DESCARTES

RED educativa digital descartes

CREACIÓN DE RECURSOS Y JUEGOS INTERACTIVOS CON DESCARTES



Jesús M. Muñoz Calle

iCartesiLibri

Creación de recursos y juegos interactivos con Descartes

Jesús M. Muñoz Calle
Doctor en Ciencias Químicas
Creador y coordinador del Proyecto AJDA



Fondo Editorial RED Descartes

Córdoba (España)
2026

Título de la obra:
Creación de recursos y juegos interactivos con Descartes

Autor:
Jesús M. Muñoz Calle
Doctor en Ciencias Químicas
Creador y coordinador del Proyecto AJDA

Código JavaScript para el libro: Joel Espinosa Longi, IMATE, UNAM.
Recursos interactivos: DescartesJS
Fuentes: Lato y UbuntuMono

Red Educativa Digital Descartes
Córdoba (España)
descartes@proyectodescartes.org
https://proyectodescartes.org

Proyecto iCartesiLibri
https://proyectodescartes.org/iCartesiLibri/index.htm

ISBN: 978-84-10368-47-7

Esta obra está bajo una licencia Creative Commons 4.0 internacional: Reconocimiento-No Comercial-Compartir Igual.

A mi querida madre
Sin ella no hubiera llegado a ser lo que soy

Tabla de contenido

Introducción **1**

1. Escenas interactivas de Descartes **5**

1.1 ¿Qué son las escenas interactivas de Descartes? 7

1.2 Instalación del editor de DescartesJS 10

1.3 Descripción del editor de DescartesJS 12

1.4 Interpretación de escenas 23

1.5 Editor de escenas 29

1.6 Atajos de teclado 34

1.7 Selector Escenas 35

1.8 Región exterior 39

1.9 La primera escena 40

1.10 La segunda escena 47

1.11 La tercera escena 50

1.12 Mi primer juego paso a paso. Paso 1 51

1.13 Mi primera escena paso a paso. Paso 1 56

2. Espacios, animaciones y elementos básicos **61**

2.1 Introducción 63

2.2 Coordenadas en escenas 66

2.3 Espacios en escenas 74

2.4 Coordenadas en espacios 84

2.5 Tipos de espacios 88

2.6 Espacios 2D 91

2.7 Espacios 3D 93
2.8 Espacios HTMLIFrame 96
2.9 Imágenes en escenas 101
2.10 Textos en escenas 109
2.11 Editor de textos con formato 117
2.12 Colores en textos y otros elementos 126
2.13 Animaciones en escenas 138
2.14 Movimiento en una escena 144
2.15 Mi primer juego paso a paso. Paso 2 151
2.16 Mi primera escena paso a paso. Paso 2 154

3. Controles 159
3.1 Introducción 161
3.2 Tipos de controles 162
3.3 Introducción de controles 164
3.4 Parámetros comunes a múltiples controles 167
3.5 Control tipo Pulsador 171
3.6 Control tipo Campo de texto 175
3.7 Control tipo Menú 178
3.8 Control tipo Barra 182
3.9 Control tipo Deslizador 185
3.10 Control tipo Botón 187
3.11 Control tipo Casilla de verificación 193
3.12 Control tipo Gráfico 198
3.13 Control tipo Texto 207
3.14 Control tipo Audio 210

3.15 Control tipo Vídeo 213
3.16 Mi primer juego paso a paso. Paso 3 217
3.17 Mi primera escena paso a paso. Paso 3 222

4. Programación en escenas 227
4.1 Introducción 229
4.2 Operadores generales 232
4.3 Operadores booleanos y su uso en condicionales 237
4.4 Operadores reservados 248
4.5 Funciones comunes intrínsecas de Descartes 257
4.6 Funciones propias del lenguaje de Descartes 262
4.7 Funciones 267
4.8 Variables 278
4.9 Vectores 283
4.10 Matrices 294
4.11 Biblioteca 305
4.12 Resumen sobre definiciones 309
4.13 Eventos 311
4.14 Algoritmos Inicio y Cálculos 320
4.15 Algunos aspectos sobre programación 335
4.16 Mi primer juego paso a paso. Paso 4 345
4.17 Mi primera escena paso a paso. Paso 4 350

5. Gráficos 2D 353
5.1 Introducción 355
5.2 Inclusión gráficos 2D 356
5.3 Parámetros comunes a los gráficos 360

5.4 Gráfico Ecuación 363
5.5 Gráfico Curva 370
5.6 Gráfico Sucesión 374
5.7 Gráfico Polígono 378
5.8 Gráfico Rectángulo 381
5.9 Gráfico Arco 384
5.10 Gráfico Punto 387
5.11 Gráfico Segmento 391
5.12 Gráfico Flecha 394
5.13 Gráfico Relleno 398
5.14 Gráfico Macro 401
5.15 Gráfico Imagen 420
5.16 Gráfico Texto 430
5.17 Familias en gráficos 2D 445
5.18 Mi primer juego paso a paso. Paso 5 454
5.19 Mi primera escena paso a paso. Paso 5 460

6. Gráficos 3D 465
6.1 Introducción 467
6.2 Inclusión gráficos 3D 469
6.3 Parámetros comunes a los gráficos 3D 473
6.4 Gráfico 3D Punto 477
6.5 Gráfico 3D Segmento 481
6.6 Gráfico 3D Polígono 484
6.7 Gráfico 3D Polígono regular 487
6.8 Gráfico 3D Triángulo 490

6.9 Gráfico 3D Curva 493
6.10 Gráfico 3D Cara 496
6.11 Gráfico 3D Superficie 499
6.12 Gráficos 3D. Cuerpos geométricos 502
6.13 Gráfico 3D Texto 518
6.14 Gráfico 3D Macro 521
6.15 Familias en gráficos 3D 533
6.16 Mi primer juego paso a paso. Paso 6 537
6.17 Mi primera escena paso a paso. Paso 6 544

7. Proyectos con escenas de Descartes 549

7.1 Introducción 551
7.2 Proyecto enfocado hacia la realización de un juego didáctico. Parte de diseño web 552
7.3 Ejemplo de web basada en un juego 555
7.3.1 Diseño y realización de la escena del juego 555
7.3.2 Índice 566
7.3.3 Introducción-presentación del juego 567
7.3.4 Instrucciones 568
7.3.5 Documentación complementaria 569
7.3.6 Integración del juego didáctico en la web 571
7.3.7 Otras posibles aportaciones o variaciones 572
7.3.8 Web del juego completa 573
7.4 Proyecto enfocado hacia la realización de un material didáctico. Parte de diseño web 579
7.5 Ejemplo de web basada en un material didáctico 582

7.5.1 Diseño y realización de las escenas de un material didáctico 582
7.5.2 Introducción-presentación 589
7.5.3 Índice 590
7.5.4 Contenidos 591
7.5.5 Actividades 591
7.5.6 Integración de escenas en la web 592
7.5.7 Web de factores de conversión completa 594
7.6 Publicación de escenas 598
7.7 Recomendaciones para la puesta en práctica 599
7.8 Ideas y ejemplos sobre proyectos con escenas de Descartes 601

Referencias 605

Introducción

Los recursos y los juegos didácticos interactivos se han consolidado como herramientas eficaces para apoyar el aprendizaje y el desarrollo de competencias en el alumnado. El Proyecto Aplicación de Juegos Didácticos en el Aula (AJDA) [1], integrado en la Red Educativa Digital Descartes [2], pone a disposición de la comunidad educativa una amplia colección de juegos didácticos interactivos y materiales relacionados, ya elaborados y listos para su uso en el aula. Estos recursos se describen con mayor detalle en la obra "Juegos didácticos. Proyecto AJDA" [3] y también pueden utilizarse a través del libro digital interactivo "Colección de juegos didácticos AJDA" [4].

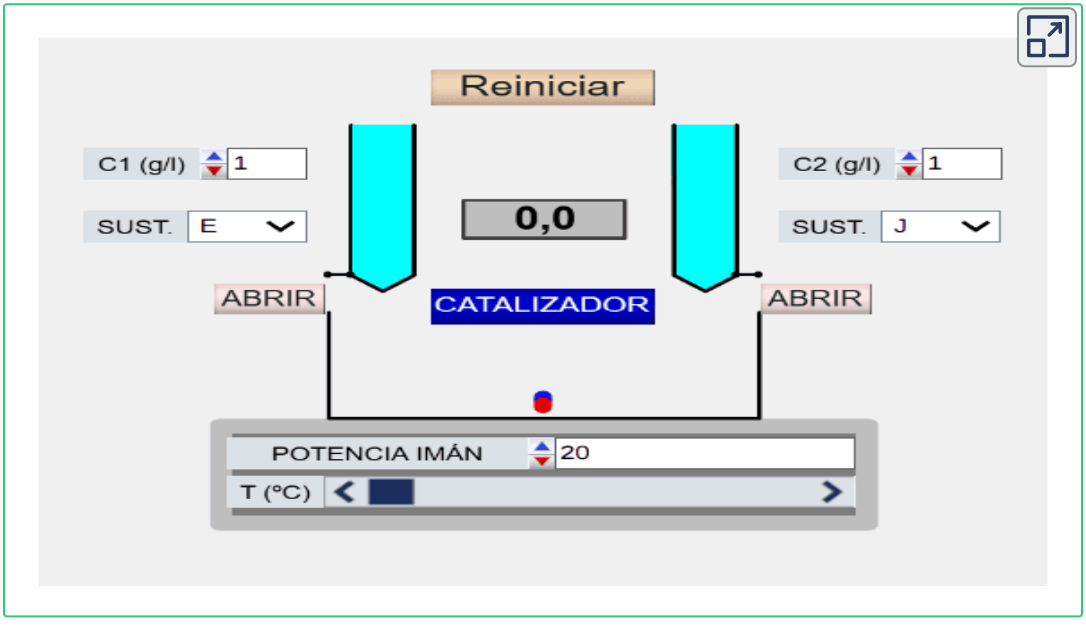
La experiencia en el aula demuestra que el uso de estos juegos, especialmente cuando se combinan con metodologías de gamificación, incrementa su eficacia y favorece una mayor implicación del alumnado. Algunos ejemplos de estas propuestas pueden encontrarse en los libros digitales interactivos "Gamificando con juegos AJDA" [5], "El Gran Torneo de Magia y Hechicería" [6] y "Los Juegos de la Alquimia" [7]. Además, estas experiencias pueden enriquecerse mediante un uso adecuado de herramientas basadas en inteligencias artificiales generativas [8], [9] y [10].

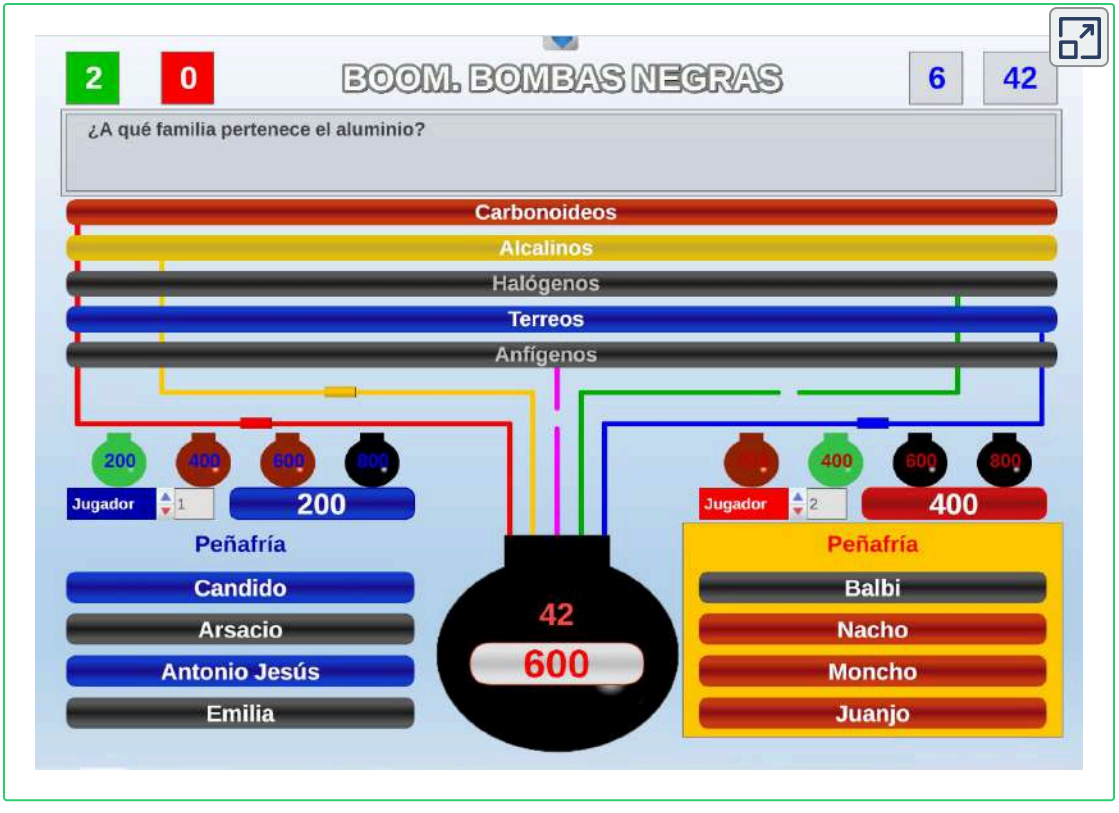
Aun así, muchos docentes prefieren diseñar sus propios juegos y materiales interactivos, o adaptar los existentes a las características de su alumnado y a su contexto educativo. Pensando en ellos, y en todas las personas interesadas en aprender a hacerlo, se presenta esta obra.

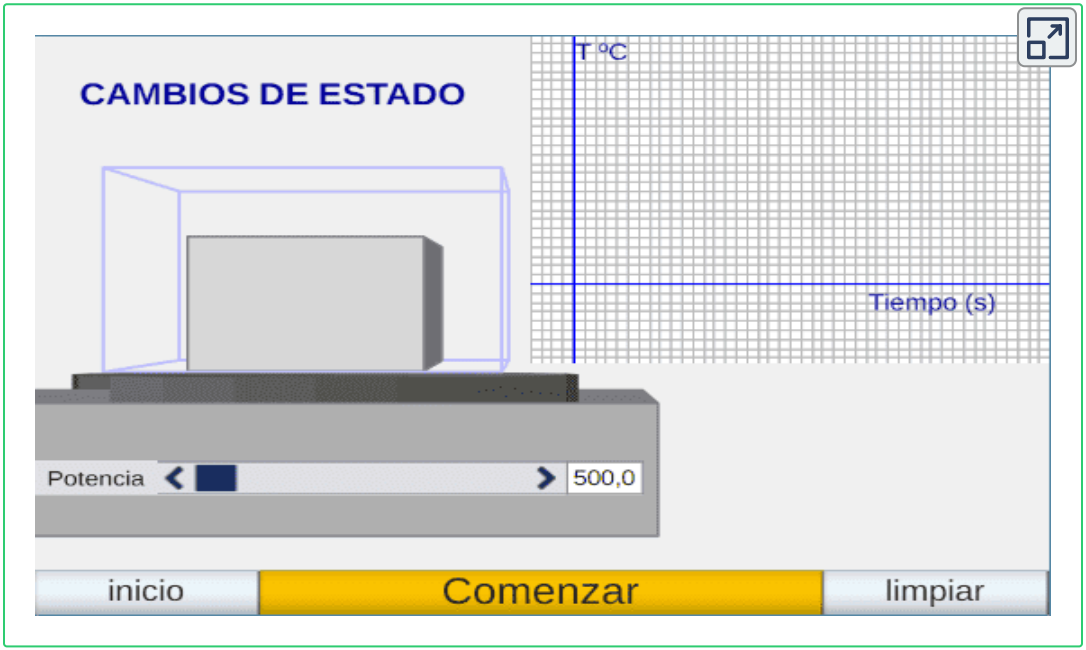
El objetivo de este texto es ofrecer una guía práctica para el diseño o la modificación de juegos didácticos y escenas animadas interactivas mediante la herramienta de autor, libre y abierta, Descartes [11], a cuya documentación técnica completa se puede acceder a través de este enlace [12].

La Red Descartes cuenta con una amplia y variada colección de recursos interactivos desarrollados con la herramienta Descartes. A través de su editorial digital, iCartesiLibri [13], se ofrecen numerosos libros interactivos de acceso gratuito que incluyen materiales de este tipo, destacando especialmente las obras [14]-[20] por su especial relación con las escenas interactivas de Descartes. El Proyecto Prometeo de la UNAM es otro portal destacado que reúne recursos similares [21]. Además, existen numerosos subproyectos de la Red Descartes que ponen a disposición del profesorado una gran diversidad de escenas interactivas de distinta naturaleza [22].

Estos materiales interactivos pueden desarrollarse para cualquier materia y nivel educativo. Pueden utilizarse en el aula mediante pizarras o paneles digitales, proyectores, ordenadores o dispositivos móviles, y visualizarse fácilmente a través de navegadores web. Asimismo, pueden publicarse en páginas web, blogs, plataformas LMS, repositorios educativos y otros entornos digitales. Mostramos a continuación algunos ejemplos de recursos interactivos realizados con Descartes, incluyendo un juego didáctico.




2
0
BOOM. BOMBAS NEGRAS
6
42
¿A qué familia pertenece el aluminio?
Carbonoideos
Alcalinos
Halógenos
Terreos
Anfígenos
200
400
600
800
Jugador
1
200
Peñafría
Candido
Arsacio
Antonio Jesús
Emilia
42
600
400
600
800
Jugador
2
400
Peñafría
Balbi
Nacho
Moncho
Juanjo


CAMBIOS DE ESTADO
T °C
Tiempo (s)
Potencia
500,0
inicio
Comenzar
limpiar



Capítulo 1

Escenas interactivas de Descartes

Capítulo 1
Escenas interactivas de Descartes

1.1 ¿Qué son las escenas interactivas de Descartes? 7

1.2 Instalación del editor de DescartesJS 10

1.3 Descripción del editor de DescartesJS 12

1.4 Interpretación de escenas 23

1.5 Editor de escenas 29

1.6 Atajos de teclado 34

1.7 Selector Escenas 35

1.8 Región exterior 39

1.9 La primera escena 40

1.10 La segunda escena 47

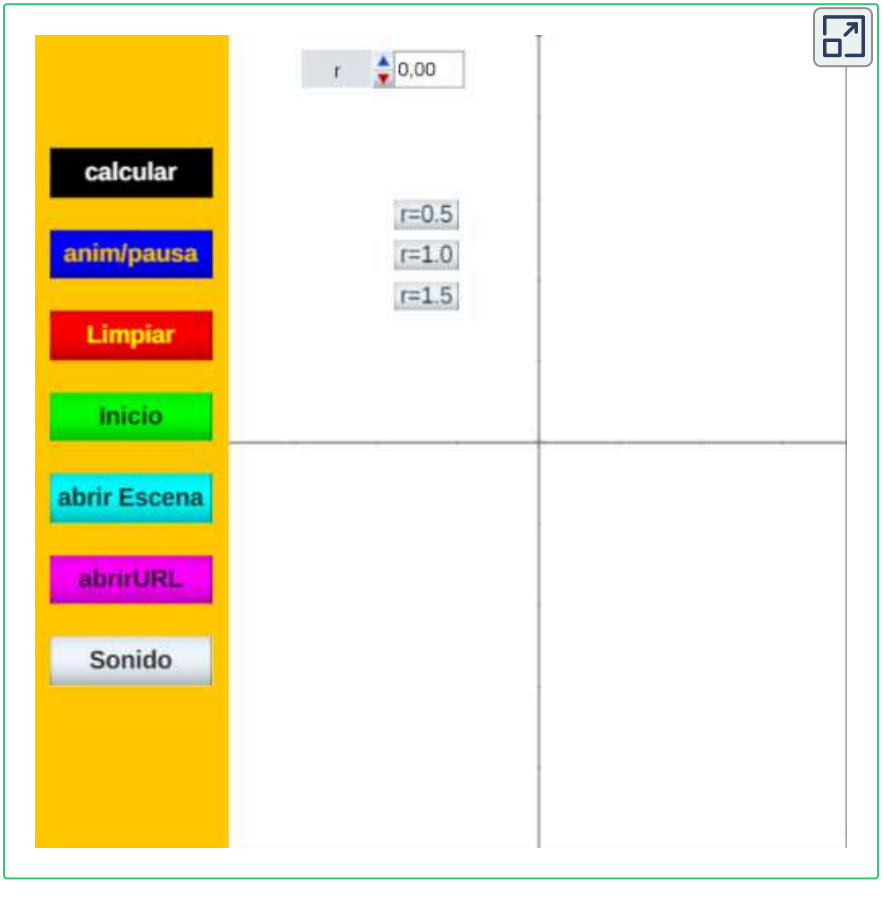
1.11 La tercera escena 50

1.12 Mi primer juego paso a paso. Paso 1 51

1.13 Mi primera escena paso a paso. Paso 1 56

1.1 ¿Qué son las escenas interactivas de Descartes?

Desde el punto de vista del usuario, una escena es un elemento interactivo integrado en una página web que se visualiza a través de un navegador. El usuario puede interactuar con ella de forma directa, sin necesidad de instalar ningún software adicional. Para que estas escenas puedan mostrarse correctamente, es necesario un **"intérprete"**, un archivo denominado **descartes-min.js**, que permite su ejecución en el navegador. Este intérprete se encarga de transformar las escenas creadas con Descartes en lenguaje JavaScript, que es el lenguaje que entienden los navegadores web.

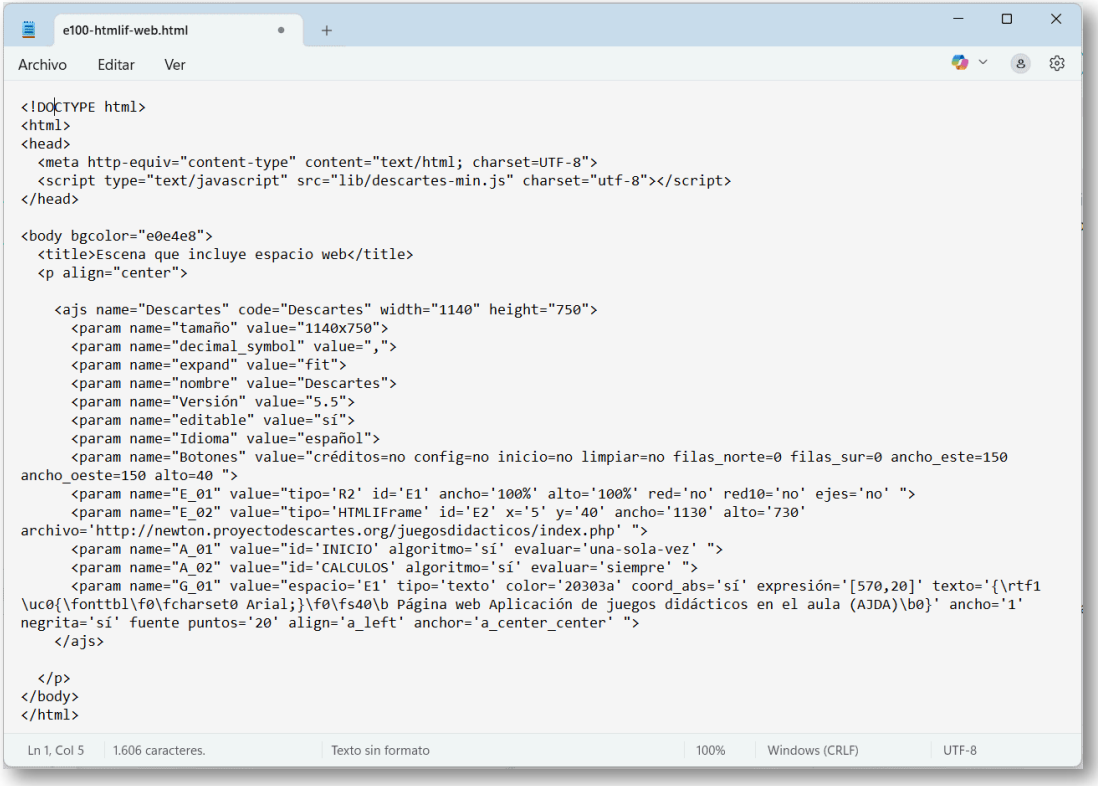


Las escenas pueden incluir espacios en dos dimensiones (2D), en tres dimensiones (3D) y espacios web, que permiten integrar cualquier página web, además de una amplia variedad de elementos interactivos: animaciones, imágenes, vídeos, textos, fórmulas matemáticas y otros recursos multimedia. El contenido de cada escena se define en función de los objetivos didácticos que se quieran alcanzar.



Desde el punto de vista del creador o programador, una escena se define mediante un conjunto de instrucciones que se integran en el código de una página web.

Tanto el código de la escena como el de la propia página pueden editarse con herramientas sencillas, como un procesador de textos, un editor de páginas web o incluso el bloc de notas. Después se deberán guardar los cambios.



Aunque es posible crear o modificar escenas interactivas de Descartes trabajando directamente sobre su código, existe una forma mucho más cómoda y accesible de hacerlo: utilizar el **editor de escenas de Descartes**. Este editor permite visualizar las escenas en tiempo real, con toda su interactividad, así como crear y modificar su contenido de forma intuitiva y, finalmente, integrarlas fácilmente en páginas web.

A lo largo de esta obra se trabajará fundamentalmente con el editor de escenas, ya que facilita el proceso de creación y modificación de escenas, incluso para usuarios sin conocimientos previos de programación.

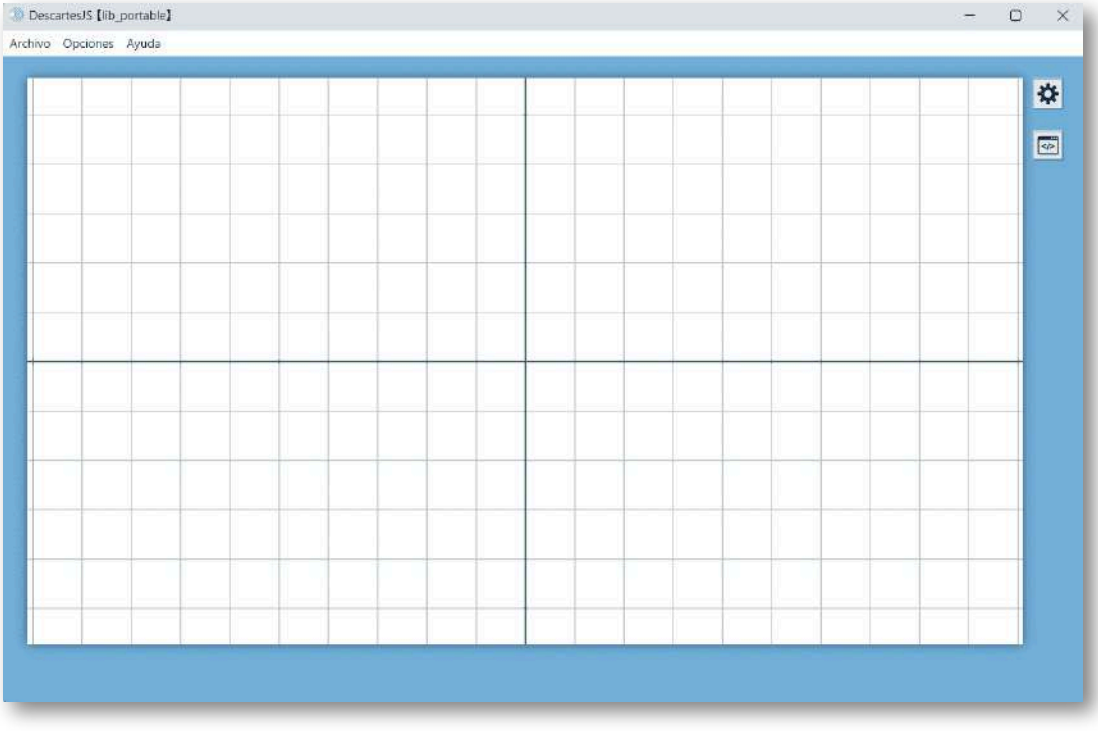
A partir de este momento, nos referiremos a las escenas interactivas de Descartes simplemente como **escenas**.

1.2 Instalación del editor de DescartesJS

El editor de escenas de DescartesJS puede utilizarse en diferentes sistemas operativos y descargarse desde el siguiente enlace, donde se encuentran disponibles las distintas versiones. Tanto el editor de Descartes como el intérprete de escenas se mantienen en constante actualización, incorporando de forma progresiva mejoras y nuevas funcionalidades. En este libro se tratarán los aspectos necesarios para crear juegos y animaciones a un nivel básico e intermedio. No obstante, quienes deseen profundizar pueden consultar y descargar la documentación completa de DescartesJS a través del enlace del "librito", situado en la esquina superior derecha de la siguiente imagen. Al pulsar sobre la imagen se abrirá la página de descarga del editor, junto con su documentación y el acceso a otras herramientas relacionadas.



Una vez descargado, basta con ejecutar el archivo para abrir el editor. En el caso de Windows, se hace pulsando dos veces con el ratón sobre el archivo ejecutable, que tiene la extensión `.exe`. Es posible que aparezca un mensaje de actualización; se recomienda aceptarlo para trabajar con la versión más reciente. A continuación, se abrirá el editor, que presentará un aspecto similar al que se muestra a continuación:



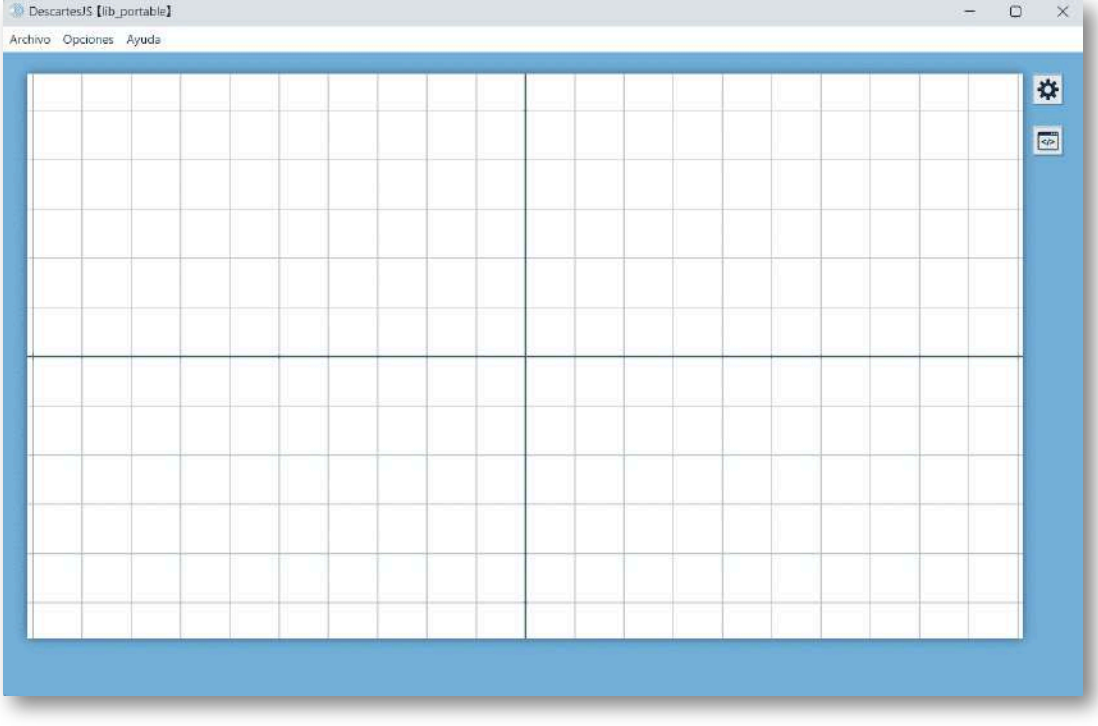
Al abrir el editor, aparece por defecto una escena en blanco sobre la que se puede comenzar a trabajar de inmediato. El editor de escenas permite, entre otras opciones, crear nuevas escenas, eliminarlas, guardarlas, recargarlas o capturar imágenes de las escenas.

Este editor se actualiza de forma periódica, incorporando mejoras y modificaciones. Por este motivo, es posible que en las distintas versiones se vayan ampliando las funcionalidades, el diseño y los recursos disponibles en la aplicación.

1.3 Descripción del editor de DescartesJS

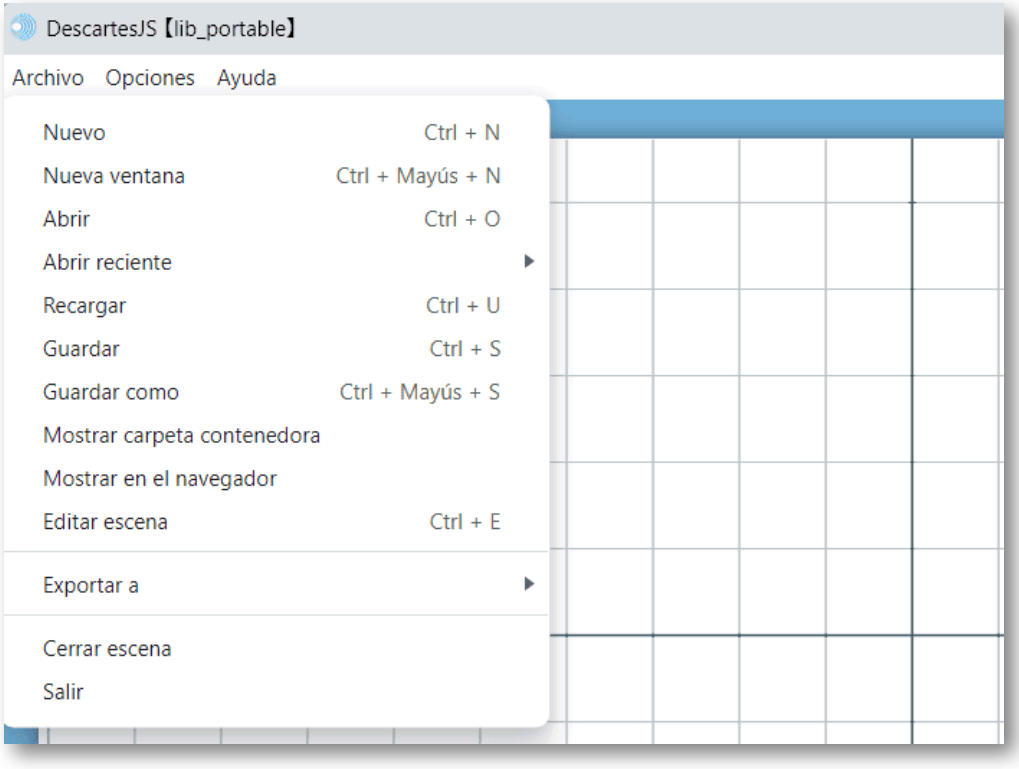
El editor de DescartesJS, al que nos referiremos simplemente como **editor de Descartes**, es la aplicación que utilizaremos principalmente para crear, modificar y guardar las escenas.

El editor dispone de un menú superior, un área de trabajo y varios iconos que facilitan el acceso a distintas funciones.



En el menú superior aparecen los apartados **Archivo**, **Opciones** y **Ayuda**. Algunos elementos del menú cuentan con atajos de teclado para un acceso más ágil. Los atajos correspondientes aparecen en un texto en gris claro a la derecha del elemento del menú en cuestión y dependen del sistema operativo.

Archivo. Este menú ofrece diversas opciones y submenús que permiten gestionar documentos y escenas dentro de Descartes.

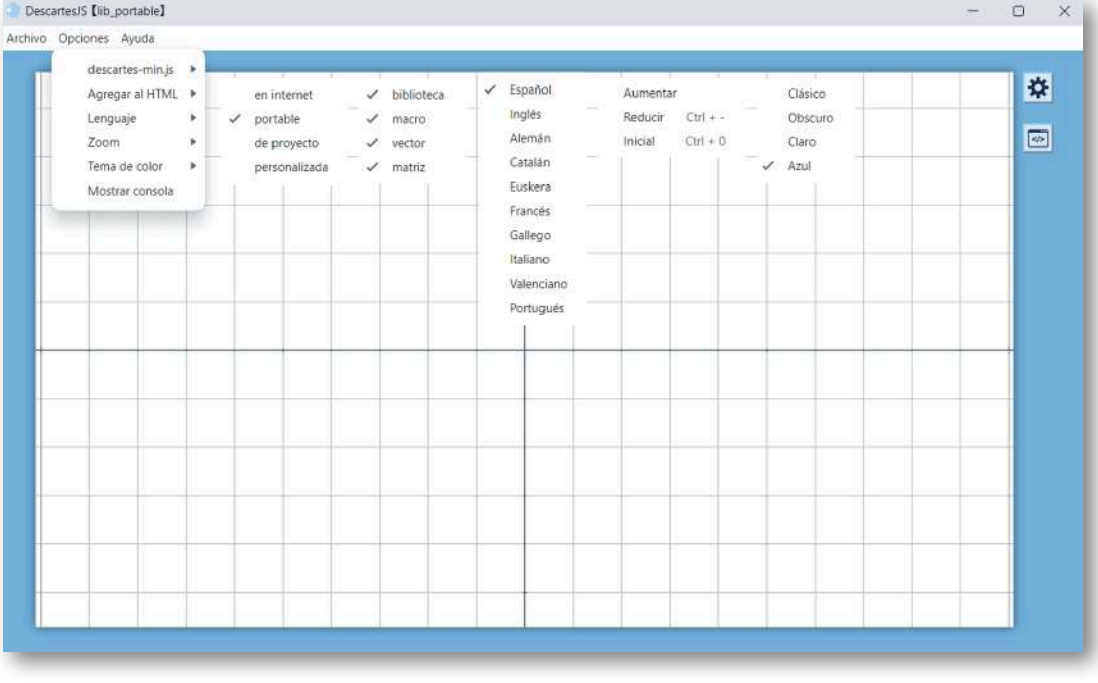


- **Nuevo:** Crea un documento de Descartes en blanco.
- **Nueva ventana:** Lanza una nueva instancia de Descartes cuando ya hay una abierta. Evita tener que lanzarlo desde la barra de tareas. Cuando se tiene una ventana de Descartes abierta y se desea abrir un segundo archivo existente, esta opción permite abrirlo en una nueva ventana.
- **Abrir:** Permite abrir un documento HTML creado en Descartes. Es posible también abrir un archivo HTML de DescartesJS arrastrándolo y soltando sobre el icono de acceso directo de DescartesJS creado en el escritorio al instalar el programa. Alternativamente, se puede abrir un HTML de Descartes mostrando el menú contextual del archivo y expandiendo el submenú **Abrir con**. En esta lista aparece una opción **nwjs** con el icono de Descartes. Al pulsar en ella, el archivo será abierto automáticamente en Descartes.

- **Abrir reciente:** Despliega un menú con los archivos usados más recientemente, incluyendo sus carpetas contenedoras, para facilitar el acceso sin buscarlos manualmente. Incluye una opción **Vaciar menú** para limpiar la lista.
- **Recargar:** Recarga el interactivo con los cambios más recientes.
- **Guardar:** Si el archivo HTML no había sido guardado antes, se abre un diálogo para seleccionar nombre y ruta; de lo contrario, se sobrescribe el archivo con los cambios. Guardar periódicamente evita pérdidas en caso de fallo de la herramienta.
- **Guardar como:** Abre un diálogo para seleccionar ruta y nombre de un nuevo archivo HTML que contenga la última versión del documento.
- **Mostrar carpeta contenedora:** Abre la carpeta donde se encuentra el archivo de la escena actualmente en edición.
- **Mostrar en el navegador:** Abre la escena en el navegador predeterminado del usuario.
- **Exportar a:** Contiene varias opciones que permiten guardar el archivo en diferentes formatos:
    - **Macro de Descartes:** Guarda contenido parcial del documento (selectores Definiciones, Programa y Gráficos) como un archivo de texto (.txt), para poder importarlo en otras escenas.
    - **Biblioteca de Descartes:** Guarda el contenido del selector **Definiciones** como un archivo de texto (.txt) que puede usarse en otras escenas.
    - **PNG:** Guarda la visualización como imagen .png.
    - **JPG:** Guarda la visualización como imagen .jpg.
    - **SVG:** Guarda la visualización como imagen .svg.
    - **PDF:** Guarda la visualización como archivo .pdf.

- **Cerrar escena:** Cierra el documento actual de Descartes. Si hay cambios no guardados, se muestra un diálogo de confirmación para evitar pérdida de datos.
- **Salir:** Cierra la aplicación DescartesJS. Si hay cambios sin guardar, se presenta un diálogo de confirmación. Si se cierra DescartesJS cerrando la ventana directamente, también se mostrará el diálogo de confirmación con opción de guardar; si se selecciona **Cancelar**, el programa no se cerrará y no se perderán los cambios.

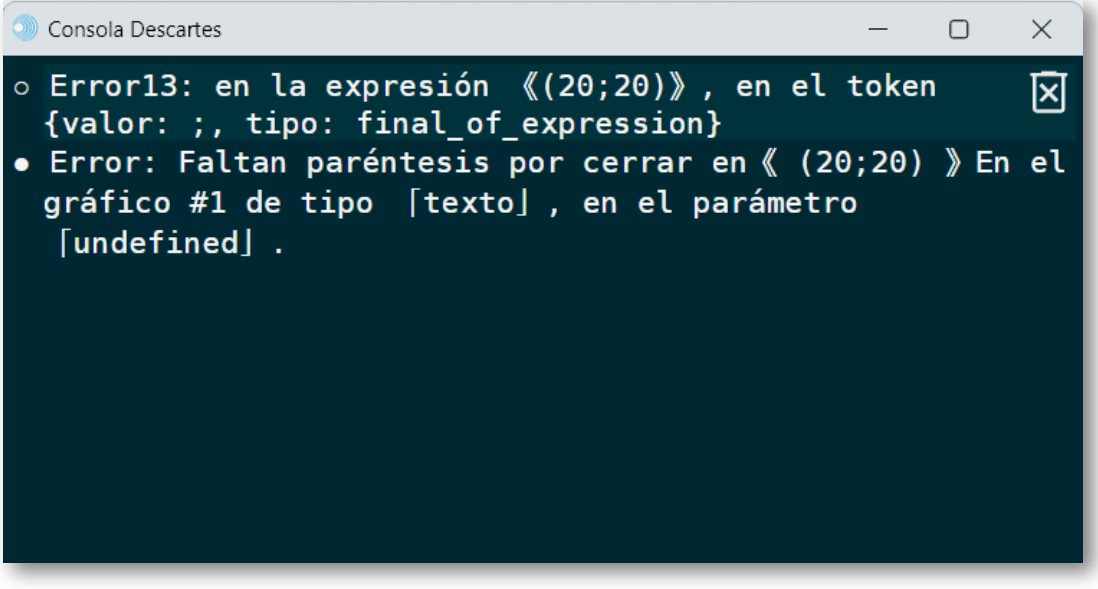
Opciones. Este menú contiene cuatro opciones: la primera gestiona la localización desde la cual se carga la biblioteca (**descartes-min.js**), la segunda configura el lenguaje mostrado en el editor, la tercera ajusta el tema de color del editor y la cuarta abre la consola de Descartes.



A continuación, desglosamos las características de los diferentes elementos de este menú:

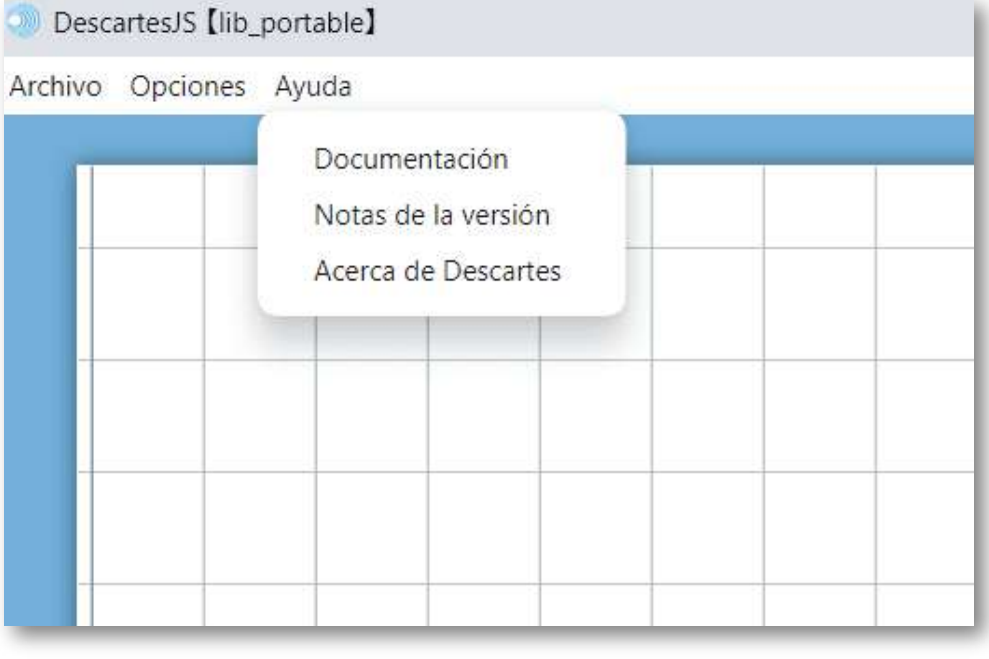
- **descartes-min.js:** Al seleccionar este menú se despliega un submenú con opciones relacionadas con el intérprete de Descartes:
    - **En Internet:** Se lee el archivo descartes-min.js desde su ubicación en línea (*https://arquimedes.matem.unam.mx/Descartes5/lib/descartes-min.js*). Garantiza siempre la versión más actual, pero las escenas no funcionarán sin conexión a Internet.
    - **Portable:** Se crea una carpeta */lib* en la misma ubicación donde se guarda el archivo, con una copia local del intérprete.
    - **De proyecto:** Permite que varios archivos Descartes compartan el mismo intérprete, ubicando la carpeta */lib* un nivel por encima de las escenas agrupadas.
    - **Personalizada:** El usuario define la carpeta donde se encuentra el intérprete, útil para proyectos con archivos distribuidos en distintos niveles.
- **Agregar al HTML:** Muestra un submenú con tres opciones activadas por defecto:
    - **Biblioteca:** Incluye el contenido de la biblioteca en un bloque dentro del archivo HTML.
    - **Macro:** Guarda la información del macro dentro de un bloque al guardar la escena.
    - **Vector:** Incluye los vectores definidos en un bloque, permitiendo su lectura incluso si el archivo de texto no está presente.
- **Lenguaje:** Permite seleccionar el idioma de los selectores y controles: español, inglés, alemán, catalán, euskera, gallego, valenciano y portugués. No se descarta que en el futuro se puedan añadir nuevos idiomas.

- **Zoom:** Controla el acercamiento al área de trabajo en el editor:
  - **Aumentar:** Incrementa ligeramente el tamaño.
  - **Disminuir:** Reduce ligeramente el tamaño.
  - **Inicial:** Regresa al tamaño original.
- **Tema de color:** Selecciona el color de fondo del editor: clásico (verde grisáceo), oscuro, claro o azul.
- **Mostrar consola:** Abre la Consola Descartes para mostrar errores de parseo y valores de variables mediante *_Print_()* o *_Trace_()*. La consola también permite visualizar errores internos en la programación (por ejemplo, divisiones por cero, raíces de argumentos negativos, y otros). Incluye un botón para borrar la consola. Cuando la cantidad de renglones en la consola supera el tamaño de la misma, aparece una barra de desplazamiento vertical para poder navegar por ella. Siempre que se agregan renglones al final la consola, la barra se desplaza al final de la misma para mostrar siempre la última línea agregada. Es útil mantenerla abierta durante programación compleja para detectar errores internos.



Ayuda. Incluye apartados de información sobre DescartesJS.

- **Documentación:** Abre la documentación desde el sitio web correspondiente en el navegador predeterminado.
- **Notas de la versión:** Muestra una ventana con el número de la última versión y un listado de cambios y mejoras. Incluye un hipervínculo *Versiones anteriores* para consultar los cambios históricos de todas las versiones previas.
- **Acerca del editor Descartes:** Despliega una ventana con el logo de DescartesJS, la versión del editor y de la biblioteca, así como los créditos de la herramienta.

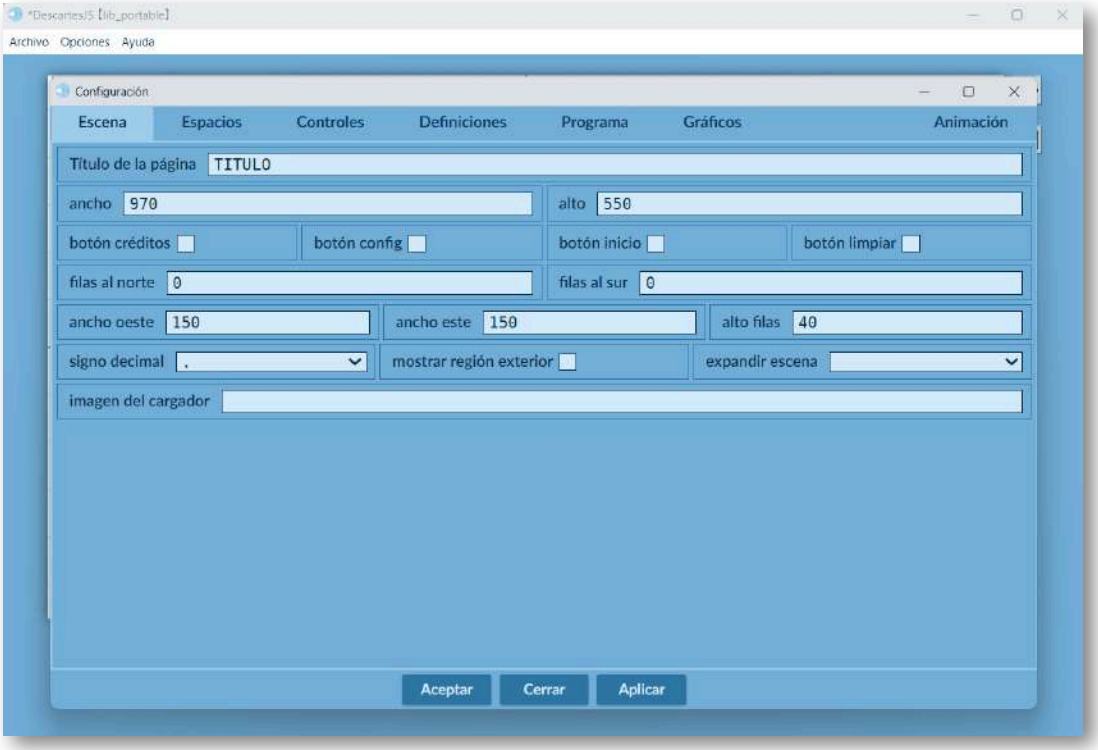


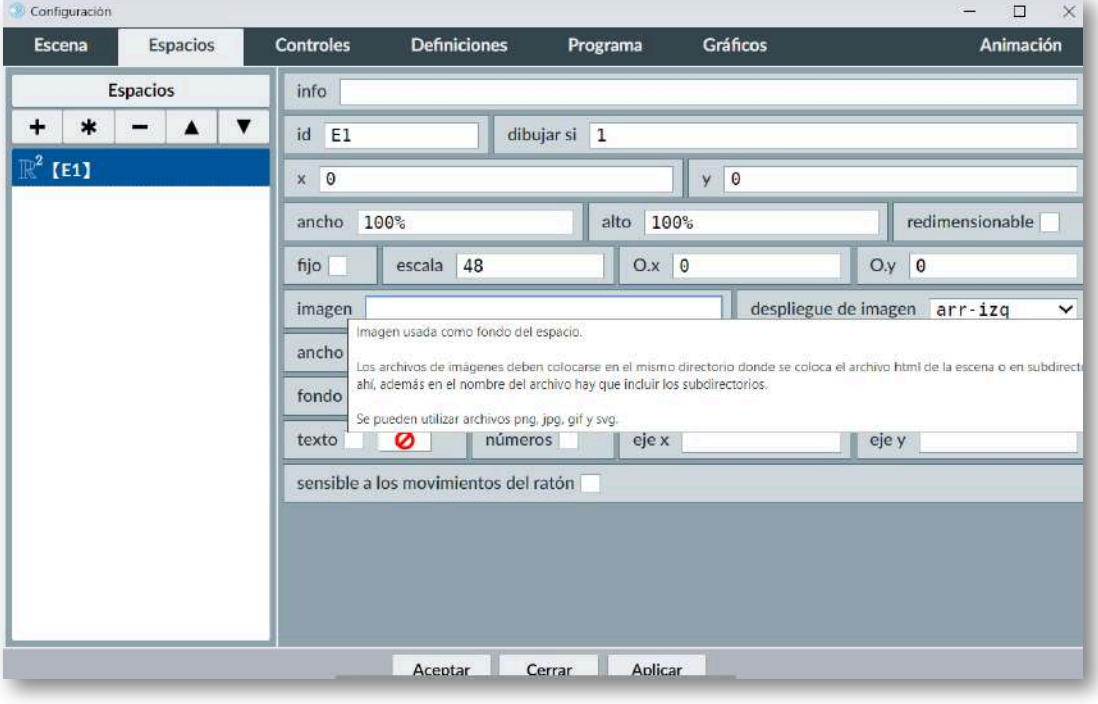
Debajo del menú superior se encuentra el área de trabajo, que ocupa la mayor parte del espacio del editor. En ella se visualizan las escenas de forma gráfica, tal y como se mostrarán en un navegador web. En este espacio podemos observar cómo la escena se va creando y evolucionando a medida que trabajamos en ella. De forma inicial y por defecto, el editor carga una escena que consta únicamente de un espacio 2D en blanco, con una rejilla de ejes de coordenadas.



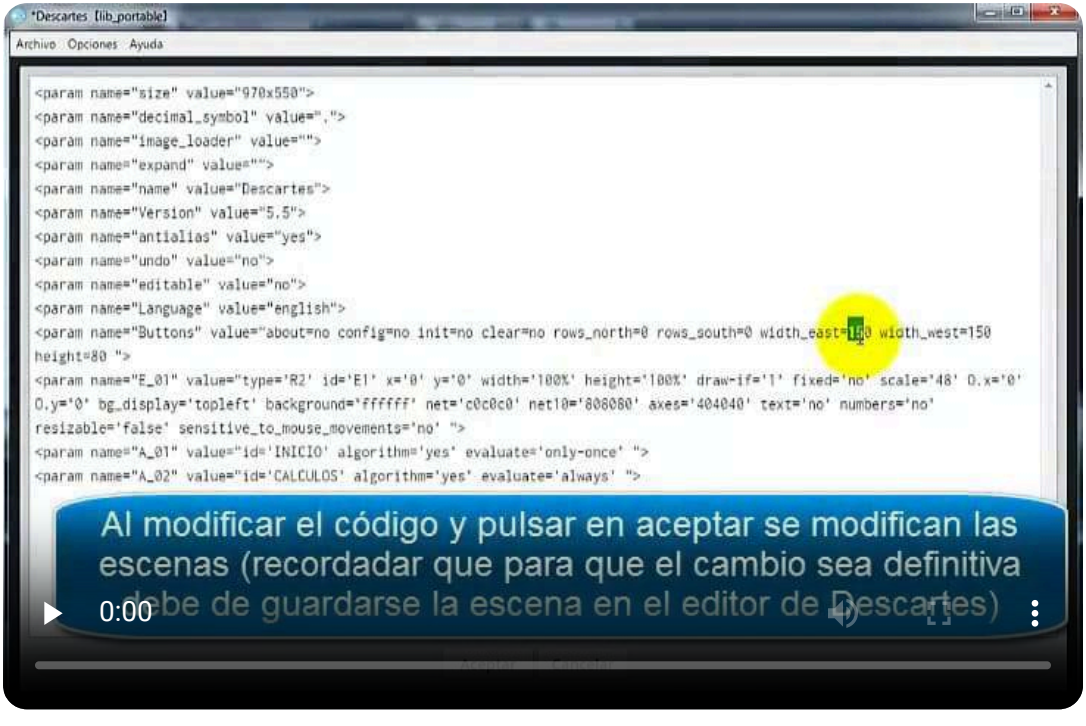
A la izquierda de la zona de trabajo, en su zona superior, hay un icono con forma de rueda dentada que abre el **editor de configuración de la escena**, al que nos referiremos simplemente como **editor de la escena** o **editor de escenas**. Es importante no confundir este editor con el editor de Descartes (que es el que estamos tratando en este apartado). El editor de la escena es la herramienta que permite crear, configurar y modificar las escenas, y será el principal objeto de estudio de esta obra.

El editor contiene diferentes menús y parámetros. Si se mantiene el cursor del ratón sobre el nombre de cualquier campo del editor durante más de un segundo, aparecerá un mensaje emergente o tooltip que explica la función y el funcionamiento de ese campo en el idioma seleccionado. Al mover el ratón, el mensaje desaparece automáticamente.


*Descartes/5 [lib_portable]
Archivo Opciones Ayuda
Configuración
Escena Espacios Controles Definiciones Programa Gráficos Animación
Título de la página TITULO
ancho 970
alto 550
botón créditos
botón config
botón inicio
botón limpiar
filas al norte 0
filas al sur 0
ancho oeste 150
ancho este 150
alto filas 40
signo decimal .
mostrar región exterior
expandir escena
imagen del cargador
Aceptar Cerrar Aplicar


Configuración
Escena Espacios Controles Definiciones Programa Gráficos Animación
Espacios
+ * − ▲ ▼
ℝ² [E1]
info
id E1
dibujar si 1
x 0
y 0
ancho 100%
alto 100%
redimensionable
fijo
escala 48
O.x 0
O.y 0
imagen
despliegue de imagen arr-izq
Imagen usada como fondo del espacio.
Los archivos de imágenes deben colocarse en el mismo directorio donde se coloca el archivo html de la escena o en subdirect
ahí, además en el nombre del archivo hay que incluir los subdirectorios.
Se pueden utilizar archivos png, jpg, gif y svg.
ancho
fondo
texto
números
eje x
eje y
sensible a los movimientos del ratón
Aceptar Cerrar Aplicar

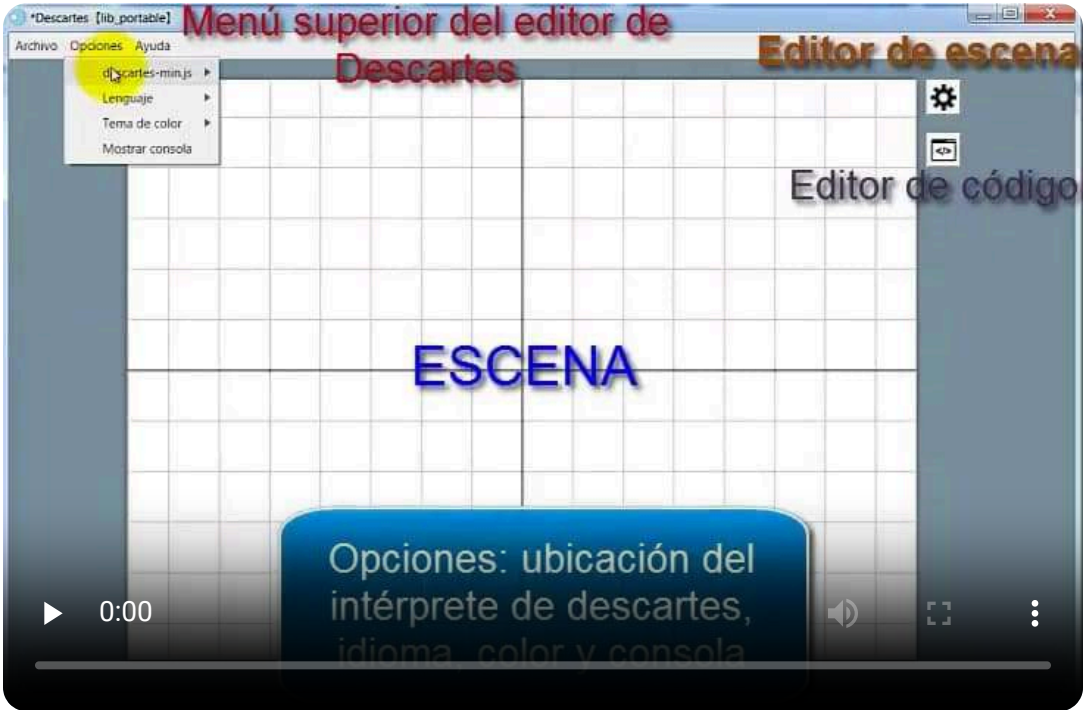
Debajo del control de acceso al editor de escenas se encuentra el icono </> que permite abrir el código de la escena para consultarlo o modificarlo, de forma similar a como puede hacerse con un editor de textos. Tras realizar cualquier cambio en el código, es necesario pulsar la opción **Aceptar** para que las modificaciones se guarden correctamente. Esta es la forma de intervenir directamente sobre el código de programación de una escena.



En este libro no se abordará la programación directa del código de DescartesJS, ya que dicho código es generado de forma automática por el editor de escenas. No obstante, es importante tener en cuenta que el código que comienza con el parámetro `ajs` y finaliza con `/ajs` es el que debe insertarse en una página web para incluir la escena. Además, para que la escena funcione correctamente, es necesario que estén disponibles todos los elementos que esta utilice, como imágenes, audios, vídeos u otros recursos.

Los usuarios más avanzados que conozcan el funcionamiento del código de Descartes pueden intervenir directamente sobre él. Cada línea de código corresponde a una instrucción de la escena y puede utilizarse, por ejemplo, para incluir una imagen, un texto, un espacio en 2D o 3D, un control, un gráfico, un algoritmo, una función u otros elementos, indicando en cada caso sus parámetros de configuración y atributos.

En el siguiente vídeo se realiza un breve recorrido por los distintos elementos del **editor de Descartes**, mostrando sus principales componentes y funcionalidades.



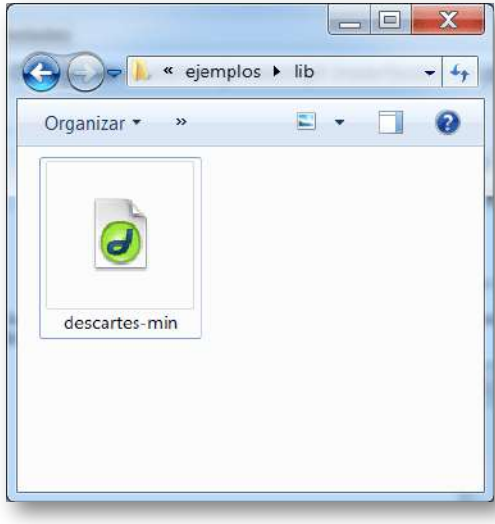
Por último, conviene recordar un aspecto importante: las escenas de Descartes se almacenan en archivos con extensión HTML. Para que los cambios realizados en una escena se guarden de forma permanente, es necesario utilizar las opciones **Guardar** o **Guardar como** del menú **Archivo** del editor de Descartes.

1.4 Interpretación de escenas

Como ya hemos comentado, para que las escenas puedan visualizarse en un navegador web es necesario contar con un archivo intérprete denominado **descartes-min.js**. Es fundamental que la escena pueda localizar a dicho intérprete, y esto puede hacerse, básicamente, de dos formas.

La primera consiste en que el archivo intérprete **descartes-min.js** acompañe al fichero de la escena. Para ello, el editor de Descartes ofrece tres opciones dentro del menú **Opciones**: **portable**, **de proyecto** y **personalizada**. En la opción **de proyecto**, el intérprete **descartes-min.js** se coloca automáticamente en una carpeta denominada **lib**, creada por el propio editor. En la imagen siguiente se muestran las páginas web ej1 y ej1_JS, que contienen escenas, junto con la carpeta **lib** que alberga el archivo **descartes-min.js**. Todas las páginas con escenas que se encuentren junto a esta carpeta serán interpretadas por el archivo **descartes-min.js** contenido en ella. Por este motivo, cuando las escenas se guardan utilizando la opción **portable**, la carpeta **lib**, que contiene el intérprete, debe acompañar siempre a dichas escenas.





Si examinamos el código de una escena de Descartes interpretada en local e insertada en una página web, veremos que la escena se encuentra delimitada entre las etiquetas `<ajs>` y `</ajs>`. Otras etiquetas básicas que se pueden encontrar en páginas web son:

`<html>` Comprende todo el documento HTML y se cierra al final con la etiqueta `</html>`.

`<title>` Define el título de la página, que es una descripción breve del contenido. Este título se muestra en la barra superior del navegador cuando se abre la página. Se cierra con `</title>`.

`<head>` Contiene el encabezado del documento. Dentro de esta etiqueta se incluye información sobre la página web que no se muestra directamente en el contenido visible de la página. Se cierra con `</head>`.

`<body>` Contiene el cuerpo del documento. Todo lo que se quiera que aparezca en la página web, incluidas las escenas de Descartes, debe estar dentro de estas etiquetas. Se cierra con `</body>`.

`<br>` Introduce un salto de línea.

`<p>` Introduce un párrafo de texto y se cierra con `</p>`.

`<span>` Permite aplicar formato a una parte del texto dentro de una línea.

`<div>` Permite crear un bloque de contenido, que puede incluir texto, imágenes o escenas, y se cierra con `</div>`.

`<b>` se utiliza para resaltar texto en **negrita**. Se cierra con `</b>`.

`<i>` se utiliza para mostrar texto en *cursiva*. Se cierra con `</i>`.

<table> Permite introducir una tabla, que ayuda a distribuir los distintos contenidos. Se cierra con </table>. Dentro de la tabla:

- <tr> y </tr>: definen las filas de la tabla.
- <td> y </td>: contienen los datos de cada celda.
- <th> y </th>: se usan para las celdas de encabezado de la tabla.

 (unordered list) se utiliza para crear listas no ordenadas, es decir, listas cuyos elementos se presentan con viñetas o bullets. Se cierra con .

 (ordered list) se utiliza para crear listas ordenadas, en las que los elementos aparecen numerados de forma automática. Se cierra con .

 (list item) define cada elemento de la lista, ya sea dentro de una lista ordenada (<ol>) o no ordenada (<ul>). Se cierra con .

```
<html>
<head><title> Ejemplo de código html</title></head>
<body>
<table>
   <tr><td>
        1ª celda de una tabla de dos filas y una  columna.
   </td></tr>
   <tr> <td>
        Esta celda está en la 2ª fila.  Una fila se abre con la
        etiqueta <tr> y  la columna con <td>. Escrito el texto
        se cierra la columna con </td > y la fila con </tr>.
   </td></tr>
</table>
</body>
</html>
```

Existen muchas más etiquetas HTML, pero su estudio no es el objetivo de esta obra. Sin embargo, es importante saber que el código de la escena debe colocarse en el lugar donde se quiera que aparezca, siempre dentro de las etiquetas <body> y </body> del documento.

Además, en la página web es necesario incluir una etiqueta que cargue el intérprete **descartes-min.js**. Esta etiqueta puede colocarse en cualquier parte del documento, aunque se recomienda incluirla entre las etiquetas <head> y </head>:

```
<script type="text/javascript" src="lib/descartes-
min.js"></script>
```

```
<html>
<head>
<title>TITULO</title>

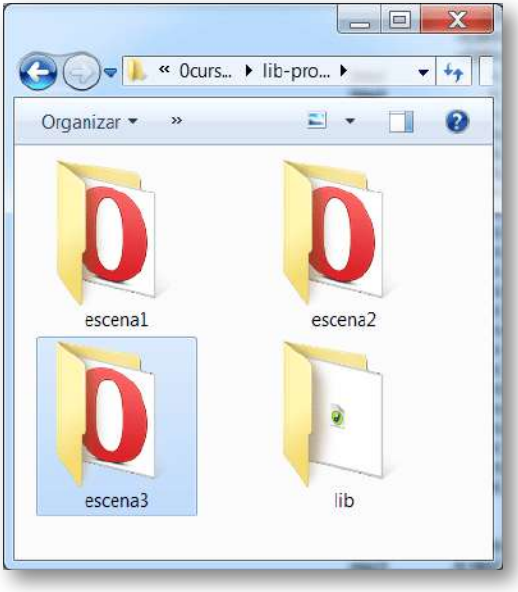
        <script type='text/javascript' src='lib/descartes-min.js'></script>
</head>
<body BGCOLOR="e0e4e8">

<p align="center">
<ajs name="Descartes" width=970 height=550
        code="Descartes.class"
        codebase="./"
        archive="lib/Descartes5.jar,lib/Descartes5_Sonido.jar,lib/Descartes5_Algebra.jar"
        MAYSCRIPT>
   <param name="tama&ntilde;o" value="970x550">
   <param name="sonido" value="s&iacute;">
   <param name="&aacute;lgebra" value="s&iacute;">
   <param name="decimal_symbol" value=",">
   <param name="antialias" value="s&iacute;">
   <param name="deshacer" value="no">
   <param name="nombre" value="Descartes">
   <param name="Versi&oacute;n" value="5.107, 2013-08-17">
   <param name="Idioma" value="espa&ntilde;ol">
   <param name="Botones" value="cr&eacute;ditos=s&iacute; config=s&iacute; inicio=s&iacute; lim
   <param name="E_01" value="tipo='R2' id='E1' despl_imagen='arr-izq' fondo='blanco' red='grisC
   <param name="C_01" value="id='n1' tipo='num&eacute;rico' interfaz='bot&oacute;n' regi&oacute
acci&oacute;n='abrir URL' par&aacute;metro='http://www.google.es'">
   <param name="A_01" value="id='INICIO' algoritmo='s&iacute;' evaluar='una-sola-vez'">
   <param name="A_02" value="id='CALCULOS' algoritmo='s&iacute;' evaluar='siempre'">
</ajs></p>

</body>
</html>
```

Cuando se utiliza la opción **de proyecto**, la carpeta **lib**, que contiene el archivo **descartes-min.js**, se sitúa un nivel por encima de la carpeta que alberga el archivo HTML de la escena.

Por ejemplo, supongamos que tenemos tres carpetas llamadas escena1, escena2 y escena3, y que cada una de ellas contiene una escena denominada ej1.html, ej2.html y ej3.html, respectivamente. Junto a estas carpetas se encuentra la carpeta **lib**, que contiene el archivo **descartes-min.js**. Este archivo será el encargado de interpretar todas las escenas ubicadas en las carpetas escena1, escena2 y escena3. De este modo, con un único archivo **descartes-min.js** en la carpeta **lib**, es posible interpretar todas las escenas contenidas en varias carpetas diferentes.





En este caso, las escenas ej1.html, ej2.html y ej3.html deben incluir la siguiente etiqueta, preferentemente situada entre las etiquetas `<head>` y `</head>`:

```
<script type='text/javascript' src='../lib/descartes-
                    min.js'></script>
```

Si se utiliza la opción **personalizada**, será necesario indicar de forma explícita la ruta en la que la escena debe localizar el intérprete **descartes-min.js**, ajustándola a la estructura de carpetas correspondiente.

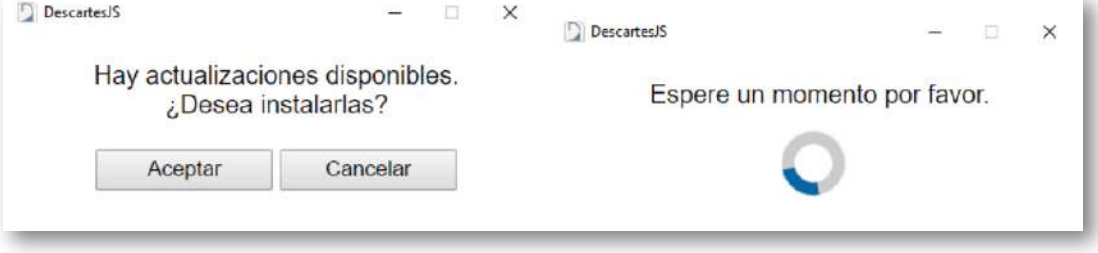
La segunda posibilidad consiste en que el intérprete **descartes-min.js** se cargue directamente desde Internet, lo que requiere disponer de conexión a la red. En este caso, el código de la escena debe contener, también de forma preferente entre `<head>` y `</head>`, la siguiente etiqueta:

```
<script type='text/javascript'
src='http://arquimedes.matem.unam.mx/Descartes5/lib/descartes-min.js'></script>
```

Es importante recordar que todas estas etiquetas son creadas e insertadas automáticamente en la página por el editor de Descartes, por lo que el usuario no necesita añadirlas manualmente, es suficiente con seleccionarla a través del menú **Opciones**.

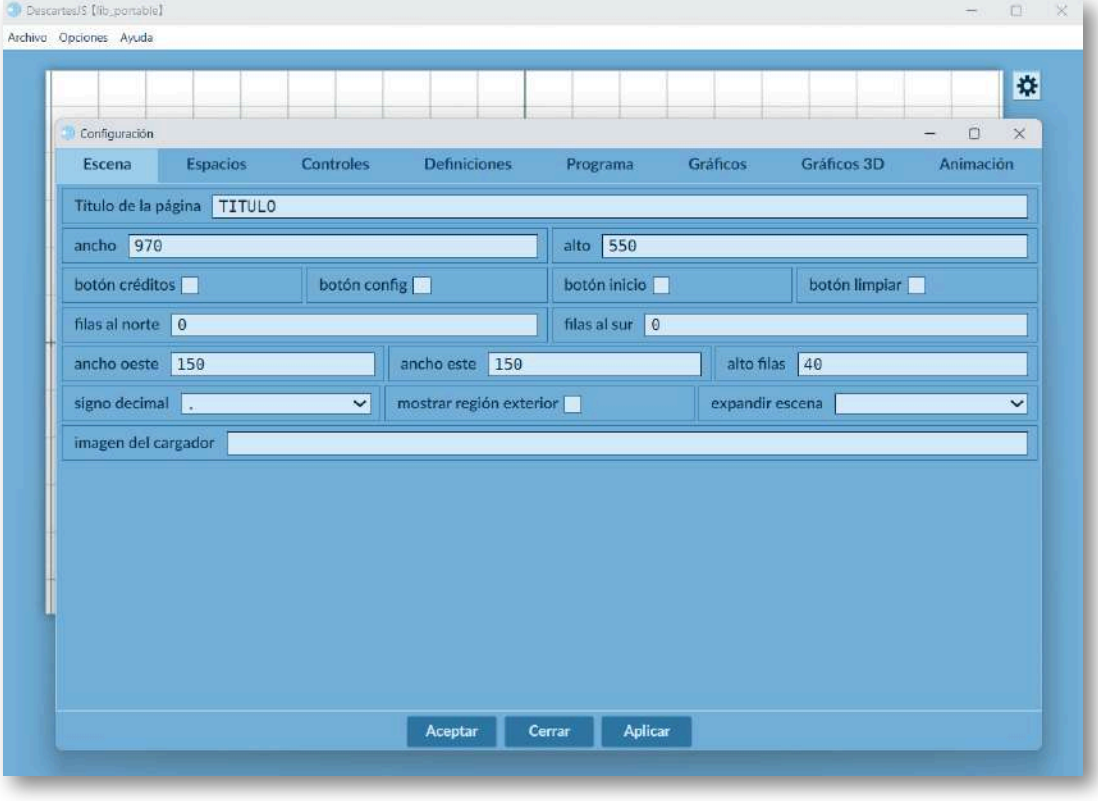
Técnicamente, el intérprete es un archivo *js* que actúa como una librería o biblioteca informática. Su función principal es traducir el lenguaje propio del código de Descartes a lenguaje JavaScript, de modo que pueda ser interpretado correctamente por los navegadores web. Por esta razón, las escenas de Descartes no pueden visualizarse ni funcionar en un navegador si no cuentan con este intérprete.

Cada vez que el editor de Descartes se abre, permite realizar una actualización automática del intérprete. El diálogo de actualización del intérprete sólo aparece si se detecta que este está desactualizado. Se recomienda siempre aceptar la actualización para que los interactivos sean visualizados correctamente.

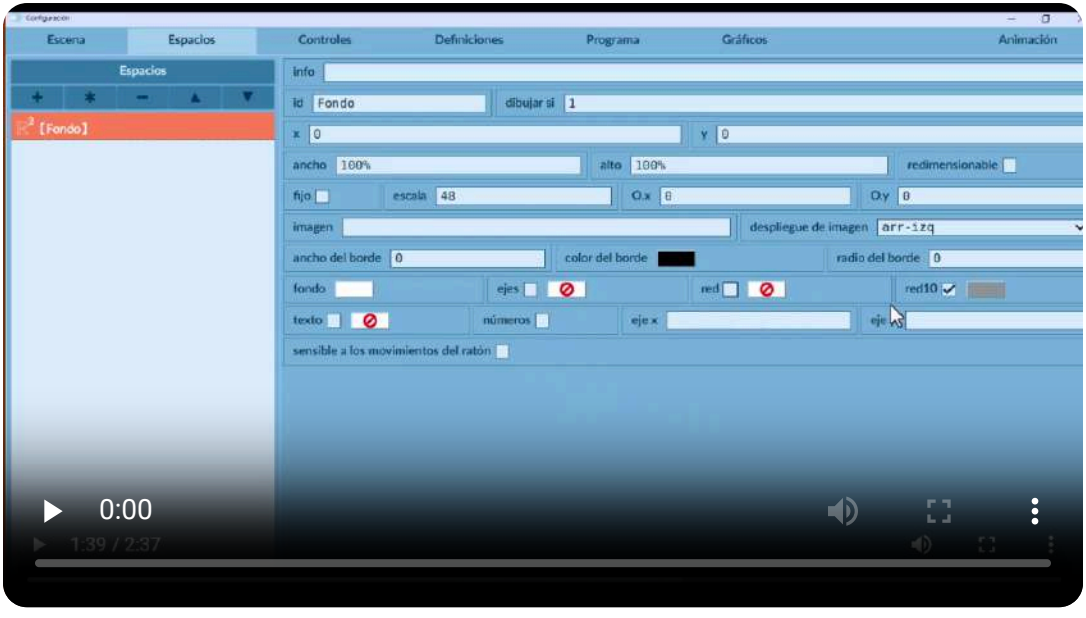


1.5 Editor de escenas

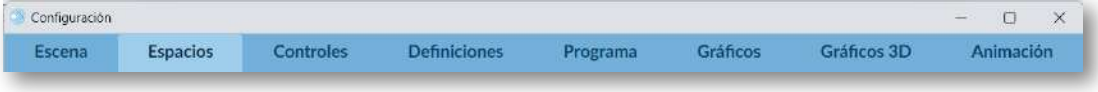
Tal y como hemos comentado, el **editor de escenas** se abre al pulsar el icono con forma de rueda dentada del editor de Descartes. Desde este espacio es posible crear y modificar cualquier escena de Descartes de manera visual y ordenada. El editor cuenta con un sistema de menús y ventanas que facilitan el trabajo y, además, al situar el ratón sobre cualquiera de sus campos durante unos instantes, aparece un mensaje emergente de ayuda que explica la función y el uso de ese elemento (tooltip).



En el siguiente vídeo se hace un recorrido por los diferentes elementos que componen el **editor de escenas**. En el mismo se presentan sus principales apartados, describiendo su función y la manera en que se integran dentro del entorno de trabajo.



El editor cuenta con un menú superior compuesto por ocho selectores, que organizan las distintas funciones de la aplicación. Al seleccionar cada uno, se muestran los contenidos y opciones correspondientes.



A continuación, se ofrece una breve descripción de los selectores del editor de escenas, aunque cada uno se abordará con mayor profundidad a lo largo de este libro.

- **Selector *Escena*:** Permite realizar cambios en la interfaz general de la escena y su interacción con el usuario, además de otras propiedades generales.
- **Selector *Espacios*:** Se utiliza para añadir, duplicar, eliminar o editar las propiedades de los espacios existentes en un interactivo. Los espacios son áreas donde se muestra material gráfico al usuario.

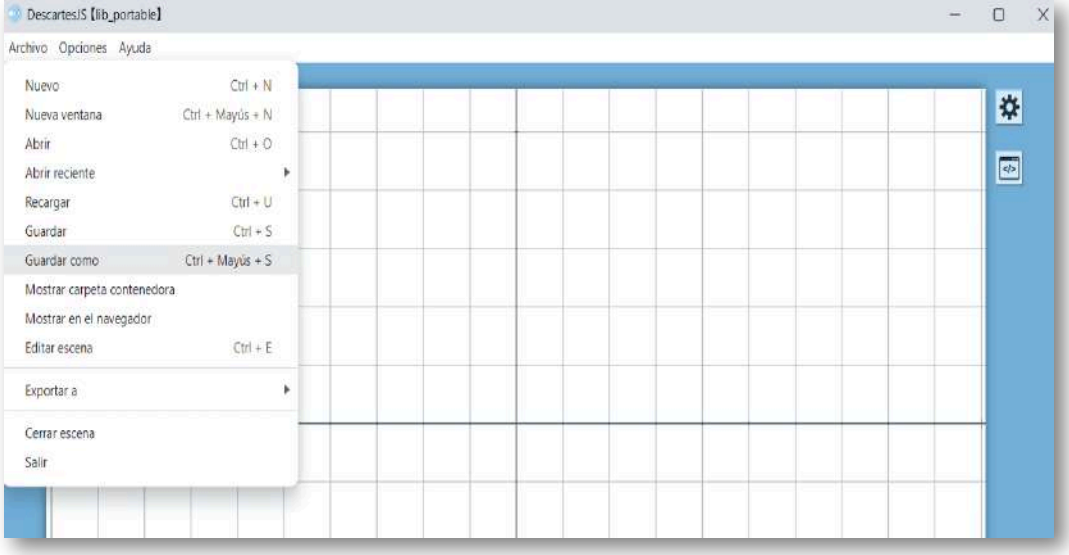
- **Selector *Controles*:** Permite añadir, duplicar, eliminar o editar las propiedades de los controles presentes en la escena. Estos son los elementos que el usuario puede manipular directamente.
- **Selector *Definiciones*:** Facilita añadir, duplicar, eliminar o editar los elementos que componen la programación del interactivo. Aquí se encuentra gran parte del "código fuente" que da funcionalidad a la escena.
- **Selector *Programa*:** Contiene los algoritmos y eventos que controlan el funcionamiento del interactivo. Incluye instrucciones de preparación inicial, así como acciones que se repiten bajo ciertas condiciones.
- **Selector *Gráficos 2D*:** Permite añadir, duplicar, eliminar o modificar las propiedades de los gráficos 2D mostrados en los espacios de la escena.
- **Selector *Gráficos 3D*:** Igual que el de *Gráficos 2D* pero para espacios en 3D. Este menú no se muestra si no hay definido ningún espacio 3D.
- **Selector *Animación*:** Se utiliza para editar las instrucciones y condiciones de las animaciones, en caso de escenas animadas.

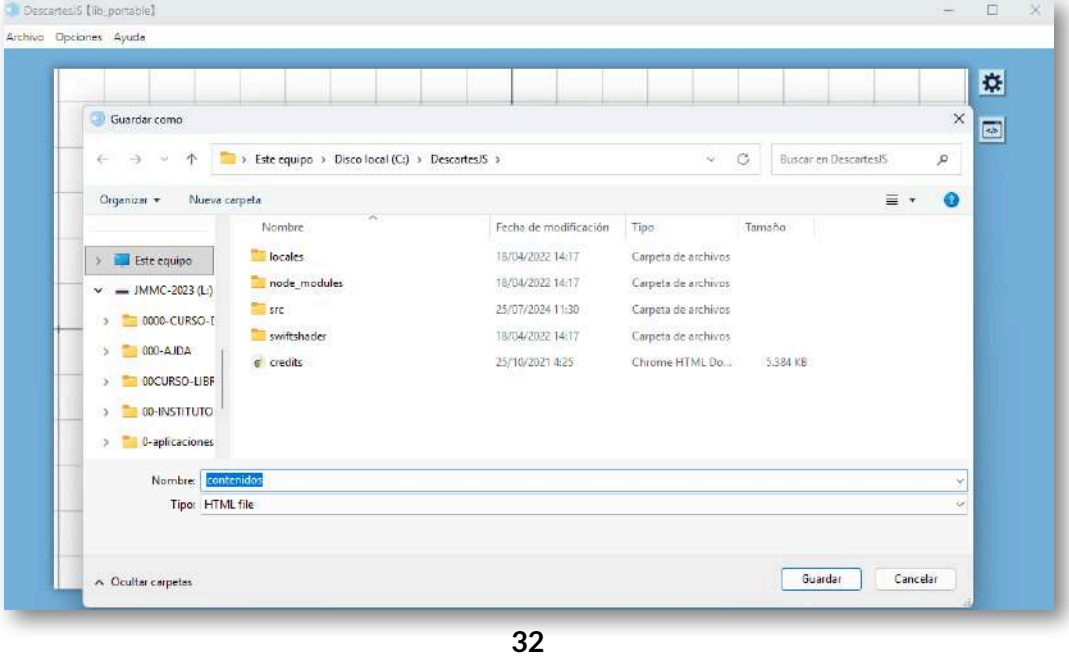
En la parte inferior del editor de escenas se encuentran siempre tres botones: **Aceptar**, **Aplicar** y **Cerrar**.



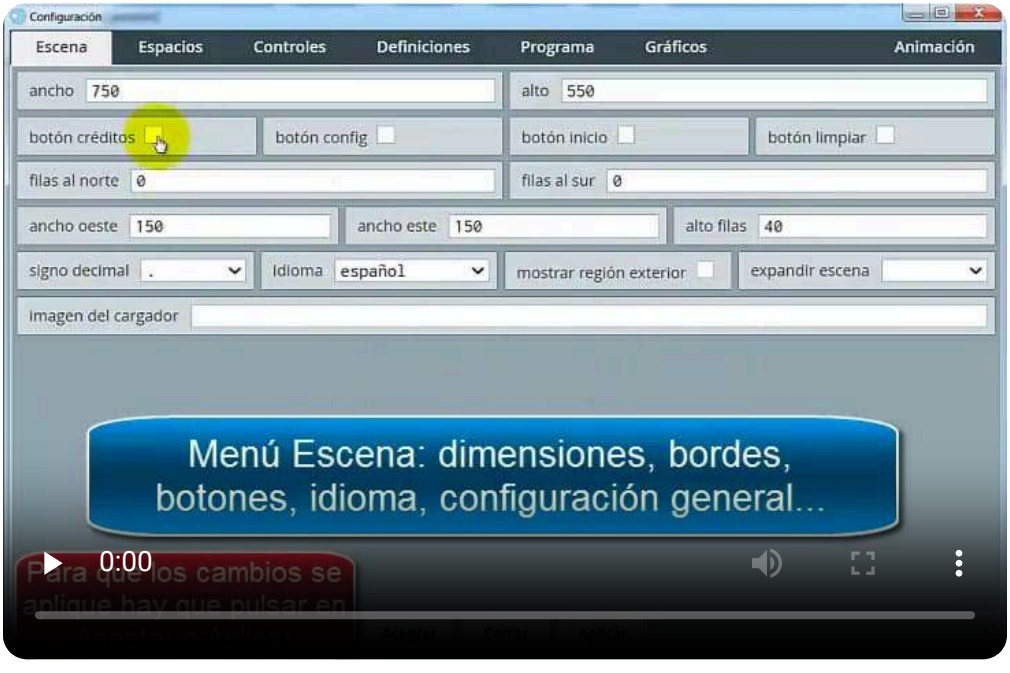
Es importante utilizarlos correctamente para que los cambios realizados se apliquen. Si se realizan modificaciones en el editor de escenas pero no se pulsa **Aceptar** o **Aplicar**, los cambios no se ejecutarán. El botón **Aceptar** aplica las modificaciones y cierra el editor de escenas; el botón **Aplicar** realiza lo mismo pero mantiene abierto el editor, permitiendo seguir trabajando; mientras que el botón **Cerrar** cierra la ventana del editor sin aplicar ningún cambio.

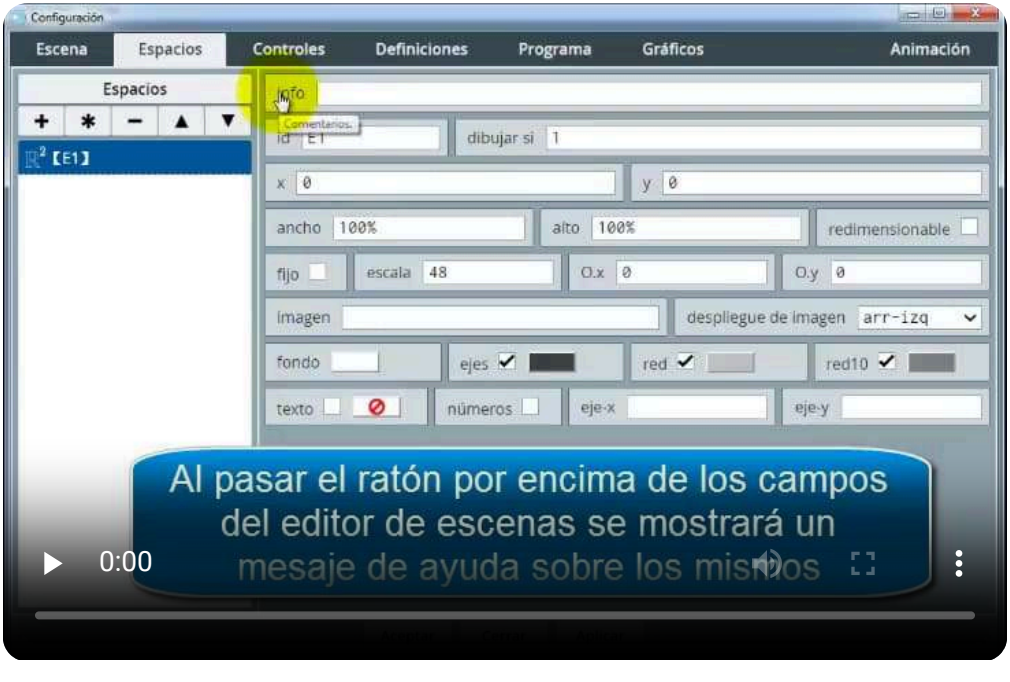
Además, hay que tener en cuenta que los cambios en las escenas no se guardan de forma definitiva hasta que se utiliza la opción **Guardar** o **Guardar como** del menú **Archivo** del editor de Descartes. Esto significa que, aunque los cambios se vean reflejados en el área de trabajo del editor de escenas, no se conservarán permanentemente hasta guardar el archivo de la escena.





En los siguientes vídeos se presenta el editor de escenas y la ayuda que proporciona mediante mensajes emergentes (tooltips).





1.6 Atajos de teclado

Atajos al editor de configuraciones y sus selectores. La tecla CONTROL se indica como **CTRL**, RETORNO como **INTRO** y retroceso como **BACKSPACE**. Para combinaciones simultáneas se usa el signo **+**.

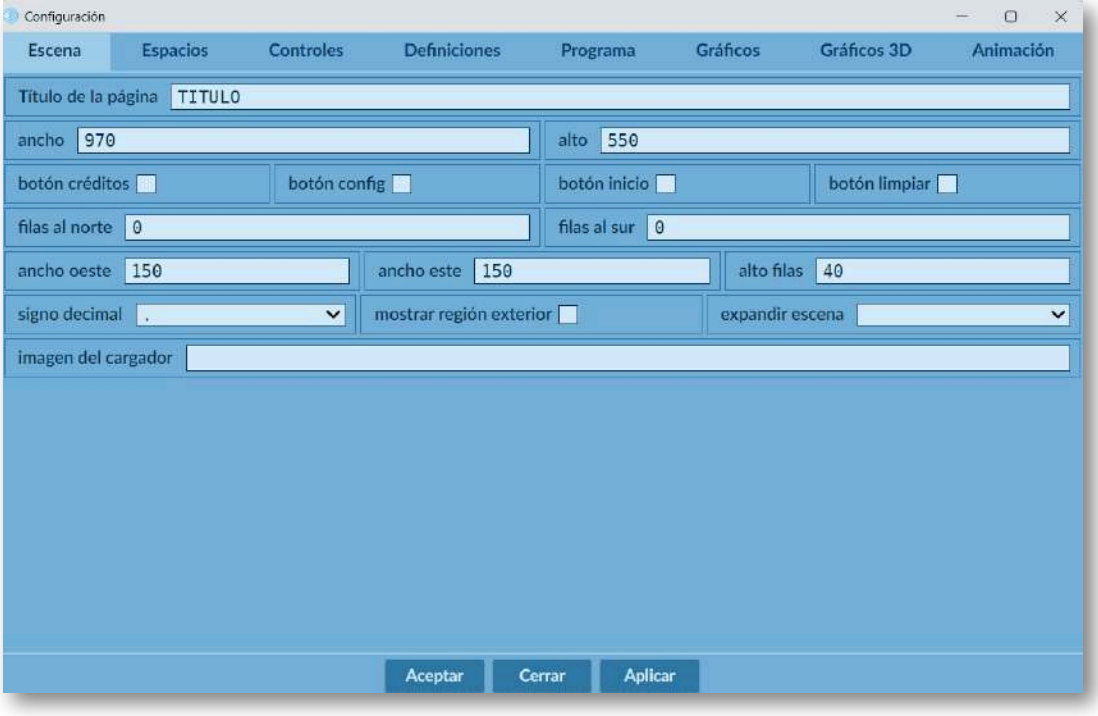
- **CTRL+E:** Abre el editor desde DescartesJS.
- **CTRL+1:** Abre el editor en el selector Escena.
- **CTRL+2:** Abre el editor en el selector Espacios.
- **CTRL+3:** Abre el editor en el selector Controles.
- **CTRL+4:** Abre el editor en el selector Definiciones.
- **CTRL+5:** Abre el editor en el selector Programa.
- **CTRL+6:** Abre el editor en el selector Gráficos 2D.
- **CTRL+7:** Abre el editor en el selector Gráficos 3D.
- **CTRL+8:** Abre el editor en el selector Animación.
- **CTRL+INTRO:** Ejecuta la acción de aplicar cambios.
- **CTRL+ALT+INTRO:** Ejecuta la acción de aceptar cambios.
- **CTRL+BACKSPACE:** Cierra el editor.
- **CTRL+SHIFT+C** (o **Command+SHIFT+C** en MacOS): Copia al portapapeles el código de configuración de la escena activa.

Atajos de navegación de elementos en el panel de lista: Los selectores Espacios, Controles, Definiciones, Programa, Gráficos y Gráficos 3D contienen un panel lateral que lista los elementos del selector. Al estar seleccionado el panel (enmarcado en azul oscuro):

- Las teclas **ARRIBA** y **ABAJO** permiten cambiar la selección al elemento superior o inferior.
- Con **CTRL+ARRIBA** o **CTRL+ABAJO** se mueve el elemento seleccionado hacia arriba o abajo en la lista.

1.7 Selector *Escenas*

El selector ***Escena*** permite controlar las propiedades más generales de la interfaz de la escena. A continuación, se presenta una descripción de los principales parámetros de este selector.

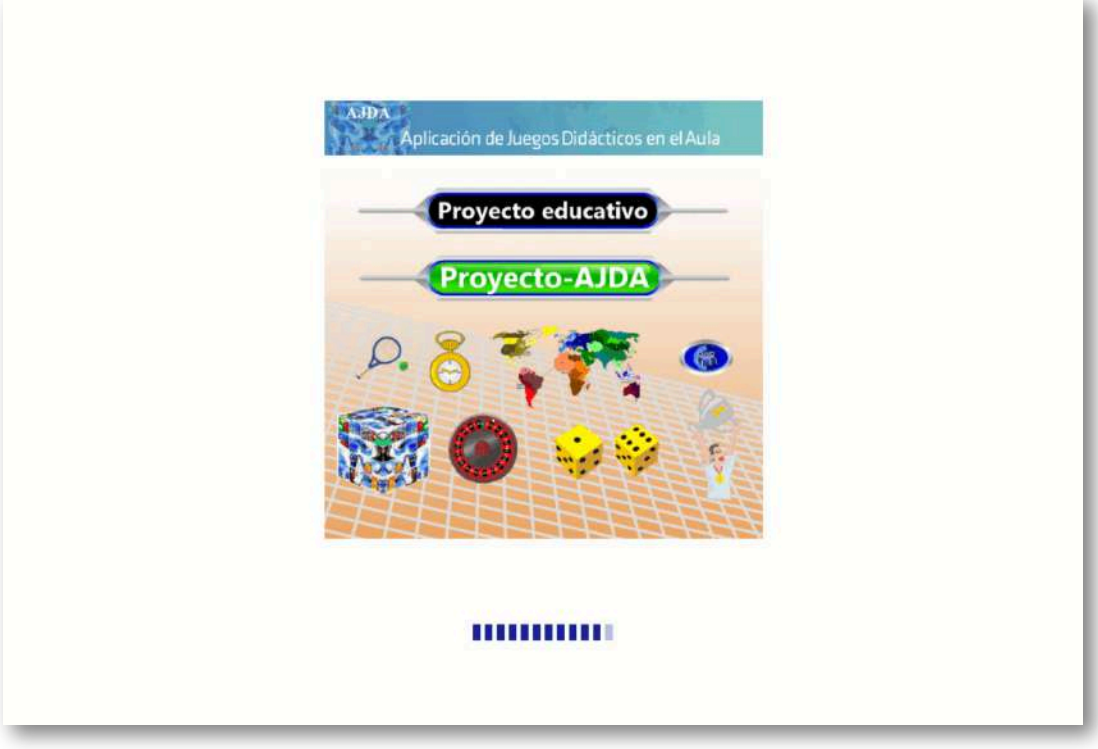


- **Título de la página:** Campo de texto donde se introduce el título del archivo HTML. Es decir, el contenido de la etiqueta `<title>...</title>` del archivo.
- **Ancho:** Campo de texto donde se indica el ancho que tendrá el interactivo, en píxeles (px).
- **Alto:** Campo de texto donde se indica la altura que tendrá el interactivo, en px.
- **Botón Créditos:** Checkbox que, al marcarlo, incluye en el interactivo un botón llamado "Créditos" que despliega los créditos y la licencia de DescartesJS en una ventana emergente.

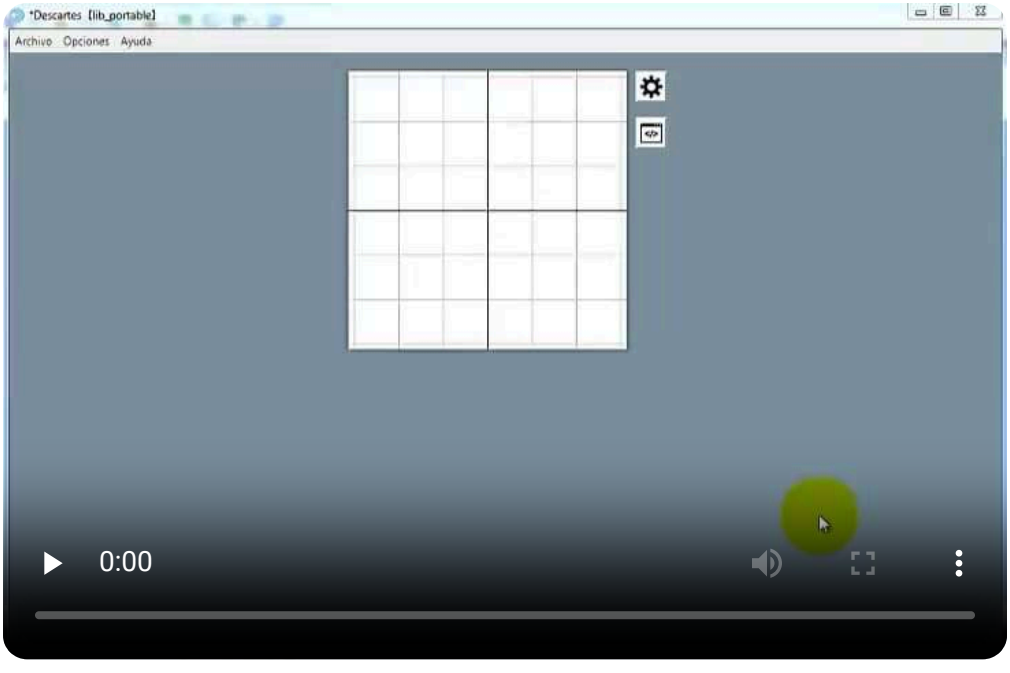
- **Botón Config:** Checkbox que añade un botón "Config". Al pulsarlo, muestra el código completo de la escena, listo para ser embebido en una página web.
- **Botón Inicio:** Checkbox que incluye un botón "Inicio". Al pulsarlo, recarga la escena desde el principio, eliminando cualquier modificación realizada por el usuario.
- **Botón Limpiar:** Checkbox que añade un botón "Limpiar", que borra cualquier rastro gráfico dejado por la interacción del usuario.
- **Filas al norte:** Campo de texto que determina cuántas filas se añadirán en la parte superior del interactivo (de color gris) para alojar controles. Los botones Créditos y Config siempre se colocan en una fila al norte, aunque no se defina aquí.
- **Filas al sur:** Campo de texto que determina cuántas filas se añadirán en la parte inferior del interactivo (de color gris) para alojar controles. Los botones Inicio y Limpiar siempre se colocan en una fila al sur.
- **Ancho oeste:** Campo de texto que determina el ancho de la columna izquierda del interactivo, mostrada solo si contiene controles.
- **Ancho este:** Campo de texto que determina el ancho de la columna derecha del interactivo, con la misma función que el ancho oeste.
- **Alto filas:** Campo de texto que define la altura, en px, de cada fila agregada por las opciones "Filas al norte" o "Filas al sur".
- **Signo decimal:** Menú que permite seleccionar si el símbolo decimal será punto (.) o coma (,).
- **Mostrar región exterior:** Checkbox que permite abrir el "espacio exterior" con clic derecho del ratón. Esta ventana emergente aloja los botones Créditos, Config, Inicio y Limpiar, así como otros controles definidos para exterior.

- **Expandir escena:** Menú que controla cómo se ajusta la escena al tamaño del navegador:
  - **Cubrir:** La escena se expande para cubrir todo el área disponible manteniendo la misma escala.
  - **Escalar:** La escena se reescala proporcionalmente para ajustarse al tamaño del navegador.
- **Imagen del cargador:** Campo de texto donde se indica la ruta de una imagen que se mostrará mientras se carga la escena en un navegador. Si se deja vacío, se muestra el logo de DescartesJS. Esta imagen no se visualiza en el editor.

A continuación, se muestra un ejemplo de imagen de carga. Esta imagen se utiliza en los juegos didácticos del Proyecto AJDA y se muestra desde que se accede al juego hasta que este se ha cargado por completo.



En los siguientes vídeos se muestra cómo configurar el tamaño y las distintas zonas de una escena.





1.8 Región exterior

El **espacio o región exterior** es una ventana emergente que puede alojar controles adicionales. Esta ventana siempre incluye los botones **Créditos**, **Config**, **Inicio** y **Limpiar**, independientemente de si están visibles en el interactivo principal. Además, si existen otros controles cuya ubicación se haya definido como "exterior", también se mostrarán en esta ventana.



El espacio exterior se abre con un clic derecho del ratón y solo funciona si está activada la opción **Mostrar región exterior** en el selector *Escena*.

El significado de los controles que aparecen por defecto en el espacio exterior es el siguiente:

- **Créditos:** Muestra la autoría, las instituciones y la licencia bajo la que se publica la aplicación.
- **Config:** Abre una ventana que muestra el código completo de la escena.
- **Inicio:** Reinicia la escena, dejándola como estaba al abrirse por primera vez.
- **Limpiar:** Borra cualquier rastro gráfico que haya quedado por interacción del usuario.
- **Cerrar:** Cierra el menú exterior.

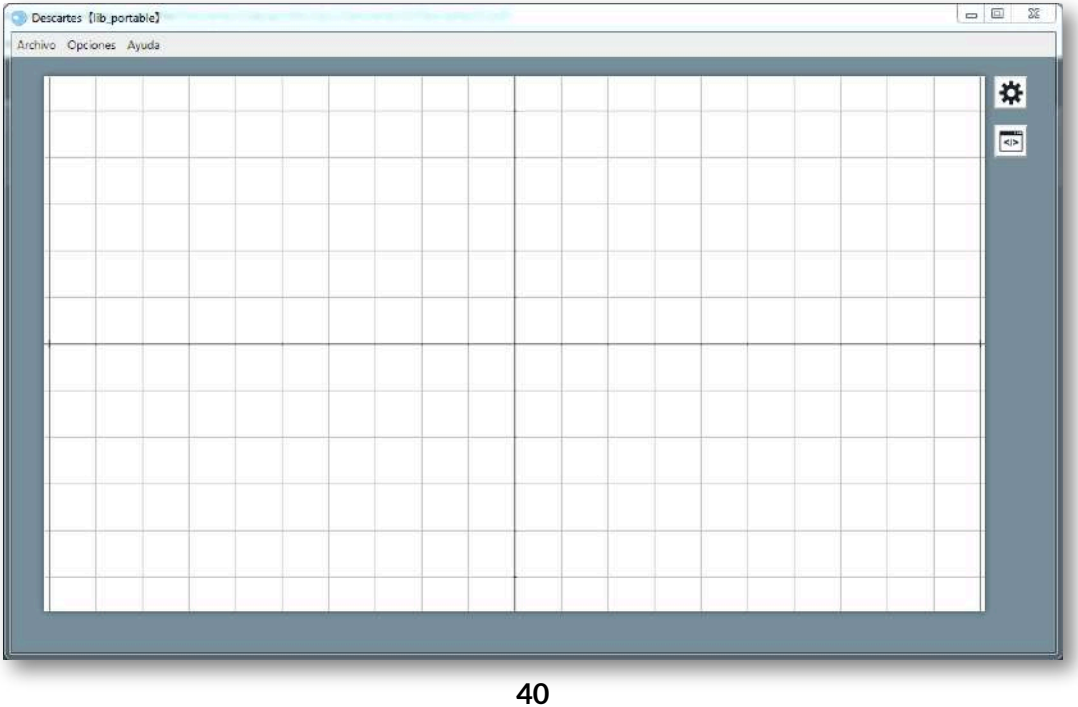
1.9 La primera escena

Vamos a describir los pasos para crear una primera escena sencilla con Descartes:

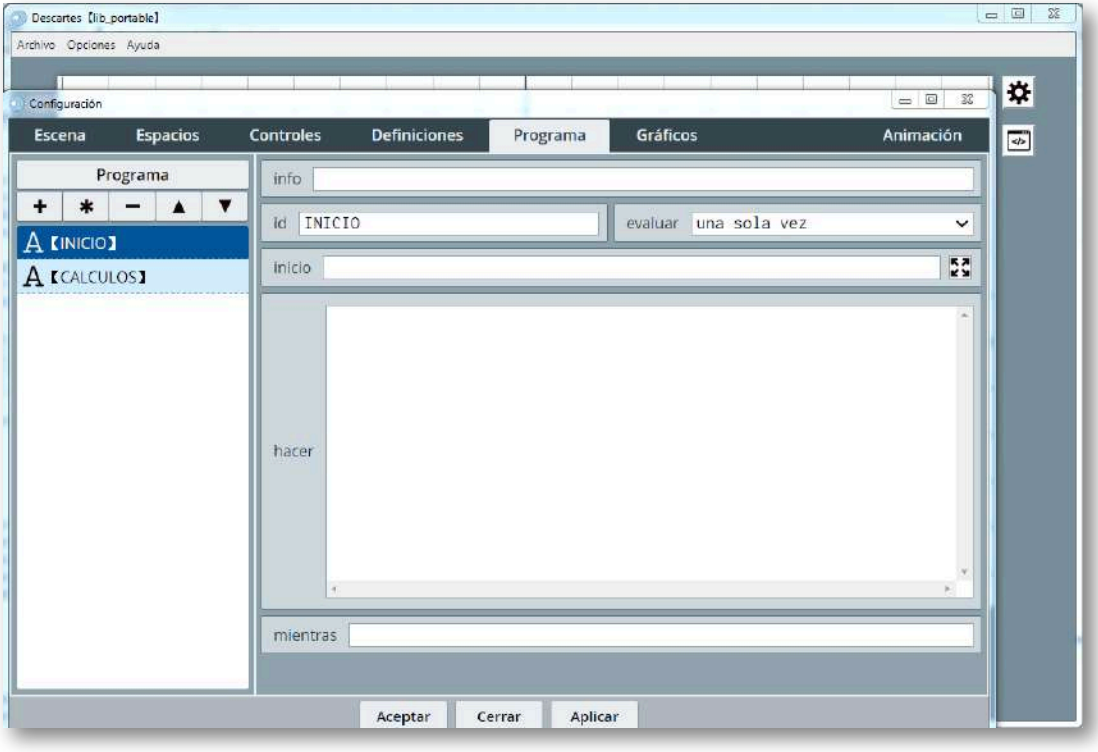
1. Abrir el **editor de Descartes**.
2. Abrir el **editor de escenas** haciendo clic en el icono de la rueda dentada.
3. Insertar los **elementos** deseados en la escena, como controles, textos, figuras, imágenes, espacios, etc.
4. **Guardar los cambios** y la escena generada.

A continuación, detallaremos cada uno de estos pasos creando una escena sencilla. En esta primera escena, añadiremos un texto que diga: *"Esta es mi primera escena con Descartes"*.

El primer paso es abrir el **editor de Descartes**, que nos permitirá crear y gestionar nuestras escenas interactivas.

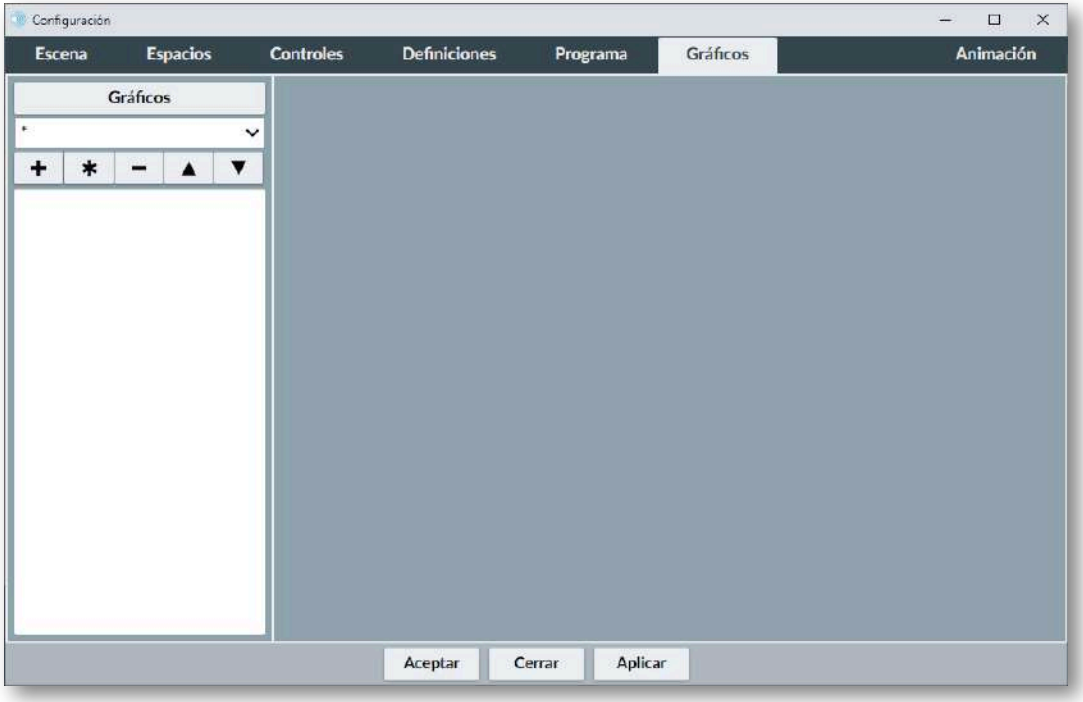


Para configurar nuestra escena, abrimos el **editor de configuración de la escena** haciendo clic en el icono de la rueda dentada del editor de Descartes.

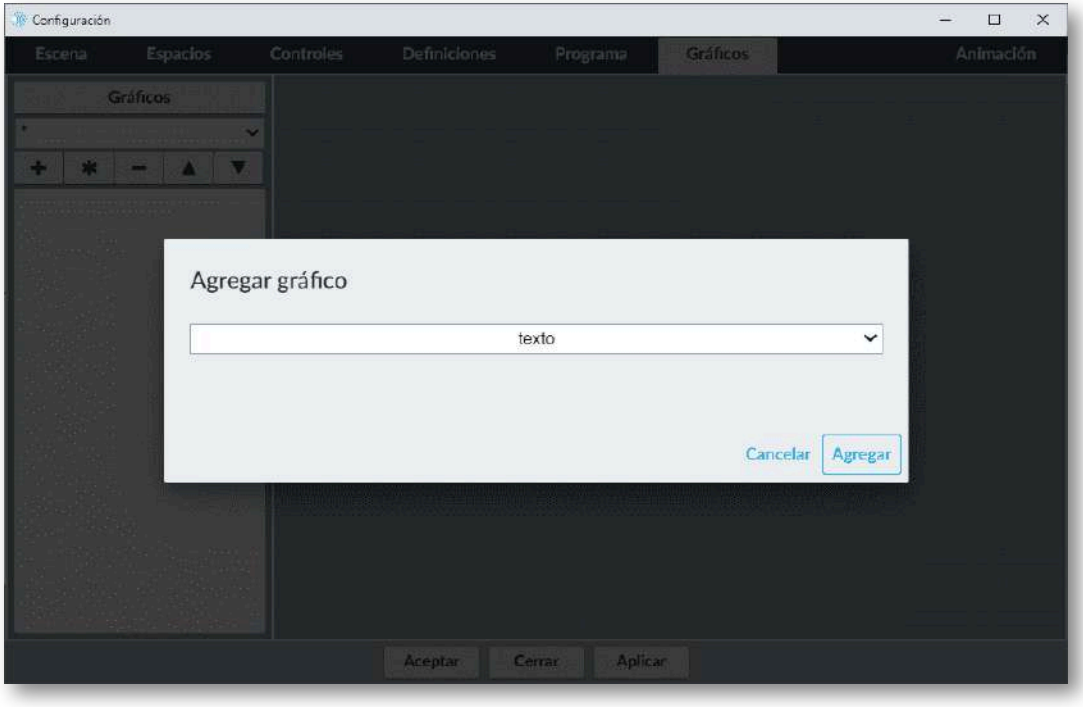


Ahora vamos a introducir un texto en la escena (su manejo se detallará con más amplitud en capítulos posteriores). Para ello daremos los siguientes pasos:

1. En el menú superior del panel de control, seleccionamos **Gráficos**.
2. Observamos la parte izquierda del editor, que corresponde a la zona donde se introducen los elementos propiamente dichos del selector *gráficos 2D*.
3. Pulsamos el botón **+** para añadir un nuevo gráfico 2D a la escena.

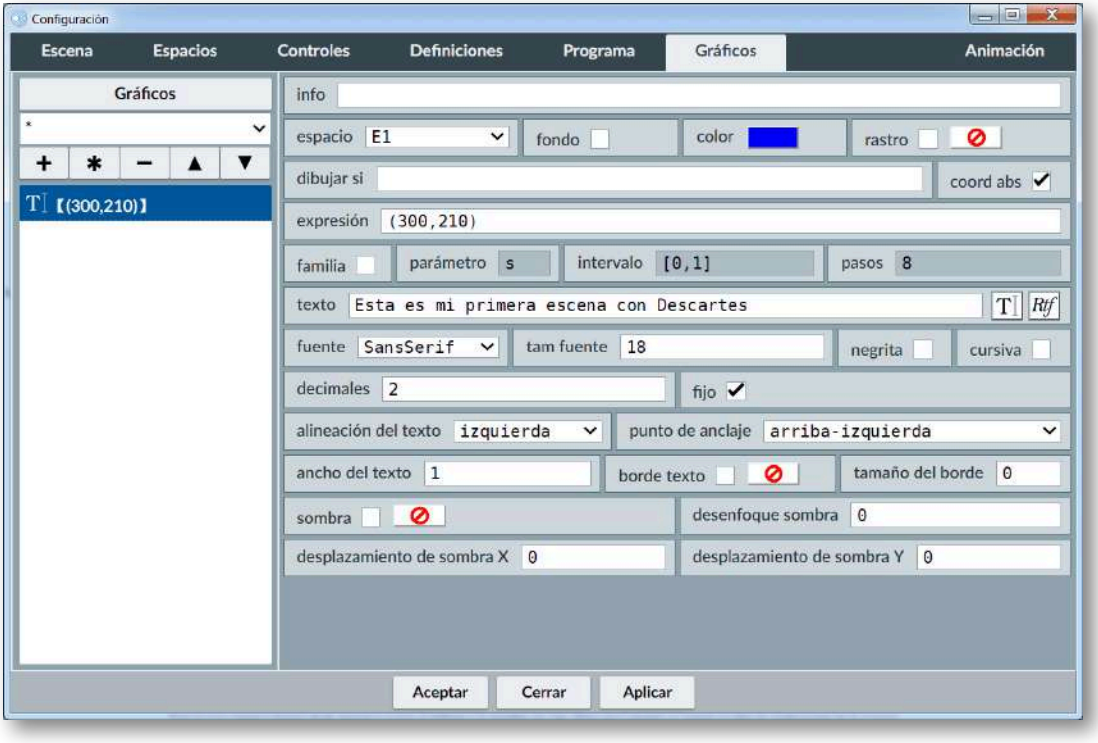


En el menú emergente que aparece elegiremos la opción texto.

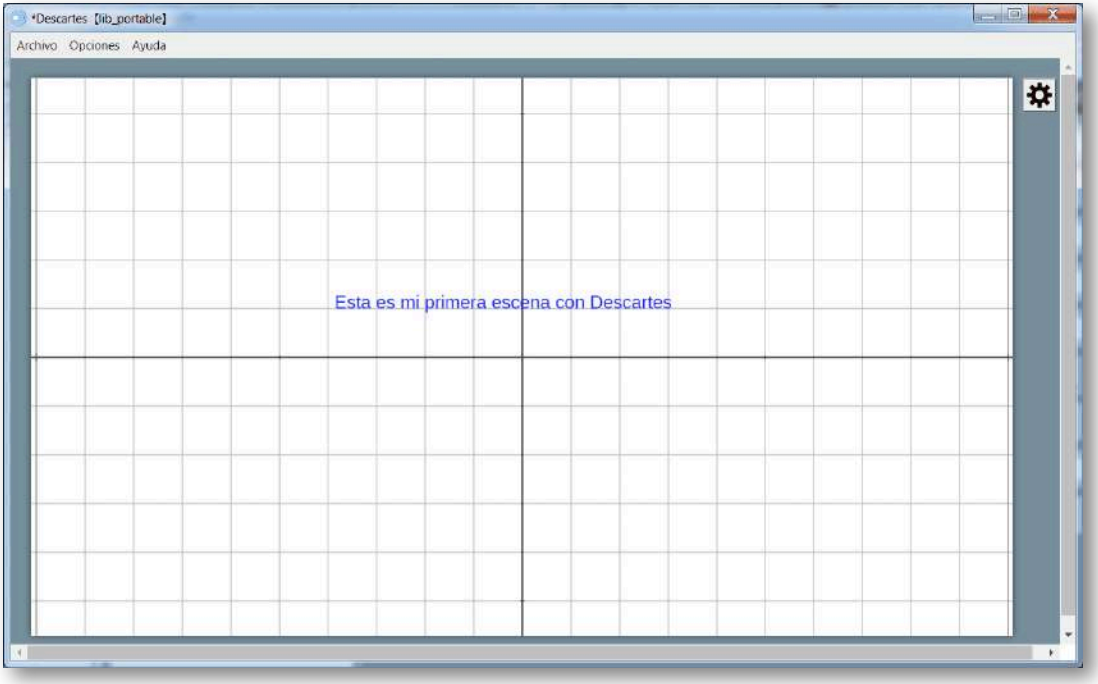


Una vez aceptados los cambios, configuramos los parámetros básicos del texto que vamos a introducir. En nuestro ejemplo, establecemos la posición en las coordenadas **(300, 210)**, seleccionamos el color **azul** para la letra y, en el campo **Texto**, escribimos el mensaje: *"Esta es mi primera escena con Descartes"*.

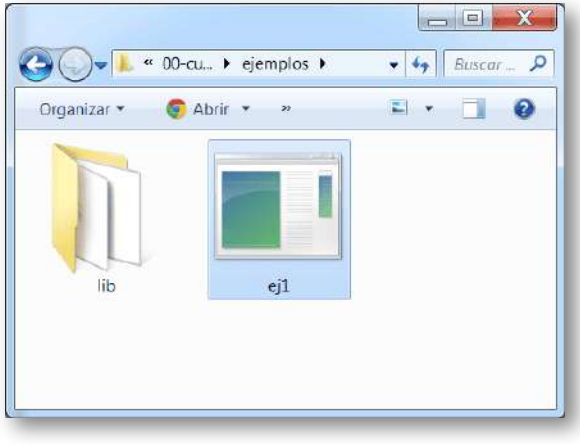
Los elementos de texto disponen de numerosas opciones adicionales, como estilos de fuente, alineación, tamaño, formato y otros ajustes, que se irán explorando más adelante.



Para que los cambios tengan efecto es necesario pulsar el botón **Aplicar** o **Aceptar**, ambos se encuentran en la zona inferior del editor de escenas. En el caso de pulsar **Aceptar**, además de aplicarse los cambios, se cerrará el editor de configuración de la escena. Si se pulsa el botón **Cancelar**, todos los cambios realizados desde la última vez que fueron aplicados o aceptados se perderán, tanto en el editor como en la escena.

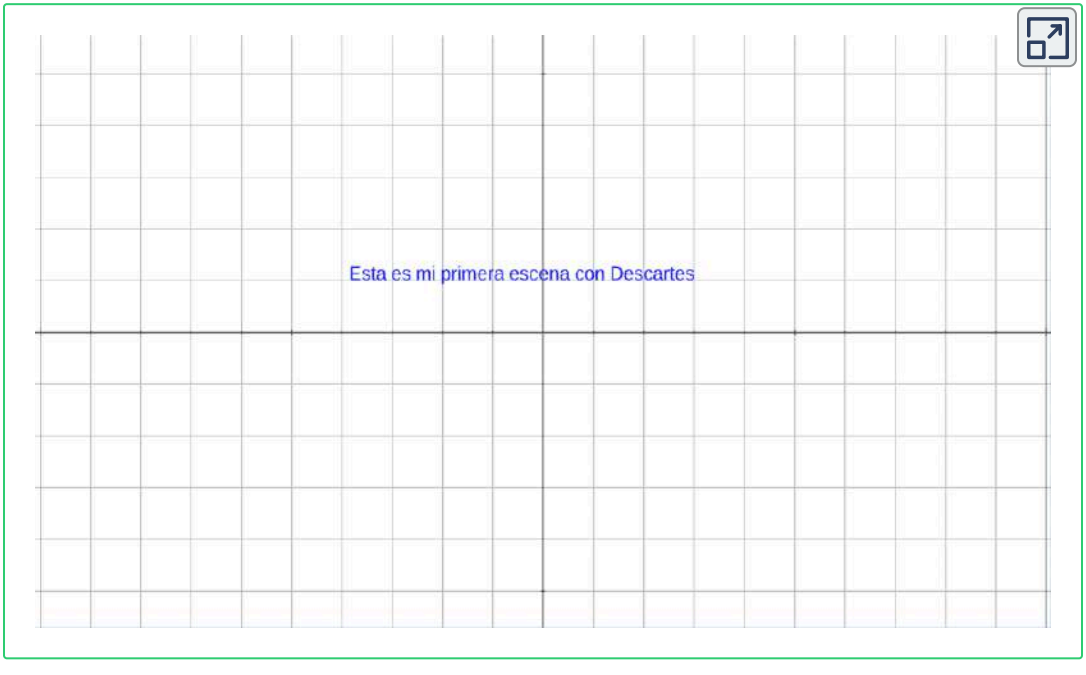


Finalmente, debemos guardar la escena desde el editor de Descartes. Antes de hacerlo, conviene comprobar en el menú **Opciones** la ubicación del intérprete de Descartes. En nuestro ejemplo, seleccionaremos la opción **portable**. A continuación, en el menú superior **Archivo**, elegimos **Guardar como** y guardamos la escena con el nombre que deseemos. En este caso, la llamaremos **ej1** y la guardaremos dentro de una carpeta llamada **ejemplos**.



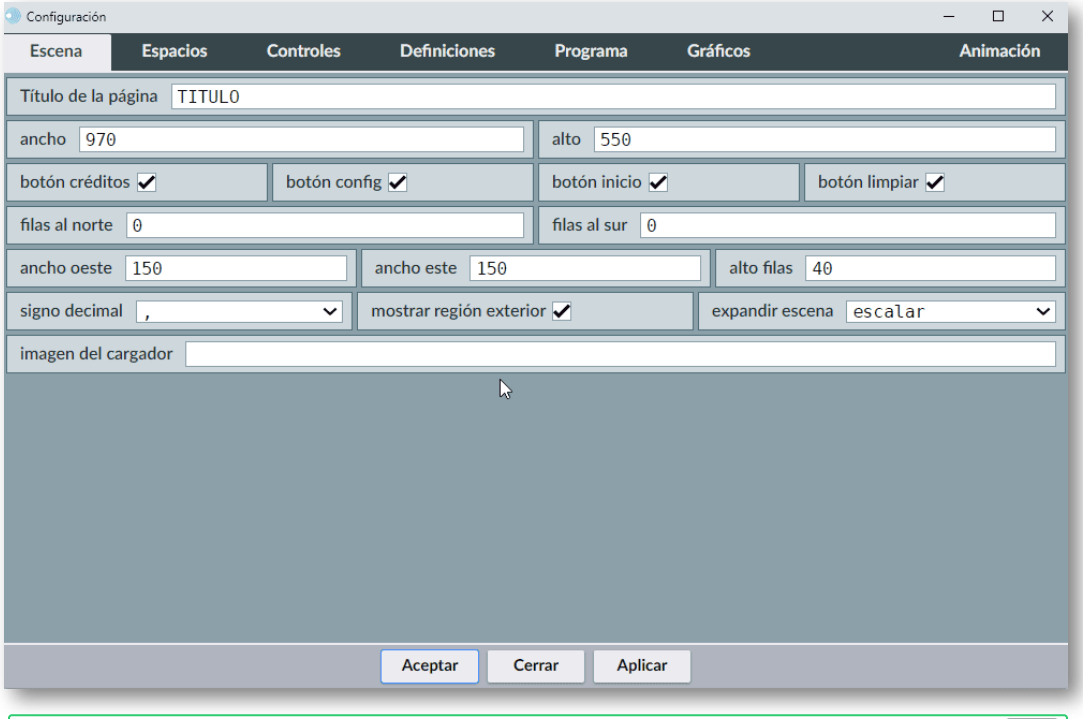
Al hacerlo, el editor crea el archivo **ej1.html**, que contiene la escena de Descartes, y también genera la carpeta **lib**, en la que se encuentra el intérprete de Descartes, **descartes-min.js**, necesario para que la escena pueda visualizarse correctamente.

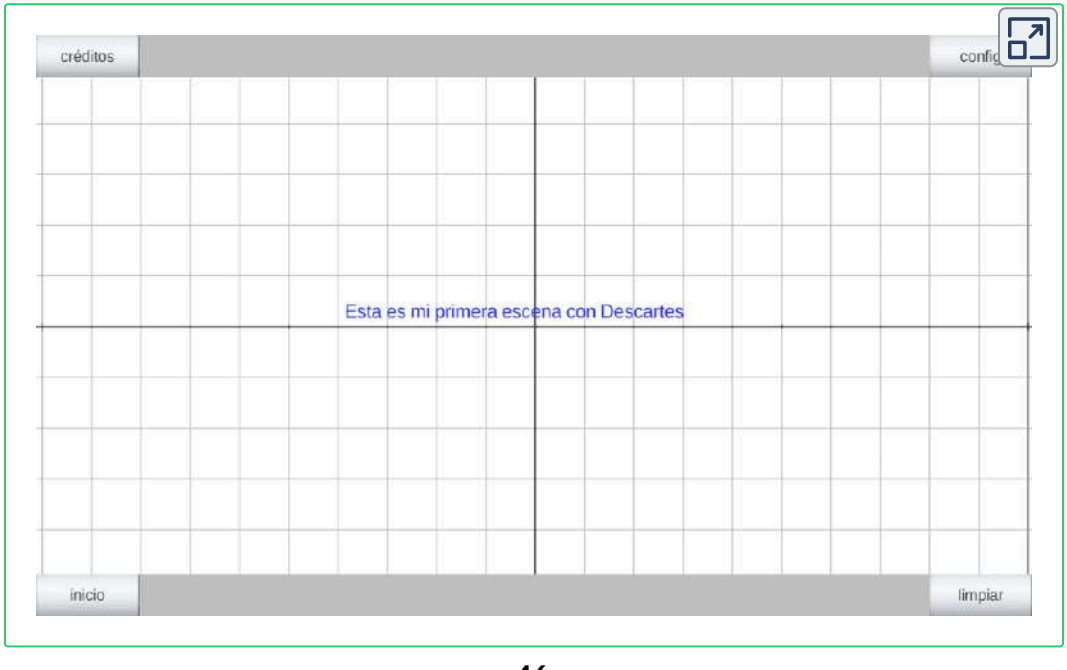
Al abrir el archivo **ej1.html** desde un navegador web, la escena se presentará tal y como se muestra en el siguiente interactivo. Al interactuar con ella mediante el ratón, se puede comprobar que no se trata de una imagen estática, sino de una escena interactiva. Además, al pulsar con el botón derecho del ratón, se mostrará el espacio o región exterior de la escena.



Esta escena puede modificarse utilizando el editor de Descartes, a través de su editor de escenas y posteriormente guardando los cambios realizados en el editor de Descartes. Otra posibilidad para intervenir sobre la escena consiste en abrir su código con un editor de texto, realizar las modificaciones necesarias y guardar los cambios efectuados.

Al activar los cuatro controles del selector **Escenas**, se representará de la siguiente manera:

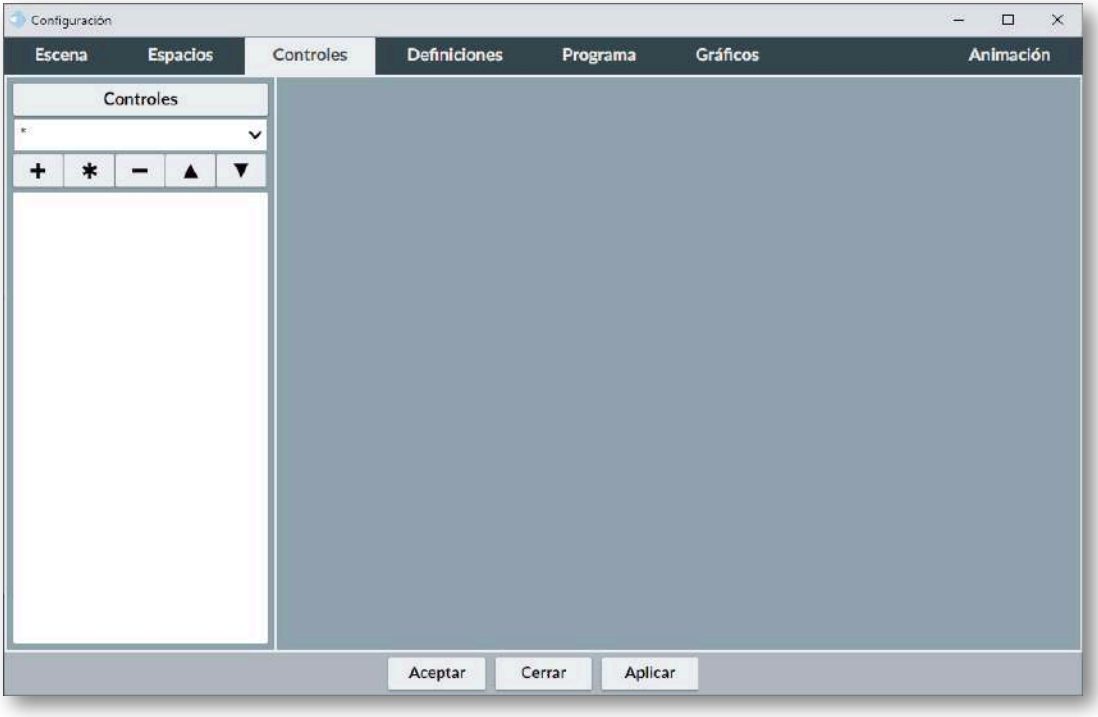




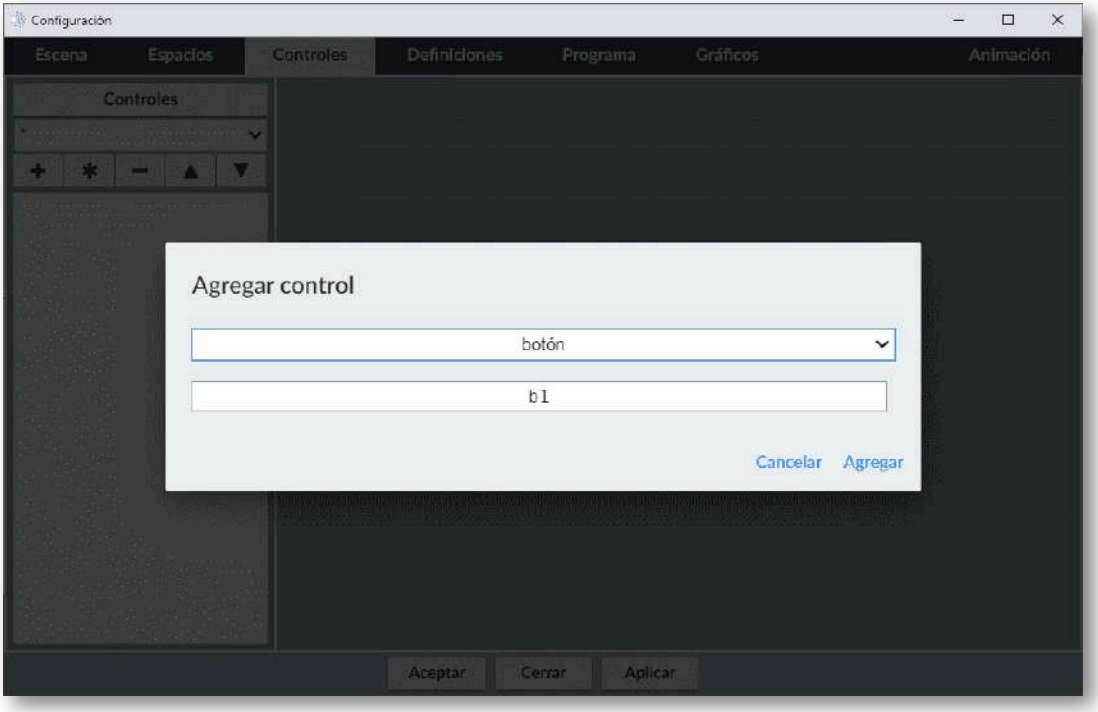
1.10 La segunda escena

Pasamos ahora a crear una segunda escena sencilla con Descartes. Para ello, repetiremos los mismos pasos indicados en la creación de la primera escena.

Tras abrir el **editor de Descartes** y después **su editor de escenas** (control rueda dentada), vamos a introducir un control de tipo botón que nos permita abrir una página web (este procedimiento se detallará con mayor profundidad en secciones posteriores). Para ello, en el menú superior del panel de control seleccionamos **Controles**, observamos la parte izquierda del editor —correspondiente a la zona donde se gestionan los controles— y pulsamos el botón **+** para añadir uno nuevo.



En el menú emergente que aparece, seleccionaremos la opción **botón** para añadir este tipo de control a la escena.

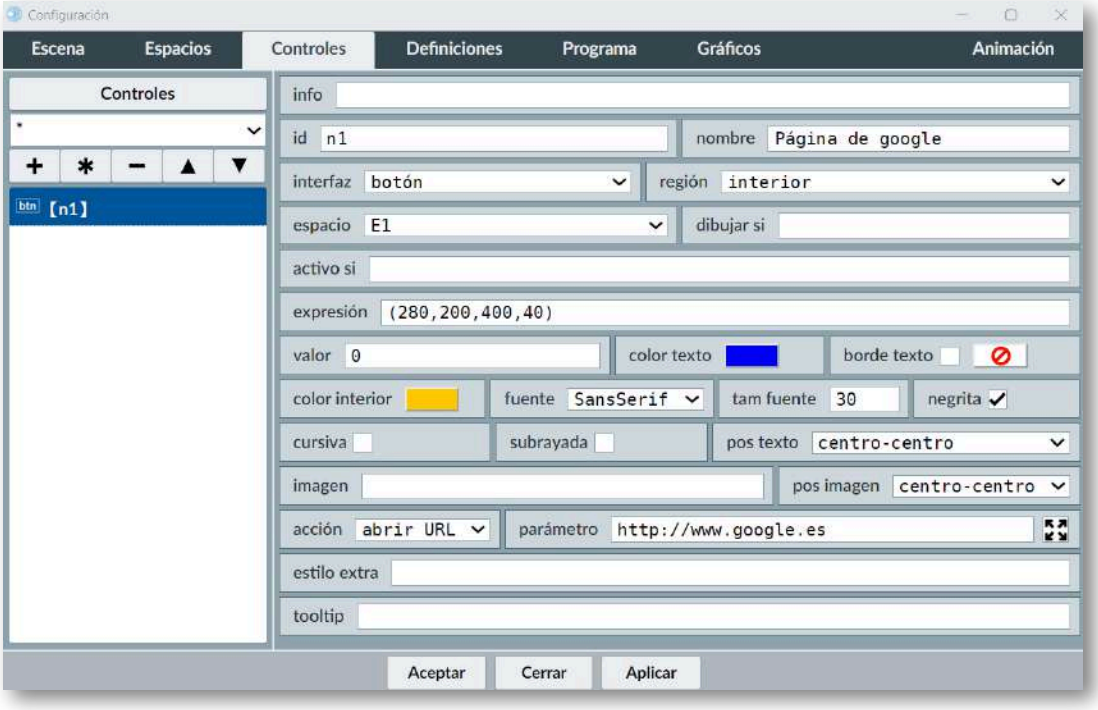


Una vez aceptado, configuramos el control de tipo botón indicando sus parámetros básicos: nombre ("Página de Google"), región (Interior), posición y tamaño (**280, 200, 400, 40**), color texto (azul), color interior (naranja) y tamaño de la fuente (19). En el apartado **Acción** seleccionamos *Abrir URL* y, en el campo **Parámetro**, introducimos la dirección web que deseamos abrir. En este ejemplo utilizaremos el enlace https://www.google.es.

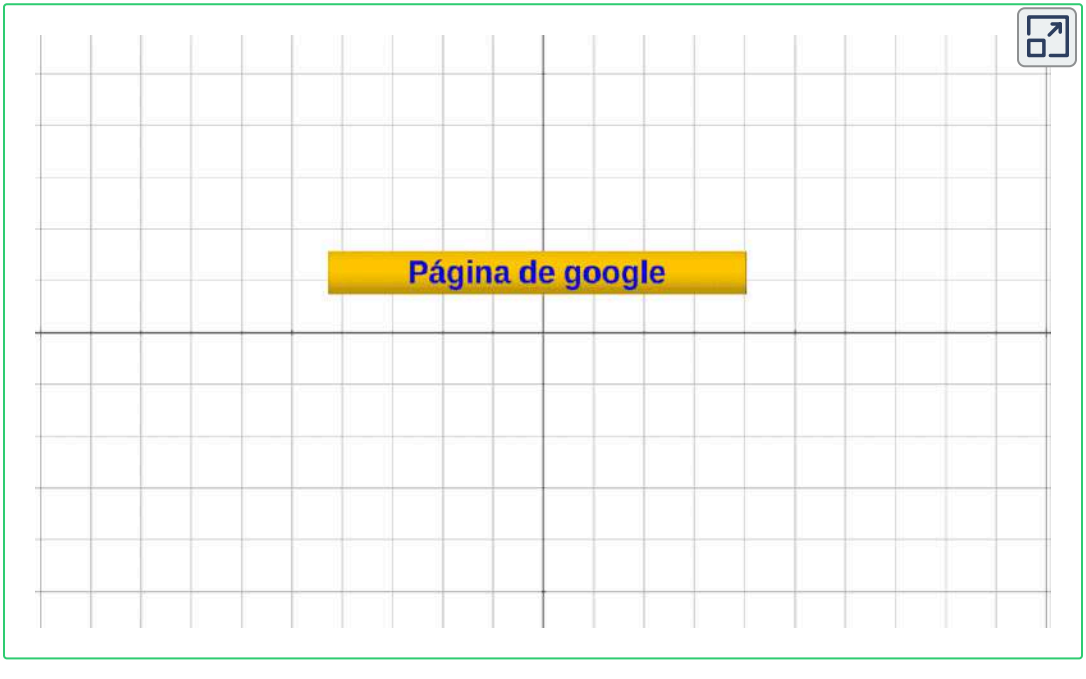
Los controles admiten numerosas opciones de configuración, tanto en su apariencia como en las acciones que pueden realizar, las cuales se irán explorando a lo largo de este libro.

Para que los cambios tengan efecto es necesario pulsar **Aplicar** o **Aceptar**. En este último caso, además de aplicarse los cambios, se cerrará el editor de configuración de la escena.

También debe guardarse la escena (**ej2.html**) desde el editor de Descartes para que los cambios se almacenen permanentemente.



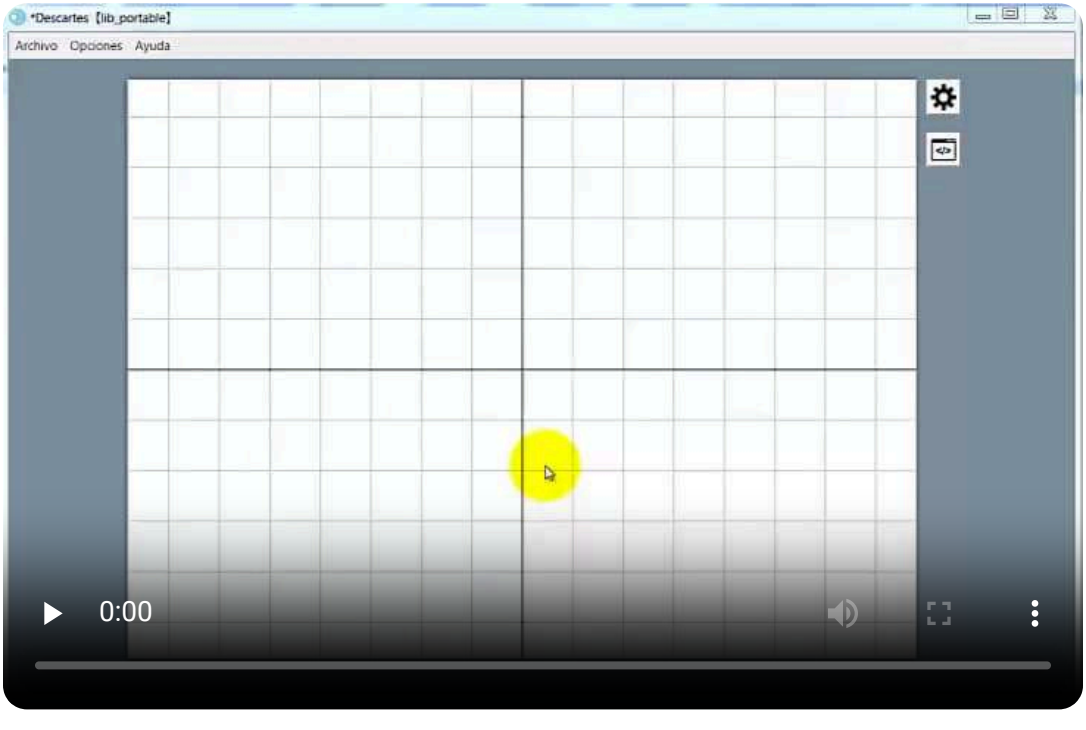
Nuestra segunda escena interactiva quedará de la siguiente forma:



1.11 La tercera escena

En este apartado se mostrará un vídeo en el que se ilustrará, paso a paso, el proceso de creación de una escena con varios elementos utilizando el editor de Descartes. Las principales características de esta escena son:

- Representación gráfica de varias ecuaciones matemáticas sencillas.
- Introducción de texto en un espacio 2D, sin añadir elementos de formato.
- Modificación de parámetros, especialmente del color, de los elementos introducidos.
- Definición de los parámetros básicos de la escena (en esta ocasión se mantienen los valores por defecto).

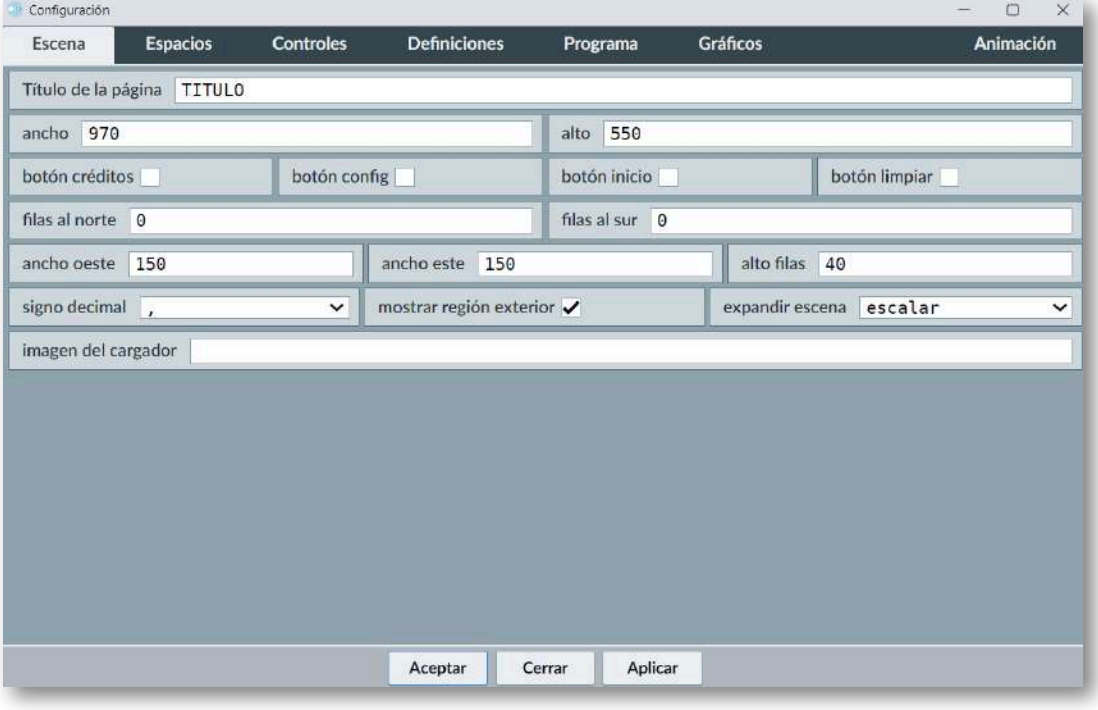


1.12 Mi primer juego paso a paso. Paso 1

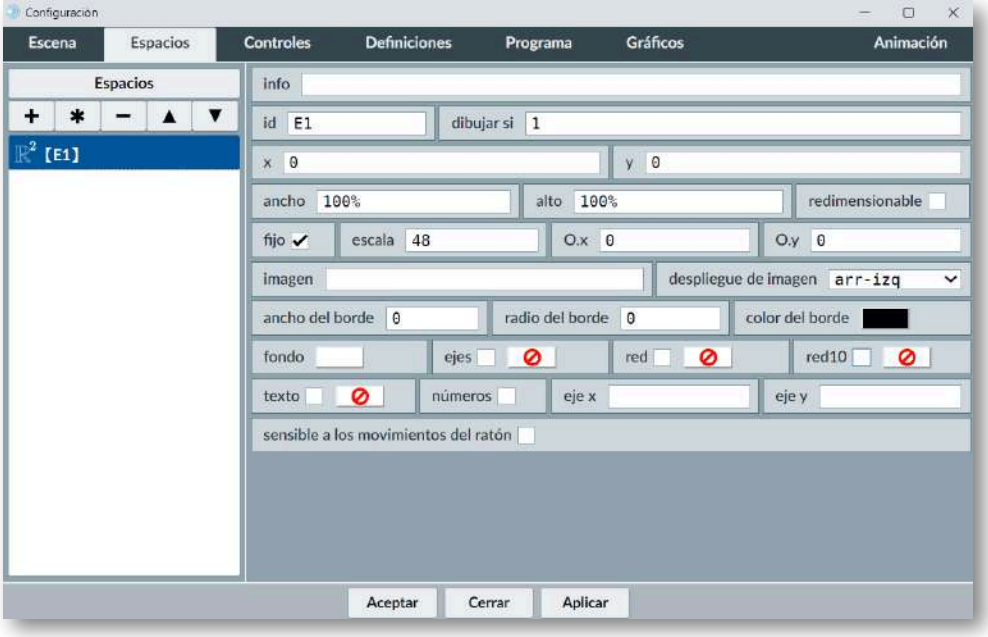
En este apartado, que estará presente en todos los capítulos del libro, se irá diseñando un juego didáctico paso a paso. En cada capítulo se irán incorporando a la escena del juego los elementos que se vayan aprendiendo.

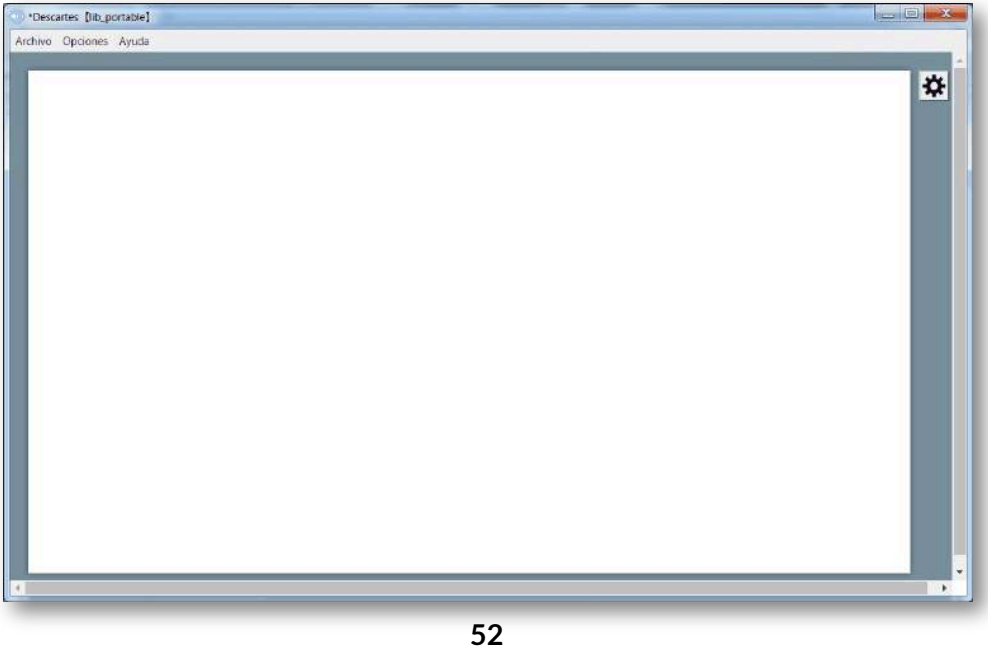
El título del juego es **«Encuentra la bolita»** y consistirá en localizar una pequeña bola que se esconderá aleatoriamente detrás de una de tres posibles casillas. El jugador gana si encuentra la bolita en el primer intento.

En primer lugar, utilizando el editor de Descartes, creamos una nueva escena y mediante el selector *Escena* del editor de escenas, le asignamos unas dimensiones de **970×550** píxeles. En expandir escena marcamos la opción **Escalar**. No activamos ninguno de los otros botones disponibles. A continuación, pulsamos en **Aplicar**.

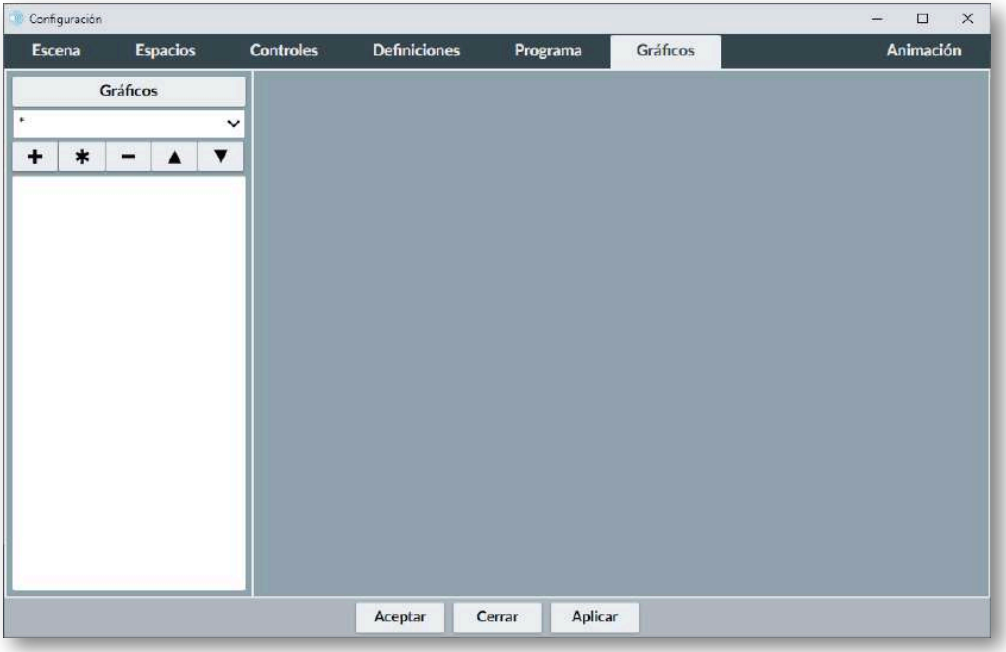


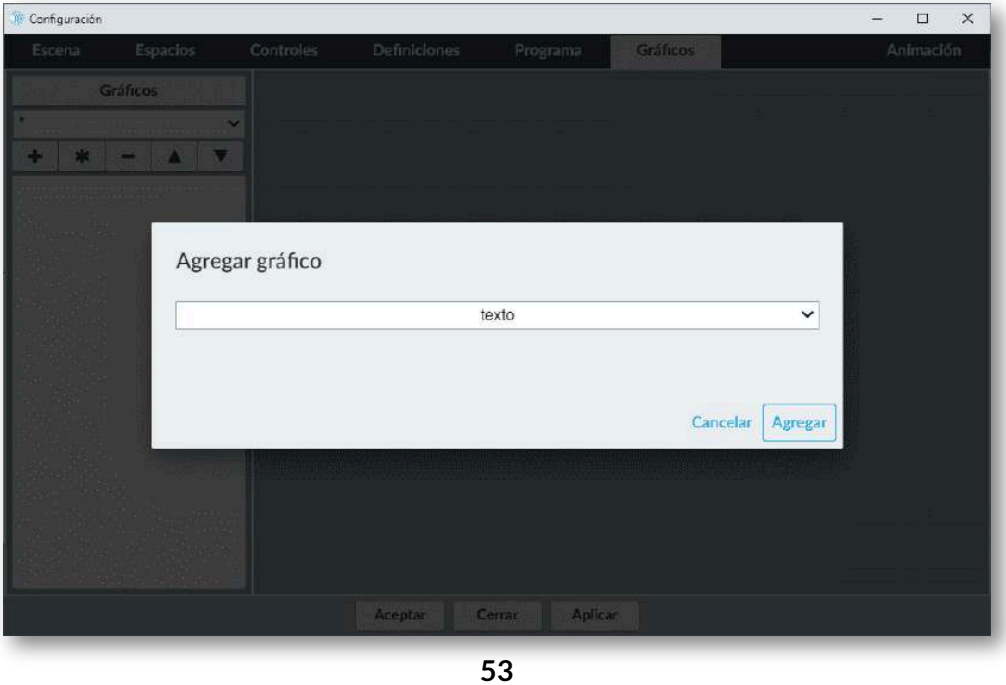
En el menú *Espacios*, seleccionamos el espacio creado por defecto y desmarcamos las opciones **ejes**, **red** y **red 10** con el objetivo de dejar la escena completamente en blanco. Además, marcamos la casilla **fijo** para evitar desplazamientos. A continuación, pulsamos en **Aplicar**.



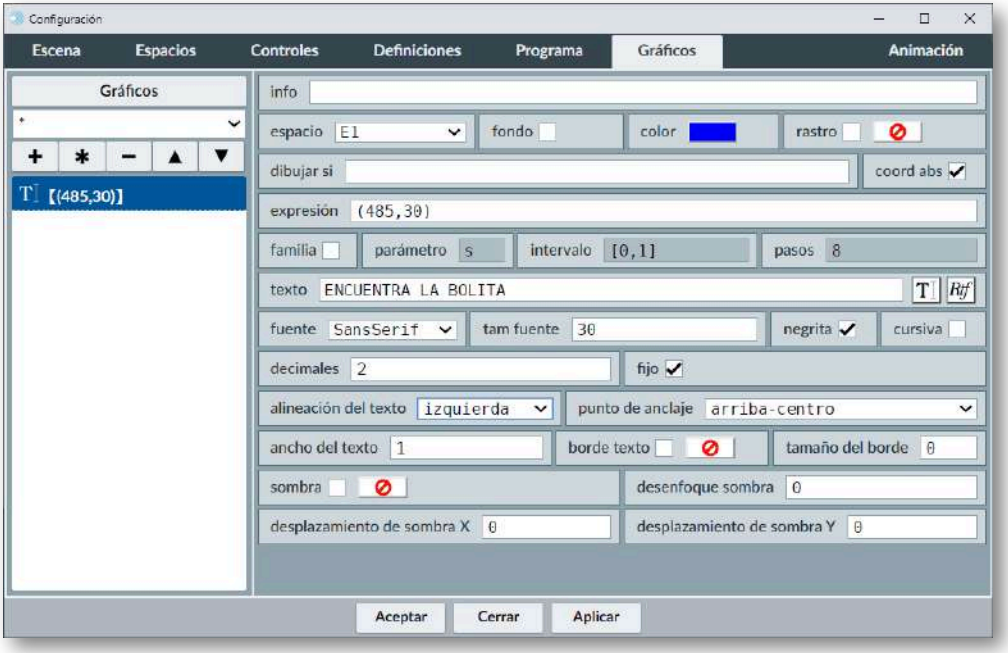


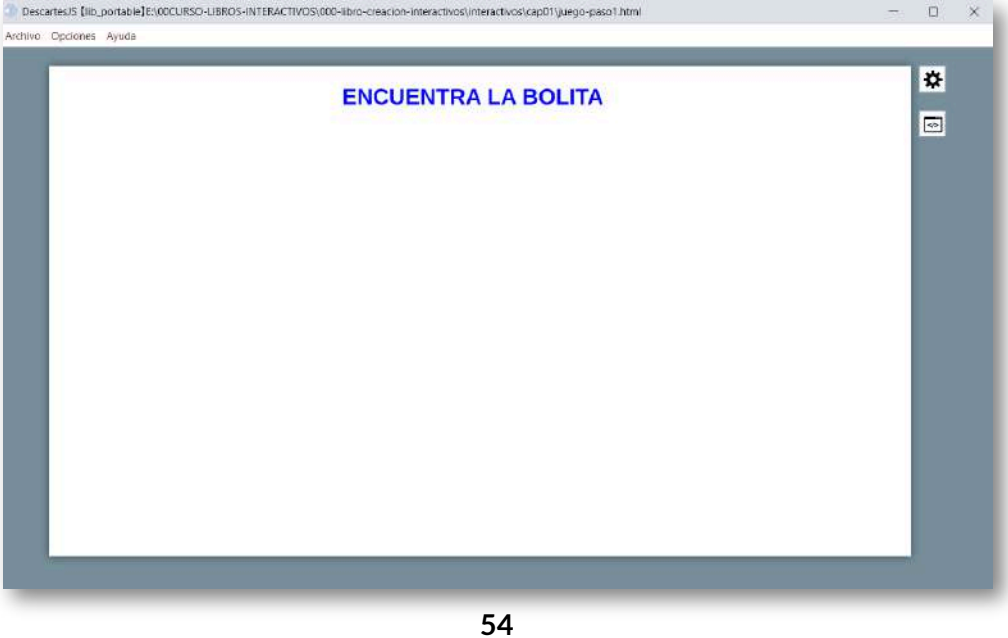
Pasamos ahora a introducir el título del juego, **“ENCUENTRA LA BOLITA”**, mediante un elemento de texto. Para ello, desde el menú *Gráficos*, pulsamos el botón **+** y añadimos un texto.





Configuramos el texto con las siguientes propiedades: color azul, estilo **negrita**, tamaño de letra el que deseemos y alineación centrada. Situamos el texto en la parte superior de la escena, alineado al centro. Finalmente, pulsamos el botón **Aplicar**.





A continuación, guardamos la escena del juego con el nombre **juego-paso1**. Con ello, ya tenemos dado el primer paso hacia la creación de nuestro primer juego didáctico.

A continuación, se muestra el interactivo del juego didáctico correspondiente al paso 1 del proceso de creación de este recurso interactivo.



Un aspecto que podemos observar en la escena correspondiente al primer paso de este juego es que el espacio es fijo. Esto implica que, aunque se pulse el botón izquierdo del ratón y se intente mover la escena, los elementos del espacio no se desplazan.

Al pulsar con el botón derecho del ratón se mostrará la región exterior de la escena.

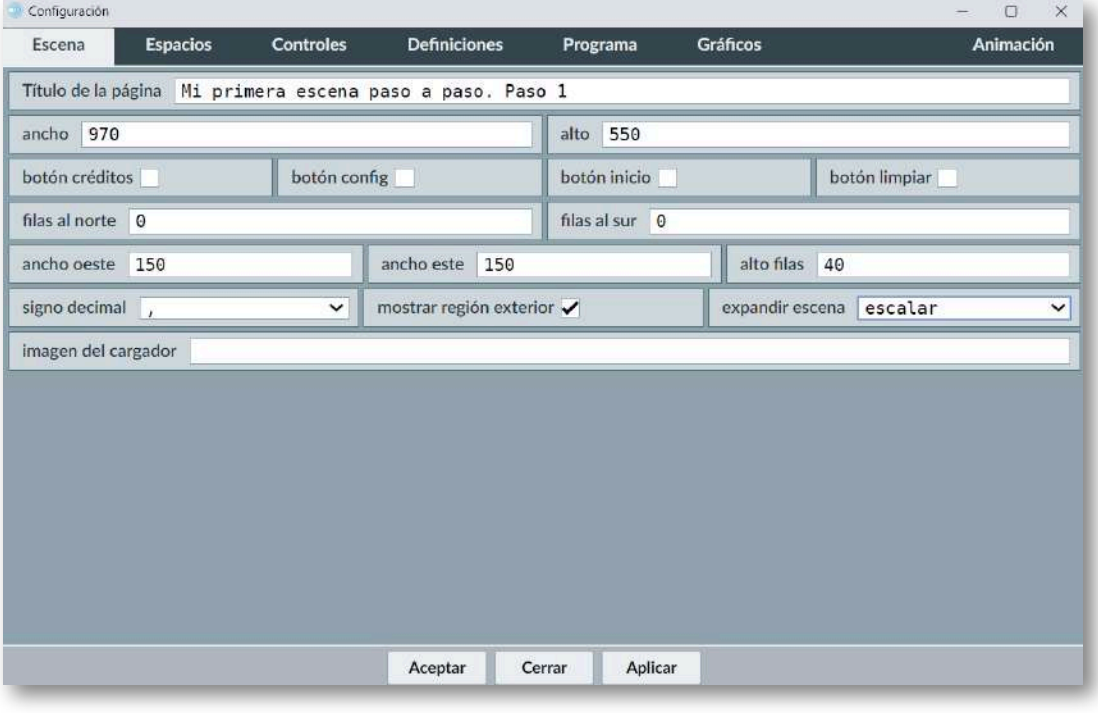
En el siguiente capítulo continuaremos con el segundo paso del proceso de diseño del juego.

1.13 Mi primera escena paso a paso. Paso 1

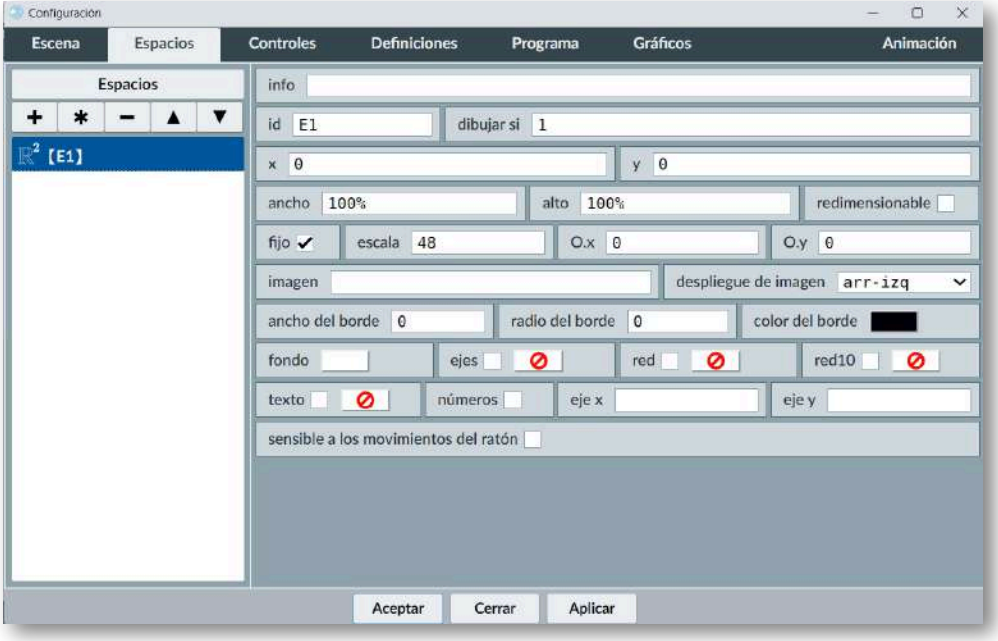
Análogamente al apartado dedicado al diseño de un juego didáctico paso a paso, procederemos de la misma manera con la creación de una primera escena sencilla que iremos completando a lo largo de los distintos capítulos.

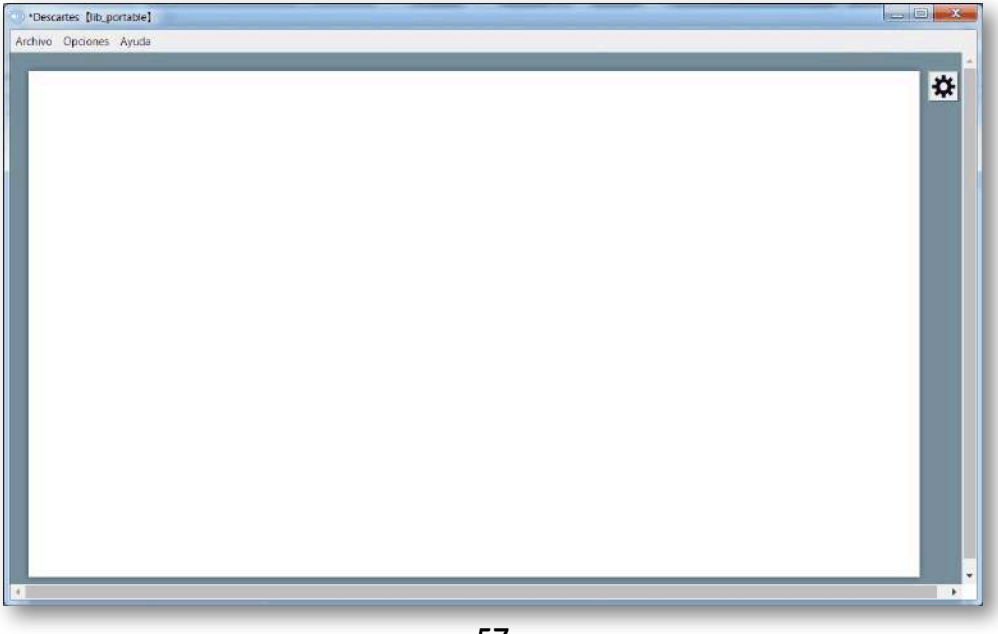
El título de la escena será **"Cuadrado y cubo"** y su objetivo será calcular y representar gráficamente algunas de las magnitudes geométricas de estas figuras: perímetro, área y volumen.

En primer lugar, utilizando el editor de Descartes, creamos una nueva escena y, a través del menú **Escena** del editor de escenas, configuramos sus parámetros iniciales. La escena tendrá las dimensiones que el editor proporciona por defecto (**970x550**), sin incluir botones en el menú de configuración. Finalmente, pulsamos el botón **Aplicar** para confirmar los cambios.

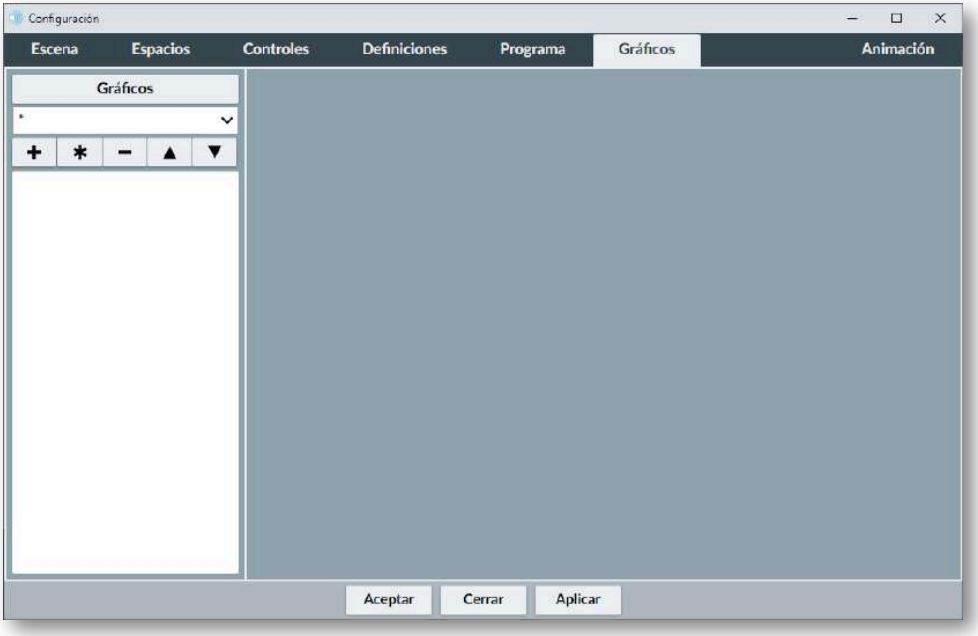


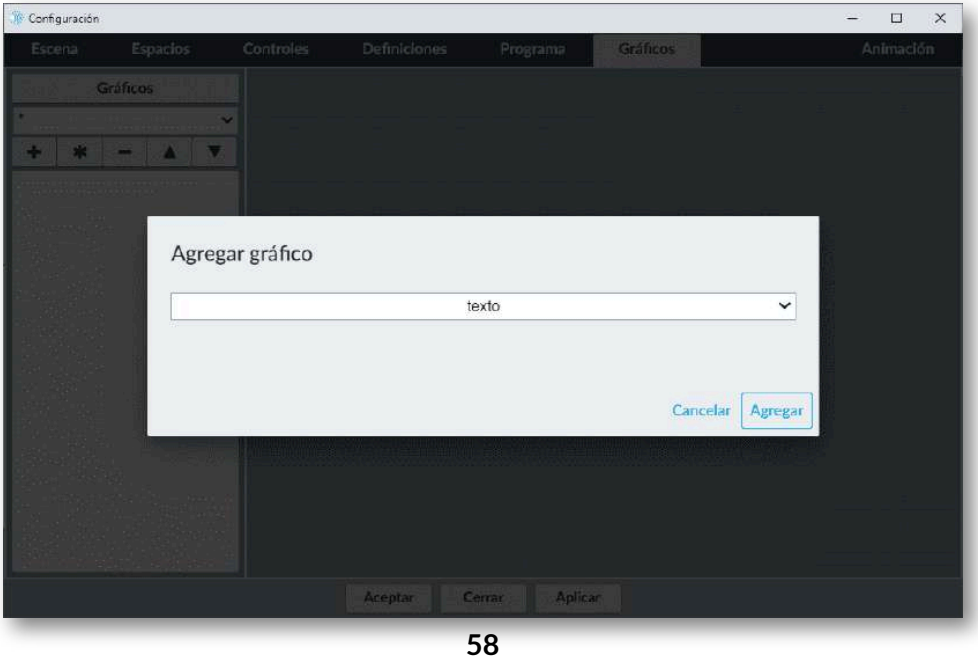
En el menú **Espacios**, dejamos el fondo blanco que aparece por defecto y desactivamos los ejes y las cuadrículas desmarcando las opciones **red**, **red 10** y **ejes**. Además, dejamos el espacio como **fijo**. Finalmente, pulsamos el botón **Aplicar** para confirmar los cambios.

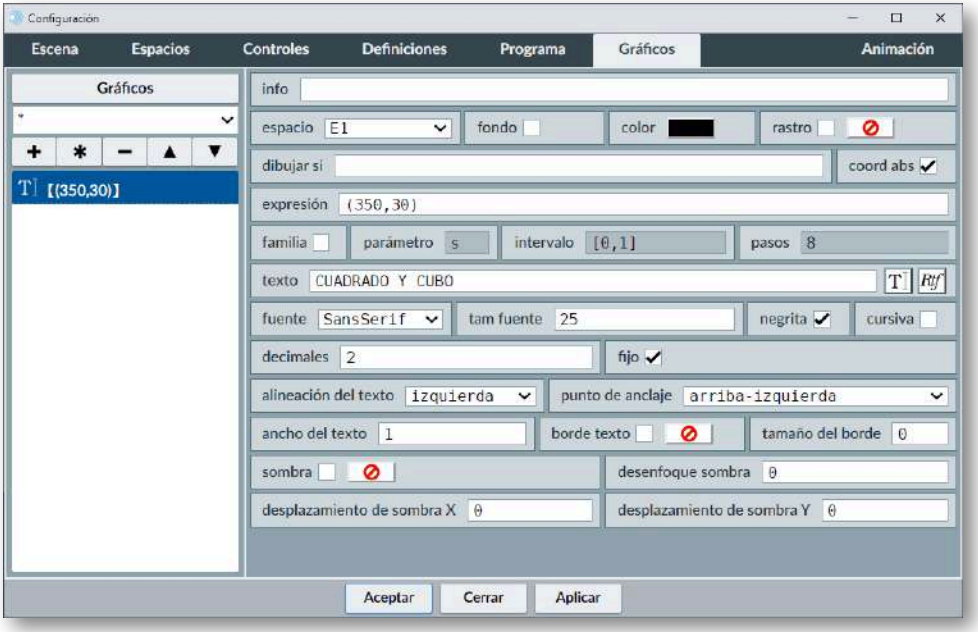




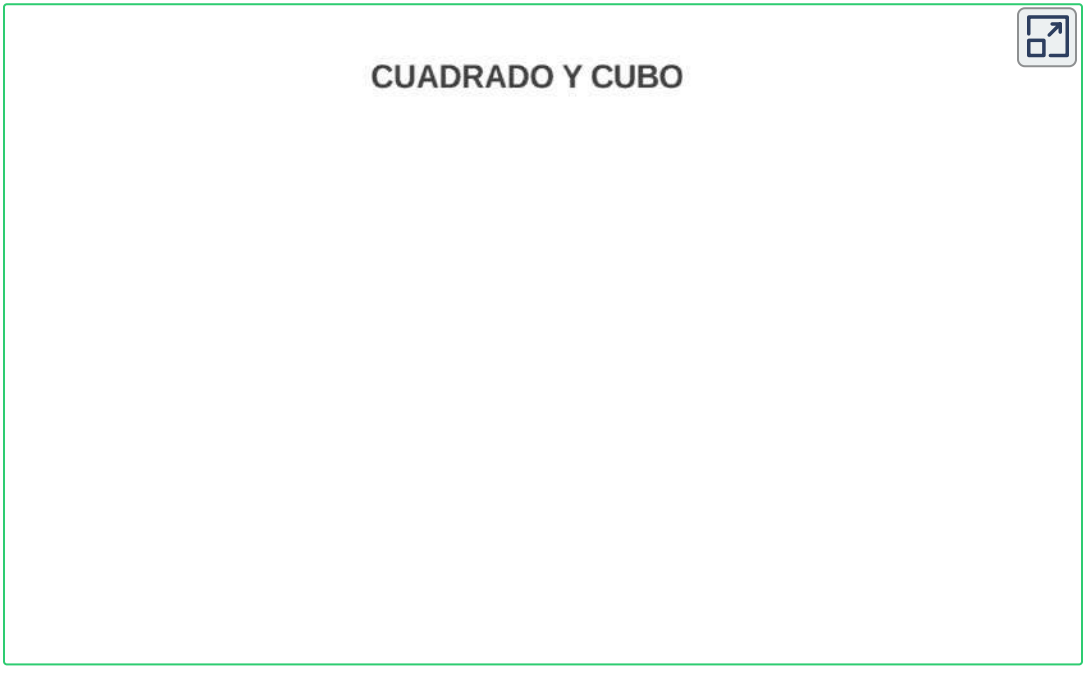
Pasamos a introducir el título de la escena **CUADRADO Y CUBO** mediante un texto, que se añade desde el menú **Gráficos** pulsando el botón **+**. El texto se configura en color gris, en negrita, con el tamaño que se desee y centrado en la parte superior de la escena.







Guardamos la escena con el nombre **escena-paso1.html** y con ello completamos el primer paso en la creación de nuestra primera escena. En el siguiente capítulo daremos el segundo paso.





Capítulo 2

Espacios, animaciones y elementos básicos

Capítulo 2
Espacios, animaciones y elementos básicos

2.1 Introducción 63

2.2 Coordenadas en escenas 66

2.3 Espacios en escenas 74

2.4 Coordenadas en espacios 84

2.5 Tipos de espacios 88

2.6 Espacios 2D 91

2.7 Espacios 3D 93

2.8 Espacios HTMLIFrame 96

2.9 Imágenes en escenas 101

2.10 Textos en escenas 109

2.11 Editor de textos con formato 117

2.12 Colores en textos y otros elementos 126

2.13 Animaciones en escenas 138

2.14 Movimiento en una escena 144

2.15 Mi primer juego paso a paso. Paso 2 151

2.16 Mi primera escena paso a paso. Paso 2 154

2.1 Introducción

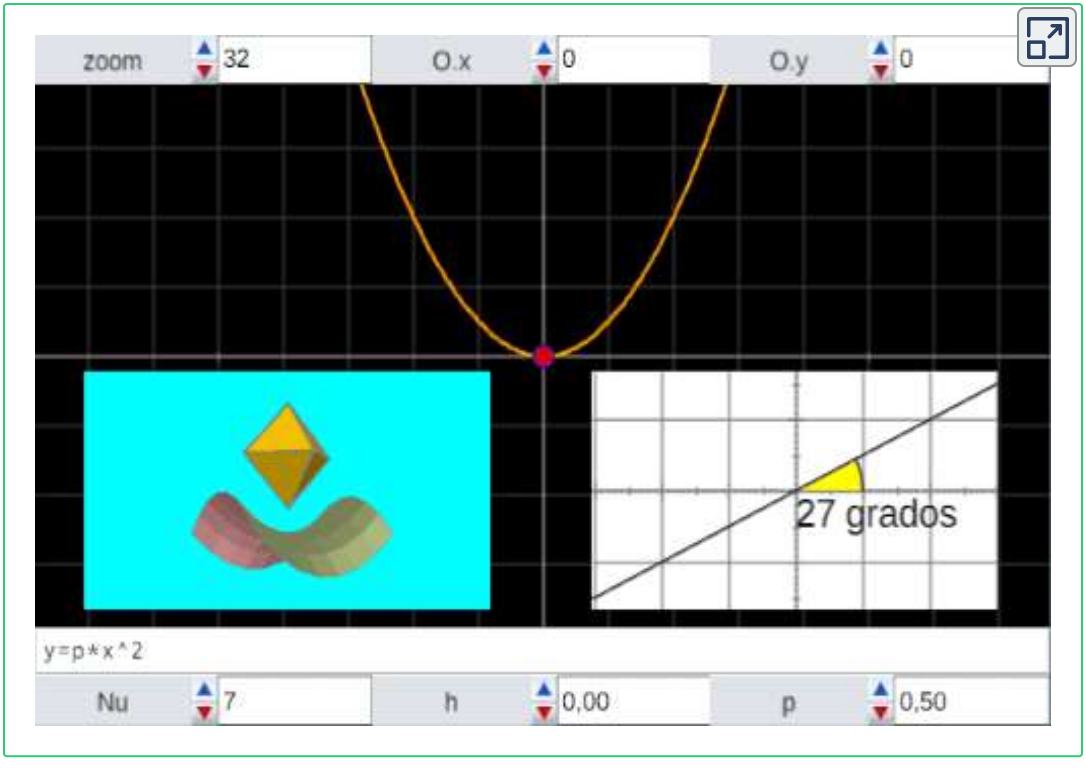
Las escenas pueden dividirse en distintas partes o **espacios**, que son las diferentes ventanas que pueden abrirse dentro de una misma escena y que pueden estar relacionadas entre sí. Como mínimo, una escena dispone de un espacio (cuando solo tiene una ventana), y como máximo puede tener tantos como se desee. Los espacios pueden tener cualquier tamaño y ocupar cualquier posición dentro de la escena, incluso pueden superponerse total o parcialmente.

- **Espacios de dos dimensiones (R2):** Son entornos que se desarrollan en un plano con ancho y alto. Permiten organizar y visualizar elementos de forma plana, como gráficos, textos, polígonos o imágenes.
- **Espacios de tres dimensiones (R3):** Añaden la profundidad a las dos dimensiones anteriores, lo que posibilita representar objetos con volumen, cuerpos geométricos y crear entornos más realistas e interactivos.
- **Espacios HTMLIframe:** Permiten insertar contenido externo mediante un archivo HTML o enlace web, por lo que permiten introducir cualquier contenido compatible con páginas web.

Además, las escenas pueden estar animadas o no; es decir, podemos dotar tanto a la escena como a sus espacios de movimiento o secuenciación.

Por tanto, en este capítulo aprenderemos a insertar en nuestras escenas los siguientes elementos básicos: **espacios**, **animaciones**, **imágenes** y **textos**.

En la siguiente escena se muestra un ejemplo de interactivo con varios espacios: uno principal de fondo negro y dos más pequeños, uno de fondo celeste y otro blanco. En cada uno de ellos se pueden definir los elementos que se deseen.

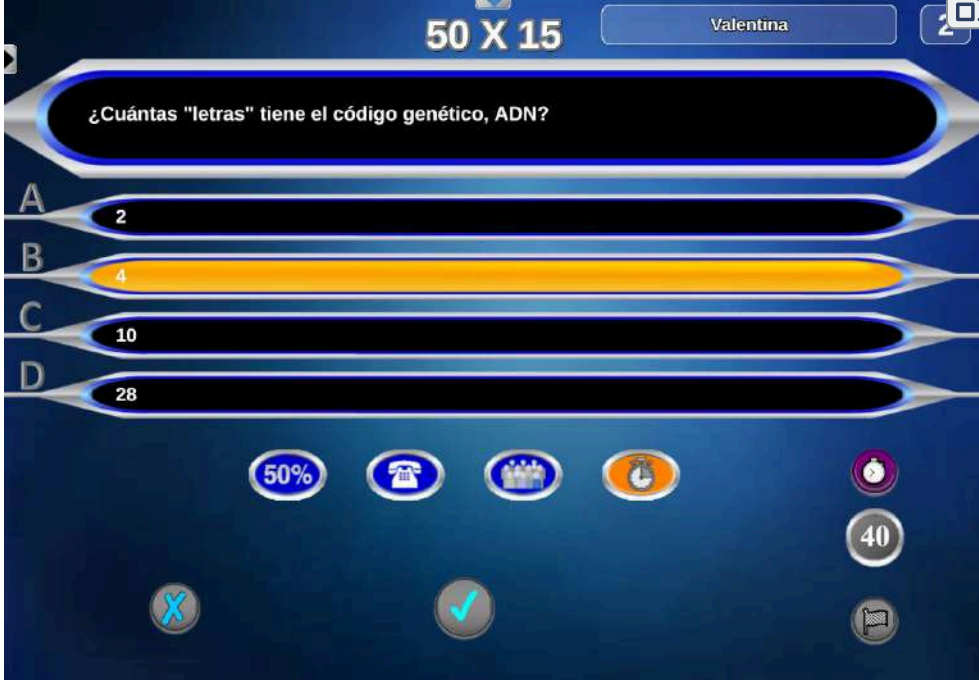


Las escenas de DescartesJS que implementan juegos didácticos suelen incluir múltiples elementos: varios espacios, imágenes, textos, animaciones, sonidos, controles y otros recursos interactivos de diferente naturaleza.

Todo ello permite crear experiencias dinámicas y atractivas para el usuario. En la siguiente imagen, correspondiente al juego **"Identity"**, pueden observarse claramente los distintos componentes que forman la escena: diferentes espacios de trabajo, imágenes integradas, textos informativos y una animación que controla el tiempo de juego, entre otros elementos.

Incluimos también la escena del juego didáctico interactivo **50x15**, completamente operativo. Este juego está inspirado en el conocido concurso de televisión *¿Quién quiere ser millonario?* y está íntegramente desarrollado con DescartesJS.


6
4
IDENTITY
MARTA
57
Tiempo 1
1. K. BENZ
2.
3. C. BERTHOLLET
4.
5.
6. W. WIEN
7.
8.
9.
PISTA 1
PISTA 2
1
2
3
Propuso un método para sintetizar la hormona aldosterona.
Era filósofo y afirmaba que las sensaciones estaban desligadas de los objetos.
4
5
6
Inventó una máquina que lleva su nombre consistente en una polea y dos masas.
Fue inspector general de educación, senador y ministro francés.
7
8
9
Ideó el densímetro que lleva su nombre.
Ingeniero pionero en la industria del automóvil.
Nueva tirada
CORRECTO
900

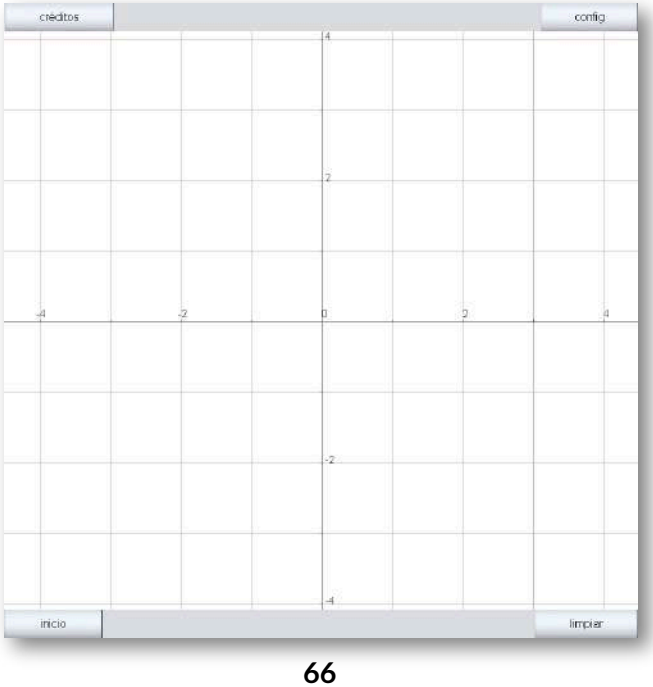

50 X 15
Valentina
¿Cuántas "letras" tiene el código genético, ADN?
A
2
B
4
C
10
D
28
50%
40

2.2 Coordenadas en escenas

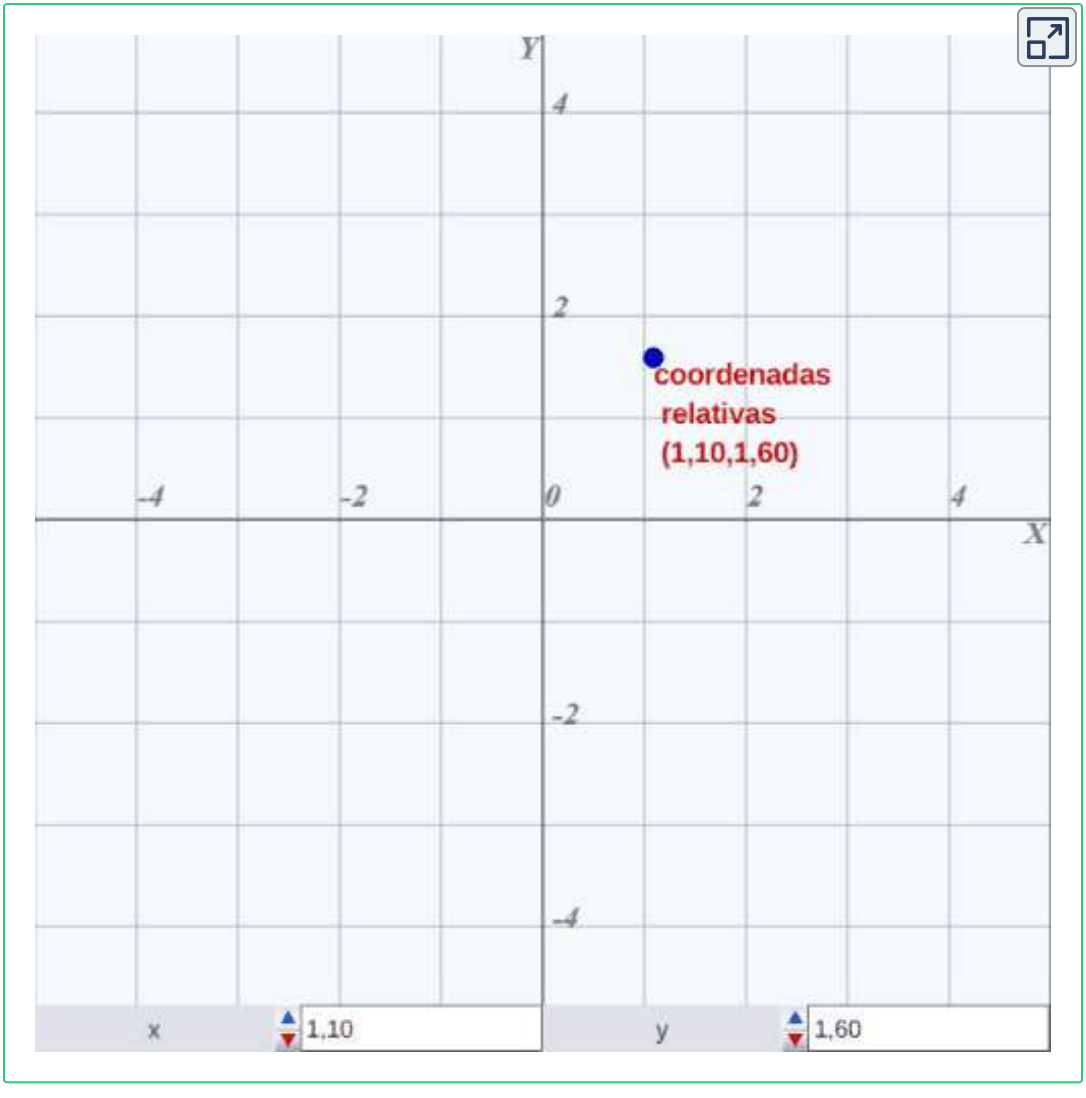
Uno de los pasos clave al trabajar con una escena es establecer correctamente su sistema de coordenadas, así como el de los espacios que la componen. Esto permite ubicar con precisión cada elemento que se incorpore —ya sean imágenes, textos, controles u otros objetos— dentro del entorno definido. Tanto en la escena principal como en sus distintos espacios pueden emplearse dos sistemas de referencia:

Sistema de coordenadas relativo

En este caso, el origen de coordenadas se sitúa en el centro de la escena o del espacio, es decir, en el punto (0,0). A partir de ahí, los valores del eje X aumentan hacia la derecha y disminuyen hacia la izquierda, mientras que los valores del eje Y aumentan hacia arriba y disminuyen hacia abajo. En la imagen correspondiente —que representa una escena con un único espacio— se muestran los ejes definidos según este sistema de coordenadas relativas.



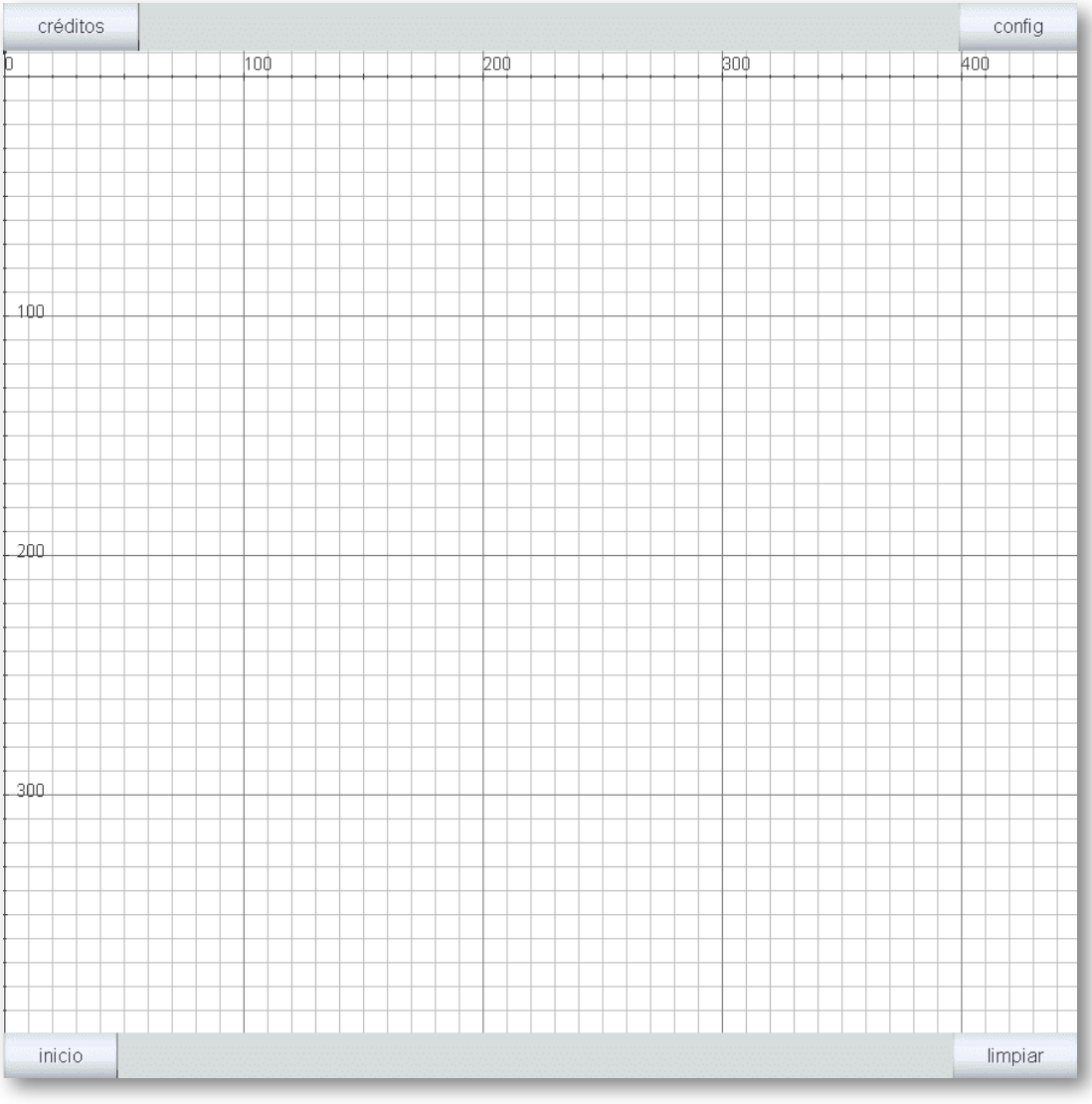
En la siguiente escena se presenta un punto cuyas coordenadas están referenciadas en un sistema relativo.



Sistema de coordenadas absoluto

En el sistema absoluto, el origen de coordenadas se encuentra en la esquina superior izquierda de la escena o de cada espacio. En este punto se ubica el (0,0). Desde ahí, la coordenada X incrementa su valor hacia la derecha y la coordenada Y aumenta hacia abajo.

En la imagen asociada —también correspondiente a una escena con un único espacio— se pueden observar los ejes definidos según el sistema de coordenadas absolutas.



En la siguiente escena se presenta un punto cuyas coordenadas están referenciadas en un sistema absoluto. Comentamos que las coordenadas pueden ser negativas, lo que implica que el objeto estaría colocado en una posición fuera del espacio.



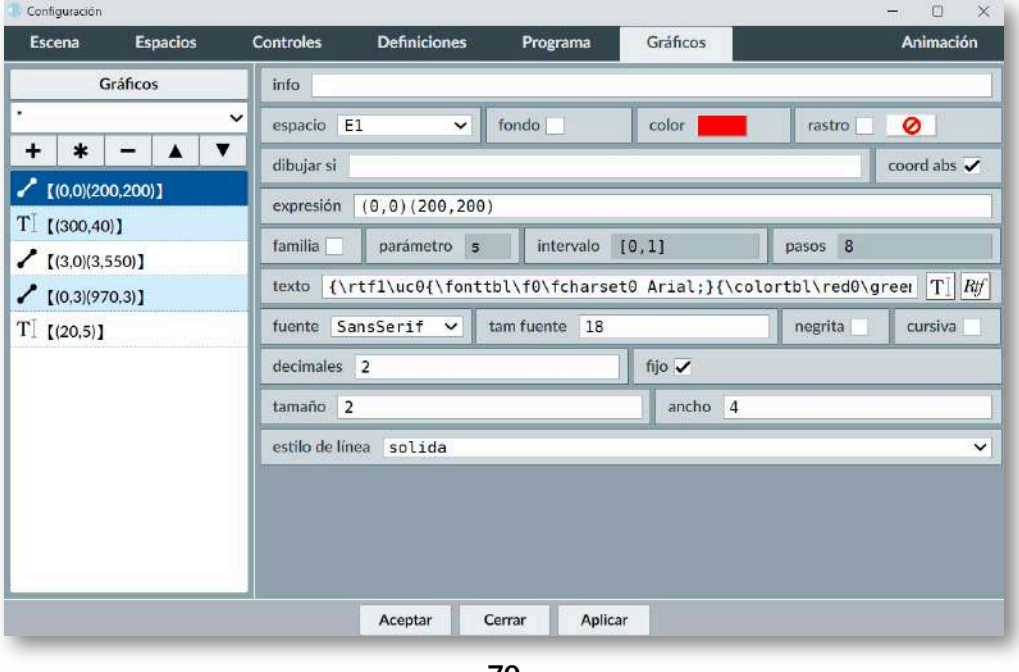
La posición de los distintos espacios dentro de una escena —en caso de que exista más de uno— debe indicarse obligatoriamente mediante coordenadas absolutas, tomando como sistema de referencia el de la propia escena, cuyo origen se sitúa en la esquina superior izquierda.

Es posible definir un espacio fuera de los límites visibles de la escena (total o parcialmente); en ese caso no se mostrará en pantalla, aunque podrá desplazarse posteriormente hasta quedar dentro del área visible. Los espacios de DescartesJS admiten incluso un cambio dinámico de sus dimensiones.

En cuanto a los objetos contenidos en un espacio, sus posiciones se determinan utilizando como referencia el propio espacio que los alberga. Existen objetos cuya ubicación debe definirse necesariamente mediante coordenadas absolutas. Sin embargo, otros elementos —como segmentos, polígonos o flechas— pueden posicionarse empleando ambos, siempre teniendo en cuenta que el sistema de referencia está vinculado al espacio contenedor.

En aquellos objetos que admiten ambos sistemas de referencia, se considerará que están definidos por defecto respecto al sistema absoluto del espacio cuando esté activada la casilla **"coord_abs"** en el panel de configuración del objeto. Si dicha opción no está marcada, se interpretará que utilizan el sistema de coordenadas relativas.

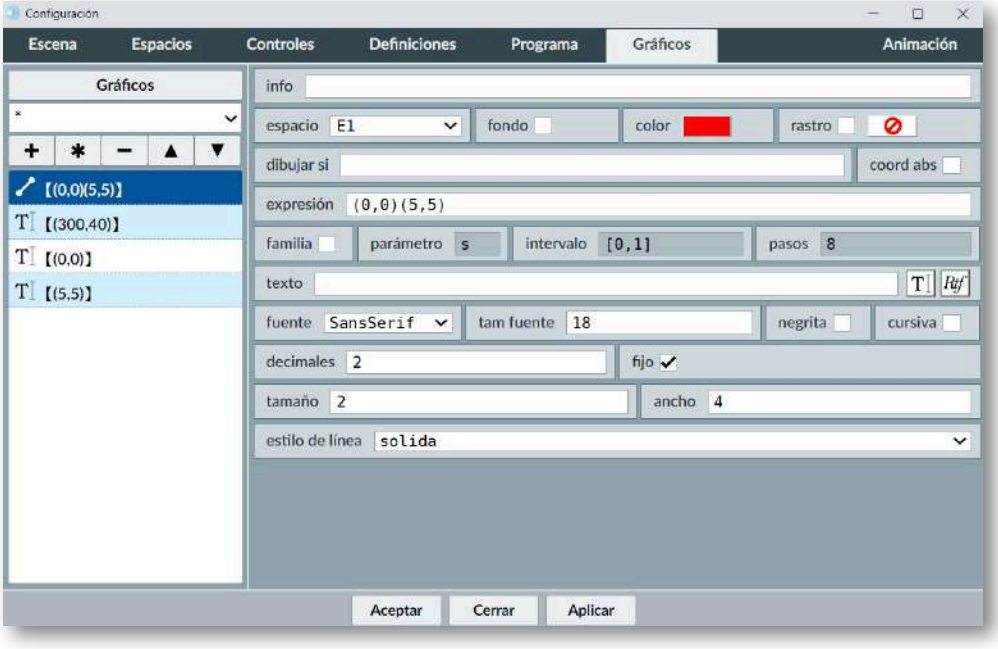
A continuación, se muestra un segmento definido mediante coordenadas absolutas, al estar activado el control **"coord_abs"**. En la escena puede observarse que el origen de coordenadas se sitúa en la esquina superior izquierda, de modo que la coordenada X aumenta hacia la derecha y la coordenada Y incrementa su valor hacia abajo.

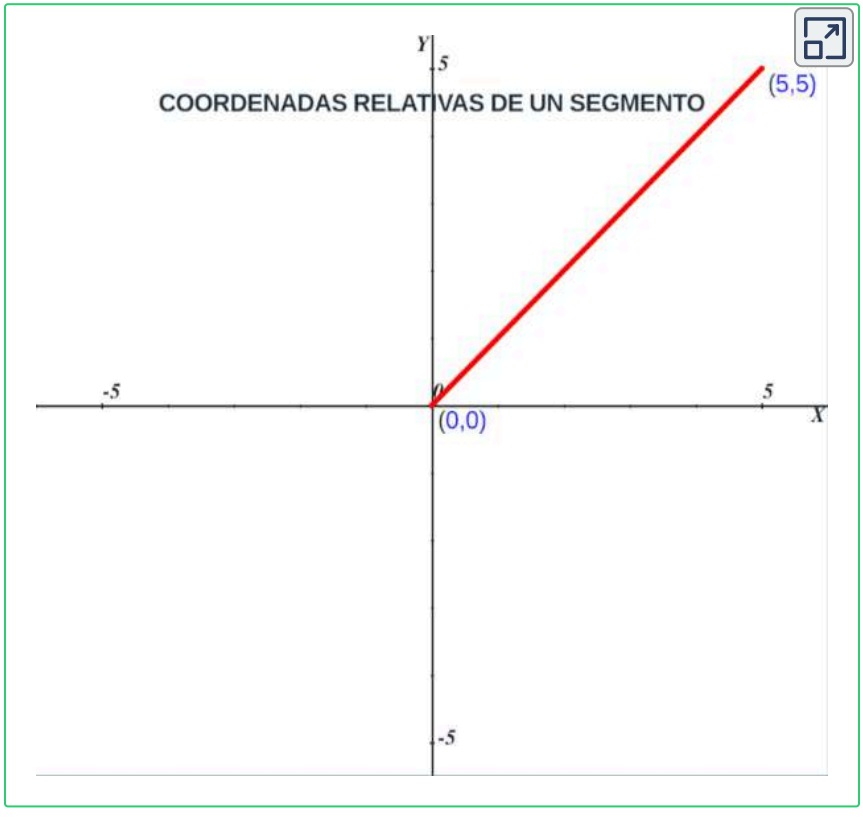




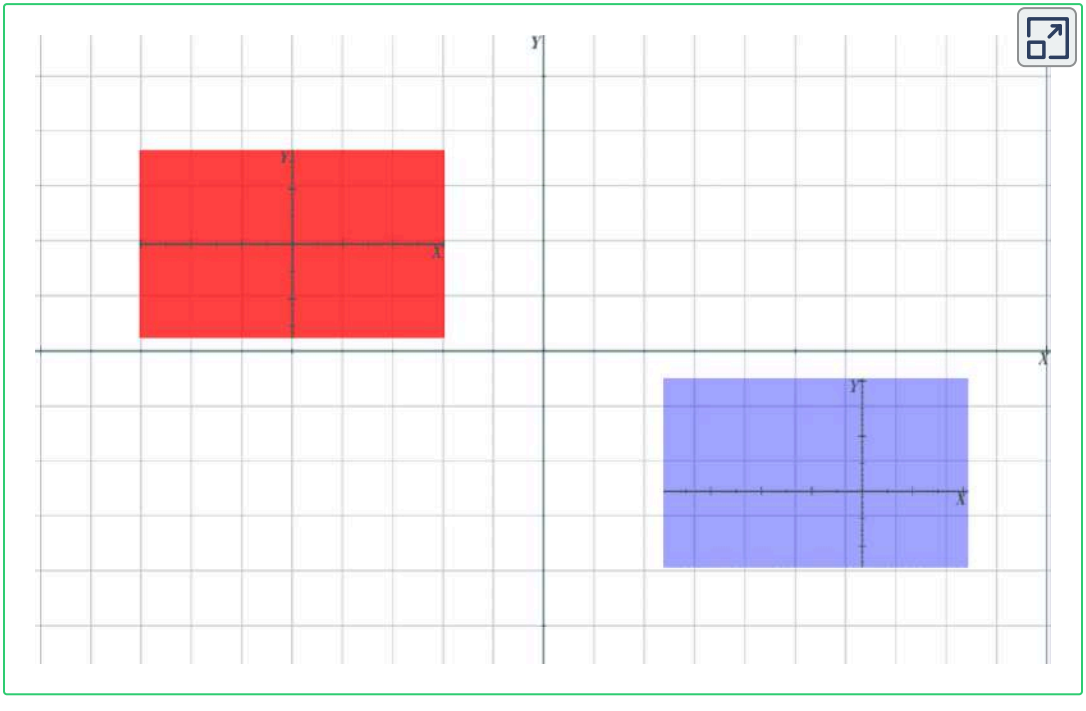
Si se desmarca el control **"coord_abs"**, la posición del objeto se definirá mediante un sistema de coordenadas relativas, definido en el espacio que contiene el elemento.

Seguidamente se muestra un segmento definido mediante coordenadas relativas, con el control **"coord_abs"** desactivado. En la escena puede observarse que el origen de coordenadas se sitúa en el centro del espacio, de modo que la coordenada X es positiva hacia la derecha y negativa hacia la izquierda, y la coordenada Y es positiva hacia arriba y negativa hacia abajo.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
【(0,0)(5,5)】
【(300,40)】
【(0,0)】
【(5,5)】
info
espacio E1
fondo
color
rastro
dibujar si
coord abs
expresión (0,0)(5,5)
familia
parámetro s
intervalo [0,1]
pasos 8
texto
fuente SansSerif
tam fuente 18
negrita
cursiva
decimales 2
fijo
tamaño 2
ancho 4
estilo de línea solida
Aceptar
Cerrar
Aplicar


COORDENADAS RELATIVAS DE UN SEGMENTO
Y
5
(5,5)
-5
0
5
X
(0,0)
-5

La posición de los distintos espacios dentro de una escena —en caso de que exista más de uno— debe indicarse obligatoriamente mediante coordenadas absolutas, tomando como sistema de referencia el de la propia escena, cuyo origen se sitúa en la esquina superior izquierda. En la siguiente escena se han introducido dos espacios 2D dentro del espacio principal de la escena.



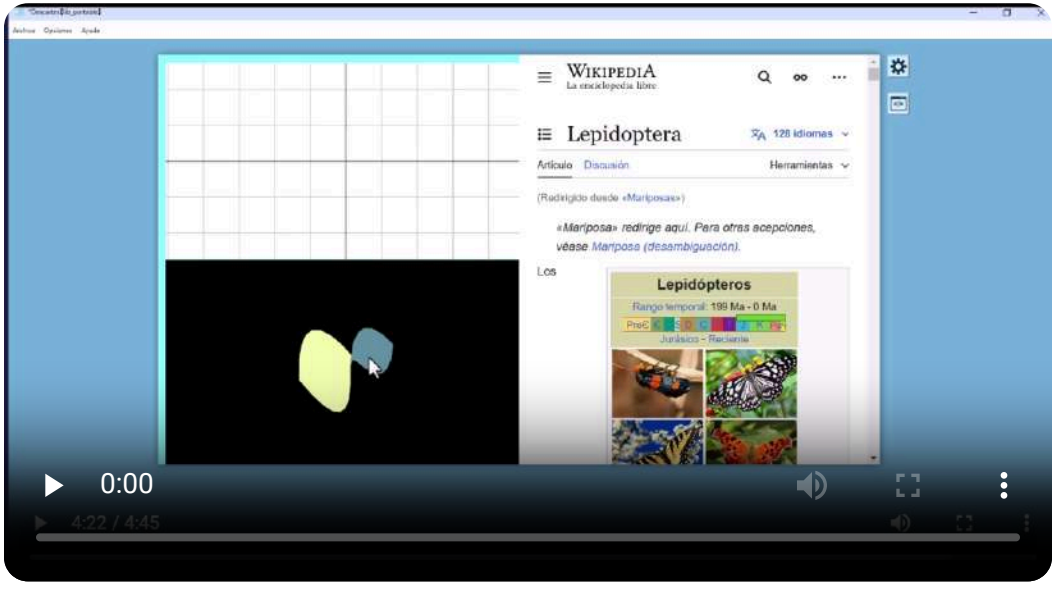
Es posible definir un espacio fuera de los límites visibles de la escena (total o parcialmente); en ese caso no se mostrará en pantalla, aunque podrá desplazarse posteriormente hasta quedar dentro del área visible, ya que la posición de los espacios se puede indicar mediante una variable.

Los espacios de DescartesJS admiten incluso un cambio dinámico de sus dimensiones. Para ello se deben introducir los parámetros **ancho** y **alto** mediante variables y tener activado el parámetro **redimensionable**.

2.3 Espacios en escenas

Una escena puede contener uno o más espacios. Para comprender mejor qué son los espacios, podemos usar el siguiente símil: imaginemos que cada espacio de una escena es una hoja de papel. Podemos tener hojas de diferentes tamaños, colocadas unas encima de otras, y cada hoja puede contener distintos escritos o elementos. Si observamos un montón de hojas apiladas desde arriba, veremos únicamente las hojas que están en la parte superior y que no están cubiertas por otras.

Como primera toma de contacto con los espacios en escenas, incluimos el siguiente vídeo explicativo, en el que se muestra cómo introducir los distintos tipos de espacios en las escenas y realizar su configuración básica. Su contenido se desarrollará con más detalle a lo largo de este libro.

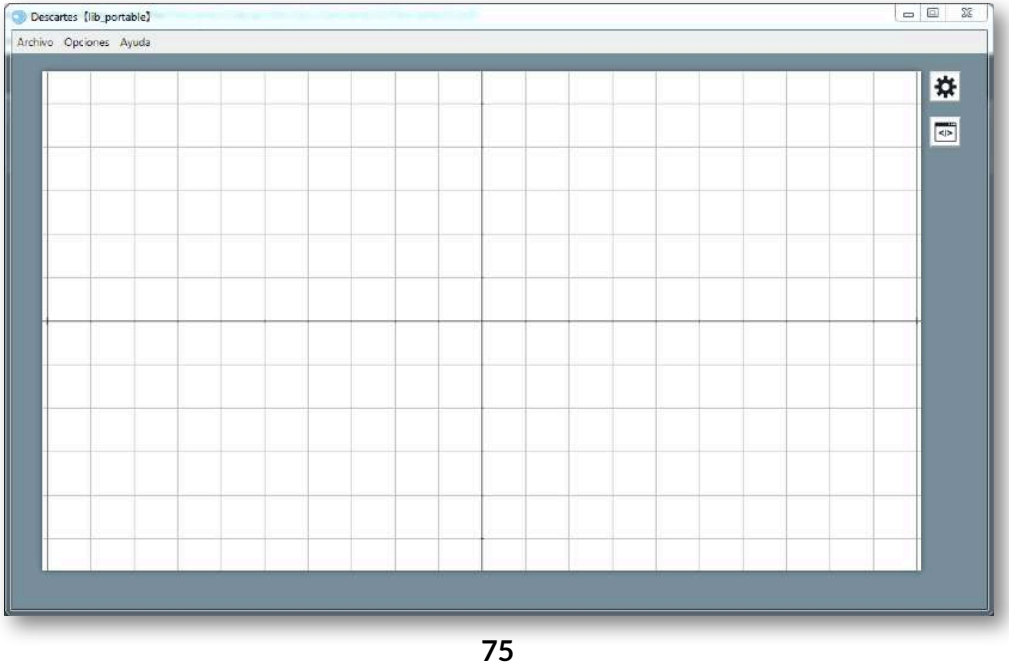


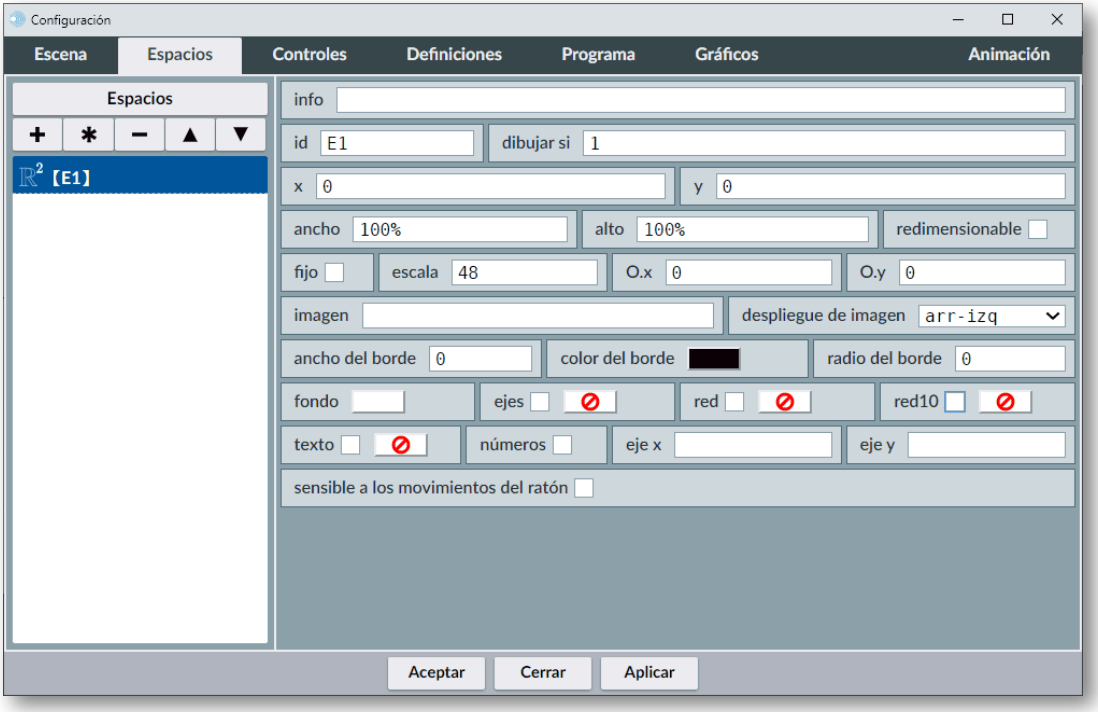
Vamos a realizar una analogía entre los espacios en las escenas de DescartesJS y las hojas de papel. Sus principales diferencias son las siguientes:

- Podemos definir condiciones para que un espacio aparezca o desaparezca según lo necesitemos.
- Los espacios pueden estar relacionados entre sí.
- Existen diferentes tipos de espacios: **2D** (bidimensionales), **3D** (tridimensionales) y **HTMLIFrame** (para mostrar ficheros con formato HTML).

Por defecto, al crear una nueva escena en el editor, esta contiene un único espacio rectangular en dos dimensiones que ocupa toda la escena y dispone de una rejilla de coordenadas. Este espacio inicial se denomina **E1** (todos los espacios deben llevar un nombre). Podemos trabajar modificando la configuración de este espacio inicial y, si lo deseamos o estimamos conveniente, añadir nuevos espacios según nuestras necesidades.

Para configurar un espacio, utilizamos el panel de configuración de la escena, activando el control **Espacio**. Al hacerlo, se mostrará la ventana donde se pueden definir los parámetros de los distintos espacios de la escena.





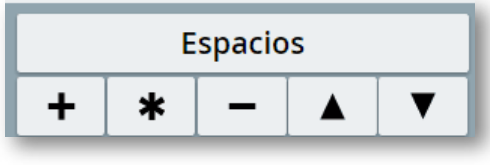
En primer lugar, comentaremos los principales parámetros que definen un espacio 2D: **nombre**, **posición**, **tamaño**, **escala**, **color de fondo**, **imagen de fondo**, **ejes de coordenadas**, **condiciones de dibujo**, entre otros.

- **Info:** Etiqueta que no se muestra en la escena y sirve al programador para identificar el espacio. También se muestra como descripción en la lista de espacios.
- **id:** Identificador del espacio. Los objetos colocados en este espacio tomarán este identificador.
- **Dibujar-si:** Permite indicar las condiciones para que el espacio aparezca o no en la escena.
- **Posición:** Ubicación del espacio respecto a la esquina superior izquierda de la escena, indicada en coordenadas absolutas en los campos X e Y. Si las coordenadas están fuera de la escena, el espacio no será visible.

- **Dimensiones:** Ancho y alto del espacio en coordenadas absolutas. El ancho se mide desde la posición X hacia la derecha y el alto desde la posición Y hacia abajo. Si el espacio se marca como **Redimensionable**, se puede ajustar su tamaño mediante variables, aunque no en porcentaje y puede superar el tamaño de la escena.
- **Fijo:** Impide que el espacio se desplace; si no está marcado, se puede mover con el ratón.
- **Escala:** Funciona como una "lupa", con valores de 0 (vista lejana) a 100.000 (vista muy ampliada). El valor por defecto es 48.
- **O.x y O.y:** Coordenadas absolutas que indican la distancia horizontal y vertical desde el centro del espacio hasta el origen de coordenadas relativas, permitiendo ubicar el origen en cualquier punto del espacio.
- **Imagen de fondo:** Permite añadir una imagen indicando la ruta del archivo (.png o .jpg). Las opciones de despliegue son:
    - **arr-izq:** arriba a la izquierda
    - **expandir:** ocupa todo el espacio
    - **mosaico:** repite la imagen para cubrir todo el fondo
    - **centrada:** coloca la imagen en el centro del espacio
- **Color de fondo:** Permite definir un color, hacer transparente o semitransparente el espacio para ver los espacios situados debajo.
- **Red y ejes y números:** Casillas que determinan si se muestran ejes, rejillas, números de coordenadas y nombres de los ejes en el espacio.
- **Texto:** Si está activado, al hacer clic izquierdo sobre el espacio se muestran las coordenadas relativas del punto seleccionado con el ratón.

- **Número:** Muestra los números en los ejes de coordenadas.
- **eje.x y eje.y:** Permiten definir los nombres que se mostrarán sobre los ejes de coordenadas.
- **sensible_a_los_movimientos_del_ratón:** Cuando está activado en espacios 2D o 3D, el programa detecta los movimientos del ratón sobre la escena y responde con acciones o eventos (por ejemplo, E1.mouse_x, E1.mouse_y o E1.mouse_clicked=1).
- **Borde o marco:** Permite definir la anchura y el radio de las esquinas del borde del espacio.

En segundo lugar, explicaremos cómo añadir, borrar o duplicar un espacio utilizando el menú situado a la izquierda del panel de configuración de espacios de la escena. El control **+** permite añadir un nuevo espacio, el botón **-** elimina el espacio seleccionado de la lista y el * duplica el espacio marcado. Las flechas ascendente y descendente sirven para mover el espacio seleccionado hacia arriba o hacia abajo en la lista. El botón **Espacios** permite acceder al código de programación de los distintos espacios definidos.

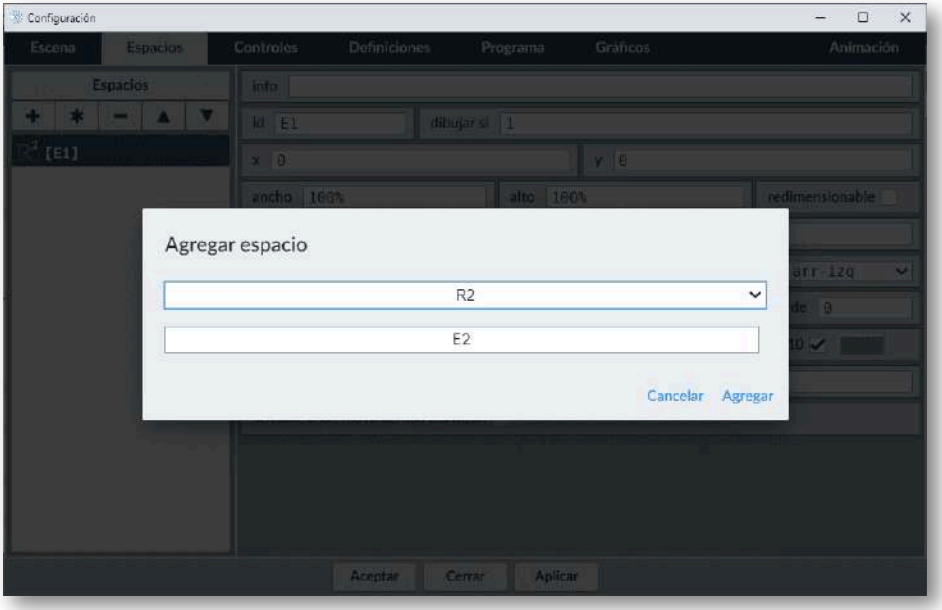


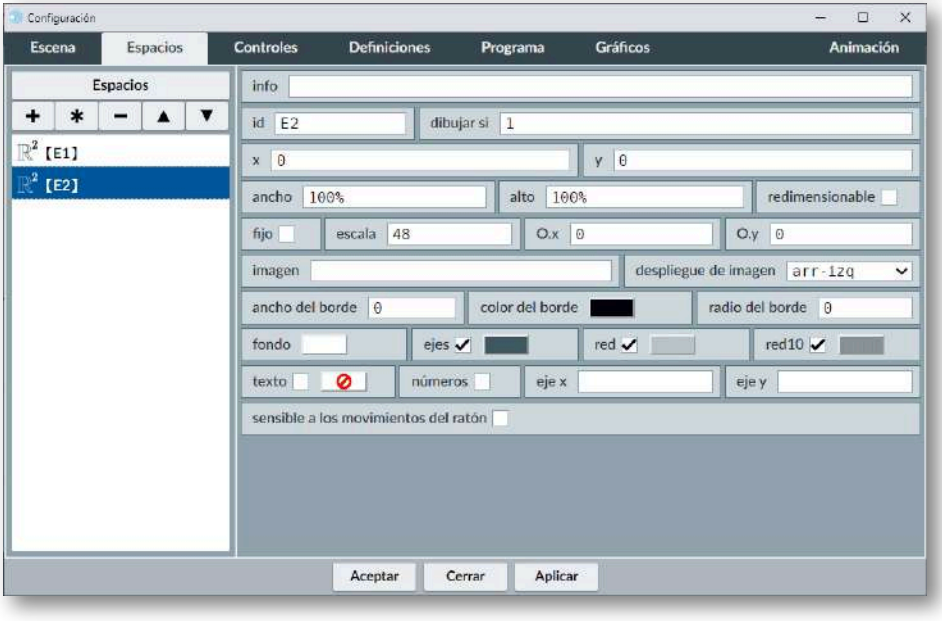
Es importante recordar que los espacios se dibujan en la escena siguiendo el orden de la lista: el último que aparece en la lista es el último que se dibuja, por lo que se mostrará encima de todos los demás.

A continuación, detallaremos cómo añadir y configurar un nuevo espacio.

Primero, pulsamos el botón **+** y se abrirá una ventana emergente donde indicaremos el tipo de espacio que queremos introducir: **R2** para un espacio bidimensional, **R3** para un espacio tridimensional, o **HTMLIFrame** para un espacio con contenido HTML.

También debemos asignar un nombre al espacio en el campo correspondiente. En nuestro ejemplo, seleccionaremos un espacio bidimensional (R2) y le asignaremos el nombre **E2**. Finalmente, pulsamos el botón **Aceptar** y el nuevo espacio se incluirá en la lista de espacios definidos, en dicha lista se mostrará su **id** y su **info**.

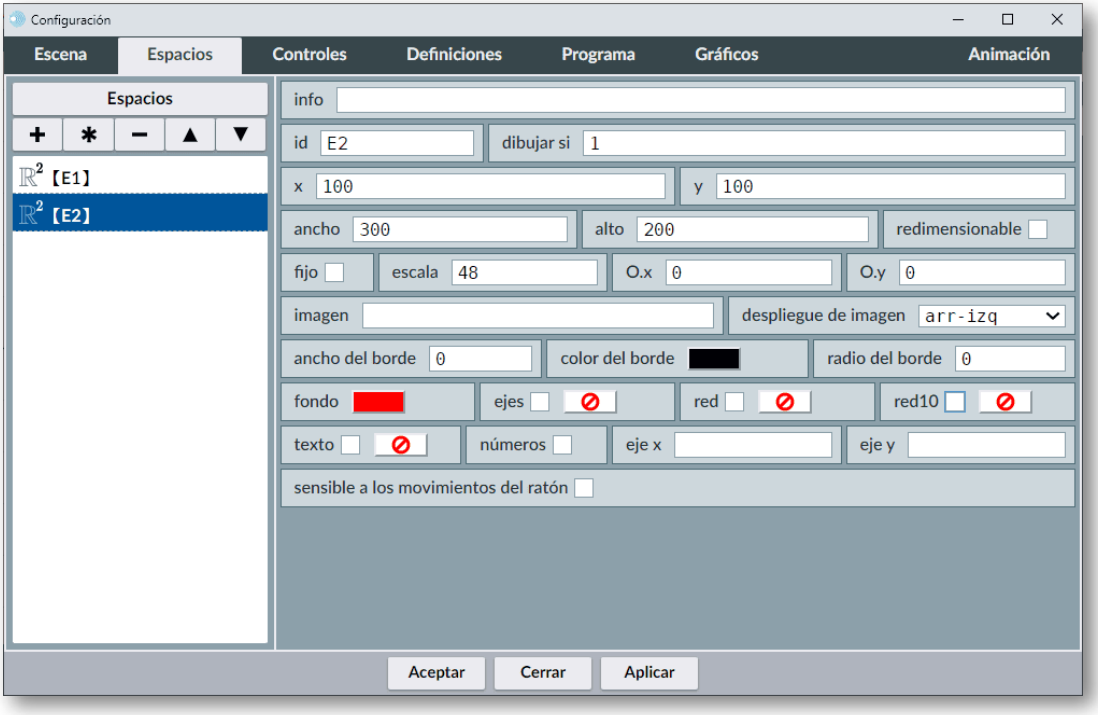


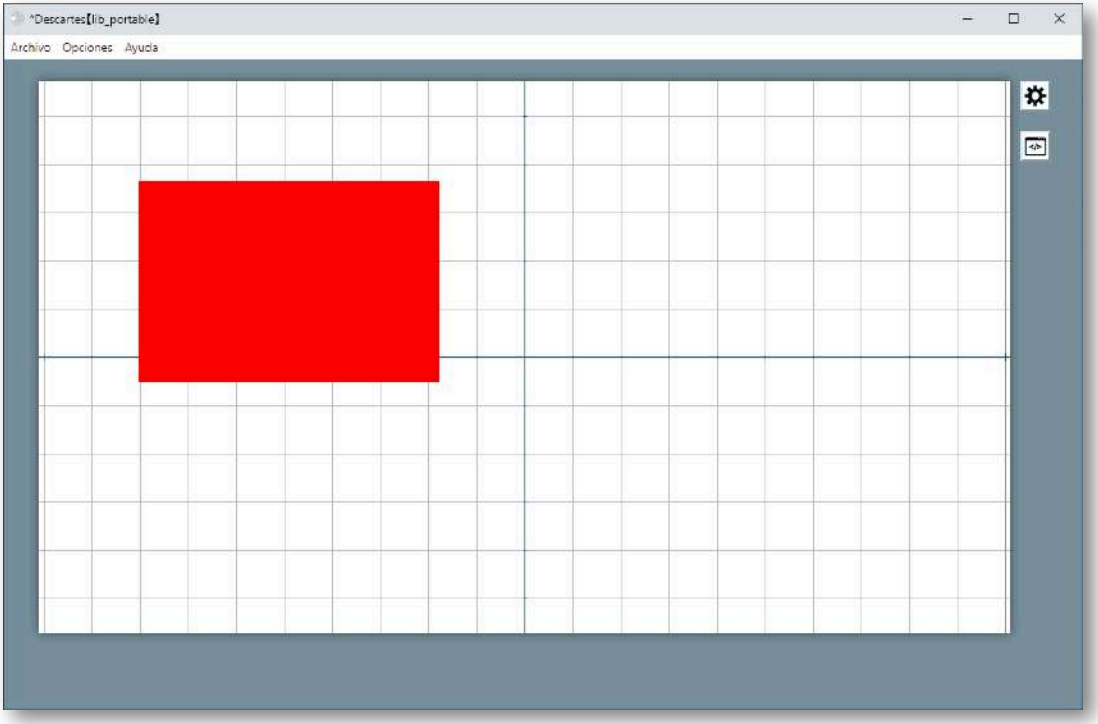


A continuación, podemos proceder a configurar algunos de los principales parámetros del espacio que hemos creado. Para ello, en la lista de espacios, pulsamos sobre el que deseamos configurar, en este caso el espacio identificado como **E2**.

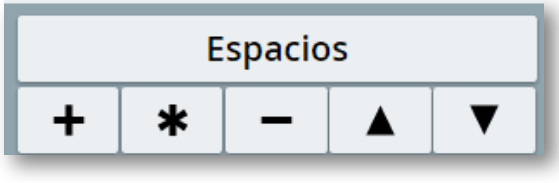
En nuestro ejemplo, la esquina superior izquierda del espacio **E2** se colocará en el punto (100,100) de la escena en coordenadas absolutas —es decir, a 100 píxeles de ancho y 100 píxeles de alto desde la esquina superior izquierda de la escena—. Sus dimensiones serán de 300 píxeles de ancho por 200 píxeles de alto (se podría indicar también mediante porcentaje relativo al espacio que lo contiene), el color de fondo será rojo, no tendrá ejes ni rejillas, y no se establecerán condiciones en **dibujar-si**.

En el editor de configuración se introducirán estos valores y, tras pulsar el botón **Aplicar** o **Aceptar** en el panel, la escena mostrará el espacio insertado.

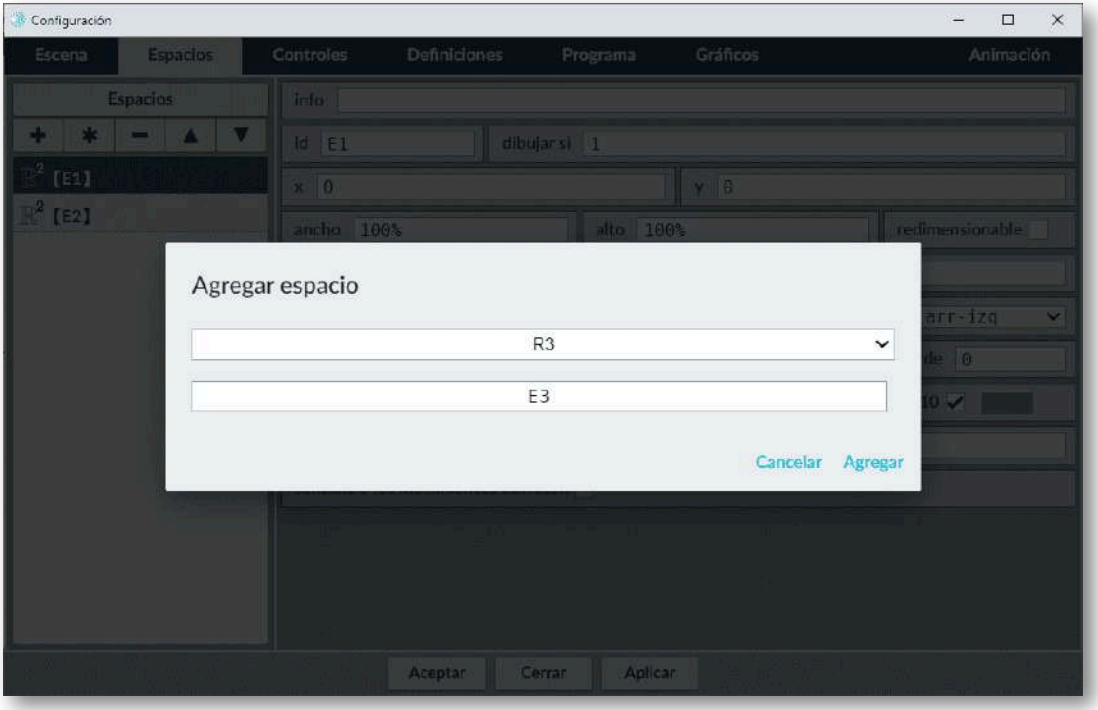


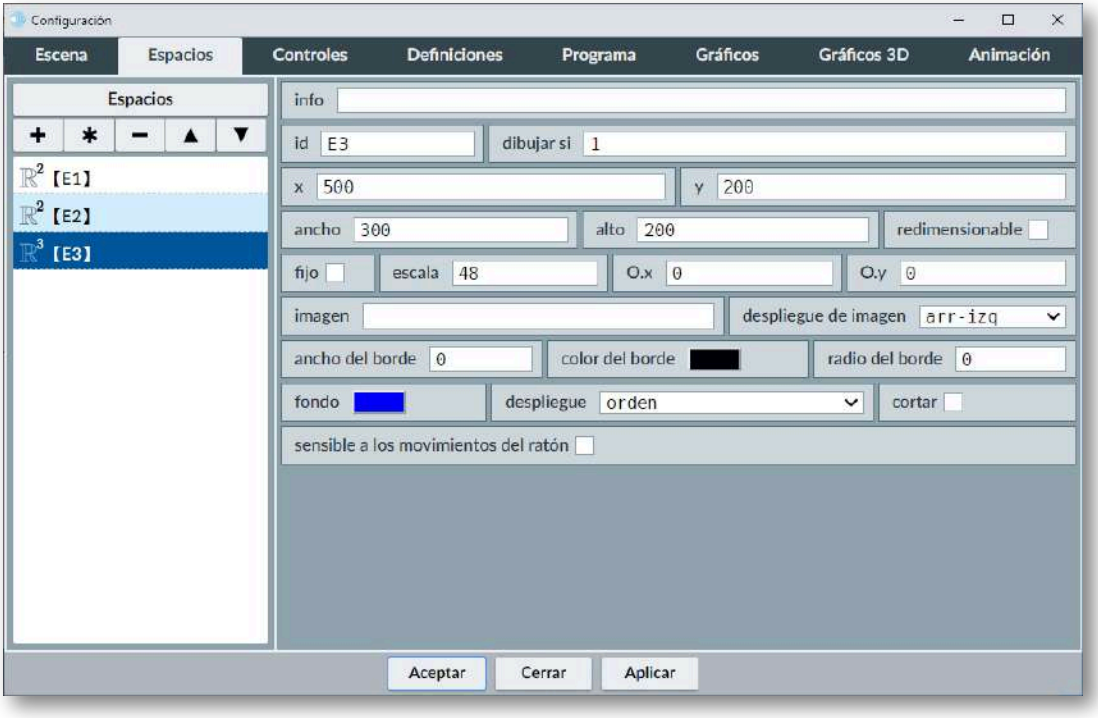


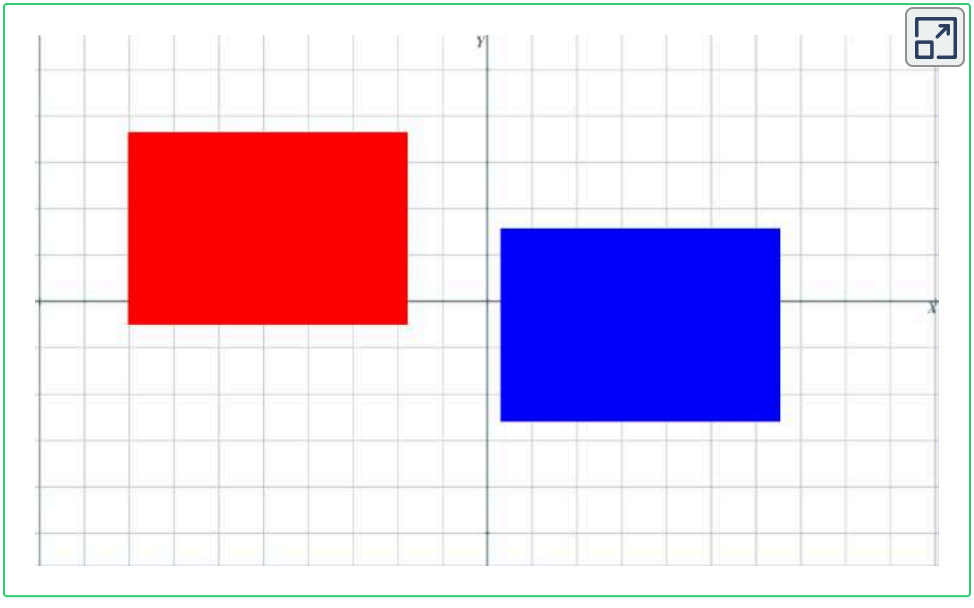
Si más adelante queremos modificar los parámetros de un espacio ya creado, debemos seleccionarlo en el menú **Espacios** a la izquierda del panel de configuración, cambiar los valores deseados y pulsar **Aplicar** o **Aceptar**. Para duplicar un espacio, nos situamos sobre él en el menú y pulsamos *, y para borrarlo usamos el botón -.



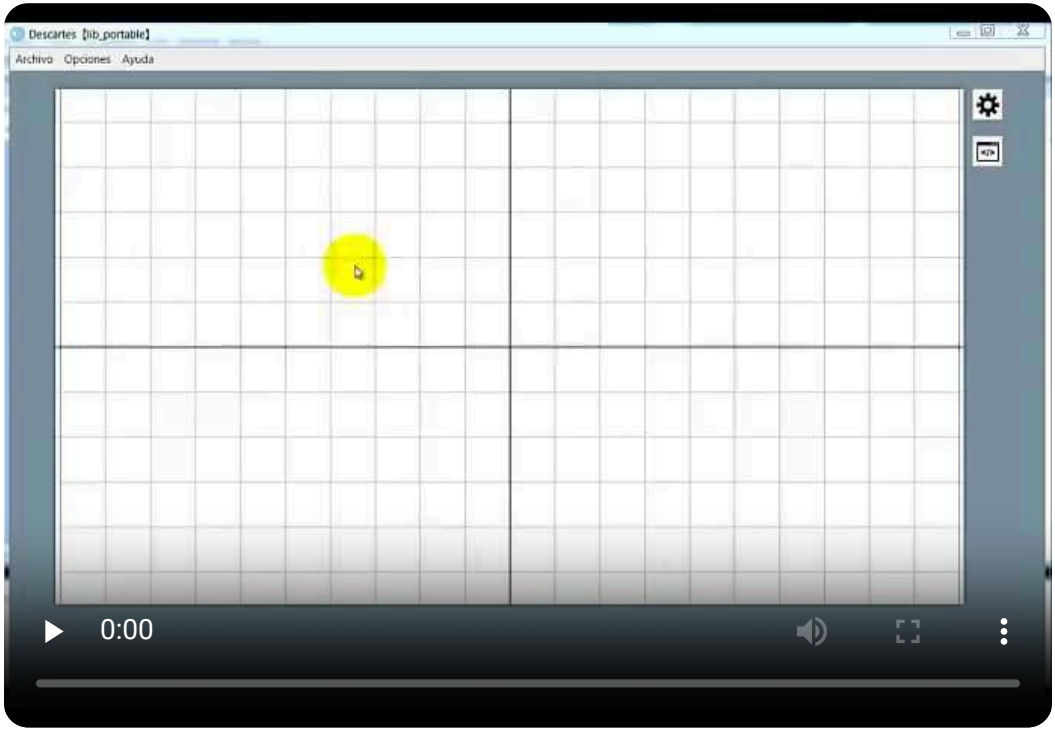
Finalmente, vamos a crear un nuevo espacio tridimensional (**R3**), con nombre **E3**, fondo azul, situado en la posición (500,200) y con dimensiones de 300 píxeles de ancho por 200 píxeles de alto. Pulsamos el botón **+**, indicamos el tipo y nombre del espacio en la ventana emergente, y configuramos los parámetros en los campos correspondientes. Como podemos observar, la configuración de espacios 2D y 3D es muy similar.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Espacios
info
id E1
dibujar si 1
x 0
y 0
ancho 100%
alto 100%
redimensionable
E1
E2
Agregar espacio
R3
E3
Cancelar
Agregar
Aceptar
Cerrar
Aplicar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Gráficos 3D
Animación
Espacios
E1
E2
E3
info
id E3
dibujar si 1
x 500
y 200
ancho 300
alto 200
redimensionable
fijo
escala 48
O.x 0
O.y 0
imagen
despliegue de imagen arr-izq
ancho del borde 0
color del borde
radio del borde 0
fondo
despliegue orden
cortar
sensible a los movimientos del ratón
Aceptar
Cerrar
Aplicar



En el siguiente vídeo se detalla la introducción de espacios en escenas.

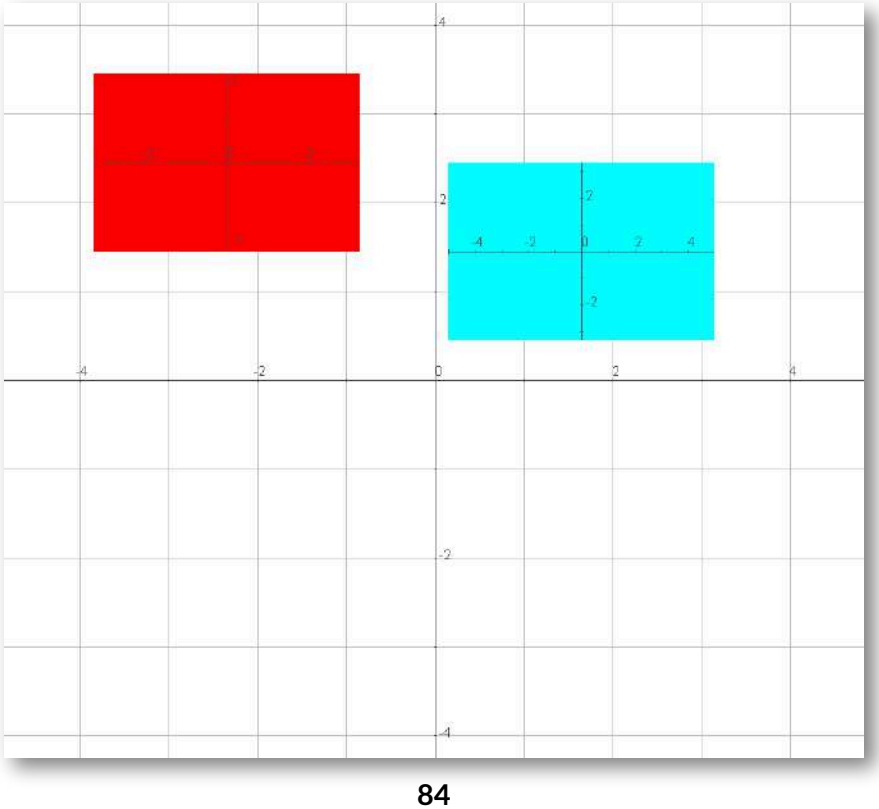


2.4 Coordenadas en espacios

Todo lo comentado sobre coordenadas en las escenas se aplica a todos los espacios, siendo especialmente útil en los espacios 2D. Así, para cada espacio se definen dos sistemas de referencia distintos: **absoluto** y **relativo**.

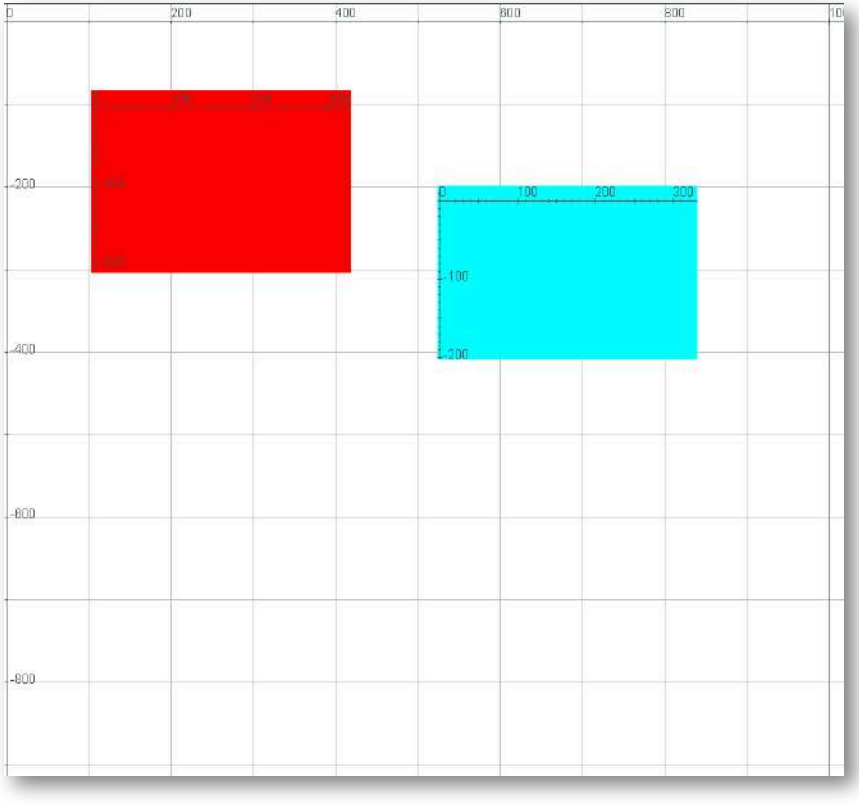
Relativo

En este sistema, el origen de coordenadas se sitúa en el centro de cada espacio (0,0). Los valores positivos de los ejes se extienden hacia arriba y hacia la derecha, mientras que los negativos van en sentido contrario. En la siguiente imagen de una escena se muestran los ejes de coordenadas relativas de cada uno de los espacios. Cuando un elemento se coloca en un espacio, su posición se establece de manera relativa al espacio que lo contiene.



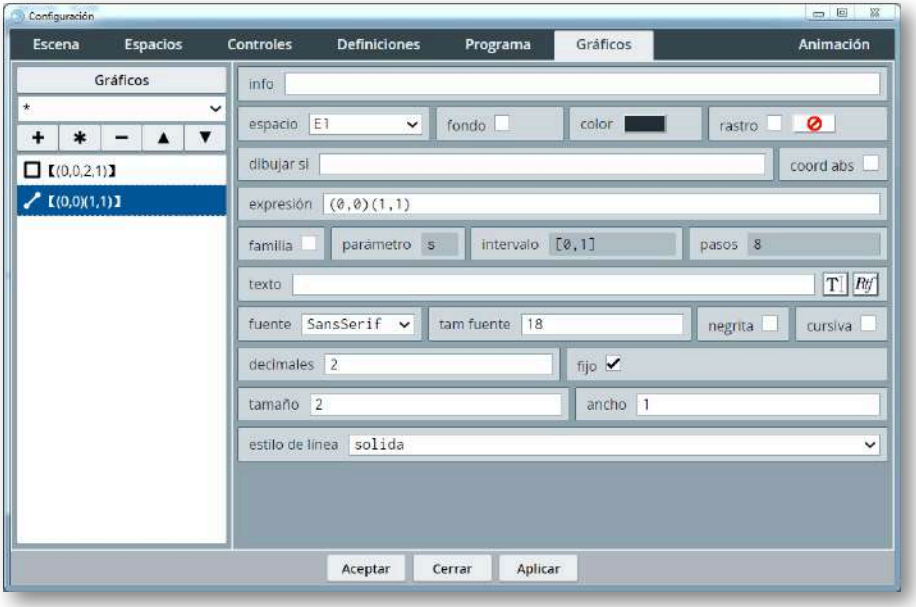
Absoluto

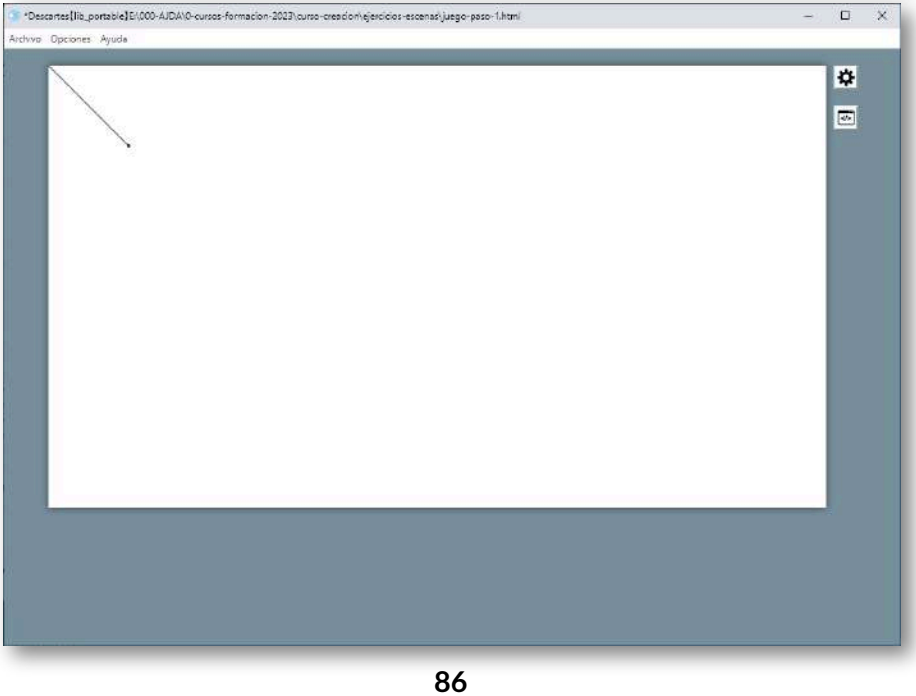
En el sistema absoluto, el origen de coordenadas de cada espacio se encuentra en su esquina superior izquierda (0,0). Desde este punto, la coordenada X aumenta hacia la derecha y la coordenada Y aumenta hacia abajo. En la siguiente imagen se indican los ejes de coordenadas absolutas de cada espacio. Al colocar un elemento en un espacio, su posición se define en relación con las coordenadas absolutas del espacio.



Hay objetos cuya posición debe definirse necesariamente en coordenadas absolutas y otros que admiten coordenadas absolutas o relativas respecto del espacio que los contiene, como segmentos, polígonos o flechas. Para los objetos que admiten ambos sistemas, se considerará el sistema absoluto si está activada la casilla **"coord_abs"** en el panel de configuración del objeto y relativo si no lo está.

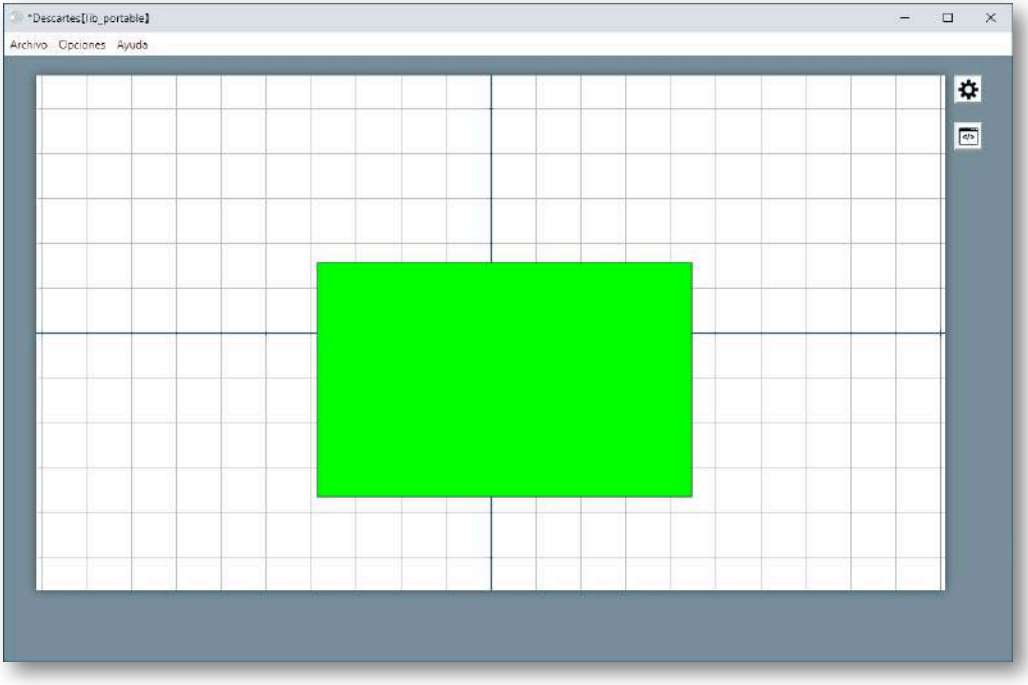
Cuando se inserta un objeto en un espacio, se debe indicar el nombre del espacio y la posición del objeto respecto a las coordenadas de dicho espacio. En el siguiente ejemplo, las coordenadas son relativas al espacio en el que se encuentra.





El rectángulo que se introduce en la siguiente escena está posicionado utilizando coordenadas absolutas.





2.5 Tipos de espacios

Según la naturaleza de los contenidos que se pueden introducir en los espacios, las escenas pueden contener los siguientes tipos de espacios:

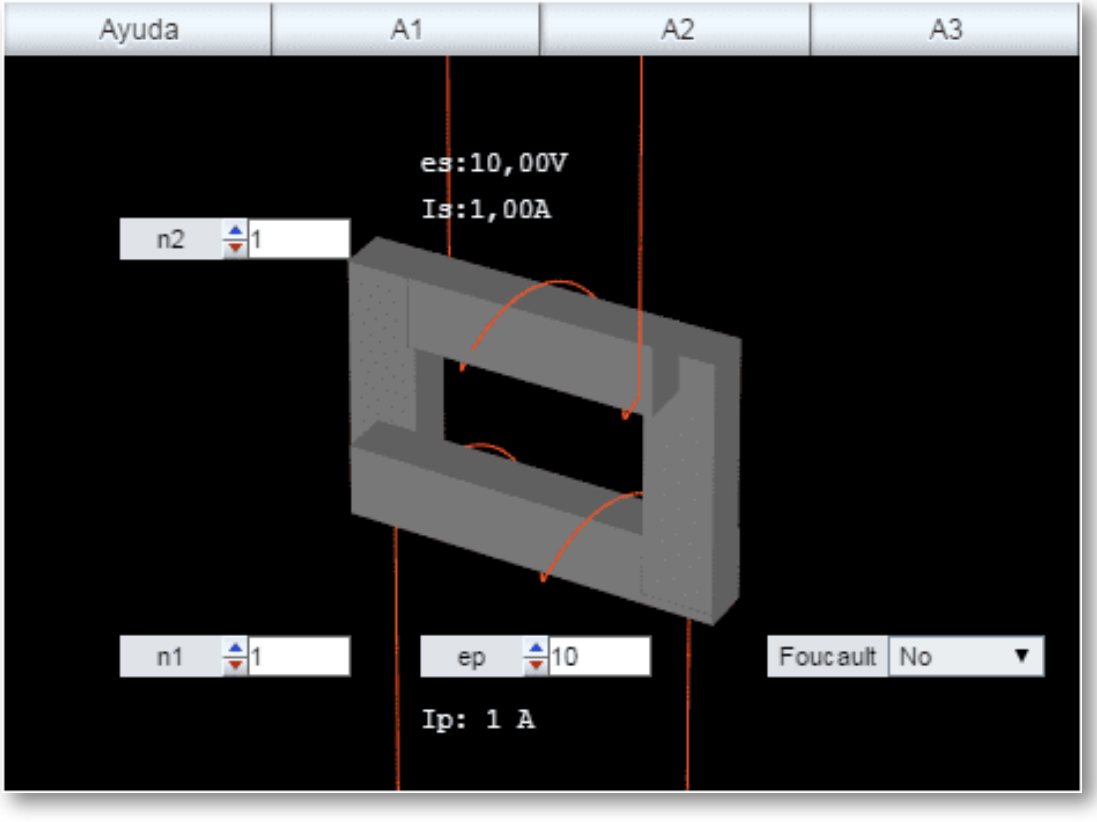
Espacios 2D (R2)

Estos espacios admiten elementos bidimensionales como textos, imágenes, puntos, ecuaciones, polígonos, segmentos, flechas, curvas, arcos, entre otros. Dedicaremos un capítulo específico a los elementos que se pueden introducir en los espacios 2D o R2. Un ejemplo de utilización de espacios 2D se encuentra en los juegos didácticos. A continuación, se muestra una captura del juego **"Duelo de magos"**.



Espacios 3D (R3)

Estos espacios permiten representar elementos tridimensionales como textos, figuras geométricas, puntos, curvas, superficies y caras. Dedicaremos un capítulo a los elementos que se pueden introducir en los espacios 3D o R3. Como ejemplo, se presenta una captura de una imagen 3D de un circuito de corriente alterna.

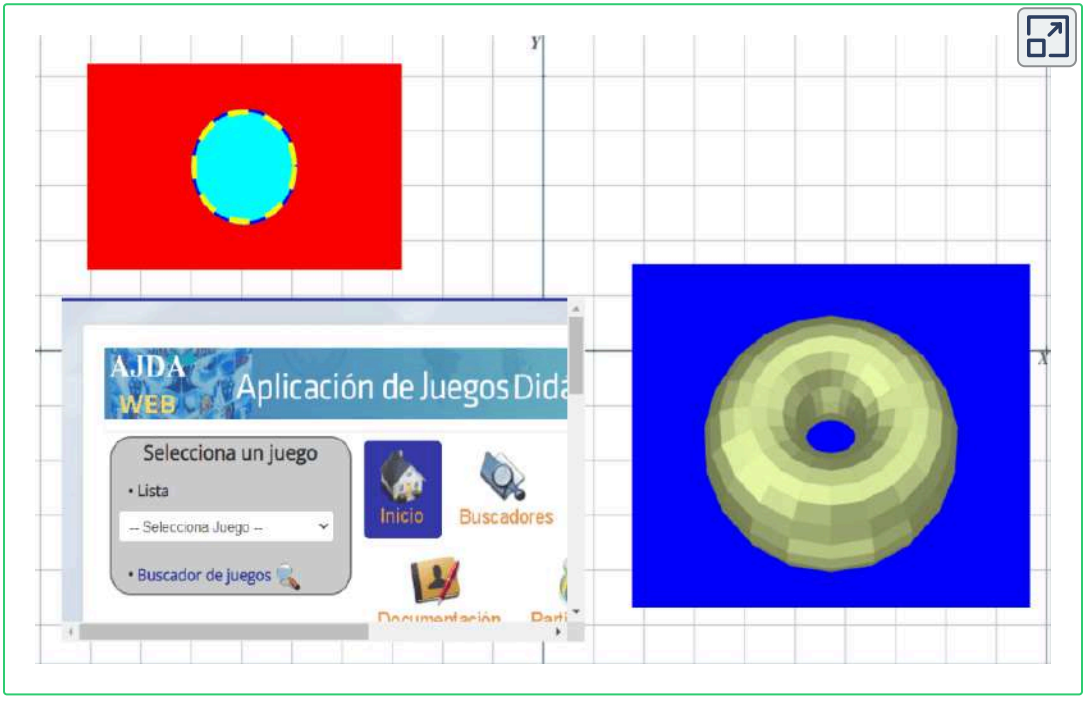


Espacios HTMLIFrame (web)

Estos espacios permiten incluir cualquier contenido de una página web o fichero HTML, como ficheros de sonido, vídeos, imágenes, GIF animados, páginas web, entre otros. En la siguiente captura se muestra una escena que contiene un espacio HTMLIFrame con un vídeo de la serie **"Águila Roja"**.

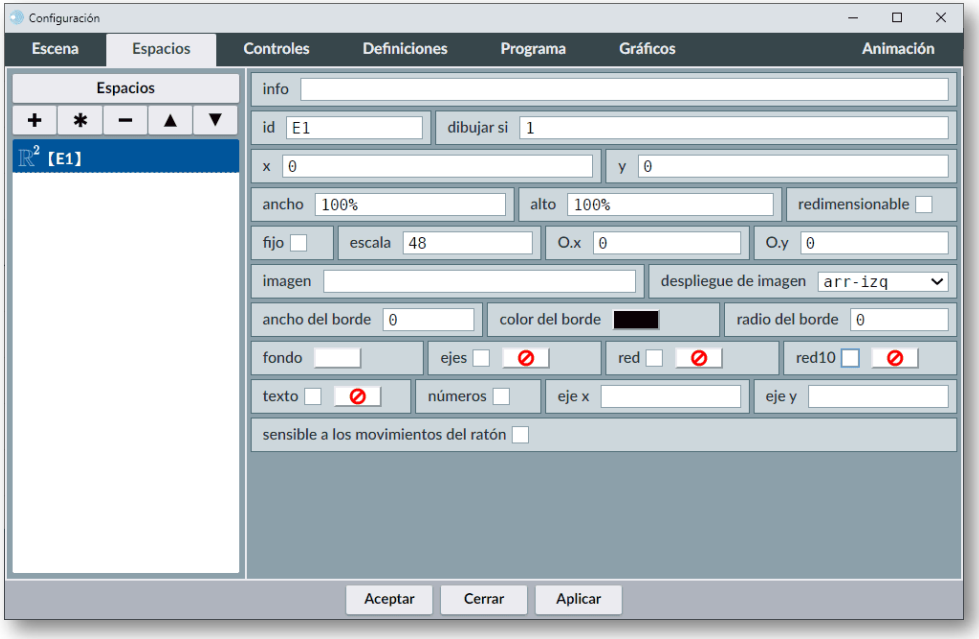


En la siguiente escena se han introducido espacios de los tres tipos diferentes que admite DescartesJS.



2.6 Espacios 2D

Los espacios 2D cuentan con los siguientes parámetros:

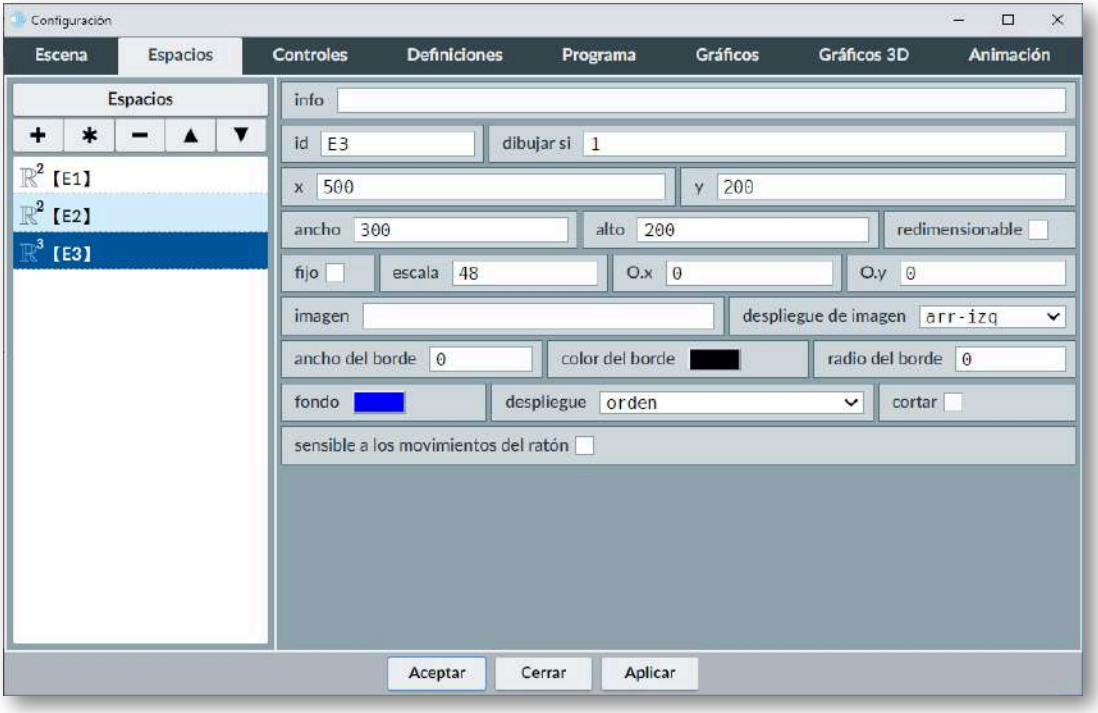


- **Info:** Etiqueta que no se presenta en la escena y sirve al programador para identificar el espacio.
- **id:** Identificador del espacio. Los objetos colocados en este espacio tomarán este identificador.
- **Dibujar-si:** Permite indicar las condiciones para que el espacio aparezca o no en la escena.
- **Posición:** Ubicación del espacio respecto a la esquina superior izquierda de la escena, indicada en coordenadas absolutas en los campos X e Y.
- **Dimensiones:** Ancho y alto del espacio en coordenadas absolutas. El ancho se mide desde la posición X hacia la derecha y el alto desde la posición Y hacia abajo. Si el espacio se marca como **Redimensionable**, el ancho y alto pueden variar, incluso saliendo fuera de la escena, pero no deben expresarse en %.

- **Fijo:** Hace que el espacio no se pueda desplazar. Si no está marcado, se puede mover con el ratón.
- **Escala:** Funciona como una "lupa", desde 0 (vista lejana) hasta 100.000 (muy ampliado), siendo 48 el valor por defecto.
- **O.x y O.y:** Coordenadas absolutas que indican la distancia horizontal y vertical desde el centro del espacio hasta el origen de coordenadas relativas.
- **Imagen:** Permite introducir una imagen de fondo indicando la ruta del archivo (.png o .jpg). Las opciones de despliegue son:
    - **arr-izq:** arriba a la izquierda del espacio
    - **expandir:** escala la imagen para ocupar todo el espacio
    - **mosaico:** repite la imagen para cubrir todo el fondo
    - **centrada:** ubica la imagen en el centro del espacio
- **Color de fondo:** Define el color del espacio y permite hacerlo transparente o semitransparente, dejando ver los espacios situados debajo.
- **Red y ejes y números:** Indica si se mostrarán los ejes, la rejilla de coordenadas, los números y los nombres de los ejes.
- **Texto:** Al activarlo, al pulsar con el botón izquierdo sobre el espacio se muestran las coordenadas relativas del punto seleccionado.
- **Número:** Muestra los números sobre los ejes de coordenadas.
- **eje.x y eje.y:** Define los nombres que se mostrarán sobre los ejes de coordenadas.
- **Sensible a los movimientos del ratón:** Cuando está activado, el programa detecta los movimientos del ratón sobre la escena y responde con acciones o eventos (por ejemplo, E1.mouse_x, E1.mouse_y o E1.mouse_clicked=1).
- **Borde o marco:** Permite definir la anchura del borde y el radio para redondear las esquinas del espacio 2D.

2.7 Espacios 3D

Los espacios 3D cuentan con los siguientes parámetros:



- **Info:** Etiqueta que no se presenta en la escena y sirve al programador para identificar el espacio.
- **id:** Identificador del espacio. Los objetos colocados en este espacio tomarán este identificador.
- **Dibujar-si:** Permite indicar las condiciones para que el espacio aparezca o no en la escena.
- **Posición:** Ubicación del espacio respecto a la esquina superior izquierda de la escena, indicada en coordenadas absolutas en los campos X e Y.
- **Dimensiones:** Ancho y alto del espacio en coordenadas absolutas. El ancho se mide desde la posición X hacia la derecha y el alto desde la posición Y hacia abajo.

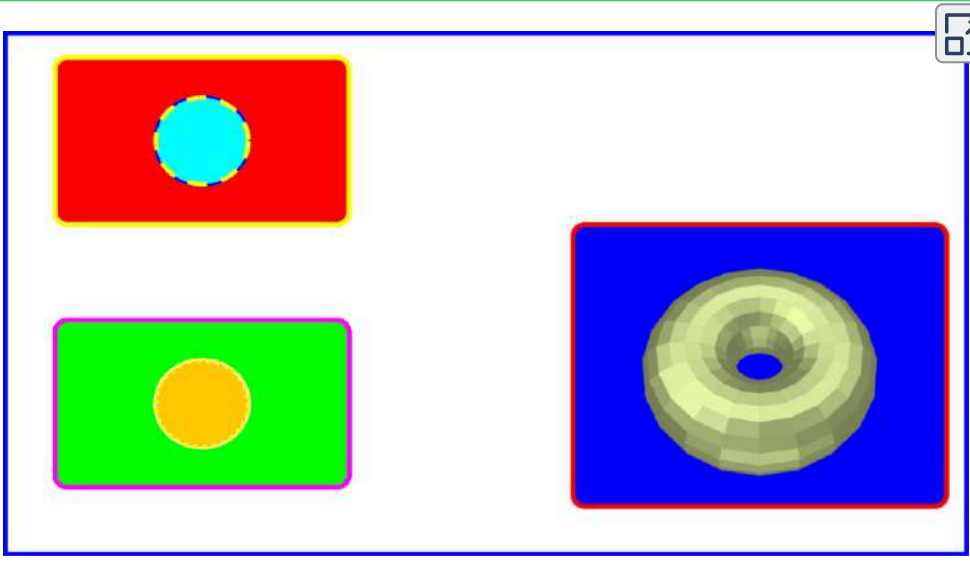
Si el espacio se marca como **Redimensionable**, el ancho y alto pueden variar, incluso saliendo fuera de la escena, pero no deben expresarse en %.

- **Fijo:** Hace que el espacio no se pueda desplazar. Si no está marcado, se puede mover con el ratón.
- **Escala:** Funciona como una "lupa", desde 0 (vista lejana) hasta 100.000 (muy ampliado), siendo 48 el valor por defecto.
- **O.x y O.y:** Coordenadas absolutas que indican la distancia horizontal y vertical desde el centro del espacio hasta el origen de coordenadas relativas, permitiendo ubicar el origen en cualquier punto del espacio.
- **Imagen:** Permite introducir una imagen de fondo indicando la ruta del archivo (.png o .jpg). Las opciones de despliegue son:
    - **arr-izq:** arriba a la izquierda del espacio
    - **expandir:** escala la imagen para ocupar todo el espacio
    - **mosaico:** repite la imagen para cubrir todo el fondo
    - **centrada:** ubica la imagen en el centro del espacio
- **Color de fondo:** Define el color del espacio y permite hacerlo transparente o semitransparente, dejando ver los espacios situados debajo.
- **Despliegue:** Permite seleccionar distintas formas de desplegar los objetos en 3D: **orden** es más rápido y **pintor** es más preciso.
- **Cortar:** Indica cómo se cortan las superficies de los cuerpos 3D en las intersecciones entre ellos.
- **Sensible a los movimientos del ratón:** Cuando está activado, el programa detecta los movimientos del ratón sobre la escena y responde con acciones o eventos (por ejemplo, E1.mouse_x, E1.mouse_y o E1.mouse_clicked=1).
- **Borde o marco:** Permite definir la anchura del borde y el radio para redondear las esquinas del espacio 3D.

En la siguiente escena presentamos diferentes imágenes de fondo de espacios introducidas tanto en espacios 2D como en espacios 3D.



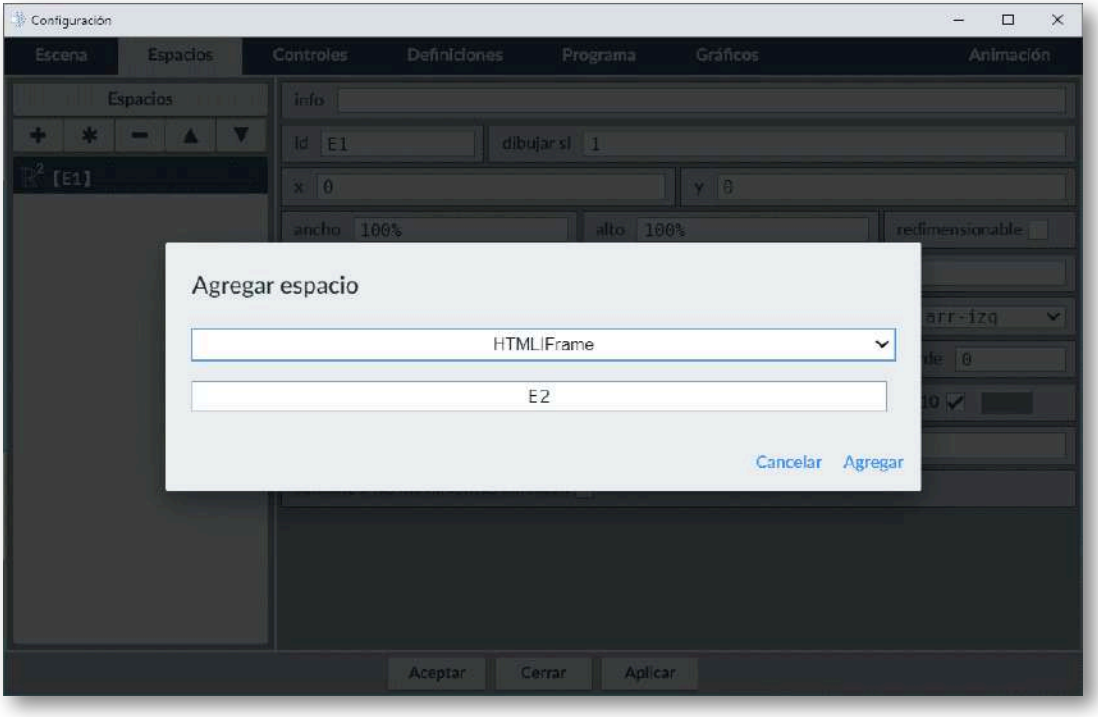
A continuación, se presentan diferentes espacios con bordes.



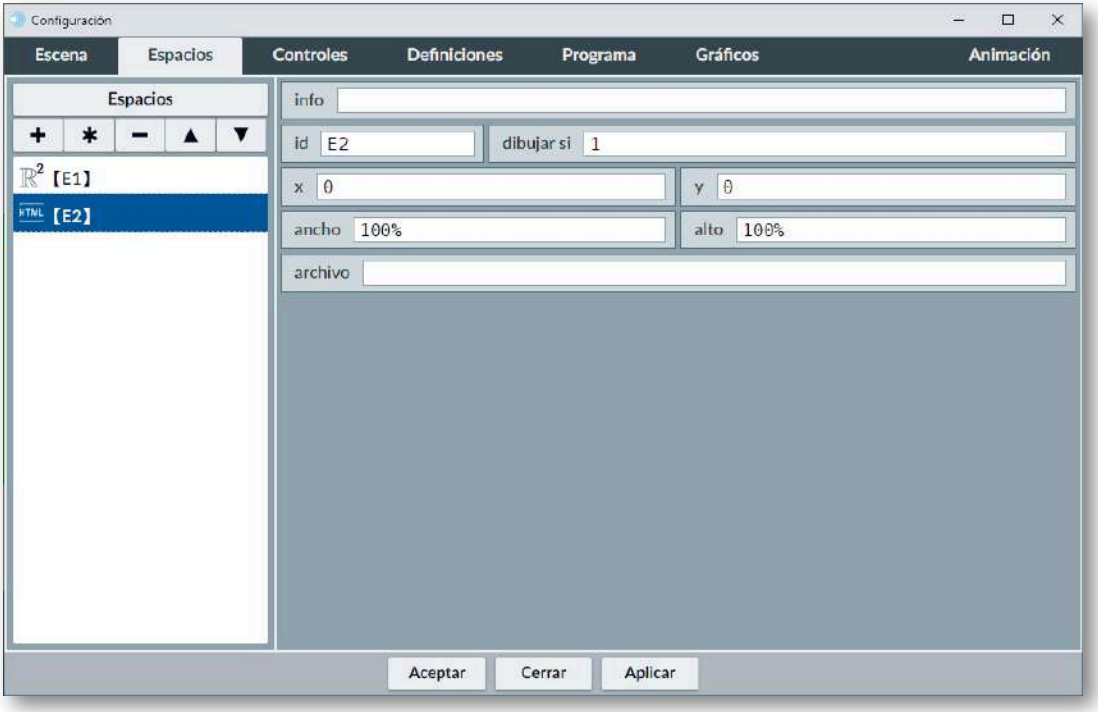
2.8 Espacios HTMLIFrame

Los espacios HTMLIFrame se utilizan para introducir en las escenas cualquier contenido que pueda contener un fichero HTML o una página web, como vídeos, audios, GIF animados, páginas web, entre otros. Estos contenidos pueden añadirse desde una página web o desde un archivo almacenado en nuestro ordenador.

Para crear un espacio HTMLIFrame, marcamos el campo **Espacio** en el menú superior del panel de configuración de la escena, pulsamos **+** en el menú izquierdo de espacios, y en la ventana emergente seleccionamos **HTMLIFrame** para introducir un espacio de este tipo. A continuación, le damos un nombre, por ejemplo, **E2**, y pulsamos **Aceptar**.



Hemos creado un espacio HTMLIFrame con el nombre **E2** (id). (el nombre se puede modificar si se desea).



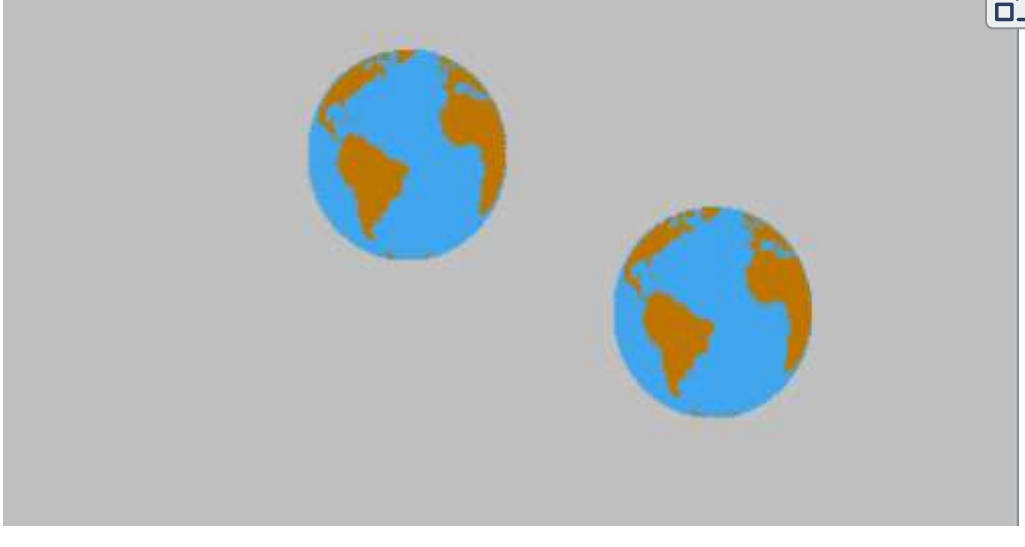
Ahora procedemos a configurar los diferentes campos y parámetros de este tipo de espacios:

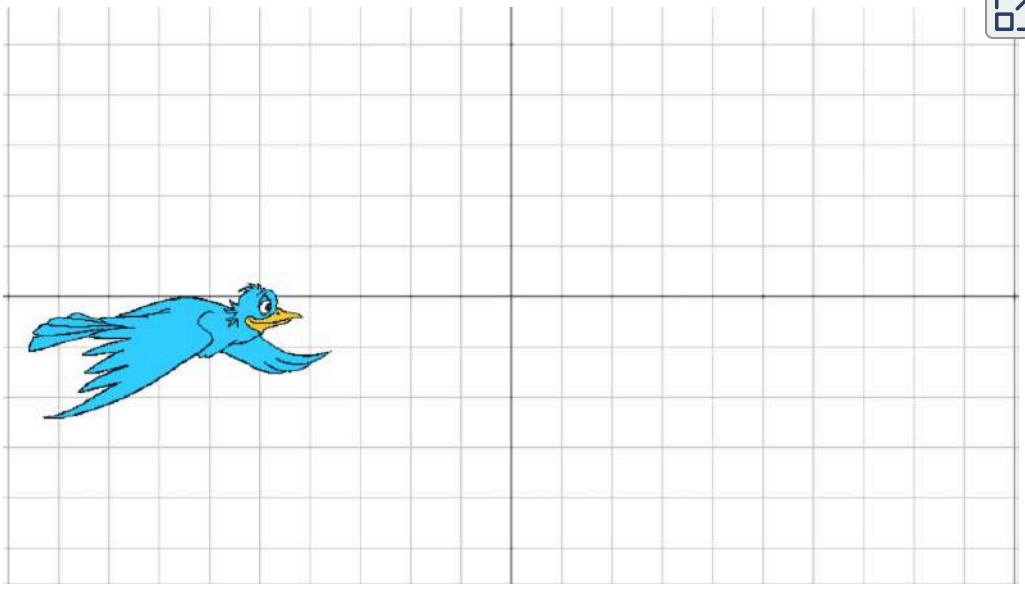
- **Info:** Etiqueta que no se muestra en la escena y sirve al programador para identificar el espacio. Esta información se muestra en la lista de espacios.
- **id:** Identificador del espacio. Los objetos colocados en este espacio tomarán este identificador.
- **Dibujar-si:** Permite establecer condiciones para que el espacio aparezca o no en la escena.
- **Posición:** Ubicación del espacio respecto a la esquina superior izquierda de la escena, indicada en coordenadas absolutas en los campos X e Y.
- **Dimensiones:** Ancho y alto del espacio en coordenadas absolutas. El ancho se mide desde la posición X hacia la derecha y el alto desde la posición Y hacia abajo.

- **Archivo:** Contenido que se mostrará en el espacio. Puede ser la dirección de una página web (por ejemplo, https://newton.proyectodescartes.org/juegosdidacticos/) o un archivo de nuestro ordenador, como vídeo, audio, GIF animado o fichero HTML.

A continuación, vamos a ilustrar algunos ejemplos más de usos de este tipo de espacios:

- **Audio o vídeo:** En el campo **archivo** se indica la ruta del fichero de audio o vídeo que se desea reproducir, por ejemplo, barco-a-venus.mp3 o ej-calculadora.mp4. Sin embargo, se recomienda introducir estos elementos mediante sus controles específicos, ya que permite un mayor control sobre la reproducción de los mismos.
- **Imagen o GIF animado:** Se indica en **archivo** la ruta del fichero que se quiere mostrar, por ejemplo, animado.gif. Vamos a mostrar dos ejemplos. En el primero el espacio en el que está el gif animado no tendrá asociado un desplazamiento pero en el segundo sí lo tendrá.





- **Páginas web o escenas locales:** Se puede insertar cualquier página web o escena con extensión HTML, ya sea de Internet o desde el equipo local.



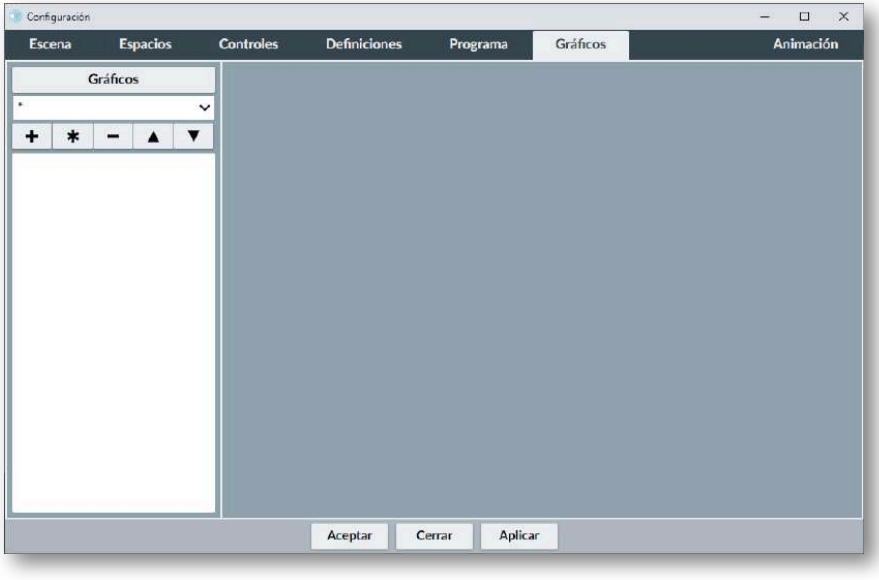


Es importante tener en cuenta que, cuando se incorpore en un espacio **HTMLIFrame** un archivo (vídeo, audio, página HTML, etc.) que no esté alojado en Internet —es decir, que no proceda de una dirección https://—, dicho archivo debe acompañar siempre a la escena y conservar su ubicación relativa. De lo contrario, al mover o publicar la escena, el recurso no podrá localizarse y no se cargará correctamente.

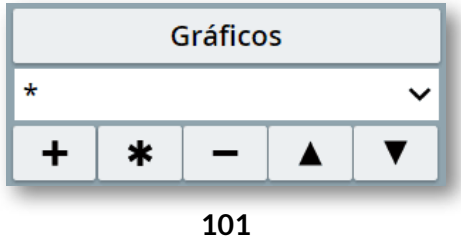
Si se trata de una página web, no es suficiente incluir solo el archivo principal (de extensión .html); deben añadirse también todos sus recursos asociados (CSS, imágenes, interactivos, audios, vídeos, scripts, etc.). Asimismo, es necesario mantener la misma estructura de carpetas para evitar errores en las rutas y asegurar su correcta visualización.

2.9 Imágenes en escenas

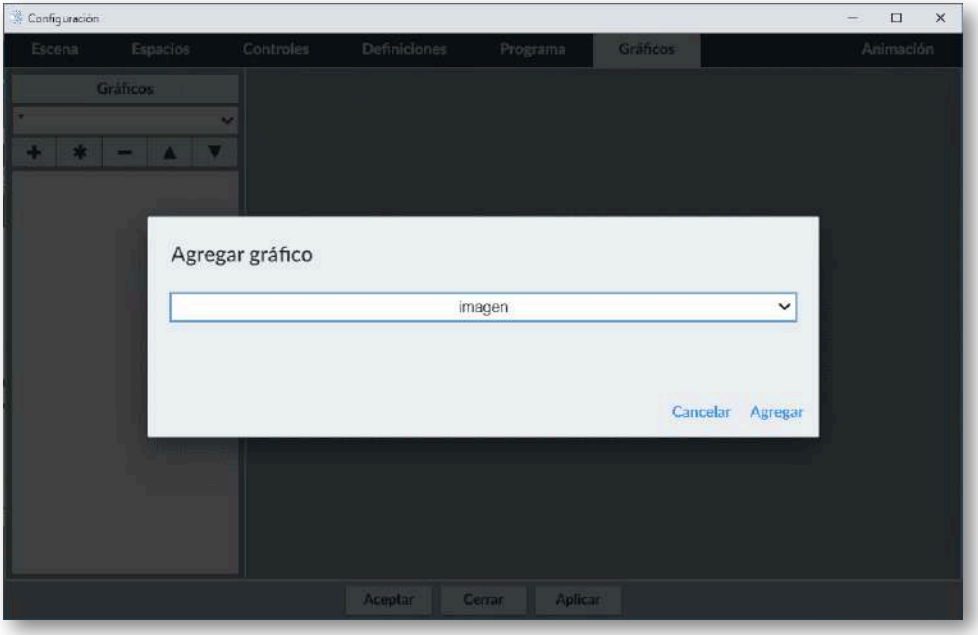
Para insertar una imagen en una escena —más concretamente, en un espacio de la escena—, primero debemos abrir el editor de configuración marcando el control **Gráficos**. Es recomendable guardar la escena antes de empezar a añadir imágenes; es decir, nada más abrir el editor conviene guardar la escena "en blanco" con el nombre deseado en la carpeta correspondiente de nuestro equipo.

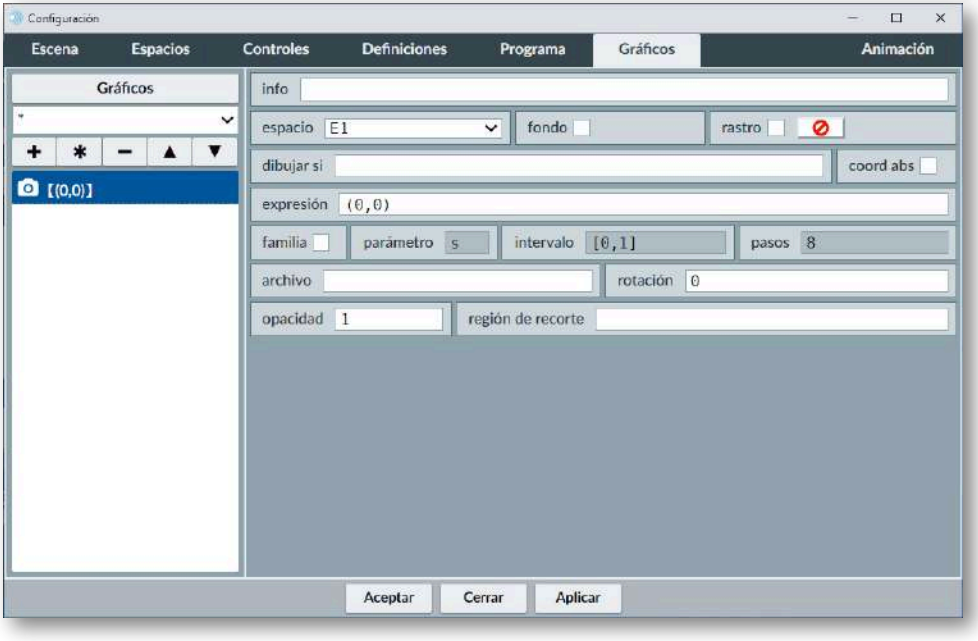


A continuación, utilizamos el menú situado a la izquierda, donde se irán incorporando los distintos elementos gráficos que añadamos a la escena. El control **+** permite crear un nuevo elemento, el botón **-** elimina el elemento seleccionado y el * lo duplica. Las flechas permiten modificar el orden de los objetos en la lista (los que están más abajo se dibujan por encima de los que están más arriba). El botón **Gráficos** muestra el código asociado a los elementos creados.



Al pulsar el botón **+**, se abre una ventana emergente donde debemos indicar el tipo de elemento que queremos introducir. En este caso, seleccionamos **Imagen** y, tras pulsar **Aceptar**, el nuevo elemento se añadirá a la lista de gráficos.





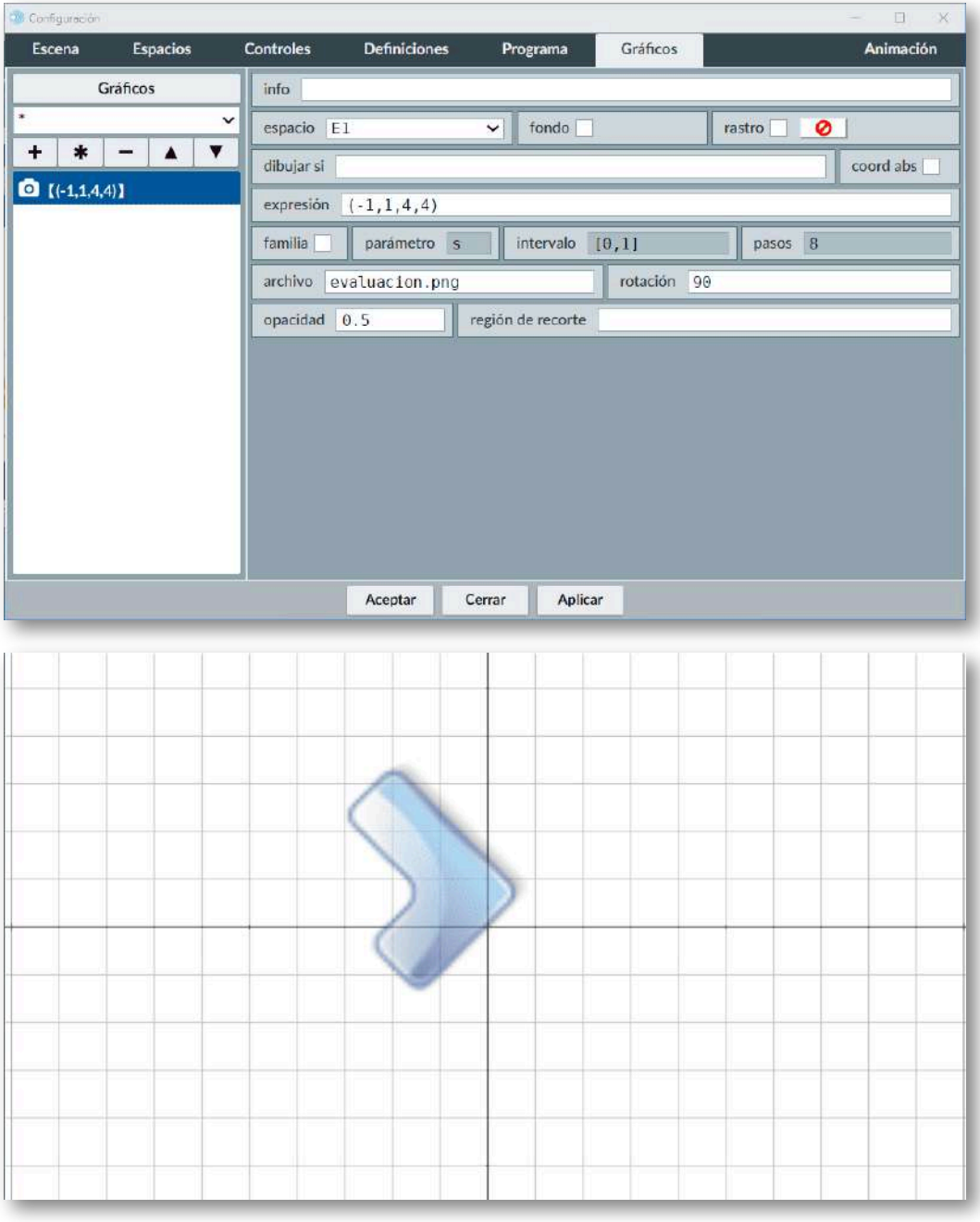
A continuación, mostramos los distintos parámetros de configuración de las imágenes en los campos del editor de DescartesJS. Comentamos los más utilizados:

- **Info:** Etiqueta informativa que no se muestra en la escena y sirve para identificar la imagen en el editor y en la lista de elementos gráficos.
- **Espacio:** La imagen debe colocarse dentro de un espacio de la escena. Aunque solo exista uno, es obligatorio indicar en qué espacio se insertará.
- **Fondo:** Si se activa esta opción, la imagen quedará fija como fondo. Es útil cuando no va a sufrir cambios ni desplazamientos, ya que optimiza el funcionamiento.
- **Rastro:** Permite que la imagen deje una huella visible cuando se desplaza por el espacio.
- **Dibujar-si:** Permite establecer las condiciones para que la imagen se muestre o no en la escena.
- **Coord_abs:** Indica si la posición se define mediante coordenadas absolutas o relativas.
- **Expresión:** Define la posición de la imagen dentro del espacio. Se pueden utilizar coordenadas relativas o absolutas. Si la opción **coord_abs** está desmarcada, se trabajará con coordenadas relativas. Por ejemplo:
    - **(0,0)**: sitúa la imagen en el centro del espacio.
    - **(0,0,1,1)**: los dos primeros valores indican la posición y los dos últimos la escala (1,1 equivale al 100% del tamaño original).
    - **(0,0,2,2)**: la imagen se muestra al doble de su tamaño original (aumento al 200%).
    - **(0,0,0.5,0.5)**: la imagen se muestra a la mitad de su tamaño (reducción del 50%).

- **Familia, parámetro, intervalo y pasos:** Permiten repetir la imagen varias veces según los valores indicados. Este apartado se desarrolla con más detalle en otra sección del libro.
- **Archivo:** Debe indicarse el nombre y la extensión del fichero de imagen. Si se encuentra dentro de una carpeta, hay que escribir su ruta relativa completa. Por ejemplo, si la imagen ejem.png está dentro de la carpeta images (situada al mismo nivel que la escena), se escribirá: images/ejem.png. Si la ruta no es correcta, la imagen no se mostrará. Por ello, es fundamental que los archivos de imagen acompañen siempre al archivo de la escena respetando la estructura de carpetas. Al copiar una escena, deben copiarse también sus imágenes asociadas.
- **Rotación:** Permite girar la imagen el número de grados indicado.
- **Opacidad:** Controla el nivel de transparencia, con valores entre 0 (totalmente transparente) y 1 (sin transparencia).
- **Región de recorte:** Mediante los parámetros **(x, y, w, h)** se define el rectángulo de la imagen que se presentará en la escena. Los valores **x** e **y** indican la posición inicial dentro de la imagen original, mientras que **w** y **h** determinan el ancho y alto del área visible.

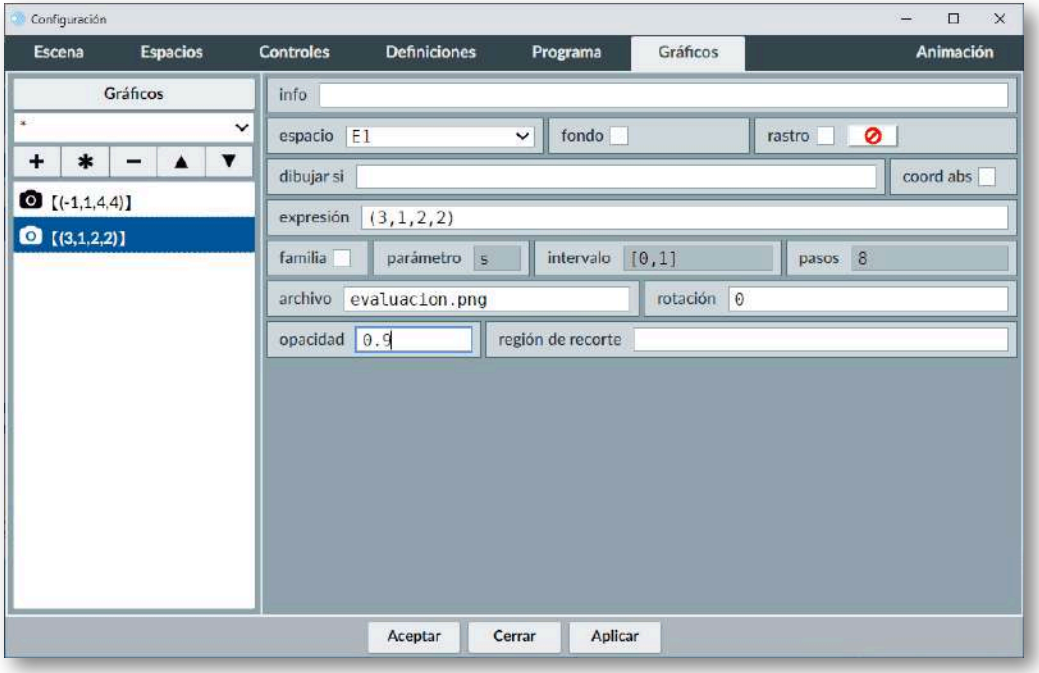
En nuestro ejemplo vamos a introducir una imagen cuatro veces mayor que la original, situada en las coordenadas (-1,1), rotada 90º, con el nombre **evaluacion.png**, semitransparente, ubicada en la misma carpeta que la escena y sin establecer ninguna condición en **dibujar-si**. En el editor de configuración se introducirán estos valores en los campos correspondientes.

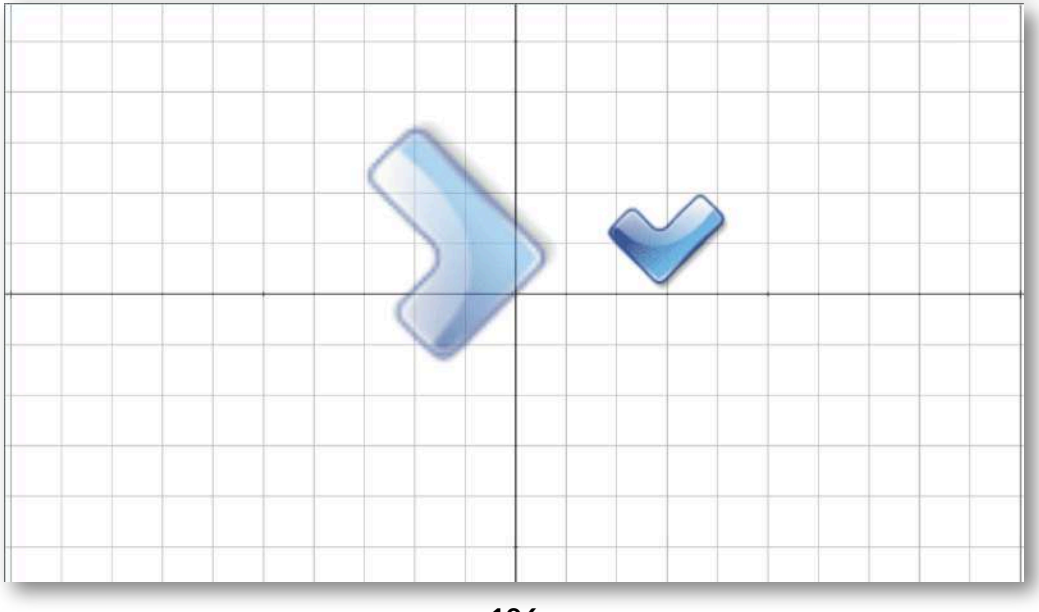
Tras pulsar el botón **Aplicar** o **Aceptar**, la escena mostrará la imagen insertada. Es importante recordar que el archivo **evaluacion.png** debe estar guardado en la misma carpeta que la escena para que pueda visualizarse correctamente.



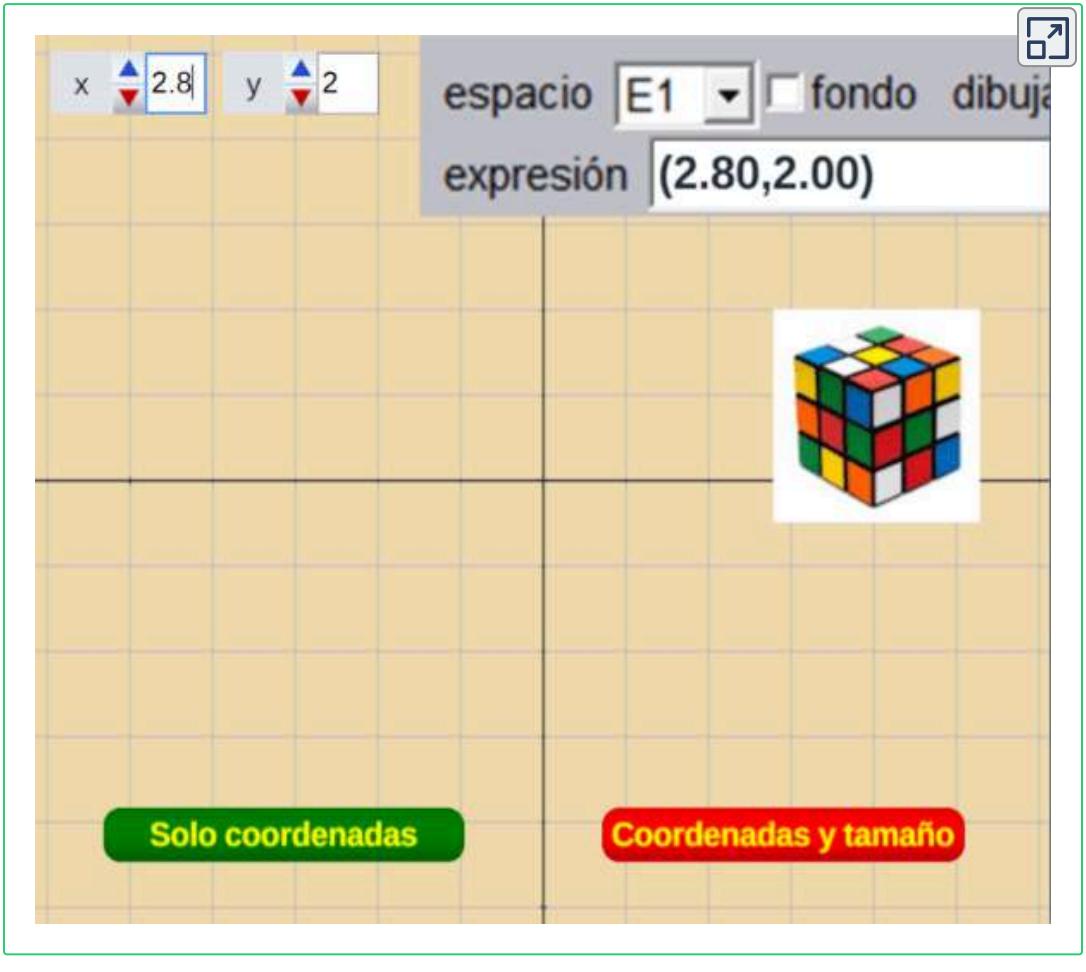
Si posteriormente queremos modificar los parámetros de la imagen añadida, debemos seleccionarla en el menú **Gráficos** situado a la izquierda del editor, cambiar los valores deseados y pulsar **Aplicar**.

Para duplicar la imagen, nos situamos sobre ella en el menú **Gráficos** y pulsamos *. Si queremos eliminarla, utilizamos el botón -. En nuestro caso, vamos a duplicar la imagen y, en la copia, modificaremos la posición, el tamaño y la rotación.

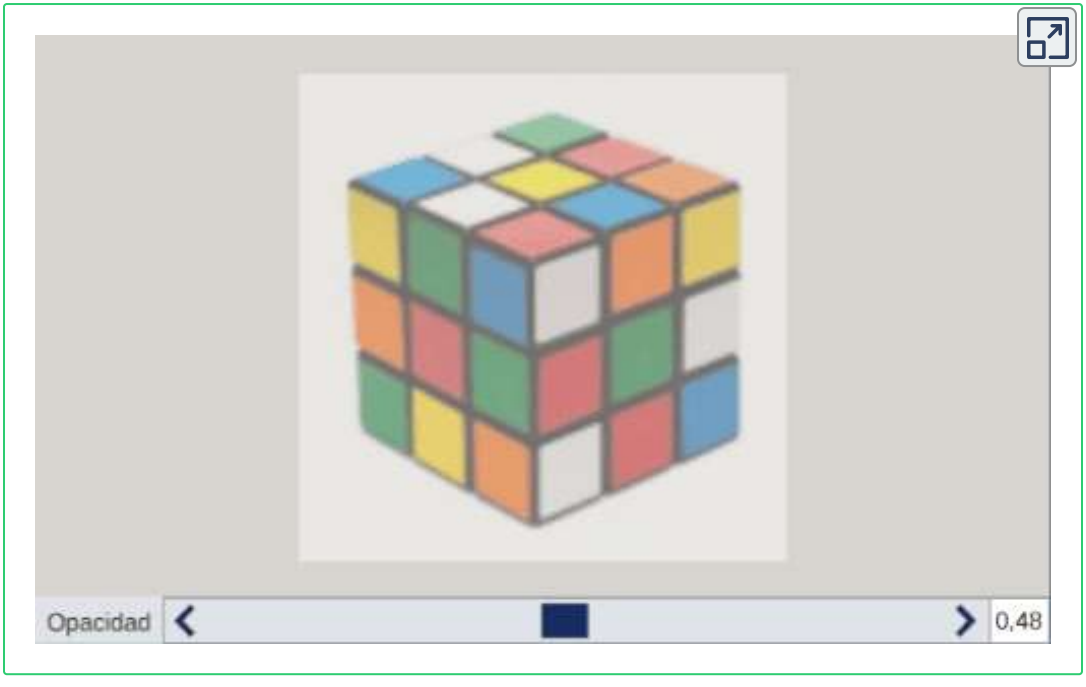




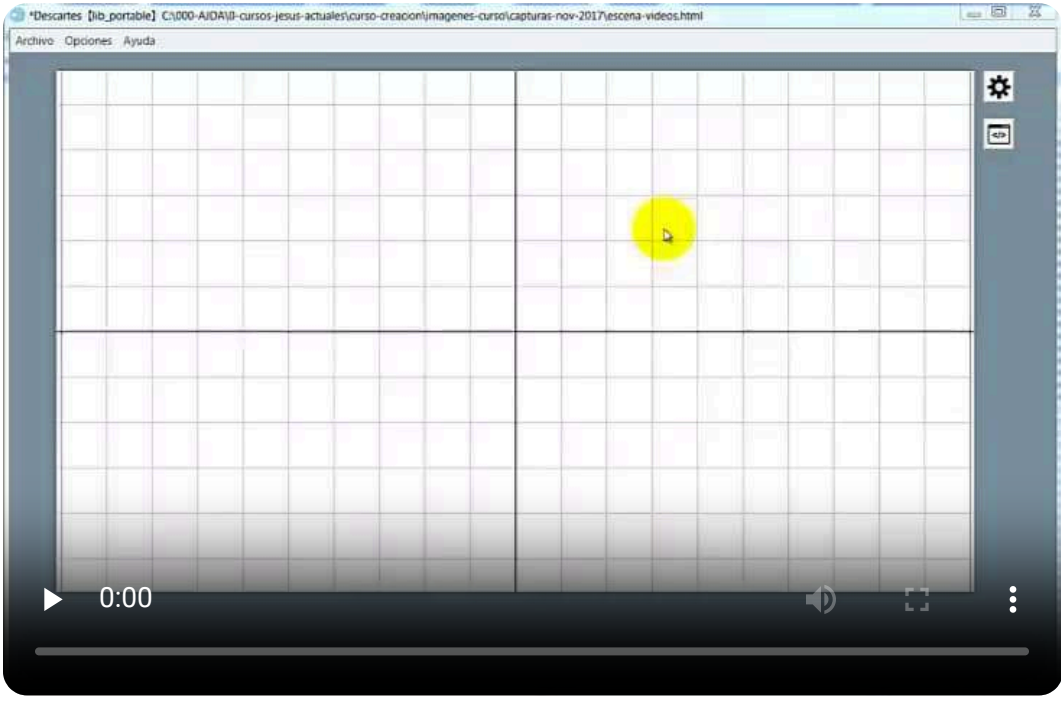
En la siguiente escena se puede experimentar cómo modificar la posición de una imagen en una escena, sus dimensiones y la rotación o giro de la misma. Además, se indica cómo escribir la notación de la posición y el tamaño en el campo **expresión** de la configuración de la imagen.



En el interactivo que se presenta a continuación se puede apreciar cómo utilizar la transparencia u opacidad de las imágenes en escenas de Descartes. Mediante un control tipo barra de deslizamiento se puede modificar el parámetro de opacidad de la imagen dentro del intervalo de cero a uno.

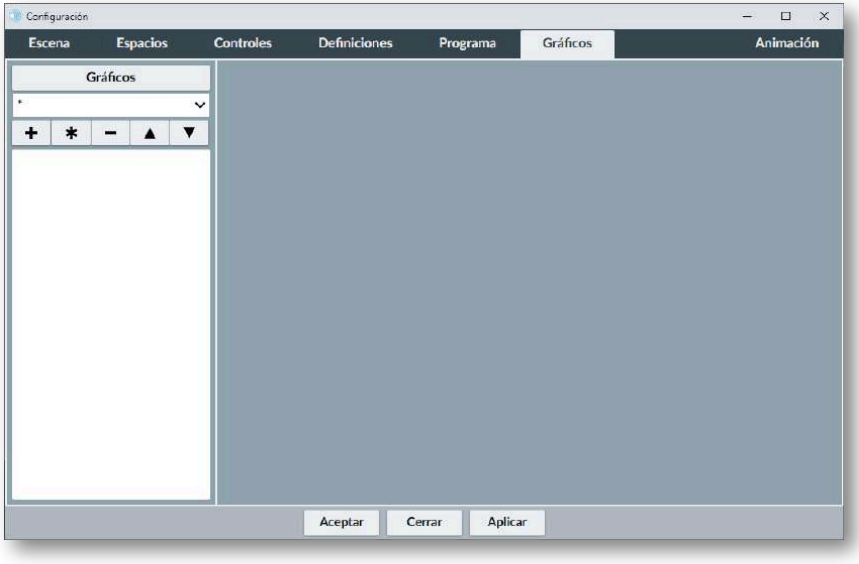


En el siguiente vídeo se ilustra la introducción de imágenes.

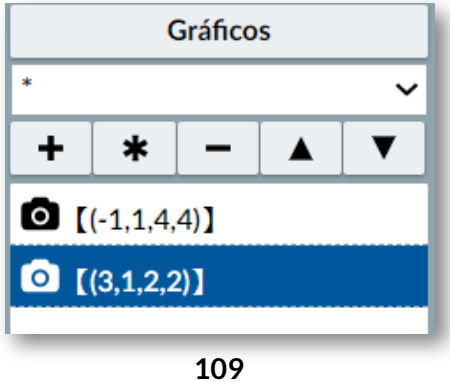


2.10 Textos en escenas

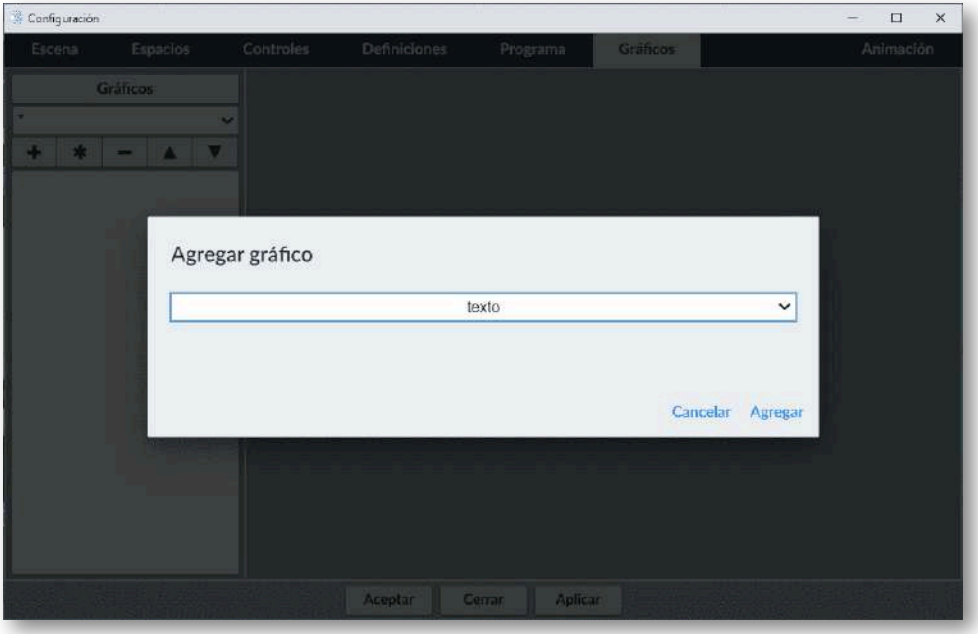
Para insertar textos en una escena —más concretamente, en un espacio de la escena—, primero debemos abrir el panel de configuración marcando el control **Gráficos**.

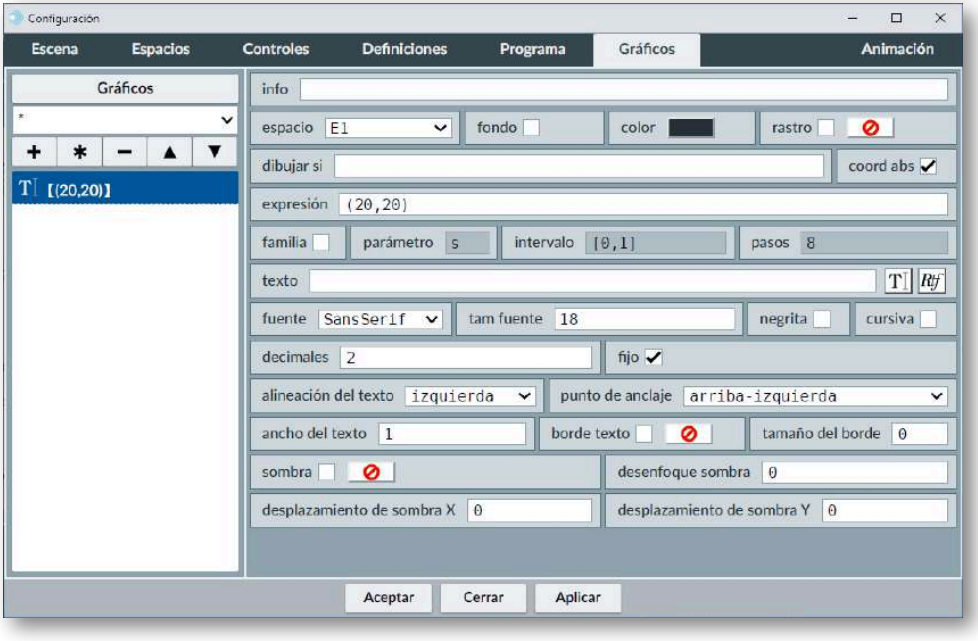


A continuación, nos dirigimos al menú situado a la izquierda, donde se irán incorporando los distintos elementos gráficos que añadamos a la escena. El control + permite crear un nuevo elemento, el botón - elimina el elemento seleccionado y el * lo duplica. Las flechas permiten modificar el orden de los objetos en la lista (los que están más abajo se dibujan por encima de los situados más arriba). El botón **Gráficos** muestra el código asociado a los elementos creados.



Al pulsar el botón **+**, se abrirá una ventana emergente en la que debemos indicar el tipo de elemento que queremos introducir. En este caso seleccionamos **Texto** y, tras pulsar **Aceptar**, el nuevo elemento se añadirá a la lista.





A continuación, se pueden configurar todos los parámetros del texto en los distintos campos del editor. Comentaremos los más utilizados:

- **Info.** Etiqueta informativa que no se muestra en la escena y sirve para que el programador identifique el elemento.
- **Espacio.** Indica el espacio de la escena dentro del cual se colocará el texto. Debe especificarse aunque solo exista un único espacio definido.
- **Fondo.** Permite indicar si el texto quedará fijo en el fondo de la escena o no.
- **Color.** Define el color de la letra (menú situado junto al campo fondo).
- **Rastro.** Indica si queremos que, cuando el texto se desplace por el espacio en el que está situado, deje un rastro o huella por donde ha pasado.
- **Dibujar-si.** Permite establecer las condiciones necesarias para que el texto aparezca o no en la escena.
- **Coord abs.** Permite seleccionar si se utilizan coordenadas absolutas o relativas.
- **Expresión.** Define la posición del texto dentro del espacio. En este caso se debe indicar en coordenadas absolutas.
- **Familia, parámetro, intervalo y pasos.** Son parámetros que permiten que el texto se repita varias veces según los valores indicados. Este punto se desarrolla con mayor detalle en otro apartado del libro.
- **Alinear y punto de anclaje.** Son opciones de alineación del texto en la escena y están relacionadas entre sí. Se considera que el texto se encuentra dentro de una caja cuya anchura viene determinada por la línea más larga y cuya altura corresponde al total de líneas del texto. El punto de anclaje hace referencia a la posición de la caja de texto, mientras que la alineación se refiere a la posición del texto dentro de dicha caja.

- **Decimales.** Indica el número de decimales que mostrarán las expresiones numéricas. Si se marca la opción **fijo**, todas las expresiones tendrán obligatoriamente ese número de decimales; si no, podrán redondearse a menos decimales.
- **Ancho del texto.** En textos no enriquecidos mediante estilos de formato, indica la anchura máxima de una línea antes de que se produzca un salto automático a la siguiente. Para que se tenga en cuenta, el valor mínimo debe ser 20. Por ejemplo, si se indica un ancho de 60, cuando la línea alcance 60 px se insertará un salto de línea.
- **Borde.** Permite añadir un borde a las letras, indicando tanto su tamaño como su color.
- **Texto.** Se puede escribir directamente una frase en el espacio situado a la derecha del botón **texto**, siempre que no se necesite utilizar un formato complejo. Para textos con estilos más elaborados, se puede pulsar el botón **texto** y acceder a un pequeño editor que permite utilizar distintos tipos de letra, superíndices, subíndices, letras griegas, etc., que se detallarán en otro apartado del capítulo.
- **Sombra.** Permite añadir un efecto de sombra al texto, configurando su desenfoque y su desplazamiento en los ejes X e Y.

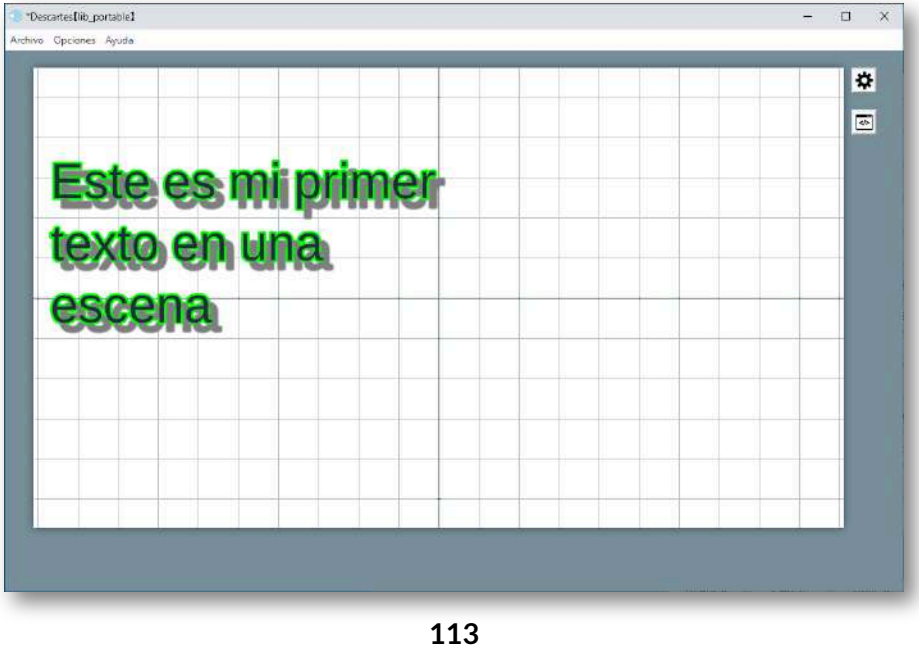
En lugar de un texto fijo, también podemos escribir el nombre de una variable entre corchetes. En ese caso, se mostrará en la escena el valor que tenga dicha variable en cada instante. Esta herramienta resulta muy útil durante la construcción de escenas, ya que permite visualizar los valores de las variables en tiempo real, ayudando a detectar errores o comprobar su evolución.

Por ejemplo, si la variable **a** vale 5 y escribimos en el campo texto **[a]**, en la escena se mostrará el valor 5. Si posteriormente **a** pasa a valer 10, automáticamente se visualizará 10.

En este ejemplo introducimos el texto **"Este es mi primer texto en una escena"** en el campo texto, con la posición absoluta **[20,100]**, con un tamaño de letra de **36**, color de borde **verde**, con **sombra** activada y un **ancho de texto** que provocará un salto de línea a los **500 px**.



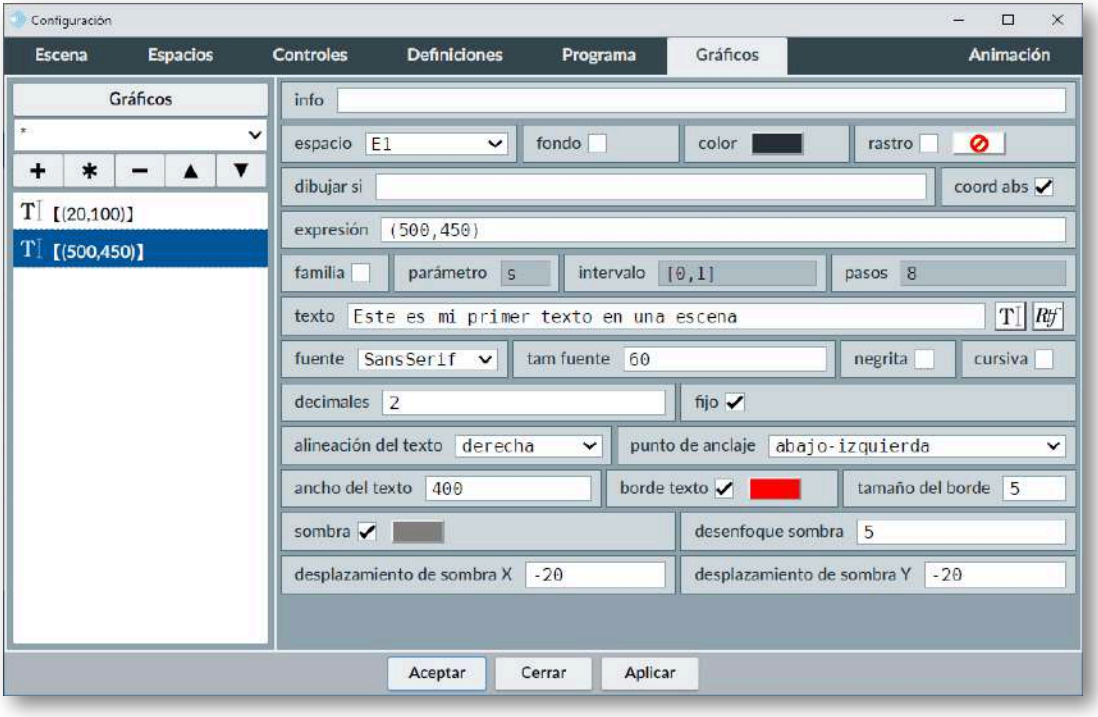
Al en **Aplicar** o **Cerrar** se mostrará la escena con el texto insertado.

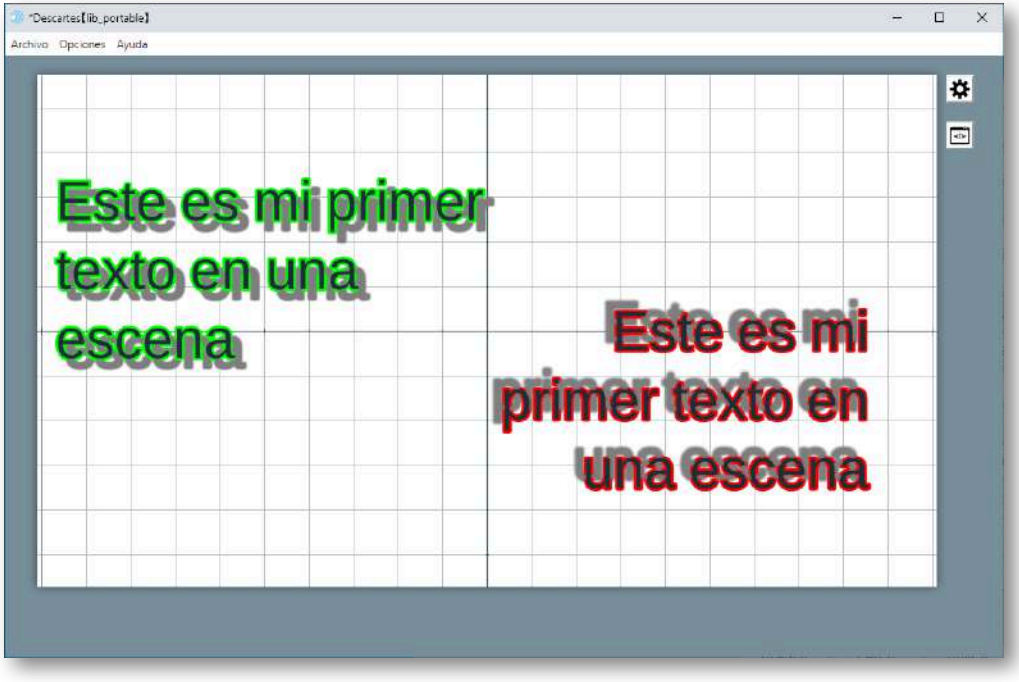


Si posteriormente queremos modificar los parámetros del texto introducido, debemos seleccionarlo en el menú **Textos** de la izquierda del editor, cambiar los parámetros deseados y pulsar **Aplicar**. Si queremos duplicarlo, nos situamos sobre él en el menú **Gráficos** y pulsamos *; si queremos eliminarlo, pulsamos -.

Vamos a duplicar el texto y a la copia le modificaremos el contenido, la posición, las opciones de alineación, la sombra, la anchura del texto y el color del borde.

Conviene recordar que los textos agregados a la lista se dibujan de abajo hacia arriba. Esto significa que, si varios textos coinciden en la misma posición, el que se encuentra en último lugar en la lista se dibujará por encima de los anteriores; el penúltimo quedará debajo de este, y así sucesivamente. Con el pulsador situado entre * y - se pueden subir o bajar los elementos en la lista para modificar su orden de dibujo.

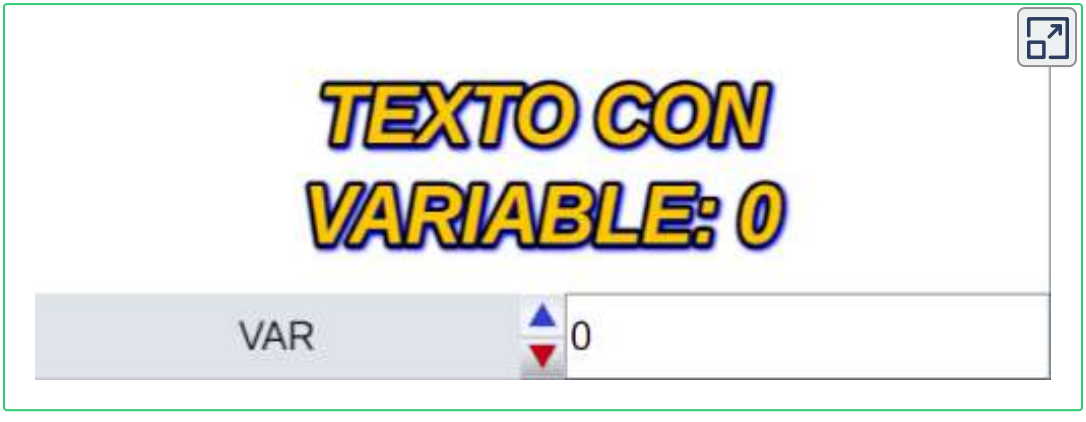




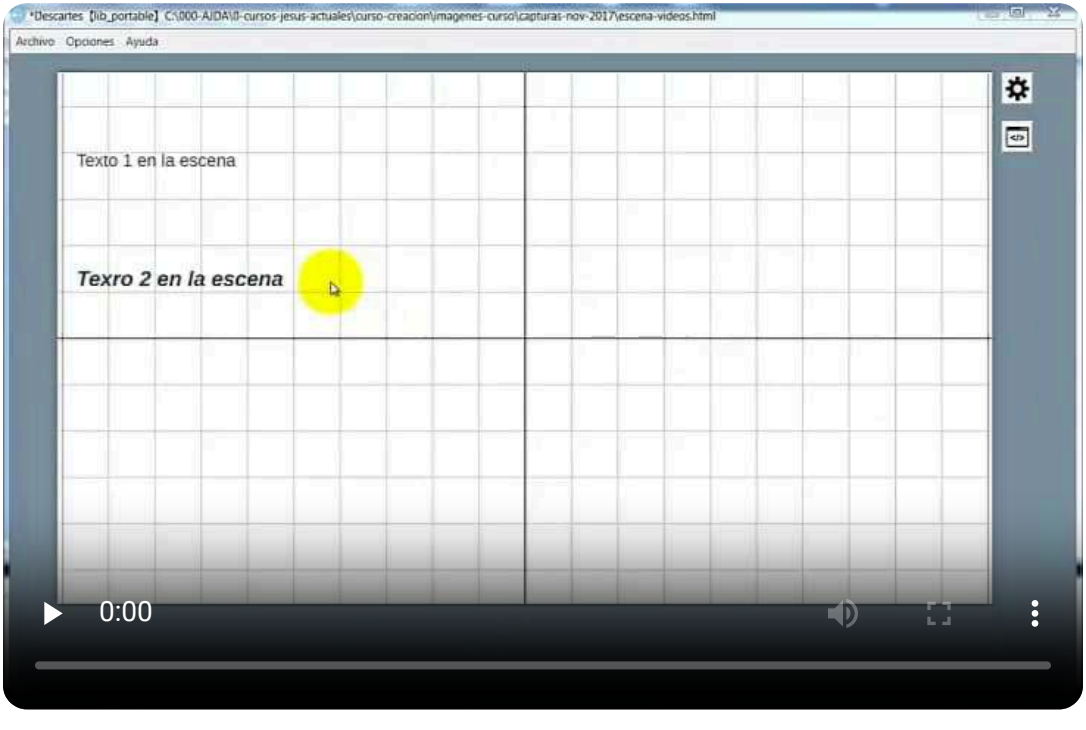
Se muestra una escena con texto con un formato "sencillo".



Presentamos una escena que contiene un texto en el que aparece una variable. El valor de dicha variable puede modificarse mediante el control situado en la parte inferior.

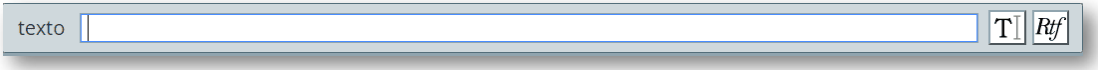


En el siguiente vídeo se presenta como introducir de forma sencilla textos en escenas de Descartes.

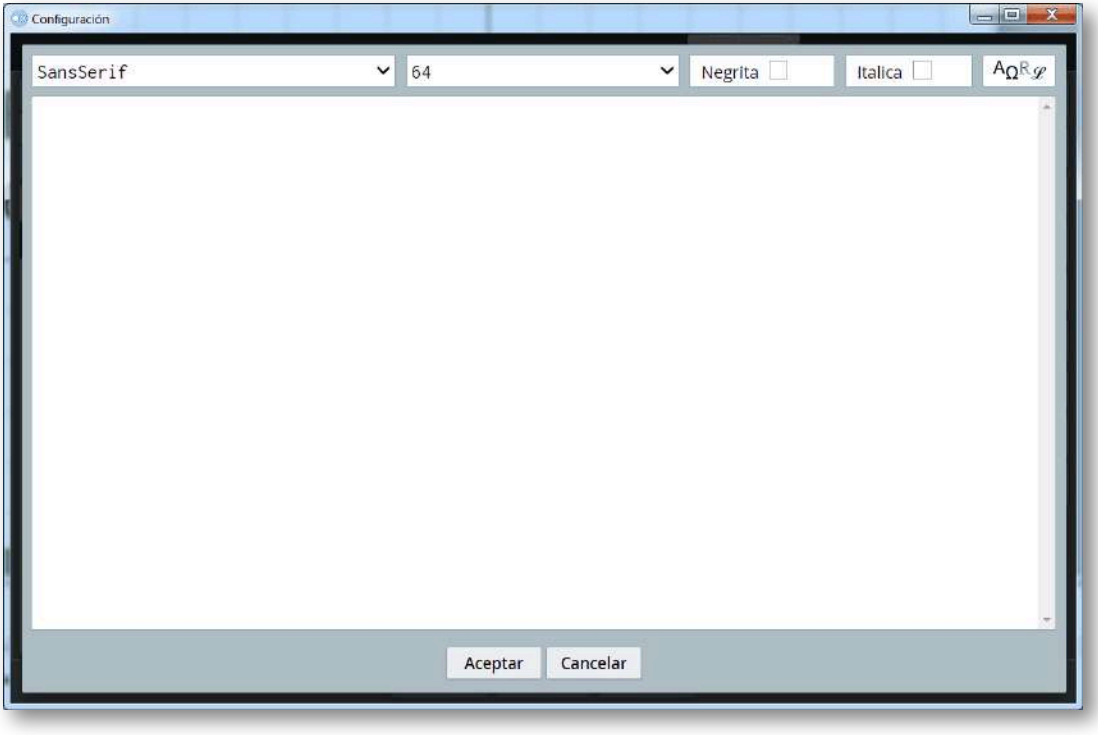


2.11 Editor de textos con formato

En el panel de configuración de un texto (y de otros elementos de la escena que incluyan el parámetro **texto**) aparece el campo **texto** y, a su derecha, dos botones: **T** y **Rtf**.



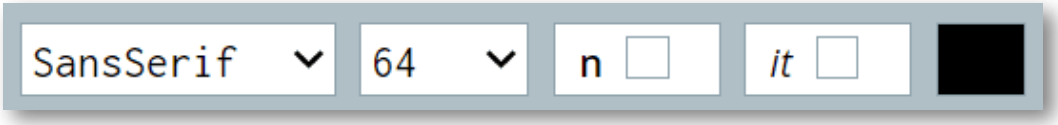
El botón **T** abre un editor de texto simple que permite cambiar aspectos básicos tales como el tamaño y el tipo de letra, aplicar negrita o cursiva, introducir variables e insertar símbolos.



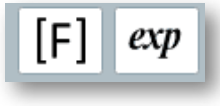
El botón **Rtf** abre un editor de texto ampliado que, además de las opciones del editor simple, permite introducir fórmulas matemáticas, aplicar colores diferentes al texto, subrayado, insertar variables con indicación del número de decimales, representación de vectores y matrices, etc.

Edición de textos con formato

La parte izquierda de la barra de herramientas del editor contiene los controles para definir el tipo, tamaño, estilo y color de la letra.

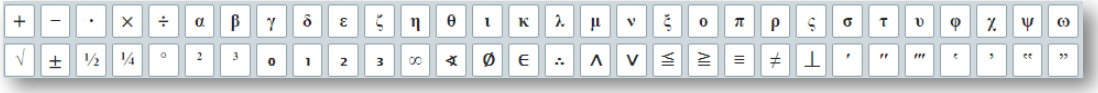


- Un selector para elegir el **tipo de letra**. Ofrece tres opciones: **Monospaced**, **Serif** o **SansSerif**, equivalentes a Courier, TimesRoman y Helvética (o Arial), respectivamente. El valor seleccionado indica el tipo de letra en la posición actual del cursor.
- Un selector para elegir el **tamaño de la fuente** (en puntos). Incluye casi todos los valores pares entre 8 y 80. Desde 28 hasta 48 avanza de cuatro en cuatro, y a partir de ahí de ocho en ocho. El valor seleccionado indica el tamaño del texto en la posición del cursor.
- Tres interruptores para definir el **estilo de la fuente**: negrita y cursiva (itálica), que pueden combinarse libremente.
- Un cuadro de **color** que permite abrir el selector para elegir el color del texto.
- Los botones **F** y **exp**, que permiten insertar variables de la escena y fórmulas matemáticas.



En la parte derecha, así como en la segunda y tercera línea de la barra de herramientas, se encuentran los botones destinados a la edición de fórmulas matemáticas y a la inserción de símbolos.





Se presenta una escena que contiene texto elaborado con el editor de textos de Descartes.

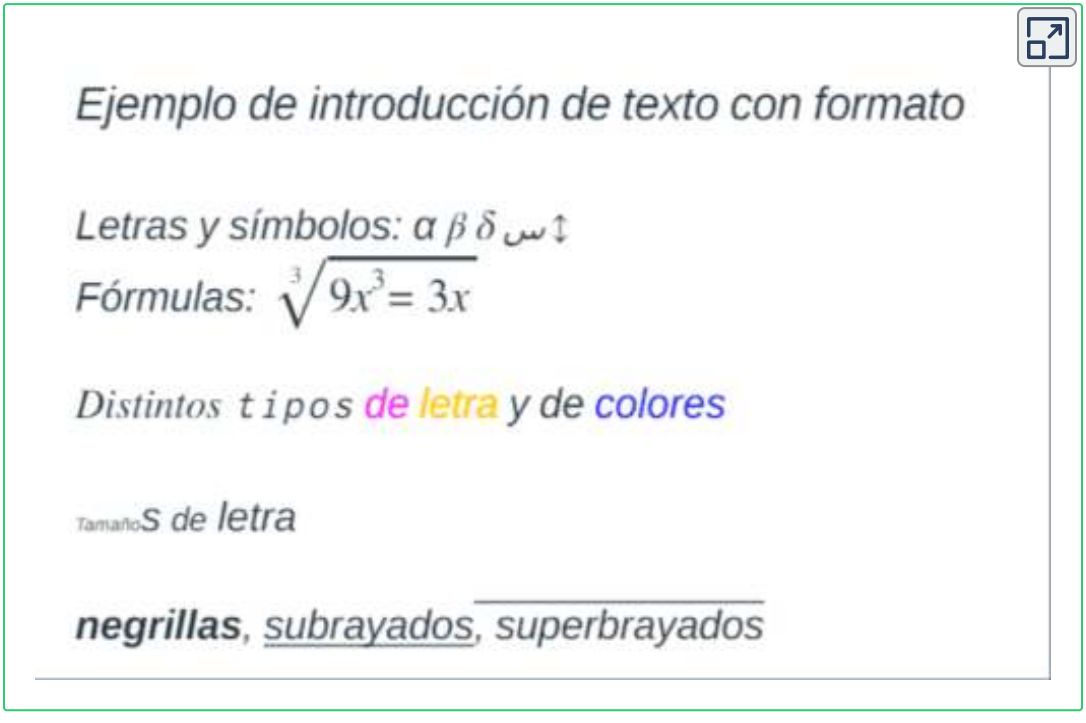


Tabla de símbolos

Descartes utiliza el sistema de símbolos **UNICODE**. UNICODE (caracteres comprendidos entre 0000 y 26FF, numeración hexadecimal) es un sistema de codificación que incluye las letras de prácticamente todos los alfabetos del mundo, así como numerosos caracteres especiales. Probablemente en el futuro los ordenadores podrán representar la totalidad de estos caracteres.

En los equipos actuales muchos de estos símbolos ya están integrados, aunque no todos. Por ello, puede ocurrir que un símbolo se visualice correctamente en un ordenador y no en otro.

Afortunadamente, los símbolos más utilizados, como las letras griegas, aparecen ya en casi todos los sistemas. Cuando un símbolo no está disponible, suele representarse mediante un cuadrado.

Al pulsar el botón de introducción de símbolos (el situado más a la derecha del menú), se abre una ventana que permite seleccionar el símbolo deseado. En la parte izquierda aparece una lista con alfabetos y grupos de símbolos, correspondientes a los principales conjuntos del sistema UNICODE. Al seleccionar uno de ellos, se muestran a la derecha los caracteres disponibles. Al pulsar sobre el símbolo elegido, la ventana se cierra y el carácter se inserta en el editor en la posición donde se encontraba el cursor.



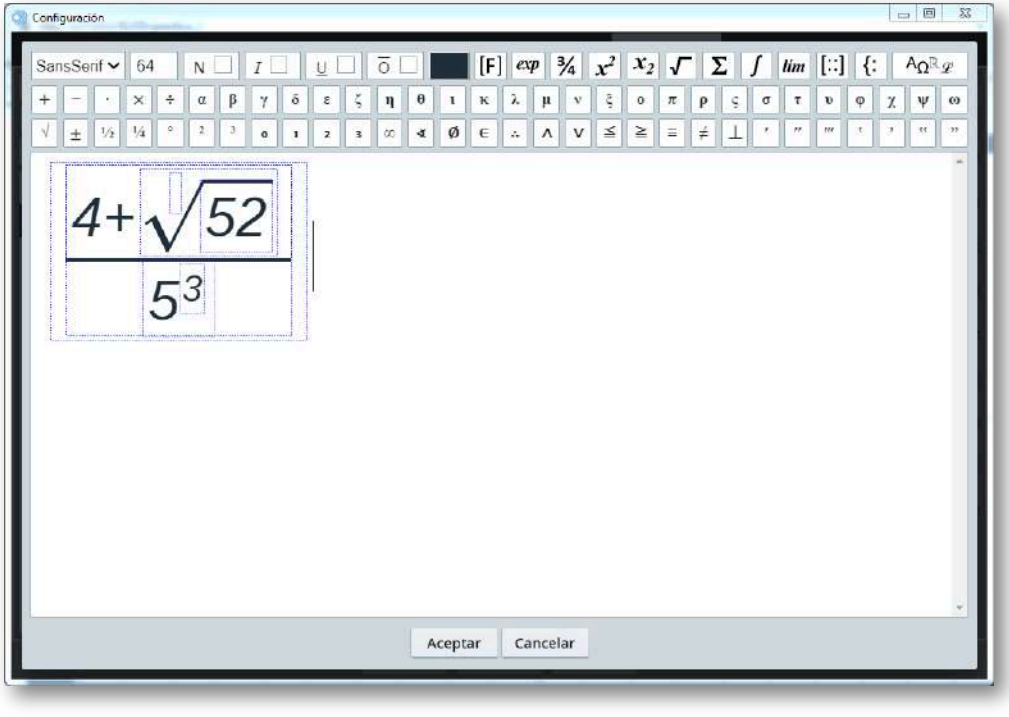
Editor de fórmulas

Las fórmulas matemáticas elementales están compuestas principalmente por texto común junto con elementos como fracciones, raíces cuadradas, subíndices, superíndices, etc.

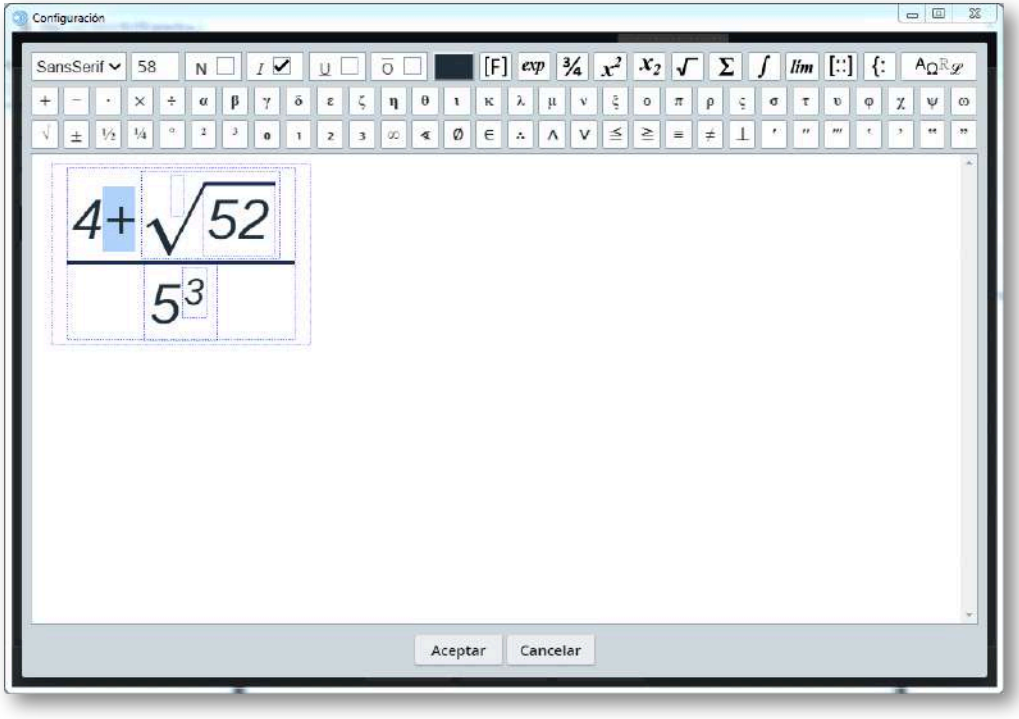
La complejidad en la presentación de una fórmula no proviene tanto de la variedad de sus elementos, sino de la posibilidad de anidarlos unos dentro de otros tantas veces como se desee, lo que puede dar lugar a expresiones muy complejas.

En Descartes, las fórmulas no existen de forma independiente, sino que forman parte de los textos. Se insertan y editan dentro de la ventana de edición de texto. Al pulsar el botón **[F]** situado a la derecha de la barra de controles de edición, se inserta un rectángulo en el área de texto: el **marco de fórmulas**, donde se puede editar la expresión deseada.

Al mismo tiempo se activan los botones con los distintos elementos matemáticos disponibles. En el orden en que aparecen en la barra de herramientas, estos son: fracciones, superíndices y subíndices, raíces cuadradas, sumatorias, integrales, matrices y expresiones. También es posible insertar una fórmula mediante la combinación de teclas **Control + f**.



Dentro del **marco de fórmulas**, es posible desplazarse entre las distintas partes de la expresión de dos maneras: utilizando las flechas de dirección del teclado (situadas en la parte inferior derecha del mismo) o haciendo clic con el ratón sobre la zona concreta que se desea editar. La parte seleccionada dentro del marco de fórmulas quedará resaltada mediante un rectángulo.



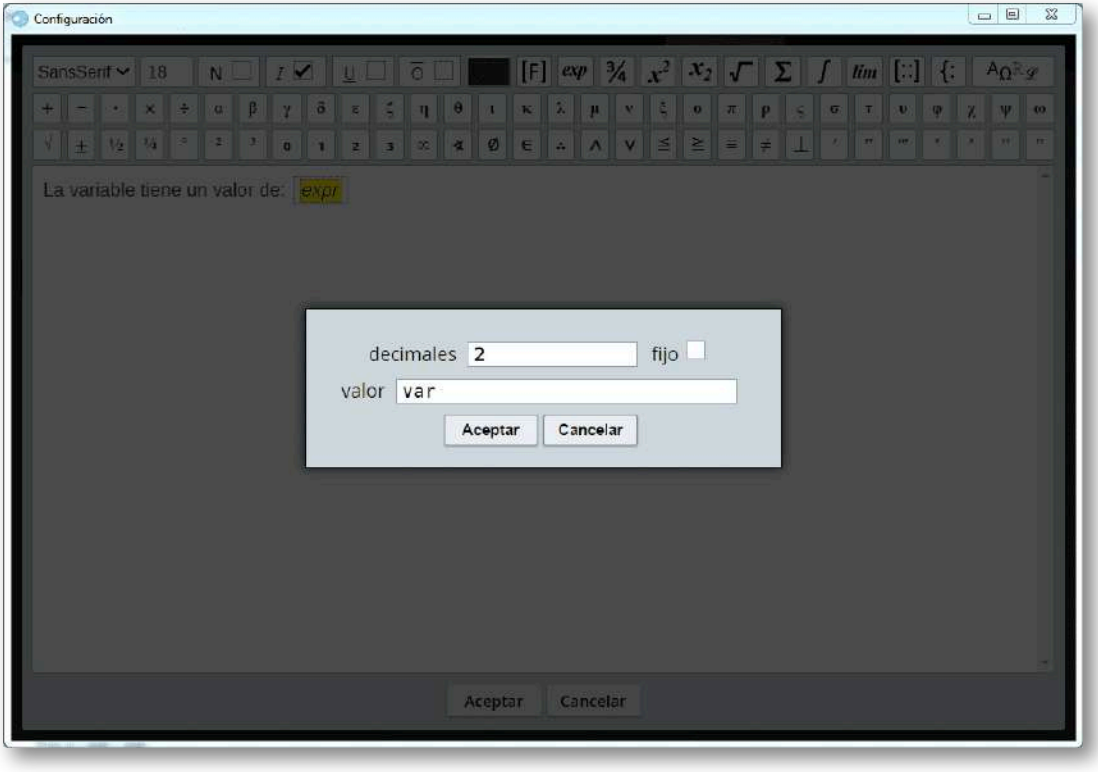
Los caracteres y símbolos dentro de las fórmulas matemáticas pueden editarse igual que cualquier otro texto. El **tipo** y el **estilo de letra** se pueden modificar incluso carácter por carácter. Sin embargo, el **color** y el **tamaño** solo se aplican a la fórmula completa, no de forma independiente a cada uno de sus elementos.

Introducción de expresiones variables dentro de las fórmulas

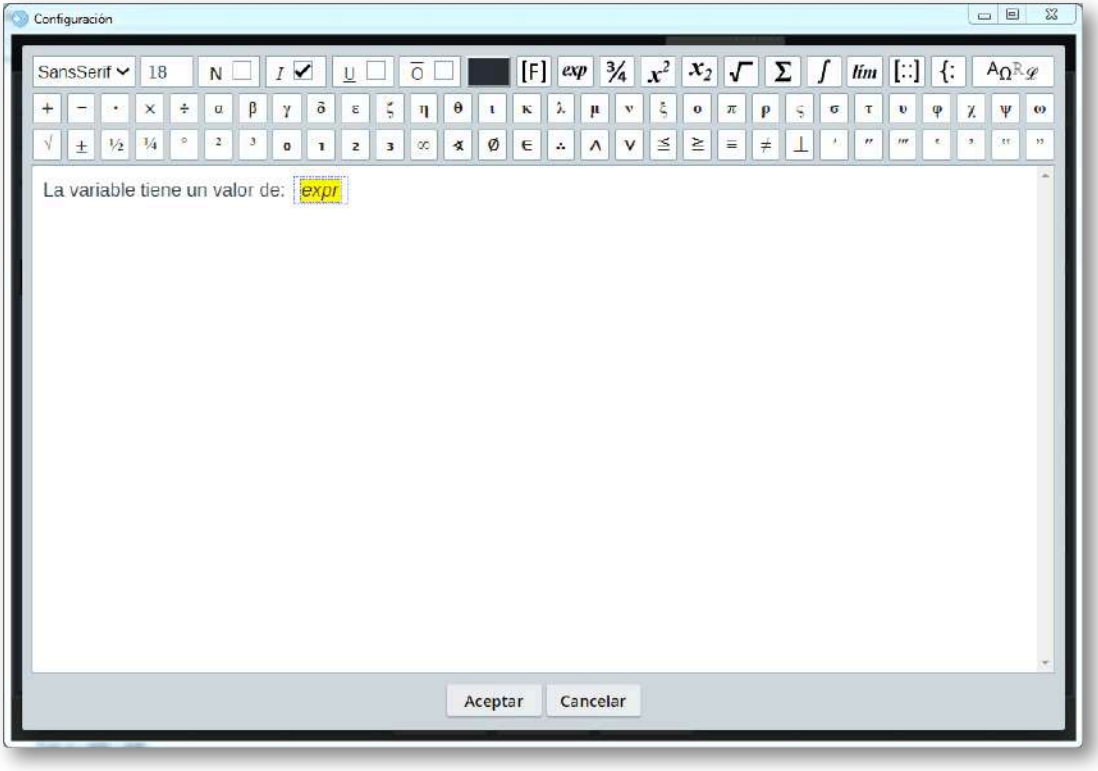
Las fórmulas que se insertan en los textos pueden incluir expresiones evaluables que contengan **variables** o **constantes** previamente definidas en el programa.

Estas variables se crean en otros apartados de la configuración de la escena, que se estudiarán más adelante en el libro.

Para insertar una variable en una fórmula, primero hay que abrir el editor de textos y pulsar el botón **[F]** del menú superior, que permite escribir expresiones matemáticas como raíces, potencias o fracciones. Al hacerlo, aparecerá un pequeño recuadro dentro del área de texto. A continuación, se hace clic dentro de ese recuadro y se pulsa la tecla **[ex]** del menú superior. En ese momento se mostrará la etiqueta **[expr]** dentro del recuadro. Al hacer clic sobre **[expr]**, se abrirá una ventana emergente donde se podrá escribir la expresión matemática deseada.

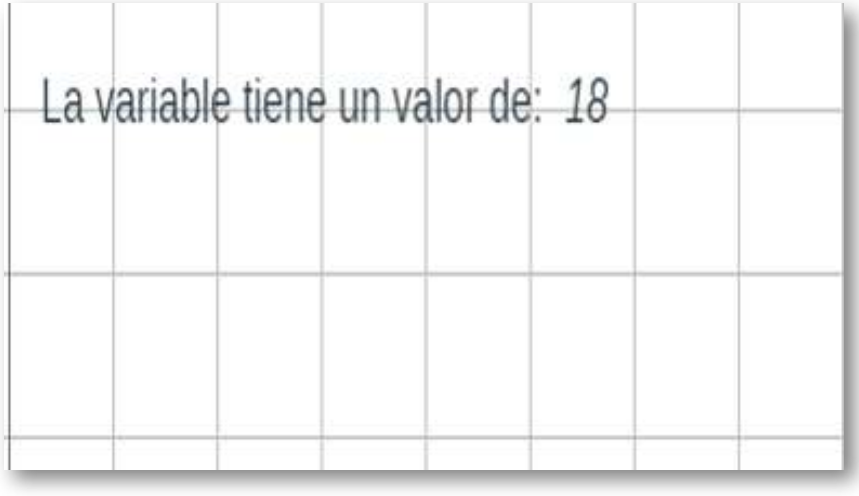


En el siguiente ejemplo, introducimos una variable llamada **var** dentro de un texto (esta variable debe haberse definido previamente en otro elemento de la configuración de la escena).

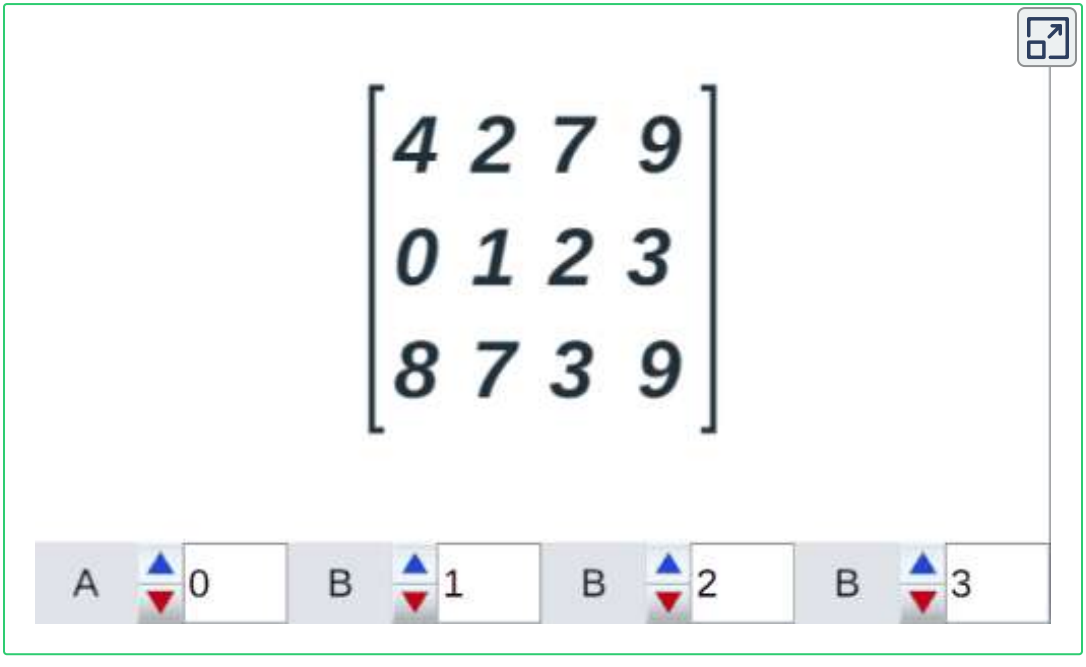


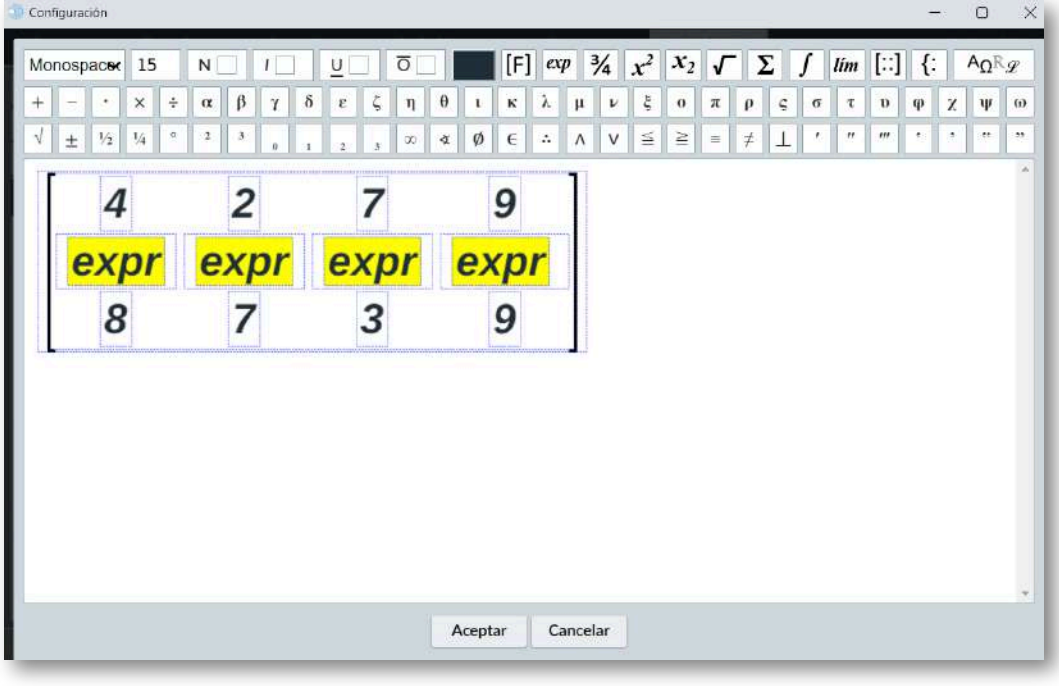
En el editor de texto, la variable se verá de la siguiente forma:

Mientras que en el editor de escenas o en el navegador se mostrará el valor actual de la variable **var**, que en este caso es 18.



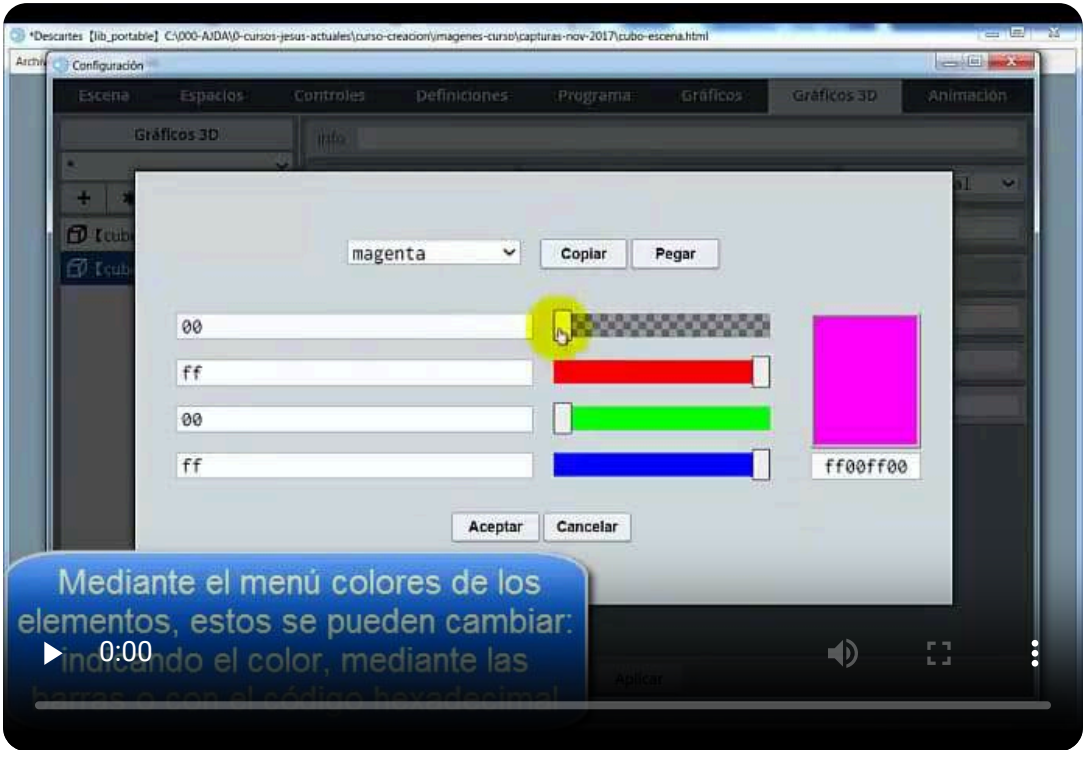
Mostramos una escena que contiene una matriz en la que se pueden introducir los elementos de la segunda fila mediante variables.





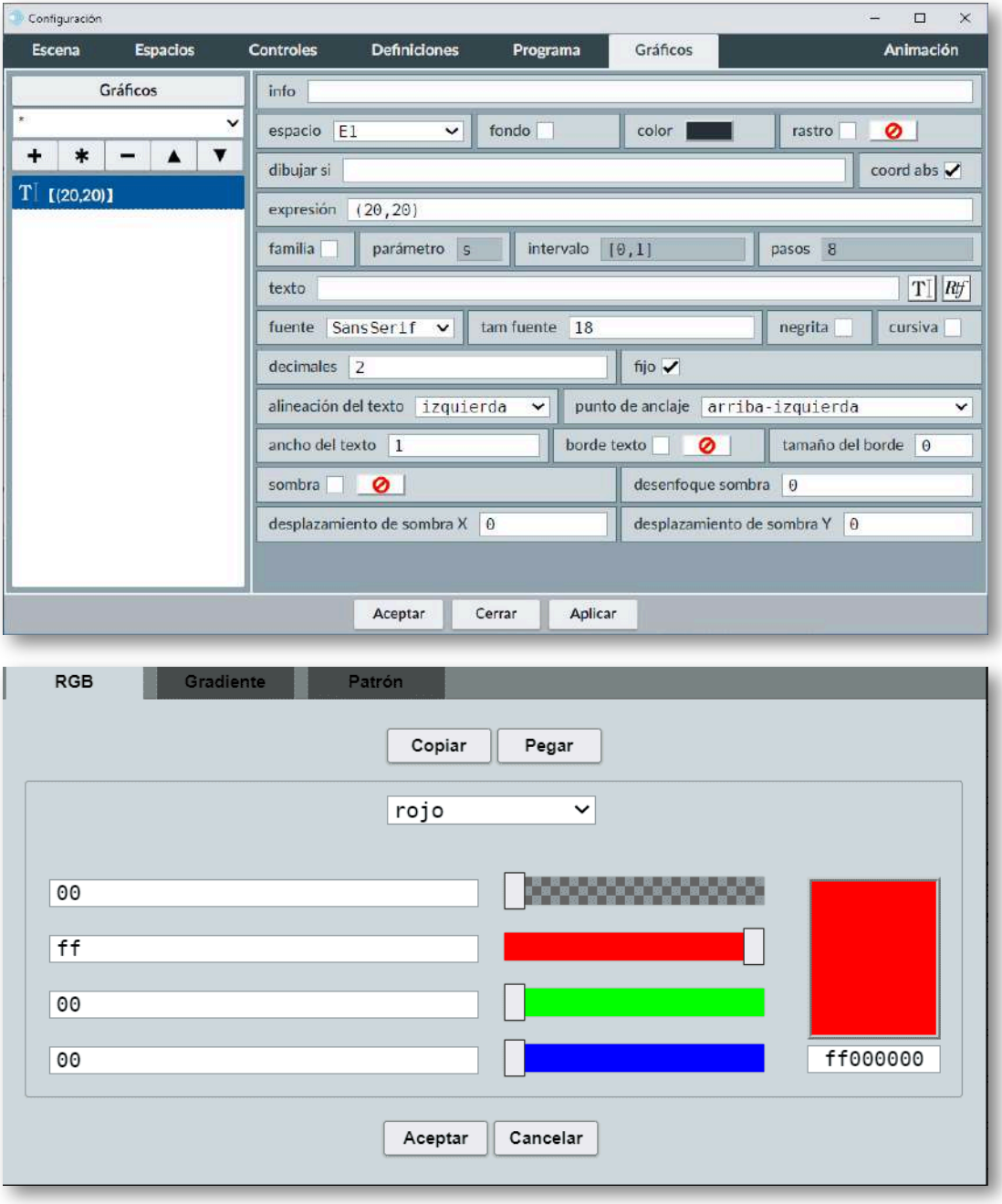
2.12 Colores en textos y otros elementos

Inicialmente vamos a mostrar un vídeo para ilustrar cómo trabajar con el editor de colores de Descartes.



En el editor de textos con formato y en muchos otros objetos y elementos de la aplicación aparece el **editor de colores**, que permite definir el color de un elemento o de un texto. Este editor se abre al pulsar el botón de configuración de color (un pequeño cuadrado que muestra el color seleccionado), disponible en todos los elementos donde es posible modificar el color. Por ejemplo, en el editor de configuración de texto, se accede a él haciendo clic en el cuadrado de color situado a la derecha de los campos **Fondo** y **Color del borde de texto**.

Pasamos a explicar cómo trabajar con el editor de colores.



En la parte superior del cuadro de diálogo que se abre aparecen tres pestañas: **RGB** (relleno sólido), **Gradiente** (relleno mediante un degradado de color) y **Patrón** (relleno con una imagen patrón).



Pasamos a comentar los diferentes elementos que contiene la pestaña de color de relleno sólido o RGB:

- **Copiar:** Botón que permite copiar el color actual a un portapapeles interno de Descartes para usarlo en otros elementos del interactivo, útil para mantener consistencia de color y transparencia entre objetos.
- **Pegar:** Botón que permite pegar un color previamente copiado, aplicándolo al elemento en edición.
- **Menú de colores:** Menú desplegable con colores prediseñados para seleccionar rápidamente.
- **Campo de texto y barra de transparencia:** Controla qué tanto se permite ver de los objetos detrás del objeto editado. Se puede ajustar mediante la barra o introduciendo valores en el campo de texto.
- **Campo de texto y barra del color rojo:** Define el componente rojo del color del objeto, ajustable mediante la barra o el campo de texto.
- **Campo de texto y barra del color verde:** Define el componente verde del color del objeto, ajustable mediante la barra o el campo de texto.
- **Campo de texto y barra del color azul:** Define el componente azul del color del objeto, ajustable mediante la barra o el campo de texto.
- **Panel de muestra del color:** Cuadro de vista previa que muestra el color del objeto editado junto con su código hexadecimal de 8 dígitos: los primeros 6 dígitos corresponden al color y los últimos 2 a la transparencia.

Comenzando por el relleno sólido, en la zona superior se encuentra un selector desplegable que permite elegir el color por su nombre. A la izquierda se muestran los valores de **transparencia** y las cantidades de **rojo**, **verde** y **azul** que componen el color, expresadas en notación hexadecimal. En la parte central aparecen cuatro barras deslizantes con las que se pueden ajustar manualmente la transparencia y los niveles de rojo, verde y azul. A la derecha hay un recuadro con el color resultante según la configuración seleccionada.

Además de esta funcionalidad, también es posible seleccionar un color mediante una herramienta tipo **gotero**. Para ello, basta con pulsar en el panel de previsualización del color (el rectángulo situado sobre el código hexadecimal). Al hacerlo, aparece un selector de color con forma de círculo y una cuadrícula en su interior. Este selector sigue el movimiento del ratón y puede colocarse sobre el color que se desee muestrear, ya sea dentro o fuera de DescartesJS.

Cuando el color deseado quede dentro del cuadrado central del círculo, se hace clic y automáticamente se obtiene el código hexadecimal correspondiente en el parámetro situado debajo del panel de previsualización del color. Esta herramienta permite evitar el uso de aplicaciones externas para capturar colores, agilizando así la obtención y traslado de códigos de color.



Si el color ha de introducirse en forma de texto, se puede introducir en forma hexadecimal y forma decimal.

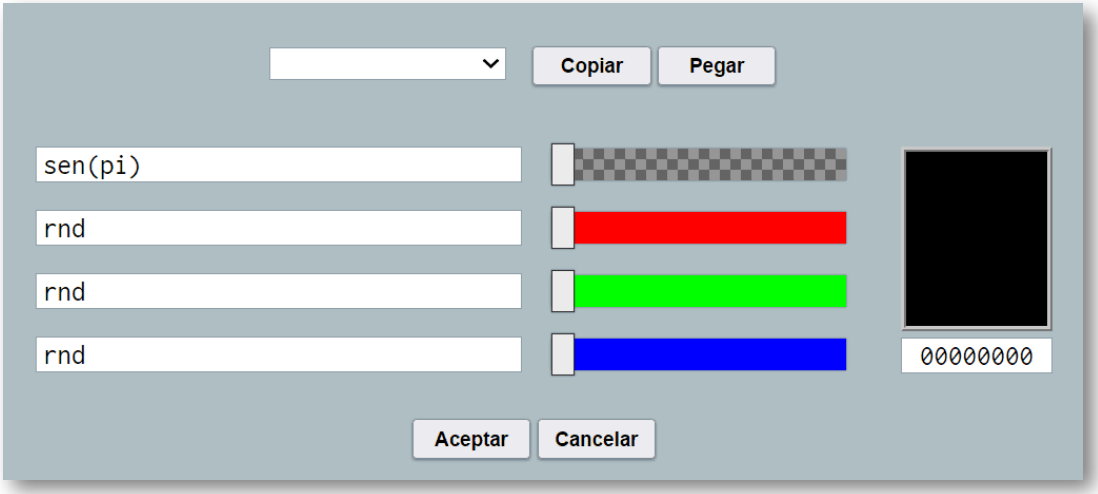
La siguiente tabla recoge las expresiones hexadecimales asociadas a los colores con nombre. En cada código, las dos primeras cifras indican la cantidad de **rojo**, la tercera y la cuarta la de **verde**, y las dos últimas la de **azul**. Todos estos colores son completamente opacos, por lo que el componente de transparencia es cero. En Internet pueden encontrarse numerosas tablas que relacionan los nombres de los colores con su notación hexadecimal.

negro	000000
grisObscuro	404040
gris	808080
grisClaro	c0c0c0
blanco	ffffff
magenta	ff00ff
azul	0000ff
turquesa	00ffff
verde	00ff00
amarillo	ffff00
naranja	ffc800
rojo	ff0000
rosa	ffafaf

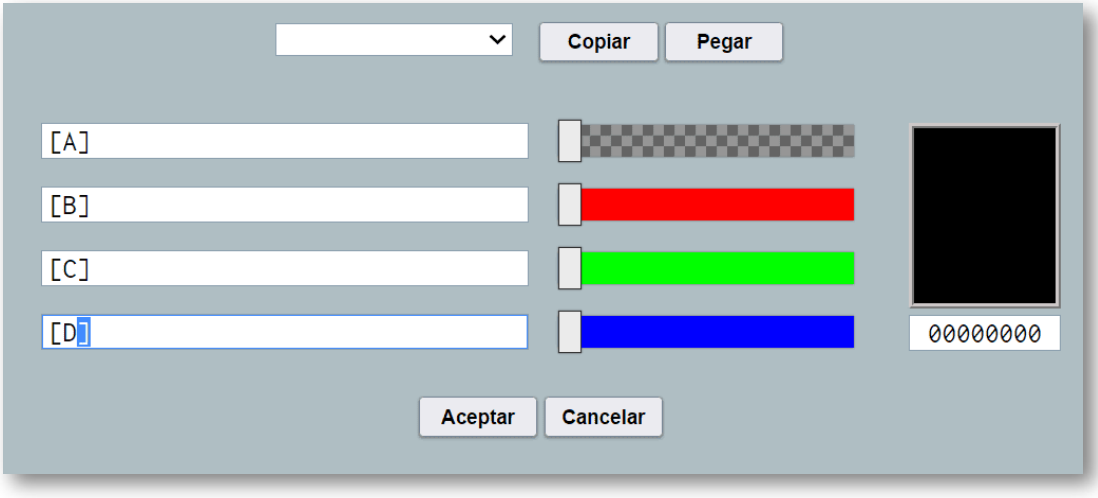
Si el componente de **transparencia** es mayor que cero, su valor se añade a la izquierda de la representación hexadecimal del color. Por ejemplo, **aaffff00** corresponde a un amarillo bastante transparente, mientras que **440000ff** representa un azul ligeramente transparente.

También es posible definir colores mediante **expresiones variables** o **expresiones matemáticas** (incluyendo variables definidas en la escena), cuyos valores deben estar comprendidos entre 0 y 1. Esto permite generar gráficos con gamas de color variables e incluso crear elementos cuyo color cambie dinámicamente a lo largo de la escena.

Por ejemplo, la función **rnd** permite obtener números aleatorios comprendidos entre 0 y 1. De esta forma, cada vez que se actualice la escena, el valor del color será diferente.

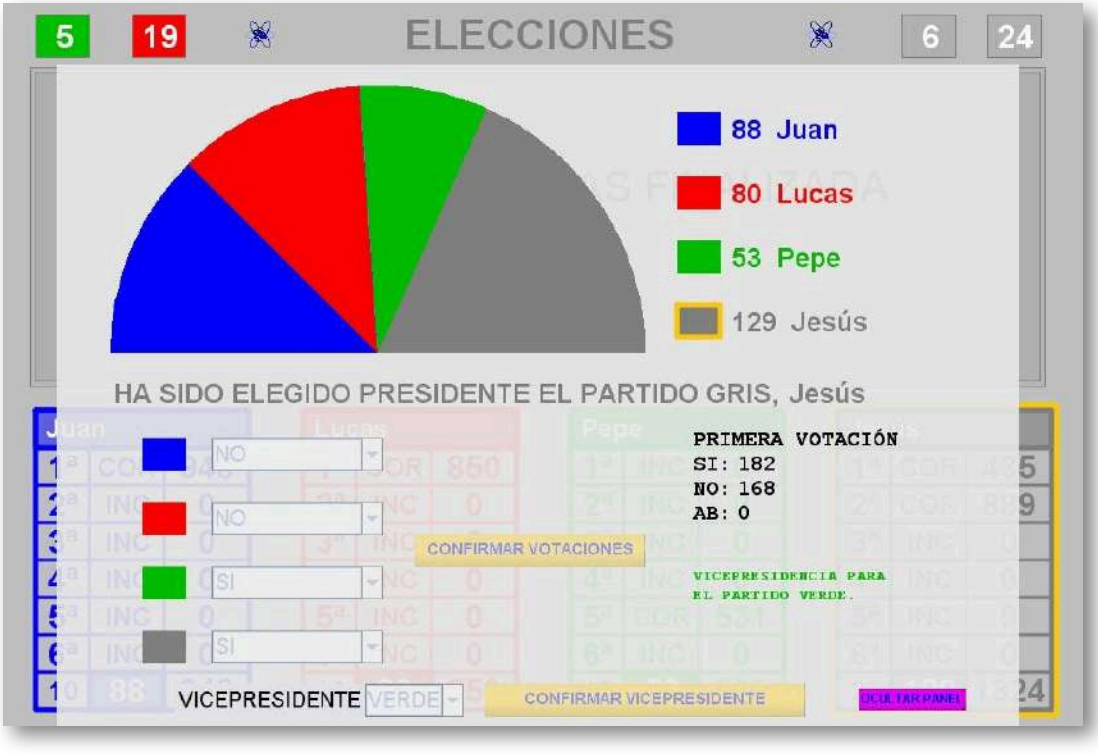


Si se utilizan valores de **variables** (cuyo valor debe estar comprendido entre 0 y 1) para definir colores, estos deben escribirse entre **corchetes**. Por ejemplo, si queremos definir el color mediante las variables **B**, **C** y **D**, y la transparencia mediante **A**, se deberá escribir **[B]** en el campo correspondiente al rojo, **[C]** en el verde, **[D]** en el azul y **[A]** en la transparencia.



Si se desea copiar un color que ya aparece en la escena, basta con abrir la ventana de configuración correspondiente y pulsar **Copiar**. Después, se abre la ventana donde se quiere aplicar ese color y se pulsa **Pegar**.

El uso de **transparencias** en los espacios permite que estos sean total o parcialmente transparentes, de modo que se pueda ver el contenido de otros espacios situados debajo. No obstante, los elementos contenidos en un espacio se seguirán dibujando aunque este sea transparente. Por ejemplo, en el juego Elecciones puede observarse cómo un espacio semitransparente superpuesto deja ver parcialmente los elementos del espacio inferior.

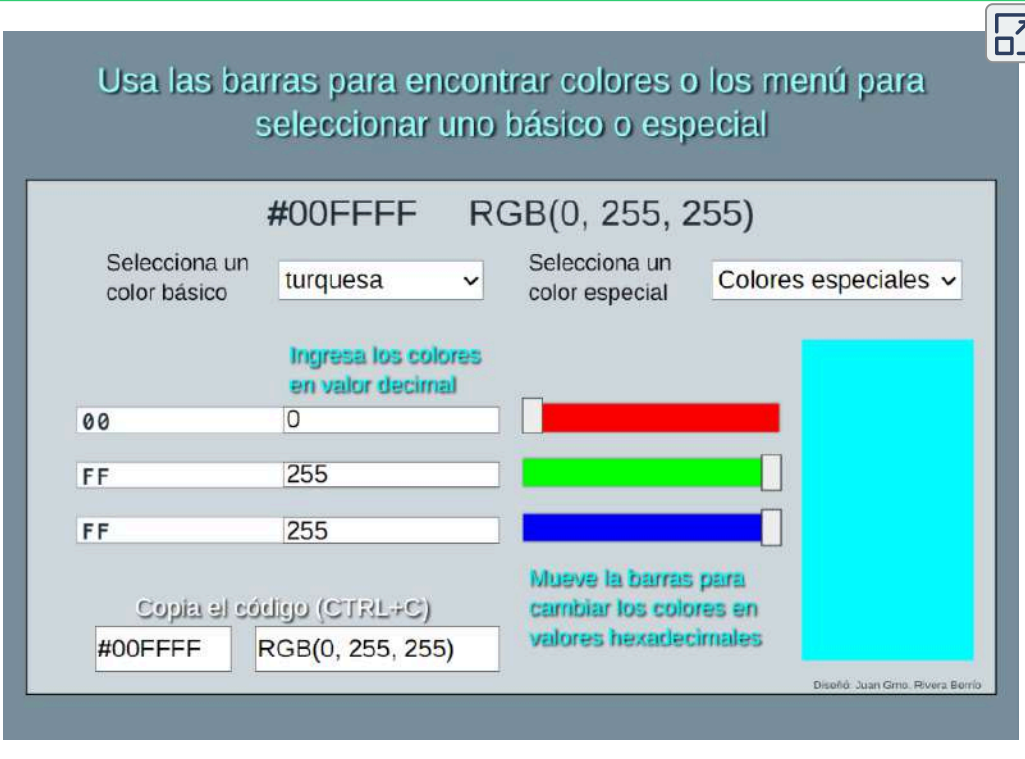


Cuando la transparencia se utiliza en elementos, dicho elemento pasa a ser transparente o semitransparente, permitiendo ver total o parcialmente los objetos o espacios que hay detrás de él.

A continuación, incluimos un interactivo que permite practicar con el menú de selección de colores de Descartes y familiarizarse con su funcionamiento. A través de él se pueden explorar las distintas formas de elegir y ajustar un color dentro de una escena.

El interactivo muestra, además, la conversión automática del color seleccionado a formato RGB y hexadecimal, lo que facilita su uso posterior en otros elementos de la escena o en aplicaciones externas.

También incorpora un menú de colores básicos y otro de colores especiales, que permiten seleccionar determinados tonos de forma directa y rápida, sin necesidad de ajustarlos manualmente.



En el fondo del interactivo se aprecian distintos gradientes y degradados de color. Este tipo de efectos también pueden incorporarse fácilmente en las escenas de Descartes.

Además del relleno sólido, es posible aplicar **gradientes** o degradados de color. Esta pestaña incluye los siguientes elementos:



- **Copiar:** Botón para copiar la definición actual de gradiente en la pestaña.
- **Pegar:** Botón que permite implementar la definición copiada con el botón **Copiar** anteriormente.
- **Barra de muestra:** Muestra visual de cómo se verá la definición actual del gradiente.
- **x1, y1, x2, y2:** Campos de texto para introducir coordenadas absolutas del segmento sobre el que se define el gradiente: x1, y1 para el primer extremo; x2, y2 para el otro extremo. El gradiente se traza siguiendo dicho segmento y las coordenadas se toman respecto al contenedor donde se aplicará (fondo de un espacio, área de un botón, etc.).
- **+**: Botón para agregar paradas. Cada parada permite definir un gradiente distinto e incluye los siguientes controles:
    - **Parada:** Campo de texto donde se introduce un número entre 0 y 1 que indica dónde empieza el gradiente. 0 corresponde al inicio del lienzo, 1 al final.

- **Selector de color:** Botón con recuadro de color que despliega una paleta al oprimirlo. Se puede seleccionar un color pulsando sobre la paleta o introduciendo un valor numérico en formato RGB, HSL o hexadecimal.
- -: Botón que elimina la parada correspondiente.

Resumidamente, las coordenadas **(x1, y1)** y **(x2, y2)** definen el inicio y fin de la recta sobre la cual se aplicará el gradiente. Se pueden añadir múltiples puntos de parada, y en cada parada se asigna un color y una posición relativa entre 0 y 1, donde 0 corresponde al inicio de la recta y 1 al final.

El resultado es una transición progresiva de colores aplicada al elemento, como por ejemplo a un rectángulo con bordes redondeados.



La última opción relacionada con el color es el uso de una **imagen patrón** como relleno. En este caso, el elemento se cubrirá completamente con la imagen seleccionada. La imagen comienza a dibujarse en el origen del lienzo o canvas. De ahí, si sobra espacio, se repite (a manera de mosaico) hasta cubrirlo todo.

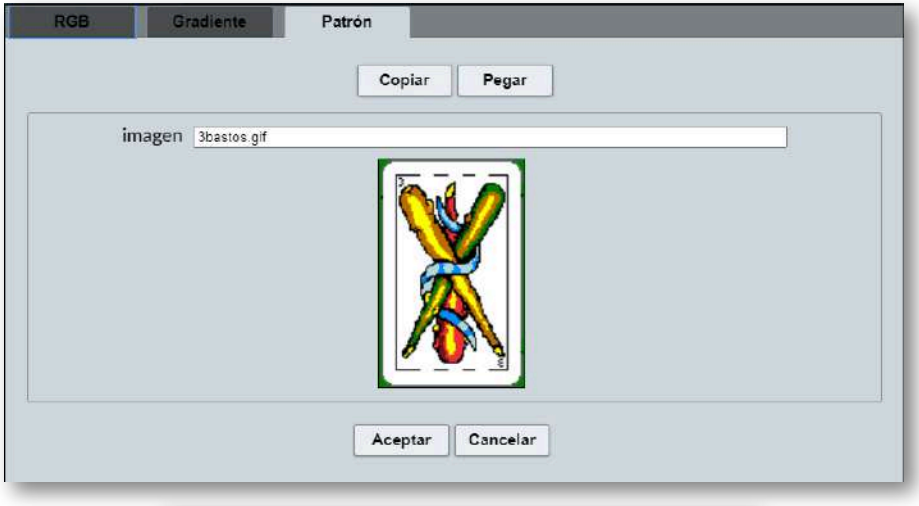
En la pestaña **Patrón** de la herramienta de control de colores se puede incluir una imagen (jpg, png o gif estática) en el objeto definido (como un espacio o el fondo de un botón). Los elementos que se pueden configurar son:

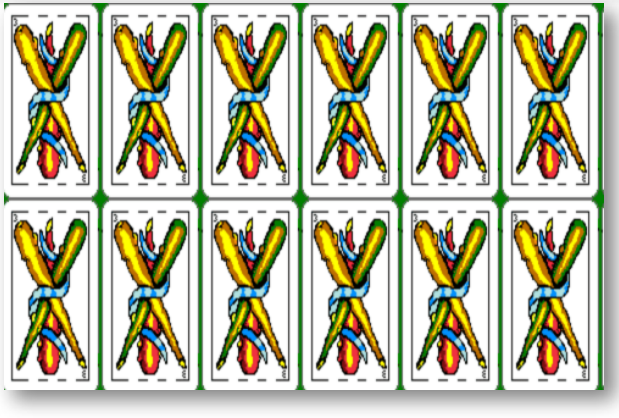
- **Copiar y pegar:** botones que permiten copiar y pegar una definición de patrón de imagen, funcionando de manera similar a las otras pestañas.

- **Imagen:** campo de texto en el que se introduce la ruta de la imagen a usar. Si se utiliza una dirección relativa al archivo HTML de la escena, es recomendable que la escena esté previamente guardada.

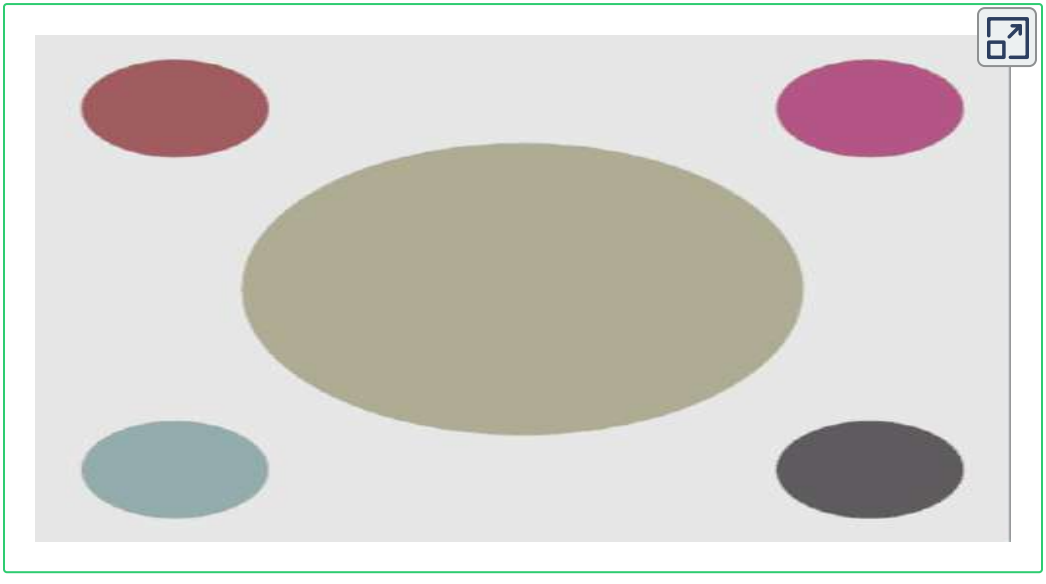
Cuando se aplica un gradiente a un control tipo barra, el color de la barra cambia dinámicamente al mover el manipulador. Si se utiliza un patrón de imagen, el comportamiento depende de cómo se asocie:

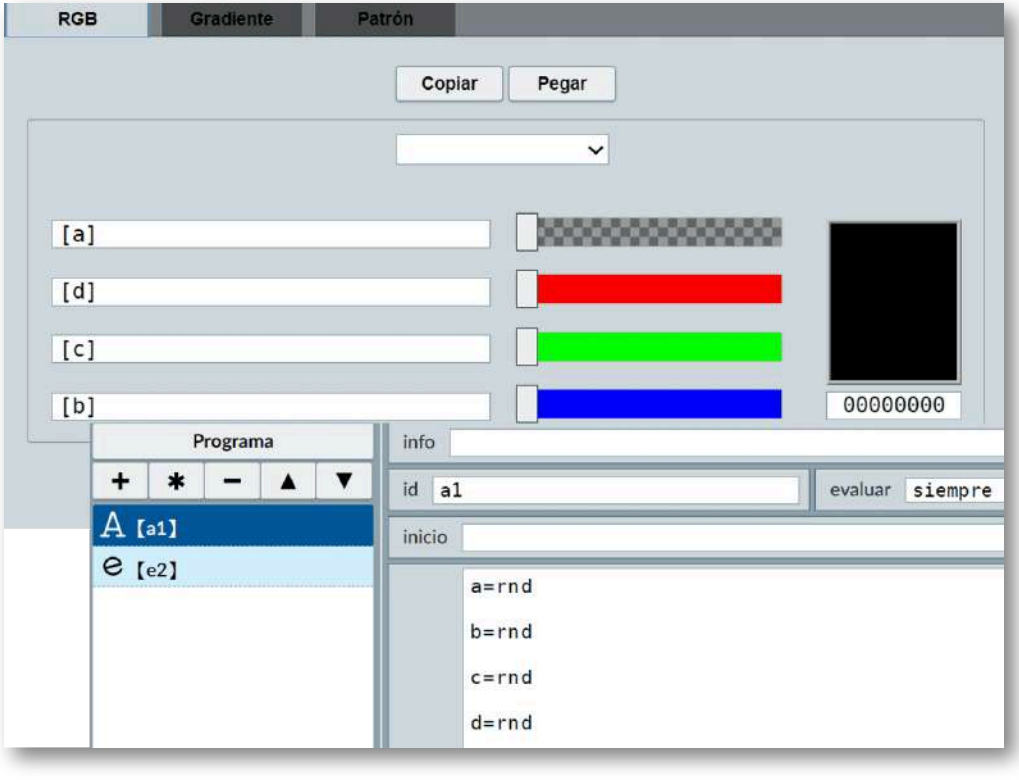
- Si el patrón se asigna al color interior de la barra, el manipulador muestra el mosaico de la imagen.
- Si el patrón se asigna al color de la barra, el mosaico de la imagen se mostrará detrás del manipulador.





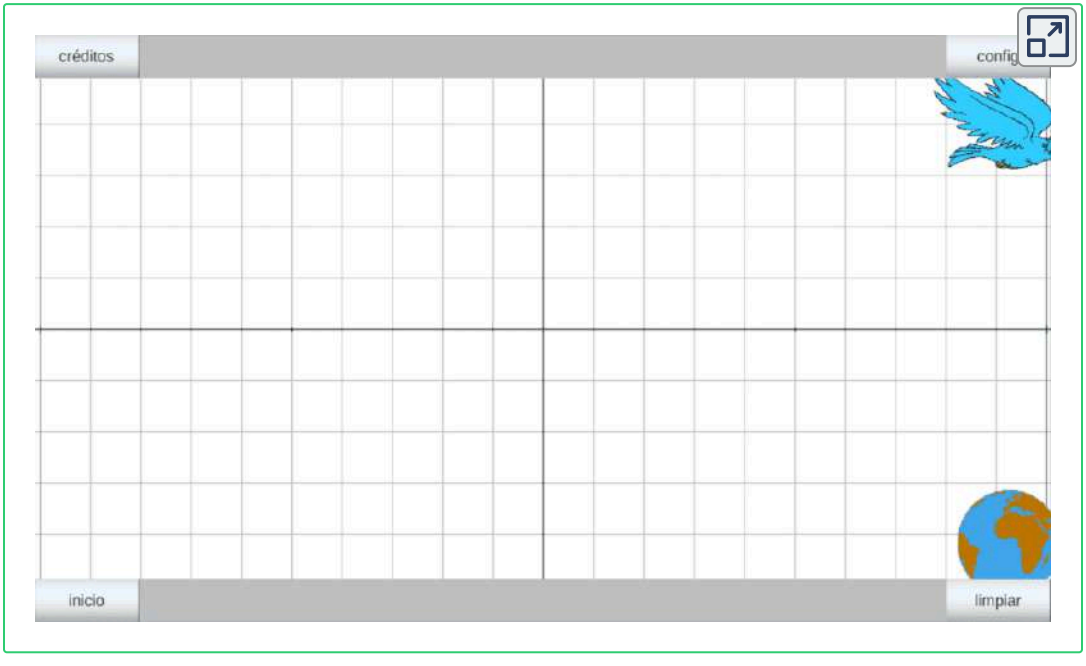
Escena en la que se muestran los colores de una escena que varían de forma aleatoria.





2.13 Animaciones en escenas

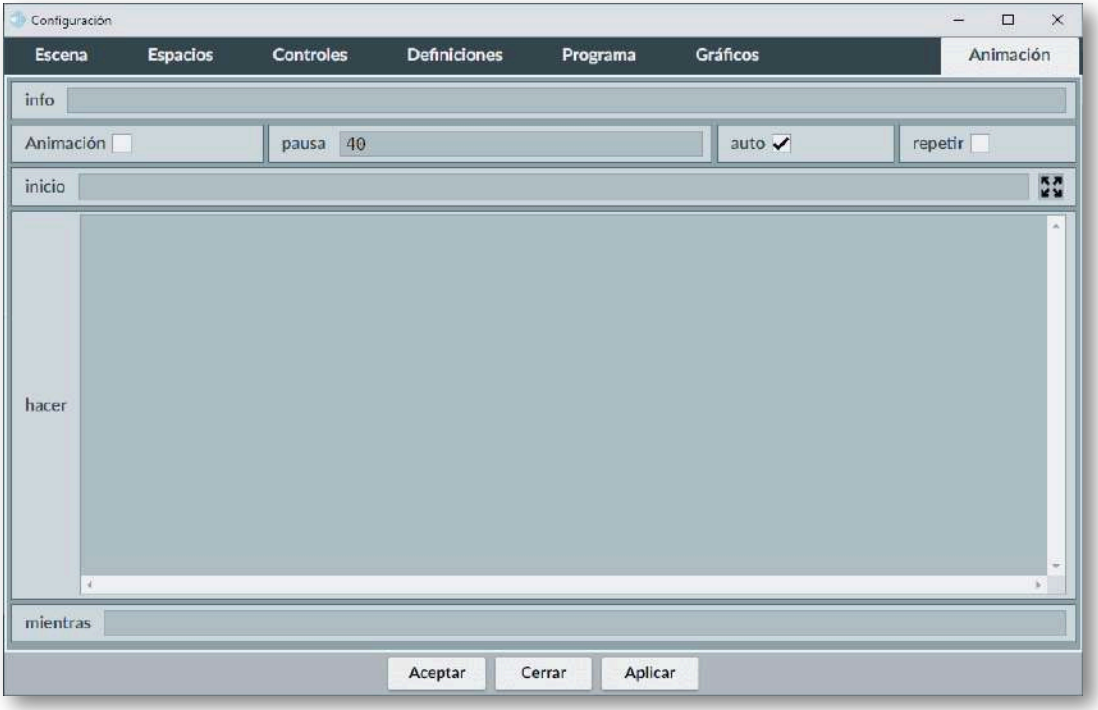
La **animación** de una escena es el elemento que le aporta movimiento y permite establecer una secuencia o evolución en el tiempo. Mostramos una escena con una animación sencilla en la que se presentan dos imágenes en movimiento.

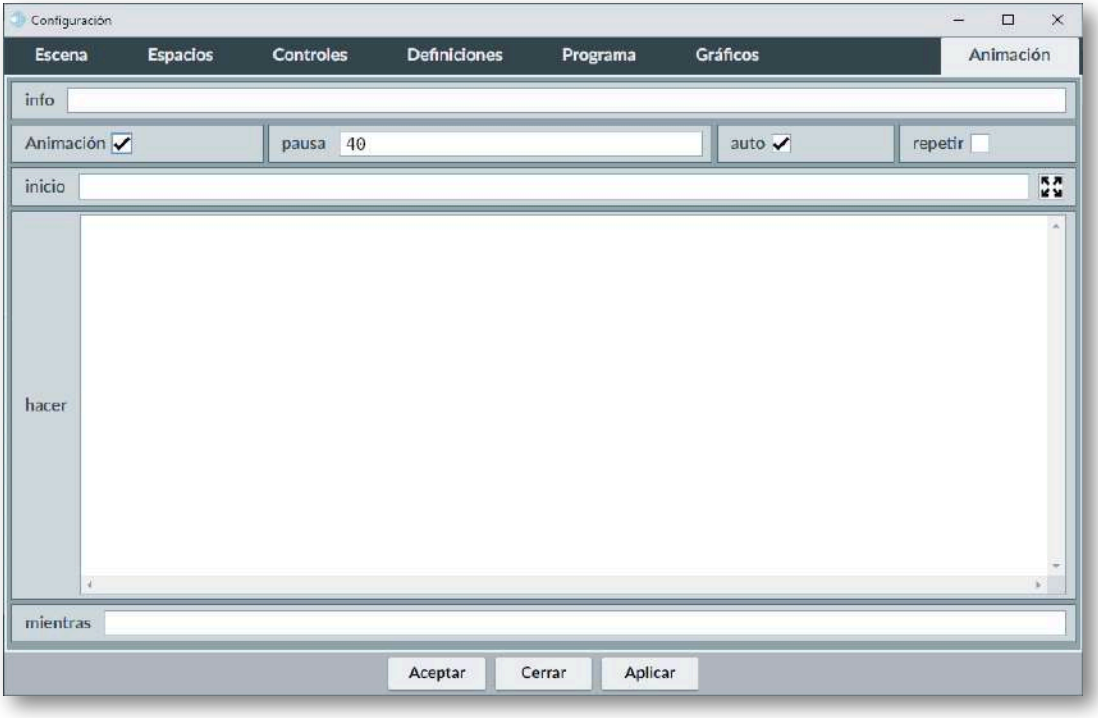


Para acceder al menú para realizar animaciones de interactivos dentro del editor de configuración de la escena, hay que pulsar el selector **Animación**, situado a la derecha del menú del editor de escenas.

Por defecto, la animación aparece desactivada. Para ponerla en funcionamiento, es necesario hacer clic en el campo **Animación** y activarlo.

Mostramos capturas de pantalla del menú de configuración de **Animación** antes y después de su activación.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
info
Animación
pausa
40
auto
repetir
inicio
hacer
mientras
Aceptar
Cerrar
Aplicar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
info
Animación
pausa
40
auto
repetir
inicio
hacer
mientras
Aceptar
Cerrar
Aplicar

Parámetros del menú de animación:

- **Info**: Campo destinado a añadir comentarios o información para el programador. Este contenido no se muestra en la escena.
- **Animación**: Permite activar o desactivar la animación de la escena.
- **Pausa**: Controla la velocidad de la animación. Al aumentar su valor, la animación se ralentiza.
- **Auto**: Indica si la animación se inicia automáticamente al cargar la escena o si es necesario activarla manualmente.
- **Repetir**: Si se activa esta opción, la animación se ejecuta en bucle de forma indefinida.

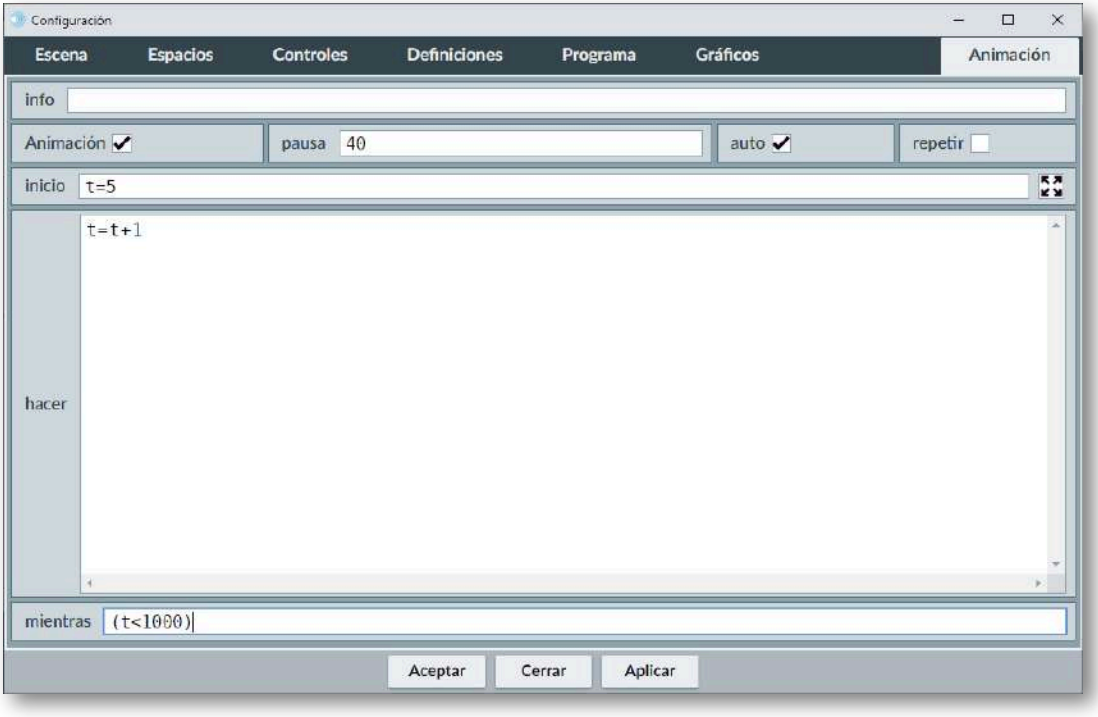
Los campos **inicio**, **hacer** y **mientras** funcionan igual que en cualquier otro algoritmo:

- **Inicio**: Permite definir los valores iniciales de las variables.
- **Hacer**: Indica las operaciones que se deben realizar mientras el algoritmo esté en ejecución.
- **Mientras**: Define las condiciones que deben cumplirse para que el algoritmo siga funcionando.

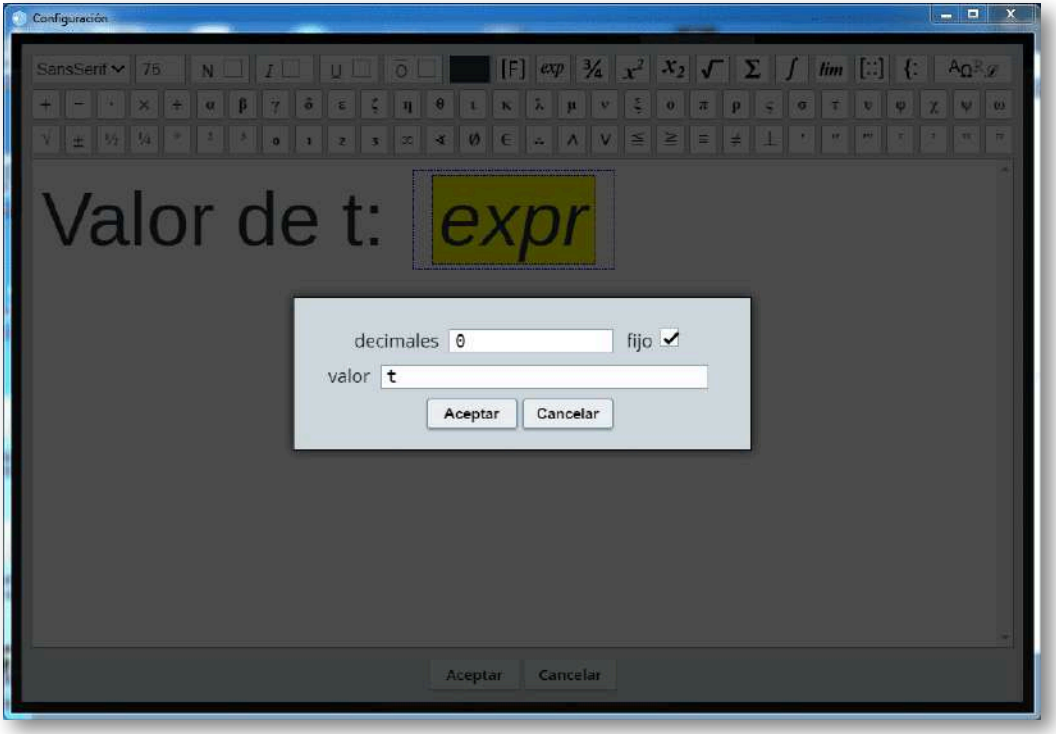
A continuación, comentamos el funcionamiento de la animación mediante un ejemplo. Diseñaremos una animación sencilla consistente en un contador que irá desde el valor 5 hasta el valor 1000, aumentando de uno en uno.

- Activamos el menú **Animación** y el control **Auto** para que la animación comience automáticamente al abrir la escena, y dejamos desactivada la opción **Repetir** para que la animación no se ejecute indefinidamente. En el campo **Pausa** dejamos el valor por defecto (si quisiéramos que la animación fuera más lenta, aumentaríamos este valor, y viceversa).

- En el campo **Inicio**, asignamos a la variable **t** el valor 5, es decir: **t = 5**. Esto sitúa nuestro contador en 5 al comienzo de la animación.
- En el campo **Hacer**, indicamos que el valor de **t** aumente en una unidad en cada ciclo: **t = t + 1**. Si quisiéramos que contase de dos en dos, escribiríamos **t = t + 2**, o de tres en tres **t = t + 3**.
- En el campo **Mientras**, definimos la condición que debe cumplirse para que el contador continúe: **t < 1000**. De esta forma, cuando **t** alcance 1000, la animación se detendrá automáticamente.



- Ahora necesitamos que el valor de **t** se muestre en la escena. Para ello, creamos un **texto con formato** que indicará el valor de la variable **t**. (También se podría haber escrito directamente en el campo texto **[t]** sin usar el editor de texto, pero en ese caso las opciones de formato serían más limitadas).


Configuración
SansSerif
76
Valor de t:
expr
decimales
0
fijo
valor
t
Aceptar
Cancelar
Aceptar
Cancelar


Configuración
SansSerif
76
Valor de t:
expr
Aceptar
Cancelar

De esta forma, la escena se actualizará automáticamente y mostrará el valor de **t** al inicio, durante y al final de la animación.



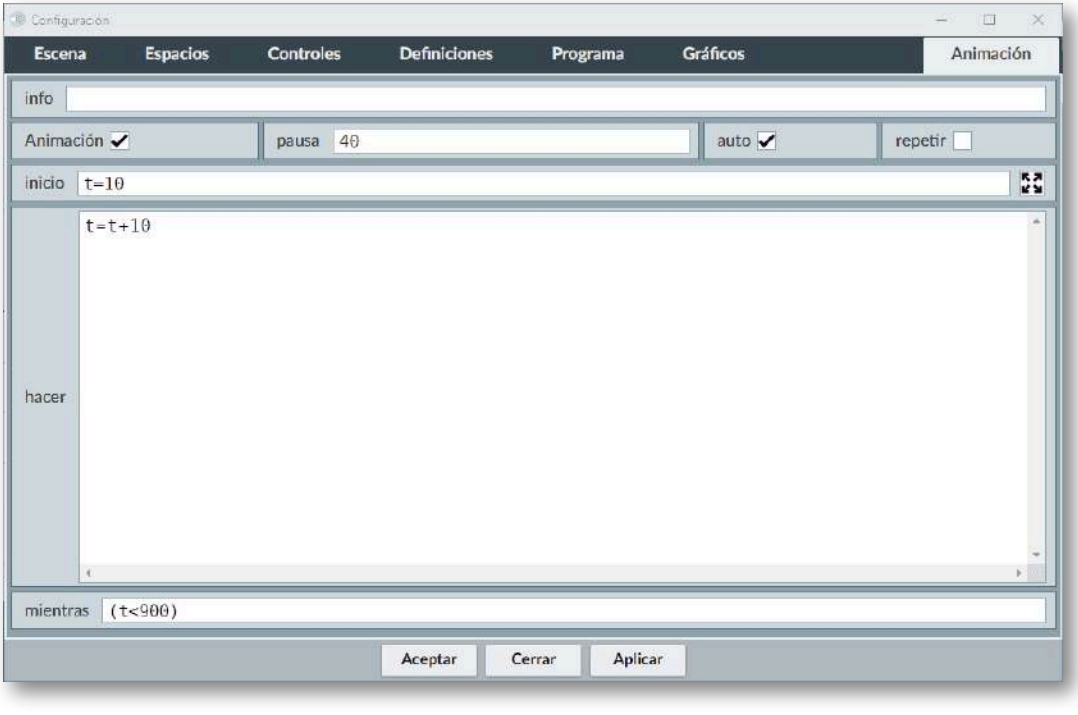


2.14 Movimiento en una escena

Vamos a diseñar una sencilla escena con animación en la que haremos que una imagen se mueva por el espacio en el que está definida.

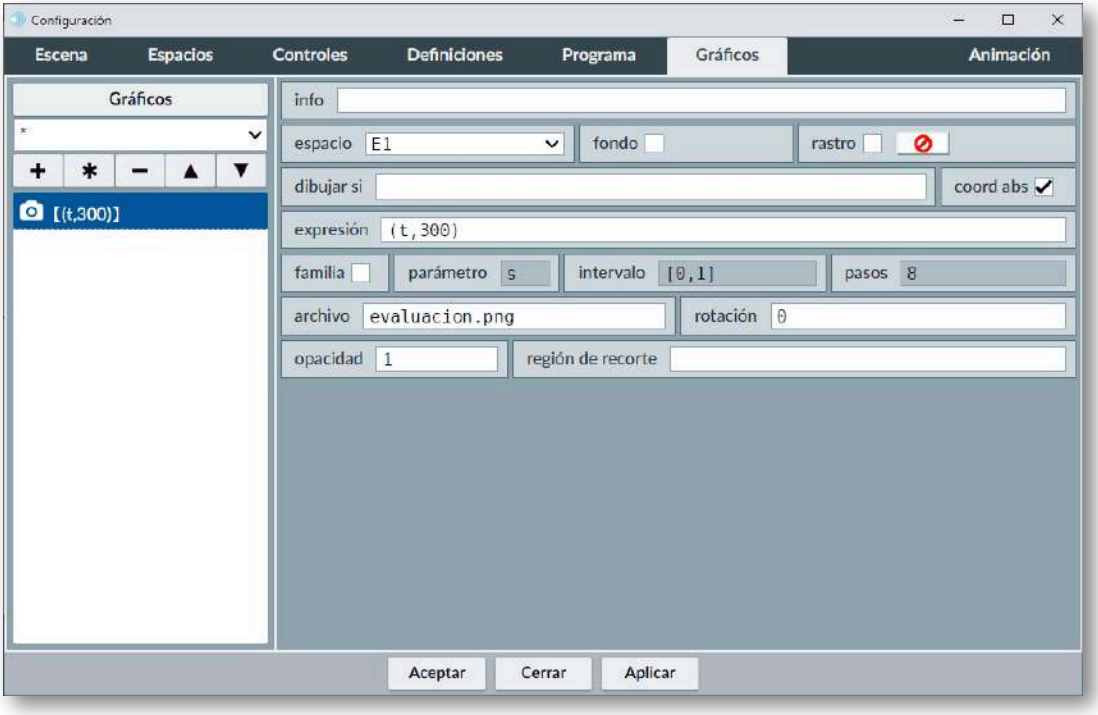
En primer lugar, definimos la animación:

- Activamos los campos **menú Animación** y **Auto** para que la animación comience automáticamente al abrir la escena. No activamos **Repetir** para que no se ejecute indefinidamente y dejamos el valor de **Pausa** sin modificar.
- En **Inicio**, asignamos a la variable **t** el valor 10, es decir: **t = 10**.
- En **Hacer**, indicamos que el valor de **t** aumente de 10 en 10 en cada ciclo: **t = t + 10**.
- En **Mientras**, establecemos la condición de que los ciclos se repitan mientras **t** sea menor que 900: **t < 900**.



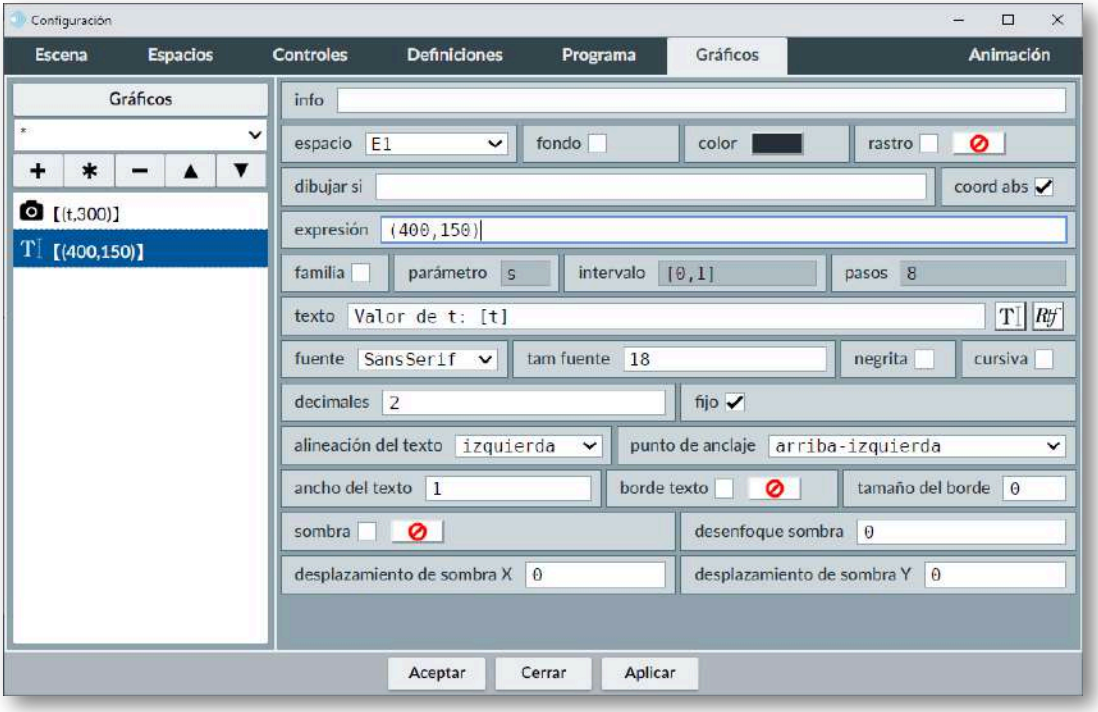
Ahora insertamos una imagen a partir del archivo **evaluacion.png** (es importante que la imagen esté en la misma carpeta que la escena o en la carpeta del editor si se está editando).

La posición de la imagen se definirá como **(t, 300)**, asegurándonos de que esté marcado el campo **coord_abs**. De esta forma, la coordenada horizontal variará a medida que avance la animación, desplazando la imagen de izquierda a derecha.

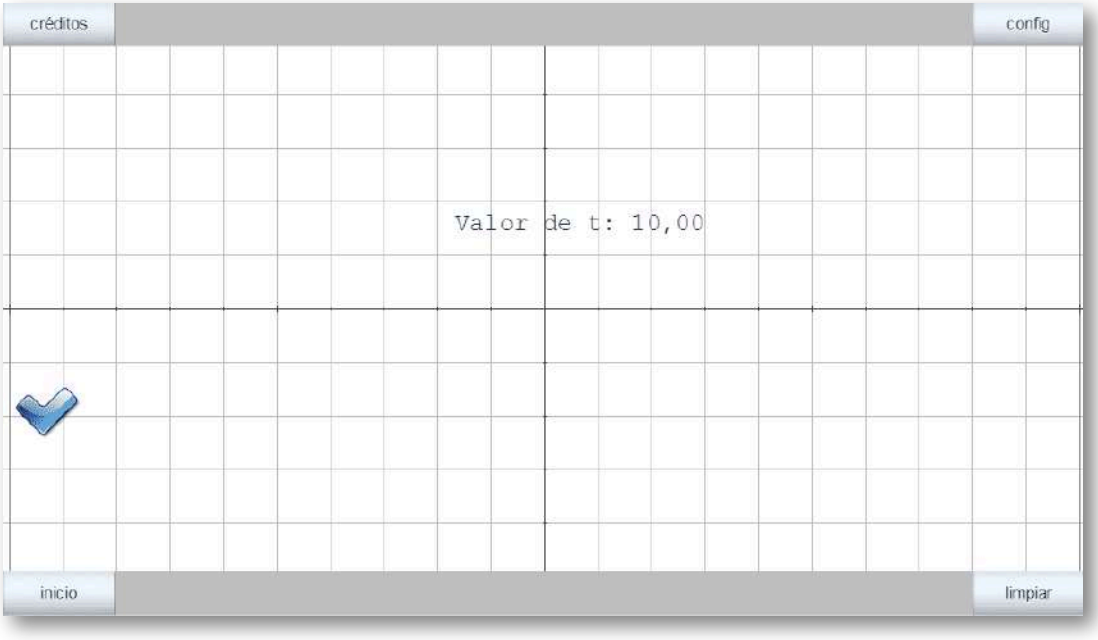


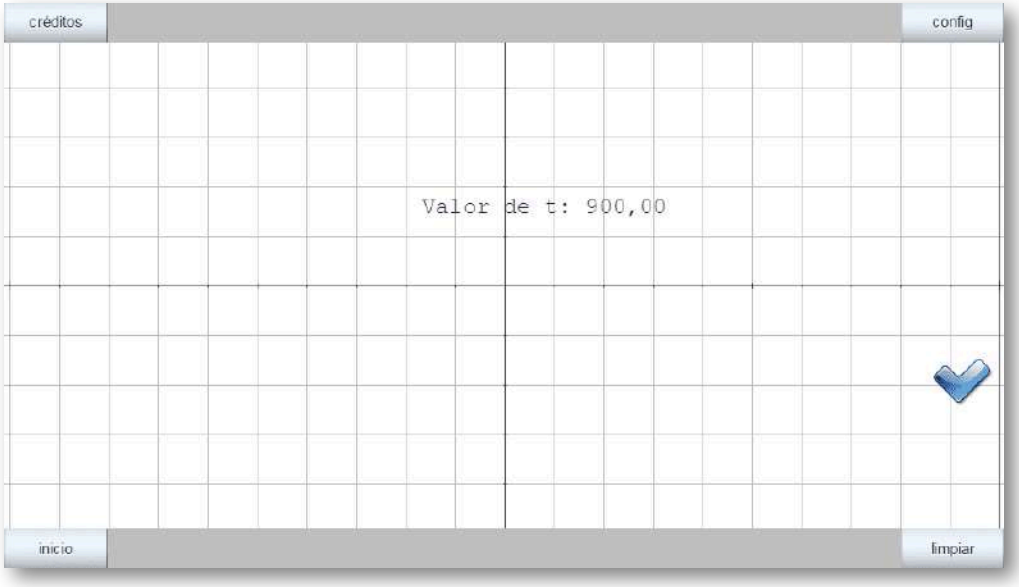
También introduciremos un texto para mostrar el valor de la variable **t** en cada instante. En el campo **texto** escribimos: **Valor de t: [t]**, donde **[t]** mostrará el valor de la variable **t** en tiempo real durante la animación.

En la siguiente captura de pantalla podemos apreciar que tanto la imagen como el texto que contiene la representación del valor de la variable **t** pertenecen al selector **Gráficos**.

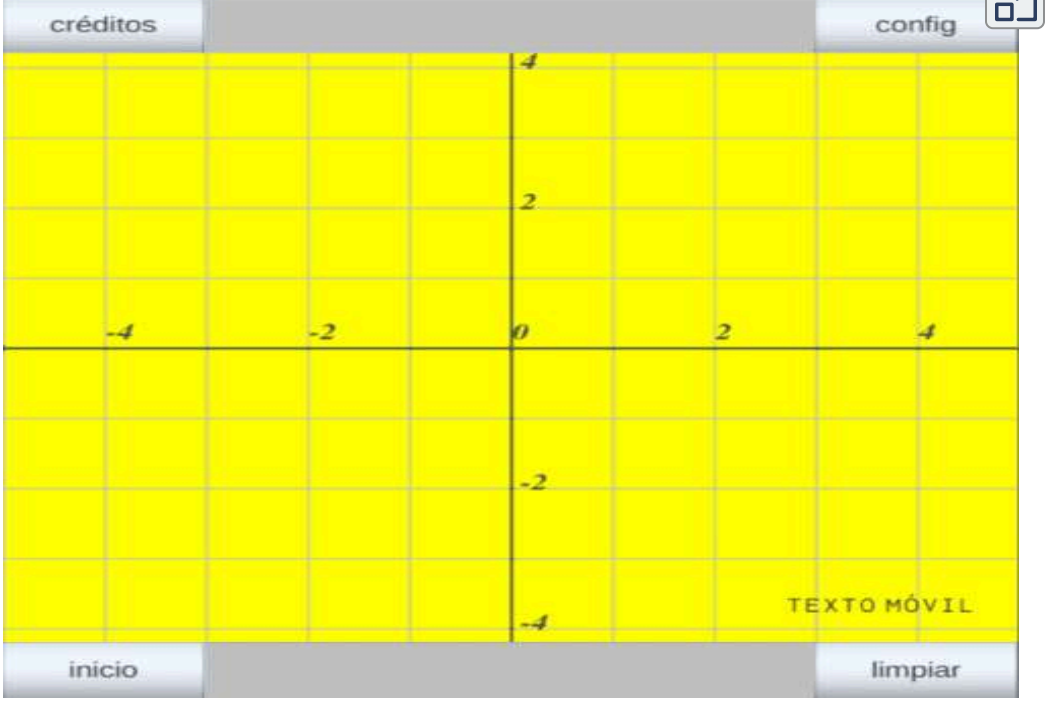


A continuación, se muestran imágenes de la escena al inicio, durante y al final de la animación:

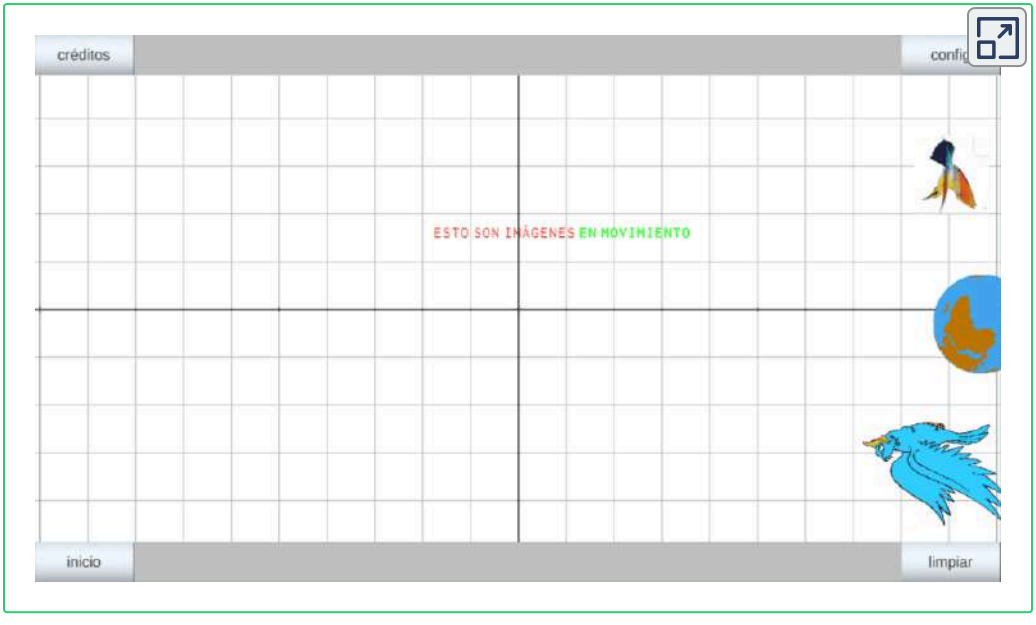




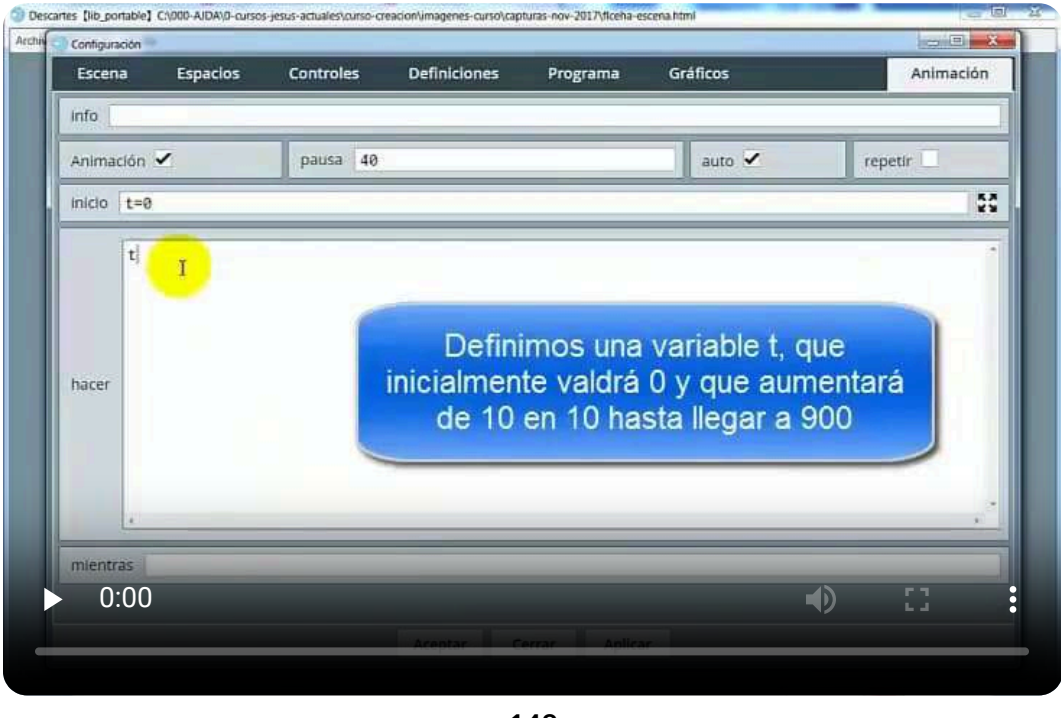
En la siguiente escena podemos ver como un texto se mueve de esquina a esquina.

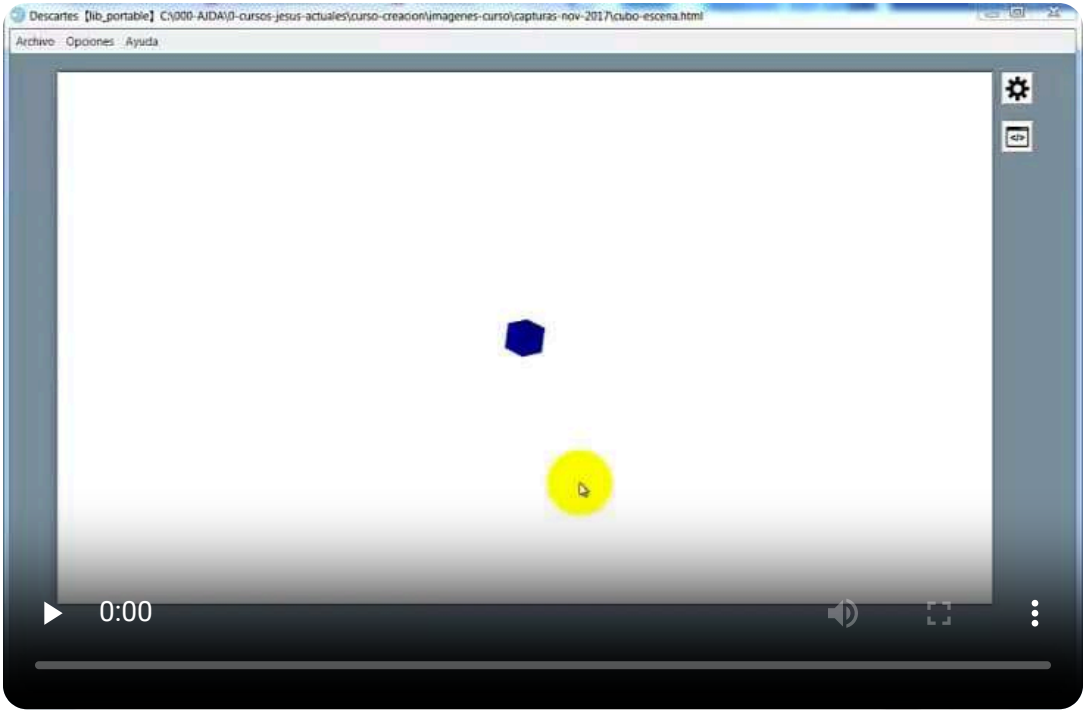


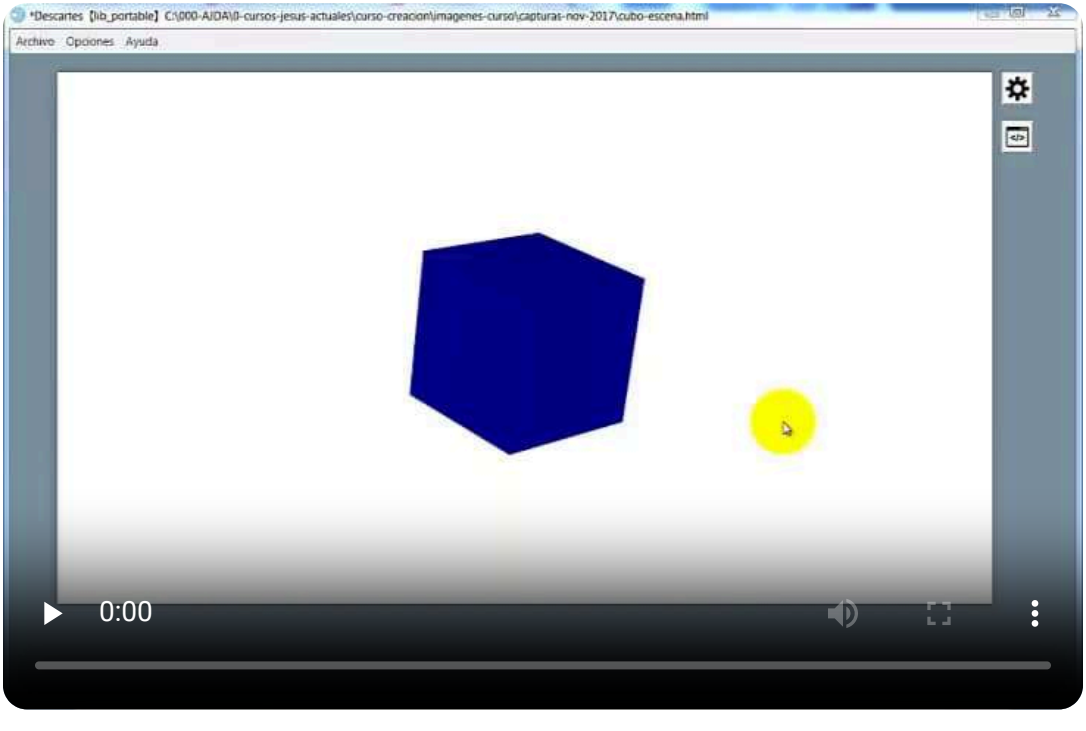
En la próxima escena se animan imágenes en traslación y rotación.



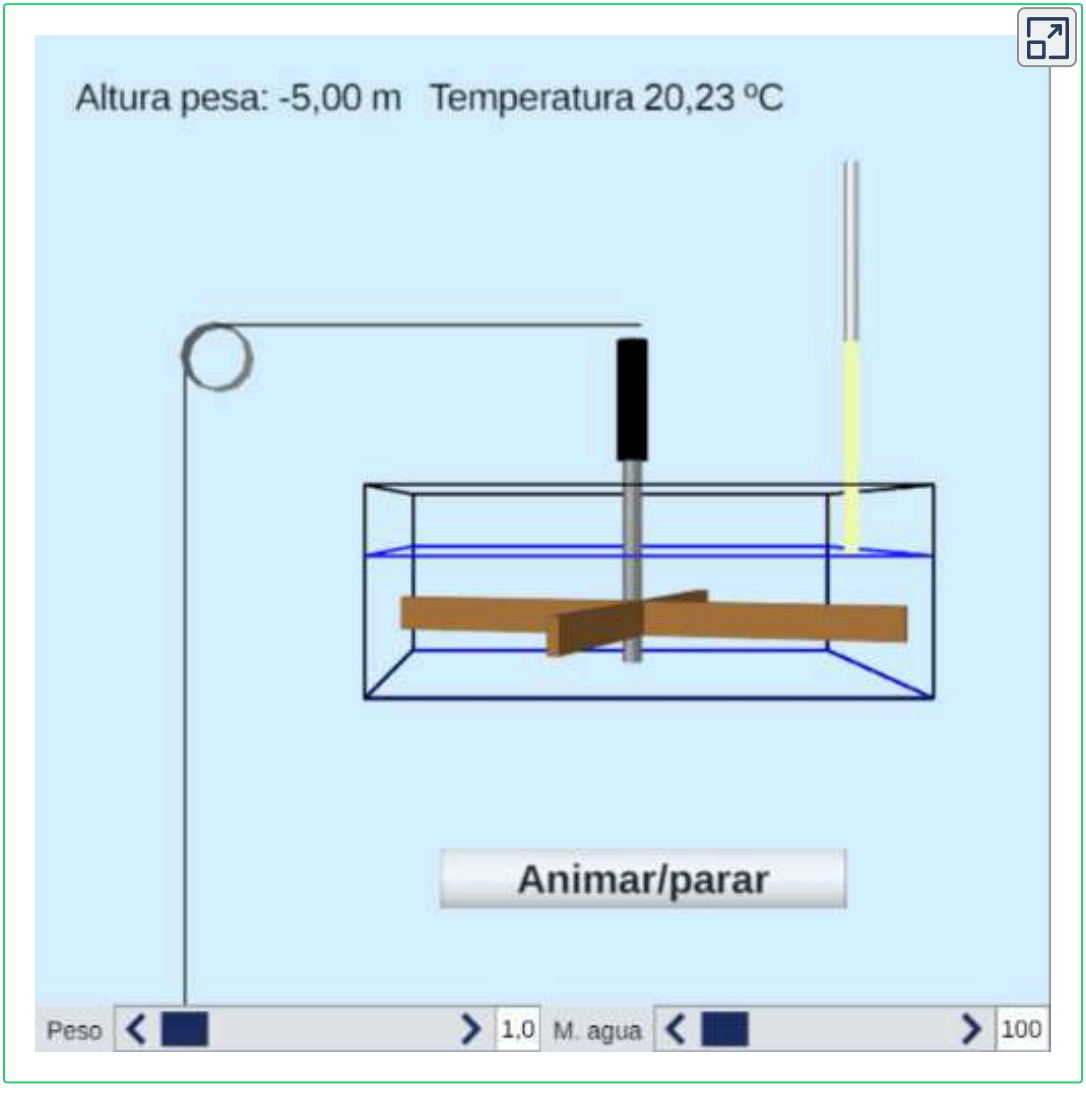
En los próximos vídeos se ilustra la creación de escenas animadas.




Descartes [lib_portable] C:\000-AJDA\0-cursos-jesus-actuales\curso-creacion\imagenes-curso\capturas-nov-2017\cubo-escena.html
Archivo Opciones Ayuda
0:00

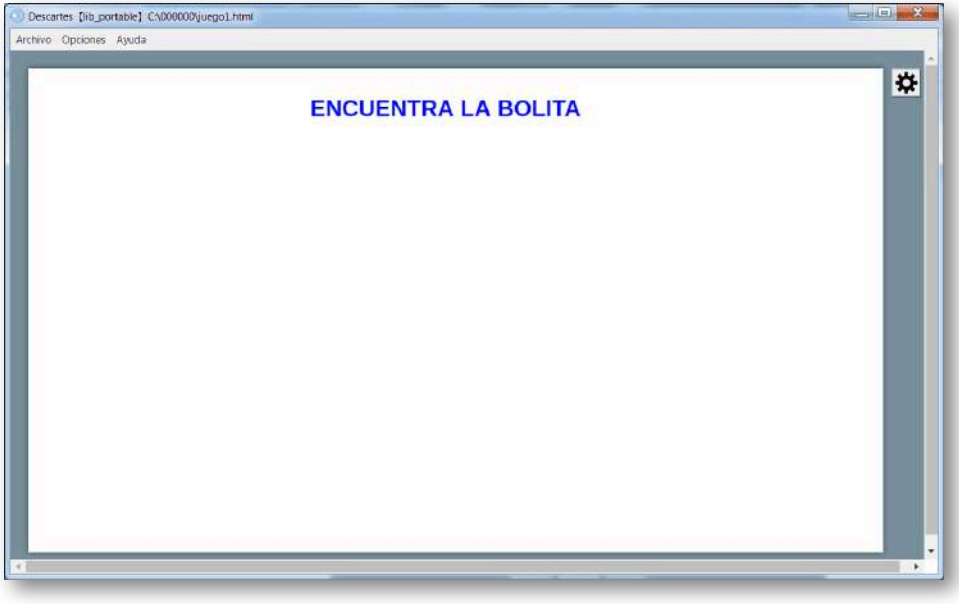

*Descartes [lib_portable] C:\000-AJDA\0-cursos-jesus-actuales\curso-creacion\imagenes-curso\capturas-nov-2017\cubo-escena.html
Archivo Opciones Ayuda
0:00

A modo de ilustración práctica, presentamos una escena animada que recrea el experimento de Joule, relacionado con la equivalencia entre el trabajo mecánico y el calor. Este interactivo está diseñado sobre un espacio 3D e incorpora distintos cuerpos geométricos tridimensionales, así como elementos dinámicos que permiten visualizar el proceso físico de forma más intuitiva. La animación facilita la comprensión del fenómeno, mostrando cómo el trabajo realizado se transforma en energía térmica.



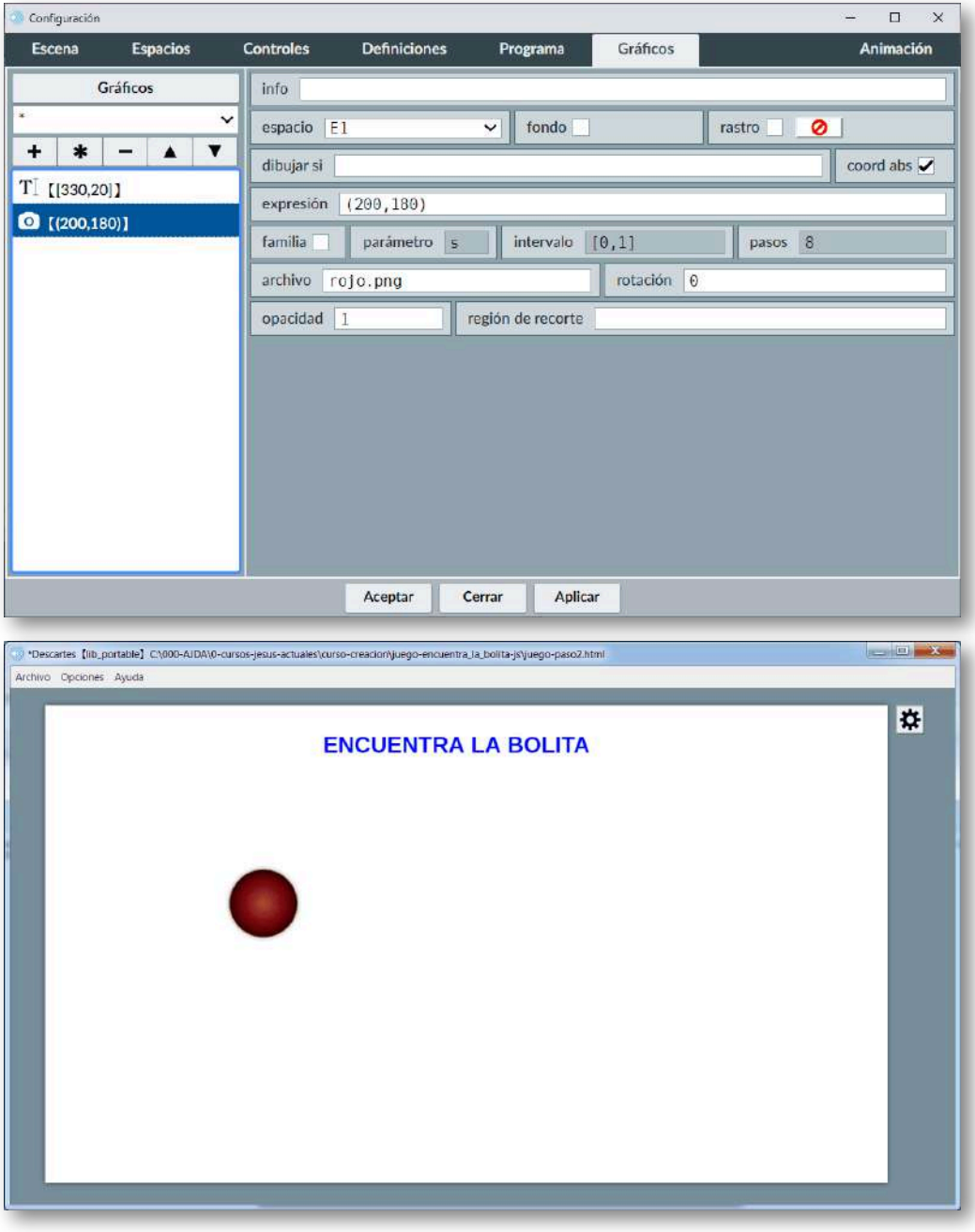
2.15 Mi primer juego paso a paso. Paso 2

Vamos a realizar el **paso 2** del juego que comenzamos en el capítulo anterior. En primer lugar, abrimos el juego con el editor de **Descartes** mediante el menú **Archivo > Abrir** y seleccionamos el archivo llamado **juego-paso1.html**.

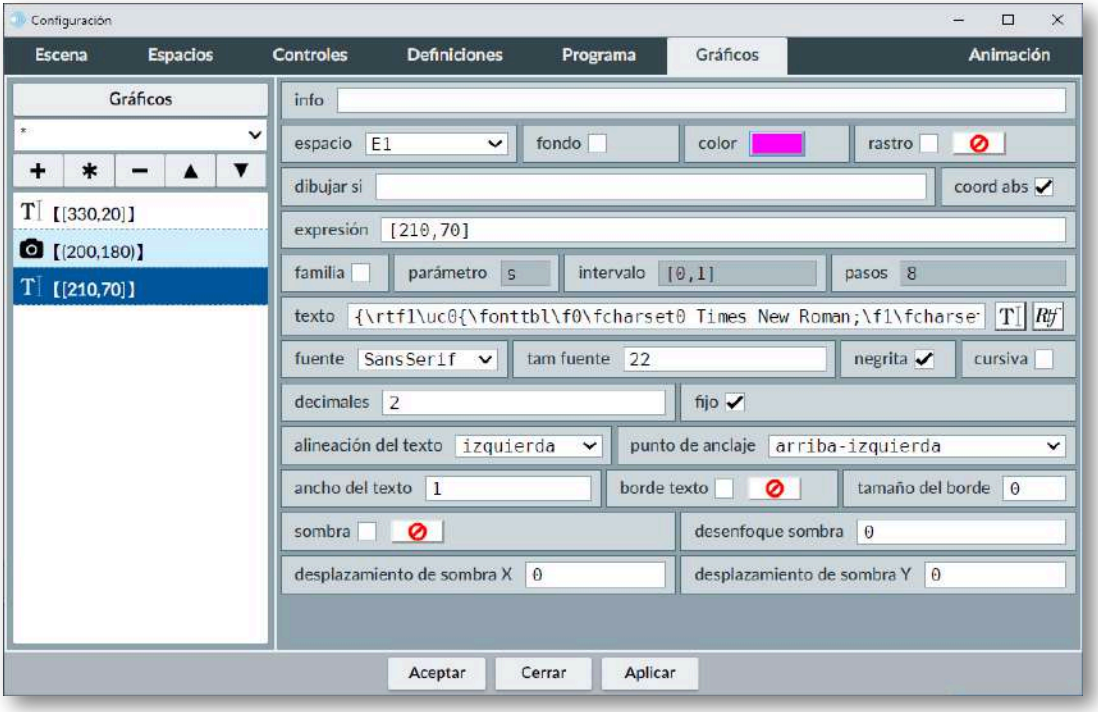


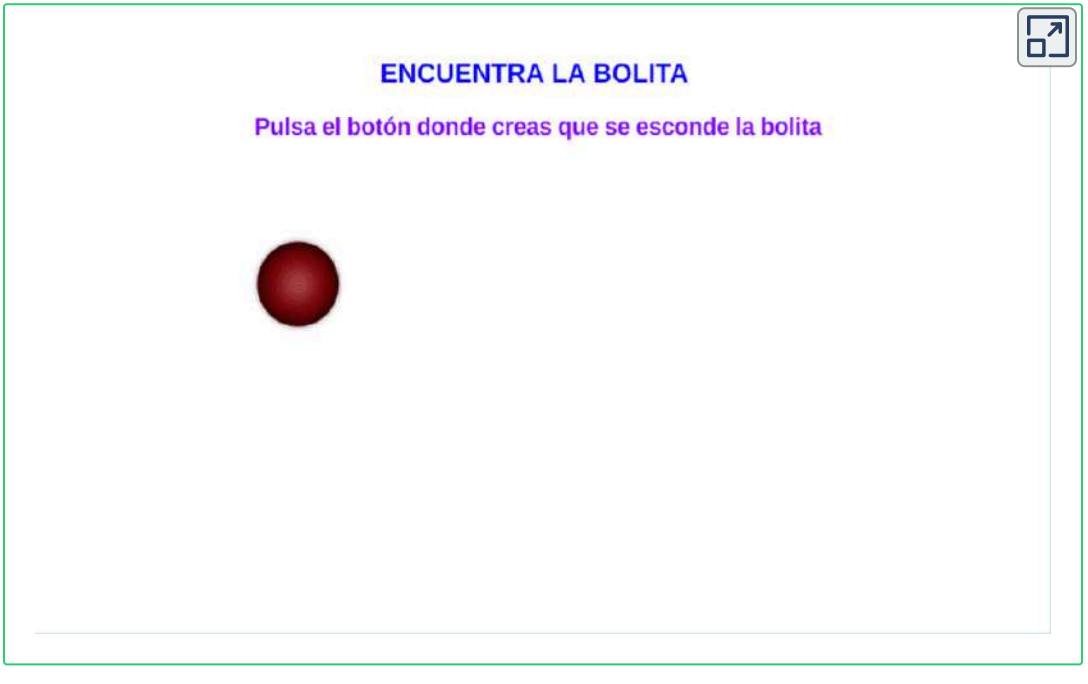
A continuación, introduciremos la imagen de una bolita roja, que se puede descargar en el siguiente enlace, y que deberá guardarse junto al fichero de la escena. Para ello, en el editor de escenas, utilizaremos el menú **Gráficos** (botón **+** > **Agregar imagen**) para insertar la imagen del fichero llamado **rojo.png**. La posición de la imagen será **(240, 140)**, asegurándonos de marcar el botón de **coordenadas absolutas**.





Finalmente, añadiremos un texto bajo el título del juego mediante el menú **Gráficos** (botón **+** > **Agregar texto**), con el formato que deseemos, con la frase: **"Pulsa el botón donde creas que se esconde la bolita"**.

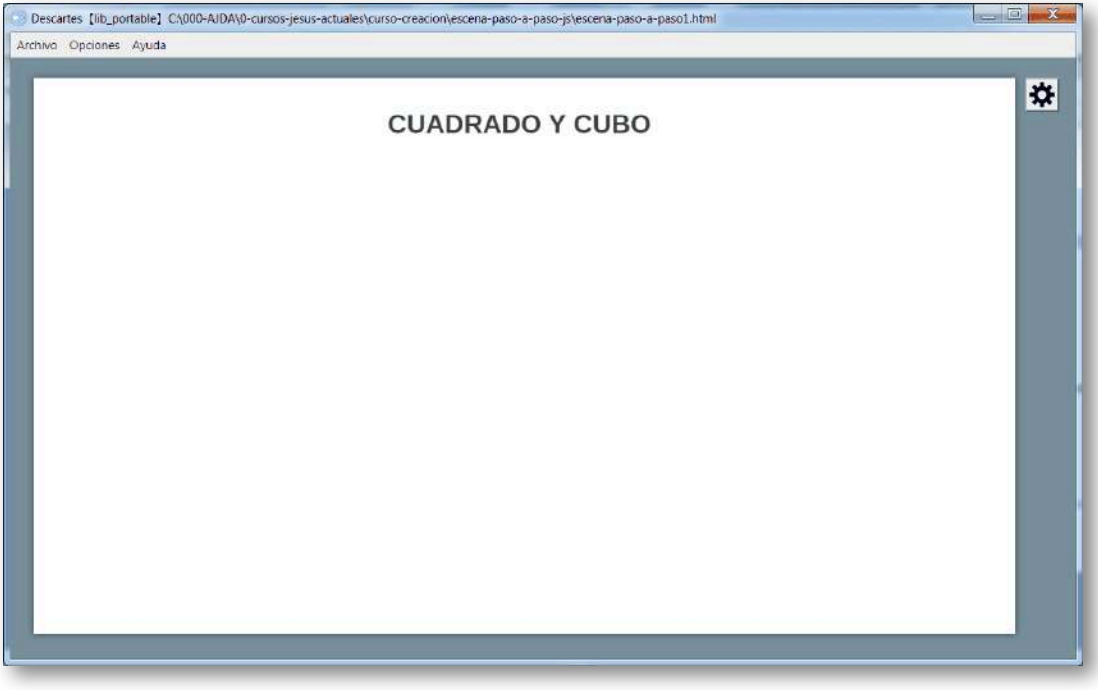




Guardamos la escena del juego con el nombre **juego-paso2**.

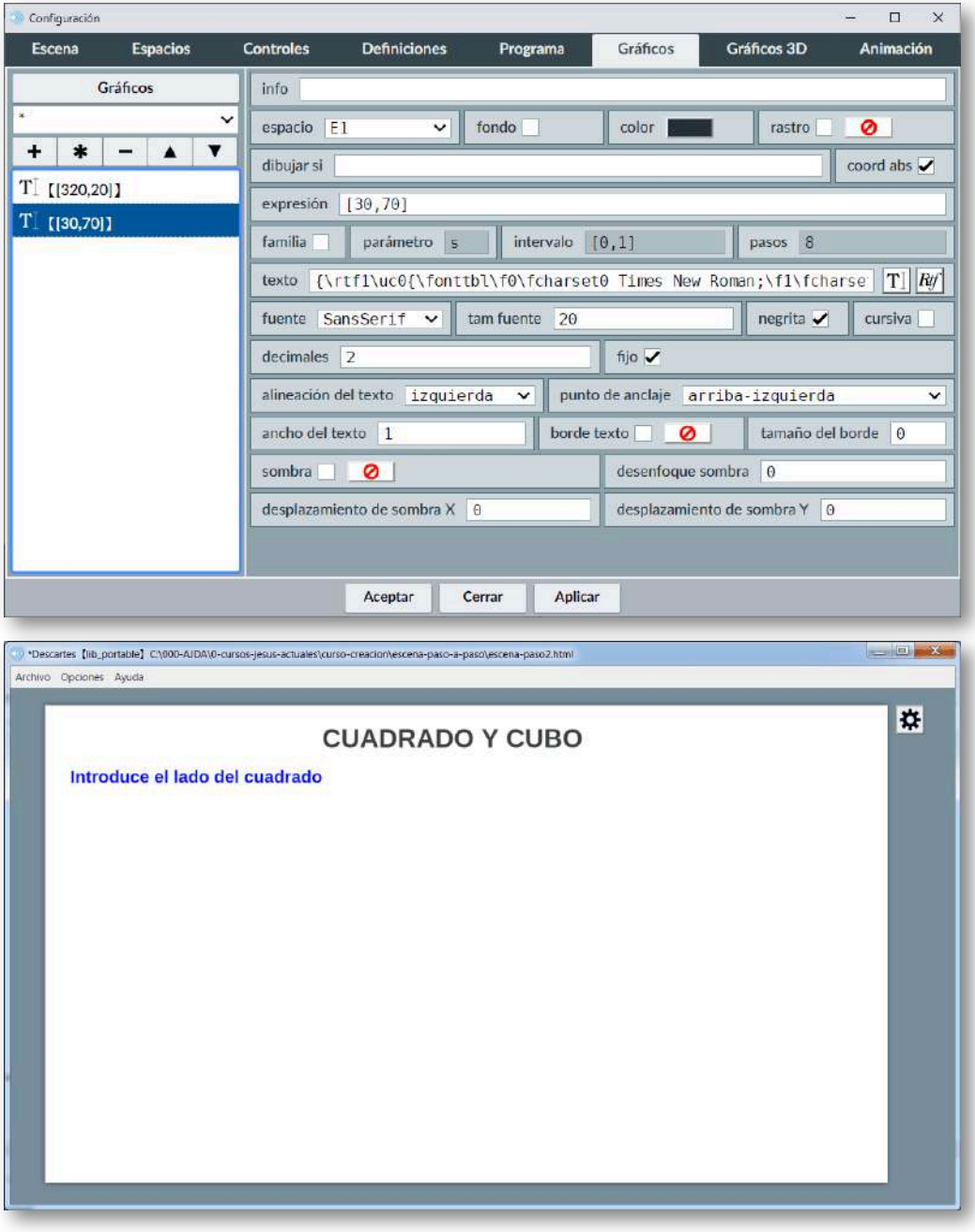
2.16 Mi primera escena paso a paso. Paso 2

Vamos a realizar el **paso 2** de nuestra primera escena que comenzamos en el capítulo anterior. En primer lugar, abrimos la escena con el editor de **Descartes** mediante el menú **Archivo > Abrir** y seleccionamos el archivo llamado **escena-paso1**.

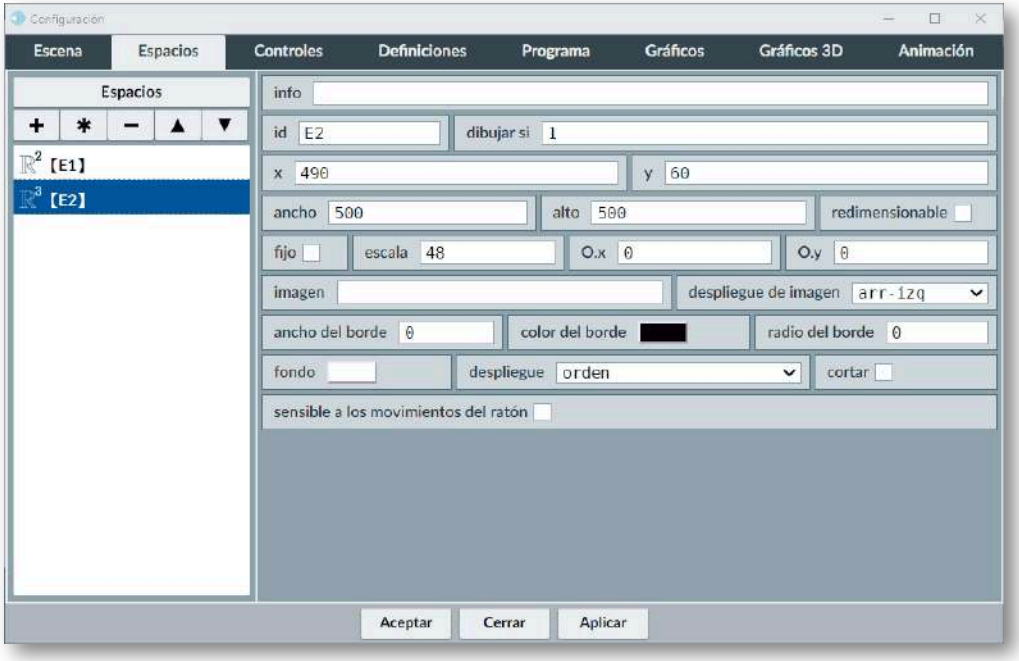


Añadiremos un texto bajo el título del juego, a la izquierda, mediante el menú **Gráficos** (botón **+** > **Agregar texto**), con el formato que deseemos, con la frase: **"Introduce el lado del cuadrado"**.

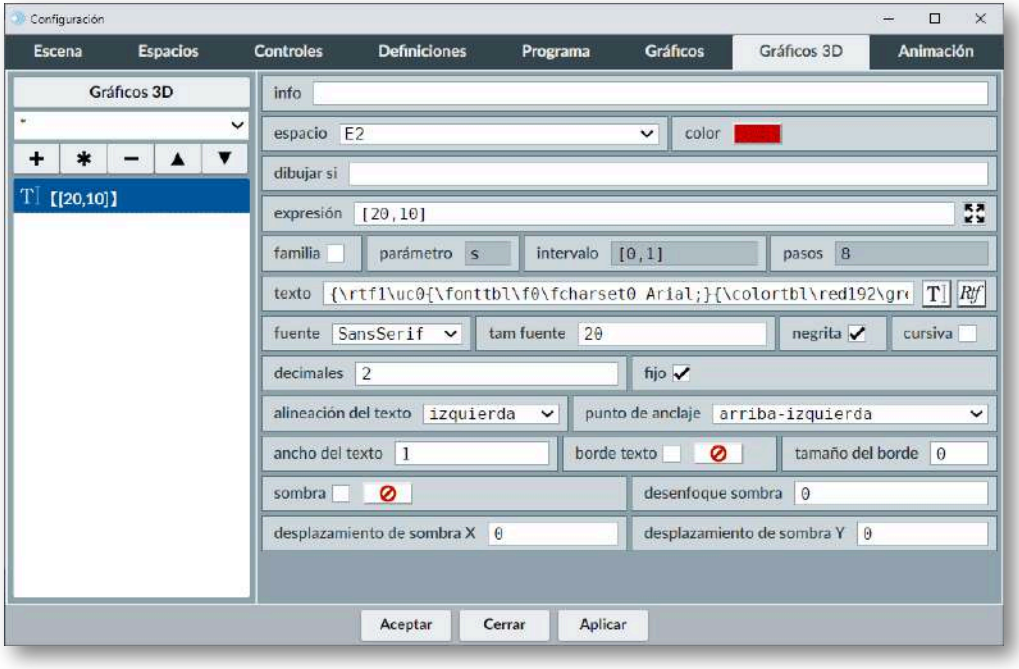


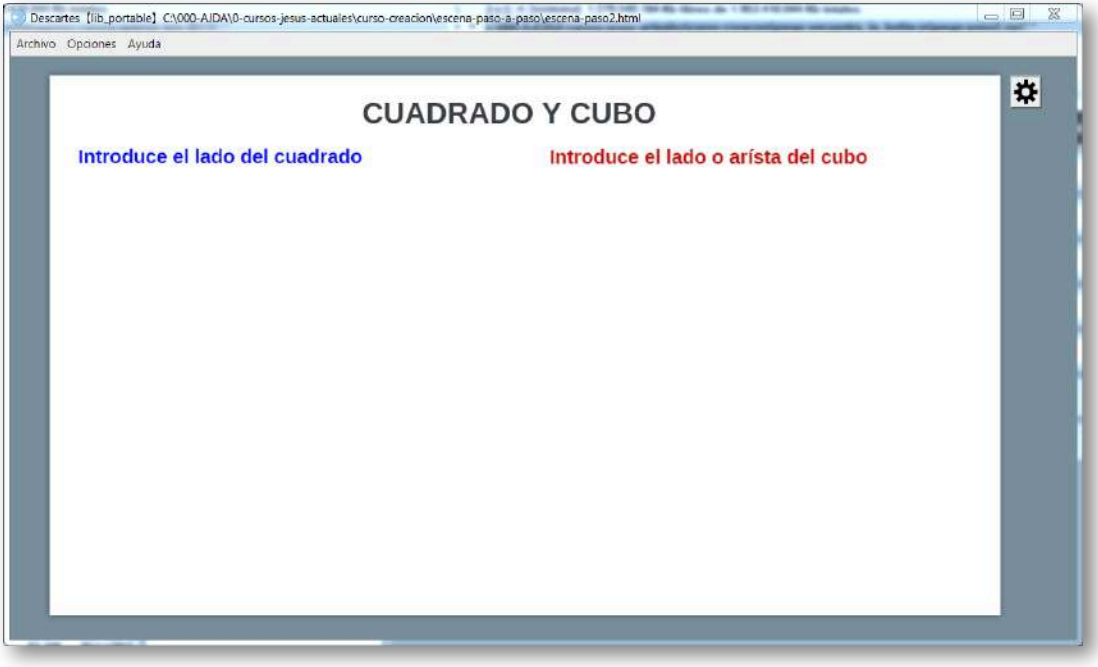


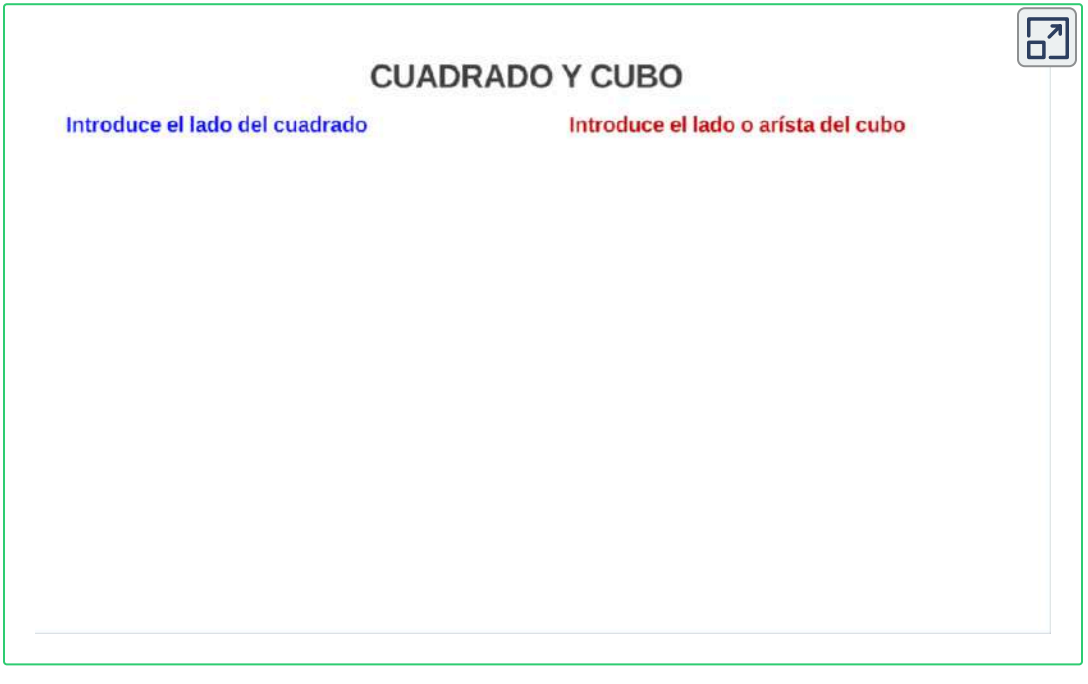
Seguidamente, creamos un espacio **3D** mediante el menú **Espacio** (botón **+** > **Agregar espacio R3**), con fondo transparente (no se distinguirá del color del espacio 2D), en la posición **(490, 40)** y con un ancho y alto de **500**.



Incluiremos un texto dentro del espacio 3D mediante el menú **Gráficos 3D** (botón **+** > **Agregar texto**), con el formato que deseemos, con la frase: **"Introduce el lado o arista del cubo"**.







Finalmente, guardamos la escena con el nombre **escena-paso2**.



Capítulo 3

Controles

Capítulo 3
Controles

3.1 Introducción 161

3.2 Tipos de controles 162

3.3 Introducción de controles 164

3.4 Parámetros comunes a múltiples controles 167

3.5 Control tipo Pulsador 171

3.6 Control tipo Campo de texto 175

3.7 Control tipo Menú 178

3.8 Control tipo Barra 182

3.9 Control tipo Deslizador 185

3.10 Control tipo Botón 187

3.11 Control tipo Casilla de verificación 193

3.12 Control tipo Gráfico 198

3.13 Control tipo Texto 207

3.14 Control tipo Audio 210

3.15 Control tipo Vídeo 213

3.16 Mi primer juego paso a paso. Paso 3 217

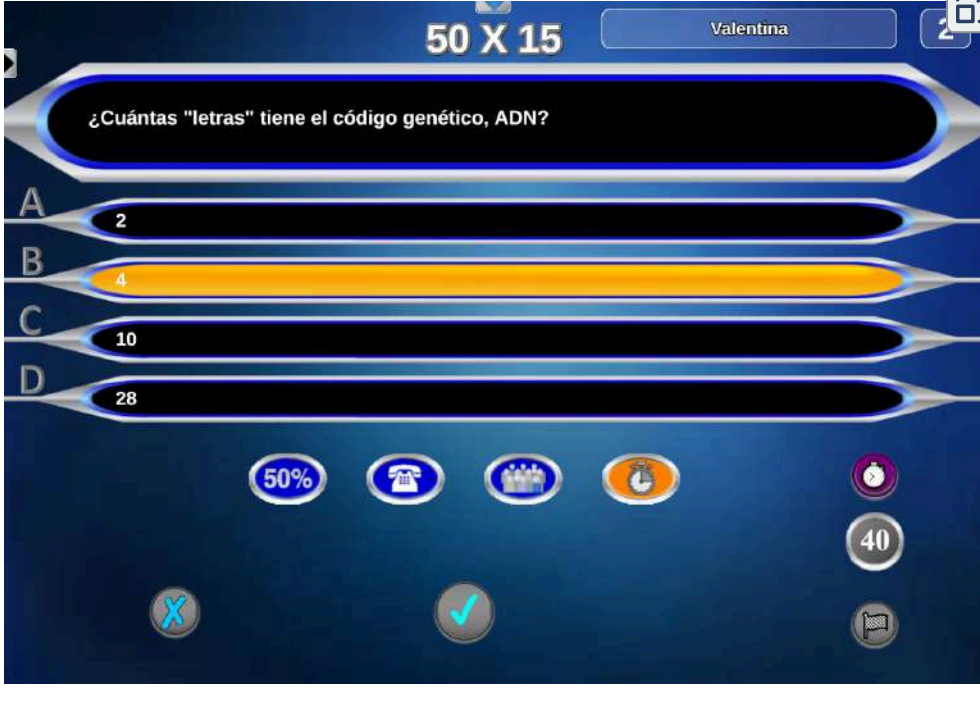
3.17 Mi primera escena paso a paso. Paso 3 222

3.1 Introducción

Los controles son elementos de los interactivos que permiten al usuario modificar los parámetros de la escena mientras la está utilizando o provocar algún tipo de acción dentro de la misma. Entre ellos se encuentran botones, menús, barras de deslizamiento, controles gráficos, sonidos o vídeos, que hacen posible la interacción directa con la escena.

En este capítulo estudiaremos los distintos tipos de controles que pueden emplearse en las escenas de Descartes.

Como ejemplo de escena con una amplia variedad de controles, incluimos la escena del juego didáctico AJDA "50x15", una versión adaptada del conocido concurso de televisión ¿Quién quiere ser millonario?



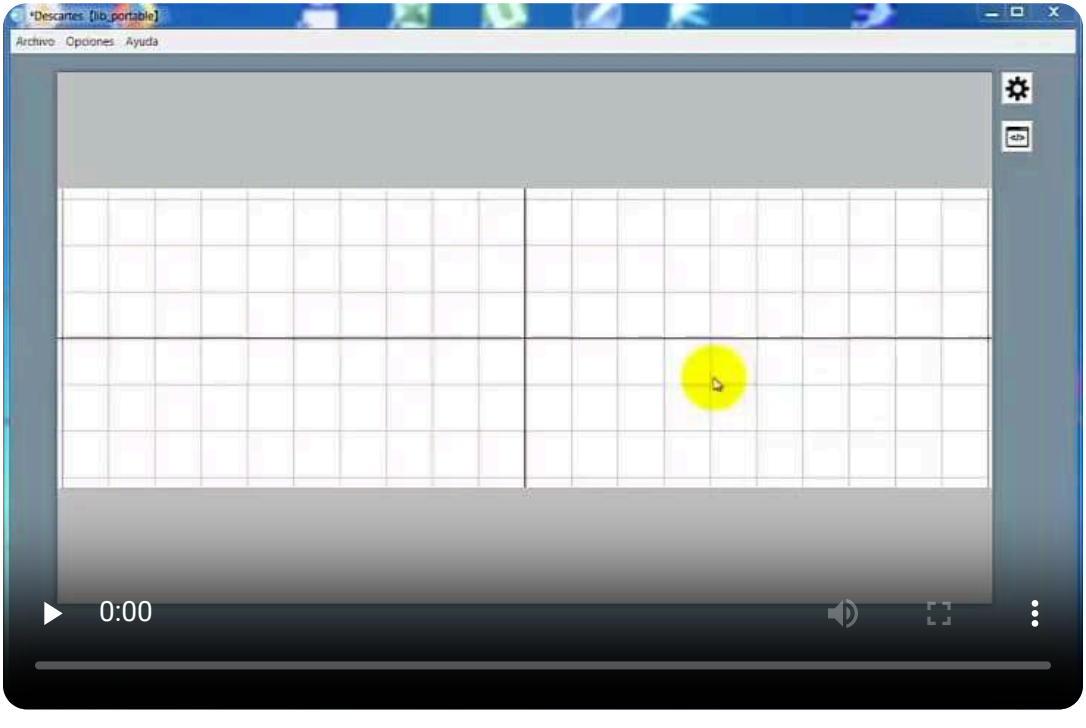
3.2 Tipos de controles

Las escenas pueden incorporar distintos tipos de controles que permiten al usuario interactuar y modificar su funcionamiento. Permiten cambiar el valor de una variable o ejecutar determinadas acciones. A través de ellos se pueden realizar cálculos, ejecutar funciones definidas en sus parámetros, iniciar, reiniciar o animar la escena, abrir una página web u otra escena interactiva, o reproducir un archivo sonoro, entre otras posibilidades. A los seis primeros de la siguiente relación se les denomina **controles numéricos**.

- **Pulsador.** Permite introducir valores numéricos, bien escribiéndolos directamente en su campo, bien utilizando los botones de incremento y decremento.
- **Menú.** Ofrece un conjunto de opciones seleccionables en formato texto; cada opción tiene asociado un valor numérico que es el que se asigna en la escena.
- **Barra.** Permite fijar un valor numérico desplazando un control deslizante dentro de un intervalo determinado.
- **Deslizador.** Los deslizadores (o sliders en inglés) son controles con comportamiento similar a la barra, pero donde el objeto deslizable es un círculo que se desliza a lo largo de un eje.
- **Botón.** Ejecuta una acción concreta al pulsarlo.
- **Campo de texto.** Permite introducir en la escena un valor numérico o literal.
- **Casilla de verificación.** Puede marcarse o desmarcarse para activar o desactivar una opción. También puede utilizarse como botón de opción cuando varias casillas están relacionadas entre sí.
- **Gráficos.** Son objetos interactivos que permiten modificar parámetros directamente sobre la escena.

- **Texto.** Son áreas en las que el usuario puede escribir o corregir textos y fórmulas, y comprobar si la respuesta es correcta comparándola con la solución, accesible mediante el botón S (Solución).
- **Audio.** Consiste en un reproductor de audio que puede ser activado en el interactivo. Se puede colocar sólo en el interior de un espacio. Reproduce archivos *mp3* y *wav*.
- **Vídeo.** Consiste en un reproductor de vídeo que puede ser activado en el interactivo. Su funcionamiento es muy similar al control de audio. Se utiliza para reproducir archivos con formato *mp4*, *WebM* y *ogv*.

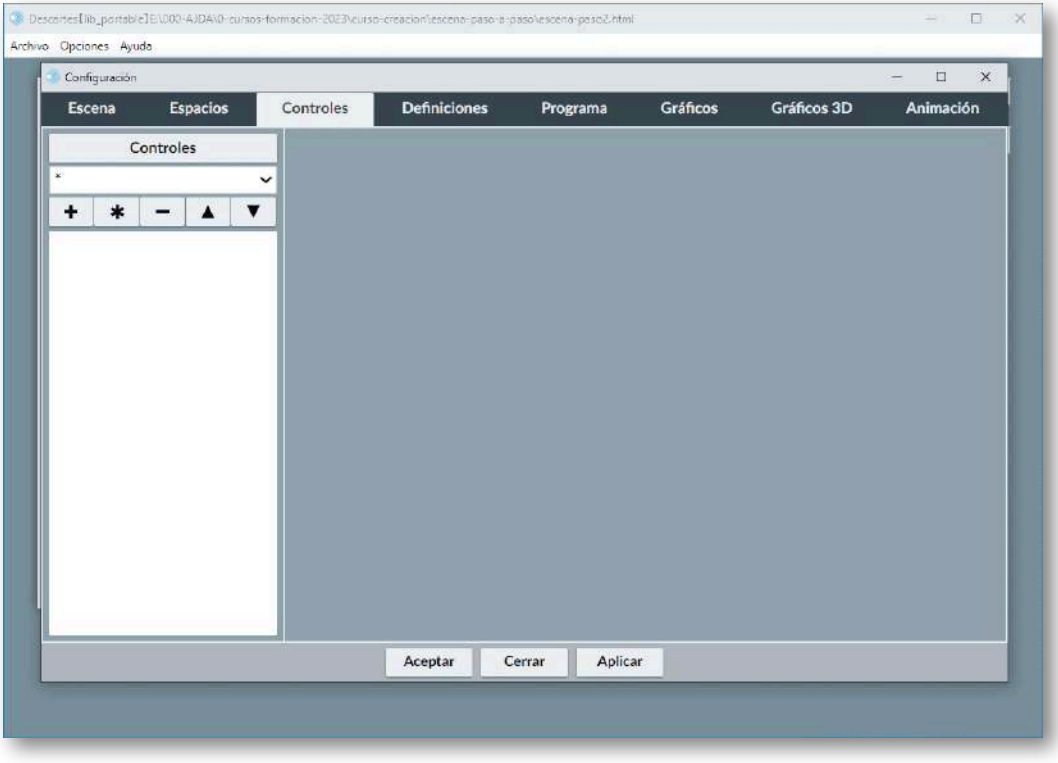
En el siguiente vídeo se realiza una presentación general de los distintos tipos de controles que pueden utilizarse en las escenas de Descartes, así como del procedimiento para incorporarlos y configurarlos.



3.3 Introducción de controles

El procedimiento para introducir un control en una escena es similar al que se emplea para añadir imágenes o textos.

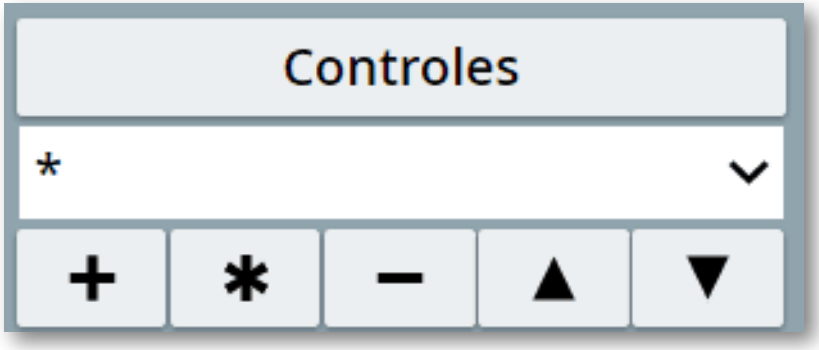
1. Acceder al panel de configuración de la escena y pulsar sobre el apartado **Controles**.

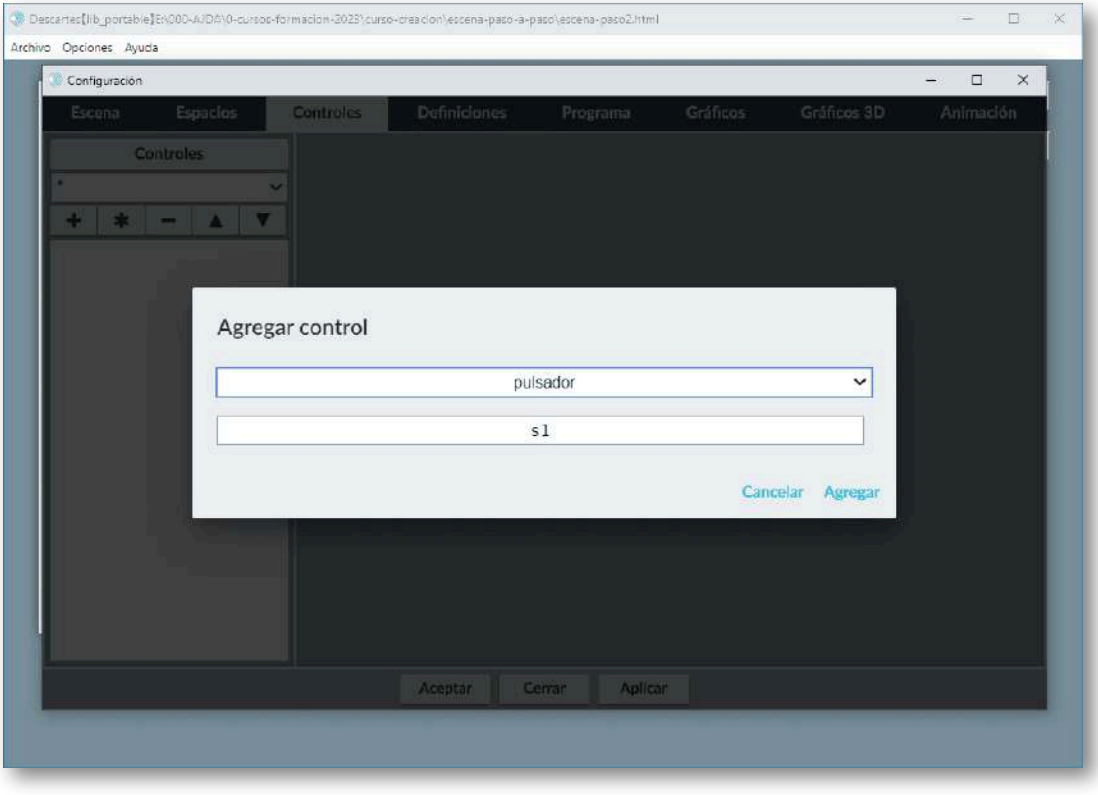


2. Añadir un nuevo control desde el panel de controles situado a la izquierda.

    1. Pulsar el botón **+** para crear un nuevo control (el botón **-** lo elimina, * lo duplica y los pulsadores permiten subir o bajar los controles ya creados). El botón **Controles** permite visualizar el código de los controles definidos.

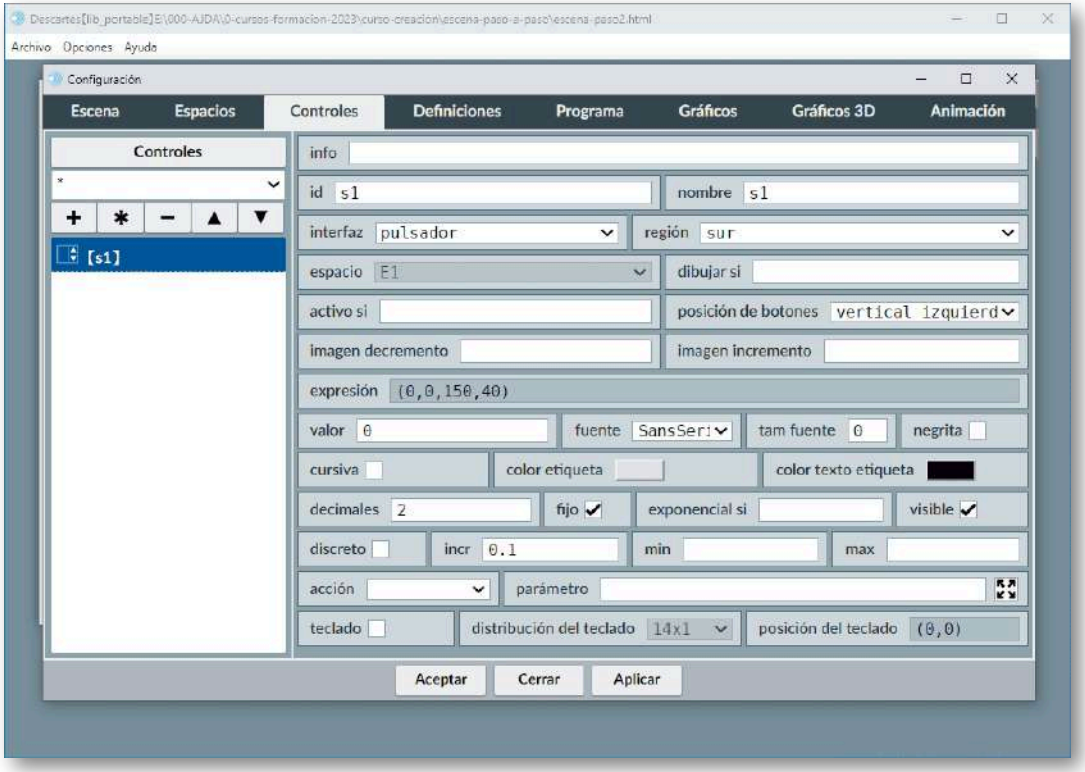
    2. Seleccionar en el menú el tipo de control que se desea añadir, por ejemplo, **Pulsador**.



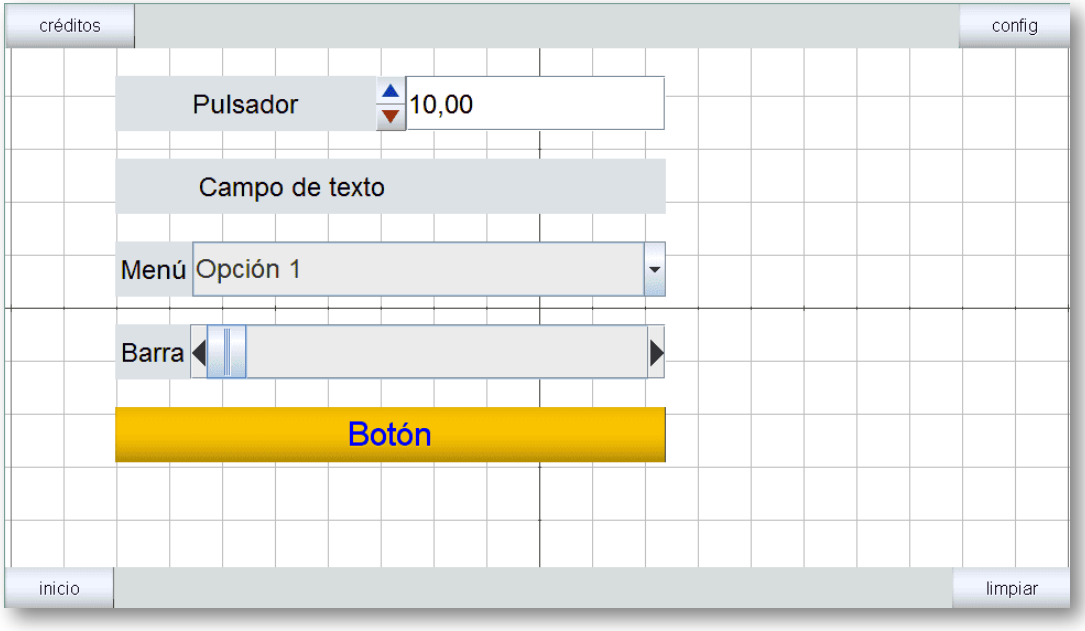


3. Configurar los parámetros del control (espacio, nombre, tipo, condición de dibujar, valor, condición de activo, acción, parámetro, entre otros).

4. Por último, pulsar en **Aplicar** o **Aceptar** para que el control quede creado en la escena.



En la siguiente imagen se muestran cómo quedan algunos controles.



3.4 Parámetros comunes a múltiples controles

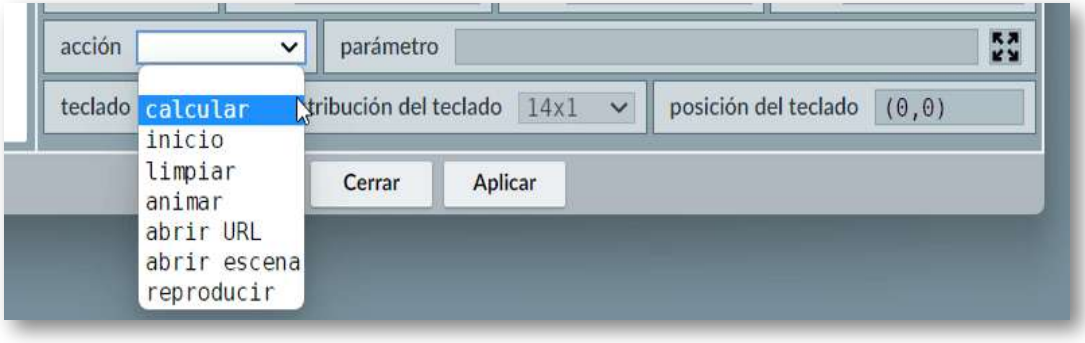
A continuación, se describen los parámetros comunes que pueden aparecer en la configuración de los controles:

- **info.** Campo de texto para introducir una breve descripción del control. Lo que se escriba aquí es lo que aparecerá en la lista del panel de controles, facilitando su localización.
- **id.** Identificador del control dentro de Descartes. Es el nombre de la variable asociada al control y no puede comenzar por un número. Puede definirse al crear el control o modificarse posteriormente.
- **nombre.** Texto que verá el usuario en el interactivo. Solo se aplica a controles numéricos. Puede coincidir o no con el identificador. Si se deja vacío o se escribe *_nada_*, no se mostrará etiqueta.
- **interfaz.** Menú que determina el tipo de control: pulsador, campo de texto, menú, barra, deslizador o botón.
- **región.** Indica dónde se ubicará el control:
  - **sur.** Barra horizontal inferior.
  - **norte.** Barra horizontal superior.
  - **este.** Barra vertical derecha.
  - **oeste.** Barra vertical izquierda.
  - **exterior.** Fuera del interactivo (requiere activar “mostrar región exterior”).
  - **interior.** Dentro de un espacio concreto.
- **espacio.** Permite elegir el espacio que alojará el control cuando la región es interior.
- **dibujar si.** Expresión booleana que determina si el control es visible (1) o no (0).

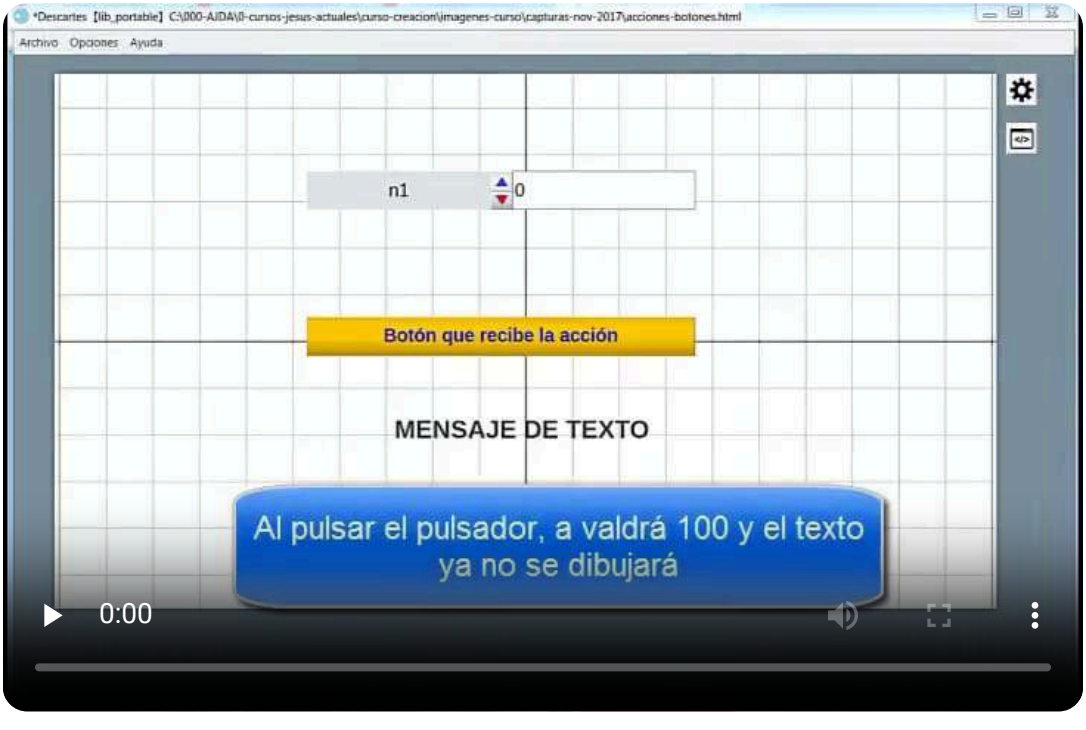
- **activo si.** Expresión booleana que determina si el control puede utilizarse (1) o queda desactivado (0).
- **expresión.** Define posición (x,y) y tamaño (ancho, alto) del control en píxeles cuando está en región interior. Siempre usa coordenadas absolutas.
- **valor.** Valor inicial del control al cargar el interactivo.
- **fuente, tam fuente, negrita, cursiva, color etiqueta, color texto etiqueta.** Opciones de formato de la etiqueta del control (solo en controles numéricos).
- **decimales.** Número de decimales que se mostrarán.
- **fijo.** Obliga a mostrar siempre el número de decimales indicado.
- **exponencial si.** Es un campo de texto que admite una expresión booleana (es decir, cuyo valor puede ser 0 o 1). Si el valor de la expresión es 0, nunca se usará la notación exponencial, pero si es 1, es posible que esta se muestre si el interactivo lo determina pertinente.
- **visible.** Determina si se muestra el valor del parámetro asociado al control.
- **discreto.** Obliga a que los valores varíen en múltiplos del incremento indicado.
- **incr.** Tamaño del incremento en controles como pulsador o barra.
- **min / max.** Valores mínimo y máximo permitidos en los valores de los controles.
- **acción.** Define qué ocurre al usar el control. Puede ser: calcular, inicio, limpiar, animar, abrir URL, abrir escena o reproducir. Cuando el parámetro acción de un control se encuentra sin acción asociada, o bien con acción inicio, limpiar, o animar, el campo parámetro asociado a la acción se inactiva, debido a que dichas acciones no involucran parámetros.

- **parámetro.** Campo asociado a la acción. Contiene asignaciones, direcciones URL, nombres de escenas o archivos de audio, según el tipo de acción.
    - **Si la acción es calcular:** el campo **parámetro** debe contener asignaciones de valores y/o llamadas a funciones que se ejecutarán al activar el control.
    - **Si la acción es abrir URL:** el **parámetro** puede incluir una dirección relativa (dentro de la misma carpeta del interactivo) o una dirección absoluta que comience, por ejemplo, por https://. Tras la dirección puede añadirse la indicación **target=_self**, **_blank**, **_parent** o **_top** para indicar dónde se abrirá el destino.
    - **Si la acción es reproducir:** el **parámetro** debe contener el nombre del archivo de audio en formato .mp3, incluyendo su extensión. Si el archivo se encuentra en una subcarpeta, es necesario especificar también la ruta correspondiente.
    - **Si la acción es abrir escena:** el **parámetro** debe indicar el nombre del archivo .html de la escena que se desea abrir, ubicada en la misma carpeta o en una subcarpeta del interactivo. Al igual que en **abrir URL**, puede añadirse **target=_self** para que la escena se abra en la misma pestaña, en lugar de hacerlo en una nueva.
- **teclado.** Activa un teclado virtual (útil en dispositivos móviles). Se encuentra presente sólo para los controles que involucran un campo de texto, tales como pulsador, campo de texto, menú y barra. Si la casilla está marcada, se activan los parámetros distribución del teclado y posición del teclado, descritos en breve.
- **distribución del teclado.** Permite elegir el tipo de teclado virtual (numérico o alfabético, con distintas disposiciones).
- **posición del teclado.** Coordenadas absolutas de la esquina superior izquierda del teclado virtual dentro de la escena.

Como hemos visto, la utilización de los controles puede ir asociada a una acción determinada, tal y como se muestra en la siguiente imagen.



En el próximo vídeo se presenta, de forma breve, la ejecución de distintas acciones mediante el uso de varios controles numéricos.



En los siguientes apartados detallaremos las particularidades de los diferentes tipos de controles que pueden incorporarse en las escenas.

3.5 Control tipo Pulsador

Este tipo de control se asocia a una variable que toma el mismo nombre que el identificador del control tipo pulsador. Permite aumentar o disminuir el valor de dicha variable mediante los botones de incremento y decremento que incorpora el propio control. Permite definir el valor del incremento y del decremento que se produce al pulsar sobre los correspondientes pulsadores.

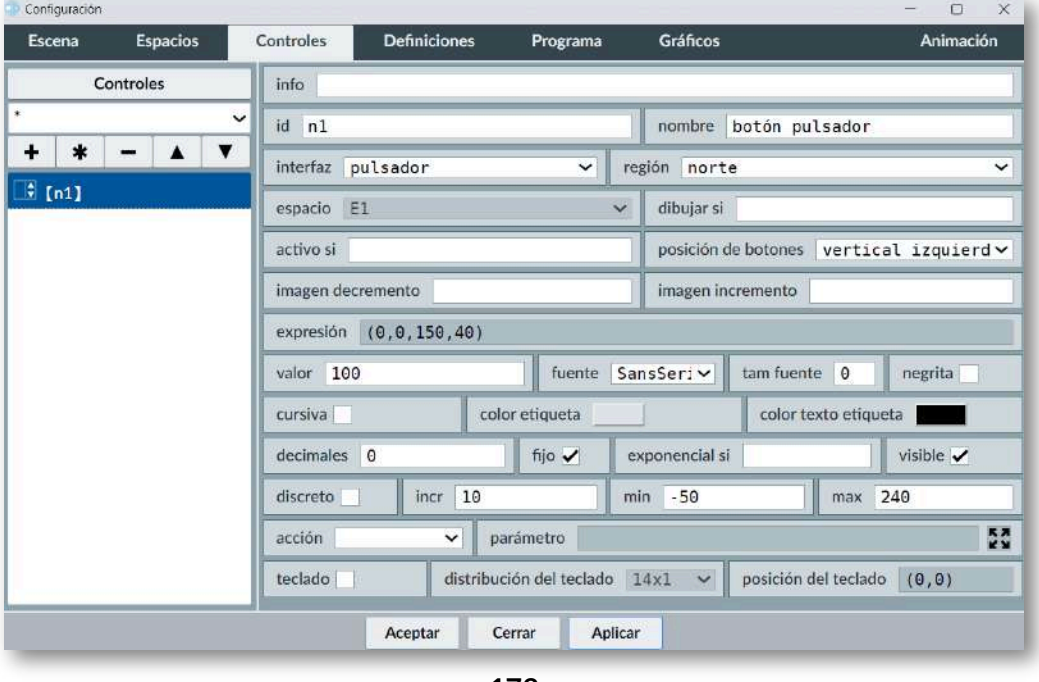
Además, puede incluir un campo de texto en el que el usuario puede escribir directamente un valor o una expresión para modificar su valor.

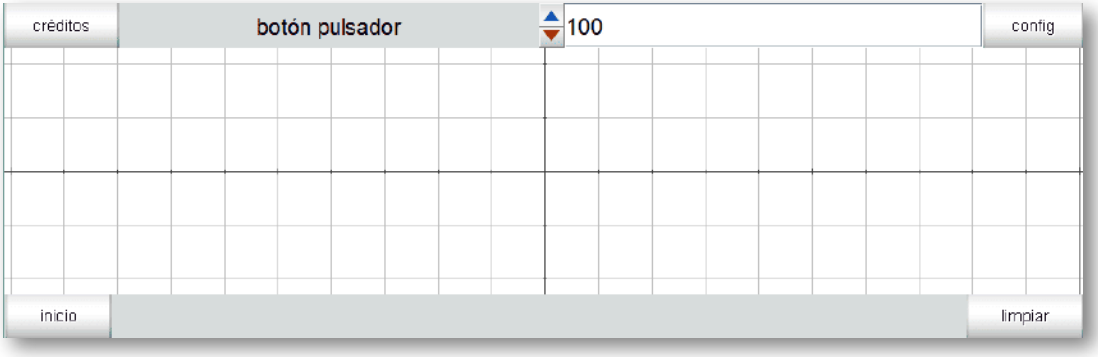
La mayoría de sus opciones coinciden con los elementos comunes a los controles; a continuación, se describen los parámetros específicos del pulsador:

- **posición de botones.** Menú que determina cómo se distribuyen los botones de decremento e incremento respecto al campo de texto del pulsador. Las opciones disponibles son:
    - **vertical izquierda** (opción por defecto): los botones se sitúan a la izquierda del campo de texto, uno encima del otro (incremento arriba y decremento abajo).
    - **vertical derecha**: los botones se disponen verticalmente a la derecha del campo de texto.
    - **horizontal izquierda**: los botones se ubican horizontalmente (decremento a la izquierda e incremento a la derecha), situados a la izquierda del campo de texto.
    - **horizontal derecha**: los botones se disponen horizontalmente a la derecha del campo de texto.
    - **horizontal extremos**: los botones se sitúan en los extremos y el campo de texto queda entre ambos.

- **imagen decremento.** Campo de texto en el que puede indicarse la ruta de una imagen que sustituirá el botón de decremento. La ruta debe ser relativa a la ubicación del archivo HTML de la escena. La imagen solo será visible una vez guardado y recargado el archivo, y se ajustará automáticamente al tamaño del botón.
- **imagen incremento.** Funciona de igual forma que **imagen decremento**, pero la imagen indicada se asocia al botón de incremento del pulsador.

Mostramos un ejemplo de un control de tipo **pulsador** con **id** n1, situado en la región **norte**, con valor inicial 100 y sin decimales. El incremento (**incr**) es de 10 unidades y el control está configurado como **discreto**, por lo que variará de 10 en 10. Su valor no podrá ser menor que -50 ni mayor que 240. El nombre que se mostrará en el interactivo será "botón pulsador". Además, se dibujará siempre (**dibujar si** = 1), estará siempre activo (**activo si** = 1) y no tendrá ninguna acción asociada.





A continuación, mostramos la escena interactiva con el control que hemos introducido en primer lugar, activado y después desactivado.





No se ha asociado ninguna acción al pulsador, aunque podría haberse configurado para que, al activarlo, se produjera algún evento, como abrir una página web o reiniciar la escena.

Incluimos a continuación una escena que incorpora controles de tipo pulsador con distintos estilos y diferentes disposiciones de las flechas de incremento y decremento. Además, se emplean los valores de estos controles para realizar dos operaciones matemáticas: en primer lugar, la suma de los valores obtenidos y, en segundo lugar, su producto.



3.6 Control tipo Campo de texto

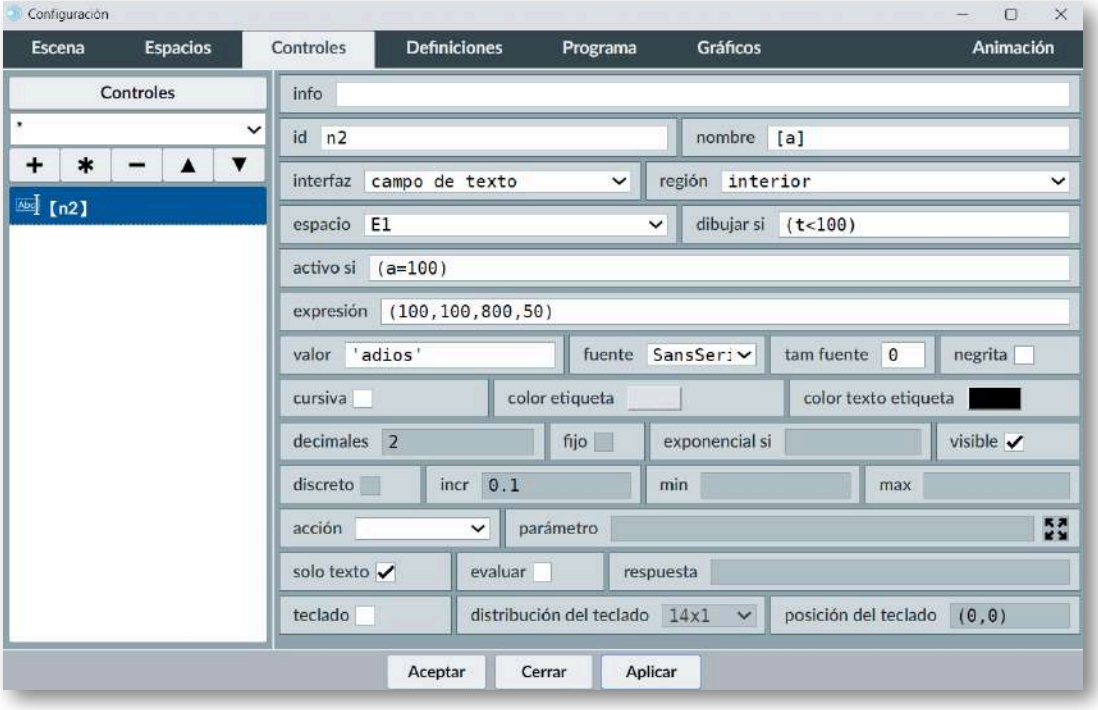
Este tipo de control numérico consiste en un campo donde el usuario puede introducir texto o números.

solo texto. Es un parámetro específico de este tipo de controles. Consiste en una casilla de verificación que indica que el contenido del campo se interpretará preferentemente como una cadena de texto y no como un número. Su activación cambia el comportamiento del control. Por ejemplo, si está marcada y el valor introducido es 123, al sumarle 3 el resultado será 1233, ya que se concatenan ambas cadenas. En cambio, si no está marcada, la operación dará como resultado 126.

Descartes incorpora cierto reconocimiento automático: si el campo está configurado como solo texto y se multiplica 123 por 3, el sistema interpretará correctamente el valor como numérico y devolverá 369 en lugar de producir un error. Cuando esta opción está activada, se deshabilitan los parámetros **decimales**, **fijo**, **exponencial si**, **discreto**, **incr**, **min** y **max**, ya que solo tienen sentido cuando el contenido del campo se trata como valor numérico.

Los campos de texto realizan una evaluación automática de expresiones siempre que la casilla **solo texto** no esté marcada. Este comportamiento es compartido con el control de tipo **Pulsador** cuando el valor del mismo es visible, permitiendo que las expresiones introducidas se interpreten y calculen automáticamente.

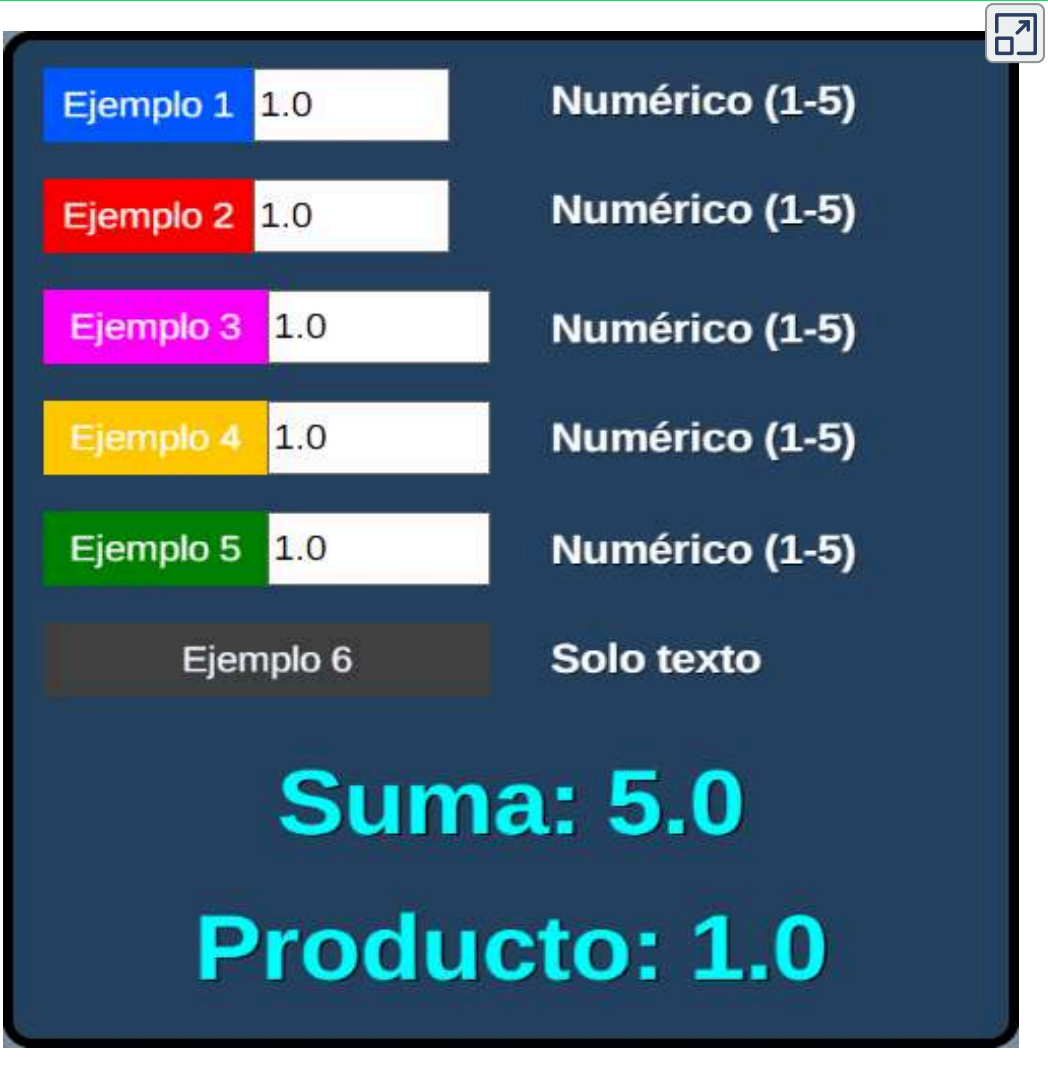
Presentamos un control de tipo campo de texto, situado en el interior en la posición (100,100) con un ancho de 800 y un alto de 50, con id **n2**. El nombre visible del control se mostrará según el valor de la variable **[a]**, que en este caso es "botón campo de texto". El control se dibujará únicamente cuando la variable **t** sea menor que 100 y estará activo solo si **a** vale


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
[n2]
info
id n2
nombre [a]
interfaz campo de texto
región interior
espacio E1
dibujar si (t<100)
activo si (a=100)
expresión (100,100,800,50)
valor 'adios'
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 2
fijo
exponencial si
visible
discreto
incr 0.1
min
max
acción
parámetro
solo texto
evaluar
respuesta
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar


créditos
config
botón de campo de texto
adios
inicio
limpiar


créditos
campo de texto
adios
inicio
limpiar

Mostramos una escena que incorpora controles de tipo campo de texto con distintos estilos. Los cinco primeros presentan un comportamiento numérico y sus valores están acotados. El sexto, en cambio, tiene activado el formato **solo texto**. Asimismo, se utilizan los valores de los controles numéricos para realizar dos operaciones matemáticas: en primer lugar, la suma de los valores introducidos y, en segundo lugar, su producto. El valor del último control no interviene en estas operaciones al no tratarse de un campo numérico.



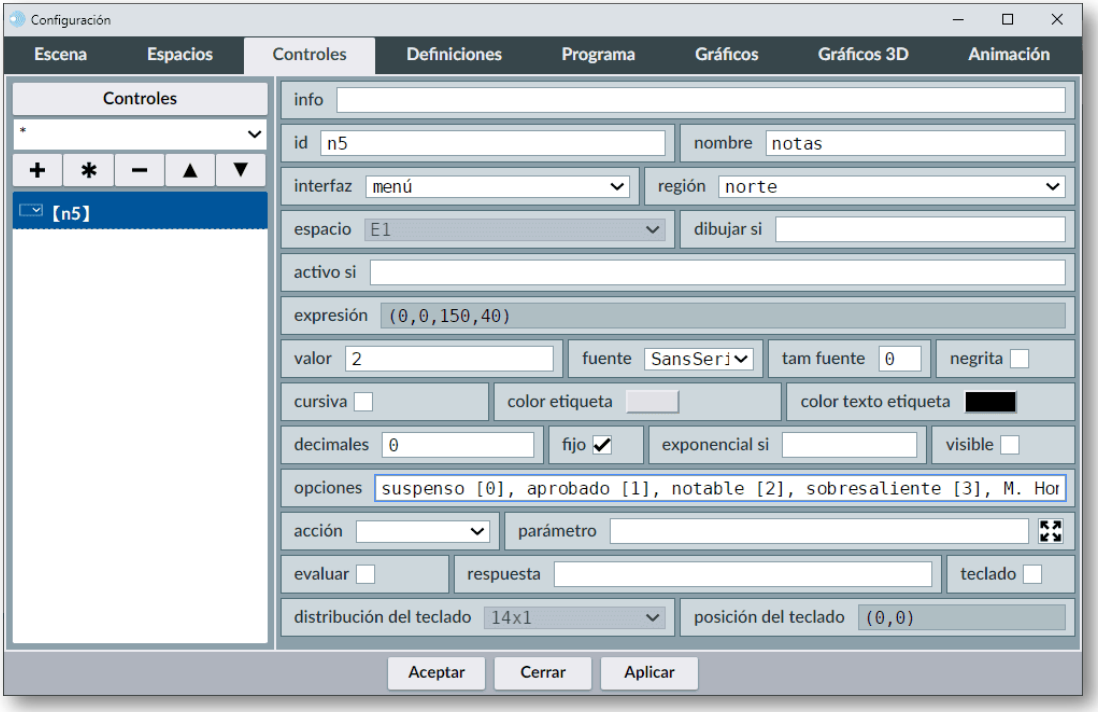
3.7 Control tipo Menú

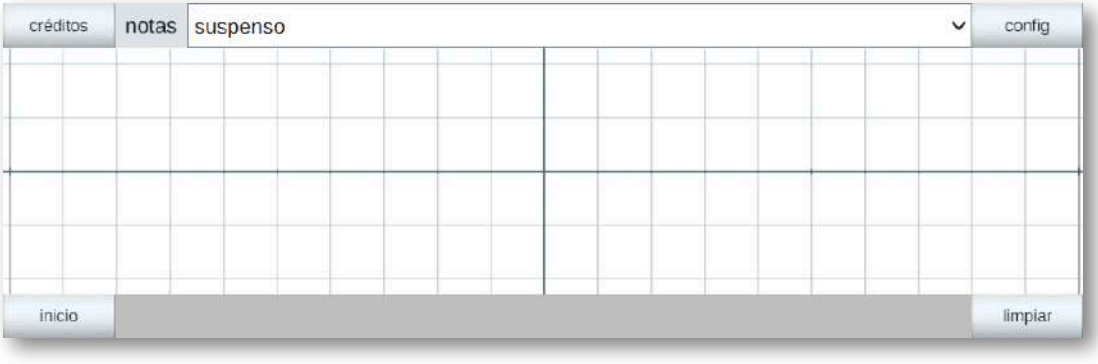
Este tipo de control numérico adopta la forma de un menú desplegable. Su principal utilidad es ofrecer al usuario la posibilidad de seleccionar entre distintas opciones dentro del interactivo. Muchos de sus parámetros son comunes al resto de controles y pueden consultarse en el apartado dedicado a los elementos comunes a los controles.

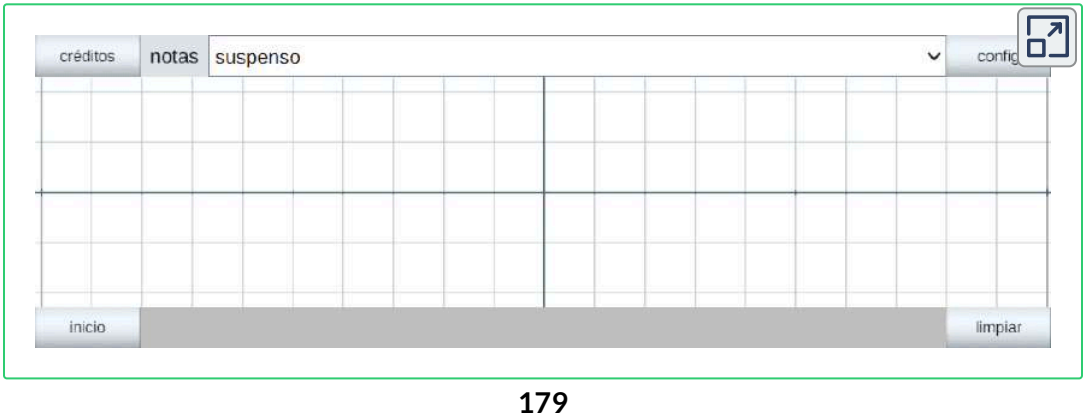
Los controles específicos del tipo **menú** son los siguientes:

- **valor.** Campo de texto que indica la opción seleccionada al iniciar el interactivo. Debe ser un número entero igual o mayor que 0. Cada opción del menú tiene asociado un valor numérico según su posición: la primera opción vale 0, la segunda 1, la tercera 2, y así sucesivamente. Por ejemplo, en un menú con las opciones "sumar", "restar", "multiplicar" y "dividir", si el valor inicial es 2, al comenzar aparecerá seleccionada la opción "multiplicar".
- **opciones.** Campo de texto donde se introducen las distintas opciones del menú, separadas por comas. Si se desea que el menú aparezca inicialmente sin ninguna opción visible seleccionada, puede escribirse un espacio antes de la primera coma y, a continuación, el resto de opciones. Cada opción determina automáticamente el valor numérico asociado al menú según su orden.

Presentamos una escena que incorpora un control de tipo **menú** con identificador **n5**, situado en la zona norte. Su valor inicial es 2 (sin decimales), por lo que al iniciarse quedará seleccionada la opción correspondiente a *notable*. Las opciones disponibles serán: suspenso [0], aprobado [1], notable [2], sobresaliente [3] y M. Honor [4]. El valor numérico asociado al control no será visible. El nombre mostrado será **notas**. El control se dibujará siempre, permanecerá activo en todo momento y no tendrá ninguna acción asociada.

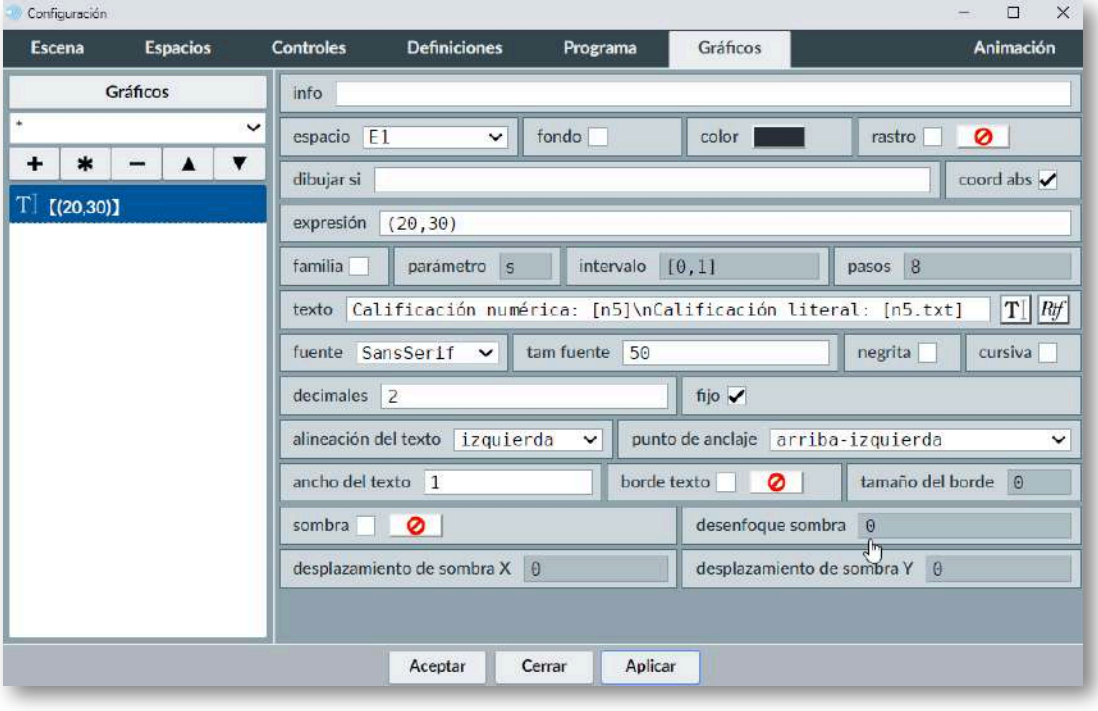

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Gráficos 3D
Animación
Controles
【n5】
info
id n5
nombre notas
interfaz menú
región norte
espacio E1
dibujar si
activo si
expresión (0,0,150,40)
valor 2
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 0
fijo
exponencial si
visible
opciones suspenso [0], aprobado [1], notable [2], sobresaliente [3], M. Hor
acción
parámetro
evaluar
respuesta
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar

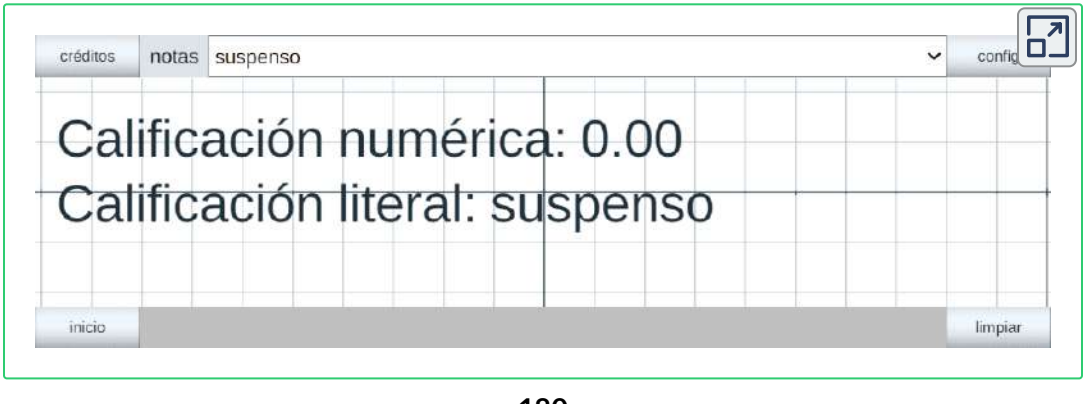

créditos
notas
suspenso
config
inicio
limpiar


créditos
notas
suspenso
config
inicio
limpiar

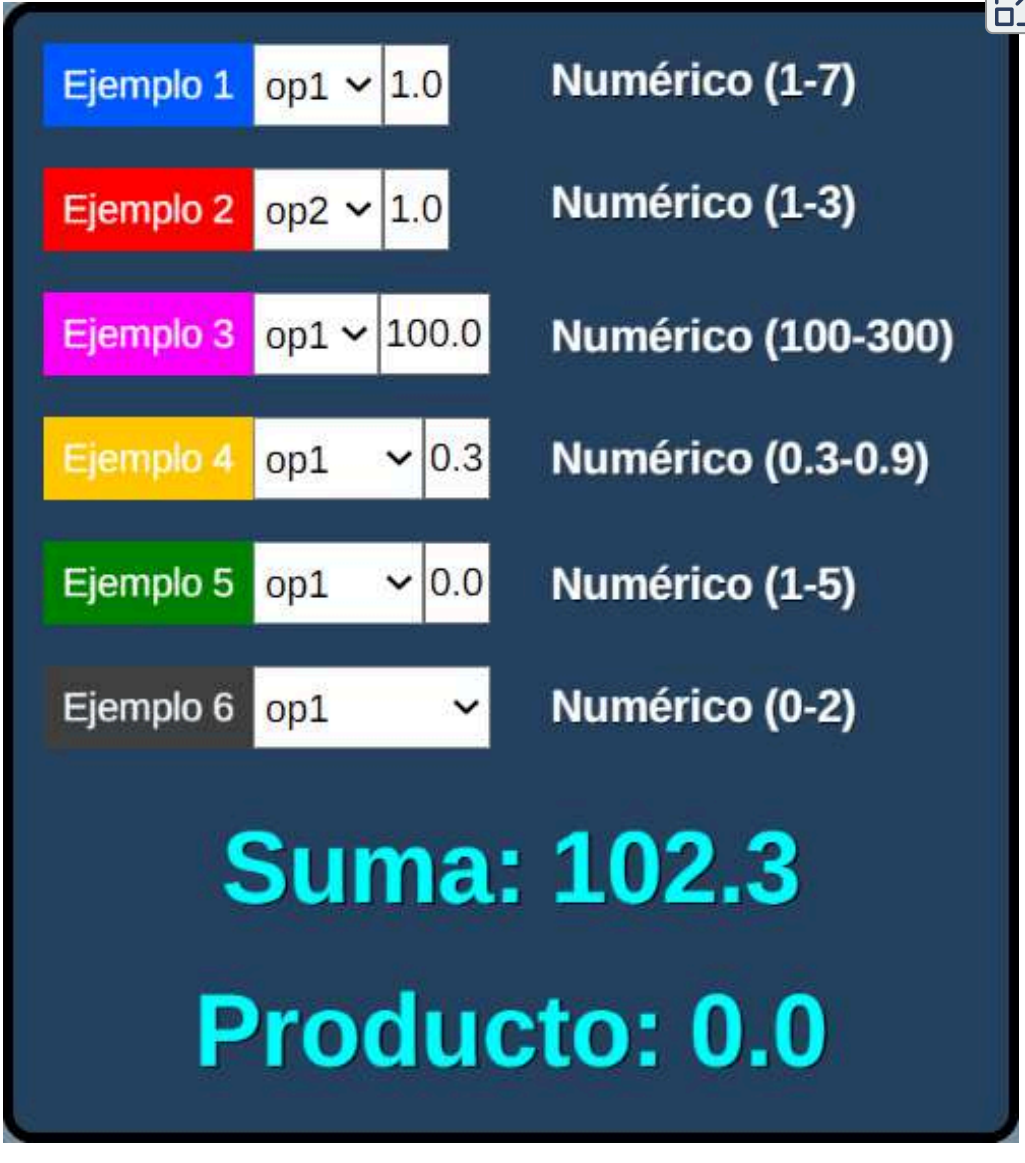
Los controles de tipo menú cuentan con la variable **.txt**, que permite recuperar el texto de la opción seleccionada.

En el ejemplo que estamos utilizando, si queremos obtener el valor numérico del control tipo menú con identificador **n5**, bastará con consultar la variable **n5**. En cambio, si deseamos obtener el texto literal de la opción seleccionada, deberemos utilizar la variable **n5.txt**.





Incluimos a continuación una escena que incorpora controles de tipo menú con distintos estilos. Cada opción textual lleva asociado un valor numérico: por defecto, la primera opción toma el valor 0, la segunda el 1 y así sucesivamente. No obstante, también es posible asignar manualmente un valor numérico concreto a cada opción.



3.8 Control tipo Barra

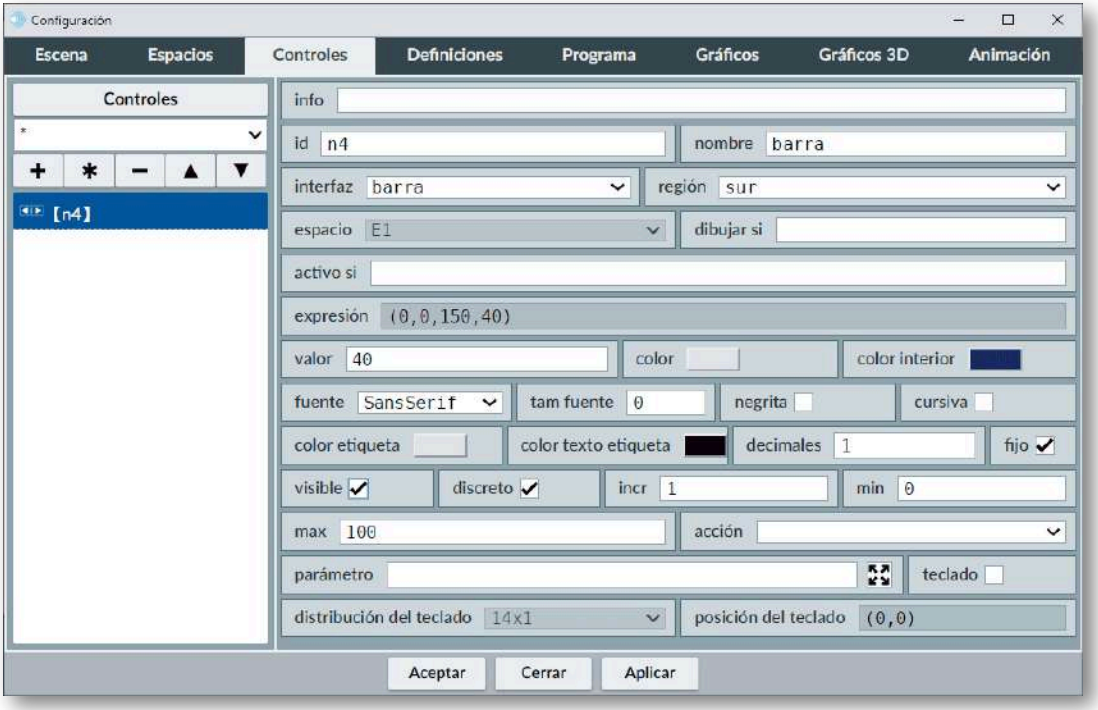
Este tipo de control numérico consiste en una barra de desplazamiento horizontal que modifica su valor al mover el cursor. Resulta especialmente útil para recorrer rápidamente un rango amplio de valores, aunque también permite ajustes más precisos cuando se requiere mayor control. Además, la barra puede incluir un campo editable que muestra su valor actual.

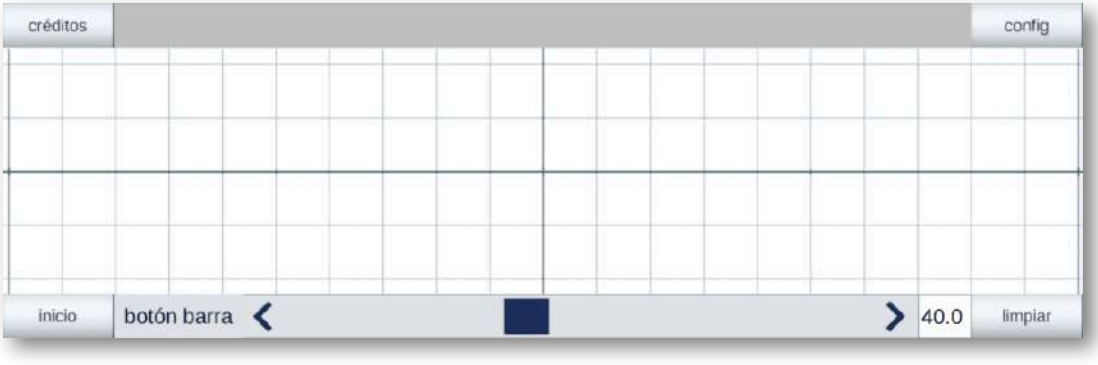
Si el alto de la barra es mayor que su ancho, el control se dispondrá con orientación vertical en lugar de horizontal y viceversa.

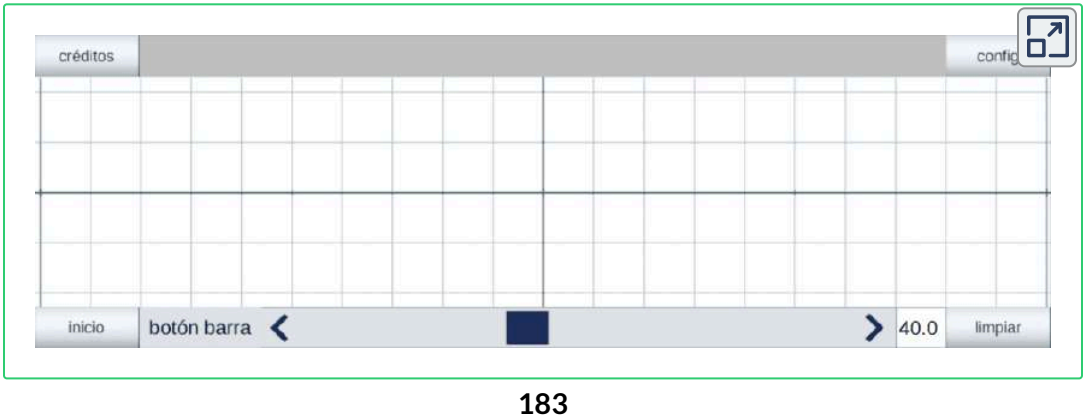
Además de los parámetros comunes, los controles propios del tipo **barra** son los siguientes:

- **color:** es un botón que abre la herramienta de selección de colores para definir el color de fondo de la barra de desplazamiento.
- **color interior:** es un botón que abre la herramienta de selección de colores para establecer el color del manipulador de la barra y de las flechas de los extremos. Si se asigna un color mediante la pestaña **Gradiente**, los extremos se pintan según los colores del gradiente y el manipulador cambia de color al acercarse a cada extremo de la barra.

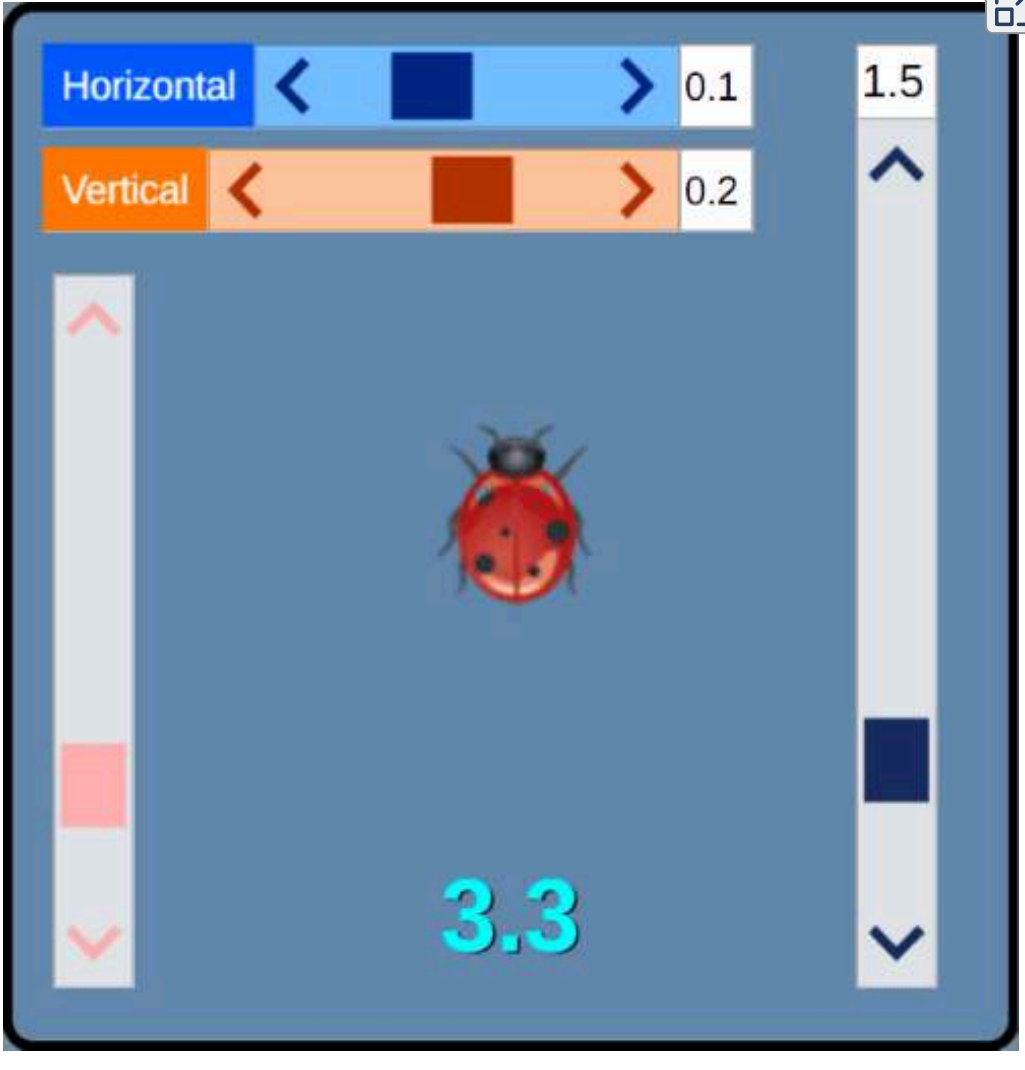
Mostramos un ejemplo de un control de tipo barra de desplazamiento con identificador **n4**, ubicado en la zona sur. Su valor inicial es 40 con un decimal, incrementándose de 1 en 1 de forma discreta, con un límite mínimo de 0 y máximo de 100. El nombre visible del control será "botón barra" y su valor se mostrará en pantalla. Este control se dibujará siempre, permanecerá activo en todo momento y no tiene ninguna acción asociada. La variable **n4** tomará los valores del control homónimo y viceversa.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Gráficos 3D
Animación
Controles
[n4]
info
id n4
nombre barra
interfaz barra
región sur
espacio E1
dibujar si
activo si
expresión (0,0,150,40)
valor 40
color
color interior
fuente SansSerif
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 1
fijo
visible
discreto
incr 1
min 0
max 100
acción
parámetro
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar


créditos
config
inicio
botón barra
40.0
limpiar


créditos
inicio
botón barra
40.0
limpiar

En la siguiente escena se incluyen diferentes controles de tipo barra de deslizamiento con variados estilos, dimensiones y configuraciones. Se presentan tanto en formato horizontal como vertical, lo que permite apreciar sus distintas posibilidades de disposición en la escena. Asimismo, se han incorporado controles con visor de valor visible y otros que no lo incluyen, para comparar su comportamiento y apariencia.

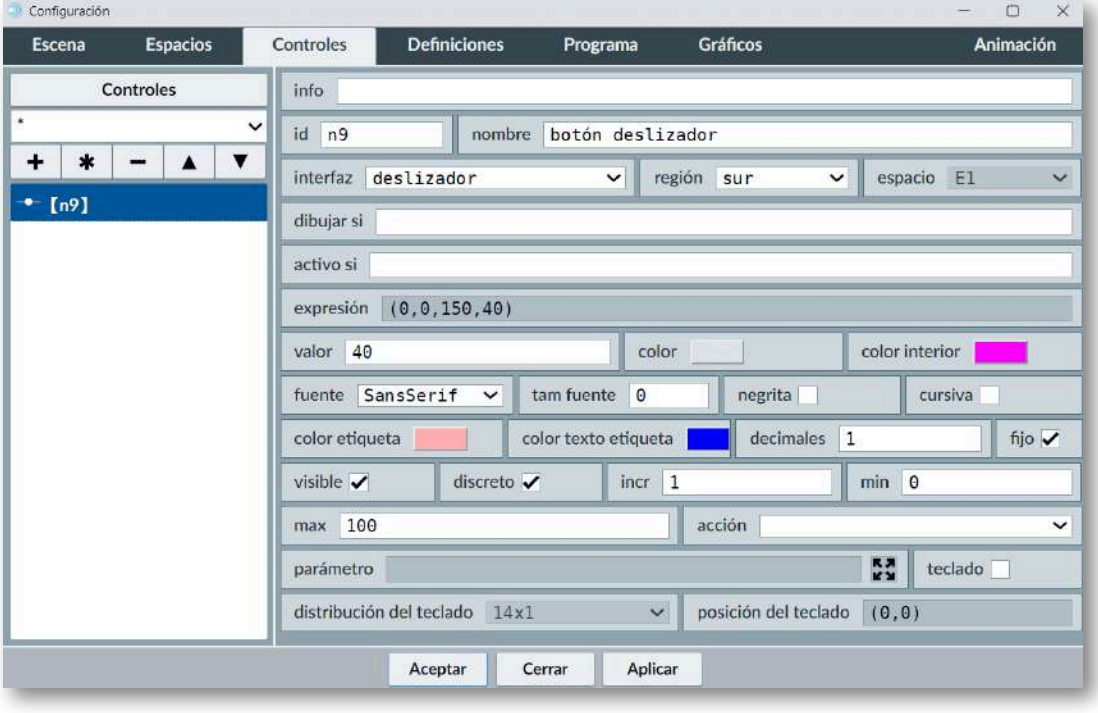


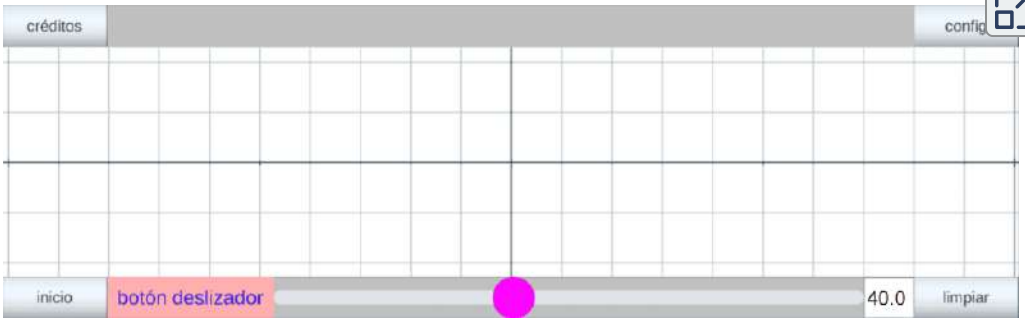
3.9 Control tipo Deslizador

Los deslizadores (o *sliders*) son controles con un funcionamiento similar al de la barra, pero en este caso el elemento móvil es un círculo que se desplaza a lo largo de un eje. El identificador del deslizador actúa como una variable cuyo valor depende de la posición del círculo y de los límites establecidos mediante los parámetros **min** y **max**.

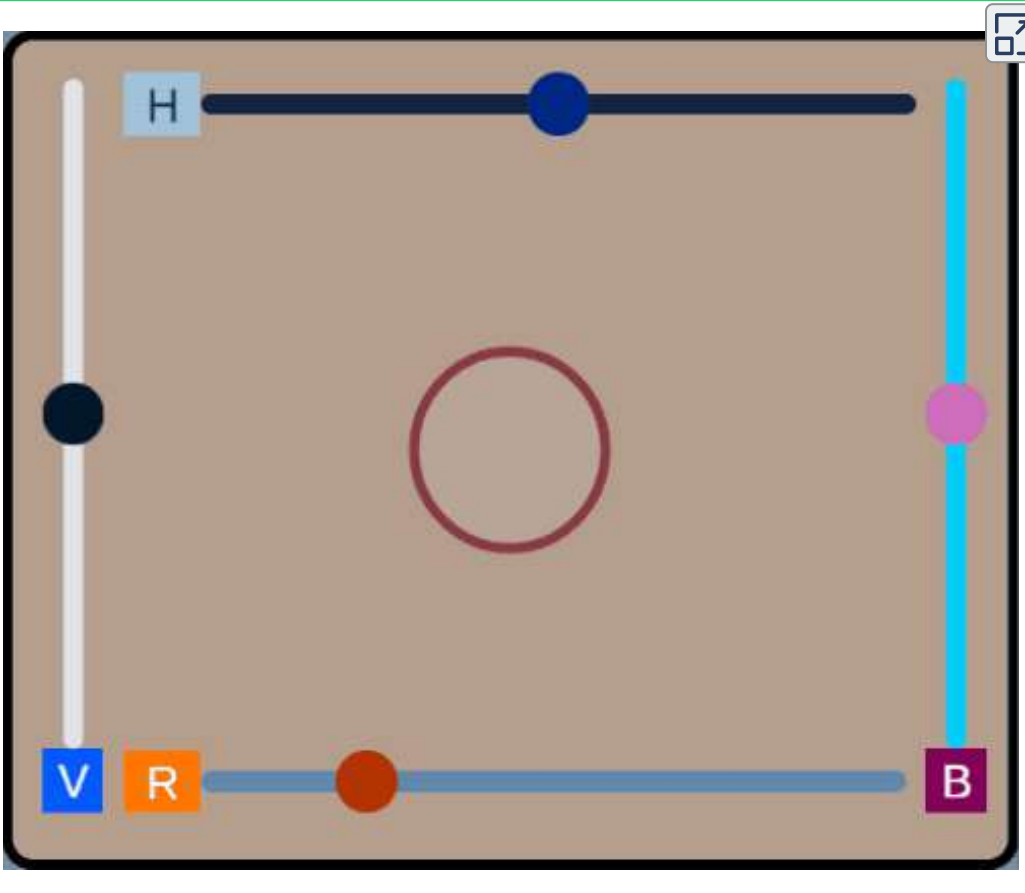
Si el ancho del deslizador es menor que su alto, el control se mostrará con orientación vertical. Los parámetros del control tipo deslizador coinciden con los del control tipo barra.

Incluimos un ejemplo de este tipo de control con identificador **n9** y nombre "botón deslizador", situado en la zona sur. Su valor inicial es 40 con un decimal, varía de 1 en 1 de forma discreta y está acotado entre 0 y 100.





En la siguiente escena se incluyen diferentes controles de tipo deslizador con variados estilos, dimensiones y configuraciones. Se presentan tanto en formato horizontal como vertical.



3.10 Control tipo Botón

Este tipo de control numérico es el más sencillo. Consiste en un botón que, al pulsarse, ejecuta una acción determinada. La mayoría de sus parámetros son comunes al resto de controles y pueden consultarse en el apartado de elementos comunes a los controles.

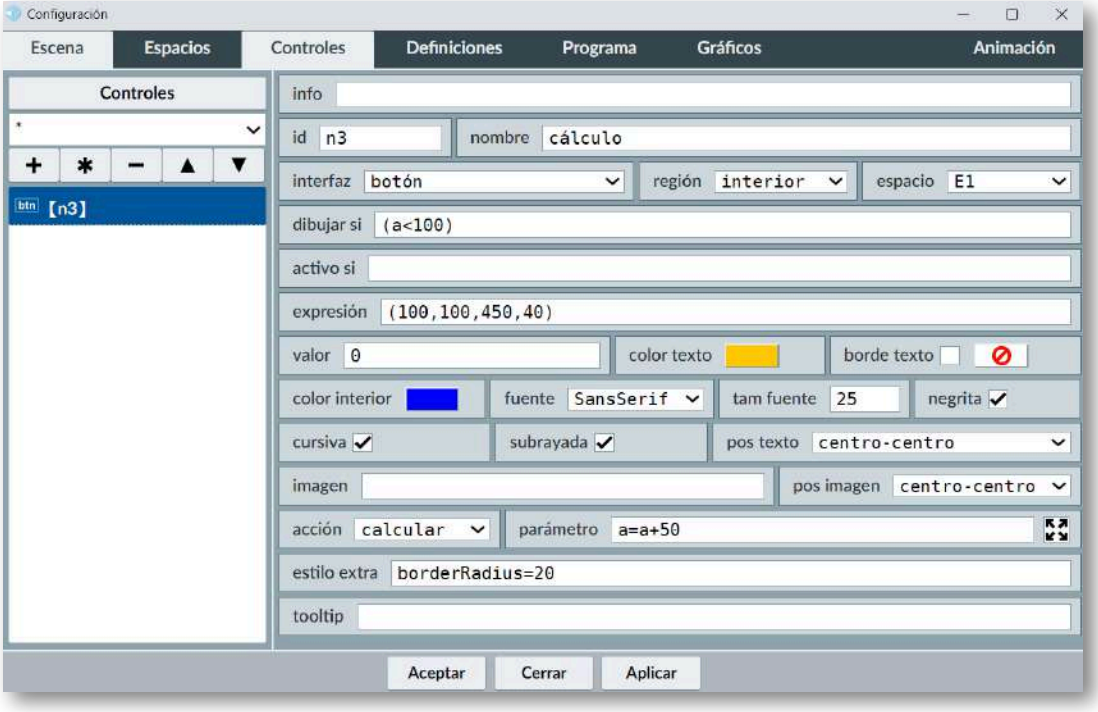
Los controles específicos del tipo **botón** son los siguientes:

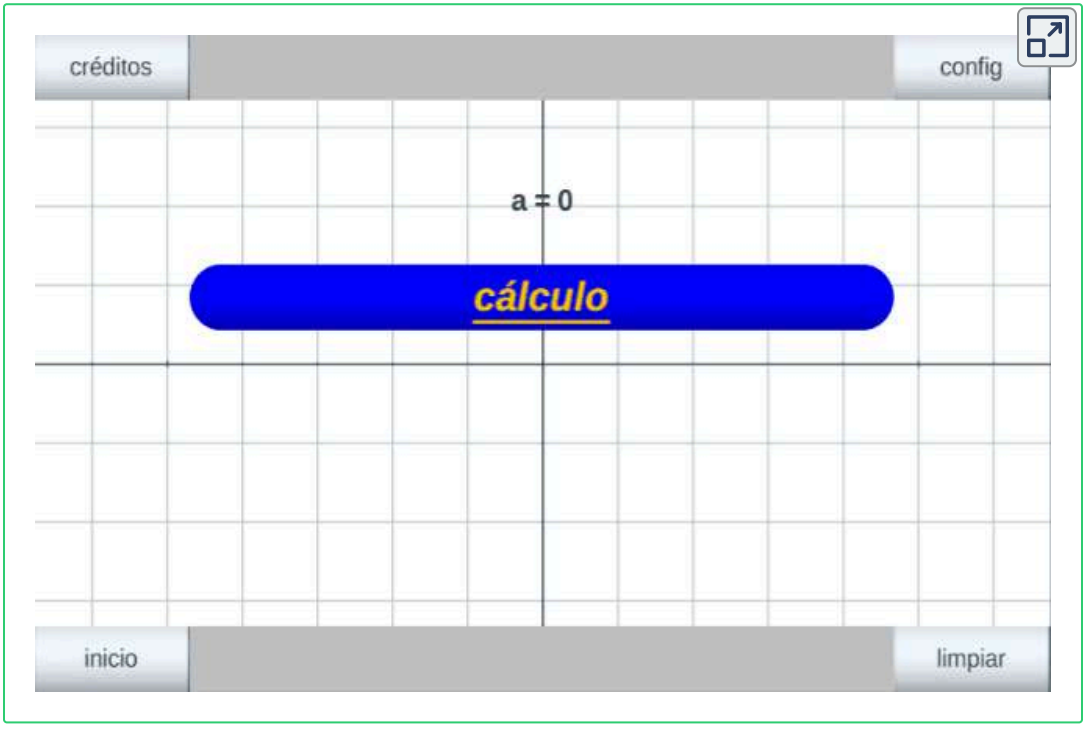
- **color texto:** abre la herramienta de selección de colores para definir el color del texto del botón.
- **borde texto:** casilla que, al activarse, permite elegir el color del borde del texto mediante la herramienta de colores.
- **color interior:** permite seleccionar el color de fondo del botón. Si se utiliza una imagen con transparencia como fondo, se recomienda ajustar aquí la transparencia al máximo.
- **fuente:** menú para elegir el tipo de letra (SansSerif, Serif o Monospaced).
- **tam fuente:** tamaño de la fuente en puntos.
- **negrita, cursiva, subrayada:** casillas que aplican estos estilos al texto del botón.
- **pos texto:** determina la posición del texto dentro del botón (arriba, centro o abajo; izquierda, centro o derecha).
- **imagen:** ruta relativa a una imagen (.png o .jpg) que se usará como fondo del botón. Si existe un archivo con el mismo nombre y sufijo **_down**, se mostrará al pulsarlo; si existe con sufijo **_over**, se mostrará al pasar el cursor por encima.
- **pos imagen:** define la posición de la imagen dentro del botón cuando no ocupa toda su superficie.
- **tooltip:** texto de ayuda contextual que aparece al mantener el cursor sobre el botón durante unos segundos.

- **estilo extra:** permite definir estilos adicionales en una sola cadena de texto, separando cada propiedad con el símbolo **|** (por ejemplo: border=3| borderRadius=10| borderColor=ff0000). Entre los parámetros disponibles se encuentran:
    - **border:** grosor del borde en píxeles.
    - **borderRadius:** radio de las esquinas redondeadas.
    - **borderColor:** color del borde en formato hexadecimal.
    - **overColor:** color interior al pasar el cursor.
    - **downColor:** color interior al pulsar el botón.
    - **inactiveColor:** color interior cuando el botón está inactivo.
    - **font:** tipo de fuente (SansSerif, Serif o Monospaced).
    - **shadowTextColor:** color de la sombra del texto.
    - **shadowBoxColor:** color de la sombra exterior del botón.
    - **shadowInsetBoxColor:** color de la sombra interior.
    - **flat:** valor 0 para fondo con degradado (por defecto) o 1 para color plano.

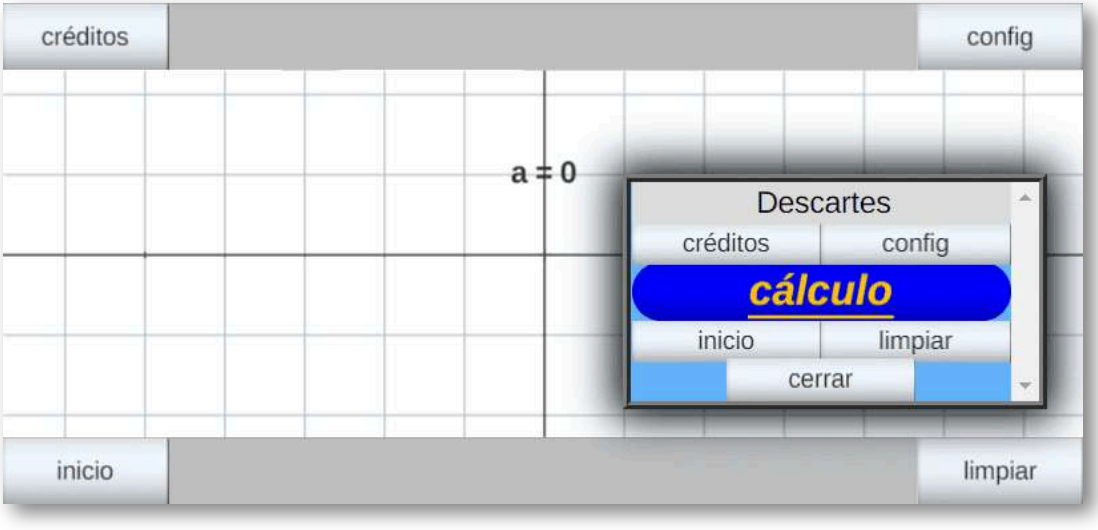
Presentamos un control de tipo **botón**, situado en el interior en la posición (300,70), con un ancho de 450 y un alto de 40, e identificador **n3**. El nombre visible será **calculo**. El fondo del botón será azul y el texto, en color naranja, aparecerá en negrita, cursiva y subrayado, con un tamaño de fuente de 25 puntos. Los bordes del botón son redondeados con una radio de borde de 20.

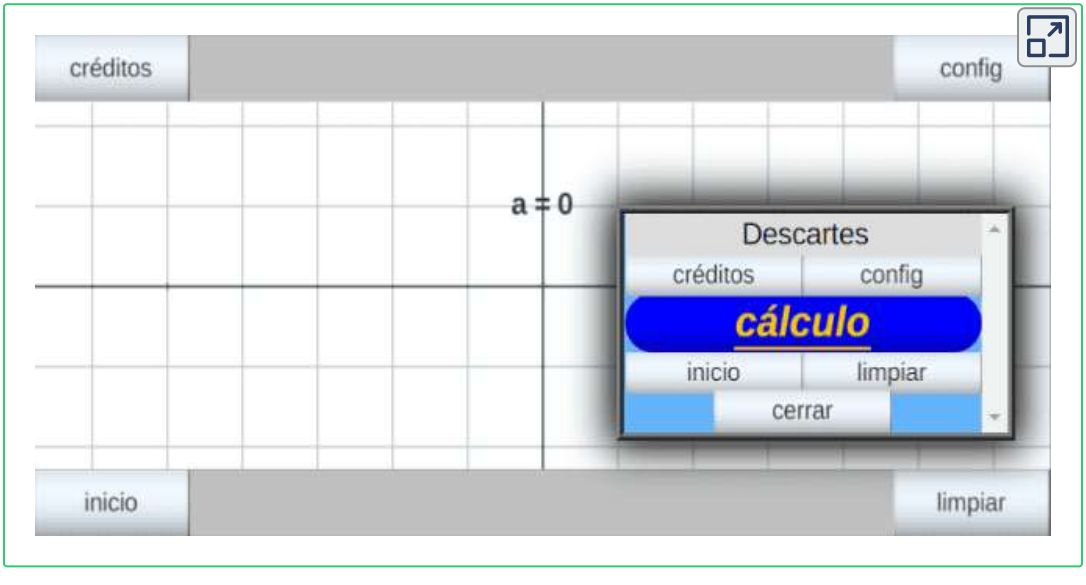
El botón se dibujará cuando la variable **a** sea menor que 100 y estará siempre activo. Tendrá asociada la acción **calcular**, de modo que cada vez que se pulse se ejecutará la asignación **a=a+50**. Tras pulsarlo dos veces, el botón dejará de mostrarse en la escena.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
btn 【n3】
info
id n3
nombre cálculo
interfaz botón
región interior
espacio E1
dibujar si (a<100)
activo si
expresión (100,100,450,40)
valor 0
color texto
borde texto
color interior
fuente SansSerif
tam fuente 25
negrita
cursiva
subrayada
pos texto centro-centro
imagen
pos imagen centro-centro
acción calcular
parámetro a=a+50
estilo extra borderRadius=20
tooltip
Aceptar
Cerrar
Aplicar

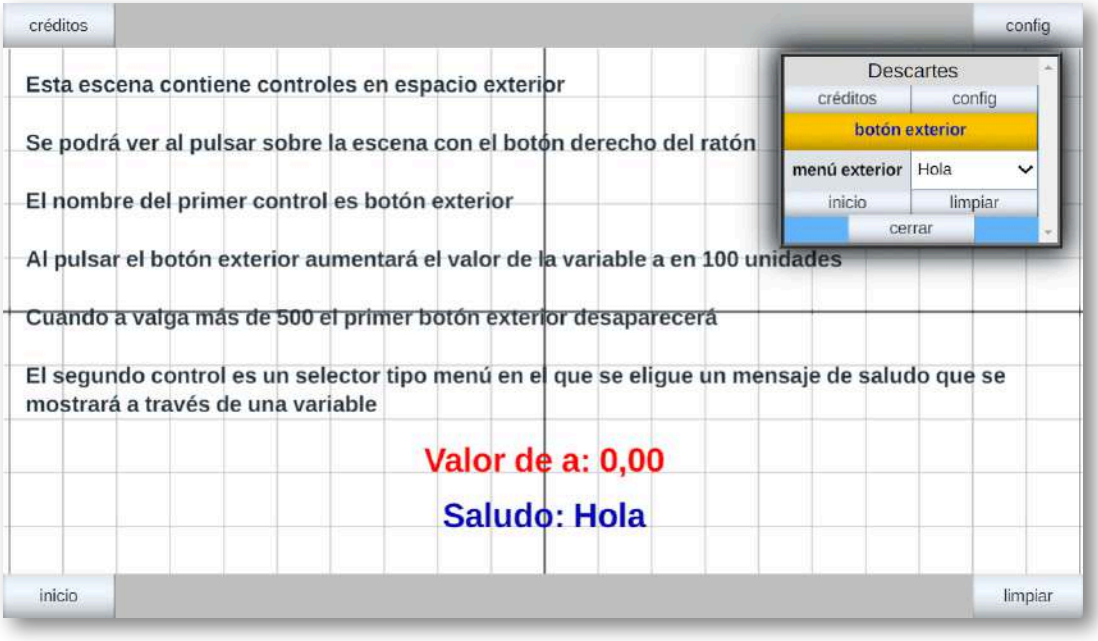

créditos
config
a = 0
cálculo
inicio
limpiar

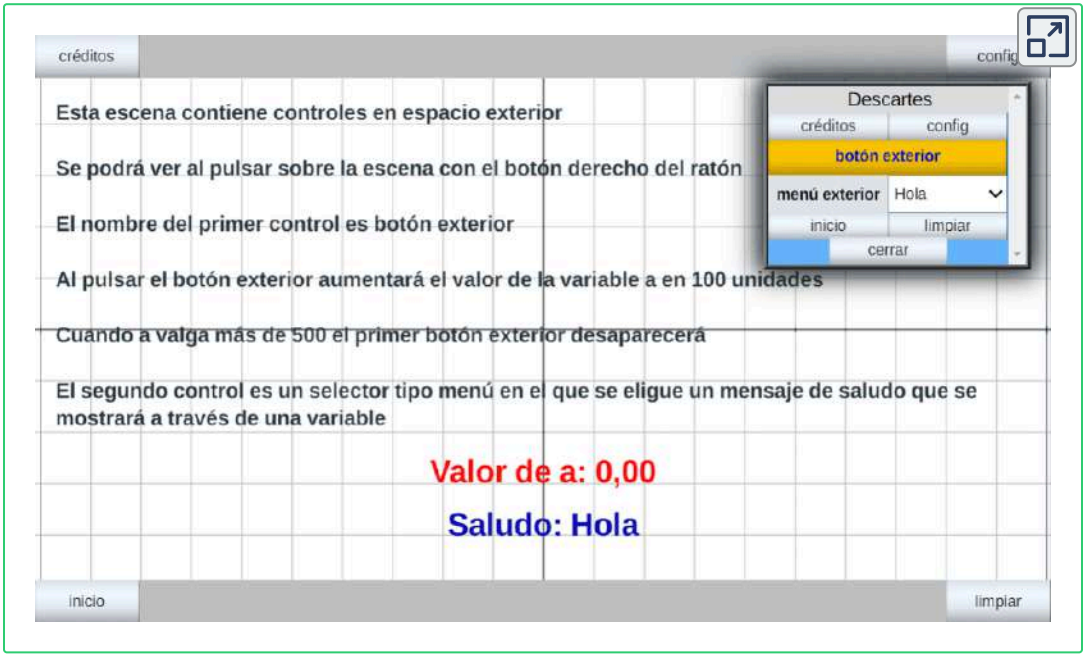
Recordamos que los controles también pueden situarse en la región exterior, siempre que esté activada en el selector *Escenas* la opción *mostrar región exterior*. Como ejemplo, incluiremos el control de la escena anterior en dicha región.





En la región exterior pueden incluirse varios controles y de distintos tipos; no obstante, no es posible incorporar controles gráficos, de texto, audio o vídeo en esta zona.





En las siguientes escenas se incluyen controles con variados estilos y configuraciones. En la segunda, además, se muestran distintas acciones asociadas a los mismos.


Botón con texto y fondo
Texto del botón
Texto del botón
Texto del botón
Botón con texto e imagen
Texto para botón
0
Botón sin fondo
Botón en estado normal


r
0,00
calcular
r=0.5
anim/pausa
r=1.0
Limpiar
r=1.5
Inicio
abrir Escena
abrirURL
Sonido

3.11 Control tipo Casilla de verificación

Las casillas de verificación, también conocidas como *checkbox*, son controles que pueden aparecer marcados o desmarcados. Resultan especialmente útiles en actividades de opción múltiple.

Existen dos formas de funcionamiento. La primera es la casilla de verificación propiamente dicha, que permite marcar varias opciones al mismo tiempo. También puede utilizarse de forma individual para indicar si se cumple o no una determinada condición.

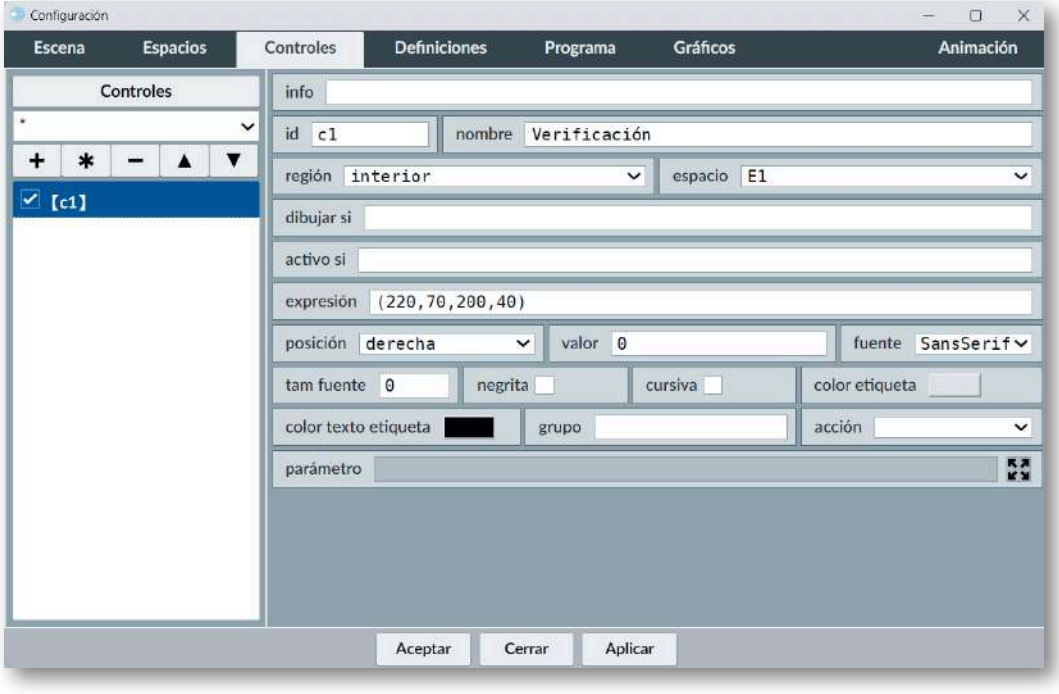
La segunda modalidad corresponde a los llamados *radio botones*. En este caso, solo puede seleccionarse una opción: al marcar una, las demás se desactivan automáticamente. Este comportamiento es apropiado cuando se requiere una única respuesta entre varias posibles.

Los parámetros propios de este tipo de control son los siguientes:

- **id:** identificador de la casilla; corresponde al nombre de la variable interna que almacena su valor.
- **valor:** valor inicial del identificador. En las casillas de verificación puede ser 0 (desmarcada) o 1 (marcada).
- **grupo:** campo de texto donde se indica el nombre del grupo al que pertenecen varias casillas cuando funcionan como *radio botones*. Al marcar una casilla del mismo grupo, las demás se desmarcan automáticamente.
- **posición:** menú que permite elegir si la casilla (el recuadro de selección) se sitúa a la derecha o a la izquierda del nombre del control.

Se incluye un control de tipo casilla de verificación, situado en la región interior en la posición y dimensiones (220,70, 200, 40), con identificador **c1**. El nombre visible del control será **Verificación**.

La casilla de verificación estará situada a la derecha de la etiqueta correspondiente al nombre del control. Además, se ha incorporado un texto que se muestra cuando la casilla está marcada y otro diferente que aparece cuando se encuentra desmarcada, de modo que el contenido visible cambia según su estado.



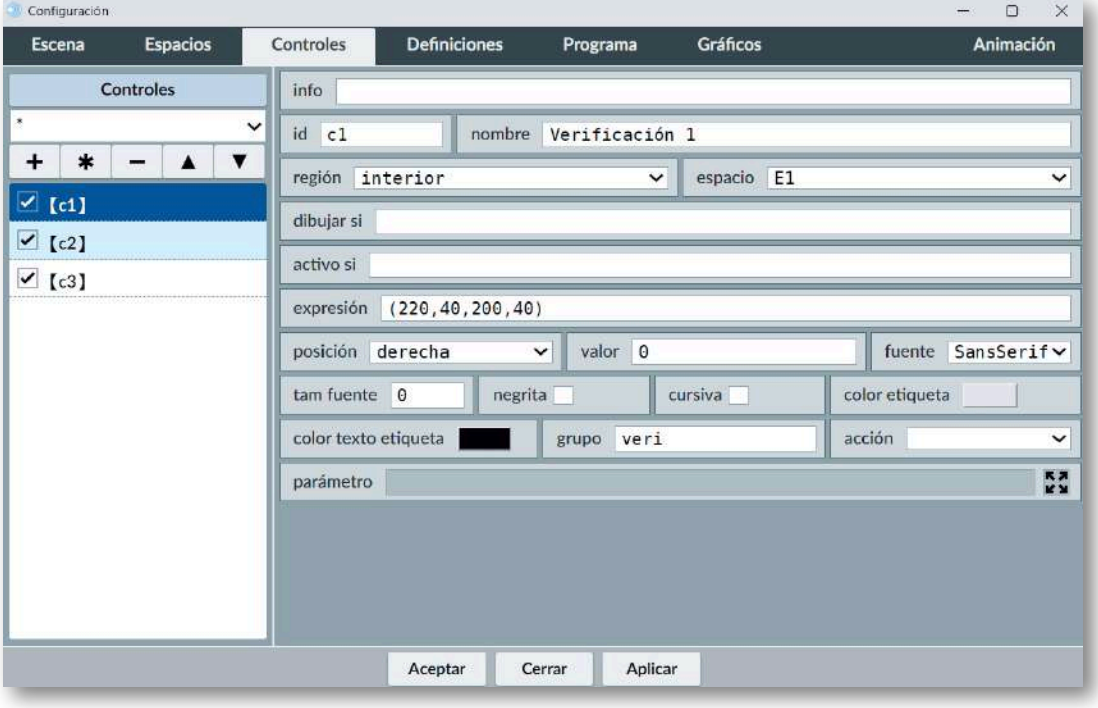


Se define un control de tipo *radio botón*, formado por varias casillas de verificación. Para ello, se sitúan tres controles con identificadores **c1**, **c2** y **c3** en la región interior, asignando a los tres el mismo nombre en el campo **grupo**; en este ejemplo, **veri**.

Al pertenecer al mismo grupo, las tres casillas actúan como un radio botón, de manera que solo una puede estar marcada a la vez. Cuando se selecciona una, las demás se desmarcan automáticamente.

Puede definirse que, al iniciarse la escena, alguna de las casillas aparezca marcada por defecto o bien que todas estén desmarcadas. En nuestro ejemplo hemos marcado inicialmente la primera casilla; para ello, se ha asignado el valor 1 a la variable asociada al control **c1**.

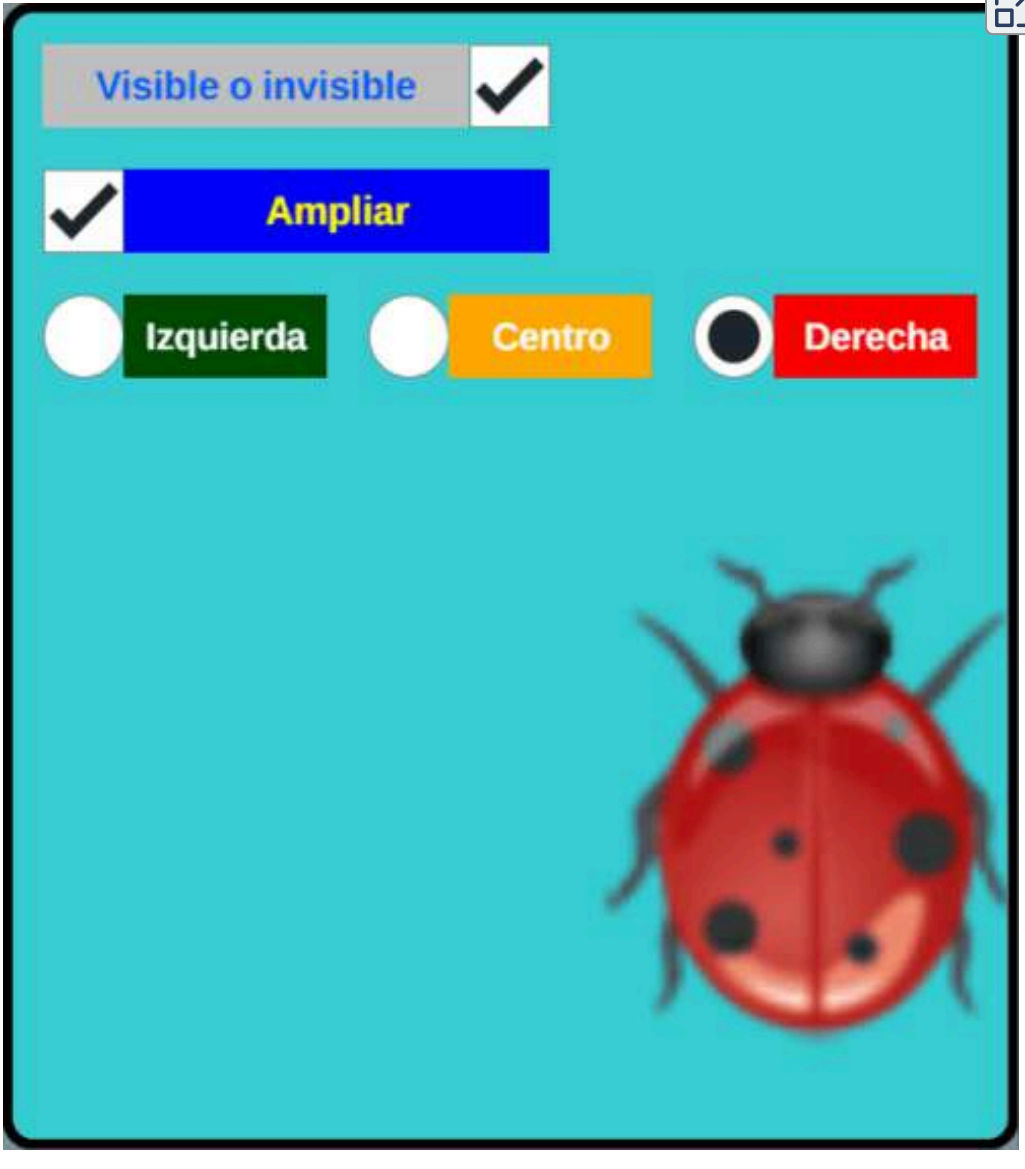
Si se desea disponer de un control que desmarque todas las casillas del radio botón, basta con asignar el valor 0 a cada una de las variables asociadas al grupo. En nuestro ejemplo, sería necesario establecer **c1=0**, **c2=0** y **c3=0**.




créditos
config
Verificación 1
Verificación 2
Verificación 3
Está marcada la casilla de verificación 1
inicio
limpiar


créditos
Limpiar
config
Verificación 1
Verificación 2
Verificación 3
Está marcada la casilla de verificación 2
inicio
limpiar

Incluimos una escena que contiene casillas de verificación independientes, así como un radio botón formado por casillas de verificación agrupadas. Además, se han aplicado distintos estilos a los controles y se han asociado acciones o efectos específicos a cada uno de ellos.



3.12 Control tipo Gráfico

Los controles gráficos, a diferencia de los controles numéricos, consisten en puntos que el usuario puede manipular directamente con el ratón o, en dispositivos móviles, arrastrar sobre la pantalla táctil.

Para agregar un control gráfico se pulsa el botón + en el selector **Controles**, lo que abre una ventana en la que se puede seleccionar la interfaz **gráfico**. Los identificadores de estos controles comienzan por defecto con la letra **g**. Son objetos que permiten modificar parámetros de la escena. Al crear un control gráfico se generan dos variables internas llamadas *id.x* e *id.y*, donde *id* es el identificador del control.

Por defecto, el control se asigna al primer espacio definido, aunque esto puede cambiarse posteriormente. Al añadirse, se mostrará un círculo cuya posición inicial (coordenadas relativas), color y tamaño pueden configurarse, así como las condiciones para que se dibuje o permanezca activo. Además, el círculo puede ser reemplazado por una imagen.

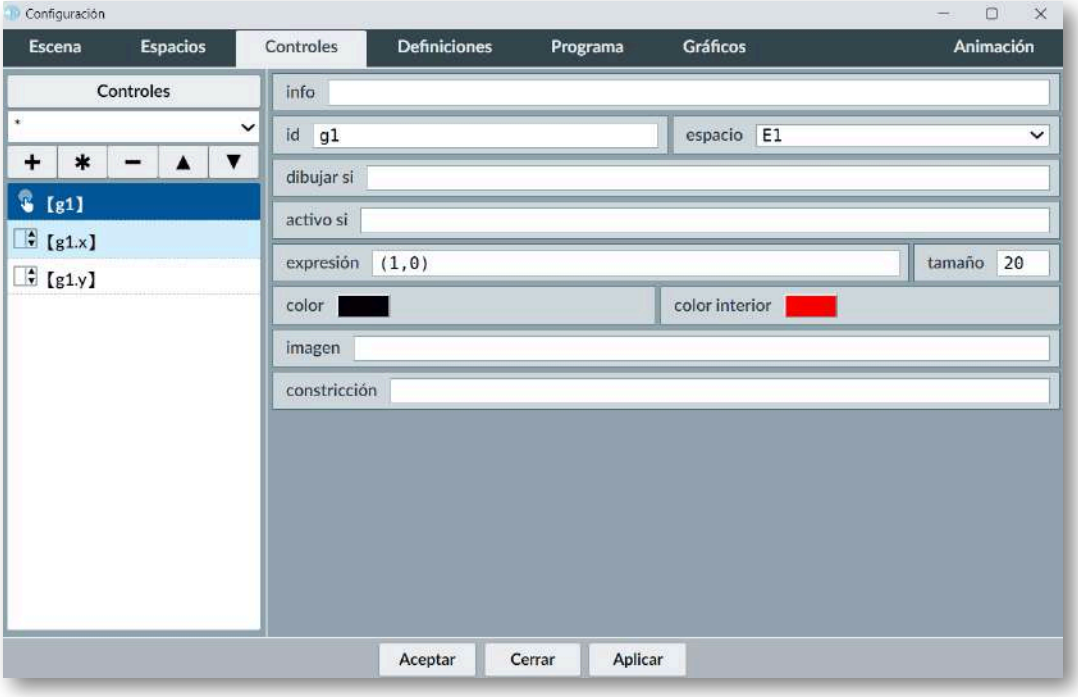
Los parámetros propios de los controles tipo gráfico son:

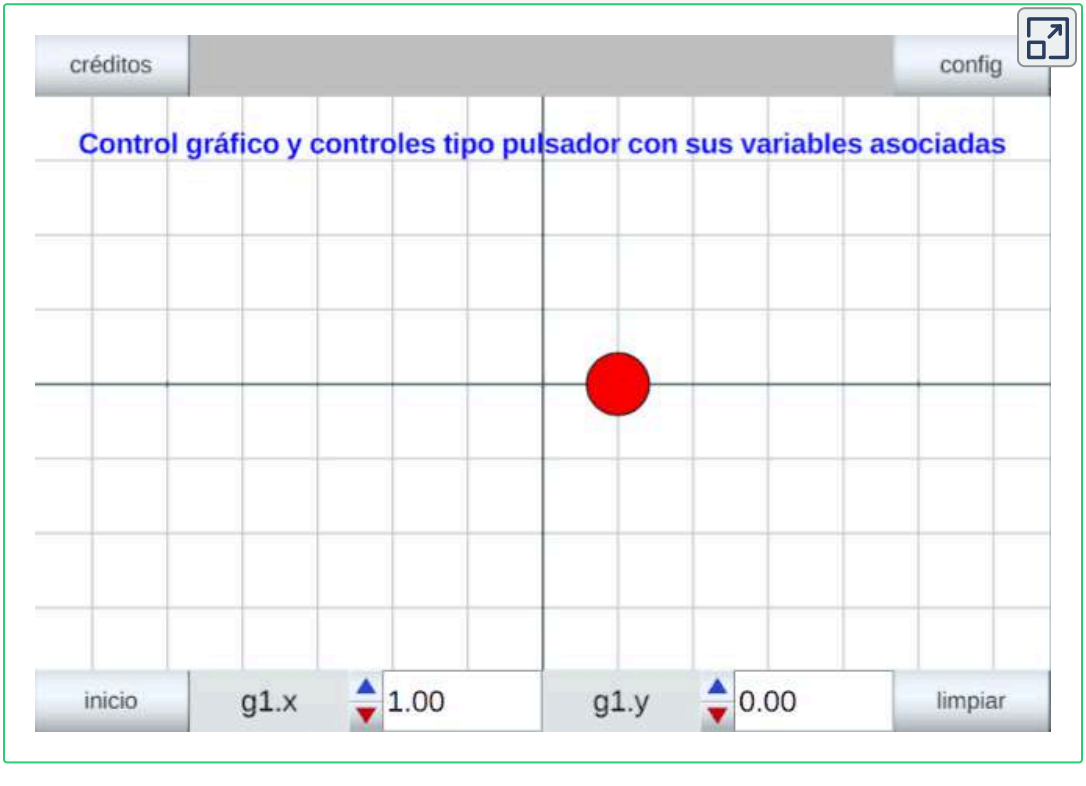
- **id:** campo de texto donde se agrega el identificador. Este permite extraer propiedades del control, como coordenadas de posición o si está siendo usado, sin ser una variable que almacene un valor como los controles numéricos.
- **espacio:** menú donde se escoge en qué espacio aparecerá el control gráfico como punto interactivo móvil.
- **expresión:** campo de texto donde se introduce la coordenada inicial del control gráfico. Por defecto, se encuentra en el origen. La posición de un control gráfico siempre se expresa en coordenadas relativas.

- **tamaño:** campo de texto para definir el radio en píxeles del círculo que representa al control gráfico.
- **constricción:** campo de texto donde se introduce una ecuación que limita el movimiento del control dentro de un área o curva determinada. Por ejemplo, $x^2 + y^2 = 4$ restringirá el control a la circunferencia de radio 2, y $x^2 + y^2 \leq 4$ permitirá que se mueva dentro de la circunferencia, pero sin salir de ella. Si la posición inicial no cumple la constricción, el editor ajustará el control al área permitida al moverlo.
- **color:** botón que abre la herramienta de control de colores para definir el color del contorno del control gráfico.
- **color interior:** botón que abre la herramienta de control de colores para definir el color del centro del círculo que representa el control gráfico.
- **imagen:** campo de texto con la ruta relativa a un archivo de imagen (jpg o png) que reemplaza al círculo y puede arrastrarse como el control gráfico.

Los controles gráficos se pueden mover de dos maneras: arrastrándolos directamente con el ratón o modificando sus coordenadas asociadas. Cuando un control gráfico tiene el foco, se muestra una circunferencia blanca o negra en su interior para indicar que está activo. Para que un control adquiera el foco, es necesario seleccionarlo previamente con el ratón.

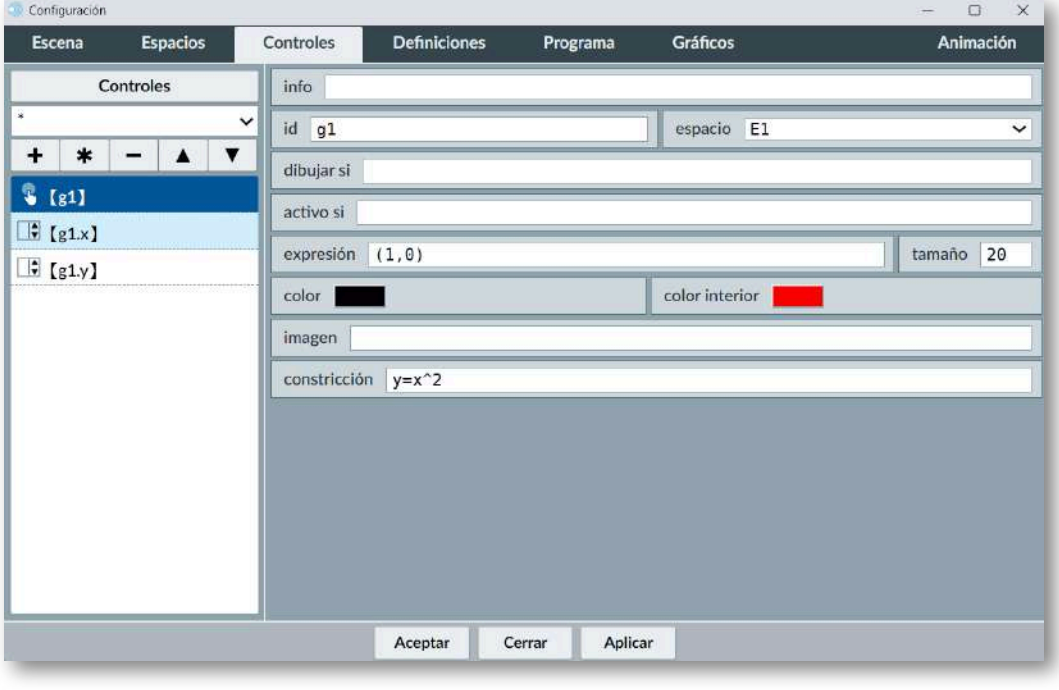
En la siguiente escena se ha incluido un control gráfico con color interior rojo, borde negro y tamaño 20, ubicado inicialmente en la posición (1,0). Su identificador es **g1** y sus variables asociadas **g1.x** y **g1.y** están vinculadas a controles tipo pulsador. Este control puede moverse tanto arrastrándolo con el ratón como utilizando los pulsadores situados en la región sur. Al no haberse definido ninguna constricción, el movimiento del control es libre por toda la escena, incluso más allá de los límites visibles.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
[g1]
[g1.x]
[g1.y]
info
id g1
espacio E1
dibujar si
activo si
expresión (1,0)
tamaño 20
color
color interior
imagen
constricción
Aceptar
Cerrar
Aplicar


créditos
config
Control gráfico y controles tipo pulsador con sus variables asociadas
inicio
g1.x
1.00
g1.y
0.00
limpiar

En ocasiones resulta interesante limitar el movimiento de un control gráfico mediante una constricción. Cuando no existe constricción, el control responde a las pulsaciones tal como se espera. Sin embargo, al aplicarse una constricción, el programa debe respetarla, lo que implica ajustar automáticamente las coordenadas del control. Debido a estos ajustes, las modificaciones introducidas manualmente por el usuario, ya sea mediante los pulsadores o escribiendo valores específicos, pueden no aplicarse de manera exacta.

Si se añade al control gráfico una constricción o condición de movimiento, el control sólo podrá desplazarse siguiendo la trayectoria indicada por dicha restricción. Por ejemplo, al definir la constricción **y = x²** (y = x^2), el control gráfico únicamente podrá moverse a lo largo de la parábola correspondiente. En este caso, la parábola se ha dibujado en la escena para que se visualice la ruta disponible, aunque su presencia gráfica no es estrictamente necesaria; la restricción se seguiría cumpliendo aunque la parábola no se mostrará.

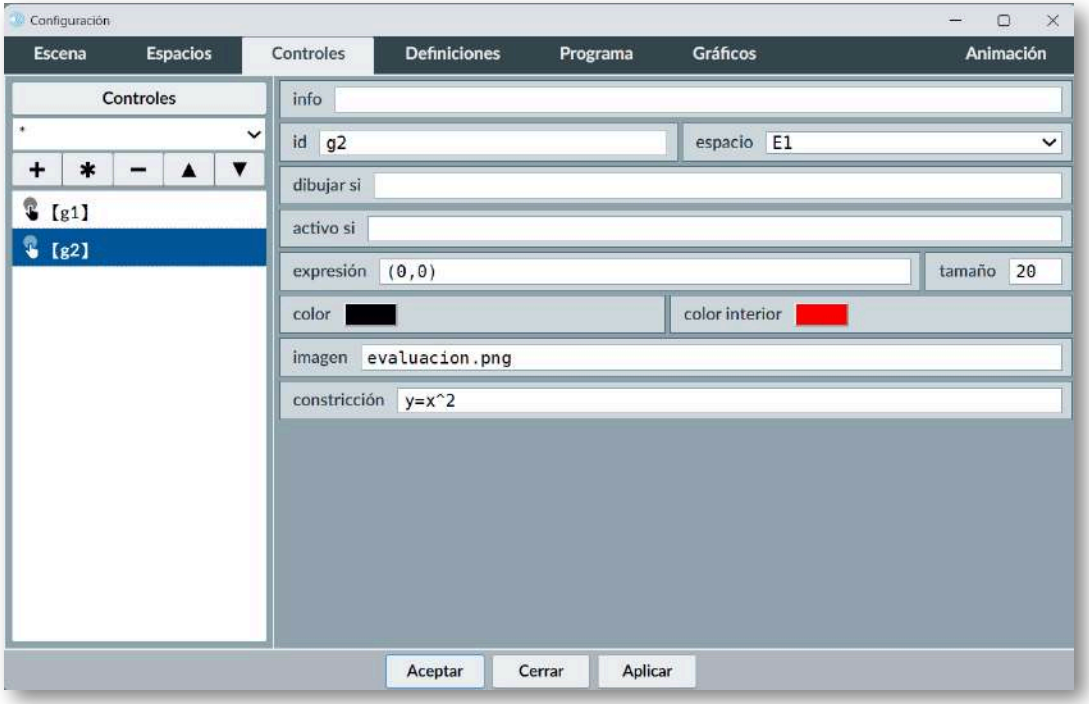


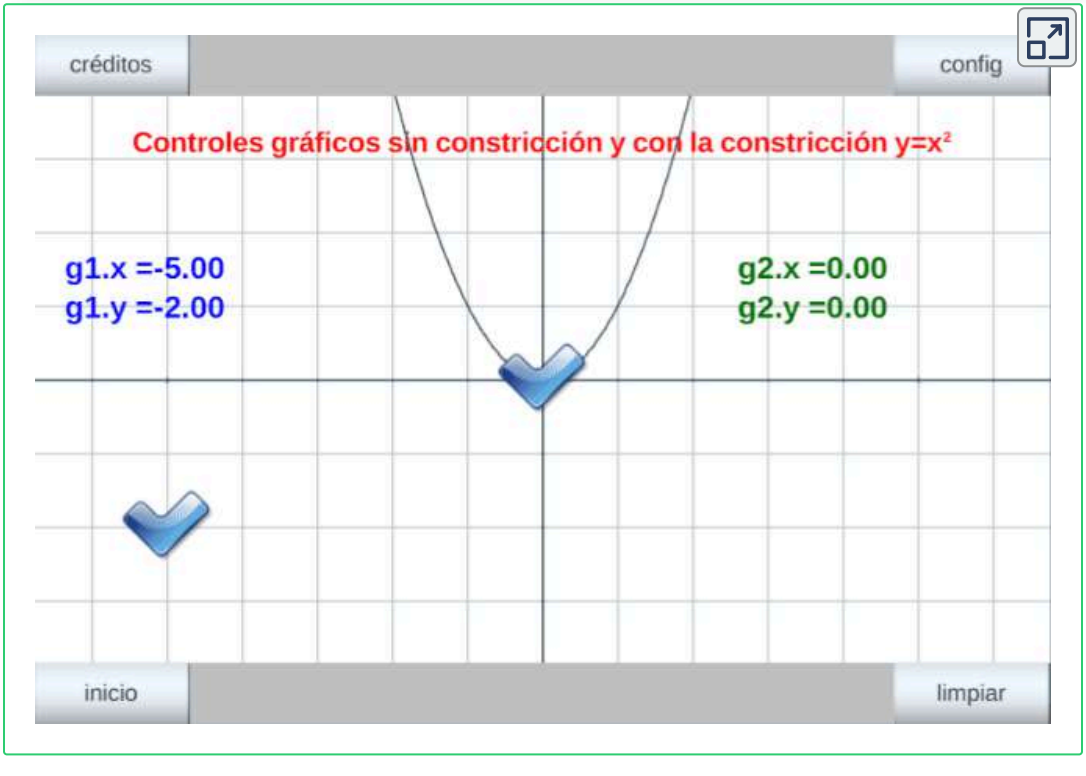


Si a un control gráfico se le asigna una imagen, por ejemplo, **evaluacion.png**, la imagen se mostrará como representación del control. Esta podrá moverse libremente por la escena, ya sea arrastrándola con el ratón o mediante otros controles que modifiquen las variables asociadas a su posición.

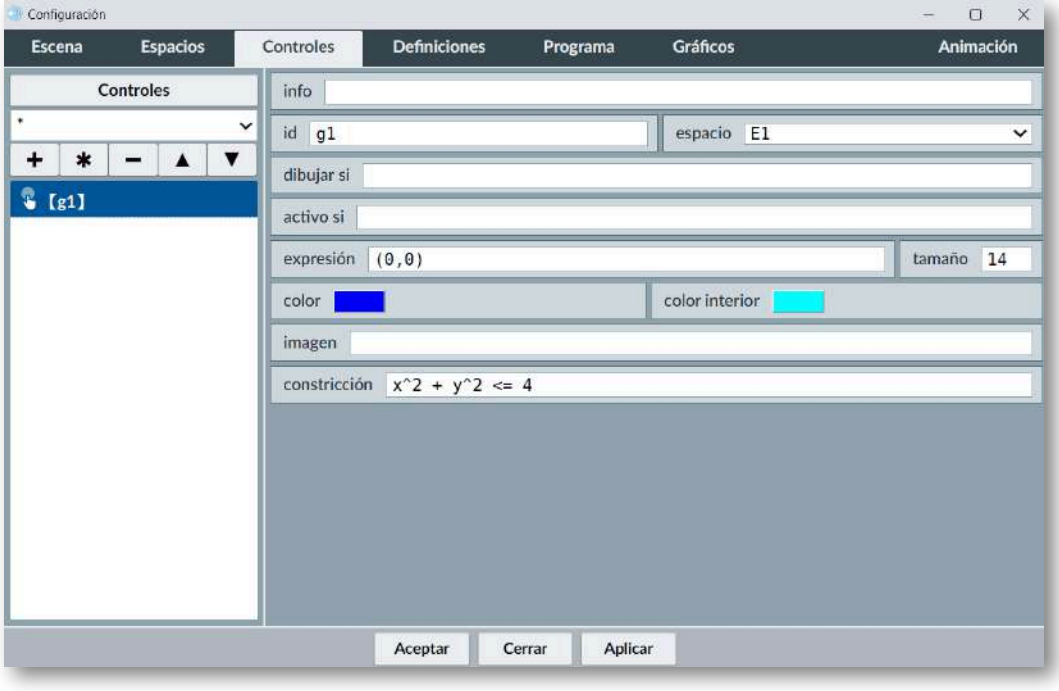
En la siguiente escena se han incluido dos controles gráficos: uno sin constricción, con identificador **g1**, y otro con la restricción definida por la parábola **y = x²**, con identificador **g2**. A ambos controles se les ha asignado la imagen **evaluacion.png**.

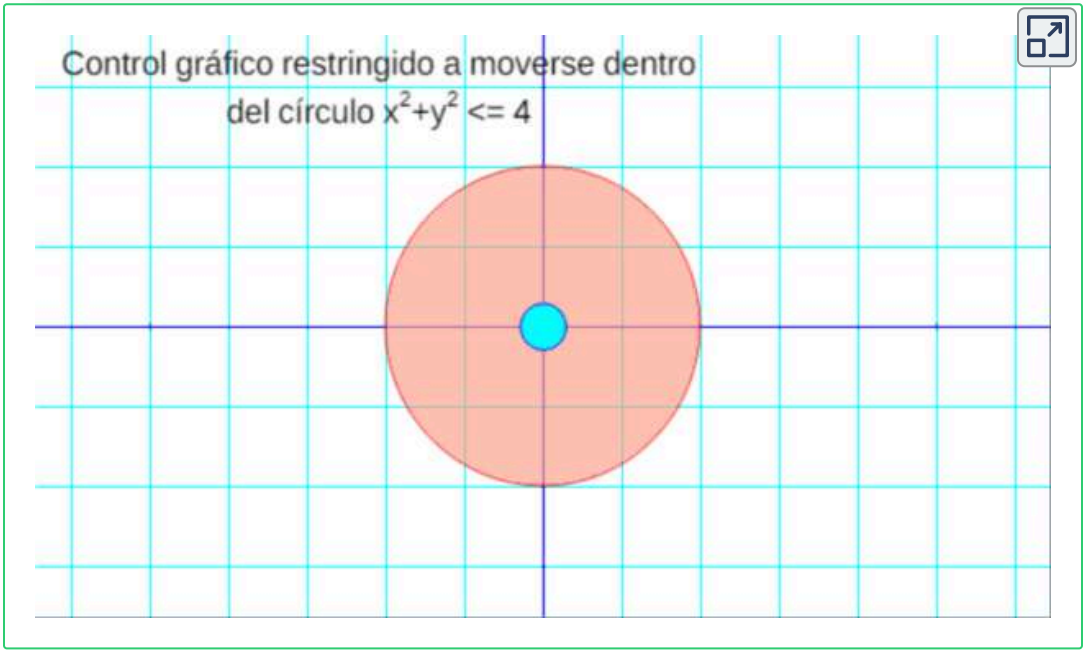
Se puede observar claramente la libertad de movimiento de **g1** frente a las limitaciones de **g2**. No se han añadido los controles asociados a sus coordenadas, aunque sí se muestran los valores de las variables correspondientes.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
[g1]
[g2]
info
id g2
espacio E1
dibujar si
activo si
expresión (0,0)
tamaño 20
color
color interior
imagen evaluacion.png
constricción y=x^2
Aceptar
Cerrar
Aplicar

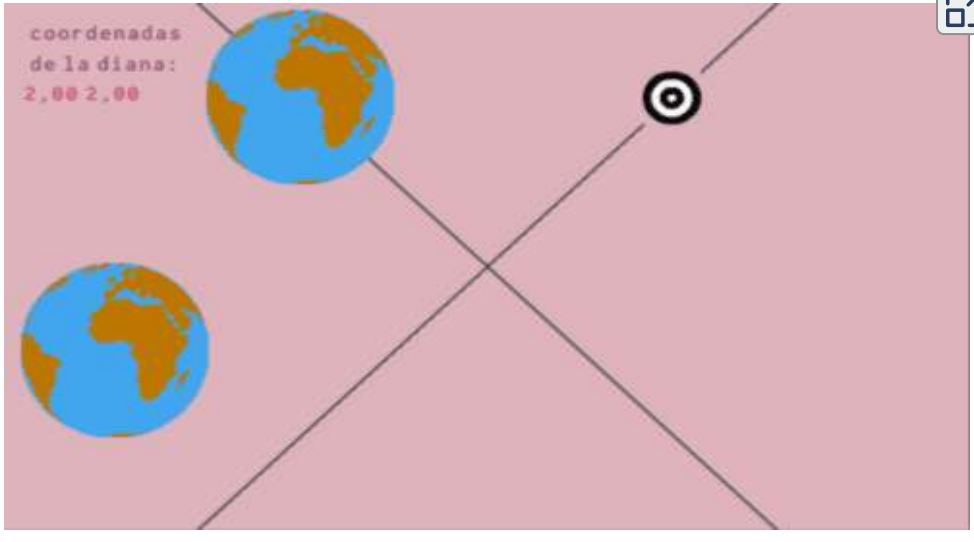

créditos
config
Controles gráficos sin constricción y con la constricción y=x²
g1.x =-5.00
g1.y =-2.00
g2.x =0.00
g2.y =0.00
inicio
limpiar

Presentamos un control gráfico cuya constricción limita su movimiento a un área específica dentro del espacio de la escena.

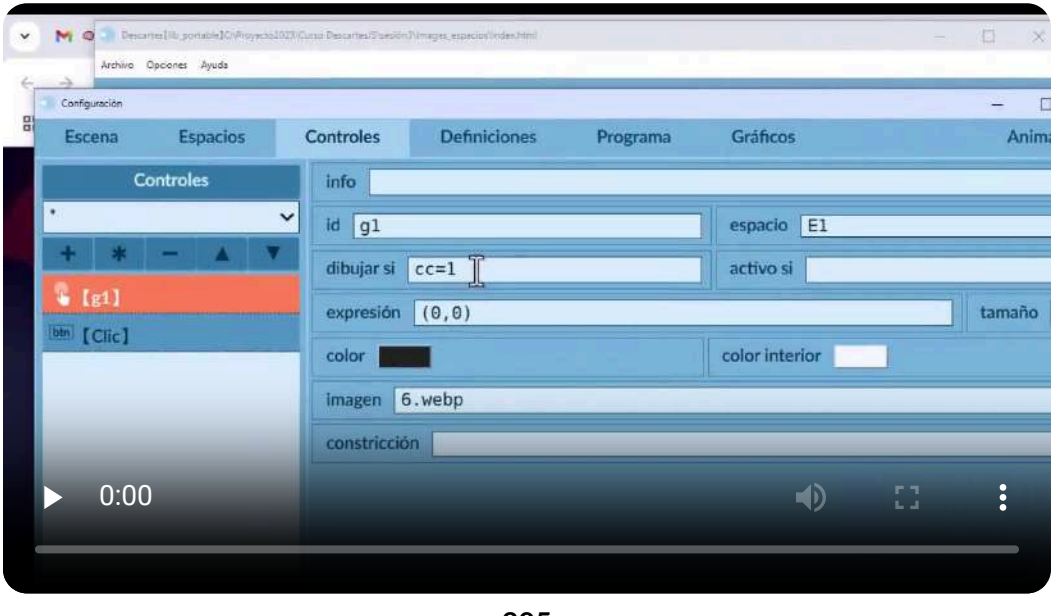




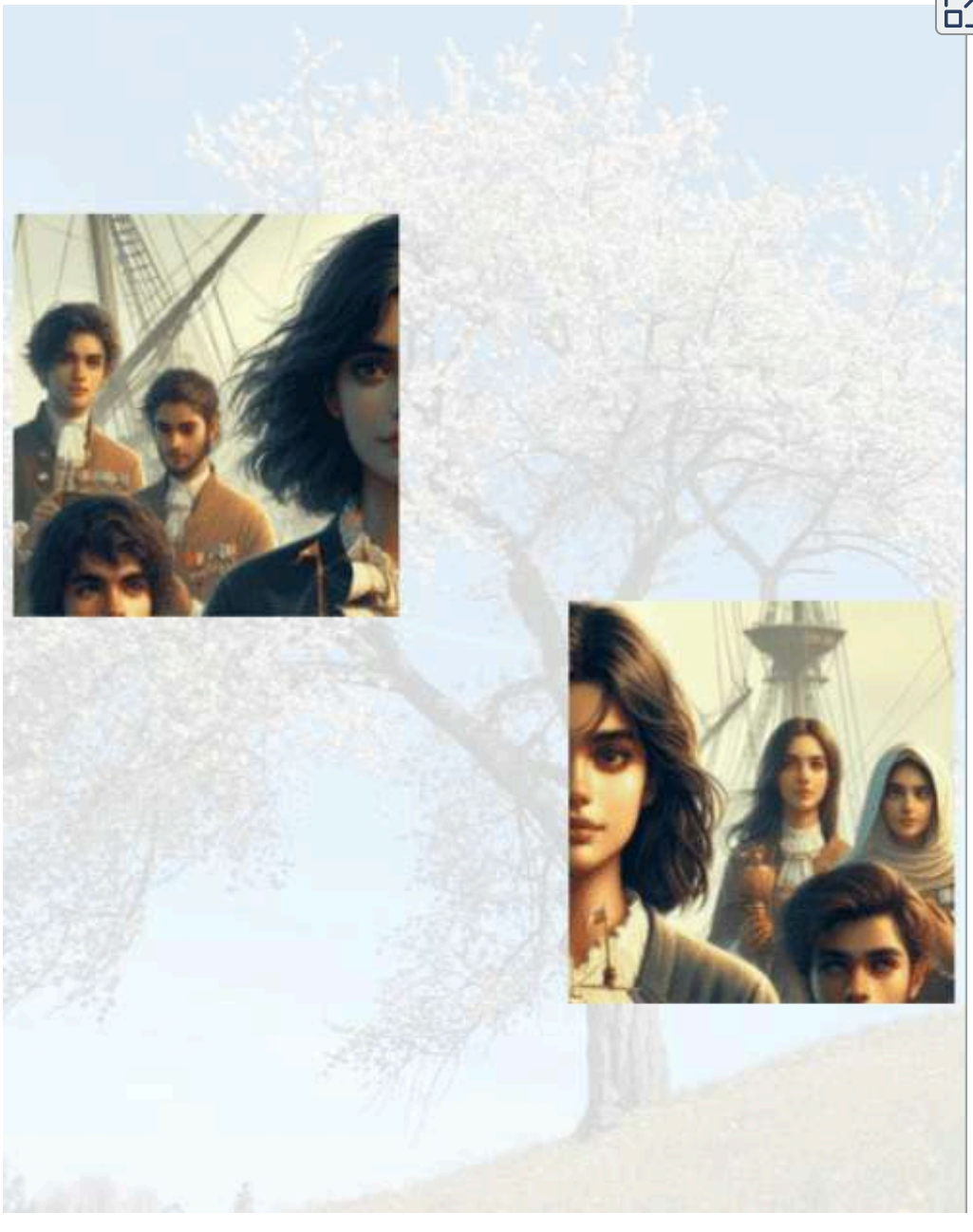
Incluimos a continuación una escena con varios controles gráficos representados mediante imágenes, algunos con constricción definida y otros con movimiento libre.



En el siguiente vídeo se muestra como asociar imágenes a controles tipo gráfico.



La siguiente escena muestra el resultado final tras asociar imágenes a los controles gráficos.



3.13 Control tipo Texto

Los controles de tipo texto consisten en bloques de texto enriquecido delimitados por un marco. El contenido puede editarse directamente desde el propio interactivo, sin necesidad de hacerlo únicamente desde el editor de configuraciones.

Además, es posible incorporar un sistema de pregunta y respuesta mediante un botón situado en la esquina inferior derecha del recuadro, lo que permite alternar entre ambos contenidos.

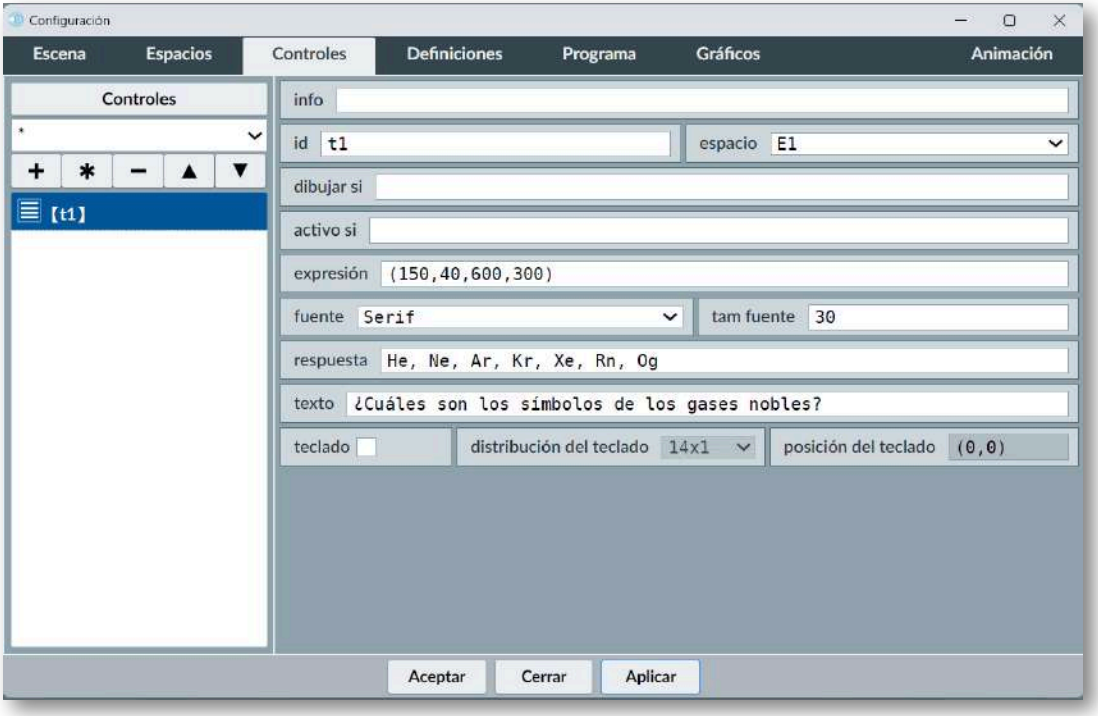
Los controles de texto son áreas en las que el usuario puede escribir o corregir textos y fórmulas. Además, permiten comprobar si la respuesta es correcta comparándola con la solución, a la que se accede mediante el botón **S** (de Solución).

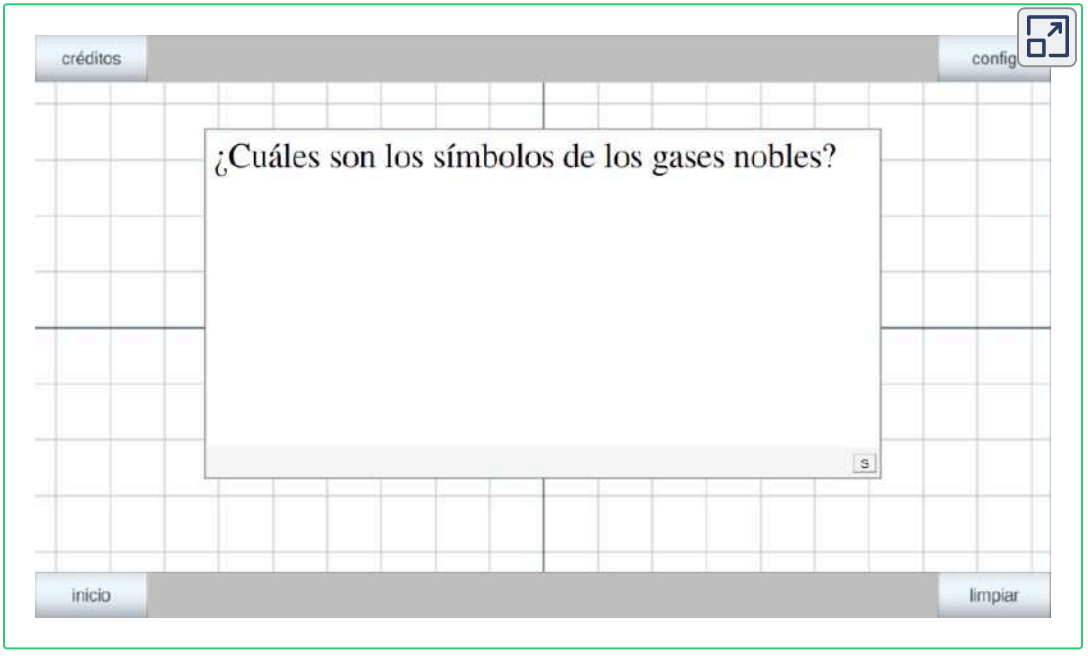
Los parámetros de este tipo de control son los siguientes:

- **fuente:** menú que permite elegir el tipo de letra utilizado en el texto.
- **tam fuente:** campo de texto donde se indica el tamaño de la fuente en puntos.
- **respuesta:** campo de texto que contiene la solución que se mostrará al pulsar el botón **S**. Esta respuesta se presenta únicamente como referencia; no se realiza una comprobación automática de lo introducido por el usuario. Si este parámetro se deja en blanco no aparece el botón **S**.
- **texto:** campo de texto destinado a plantear la pregunta o enunciado que el usuario debe responder.

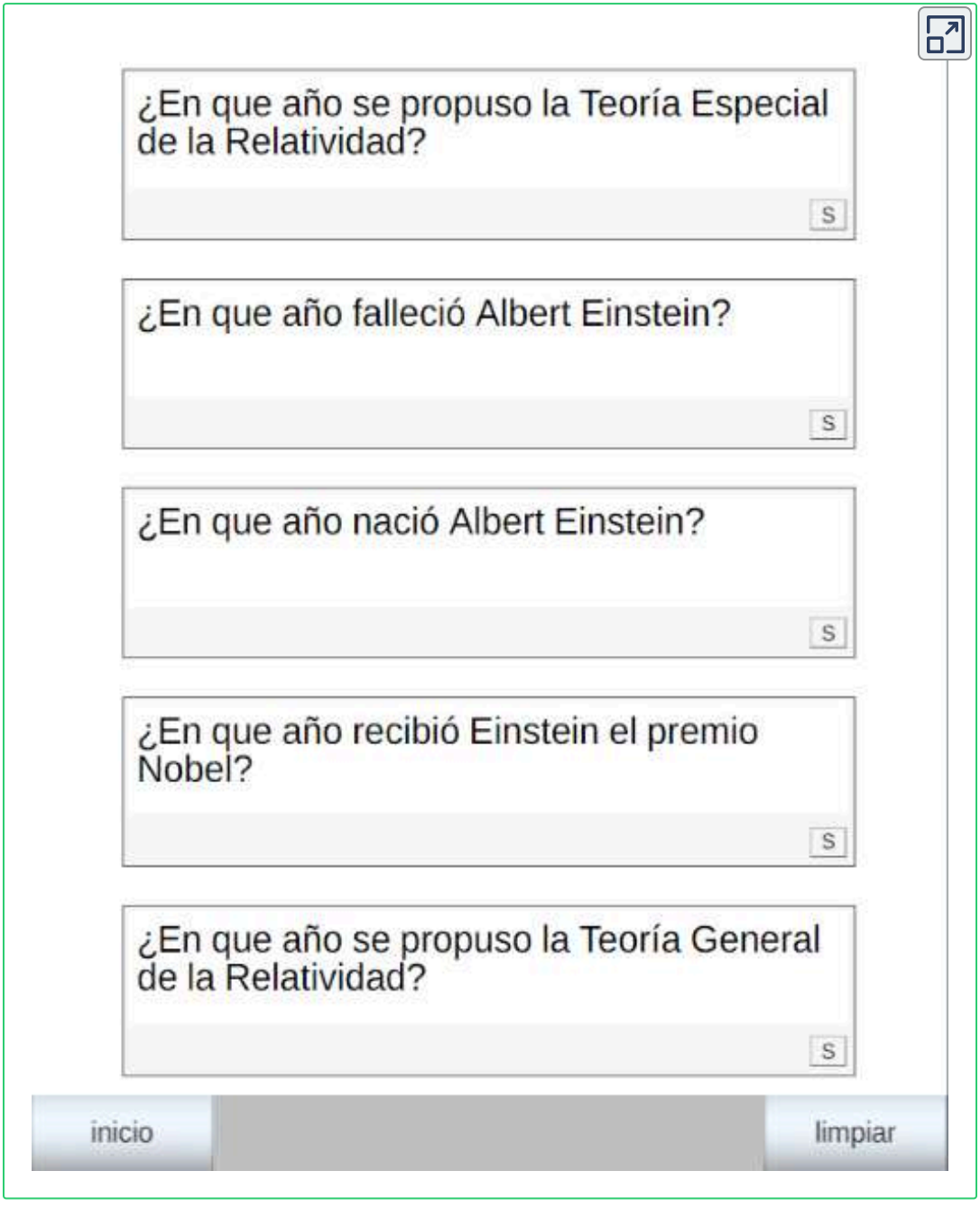
Introducimos un control de tipo texto con la pregunta: *¿Cuáles son los símbolos de los gases nobles?* El control se dibujará y permanecerá activo en todo momento. Estará situado en la posición (100, 50), con una anchura de 600 y una altura de 410, utilizando fuente **Serif** y tamaño 30.

La solución asociada al botón **S** será: **He, Ne, Ar, Kr, Xe, Rn, Og**.





En la siguiente escena se presenta una batería de controles de texto destinada a la realización de un cuestionario temático.

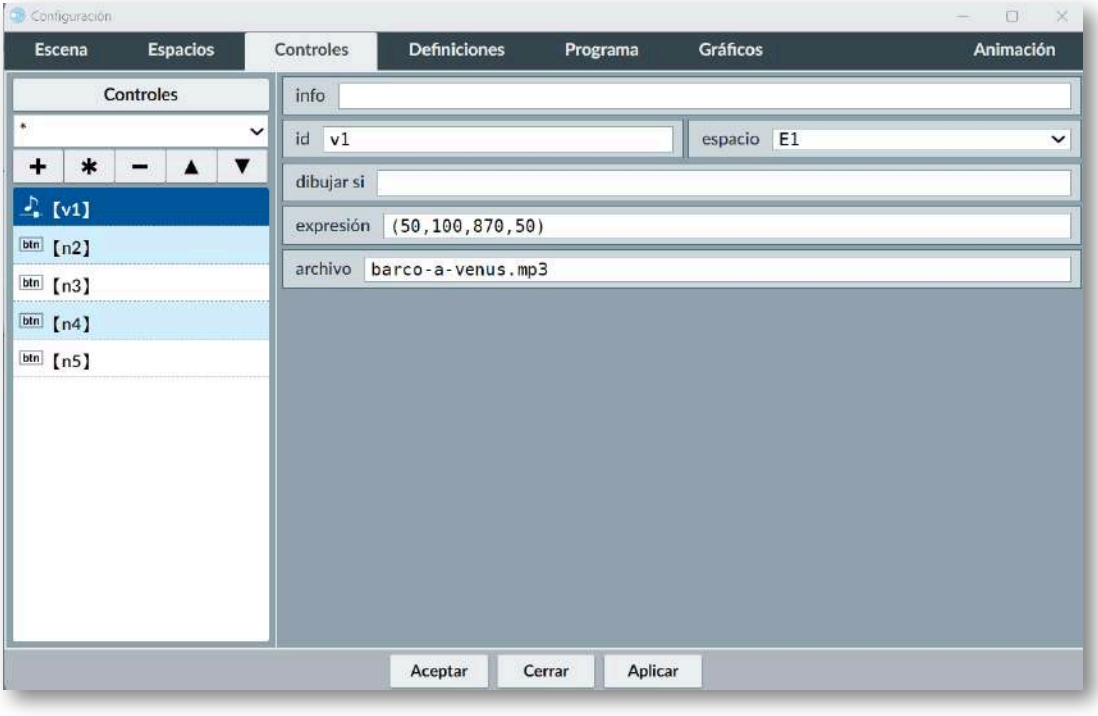


3.14 Control tipo Audio

Este control consiste en un reproductor de audio (con sus correspondientes controles) que puede activarse dentro del interactivo. Solo puede situarse en el interior de un espacio y permite reproducir archivos en formato **mp3** y **wav**.

El campo **archivo** es específico de este control. Se trata de un campo de texto en el que se introduce la ruta relativa a la ubicación del interactivo y que apunta al archivo de audio que se desea reproducir (si el archivo de audio está en el mismo nivel que la escena no hay que indicar la ruta).

Vamos a introducir en una escena un control de audio que reproduzca la canción ubicada en el archivo **barco-a-venus.mp3**. El control se colocará en la posición (50,100) con un ancho de 870 y una altura de 50.

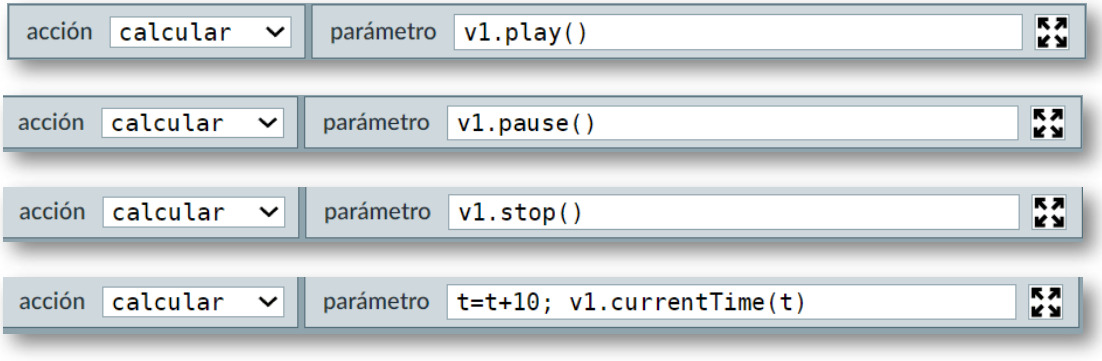




Hay funciones específicas para controlar la reproducción de los audios.

- **id.play()**: inicia la reproducción del audio o vídeo asociado al control.
- **id.pause()**: pausa la reproducción en el instante actual.
- **id.currentTime(t)**: sitúa la reproducción en el tiempo *t* indicado (en segundos).
- **id.stop()**: detiene completamente la reproducción del archivo vinculado al control.

En la escena anterior se han incorporado todas estas funciones como acciones de tipo **calcular**, incluyéndolas en el campo **parámetro** de cada uno de los cuatro controles.



En la siguiente escena se incluye otro ejemplo.



En la próxima escena se incluyen dos controles, cada uno con la acción de reproducir un audio distinto. La primera vez que se pulsa, el audio se reproduce; la segunda, se detiene y así sucesivamente.



3.15 Control tipo Vídeo

Este control consiste en un reproductor de vídeo que puede activarse dentro del interactivo y cuyo funcionamiento es muy similar al del control de audio. Permite reproducir archivos en formato **mp4**, **WebM** y **ogv**.

El campo **archivo** es específico de este control. Consiste en un campo de texto en el que se introduce la ruta relativa al archivo de vídeo que se desea reproducir. Si el archivo se encuentra en una subcarpeta, es necesario utilizar la diagonal sencilla para separar carpetas.

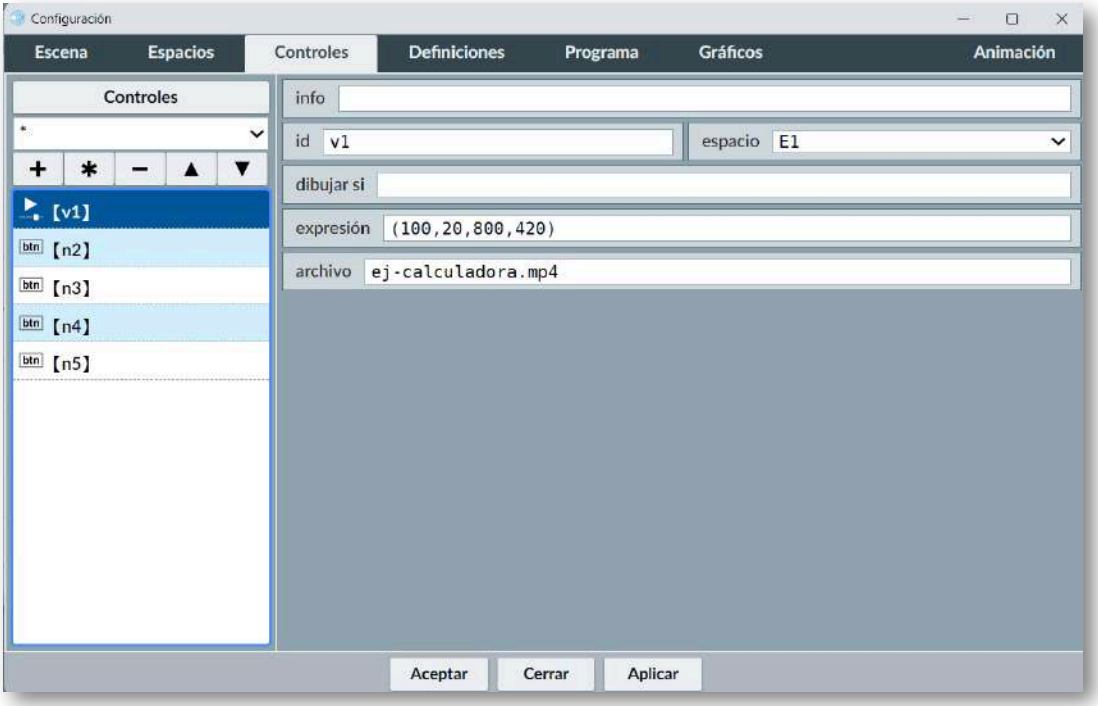
Al igual que ocurre con los controles de audio, la funcionalidad del reproductor de vídeo puede variar según el navegador. Algunos permiten extender el vídeo a pantalla completa, mientras que otros no. Estas diferencias dependen del reproductor del navegador y no de Descartes.

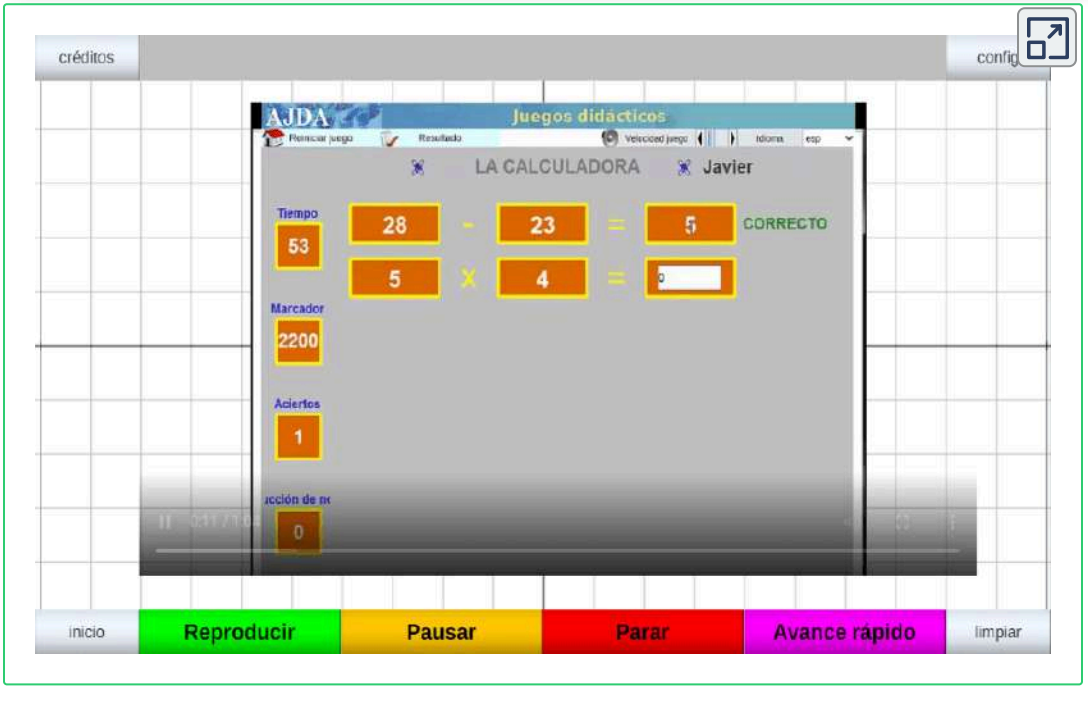
Se recomienda que al utilizar un control de vídeo, el parámetro **expresión** incluya las cuatro entradas correspondientes a posición horizontal, posición vertical, ancho y alto. Es aconsejable que el ancho y alto del control coincidan con las dimensiones originales del vídeo o, al menos, mantengan la misma proporción.

Si se desea mostrar una portada inicial para el vídeo, basta con guardar en la misma carpeta del vídeo una imagen en formato **png** con el mismo nombre que el vídeo y con dimensiones idénticas a las de este.

Las funciones **id.play()**, **id.pause()**, **id.currentTime(t)** y **id.stop()** también se aplican a los controles de vídeo y mantienen la misma funcionalidad que en los controles de audio.

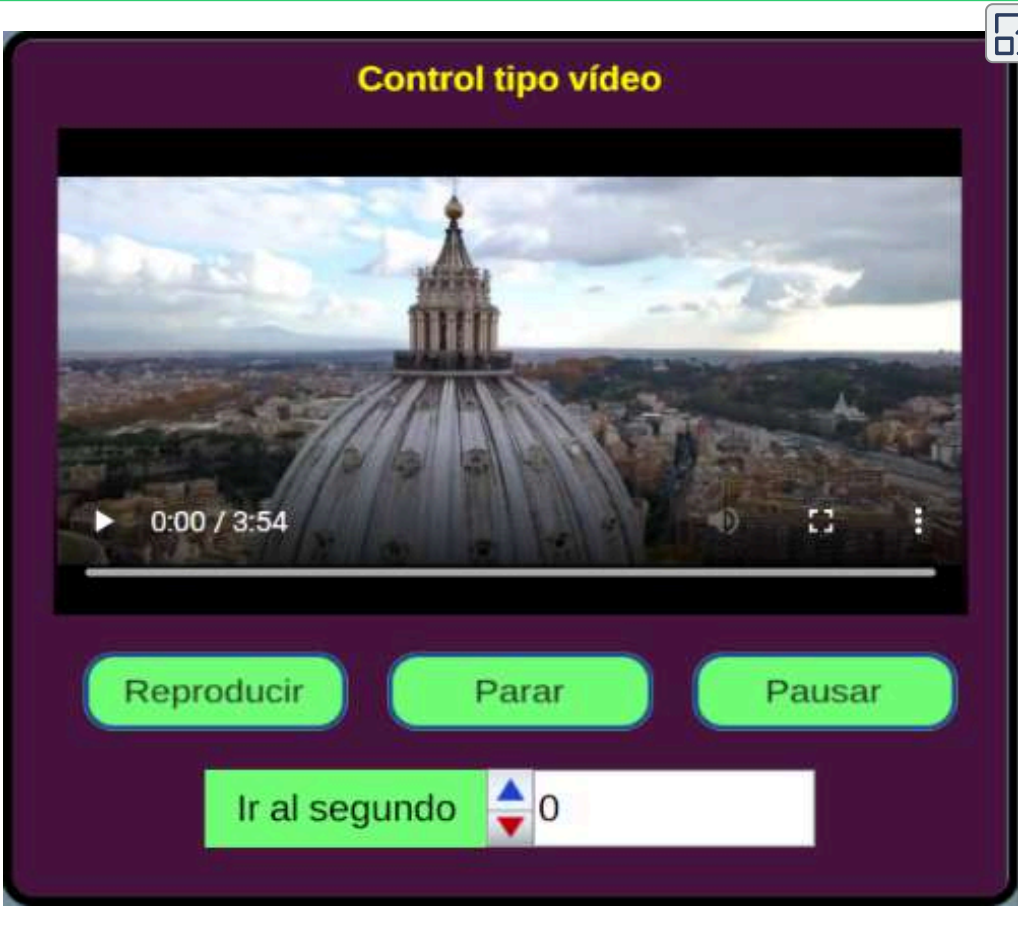
Incluimos a continuación un ejemplo de escena similar al utilizado en el control de audio, pero aplicado a un control de vídeo.





Se incluyen dos nuevos ejemplos con controles de vídeo.


Sonido. Barco a Venus de Mecano
0:01 / 4:44
Reproducir
Pausar
Parar
Vídeo.Presentación juegos didácticos
AJDA
Juegos didácticos
LA CALCULADORA
Javier
Tiempo
58
28
23
Marcador
2000
Aciertos
0
0:01 / 1:04
Reproducir
Pausar
Parar

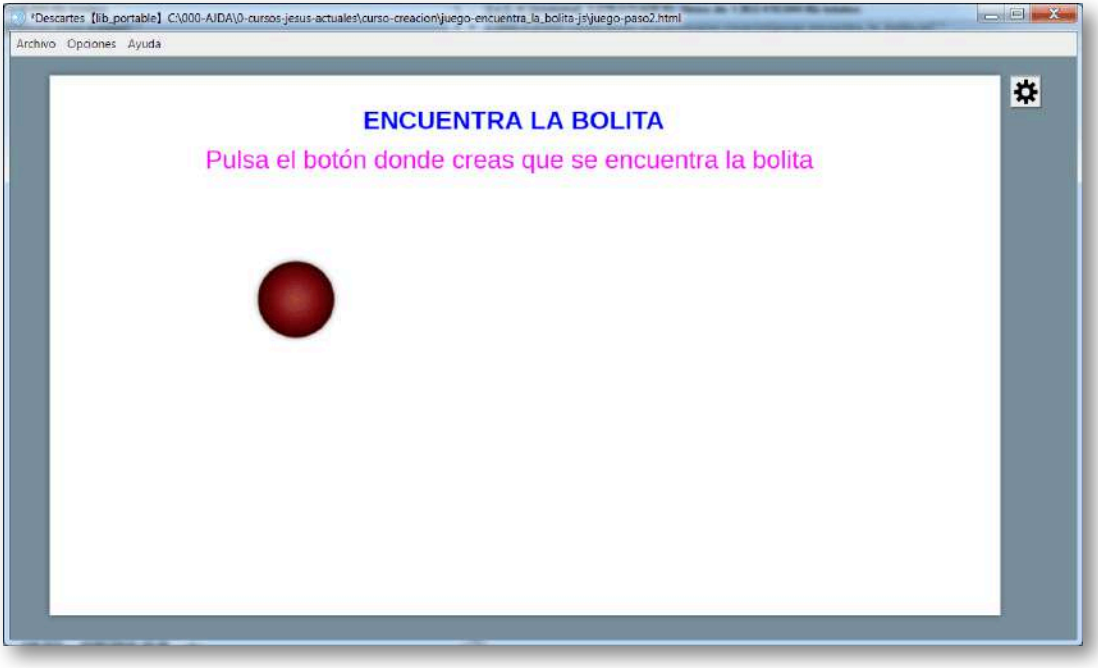

Control tipo vídeo
0:00 / 3:54
Reproducir
Parar
Pausar
Ir al segundo
0

La siguiente escena que contiene los diferentes tipos de controles.



3.16 Mi primer juego paso a paso. Paso 3

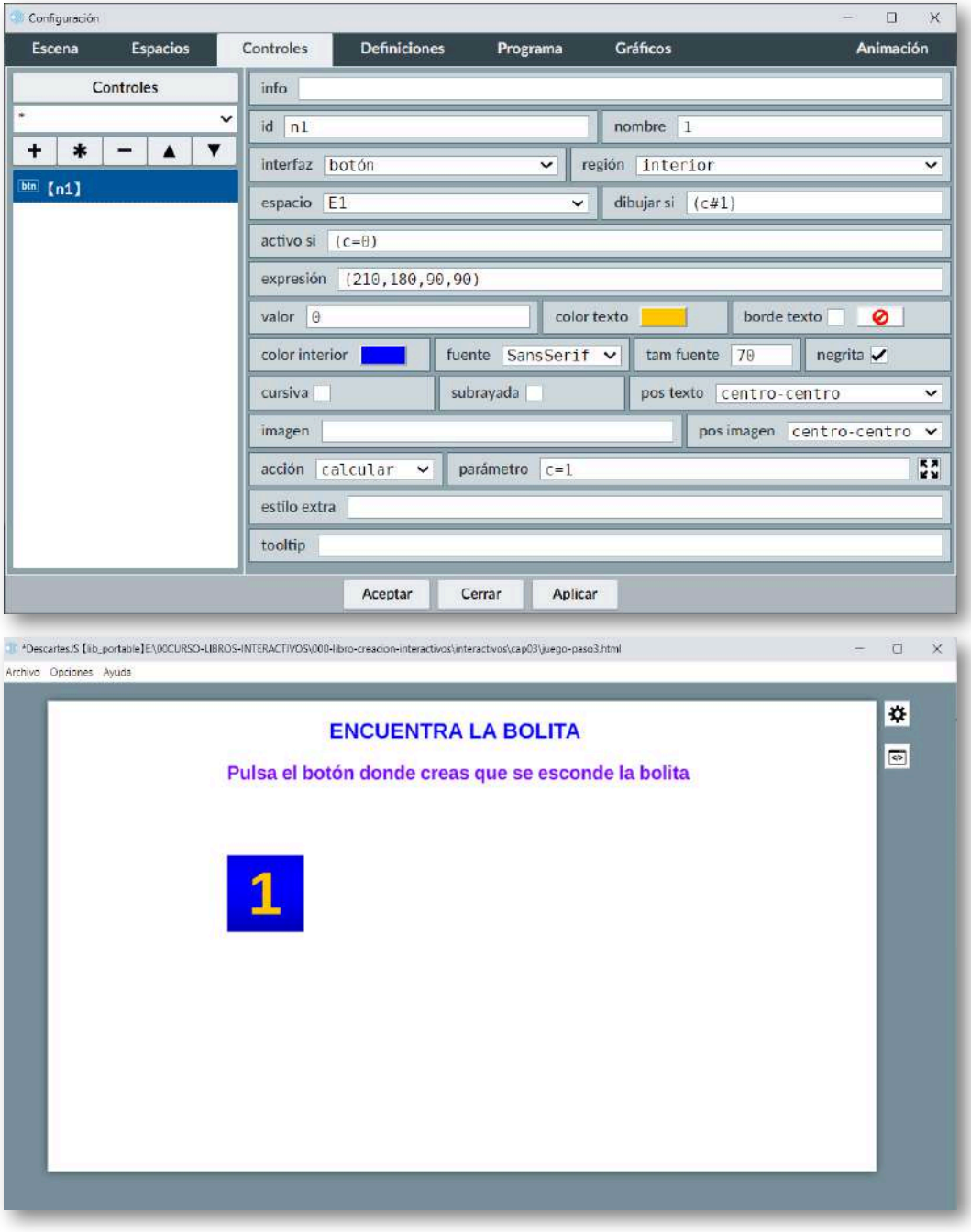
Vamos a realizar el paso 3 del juego. En primer lugar, abrimos el juego con el editor de **Descartes**, accediendo al menú *Archivo* > *Abrir* y seleccionando el archivo llamado **juego-paso2**.



A continuación, añadiremos un control que llamaremos **n1**. Para ello, utilizaremos el menú *Controles* y seleccionaremos **+ Agregar control**. Este control será de tipo **botón**, de color azul, con el texto "1" que cubrirá la bolita.

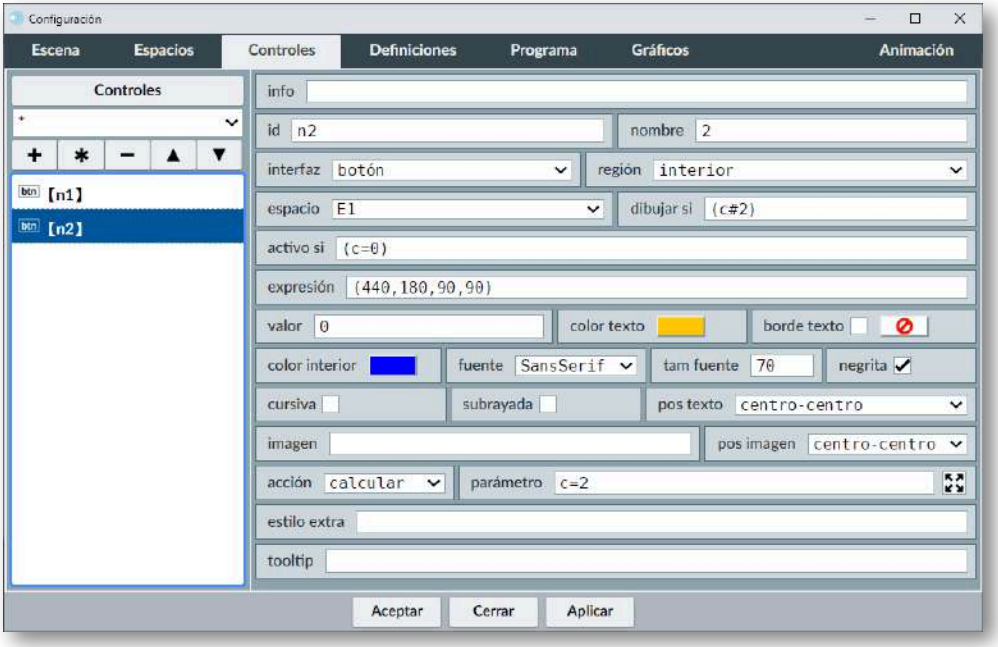
Se configurará la acción **calcular** y en el campo *parámetro* se asignará el valor **1** a la variable **c**, es decir, **c=1**. En *dibujar-si* se establecerá la condición **c#1**, lo que significa que el botón desaparecerá una vez pulsado, y en *activo-si* se pondrá **c=0**, de forma que el botón no pueda pulsarse si **c** ya ha tomado un valor mayor que 0.

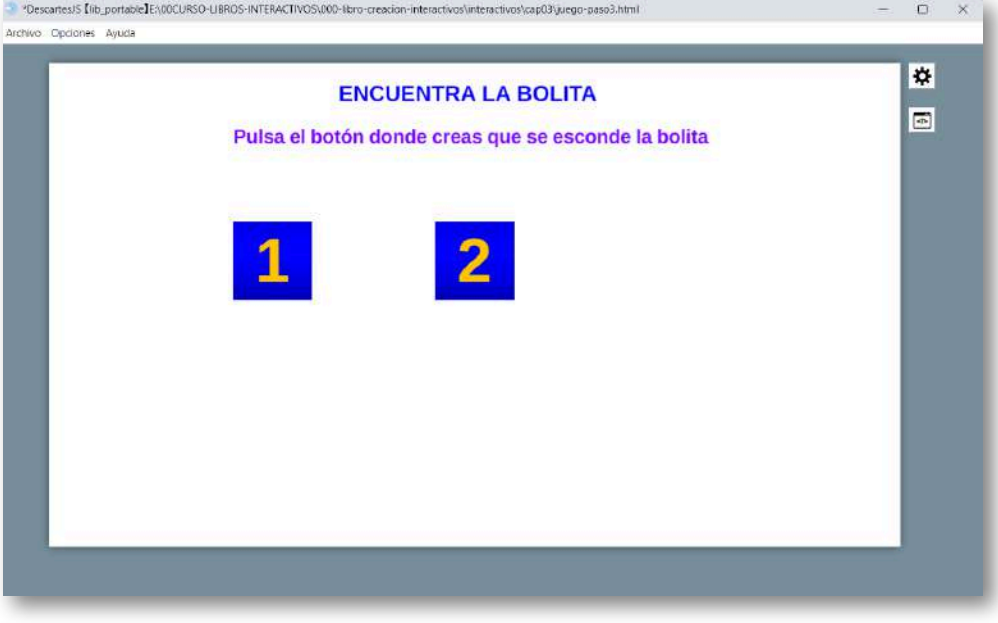
En la siguiente captura de pantalla se muestra la configuración del control **n1**.



Copiamos el control **n1** desde el menú *Controles* (botón * **Copiar control numérico**) y le asignamos el identificador **n2** y el nombre visible "2". Lo situaremos 130 puntos a la derecha del primero.

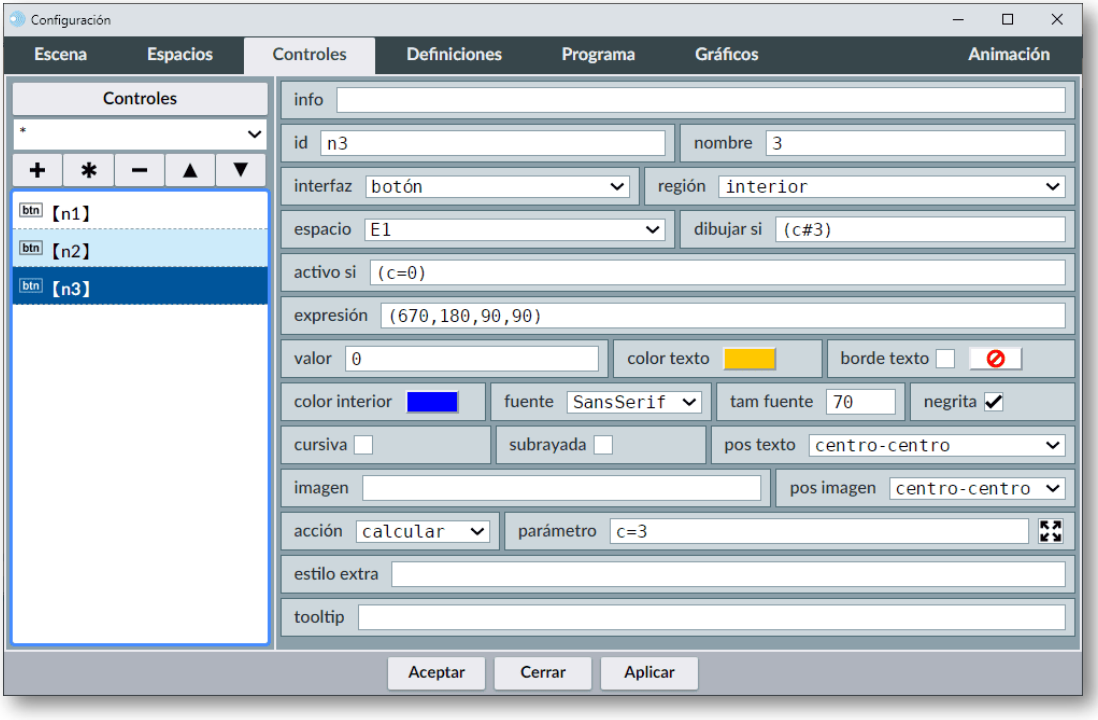
Este control tendrá la acción **calcular** y en *parámetro* asignará **c=2**. En *dibujar-si* se establecerá **c#2** para que desaparezca al pulsarlo, y en *activo-si* se indicará **c=0**, impidiendo que vuelva a activarse cuando **c** sea distinto de 0.



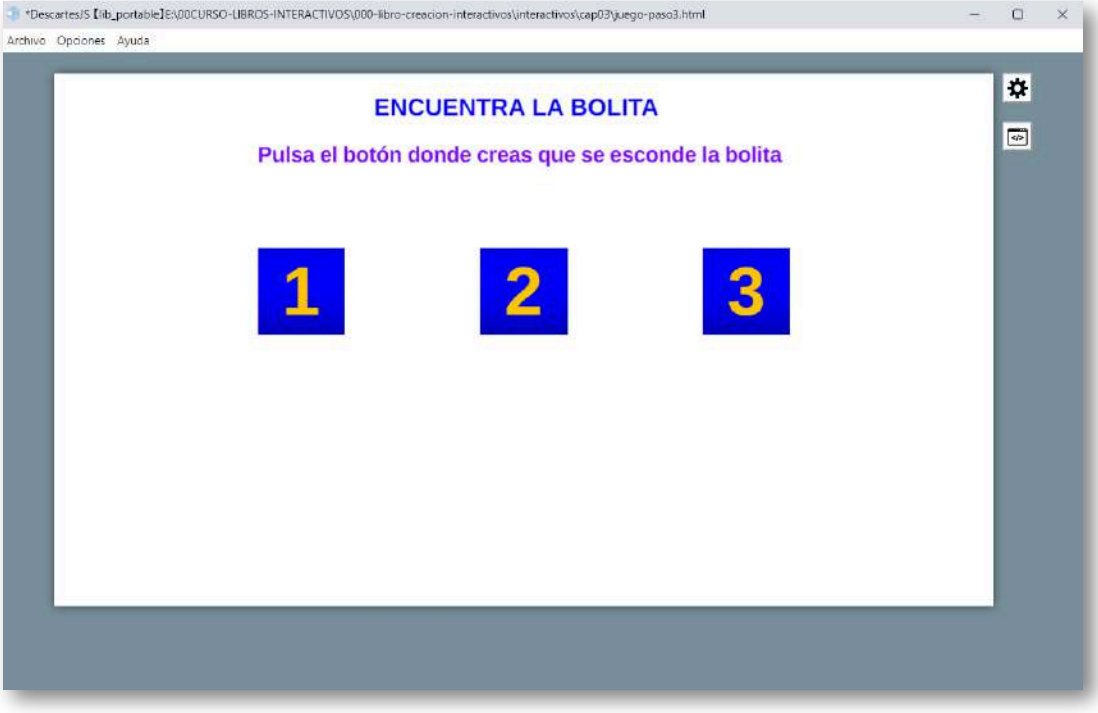


Repetimos el proceso copiando el control **n2** desde el menú *Controles* (botón * **Copiar control numérico**) y le asignamos el identificador **n3** y el nombre visible “3”. Lo situaremos 130 puntos a la derecha del segundo.

Este control tendrá la acción **calcular** y, en el campo *parámetro*, asignará **c=3**, de modo que al pulsarlo la variable tome ese valor. En *dibujar-si* se establecerá **c#3** para que el botón desaparezca una vez seleccionado, y en *activo-si* se indicará **c=0**, evitando que pueda volver a activarse cuando **c** sea distinto de 0.



Con ello hemos creado tres botones que desaparecen al ser pulsados, reforzando visualmente la elección realizada. Además, cuando se selecciona uno, los demás quedan desactivados y ya no pueden utilizarse, garantizando que solo pueda hacerse una única elección. Asimismo, según el valor que adopte la variable **c**, podremos identificar con precisión qué botón ha sido seleccionado y actuar en consecuencia dentro del programa.

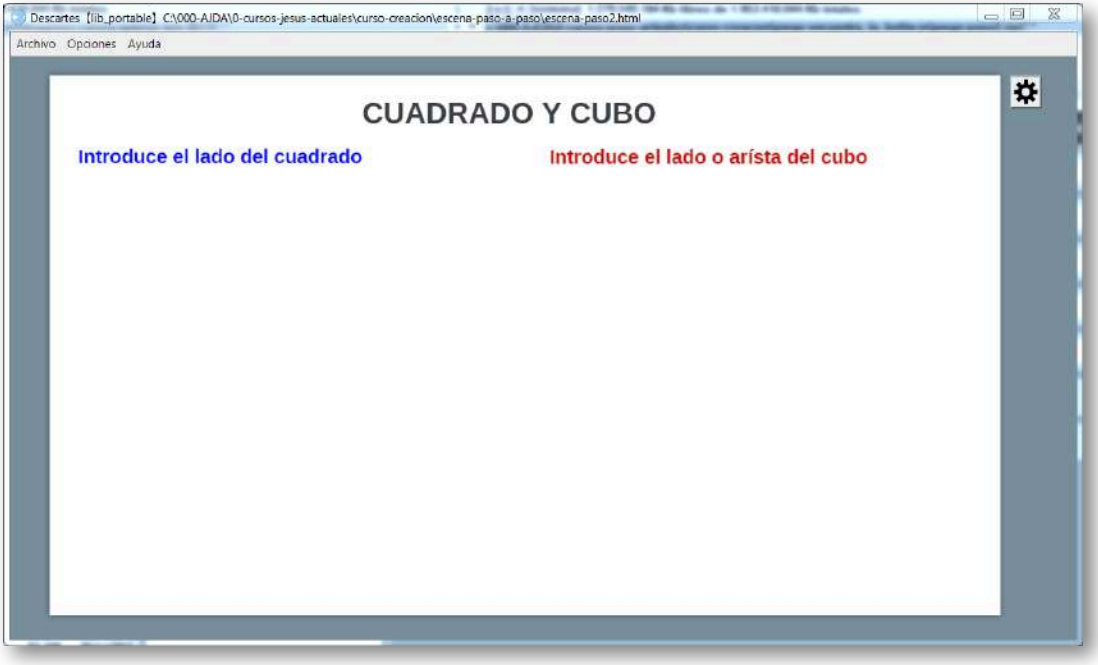




Guardamos la escena del juego con el nombre **juego-paso3**.

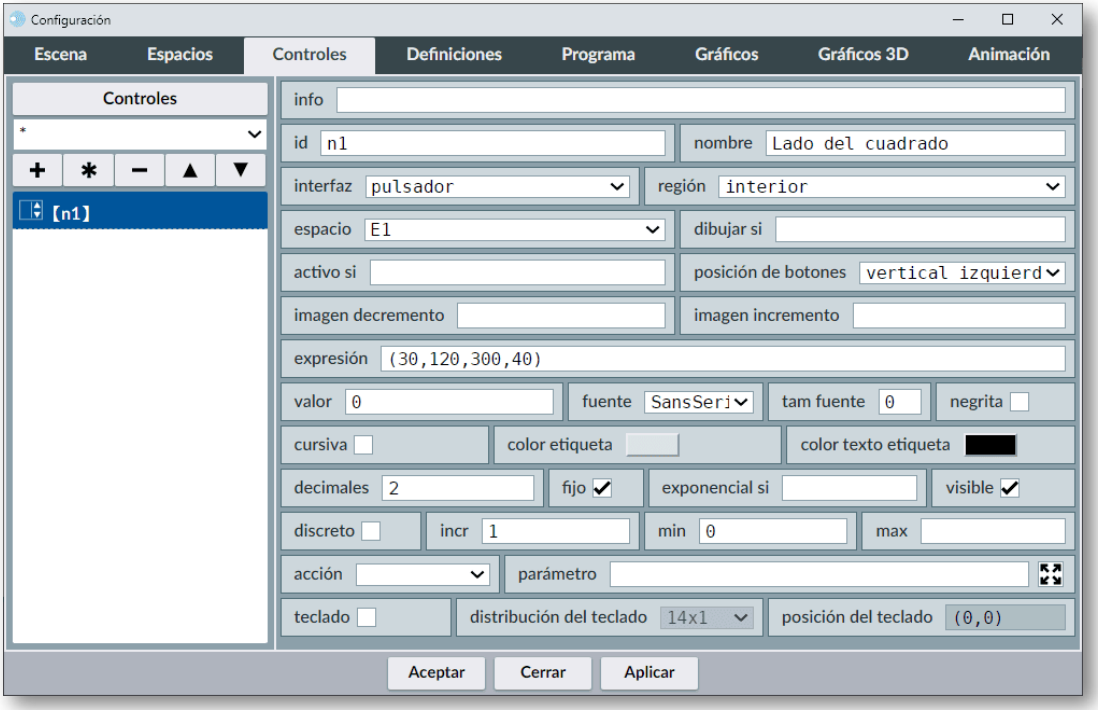
3.17 Mi primera escena paso a paso. Paso 3

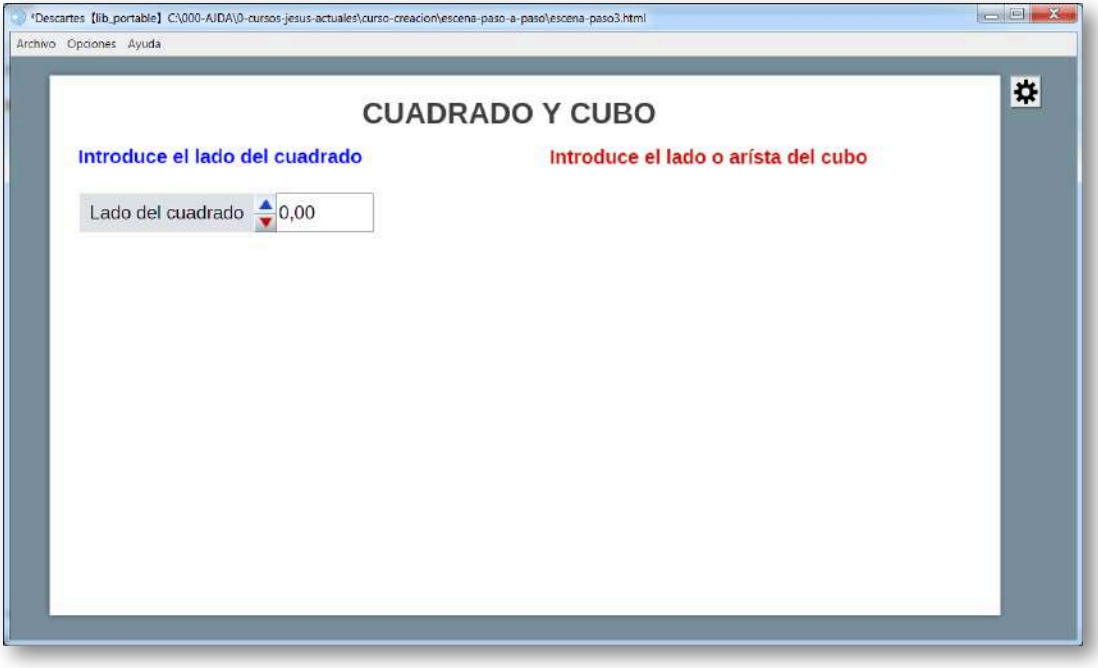
Vamos a realizar el paso 3 de la primera escena. Para comenzar, abrimos la escena con el editor de **Descartes**, accediendo al menú *Archivo* > *Abrir* y seleccionando el archivo llamado **escena-paso2**. Esto nos permitirá continuar con la configuración de los controles y variables necesarios para el desarrollo del interactivo.



A continuación, vamos a introducir un control de tipo **pulsador** con el nombre “Lado del cuadrado”, ubicado en el espacio 2D, posición interior en coordenadas absolutas (30,120,300,40), al que asignaremos el identificador **n1**. Este control permitirá al usuario definir el valor del lado del cuadrado, facilitando la interacción directa con la escena.

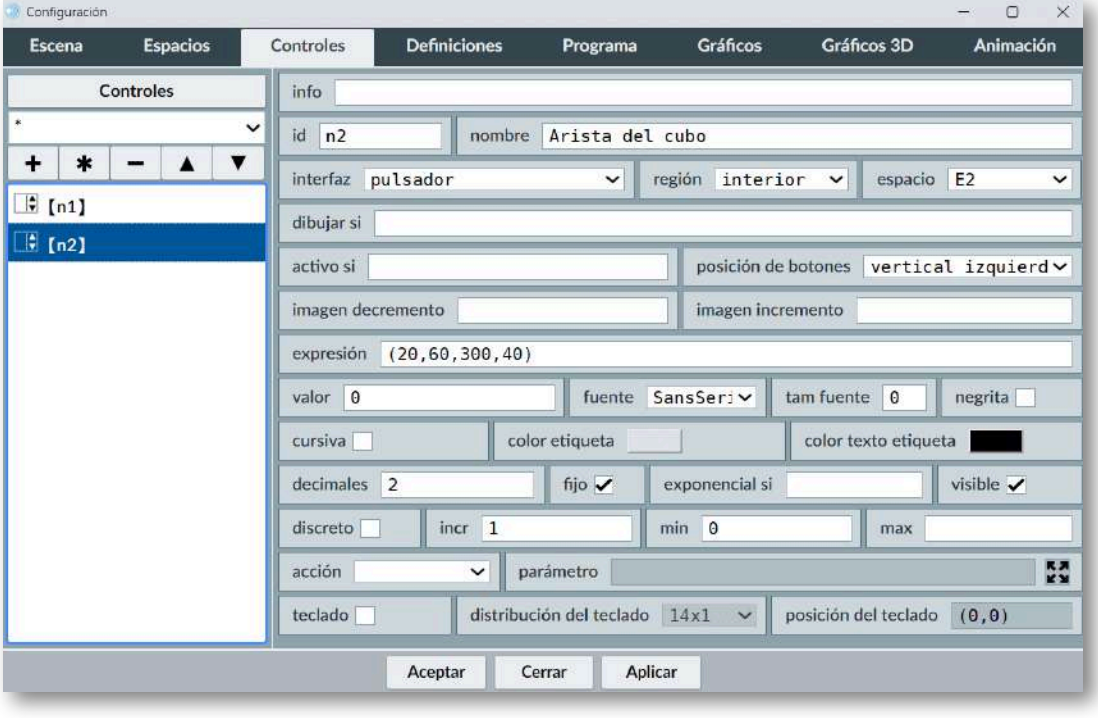
El valor mínimo que podrá tomar será 0, no se establecerá un valor máximo, y el control admitirá hasta 2 decimales, ofreciendo así precisión en la selección de valores.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Gráficos 3D
Animación
Controles
[n1]
info
id n1
nombre Lado del cuadrado
interfaz pulsador
región interior
espacio E1
dibujar si
activo si
posición de botones vertical izquierd
imagen decremento
imagen incremento
expresión (30,120,300,40)
valor 0
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 2
fijo
exponencial si
visible
discreto
incr 1
min 0
max
acción
parámetro
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar

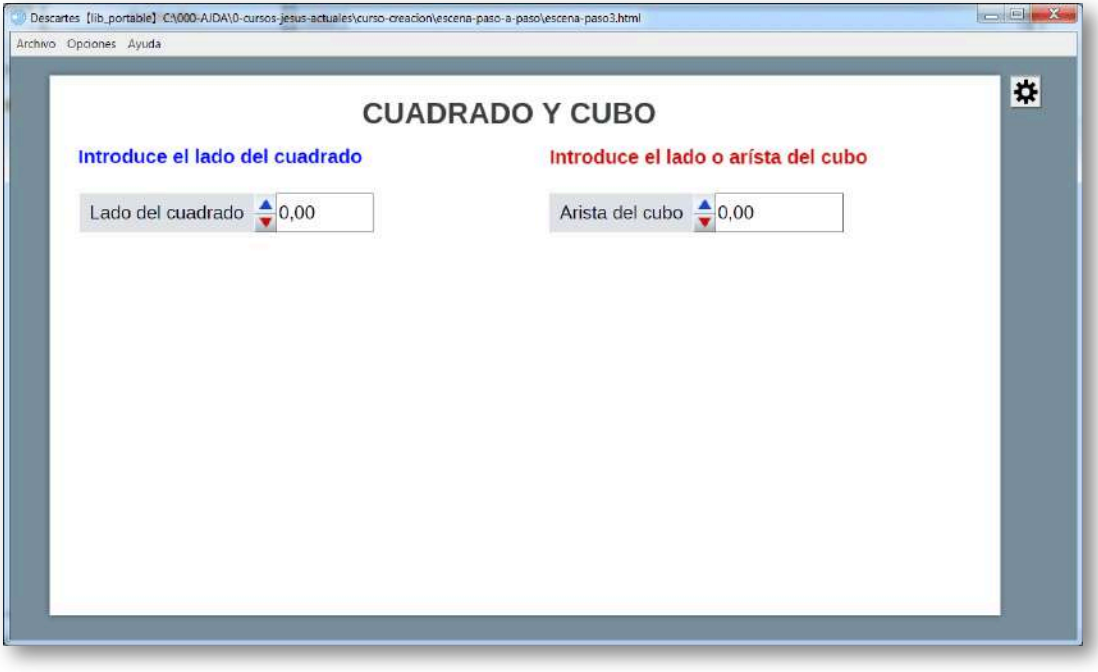

*Descartes [lib_portable] C:\000-AJDA\0-cursos-jesus-actuales\curso-creacion\escena-paso-a-paso\escena-paso3.html
Archivo Opciones Ayuda
CUADRADO Y CUBO
Introduce el lado del cuadrado
Introduce el lado o arísta del cubo
Lado del cuadrado 0,00

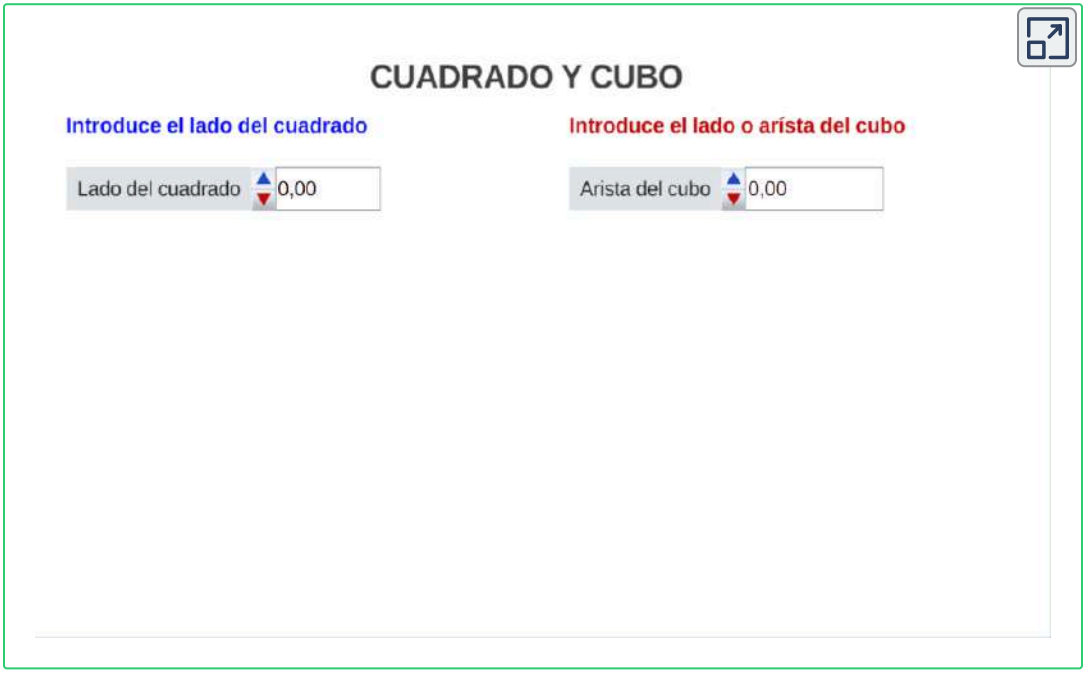
También añadiremos un control de tipo **pulsador** con el nombre "Arista del cubo", ubicado en el espacio 3D, con posición interior (30,120), una anchura de 300 y una altura de 40, al que asignaremos el identificador **n2**. Este quedará correctamente integrado dentro del entorno tridimensional de la escena.

El control permitirá establecer el valor de la arista del cubo de forma precisa. El valor mínimo no podrá ser inferior a cero, no tendrá límite máximo definido y admitirá hasta dos cifras decimales, lo que facilitará trabajar con medidas exactas.



Tras aceptar los cambios realizados, la escena contará con dos controles de tipo **pulsador**: el primero ubicado en un espacio 2D y el segundo en un espacio 3D. Estos controles permitirán definir el lado del cuadrado (figura geométrica bidimensional) y la arista del cubo (cuerpo geométrico tridimensional), respectivamente. A continuación, se muestra la correspondiente escena.





Guardamos la escena con el nombre escena-paso3.



Capítulo 4

Programación en escenas

Capítulo 4
Programación en escenas

4.1 Introducción 229

4.2 Operadores generales 232

4.3 Operadores booleanos y su uso en condicionales 237

4.4 Operadores reservados 248

4.5 Funciones comunes intrínsecas de Descartes 257

4.6 Funciones propias del lenguaje de Descartes 262

4.7 Funciones 267

4.8 Variables 278

4.9 Vectores 283

4.10 Matrices 294

4.11 Biblioteca 305

4.12 Resumen sobre definiciones 309

4.13 Eventos 311

4.14 Algoritmos Inicio y Cálculos 320

4.15 Algunos aspectos sobre programación 335

4.16 Mi primer juego paso a paso. Paso 4 345

4.17 Mi primera escena paso a paso. Paso 4 350

4.1 Introducción

En muchas ocasiones es necesario que las escenas realicen operaciones internas (no visibles en pantalla) para que funcionen correctamente. Estas operaciones pueden ser sencillas o muy complejas. En esta unidad abordaremos tanto las operaciones y funciones que vienen definidas por defecto como aquellas que puede crear el propio usuario. Por ello, este capítulo se titula **"Programación en escenas"**.

A continuación, presentamos algunos ejemplos de operaciones habituales en la programación de escenas:

- **Definir variables:** por ejemplo, asignar π = 3.1416 o establecer la aceleración de la gravedad como g = 9.81.
- **Utilizar operadores:** para establecer condiciones lógicas que permitan actuar en función de ellas.
- **Definir expresiones:** por ejemplo, indicar que la superficie S de un rectángulo de lados a y b es S = a · b.
- **Incluir eventos o acciones:** por ejemplo, si se cumple (t > 10), reproducir el archivo *musica.mp3*.
- **Definir funciones matemáticas:** por ejemplo, y = 2x + 3 o y = 4 cos(3x/2).

Descartes incluye por defecto numerosas funciones matemáticas, como funciones aritméticas, trigonométricas y logarítmicas, entre otras.

Como ejemplo, incluimos una imagen del juego **"La calculadora"**, donde resulta evidente la necesidad de incorporar distintos elementos de programación para poder realizar las operaciones matemáticas que se desarrollan en él.

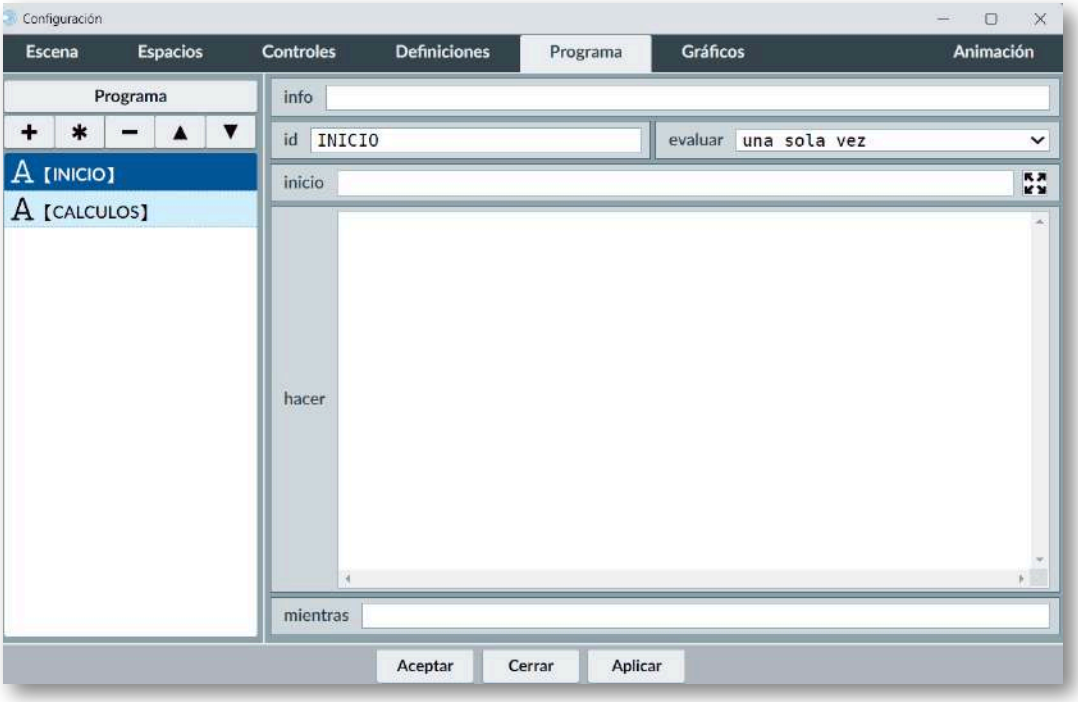


Hay dos selectores especialmente relacionados con la programación de escenas de Descartes: el **selector Programa** y el **selector Definiciones**.

El selector Programa

El selector Programa contiene parte de la programación interna que se realiza en Descartes. Incluye, por defecto, dos tipos de algoritmos presentes en cualquier escena: **INICIO** y **CÁLCULOS**, además de un elemento denominado **evento**, que permite ejecutar acciones cuando se cumple una determinada condición.

El único elemento nuevo que puede agregarse en este selector es el de tipo **evento**. Los algoritmos **INICIO** y **CÁLCULOS** ya están presentes por defecto en todas las escenas y no pueden añadirse como nuevos elementos, aunque sí pueden clonarse, cambiar su nombre o eliminarse.



El selector Definiciones

El selector Definiciones también concentra una parte fundamental de la programación en Descartes. En él se pueden crear:

- **Variables**, que pueden almacenar un valor o una expresión.
- **Vectores**, que funcionan como variables capaces de contener una lista de valores.
- **Matrices**, que amplían el concepto de vector permitiendo organizar valores en filas y columnas.
- **Funciones**, que agrupan instrucciones ejecutables cuando se desee y que pueden operar de forma cíclica.
- **Bibliotecas**, que permiten organizar y agrupar varias definiciones para mantener el código estructurado y ordenado.

Este selector no tiene ningún elemento definido por defecto; su contenido depende exclusivamente de las definiciones que incorpore el usuario.

4.2 Operadores generales

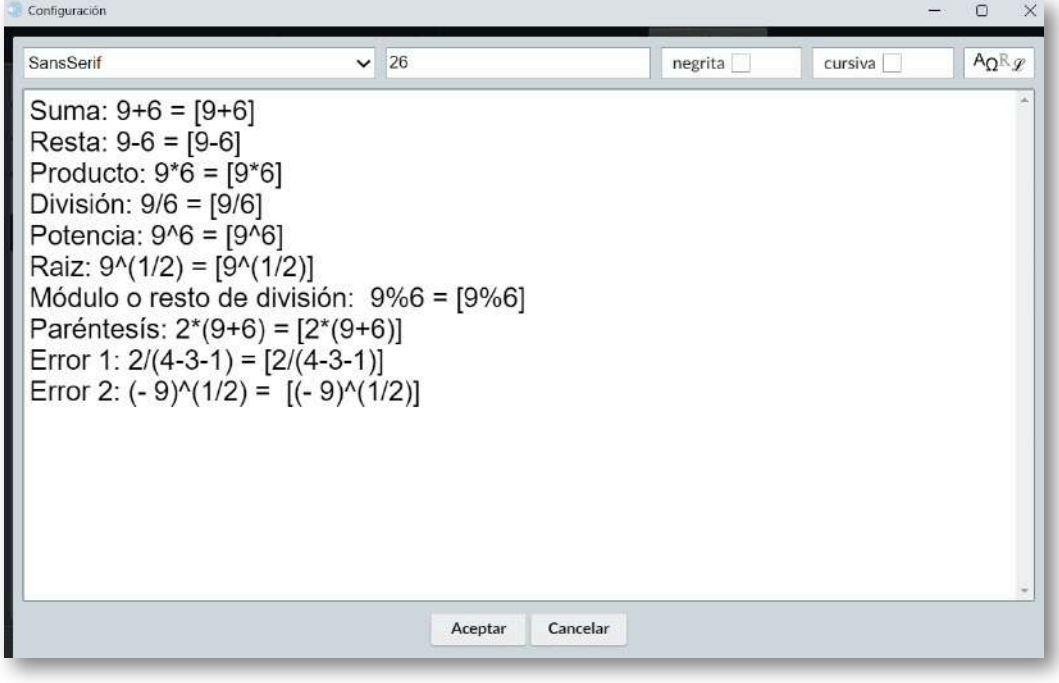
Los operadores generales son los signos matemáticos utilizados para asignar valores, realizar cálculos y agrupar expresiones. Pueden trabajar con constantes, variables, funciones y también con textos.

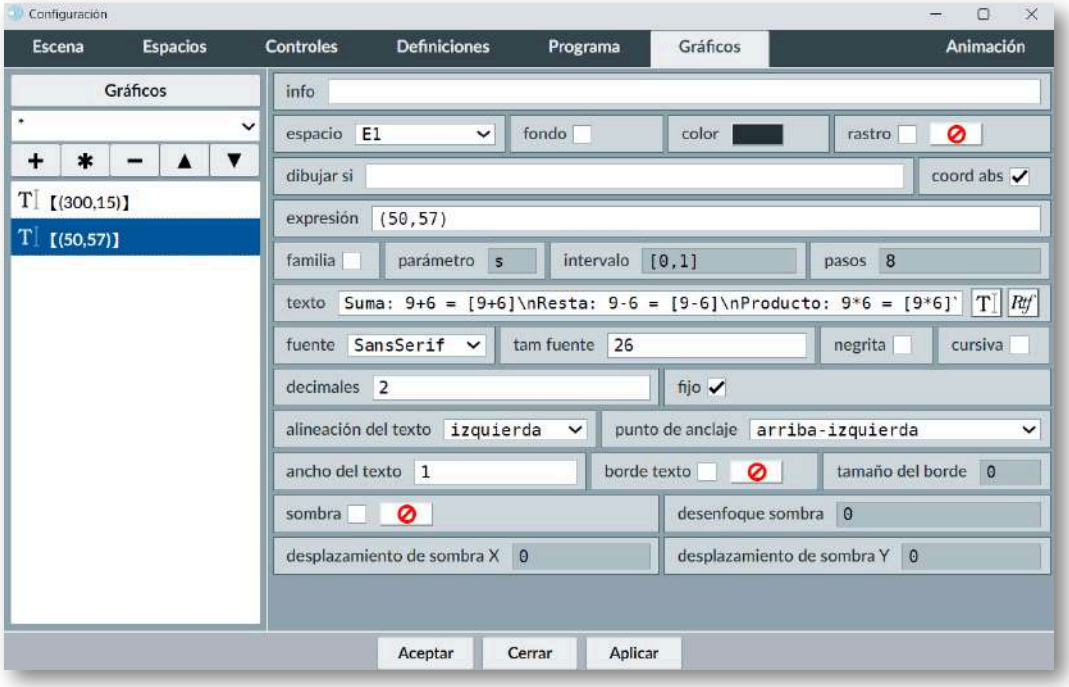
- **=: Operador de asignación.** Asigna a la variable de la izquierda el valor situado a la derecha. Ejemplos: a=5, b=2*4, c='hola'. No debe confundirse con **==**, que se utiliza para comparar valores.
- **+: Operador suma.** También permite concatenar textos. Ejemplos: a=4+3, b='te'+'saludo'.
- **-: Operador resta.** Sustracción de dos números. Ejemplo: a=8-4.
- ***: Operador producto.** Multiplicación de dos cifras. Ejemplo: a=8*pi.
- **/: Operador división.** Devuelve el cociente entre dos valores. Ejemplo: c=a/b.
- **^: Operador potencia.** Eleva el valor de la izquierda al exponente de la derecha. Ejemplos: q=2^3, r=3^(1/3). En funciones del tipo y=x^p, si p no es entero, solo se definen para x>0. Para representar correctamente raíces de índice impar en todos los reales, debe usarse la forma (x^3)^(1/5). En general, la raíz q-ésima se expresa como ^(1/q).
- **%: Operador módulo o resto.** Devuelve el residuo de una división. Ejemplo: 5%3=2.
- **( ): Paréntesis.** Permiten agrupar expresiones y modificar el orden de operaciones. Su uso es similar al de las Matemáticas, pero las limitaciones de escribir las expresiones en una sola línea hacen que los paréntesis sean aún más necesarios que cuando escribimos a mano. Así, si queremos expresar que C es 1 dividido entre la suma de A+B, debemos escribir: **C=1/(A+B)** y no **C=1/A+B**, ya que en este último caso se dividiría 1 entre A y posteriormente se sumaría B al resultado.

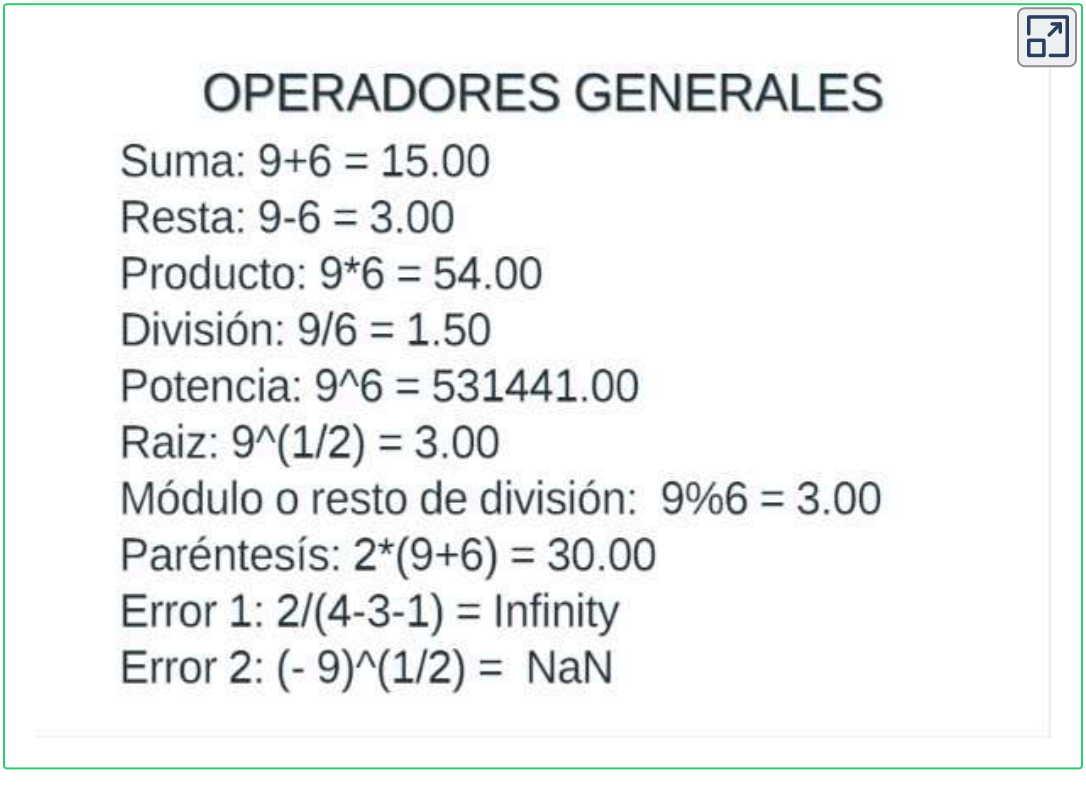
Es importante tener en cuenta los posibles errores que pueden cometerse al realizar asignaciones y operaciones en general, ya que estos pueden afectar al funcionamiento correcto de la escena.

- **Infinity**: Indica una división entre cero.
- **NaN**: Señala una operación no válida, como la raíz de un número negativo o el uso incorrecto de operadores con textos.
- **Errores en consola**: Informan de problemas como paréntesis sin cerrar o condicionales incompletas. Se recomienda aplicar cambios con frecuencia y revisar la consola para detectar errores rápidamente.

A continuación, presentamos una escena que muestra los diferentes operadores generales y algunos ejemplos de errores. Para visualizarlo de manera sencilla, hemos incluido un control de texto en el que se realizan las distintas operaciones. Recordamos que los números entre corchetes en el editor de texto se interpretan como operaciones matemáticas.




Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
【(300,15)】
【(50,57)】
info
espacio E1
fondo
color
rastro
dibujar si
coord abs
expresión (50,57)
familia
parámetro s
intervalo [0,1]
pasos 8
texto Suma: 9+6 = [9+6]\nResta: 9-6 = [9-6]\nProducto: 9*6 = [9*6]
fuente SansSerif
tam fuente 26
negrita
cursiva
decimales 2
fijo
alineación del texto izquierda
punto de anclaje arriba-izquierda
ancho del texto 1
borde texto
tamaño del borde 0
sombra
desenfoque sombra 0
desplazamiento de sombra X 0
desplazamiento de sombra Y 0
Aceptar
Cerrar
Aplicar


OPERADORES GENERALES
Suma: 9+6 = 15.00
Resta: 9-6 = 3.00
Producto: 9*6 = 54.00
División: 9/6 = 1.50
Potencia: 9^6 = 531441.00
Raiz: 9^(1/2) = 3.00
Módulo o resto de división: 9%6 = 3.00
Paréntesís: 2*(9+6) = 30.00
Error 1: 2/(4-3-1) = Infinity
Error 2: (- 9)^(1/2) = NaN

Al igual que en cualquier lenguaje de programación o incluso en calculadoras, **Descartes** sigue un orden predeterminado de operaciones. Cuando hay varias operaciones en una instrucción, Descartes primero atiende las potencias, luego multiplicaciones y divisiones, y finalmente sumas y restas, todo de izquierda a derecha.

Considere la expresión: a = 3 + 2 * 3 + 4 / 5^2

Primero se aplican las potencias: a = 3 + 2 * 3 + 4 / 25

Luego se aplican multiplicaciones y divisiones de izquierda a derecha: a = 3 + 6 + 0.16

Finalmente, sumas y restas: a = 9.16

Para modificar este orden, se usan **paréntesis**. Usar paréntesis correctamente permite mayor control sobre las operaciones y ayuda a evitar errores. Una práctica recomendada es cerrar cada paréntesis recién abierto antes de introducir su contenido.

Por ejemplo, para la raíz quinta de 32: 32^(1/5)

Sin paréntesis, 32^1/5 se evalúa como (32^1)/5 = 6.4, mientras que con paréntesis se obtiene: 32^(1/5) = 2

Los paréntesis se resuelven primero, respetando el orden tradicional dentro de ellos. Pueden anidarse, como en:

2 * (1 + 3 / (2 + 1))

Evaluando paso a paso:

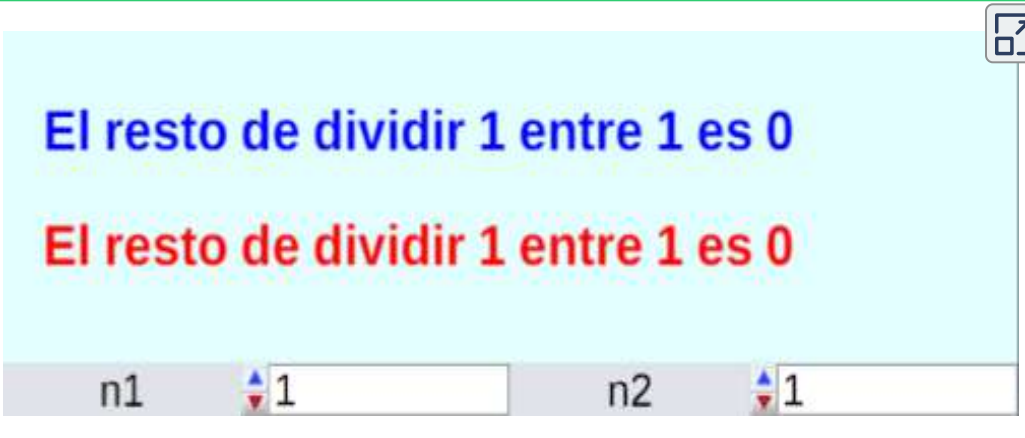
2 * (1 + 3 / 3)

2 * (1 + 1)

2 * 2 = 4

Se incluyen otras dos escenas que utilizan estos operadores.





4.3 Operadores booleanos y su uso en condicionales

Los **operadores booleanos** permiten comparar valores de elementos en Descartes y devuelven **1** si la comparación es verdadera o **0** si es falsa.

Los elementos a comparar pueden ser constantes, variables o incluso valores retornados por funciones.

Cuando la comparación es verdadera, el resultado es 1; si es falsa, el resultado es 0. A continuación,se presentan los operadores booleanos disponibles en Descartes:

- **==**: Compara si los elementos antes y después del operador son iguales. Devuelve 1 si es verdadero y 0 si es falso. Ejemplo: (2==2) = 1, (2==1) = 0. Aunque Descartes permite un solo =, se recomienda usar == para evitar confusión.
- **!=**: Compara si los elementos son distintos. Devuelve 1 si son diferentes y 0 si son iguales. Ejemplo: (2!=2) = 0, (2!=1) = 1.
- **<**: Compara si el elemento de la izquierda es menor que el de la derecha. Devuelve 1 si es verdadero y 0 si es falso. Ejemplo: (1<2) = 1, (2<1) = 0.
- **>**: Compara si el elemento de la izquierda es mayor que el de la derecha. Devuelve 1 si es verdadero y 0 si es falso. Ejemplo: (2>1) = 1, (1>2) = 0.
- **<=**: Compara si el elemento de la izquierda es menor o igual al de la derecha. Ejemplo: (2<=2) = 1, (3<=2) = 0.
- **>=**: Compara si el elemento de la izquierda es mayor o igual al de la derecha. Ejemplo: (3>=3) = 1, (3>=4) = 0.
- **!**: Niega el valor de una condición o valor. Cambia 1 a 0 y 0 a 1. Ejemplo: !(3>4) = 1, !(3<4) = 0, !1 = 0, !0 = 1.

- **|** (or): Separa dos condiciones. Devuelve 1 si al menos una es verdadera. Ejemplo: (2>3)|(3>1) = 1, (2>3)|1 = 1, (2>3)|(3>4) = 0.
- **&** (and): Separa dos condiciones. Devuelve 0 si al menos una es falsa. Ejemplo: (2>3)&(3>1) = 0, (2>3)&1 = 0, (2<3)&(3<4) = 1.

Vamos a ver algunos ejemplos prácticos de su uso:

- **Condición para dibujar un elemento:** Por ejemplo, si queremos que un texto sólo aparezca cuando la variable a sea menor que 100, indicamos en el campo dibujar-si la condición (a<100).
- **Condición para ejecutar una animación o algoritmo:** Si queremos que algo se ejecute mientras se cumpla una condición, escribimos la condición en el campo mientras de la animación. Por ejemplo, (t>=10) hará que la animación permanezca activa mientras el valor de t sea igual o mayor que 10.
- **Condición para ejecutar un evento:** Los eventos se verán más adelante en este capítulo. Por ejemplo, en el campo Condición de un evento podemos poner (b=5). Esto hará que, cuando b sea igual a 5, se realice la acción definida en el evento.
- **Condición en controles:** En los controles se pueden introducir tomas de decisión en su menú acción que están sujetas a condicionales. También se usan para decidir si estos se dibujan o están activos.
- **Dominio de una función** y **variables de forma condicionada**.
- En muchas ocasiones, el desarrollo de una escena depende de las condiciones establecidas. Para realizar comparaciones usamos:
    - **Operadores matemáticos:** >, <, =, # (mayor, menor, igual, distinto).
    - **Operadores lógicos:** &, |, ~ (Y, O, NO).

Algunos ejemplos de condiciones y operadores:

- Que a sea mayor que 3: (a>3)
- Que b sea mayor o igual que -3: (b>=-3)
- Que a sea menor o igual que 3: (a<=3)
- Que a sea igual a b: (a=b) o (a==b)
- Que a sea menor que 3: (a<3)
- Que b sea menor o igual que -3: (b<=-3)
- Que c sea distinto de 5: (c#5) o (c!=5)
- Que a sea mayor que 3 y menor que 10: (a>3)&(a<10)
- Que a sea mayor o igual que 3 y b distinto de 3: (a>=3)&(b#3)
- Que a sea igual a 3 o igual a 10: (a=3)|(a=10)
- Que a no sea menor o igual que 8: ~(a<=8)
- Que a no sea 5 ni 8: (a#5)&(a#8)

Un operador especialmente útil para decisiones condicionadas es el **operador tipo pregunta**, cuya estructura es:

- P = (condición) ? a : b

En esta estructura, P valdrá a si se cumple la condición y b si no se cumple. Los valores a y b pueden ser constantes o variables, numéricas o alfanuméricas (texto).

Ejemplos:

- P = (nota >= 5) ? 'Aprobado' : 'Suspenso'. Esto significa que si la variable nota es mayor o igual que 5, P tendrá el valor 'Aprobado', y en caso contrario 'Suspenso'.
- L = (a < 100) ? 10 : 8. Esto significa que si la variable a es menor que 100, L tomará el valor 10; en caso contrario, L tomará el valor 8.

Los operadores tipo pregunta también pueden anidarse según la estructura:

- P = (condición1) ? a : ((condición2) ? b : ((condición3) ? c : d))

En este caso, P valdrá a si se cumple la condición 1, b si se cumple la condición 2, c si se cumple la condición 3 y d si no se cumple ninguna condición. Los valores a, b, c y d pueden ser constantes o variables, numéricas o alfanuméricas. Se pueden anidar tantas condiciones como se desee, teniendo cuidado de cerrar tantos paréntesis como se abran.

Ejemplo de operador tipo pregunta anidado:

- P = (nota < 5) ? 'Suspenso' : (((nota >= 5) & (nota < 7)) ? 'Aprobado' : (((nota >= 7) & (nota < 9)) ? 'Notable' : ((nota >= 9) ? 'Sobresaliente' : 'No calificado')))

Esto significa:

- Si nota es menor que 5, P será 'Suspenso'.
- Si nota es mayor o igual que 5 y menor que 7, P será 'Aprobado'.
- Si nota es mayor o igual que 7 y menor que 9, P será 'Notable'.
- Si nota es mayor o igual que 9, P será 'Sobresaliente'.
- Si nota no cumple ninguna condición, P será 'No calificado'.

Algunas veces es necesario usar condicionales para determinar si se llama a una función o no. En Descartes no existe una forma explícita de indicar que una función se ejecute sólo si se cumple una condición y que, de lo contrario, no se ejecute. Esto es común cuando se quiere ejecutar funciones de manera condicionada.

Para resolver este problema, se pueden usar **variables mudas**. Estas variables no tienen otro propósito que permitir asociar la condicional a una función, haciendo posible que la función se ejecute únicamente cuando se cumpla la condición establecida.

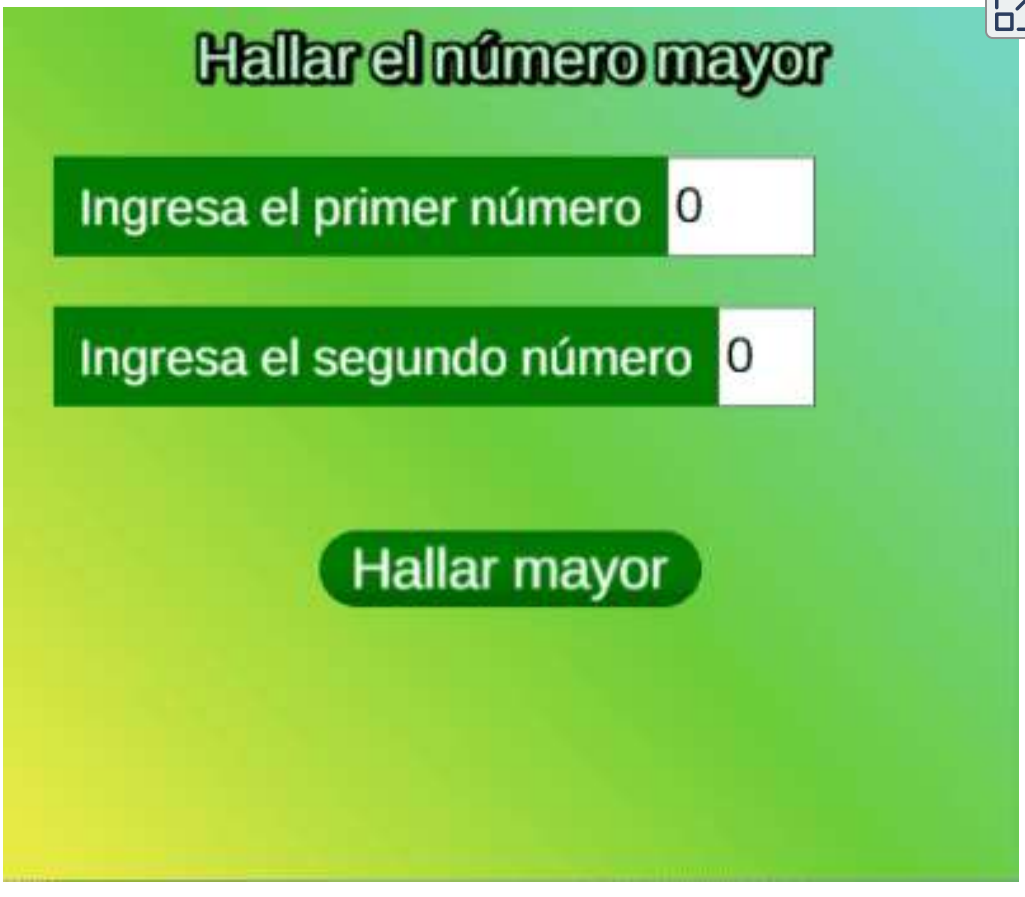
En la siguiente tabla se recogen los principales operadores de Descartes.

Operador	Descripción
(	Paréntesis izquierdo
)	Paréntesis derecho
==	Igualdad, operador binario que da un resultado booleano
!=	Desigualdad, operador binario que da un resultado booleano
#	Desigualdad, operador binario que da un resultado booleano
|	Operador binario de disyunción equivalente a OR (O)
&	Operador binario de conjunción equivalente a AND (y)
>	Mayor que, operador binario que da un resultado booleano
<	Menor que, operador binario que da un resultado booleano
>=	Mayor o igual que, operador binario que da un resultado booleano
<=	Menor o igual que, operador binario que da un resultado booleano
+	Signo más, operador binario de suma
-	Signo menos: binario de resta o unario de cambio de signo
*	Por, operador binario de multiplicación
/	Entre, operador binario de división
^	Operador binario de exponenciación (a^b = a^b)
%	Módulo, operador binario, residuo de una división
~	Operador unario booleano de negación
?	Pregunta, sólo para asignaciones condicionales (P)?a:b
:	Separador, sólo para asignaciones condicionales (P)?a:b

En la siguiente escena se muestra el funcionamiento de los operadores de comparación y de los operadores lógicos. Al modificar los valores de las variables A y B mediante sus correspondientes pulsadores, se puede observar cómo cambian los resultados de las distintas comparaciones.

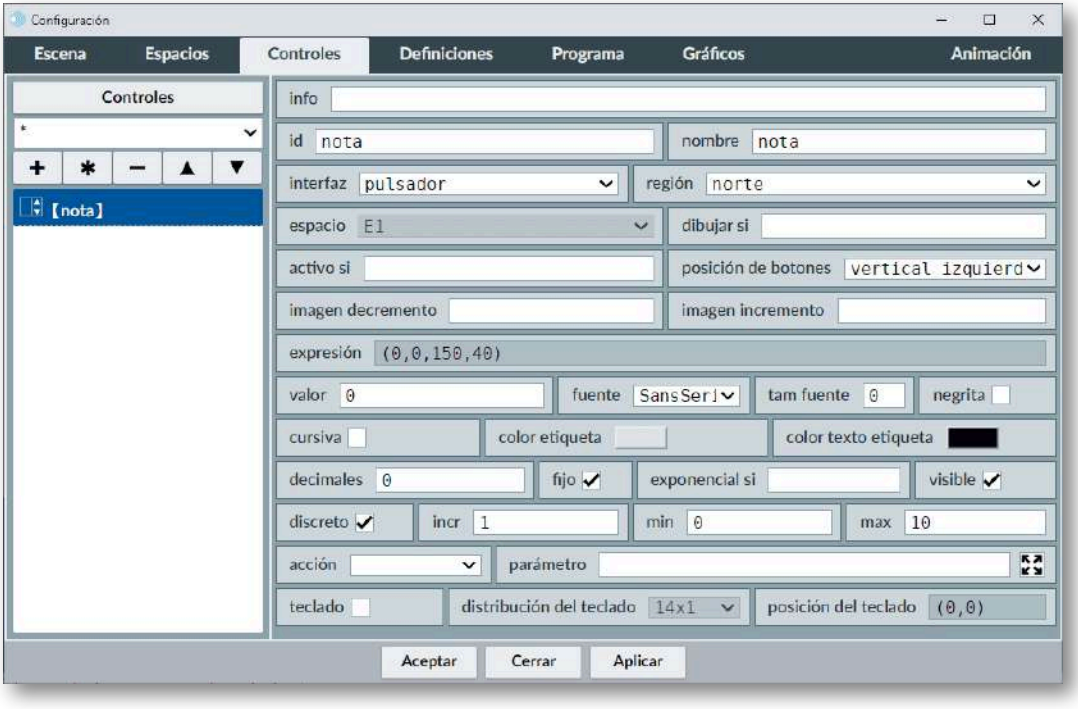


En la siguiente escena se utiliza el operador tipo pregunta y se incluye un vídeo explicativo en el que se muestra paso a paso cómo se ha creado.


Hallar el número mayor
Ingresa el primer número
0
Ingresa el segundo número
0
Hallar mayor


Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Título de la página
TITULO
ancho
400
alto
350
botón créditos
botón config
botón inicio
botón limpiar
filas al norte
0
filas al sur
0
ancho oeste
150
ancho este
150
alto filas
40
signo decimal
idioma
español
mostrar región exterior
expandir escena
escalar
imagen del cargador
Hallar el número maye
Ingresa el primer número
0
Ingresa el segundo número
0
Hallar mayor
0:00

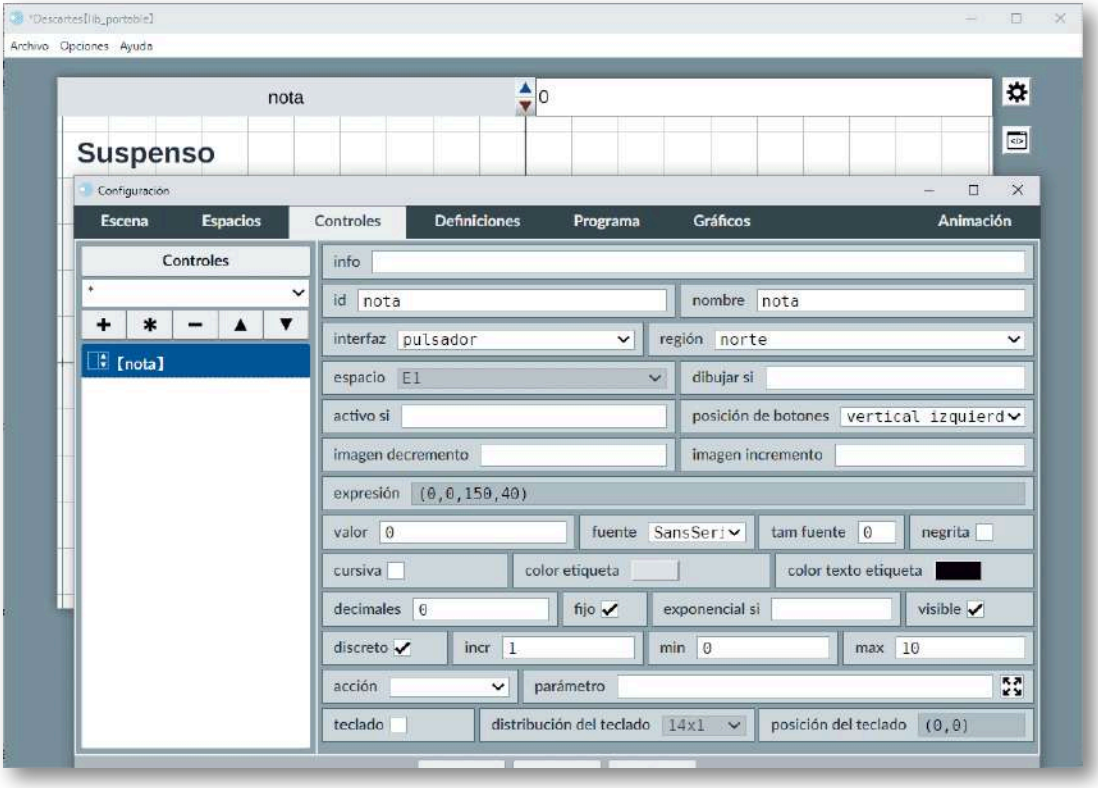
A continuación, detallaremos paso a paso el funcionamiento del operador condicional tipo pregunta anidado para ilustrar su funcionamiento. Para ello, comenzamos definiendo el control numérico nota, de tipo pulsador, situado en la zona norte de la escena. Este control no admitirá decimales, tendrá un valor mínimo de 0, un valor máximo de 10 y un incremento de 1, lo que permitirá modificar la calificación de forma sencilla y observar cómo cambia el resultado en función del valor seleccionado.



A continuación, incluimos una variable dentro de un elemento de tipo texto, situado en el selector **Gráfico**, cuyo valor será el siguiente:

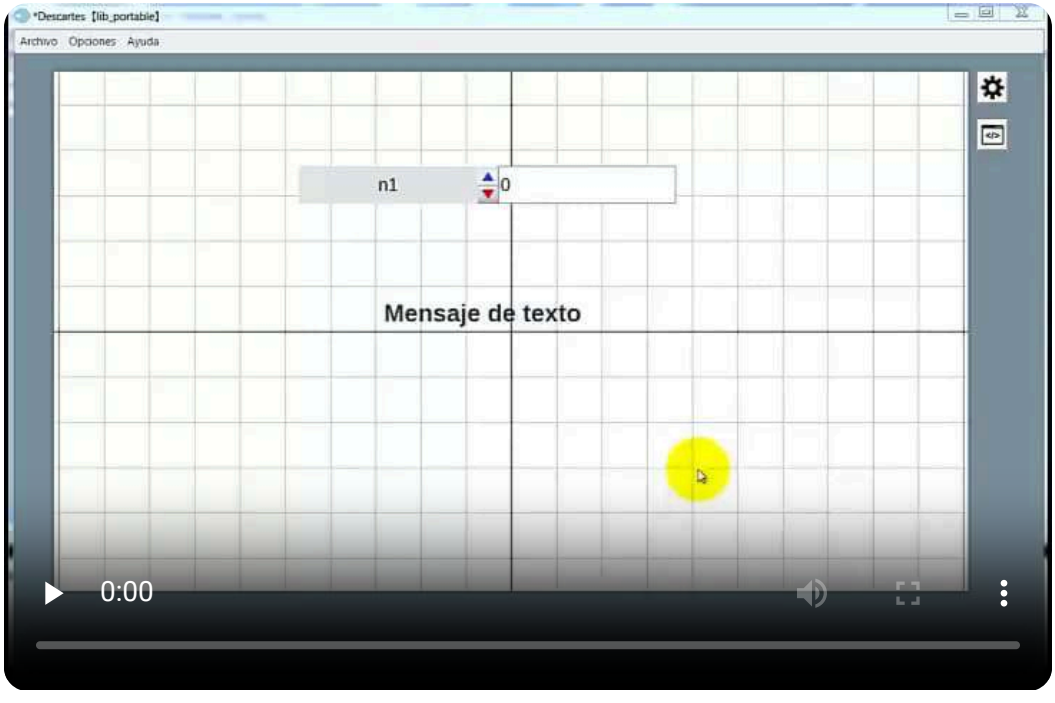
[(nota<5)?'Suspenso':(((nota>=5)&(nota<7))?'Aprobado':(((nota>=7)&(nota<9))?'Notable':((nota>=9)?'Sobresaliente':'No calificado')))].

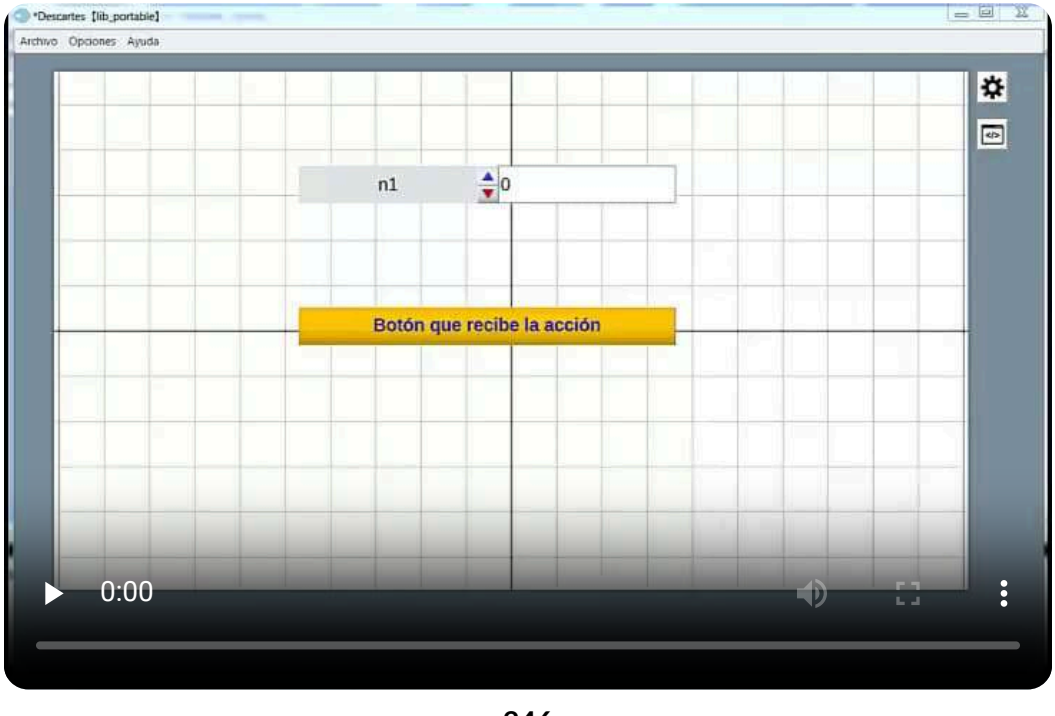
El valor de esta expresión debe escribirse entre corchetes, ya que este es el formato que se utiliza para insertar variables dentro de los textos.


*Descartes[lib_portable]
Archivo Opciones Ayuda
nota
0
Suspenso
Configuración
Escena Espacios Controles Definiciones Programa Gráficos Animación
Controles
[nota]
info
id nota
nombre nota
interfaz pulsador
región norte
espacio E1
dibujar si
activo si
posición de botones vertical izquierd
imagen decremento
imagen incremento
expresión (0,0,150,40)
valor 0
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 0
fijo
exponencial si
visible
discreto
incr 1
min 0
max 10
acción
parámetro
teclado
distribución del teclado 14x1
posición del teclado (0,0)

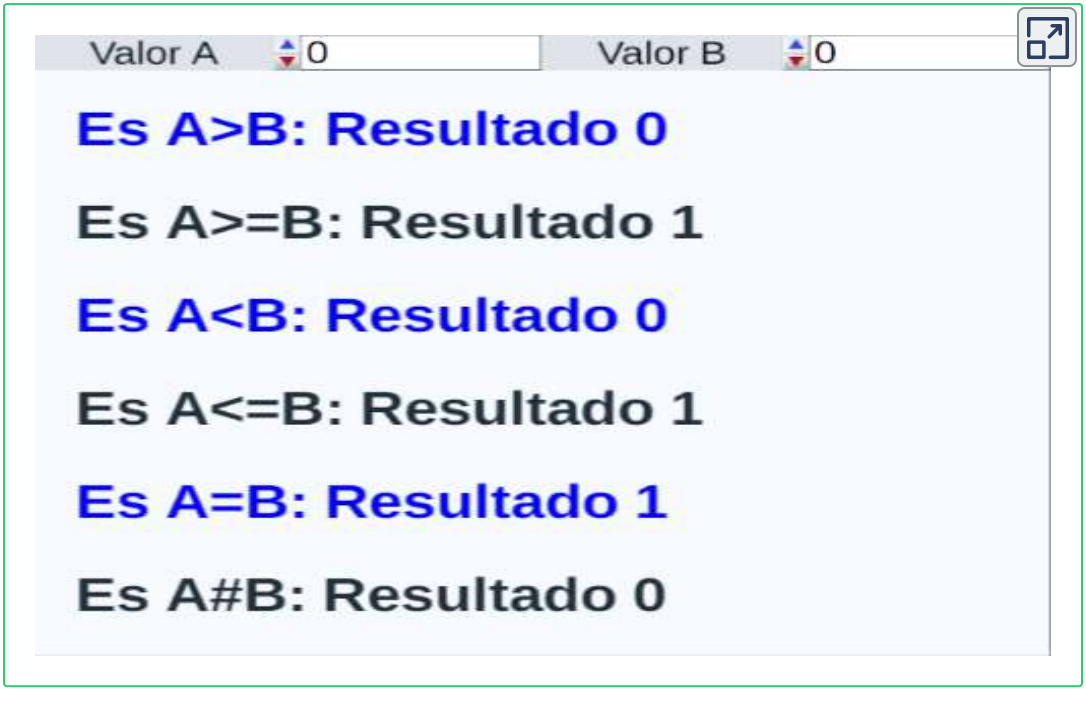

nota
0
Suspenso

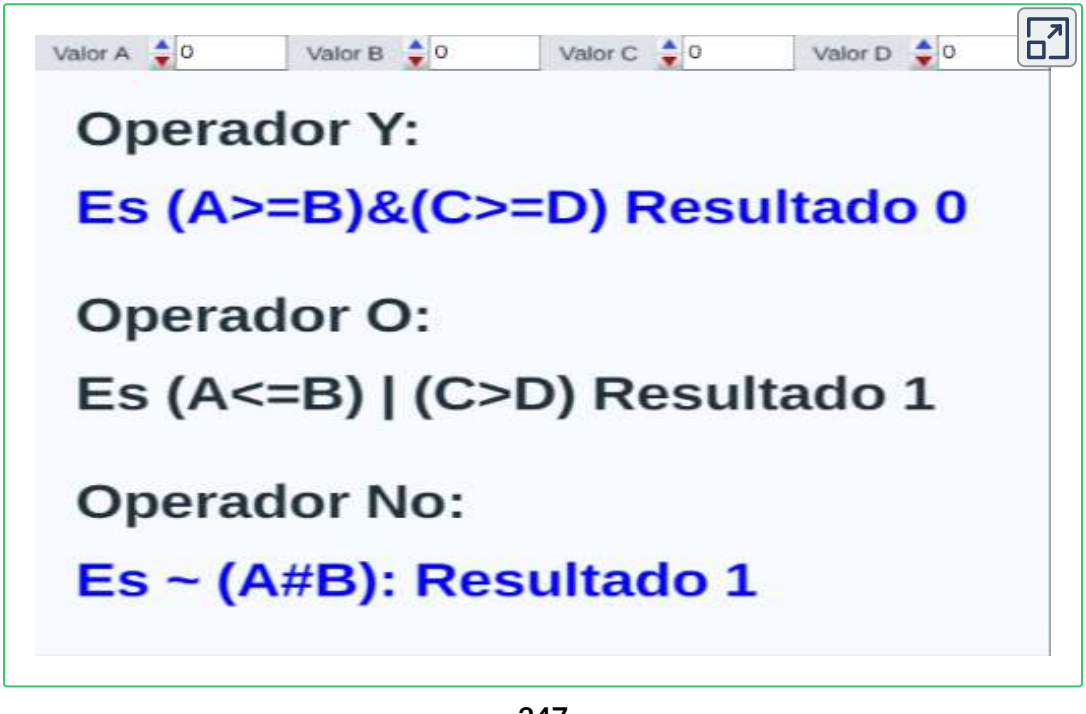
Se incluyen dos vídeos sobre operaciones con condicionales.





Dos ejemplos más de escenas con operadores booleanos son:





4.4 Operadores reservados

Existen varios identificadores o palabras reservadas que tienen un significado especial cuando se utilizan en una escena. Es decir, al emplearlos, ya cuentan con una utilidad y un comportamiento definidos por defecto.

Por ejemplo, en los espacios 2D los identificadores reservados son escala, O.x y O.y. En los espacios 3D, las variables intrínsecas son escala, observador, rot.y y rot.z.

Estas variables deben ir siempre precedidas por el identificador del espacio seguido de un punto. Es decir, su forma completa será: <id>.escala, <id>.O.x, <id>.O.y, <id>.observador, <id>.rot.y y <id>.rot.z, donde <id> representa el identificador del espacio correspondiente.

Los operadores y variables reservadas de Descartes permiten acceder a información interna de los espacios, controles y otros elementos de la escena, así como modificar su comportamiento. A continuación, se describen los más importantes:

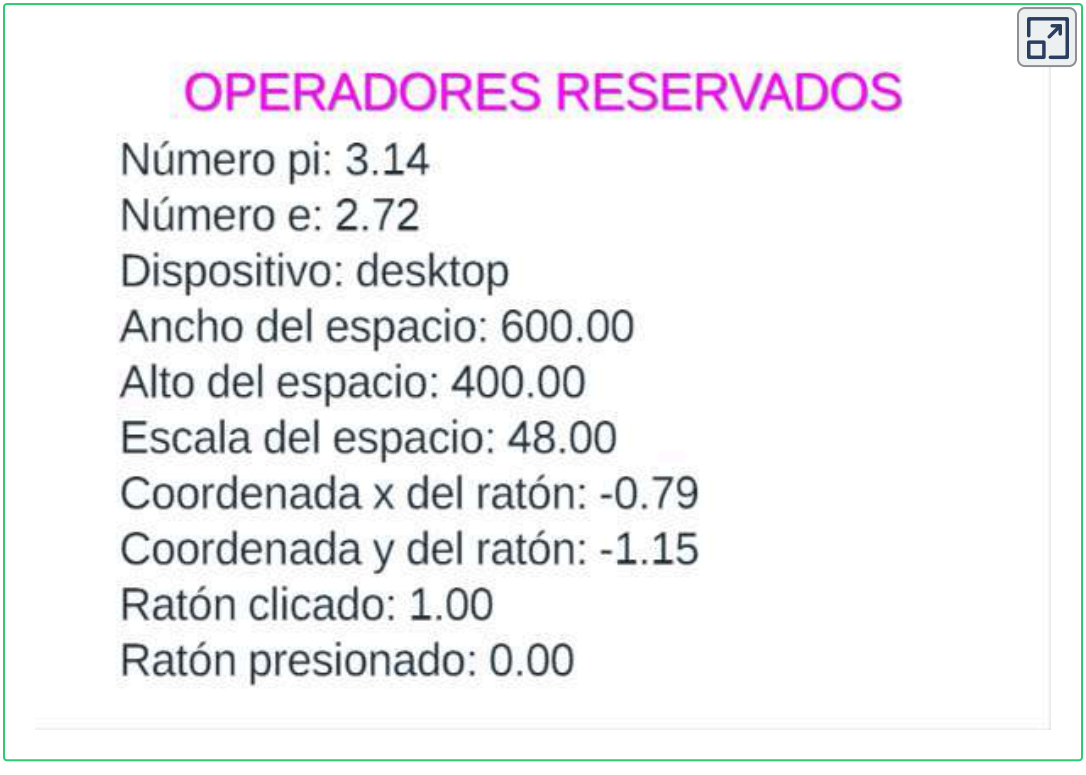
- **<Nombre del espacio>._w**: Permite conocer el ancho del espacio en píxeles (width). Por ejemplo, Esp._w indica cuántos píxeles de ancho tiene el espacio Esp.
- **<Nombre del espacio>._h**: Permite conocer el alto del espacio en píxeles (height). Por ejemplo, Esp._h indica la altura en píxeles del espacio Esp.
- **<Nombre del espacio>.Ox**: Indica el desplazamiento horizontal del origen en píxeles (offset en x). Por defecto, el origen está en el centro del espacio, pero puede desplazarse asignando valores positivos (derecha) o negativos (izquierda). Es equivalente al campo O.x del selector Espacio.

- **<Nombre del espacio>.Oy**: Indica el desplazamiento vertical del origen en píxeles (offset en y). Un valor negativo desplaza el origen hacia arriba y uno positivo hacia abajo. Es equivalente al campo O.y del selector Espacio.
- **<Nombre del espacio>.escala**: Permite conocer o modificar la escala del espacio (número de píxeles por unidad).
- **<Nombre del espacio>.rot.y**: Válida sólo para espacios tridimensionales. Guarda la rotación en grados alrededor del eje y y permite asignar una perspectiva inicial.
- **<Nombre del espacio>.rot.z**: Válida sólo para espacios tridimensionales. Guarda la rotación en grados alrededor del eje z y permite definir la perspectiva inicial.
- **<Nombre del espacio>.mouse_x**: Devuelve la coordenada horizontal del mouse en el plano cartesiano cuando el botón está presionado. Sólo se actualiza mientras el botón está oprimido.
- **<Nombre del espacio>.mouse_y**: Devuelve la coordenada vertical del mouse cuando el botón está presionado.
- **<Nombre del espacio>.mouse_clicked**: Vale 1 cuando el botón del mouse ha sido soltado tras hacer clic en el espacio; en caso contrario vale 0. Es posible configurar el espacio para que las variables del mouse se actualicen incluso sin mantener presionado el botón, activando la opción correspondiente en el selector Espacio. Sin embargo, no se recomienda abusar de esta funcionalidad, ya que incrementa el número de cálculos y puede ralentizar la escena.
- **<Nombre del espacio>.mouse_pressed**: Vale 1 mientras el botón izquierdo del mouse está presionado y 0 cuando se suelta.
- **<Identificador del control gráfico>.x**: Permite consultar o modificar la coordenada horizontal del control gráfico.

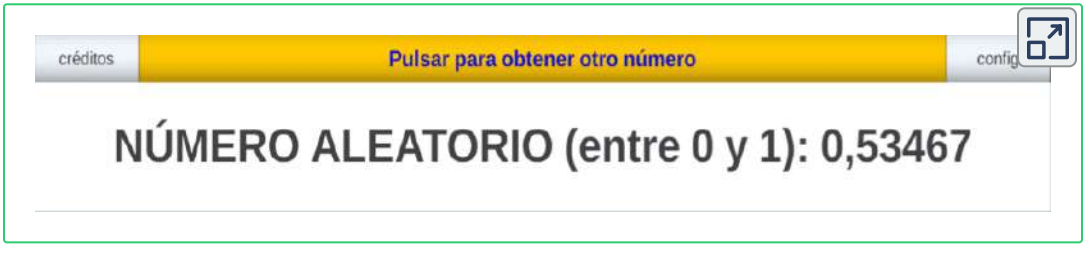
- **<Identificador del control gráfico>.y**: Permite consultar o modificar la coordenada vertical del control gráfico.
- **<Identificador del control gráfico>.activo**: Indica si el control está activo (1) o no (0). Un control permanece activo hasta que se hace clic fuera de él.
- **<Identificador del control de texto>.activo**: Indica si el control de texto está activo (1) o no (0).
- **<Identificador del control de audio o vídeo>.currentTime**: Guarda el tiempo actual, en segundos, del audio o vídeo. Para mostrarlo en un texto es necesario que exista una animación activa que refresque continuamente el valor.
- **<Identificador del menú>.txt**: Devuelve el texto correspondiente a la opción seleccionada en el menú.
- **<Identificador del vector>.long**: Devuelve el número de elementos del vector. Por ejemplo, si v1 tiene cinco entradas, v1.long devuelve 5.
- **<Identificador de la matriz>.filas**: Devuelve el número de filas de la matriz.
- **<Identificador de la matriz>.columnas**: Devuelve el número de columnas de la matriz.
- **URL.<etiqueta>**: Permite recuperar el valor asociado a una etiqueta incluida en la dirección URL donde etiqueta es un nombre de etiqueta que se da en la ruta, junto con su valor asociado. Esta variable sólo tiene sentido si la escena ha de ser abierta en un servidor.
- **rnd**: Variable que genera un número aleatorio real entre 0 (incluido) y 1 (sin llegar nunca a alcanzarlo) cada vez que se utiliza.
- **pi**: Variable numérica predefinida que representa una aproximación al valor de π. Puede utilizarse en cálculos o mostrarse en pantalla.

- **e**: Variable numérica predefinida que representa una aproximación al número de Euler o constante de Napier.
- **device**: Variable de tipo texto que indica el dispositivo en el que se ejecuta la escena. Sus posibles valores son desktop (ordenador), tablet (tableta) y mobile (teléfono móvil).
- **dispositivo**: Variable equivalente a device, que cumple la misma función.
- **_NUM_MAX_ITE_ALG_**: Variable que establece el número máximo de iteraciones permitidas en un algoritmo con repetición. Su valor por defecto es 100000, lo que evita ciclos infinitos que puedan bloquear la escena. Si es necesario, puede asignársele un valor mayor para permitir un número superior de iteraciones en determinadas simulaciones.
- **Literales especiales en expresiones gráficas:**
    - **x** y **y**: Representan la abscisa y la ordenada de los puntos en el plano para objetos gráficos tipo "ecuación". También se utilizan en las expresiones que definen las restricciones de los controles gráficos.
    - **n**: Variable de una sucesión en el objeto gráfico "sucesión". Solo toma valores enteros.
    - **u** y **v**: Parámetros en las ecuaciones paramétricas de superficies 3D. Sus valores son reales en el intervalo [0,1].
    - **x, y, z**: Coordenadas de los puntos de una superficie en 3D. Se expresan paramétricamente en términos de u y v para representar todos los puntos de la superficie.
    - **t**: Usada para definir ecuaciones paramétricas de curvas en 2D o 3D. A diferencia de u y v, puede reemplazarse por cualquier otra letra o palabra, y su rango no está limitado al intervalo [0,1].
    - **s**: Usada para definir familias de gráficos en 2D. No es una variable reservada, el autor puede cambiarla libremente.

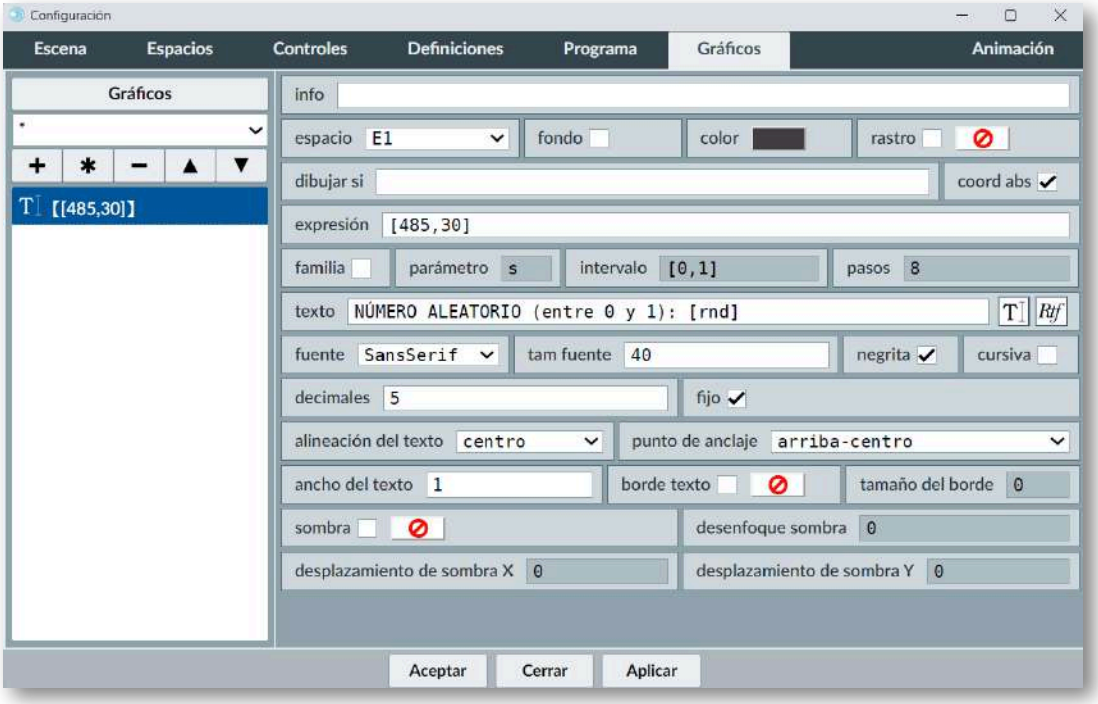
Incluimos una escena que muestra ejemplos prácticos del funcionamiento de diferentes operadores reservados.



A continuación, se incluye una escena donde se muestra cómo obtener números aleatorios con Descartes.



Mostramos una captura de la inclusión del operador reservado **rnd** en un elemento de texto. El control añadido únicamente reinicia la escena, lo que provoca que se genere y muestre un nuevo número aleatorio.



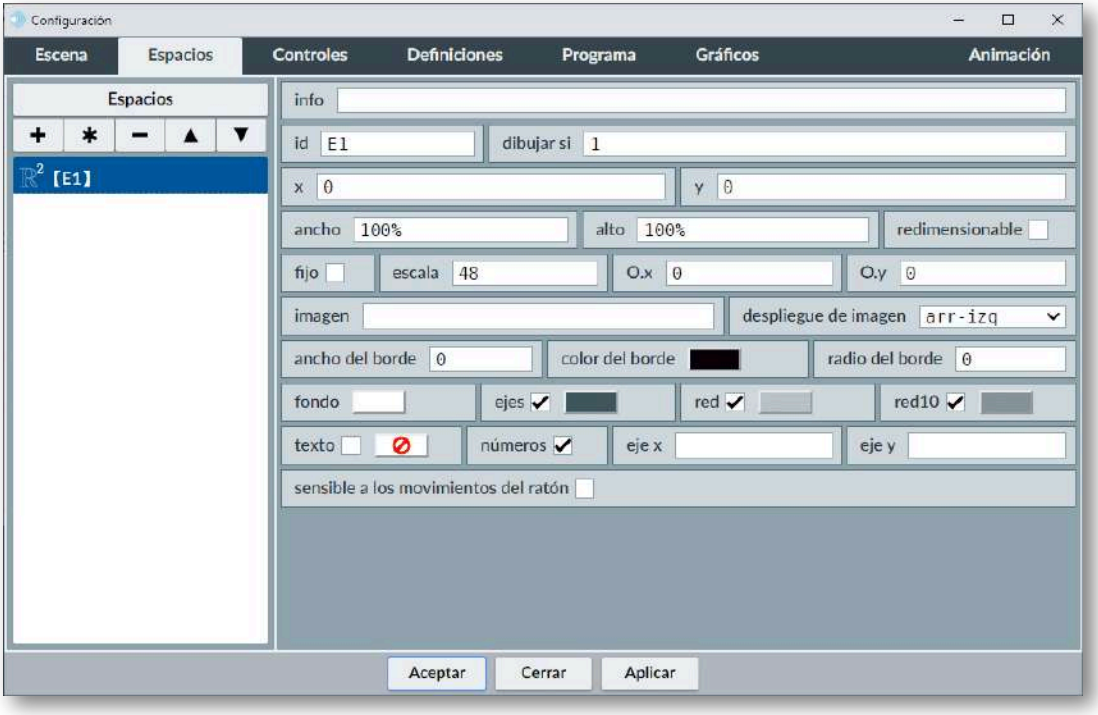
Pondremos otro ejemplo de operadores reservados, en este caso **escala**. Esta palabra se reserva para controlar la escala o "zoom" con la que se representa un espacio en la escena. Su función principal es determinar cuánto se amplía o reduce la visualización del espacio:

- A mayor valor de **escala**, más ampliado se mostrará el espacio, haciendo que los elementos parezcan más grandes.
- A menor valor de **escala**, más reducido se verá el espacio, mostrando más contenido en el mismo área.

El operador **escala** debe escribirse siempre detrás del nombre o identificador del espacio, separado por un punto. Por ejemplo: E1.escala.

En nuestra escena de ejemplo, vamos a definir un espacio llamado **E1** en dos dimensiones, con ejes, rejillas y números visibles. Para este espacio, asignamos al campo escala el valor 48, que es aproximadamente el valor predeterminado que viene por defecto.

Este ejemplo permite ver de forma práctica cómo modificar el "zoom" de un espacio y cómo la visualización cambia según el valor de la escala, siendo útil para ajustar la presentación de gráficos, coordenadas y objetos dentro del espacio.

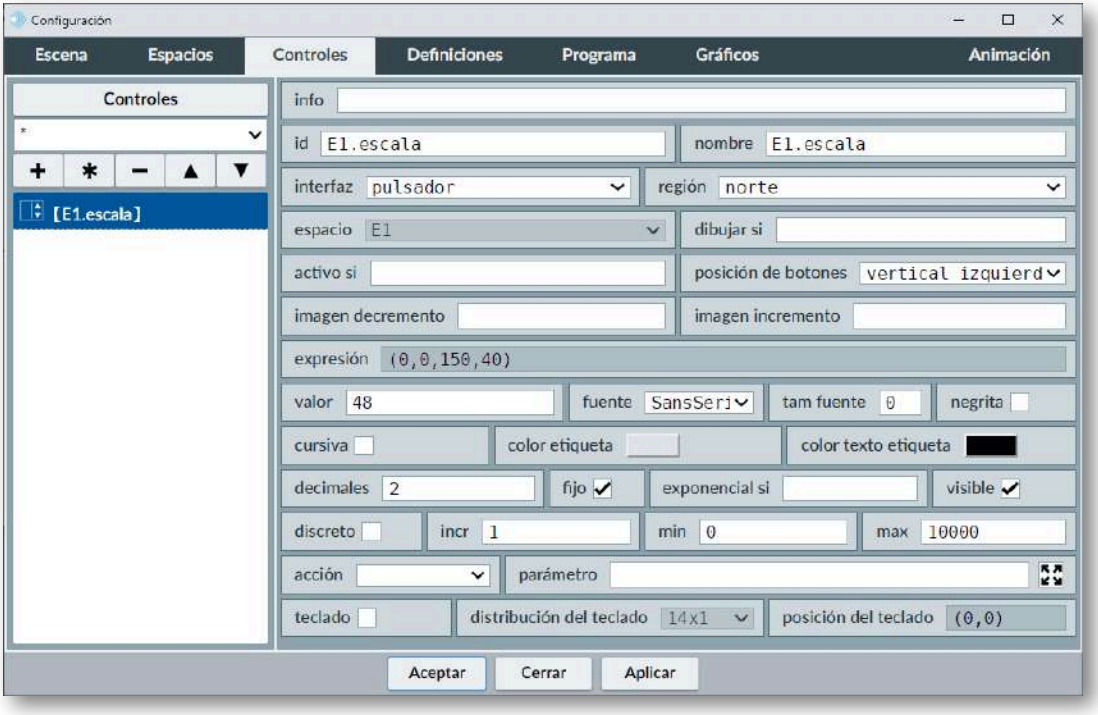


Introducimos un control numérico de tipo **pulsador** en la zona norte de la escena. Su nombre será E1.escala, es decir, el nombre del espacio seguido de un punto y el término reservado para la escala.

Las propiedades del control serán:

- Valor por defecto: 48
- Incremento: 1
- Valor mínimo: 0
- Valor máximo: 10000

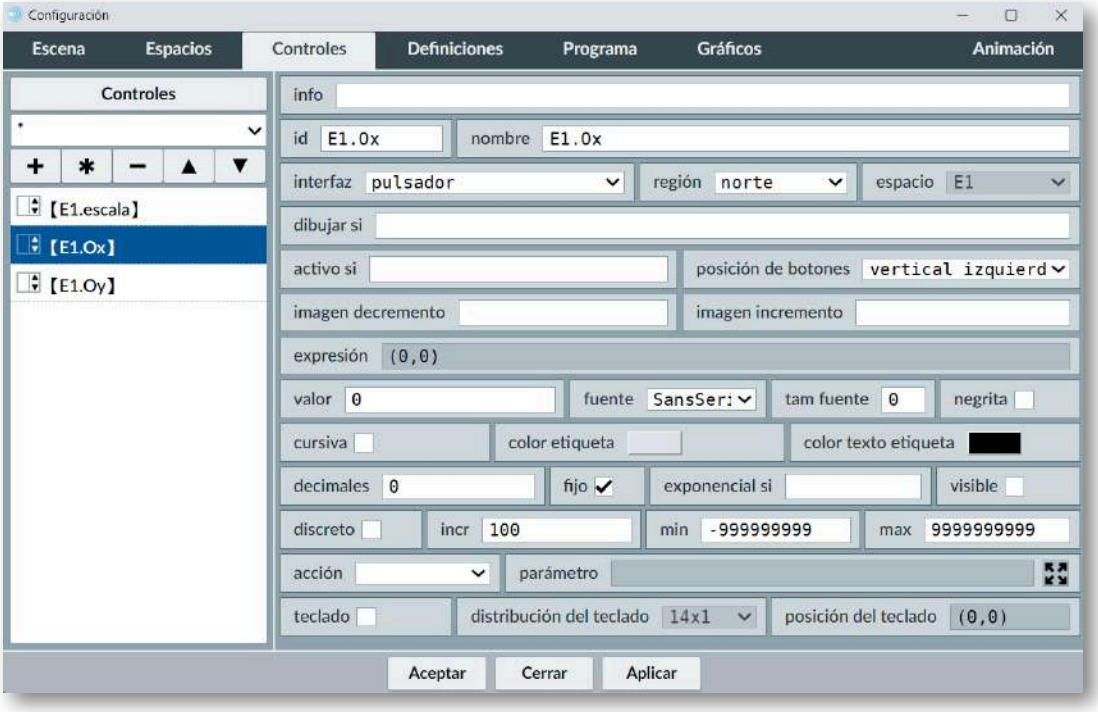
Este control permitirá modificar de manera interactiva la escala del espacio **E1**, ajustando el "zoom" de la visualización en tiempo real.

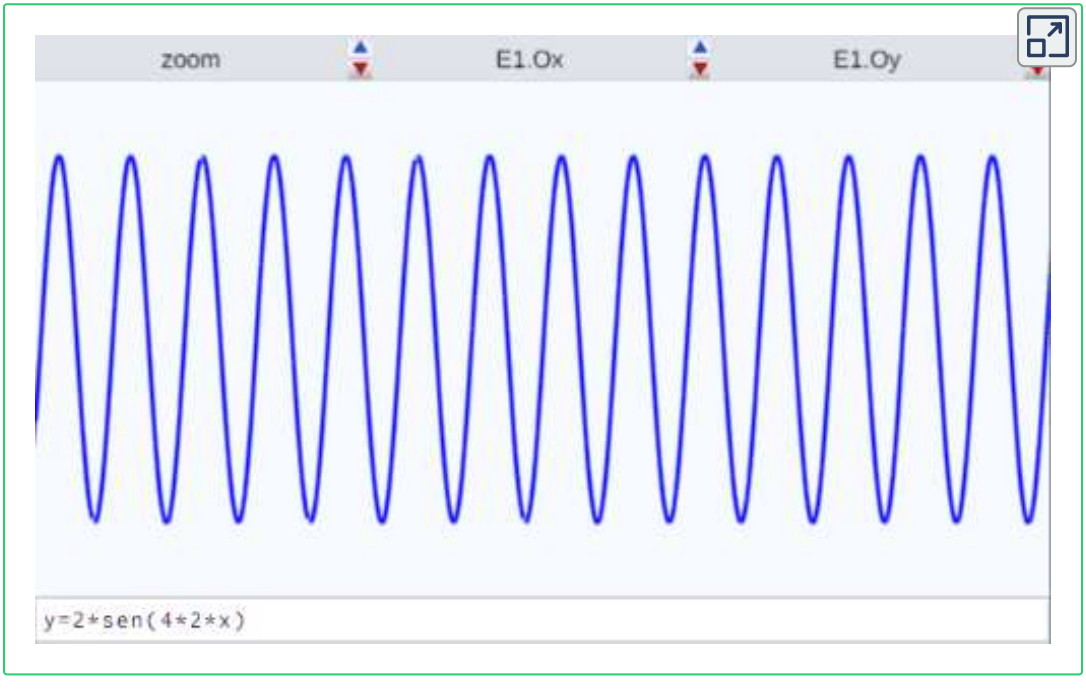


Cambiando el valor del pulsador se modificará la escala de la escena.



En la siguiente escena incluimos otros operadores reservados.





4.5 Funciones comunes intrínsecas de Descartes

Una función es el resultado de realizar ciertas operaciones matemáticas o de otra naturaleza, que pueden involucrar una o más variables. El valor resultante de estas operaciones determina el valor de la función.

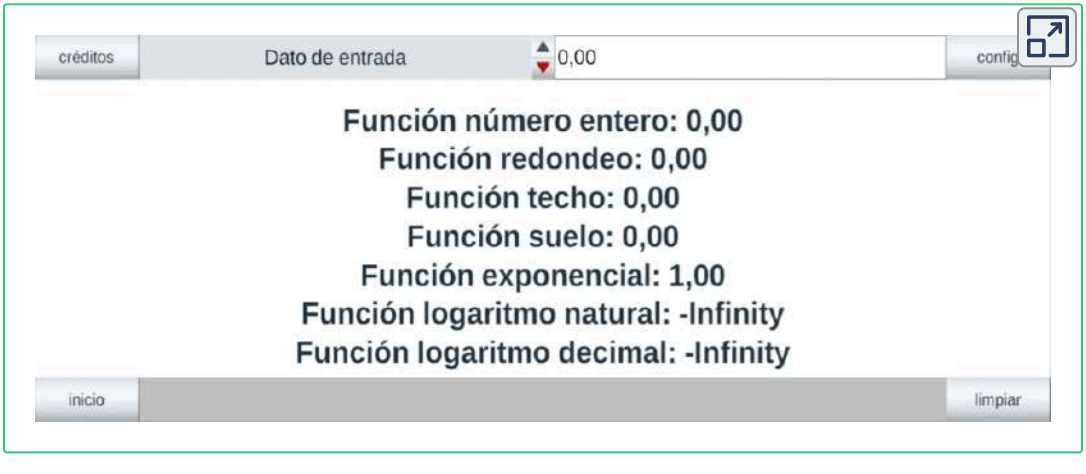
A continuación, se presenta una lista alfabética de las funciones que Descartes comparte con otros lenguajes de programación, como por ejemplo JavaScript. La mayoría son funciones matemáticas, y aunque en algunos casos el nombre pueda variar, la acción que realizan es equivalente.

- **abs(x)**: Devuelve el valor absoluto de x. Ejemplo: abs(-2) → 2.
- **acos(x)**: Arcocoseno de x en radianes. Ejemplo: acos(-1) → 3.141… rad (180°).
- **_AnchoDeCadena_(string,font,style,size)**: Calcula la longitud en píxeles de una cadena.
- **aleatorio(a,b)**: Valor aleatorio en [a,b) o [b,a) si a>b. Equivalente a rnd si a=0 y b=1.
- **asin(x)**: Arcoseno de x en radianes. Ejemplo: asin(1) → 1.570… rad (90°).
- **atan(x)**: Arcotangente de x en radianes. Ejemplo: atan(1) → 0.785… rad (45°).
- **ceil(x)**: Entero inmediato superior de x. Equivalente a techo(x).
- **charAt(string,n)**: Devuelve el carácter en la posición n de la cadena. Equivalente a _letraEn_(string,n).
- **cos(x)**: Coseno de x en radianes.
- **cosh(x)**: Coseno hiperbólico de x.
- **cot(x)**: Cotangente de x en radianes.
- **csc(x)**: Cosecante de x en radianes.

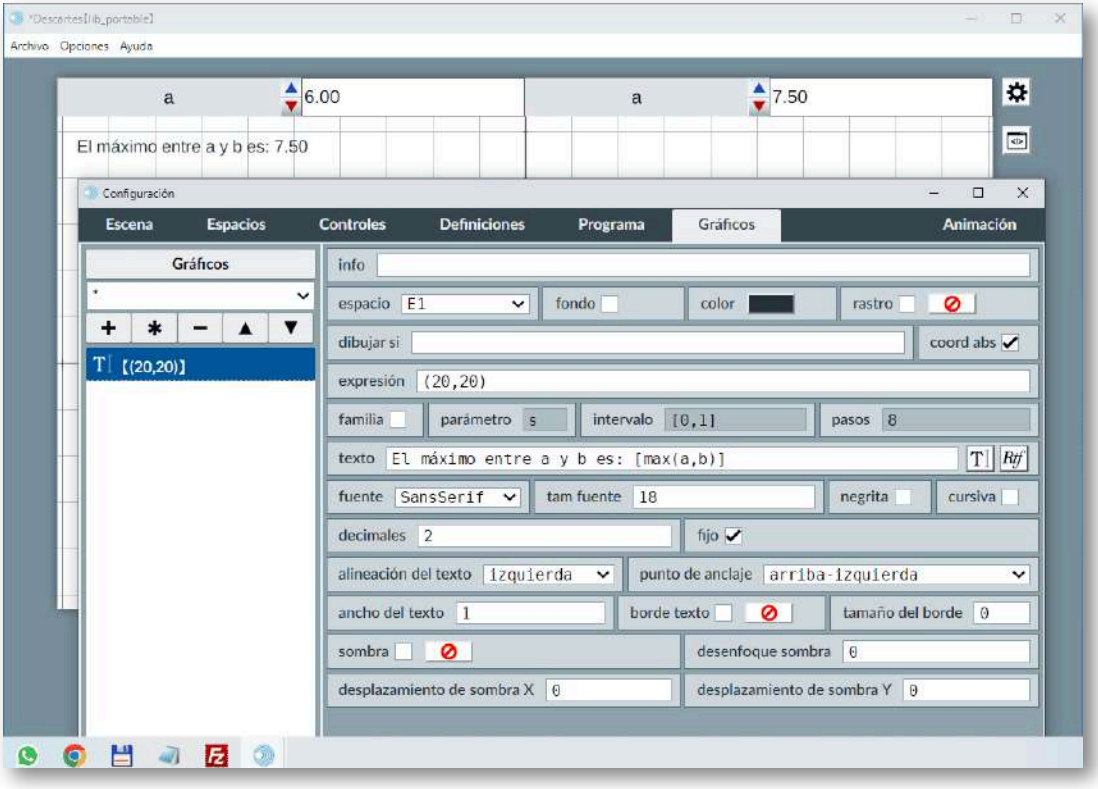
- **indexOf(string,str)**: Índice de la primera aparición de str en string. Equivalente a _índiceDe_(string,str).
- **ent(x)**: Trunca los decimales de x.
- **exp(x)**: Exponencial neperiana e^x.
- **floor(x)**: Entero inmediato inferior de x. Equivalente a piso(x).
- **lastIndexOf(string,str)**: Índice de la última aparición de str en string.
- **_letraEn_(string,n)**: Devuelve el carácter en la posición n de la cadena (inicio en 0).
- **_longitud_(string)**: Equivalente a strLength(string).
- **log(x)**: Logaritmo natural de x.
- **log10(x)**: Logaritmo base 10 de x.
- **logbase(a,x)**: Logaritmo en base a de x, equivalente a log10(x)/log10(a).
- **max(a,b)**: Devuelve el mayor de a y b.
- **min(a,b)**: Devuelve el menor de a y b.
- **piso(x)**: Equivalente a floor(x).
- **raíz(x)**: Equivalente a sqrt(x).
- **random(a,b)**: Equivalente a aleatorio(a,b).
- **redondeo(x)**: Equivalente a round(x).
- **replace(string,str1,str2)**: Sustituye todas las apariciones de str1 por str2 en string.
- **round(x)**: Redondea x al entero más cercano. Equivalente a floor(x+0.5).
- **sec(x)**: Secante de x en radianes.
- **sen(x)**: Equivalente a sin(x).
- **sgn(x)**: Devuelve +1 si x es mayor que 0 y -1 si x es menor que 0.

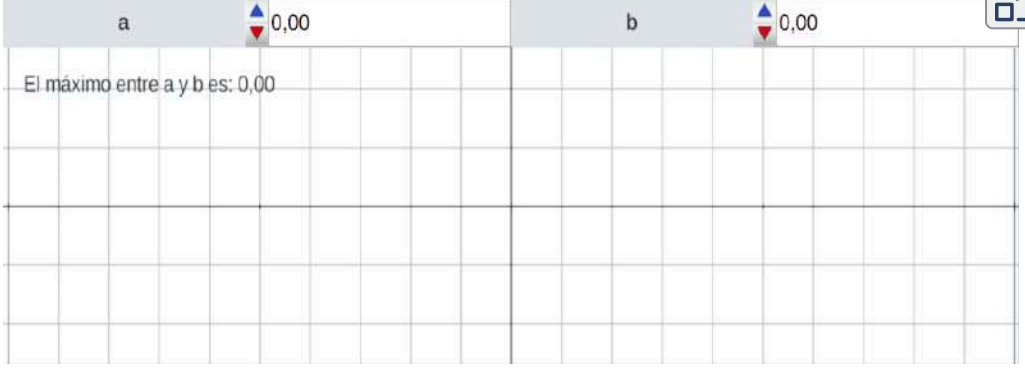
- **sin(x)**: Seno de x en radianes.
- **sinh(x)**: Seno hiperbólico de x.
- **sqr(x)**: Cuadrado de x.
- **sqrt(x)**: Raíz cuadrada de x.
- **strLength(string)**: Devuelve la longitud de la cadena en caracteres.
- **substring(string,a,b)**: Devuelve la subcadena desde índice a (inclusive) hasta b (exclusivo). Equivalente a _subcadena_(string,a,b).
- **tan(x)**: Tangente de x en radianes.
- **tanh(x)**: Tangente hiperbólica de x.
- **toFixed(x,n)**: Devuelve x como cadena con n decimales. Para operar como número, usar _Num_().
- **toLowerCase(string)**: Convierte todas las letras a minúsculas.
- **toUpperCase(string)**: Convierte todas las letras a mayúsculas.
- **trim(string)**: Elimina espacios al inicio y final de la cadena.

A continuación, se incluye una escena donde se emplean varias funciones de Descartes.

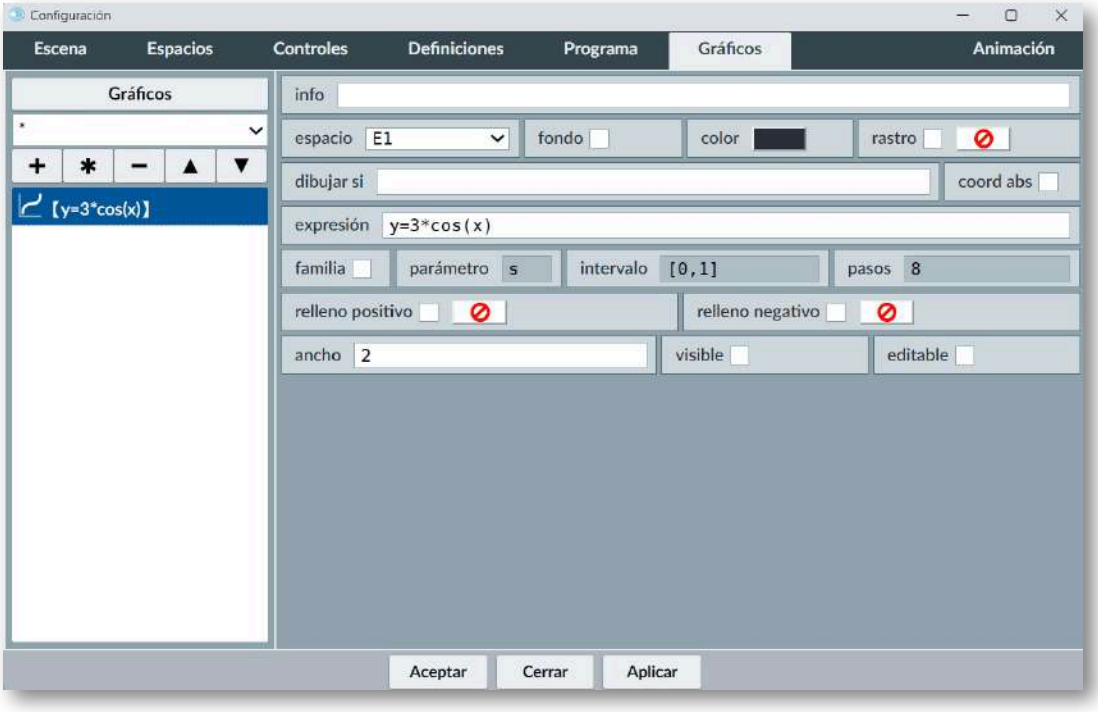


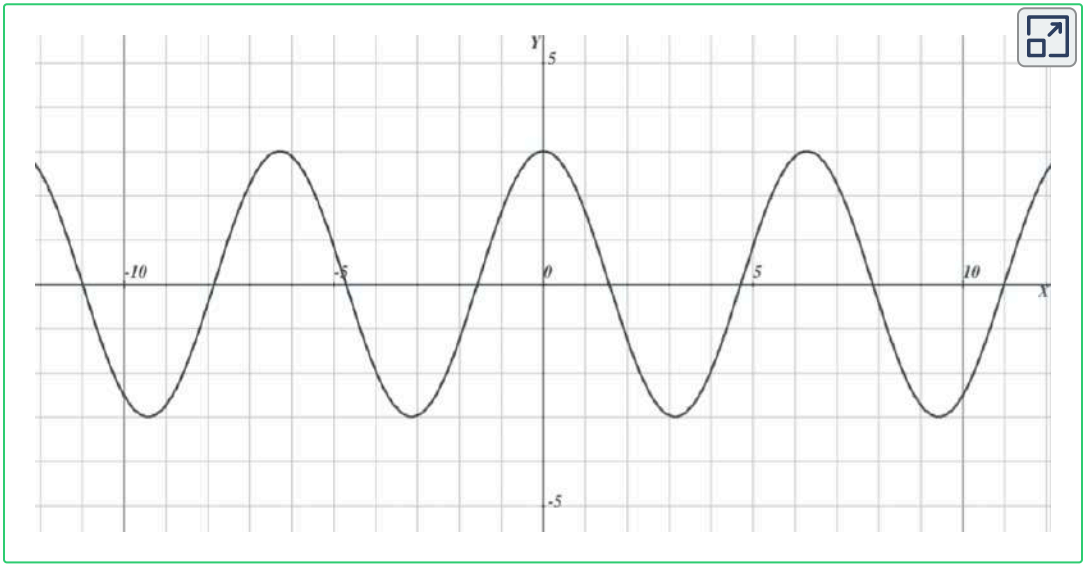
Presentamos otro ejemplo con la función **máximo** max(a,b). Primero, definimos dos controles numéricos de tipo **pulsador** en la zona norte, llamados a y b. Luego, incluimos un texto donde escribimos: "El máximo entre a y b es: [max(a,b)]".





Las funciones también pueden representarse gráficamente, como veremos más adelante. En este caso, hemos representado en un espacio bidimensional (2D) la función y = 3·cos(x).





4.6 Funciones propias del lenguaje de Descartes

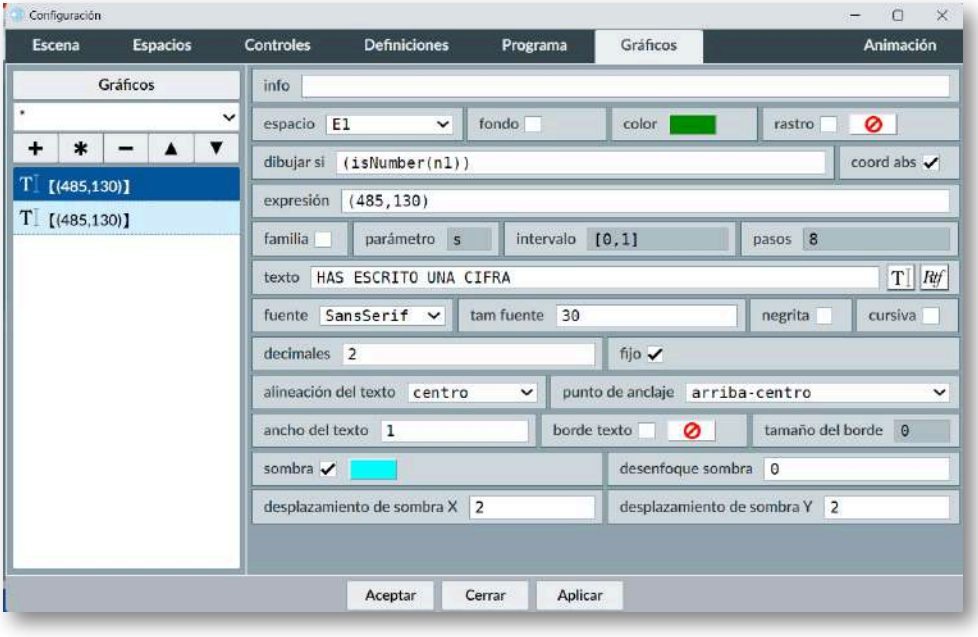
Existen funciones que son propias de Descartes y que no son habitualmente utilizadas por otros lenguajes de programación. Indicamos a continuación la relación principal de las mismas:

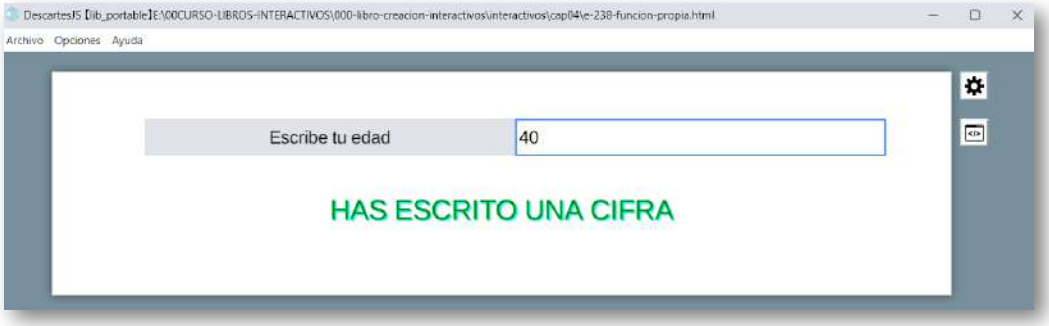
- **_Eval_(string)**: Recibe una cadena de texto que representa una expresión o función matemática y la evalúa como tal. Permite que el usuario introduzca funciones dinámicamente (por ejemplo, 'sin(x)') y que estas puedan graficarse o calcularse. Es especialmente útil cuando se desea que el usuario defina la función a trabajar.
- **_ExecBlock_(string,label)**: Evalúa un bloque de instrucciones contenido dentro de una etiqueta HTML específica (por ejemplo, <ETIQUETA>...</ETIQUETA>) en una cadena de texto. string contiene el texto completo y label indica la etiqueta a buscar. Permite ejecutar múltiples asignaciones o comandos contenidos en texto externo (por ejemplo, leído desde archivo de texto).
- **_Num_(string)**: Convierte una cadena de texto que representa un número decimal en su valor numérico real. Si la cadena no es numérica, devuelve NaN. Evita que el usuario utilice la calculadora automática de los campos de texto. Acepta tanto punto como coma decimal.
- **isNumber(x)**: Devuelve 1 si x es un número real y 0 si no lo es (NaN, Infinity o cadena de texto). Útil para condicionales que requieren validar si un resultado es numérico.
- **<identificador>.set('vr',var)**: Transfiere el valor de una variable de un espacio (principal o subordinado HTMLIFrame) a otro. El primer argumento es el nombre que tendrá la variable en el espacio receptor y el segundo es la variable emisora. No actualiza automáticamente el espacio receptor (requiere .update()).

- **<identificador>.exec('fnc',arg)**: Ejecuta una función definida en el espacio receptor. El primer argumento es el nombre de la función (cadena) y el segundo sus argumentos (como variable o cadena separada por comas). Permite comunicación activa entre escenas.
- **<identificador>.update()**: Fuerza la actualización del espacio receptor tras usar .set() o .exec(). No recibe argumentos.
- **<identificador>.changeConf(ruta)**: Recarga dinámicamente el contenido HTML de un espacio HTMLIFrame sin reiniciar el intérprete de DescartesJS. ruta es la dirección del archivo .html a cargar.
- **<identificador>.download_image(str)**: Guarda como imagen (.png) la disposición gráfica actual de un espacio. str es el nombre dado al archivo. No incluye fondo, ejes, red ni controles.
- **<control>.play()**: Inicia la reproducción de un archivo de audio o vídeo desde el principio.
- **<control>.stop()**: Detiene completamente la reproducción de audio o vídeo.
- **<control>.pause()**: Pausa la reproducción y permite continuar desde el mismo punto.
- **<control>.currentTime(t)**: Coloca el archivo de audio o vídeo en el segundo t antes de reproducirse.
- **<menú>.setOptions(str)**: Asigna dinámicamente las opciones de un menú tipo control numérico. str es una cadena con las opciones separadas por comas.
- **_MatrixToStr_('M')**: Convierte el contenido de una matriz en una cadena de texto con formato etiquetado. Útil para guardar o transferir datos.
- **_StrToMatrix_(cadena,'M')**: Vuelca el contenido de una cadena con formato adecuado en una matriz. Si la matriz no existe, se crea; si existe, se sobrescribe.

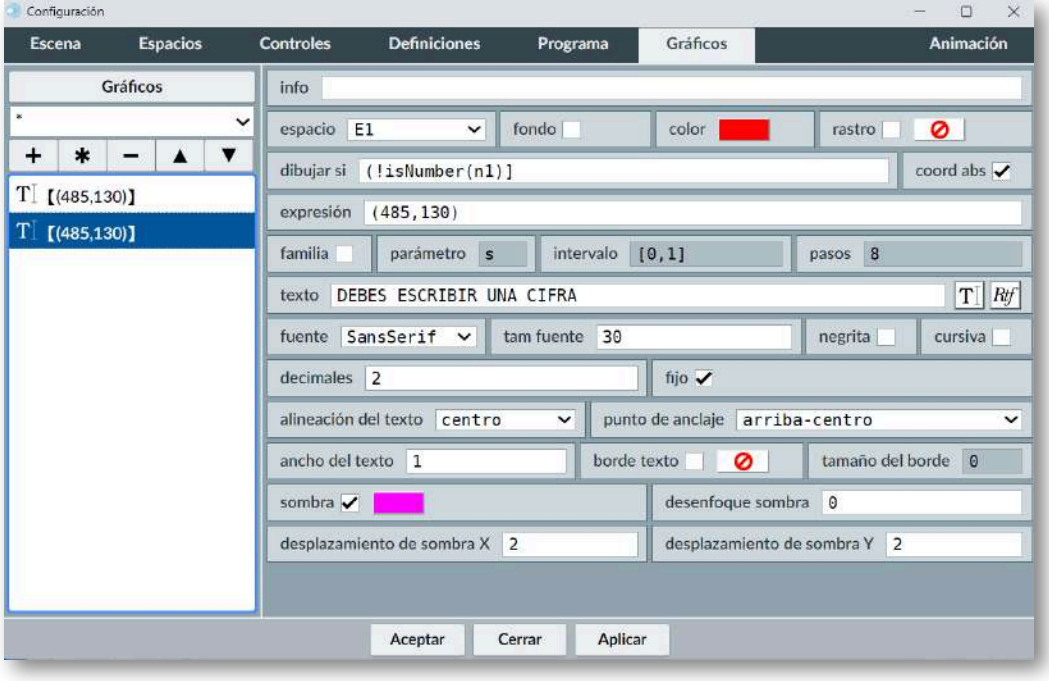
- **_VectorToStr_('V')**: Convierte un vector en cadena de texto etiquetada.
- **_StrToVector_(cadena,'V')**: Vuelca el contenido de una cadena en un vector, creándolo si es necesario.
- **_Save_('archivo.txt',contenido)**: Guarda una cadena de texto en un archivo. Lanza un diálogo de guardado.
- **_Open_('funcion')**: Abre un diálogo para que el usuario seleccione un archivo local. El nombre y contenido del archivo se almacenan en variables internas (DJS.fileName y DJS.fileContent).
- **_Load_('ruta')**: Carga directamente el contenido de un archivo en una cadena de texto. Solo funciona en el editor de DescartesJS por razones de seguridad.

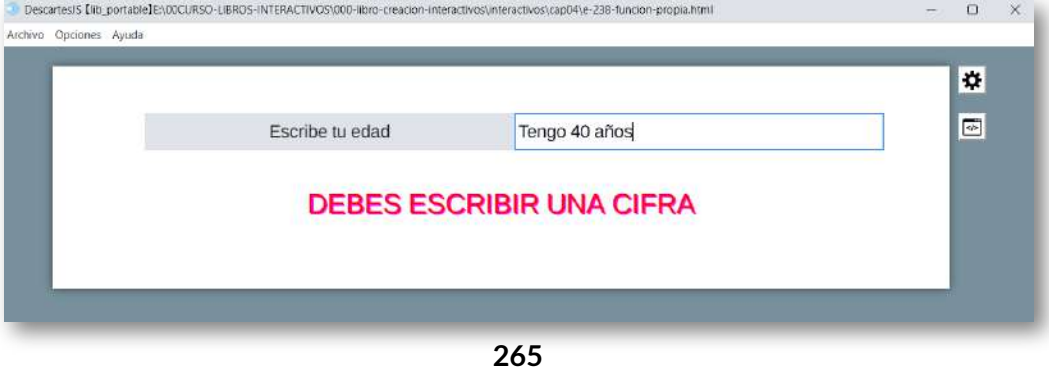
Se muestra un caso con la función propia de Descartes **isNumber(x)**. Para ello, creamos un control campo de texto con identificador n1. A continuación, mostramos un texto en pantalla cuando el valor introducido sea numérico y otro distinto cuando no lo sea.





Podemos ver que cuando se introduce una cifra aparece un texto y cuando no aparece otro.

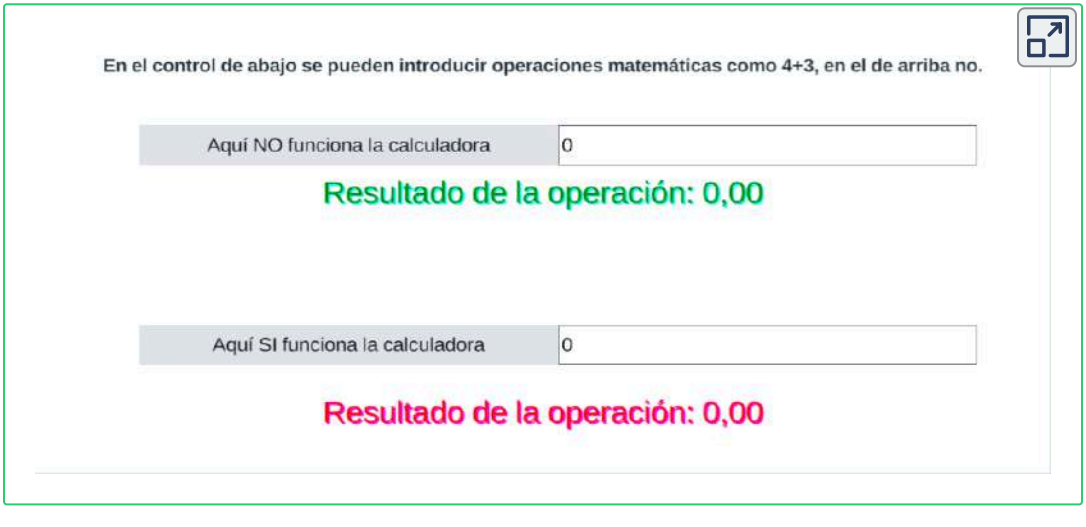






A continuación, se presenta otra escena en la que hemos utilizado la función propia de Descartes **_Num_(string)**, cuya finalidad es convertir, siempre que sea posible, una cadena de texto en un valor numérico real. El control situado en la parte superior está configurado como solo texto, por lo que cualquier entrada del usuario será interpretada estrictamente como una cadena de caracteres y no se evaluarán automáticamente operaciones matemáticas.

Sin embargo, cuando se introduce una cifra válida (por ejemplo, 3 o 4.5), la función **_Num_(string)** permite transformar esa cadena en un número real para poder utilizarlo en cálculos dentro de la escena. Si el contenido no representa un valor numérico válido, la función devolverá NaN, lo que facilita la validación de la entrada.



4.7 Funciones

Una función es el resultado de aplicar determinadas operaciones matemáticas a una o varias variables. El valor que se obtiene tras realizar esas operaciones es, precisamente, el valor de la función. Para ilustrarlo, podemos definir una función sencilla que multiplique un número por dos. Si la variable es x, la expresión será 2·x y podremos escribirla como funcion(x)=2·x, indicando así que el valor de la función depende de la variable x. Del mismo modo, si queremos definir una función que sume dos valores, por ejemplo, x e y, podremos expresarla como f(x,y)=x+y, lo que indica que la función depende de ambas variables.

En Descartes, tras el nombre de la función se escriben, entre paréntesis y separadas por comas, las variables de las que depende. En esencia, una función asigna un número real a cada valor de su variable (o a cada conjunto de valores, si depende de varias). Toda función consta de un identificador o nombre, una lista de variables —que aparecen entre paréntesis— y una expresión matemática en la que intervienen dichas variables.

Ejemplos de funciones de una variable:

Expresión	Identificador	Variable
f(x)=sen(3*x+2)	f	x
g(y)=A*exp(-(y^2)/(s^2))	g	y
Area(radio)=(pi*radio^2)/2	Area	radio

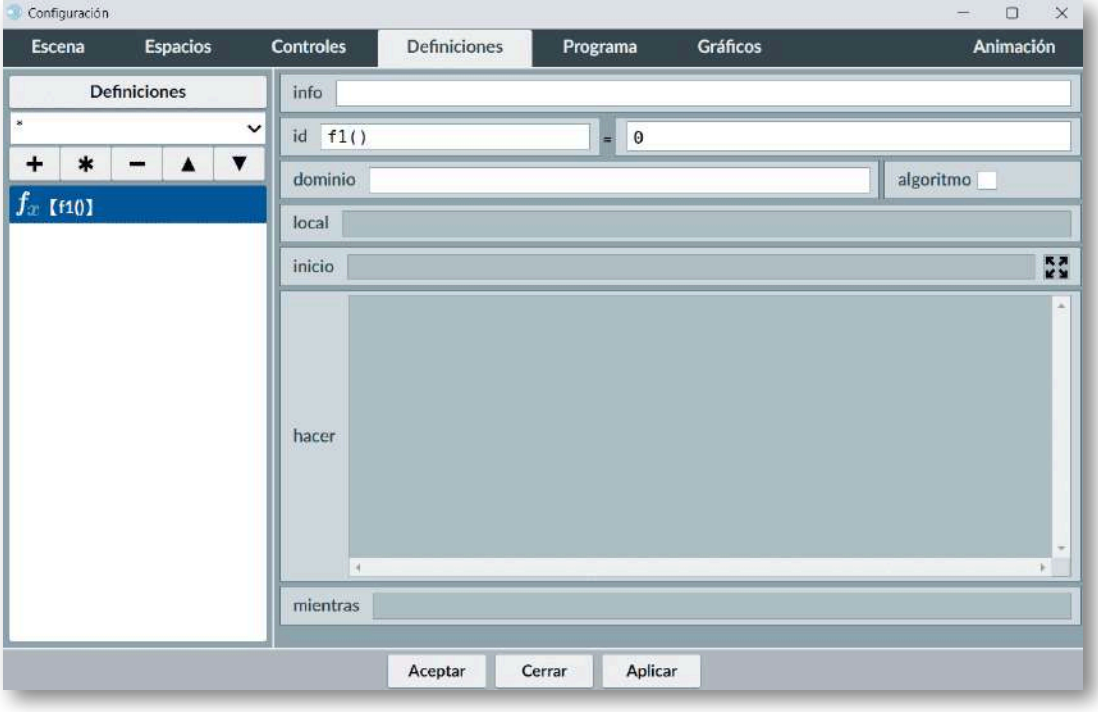
Ejemplos de funciones de dos o más variables:

Expresión	Identificador	Variables
f(x,y)=raíz(x^2+y^2)	f	x, y
Fun(x,y,z)	Fun	x, y, z
Volumen(a,b,c)=4*pi*a*b*c/3	Volumen	a, b, c

A continuación, se presenta un esquema más detallado sobre el funcionamiento y la utilidad de las funciones:

- **Estructura:** una función reúne una o varias instrucciones dentro de un mismo bloque. Estas instrucciones pueden consistir en operaciones matemáticas, asignaciones o llamadas a otras funciones, y quedan organizadas bajo un único identificador del tipo f(), f(x), f(x,y).
- **Finalidad:** su principal ventaja es la reutilización. Permiten evitar repetir el mismo conjunto de instrucciones en distintos puntos de la escena, facilitando un código más claro, ordenado y fácil de mantener.
- **Variables:** los nombres de las variables determinan el papel que desempeñan dentro de la expresión. Actúan como parámetros de entrada y permiten que la función produzca resultados distintos según los valores que reciban. Son variables, por ejemplo, x, y o z.
- **Evaluación:** una función no se ejecuta de forma automática; solo se calcula cuando es llamada explícitamente desde algún punto de la escena.
- **Origen de la llamada:** puede invocarse desde un algoritmo, un gráfico, una variable de texto, una animación, otra función o la constricción de un control gráfico, es decir, desde cualquier elemento que requiera su resultado.
- **Forma de invocación:** para llamarla se escribe su nombre seguido de los valores correspondientes entre paréntesis, por ejemplo, f(x) o Fun(x,y,z). Si tiene varios parámetros, estos se separan por comas.
- **Resultado:** al invocarla, la función calcula y devuelve su valor. No es imprescindible asignarlo a otra variable; basta con escribir la llamada para que se ejecute y produzca el resultado correspondiente.

En este sentido, las funciones constituyen uno de los pilares fundamentales de la programación, pues facilitan la estructuración, la reutilización y el mantenimiento del código en el que se incluyen. Por ello, en este libro digital interactivo se les dedicará una atención especial. A continuación, se describen los elementos propios de las funciones, a las que se accede desde el selector Definiciones del editor de escenas.



- **Información de la función (info):** comentario que no se procesa a efectos de programación y que se utiliza para realizar anotaciones.
- **Identificador de la función (id):** es el campo de texto donde se escribe el nombre con el que se invocará la función. Al llamarla, puede incluir argumentos entre paréntesis a continuación del nombre; si hay más de uno, deben separarse por comas. Incluso cuando la función no recibe argumentos, los paréntesis deben escribirse, aunque estén vacíos.

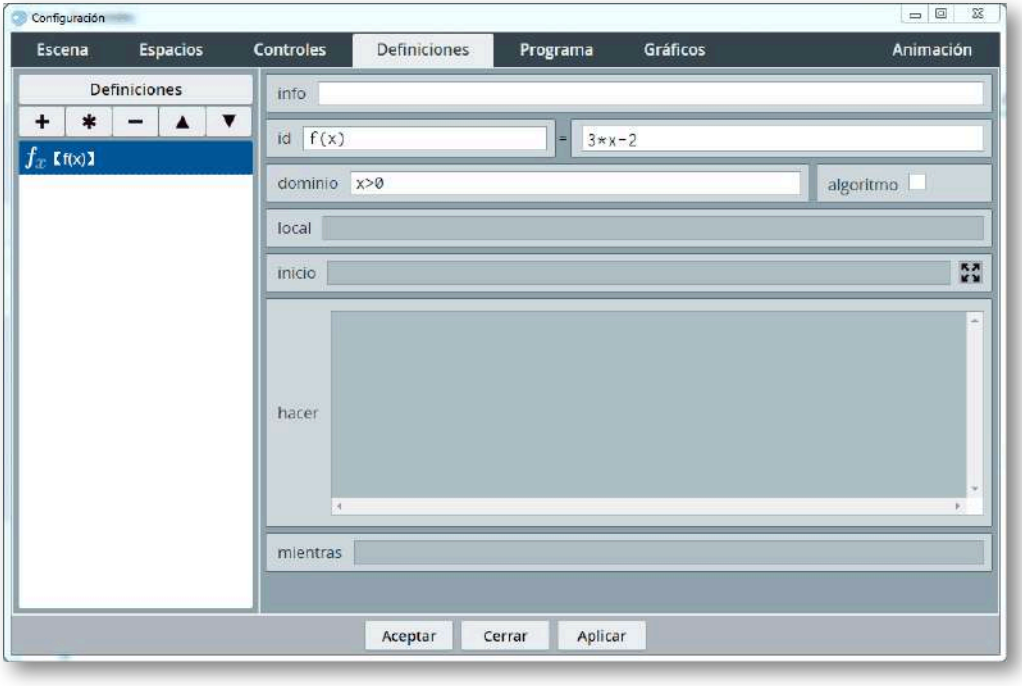
- **Valor o expresión de retorno:** es el campo situado a la derecha del símbolo =. En él se indica el valor o la expresión que la función devolverá cuando sea llamada. Ese resultado es el que se asignará, en su caso, a la variable que invoque la función. Si únicamente se desea que la función ejecute acciones sin devolver un valor, este campo puede dejarse vacío.
- **Dominio:** es un campo en el que se establece una condición lógica. La función solo se evaluará en aquellos valores para los que dicha condición sea verdadera. Por ejemplo, si el dominio incluye la condición (x > 1)&(x < −1), la función excluirá el intervalo [−1,1] en su evaluación.
- **Algoritmo:** es una casilla que, al activarse, permite definir la función como un conjunto de instrucciones y no únicamente como una expresión directa. Al marcarla, se habilitan los campos Local, inicio, hacer y mientras, cuya lógica es similar a la empleada en los algoritmos de INICIO y CALCULOS.
- **Local:** en este campo se declaran las variables que tendrán ámbito local dentro de la función. Una variable local solo puede modificar su valor dentro de esa función y no afectará a otras variables con el mismo nombre en otras partes de la escena. Esto protege el código frente a interferencias no deseadas. Cuando la función recibe argumentos, estos se consideran automáticamente variables locales. Si desde una función principal se llama a otra subordinada, las variables locales pueden heredar su valor, salvo que también se declaren como locales en la función subordinada.
- **Inicio:** es un campo de texto donde se escriben las instrucciones que deben ejecutarse antes de cualquier repetición del algoritmo. Aquí pueden incluirse asignaciones, llamadas a funciones u otras acciones preparatorias. Si se escriben varias instrucciones, deben separarse mediante punto y coma (;). Las variables aquí definidas si pueden interferir con otras del mismo nombre que existan fuera de la función.

- **Hacer:** es un panel destinado a introducir instrucciones, normalmente en varias líneas. Si el algoritmo implica un proceso repetitivo, las instrucciones que deban ejecutarse de forma cíclica deben colocarse en este apartado. Si no hay repetición, pueden escribirse aquí o en el campo inicio.
- **Mientras:** en este campo se especifica la condición que determina si las instrucciones incluidas en hacer deben repetirse. Mientras la condición sea verdadera, el ciclo continuará ejecutándose. Si se deja vacío, las instrucciones de hacer se ejecutarán una sola vez. Existe un límite máximo de 100.000 iteraciones; aunque la condición implicara más repeticiones, el proceso se detendrá al alcanzar ese número para evitar bucles infinitos y bloqueos de la escena.

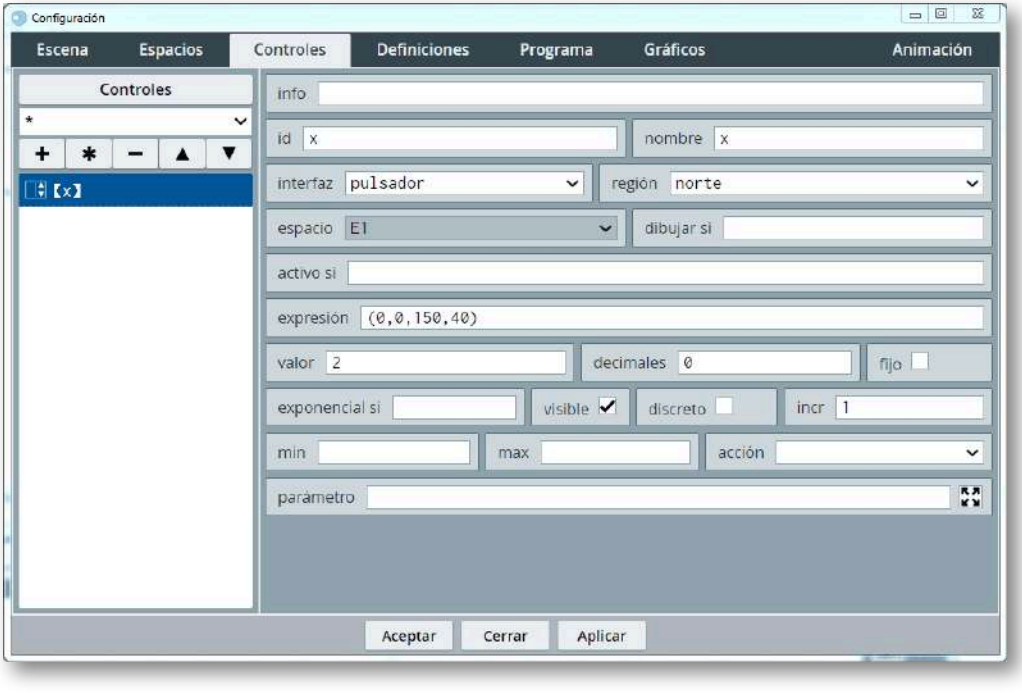
Vamos a definir una función sencilla cuyo objetivo sea tomar un número, multiplicarlo por tres y luego restarle dos. La expresión algebraica correspondiente es 3*x - 2, y la función la denominaremos f(x), indicando que su valor depende directamente de la variable x. Esta notación nos permite llamar a la función con distintos valores de x para obtener resultados diferentes según el argumento proporcionado.

Además, estableceremos una condición que limite cuándo se evaluará la función: solo se calculará cuando x > 0. De esta manera, su dominio queda restringido exclusivamente a los valores positivos de la variable, evitando que se realicen gráficos fuera de este rango.

No marcaremos el check **algoritmo**, lo que implica que la función no se ejecutará de manera iterativa o cíclica. Es decir, las instrucciones que eventualmente se incluyan en el campo **hacer** no se repetirán automáticamente, incluso si las condiciones definidas en el apartado **mientras** se cumplen. Esto garantiza que la función se evalúe únicamente cuando se invoque explícitamente, manteniendo un control preciso sobre su ejecución.

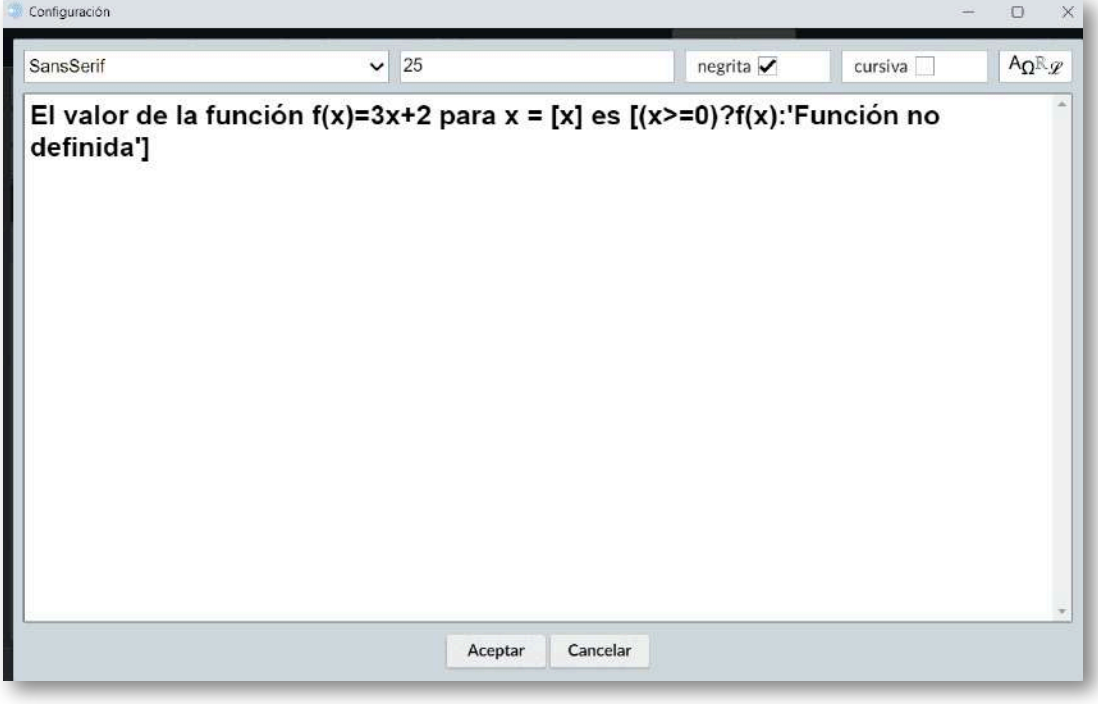


Vamos a colocar un control numérico tipo pulsador en la zona norte, el cual asignará valores a la variable x.

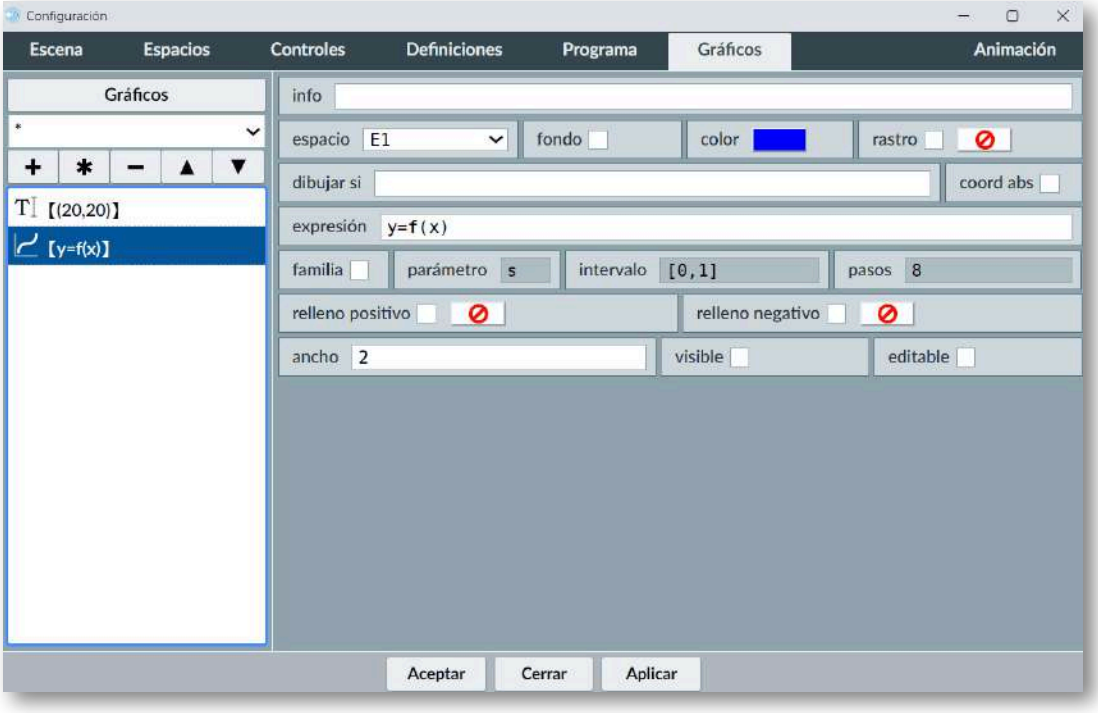


También insertaremos un texto con formato en el que se invoquen la variable x y la función f(x). Al ser llamada desde el texto, la función se ejecuta y muestra su valor en pantalla.

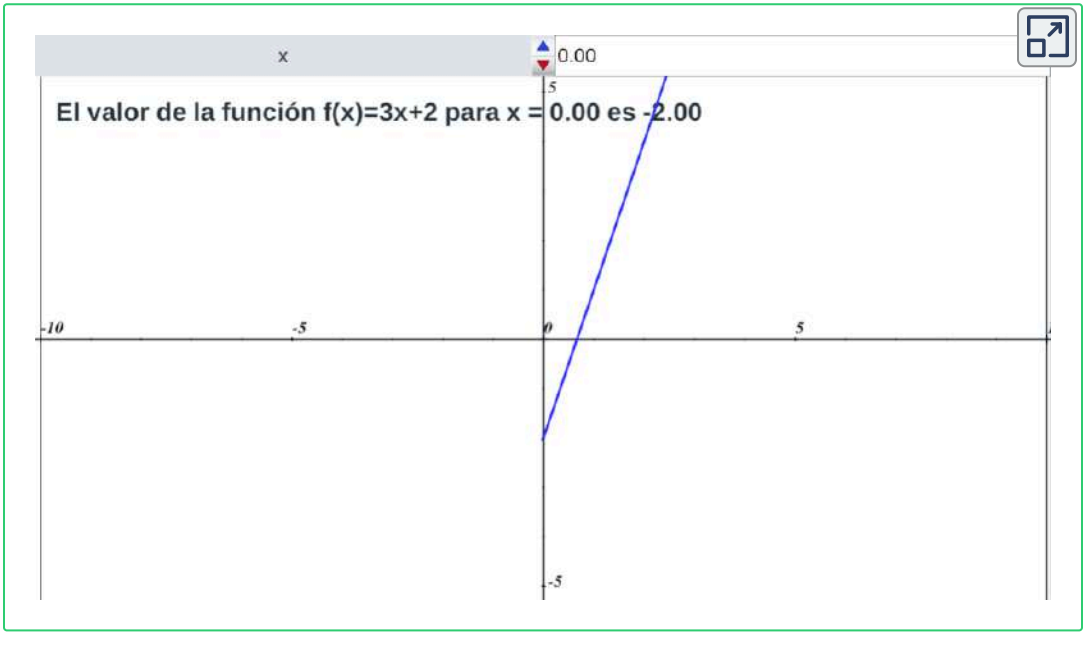
Recordemos que tanto la variable x como la función f(x) se introducen entre corchetes en el texto para que puedan ser evaluadas correctamente. Además, usamos un condicional del tipo (x >= 0) ? f(x) : 'Función no definida' para evitar que la función se evalúe fuera de su dominio. De este modo, aunque no se evalúe gráficamente, sí se obtiene el resultado numéricamente, por ello, se debe condicionar para que la función solo de valores dentro de su dominio.



Seguidamente, solicitaremos la representación gráfica de la función f(x). Para ello, en el selector Gráficos creamos un elemento tipo ecuación y en su expresión escribimos y = f(x). Además, modificamos el color a azul y duplicamos el valor del ancho de línea para mejorar la visibilidad de la gráfica.



En la escena podemos ver el resultado de la definición de la función dentro de la misma.



A veces no es posible definir una función únicamente mediante una expresión, ya que su evaluación requiere un proceso más complejo. En estos casos, el autor puede marcar el campo **algoritmo**. Al hacerlo, la función se evalúa siguiendo el procedimiento definido en los campos de texto **inicio**, **local**, **hacer** y **mientras**.

El siguiente ejemplo muestra cómo una función puede requerir un algoritmo: la función factorial. Matemáticamente, el factorial de un número entero positivo x se calcula multiplicando x por todos los enteros positivos menores que él. Por ejemplo, 4! = 4·3·2·1 = 24 y 5! = 5·4·3·2·1 = 120. En nuestra escena definiremos la función Fact(x), cuyo dominio será x >= 0 y que devolverá el factorial de x.

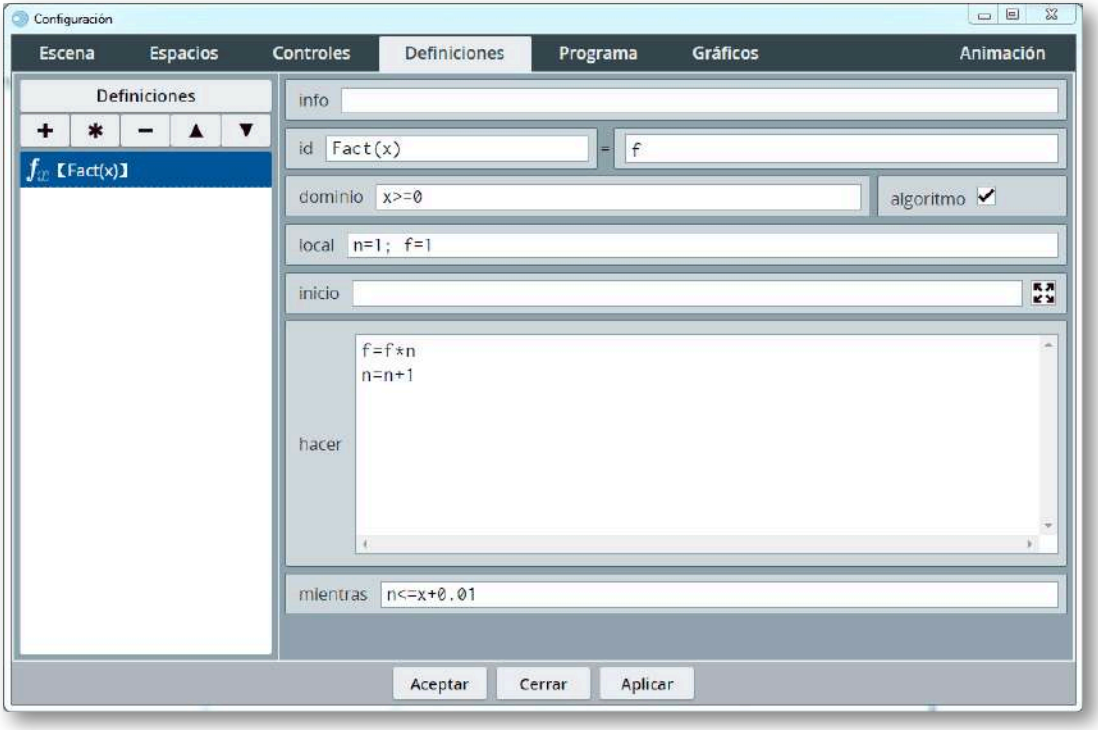
Para calcular el factorial de 3 mediante el algoritmo de Fact(x):

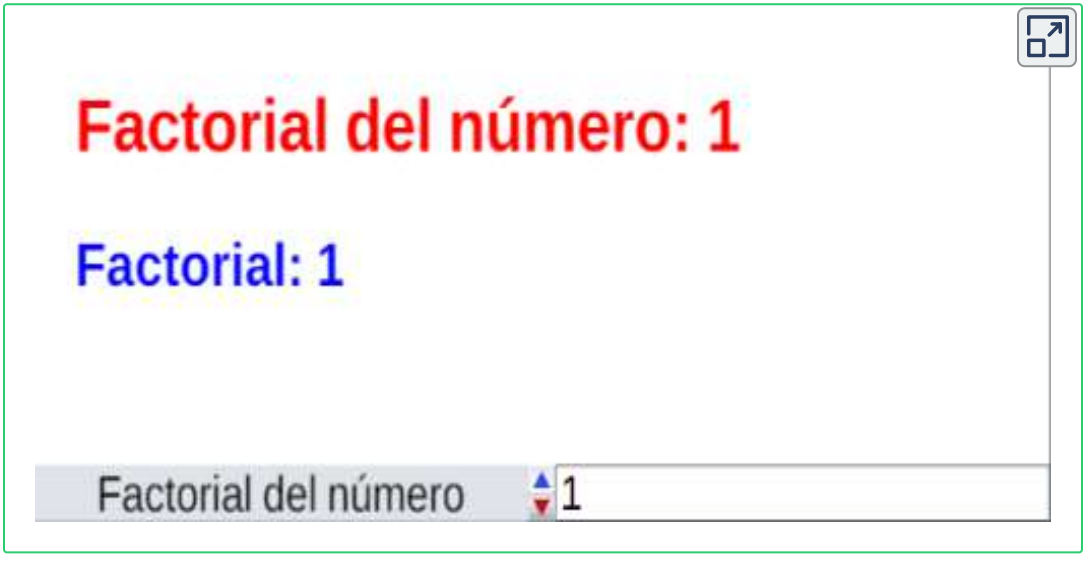
- En **inicio** se asigna n = 1 y f = 1.
- En **hacer** se realizan las operaciones f = f*n y n = n+1, mientras n <= x + 0.01.

Evaluando x = 3:

1. Primer ciclo: f=1, n=1 → f=1*1=1, n=2. La condición n <= x+0.01 se cumple (2 ≤ 3.01), se repite **hacer**.
2. Segundo ciclo: f=1, n=2 → f=1*2=2, n=3. La condición se cumple (3 ≤ 3.01), se repite **hacer**.
3. Tercer ciclo: f=2, n=3 → f=2*3=6, n=4. La condición ya no se cumple (4 > 3.01), el algoritmo se detiene. El valor final f=6, que es el factorial de 3.

A continuación, se muestra cómo queda definida la función Fact(x) y cómo se presenta la escena completa que la contiene. En dicha escena también se incluyen los textos, el control numérico y el espacio gráfico asociados a la función.

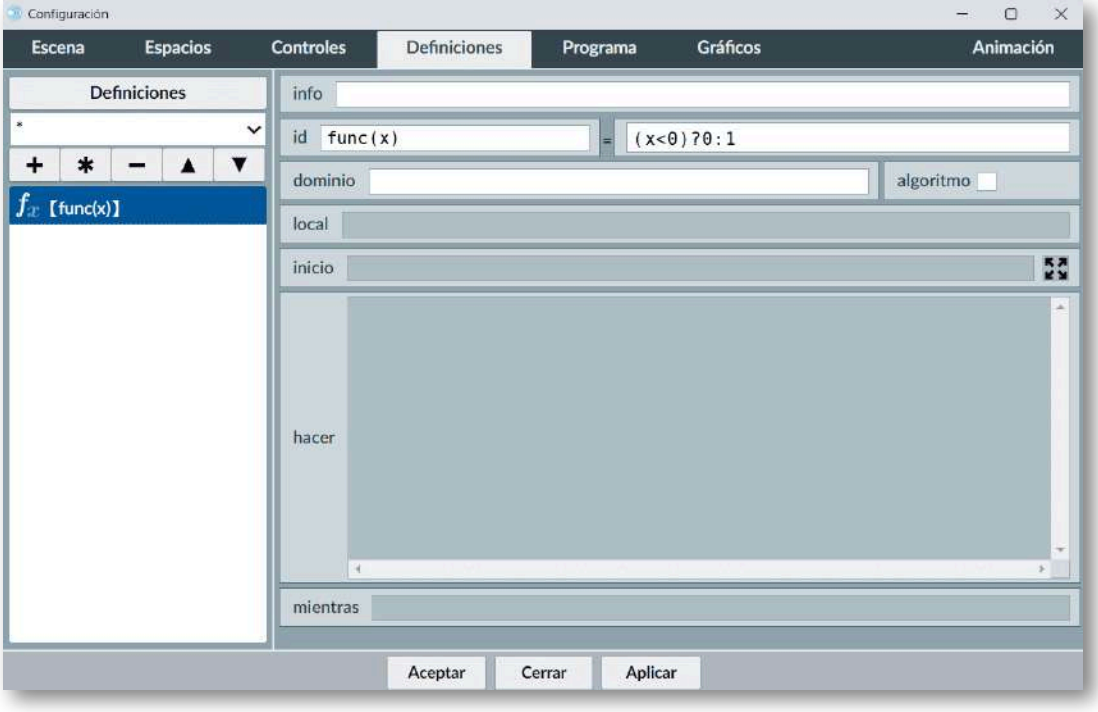


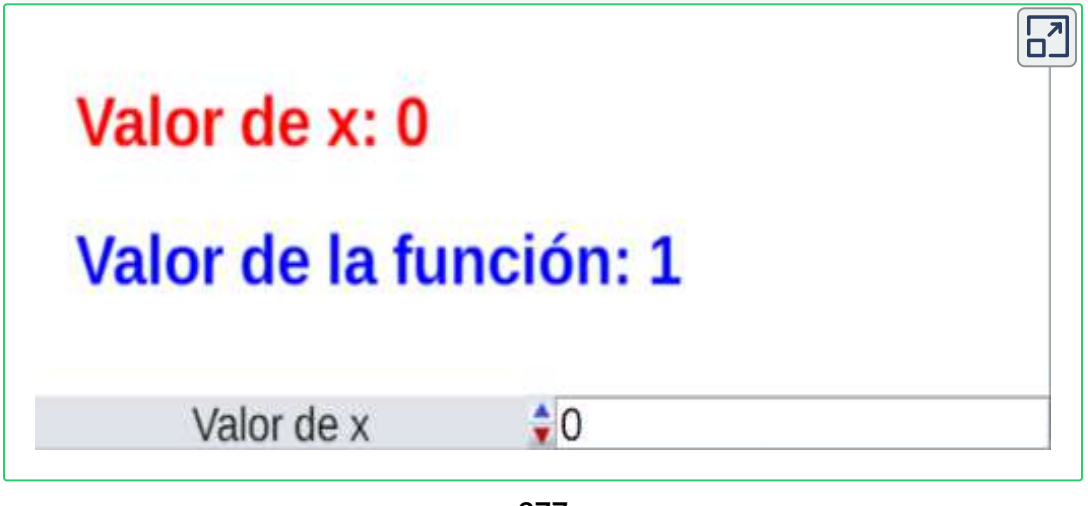


Las funciones también pueden utilizar operadores condicionales o booleanos para definirse. La expresión que aparece a la derecha del signo = puede ser, o incluir, un término condicional.

Ejemplo: func(x) = (x < 0) ? 0 : 1, donde func es una función que vale 0 cuando x < 0 y 1 cuando x >= 0.

Una expresión booleana vale 1 cuando es verdadera y 0 cuando es falsa, por lo cual la función anterior también podría escribirse como: func(x) = (x >= 0).





4.8 Variables

Las variables son elementos de programación flexibles. Su valor puede ser un valor constante o cualquier expresión definida en términos de los parámetros de los controles (es decir, el valor que tenga un control en cada momento), de otras variables que aparezcan antes en la lista o de x e y.

El valor de una variable puede ser numérico o de cadena (texto), en cuyo caso debe aparecer entre comillas sencillas, por ejemplo, 'valor de la variable'.

Las variables se evalúan cada vez que se utilizan, lo que las hace especialmente útiles. Por ejemplo, si se define una variable como R = raíz(x^2 + y^2) y se crea una gráfica de tipo ecuación con la expresión R = 2, se obtiene la gráfica de una circunferencia de radio 2. En otras palabras, las variables pueden actuar como abreviaturas de expresiones más o menos complejas.

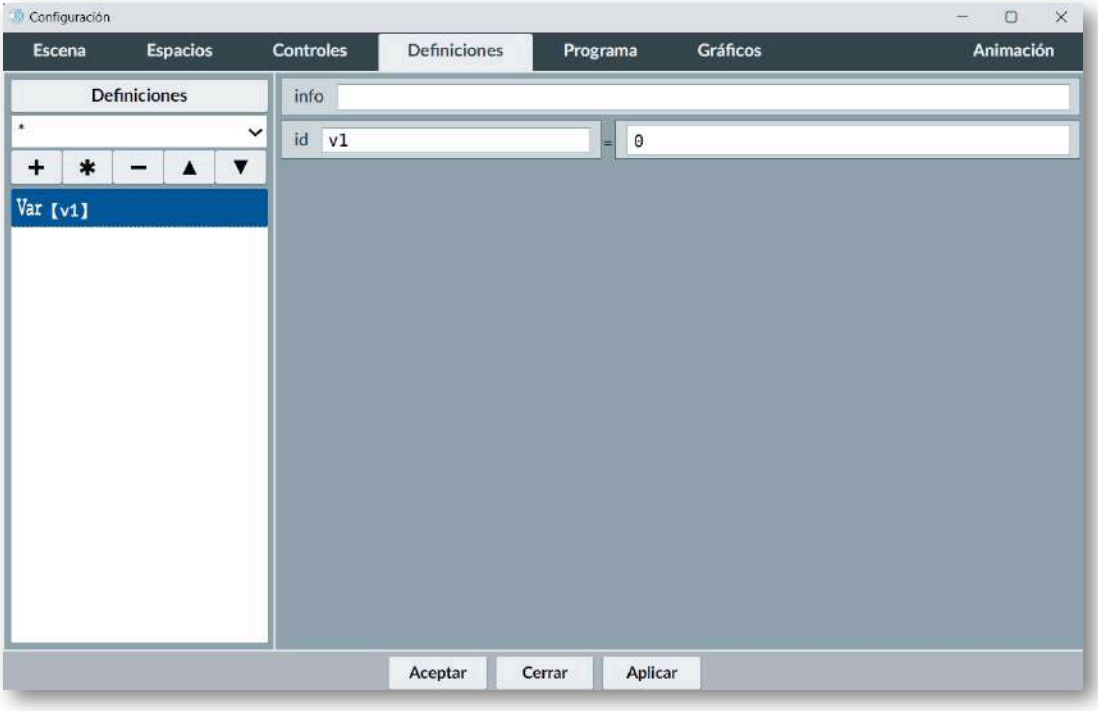
Existen dos variables numéricas "reservadas": e, correspondiente al número de Euler, y pi. Ambas están pre-inicializadas, y aunque el autor de la escena podría modificar sus valores, no es aconsejable hacerlo.

La definición de **variable** permite asignar un valor o una expresión a una variable desde el momento en que se carga el interactivo. Cuando recibe una expresión, se comporta de manera muy similar a la definición tipo función, por lo que actualmente su uso no es tan frecuente.

Además del parámetro **info**, los parámetros que se definen para las variables son:

- **Identificador:** campo de texto en el que se introduce el nombre de la variable. Este identificador se utiliza cada vez que se necesite acceder al valor de la variable.

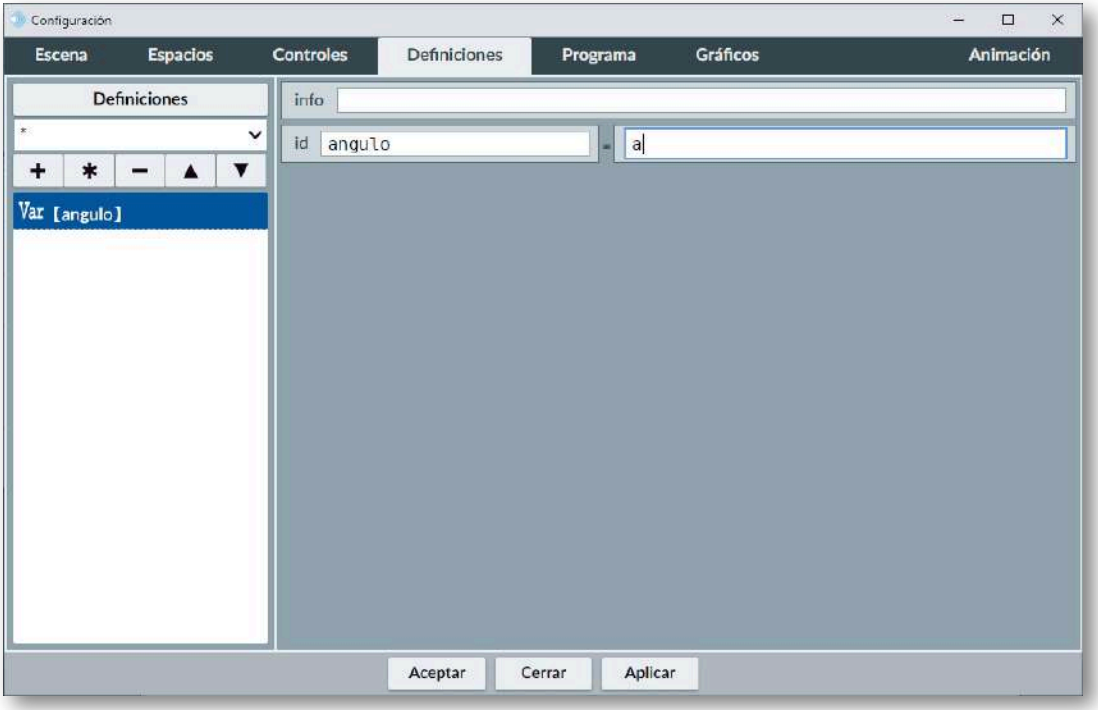
- **Valor de la variable:** campo de texto situado a la derecha del signo igual, en el que se introduce el valor o la expresión de la variable. Puede ser un número explícito o hacer referencia a otra variable mediante su identificador.

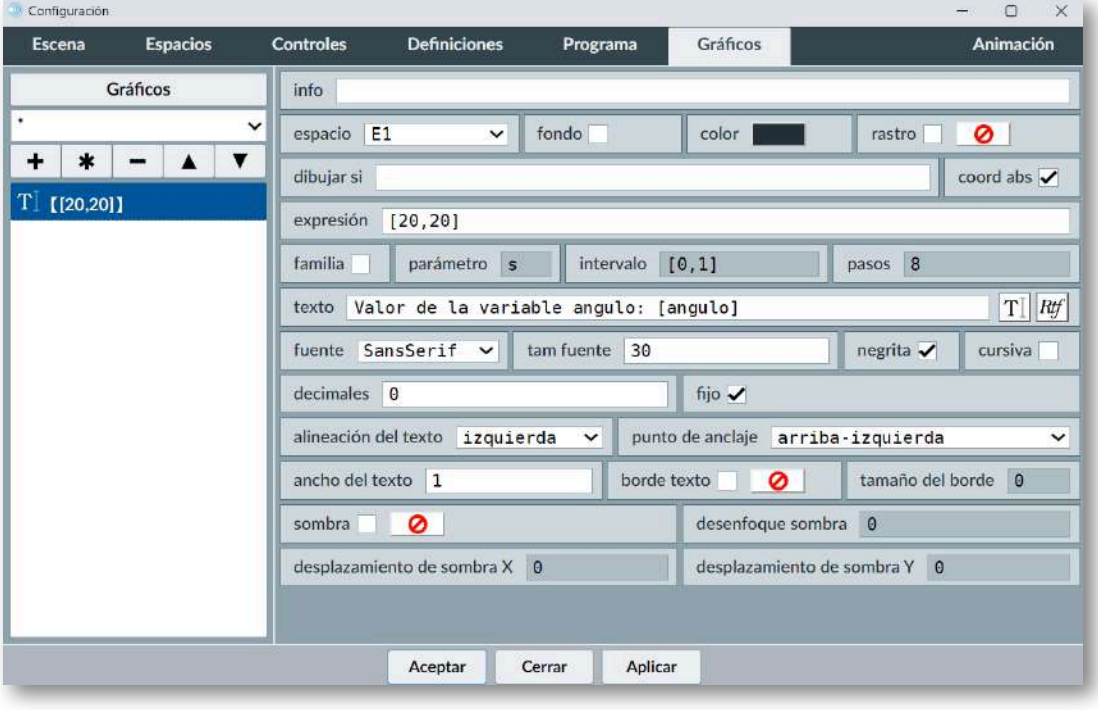


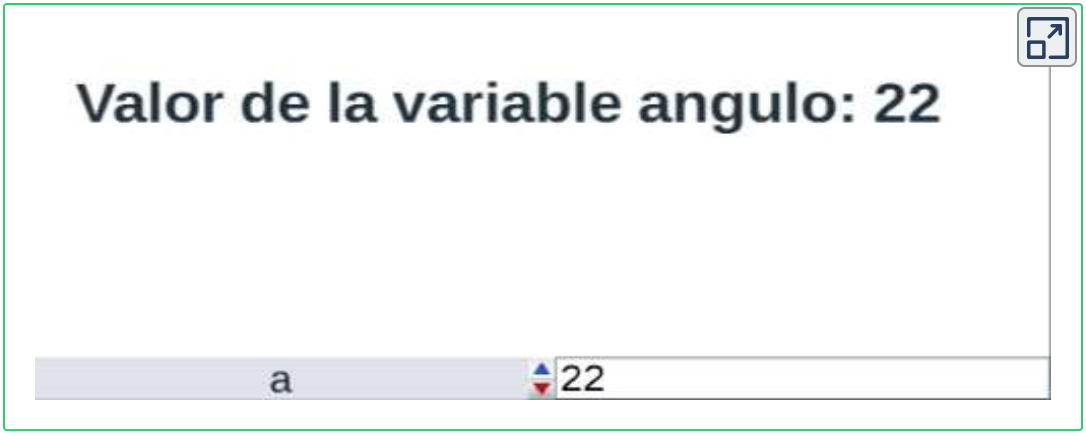
Vamos a insertar un control en la escena llamado ángulo, cuyo valor se almacenará en la variable a. Para ello, definimos un control numérico tipo pulsador con identificador a en la zona norte de la interfaz.

Observaremos que la variable a adopta los valores del control a. Para verificarlo, incluiremos en la escena un texto que muestre en pantalla el valor actual de la variable angulo.

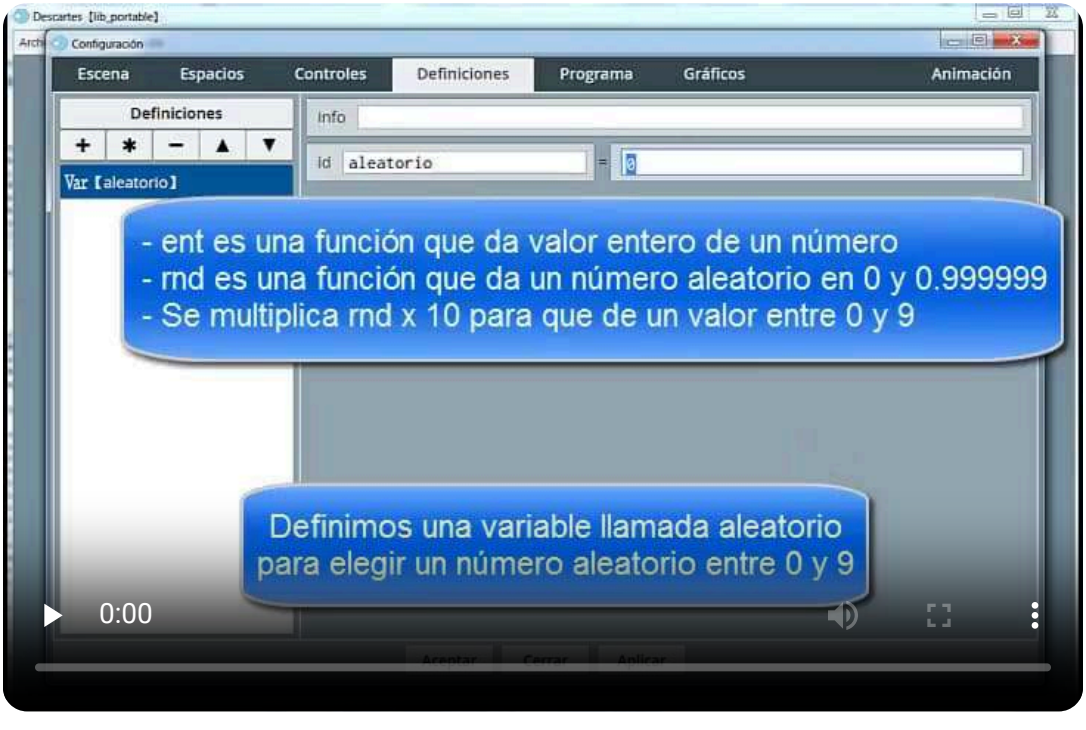
Al modificar el valor en el control, el valor de la variable mostrada también se actualizará automáticamente, reflejando el cambio en tiempo real.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Definiciones
Var [angulo]
info
id angulo
a
Aceptar
Cerrar
Aplicar

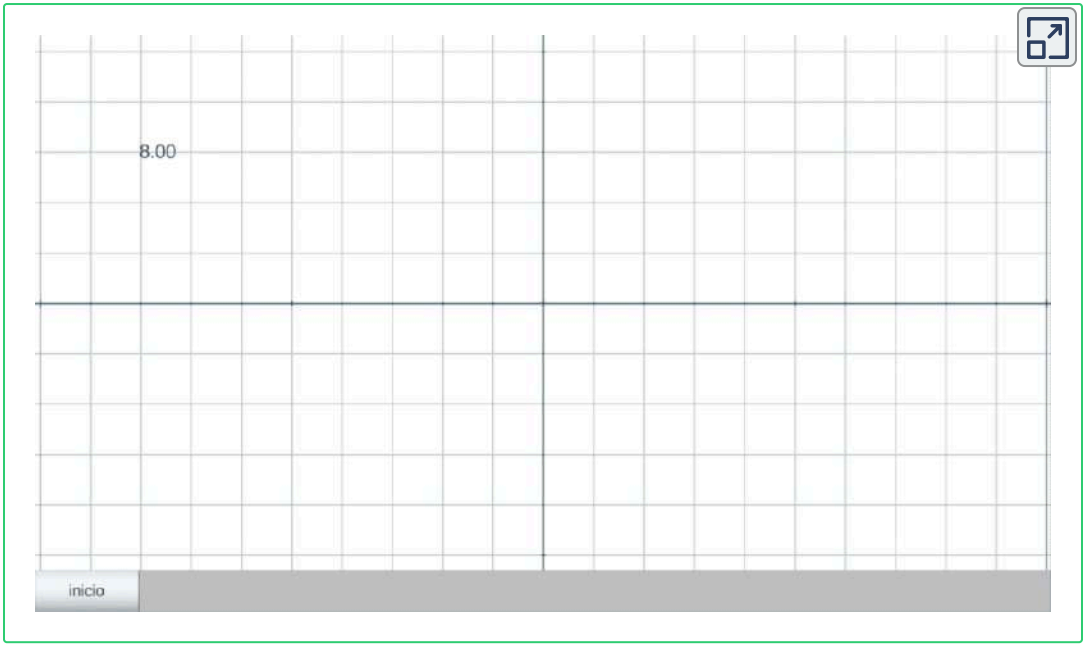

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
[[20,20]]
info
espacio E1
fondo
color
rastro
dibujar si
coord abs
expresión [20,20]
familia
parámetro s
intervalo [0,1]
pasos 8
texto Valor de la variable angulo: [angulo]
fuente SansSerif
tam fuente 30
negrita
cursiva
decimales 0
fijo
alineación del texto izquierda
punto de anclaje arriba-izquierda
ancho del texto 1
borde texto
tamaño del borde 0
sombra
desenfoque sombra 0
desplazamiento de sombra X 0
desplazamiento de sombra Y 0
Aceptar
Cerrar
Aplicar



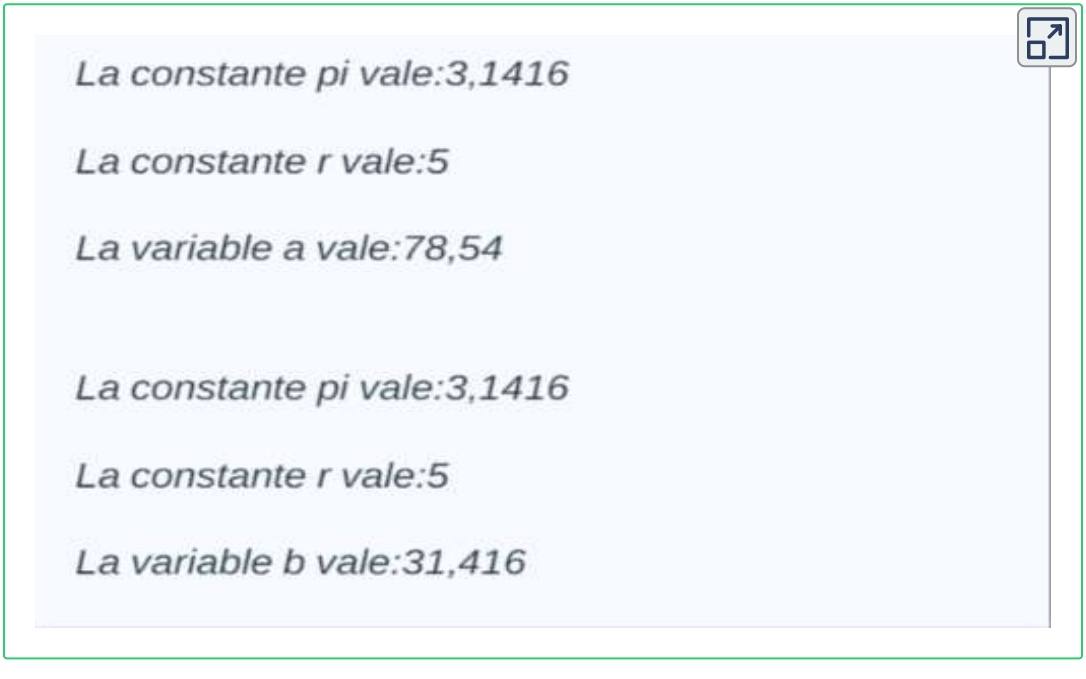
A continuación, se incluye un vídeo donde se muestra cómo obtener números aleatorios con Descartes mediante la utilización de una variable.



A continuación, se muestra la escena del vídeo, que ejemplifica el uso de variables en DescartesJS.



A continuación, se presenta otra escena en la que se realizan cálculos de fórmulas utilizando variables.



4.9 Vectores

Un **vector** es una lista de valores que pueden ser números, textos, variables, constantes, etc. Cada vector tiene un identificador o nombre, un tamaño que indica su longitud o número de elementos, y un parámetro **evaluar** que determina si se evalúa una sola vez al inicio de la escena.

El área de texto del panel de configuración se utiliza para asignar valores a los elementos del vector. También es posible usar un algoritmo para inicializarlos. Cada elemento se asigna indicando el nombre del vector seguido del número del elemento entre corchetes, comenzando la numeración desde cero. Por defecto, todos los elementos tienen valor cero.

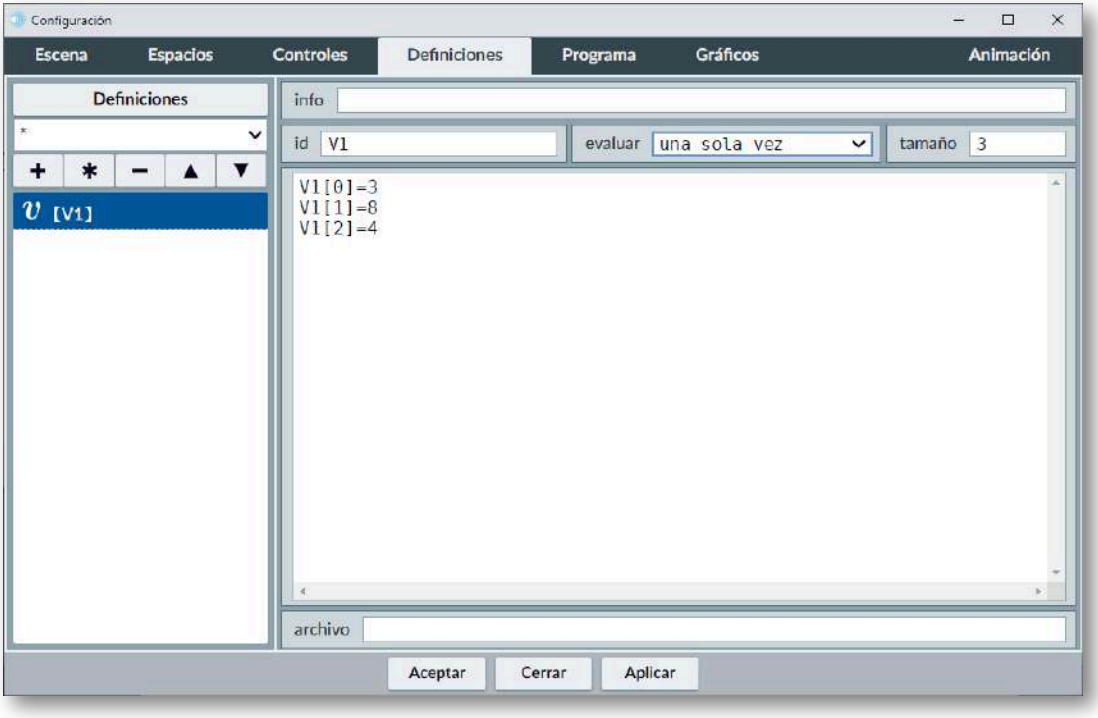
En ocasiones, se desea manejar muchas variables que guardan información relacionada entre si de alguna manera. Para ello, es más práctico definir un **vector**, que funciona como un "mueble con muchos cajones": todos comparten el mismo nombre, pero cada cajón tiene un índice que lo identifica. Por ejemplo, en el vector v1, v1[0] representa el primer cajón, v1[1] el segundo, y así sucesivamente. Cada cajón puede contener un valor o una cadena, como una variable común.

Los principales parámetros de los vectores, además del parámetro info para introducir comentarios, son:

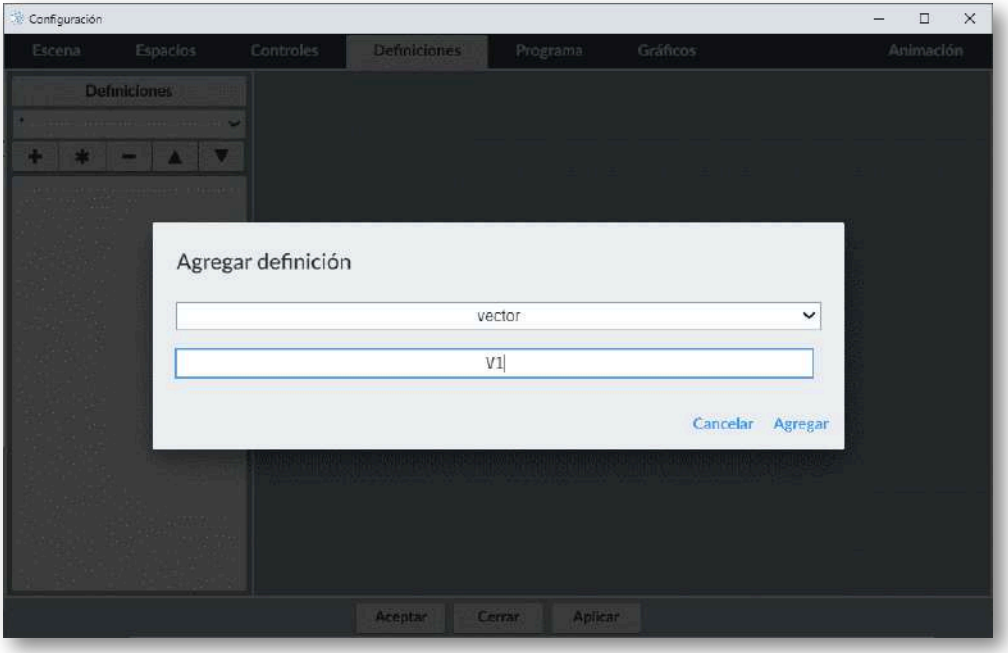
- **id:** identificador del vector. Cada entrada se accede usando corchetes, por ejemplo, Vc[2] es la tercera entrada.
- **evaluar:** una sola vez al cargar la escena. La actualización del vector suele hacerse mediante funciones.
- **tamaño:** número de entradas o de filas del vector (su definición en este campo no es relevante).

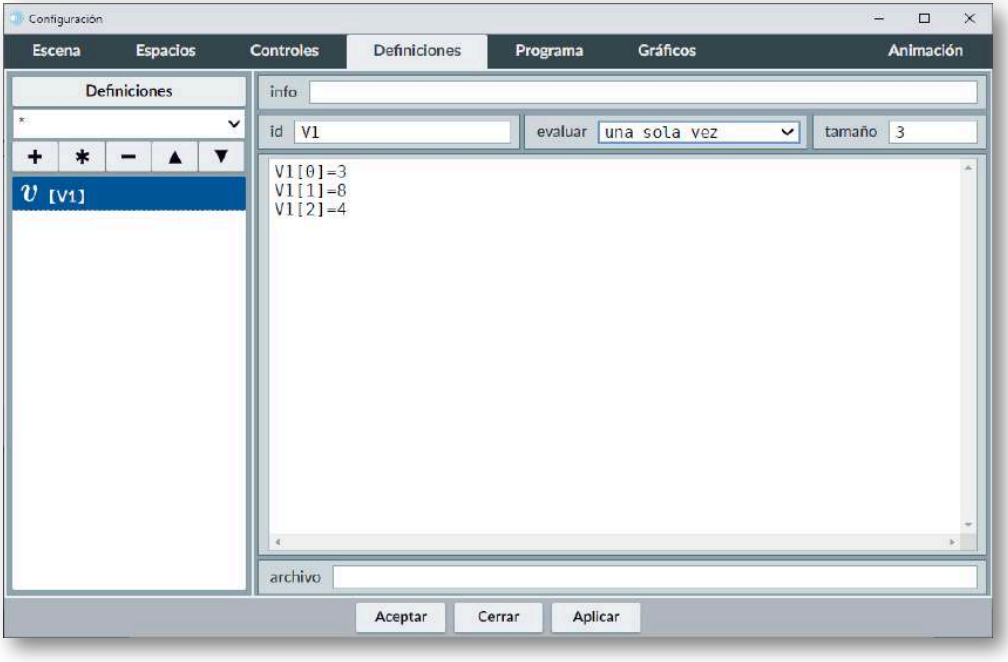
- **panel de asignaciones:** permite introducir valores iniciales para los elementos del vector. Si se modificara mediante funciones, puede dejarse vacío. Los valores alfanuméricos deben ir entre comillas simples, los numéricos no llevan comillas.
- **archivo:** ruta relativa a un archivo que contiene los datos del vector, separados por saltos de línea. El archivo debe tener codificación UTF-8 para evitar errores. Los archivos deben estar en un servidor en la red o incluirse el vector al final de la escena como script.

  Para que DescartesJS lea correctamente los datos de un archivo, el HTML debe estar guardado en la ubicación que haga válida la ruta del archivo. Una vez guardada la escena, los datos del vector pueden integrarse directamente como un script dentro del HTML, permitiendo que el vector funcione incluso si el archivo original ya no está presente, aunque se recomienda mantenerlo en su ubicación original.

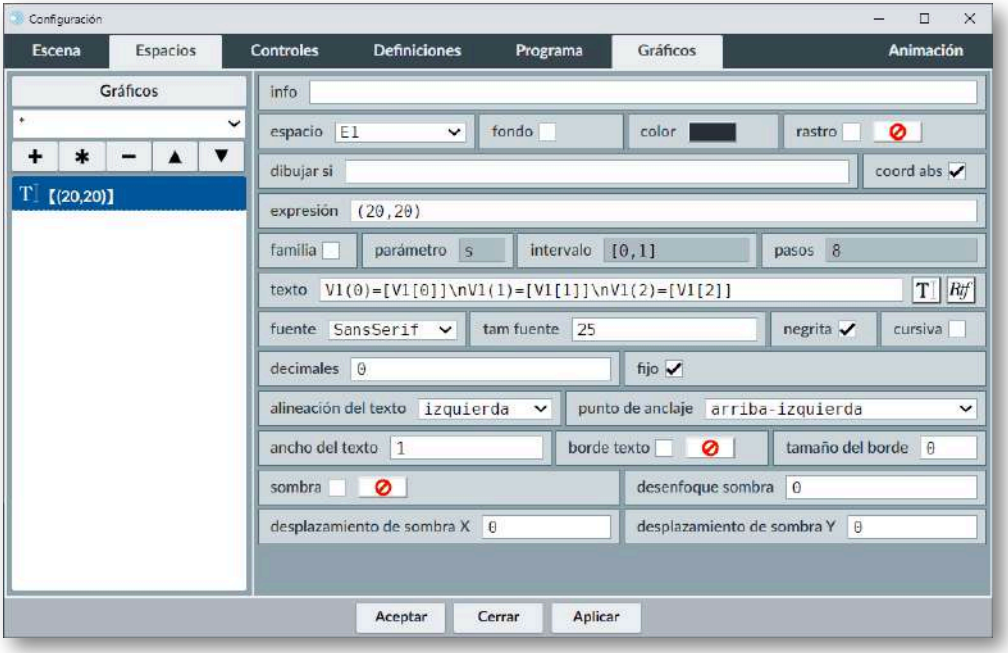


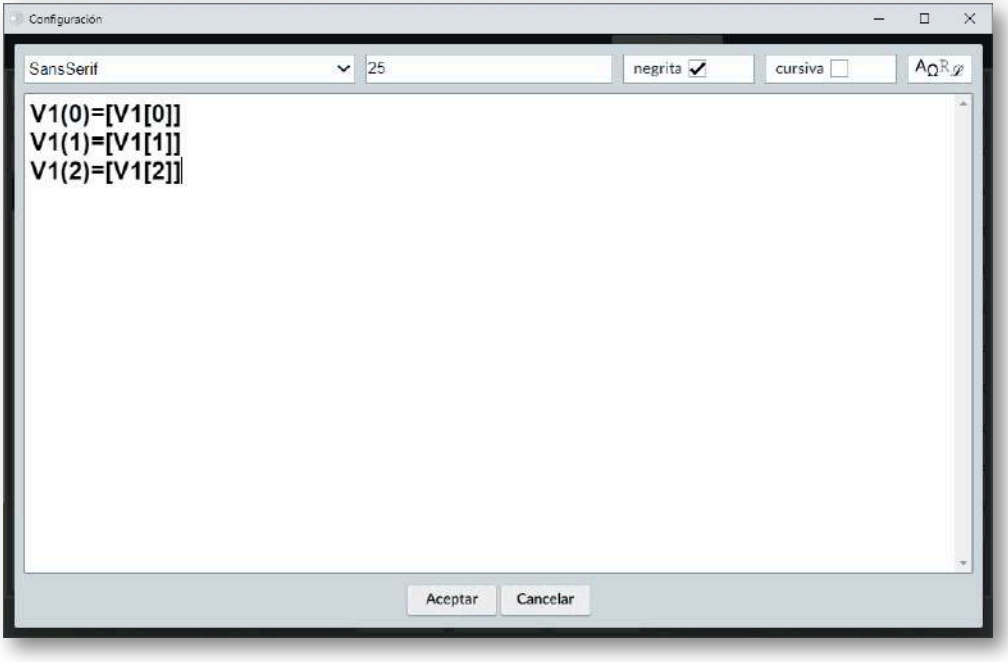
Como ejemplo, crearemos un vector llamado **V1**, que se evaluará una sola vez y tendrá tres elementos. Sus valores iniciales serán: V1[0]=3, V1[1]=8 y V1[2]=4.

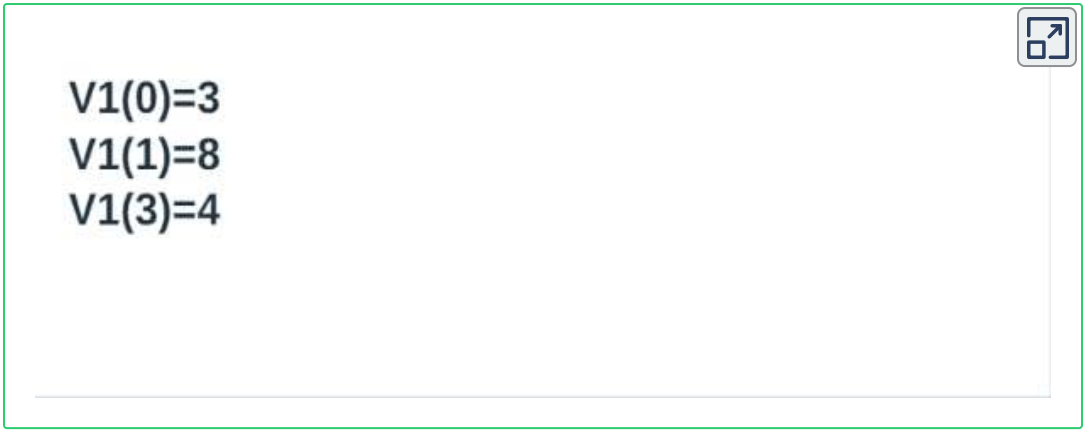




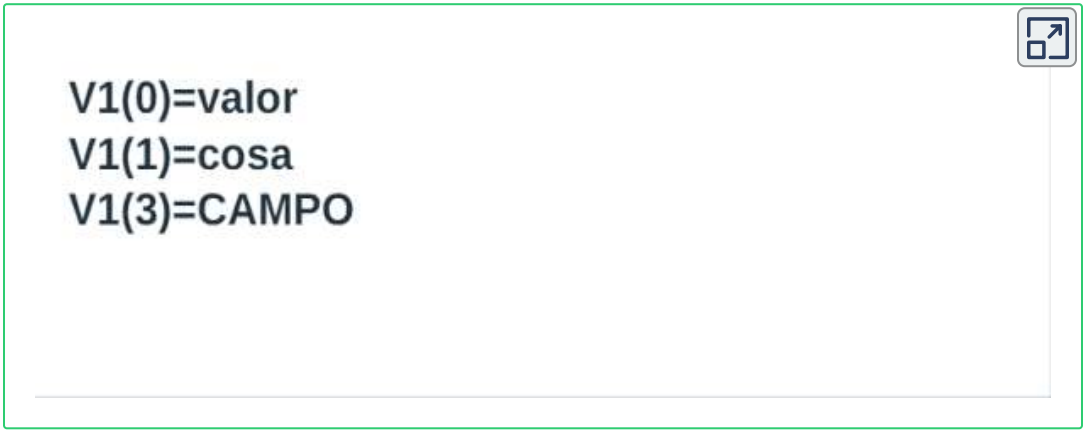
Mediante la inserción de un texto en la escena, mostraremos en pantalla los valores de los componentes del vector **V1**: V1[0], V1[1] y V1[2].





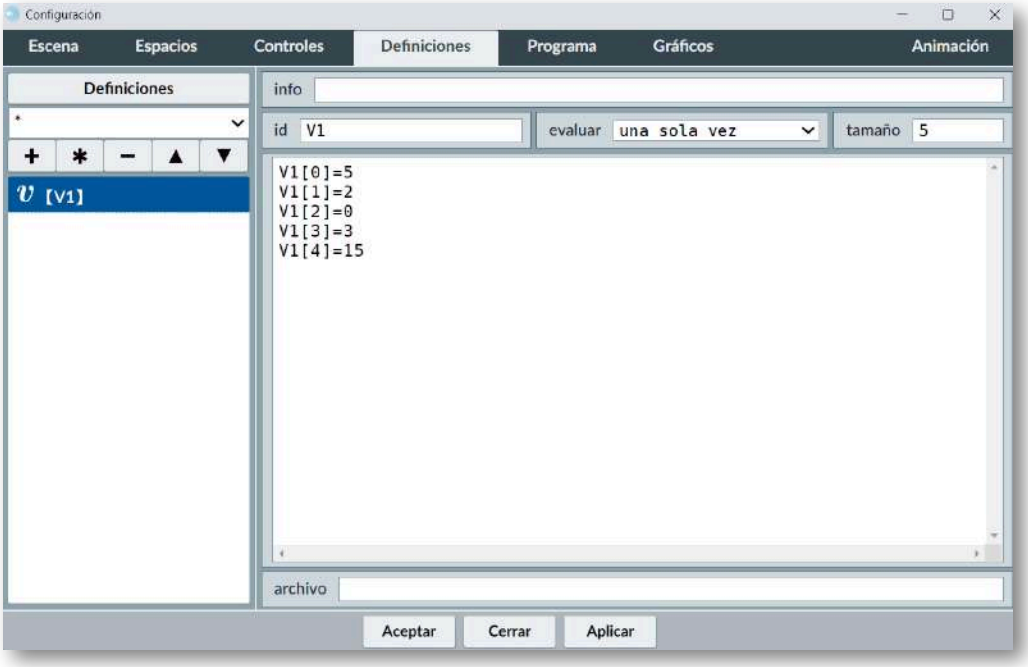


Los valores de los elementos de un vector pueden ser numéricos o de tipo cadena. En este último caso, deben escribirse entre comillas sencillas, por ejemplo: V1[0]='valor'; V1[1]='cosa'; V1[2]='CAMPO'.

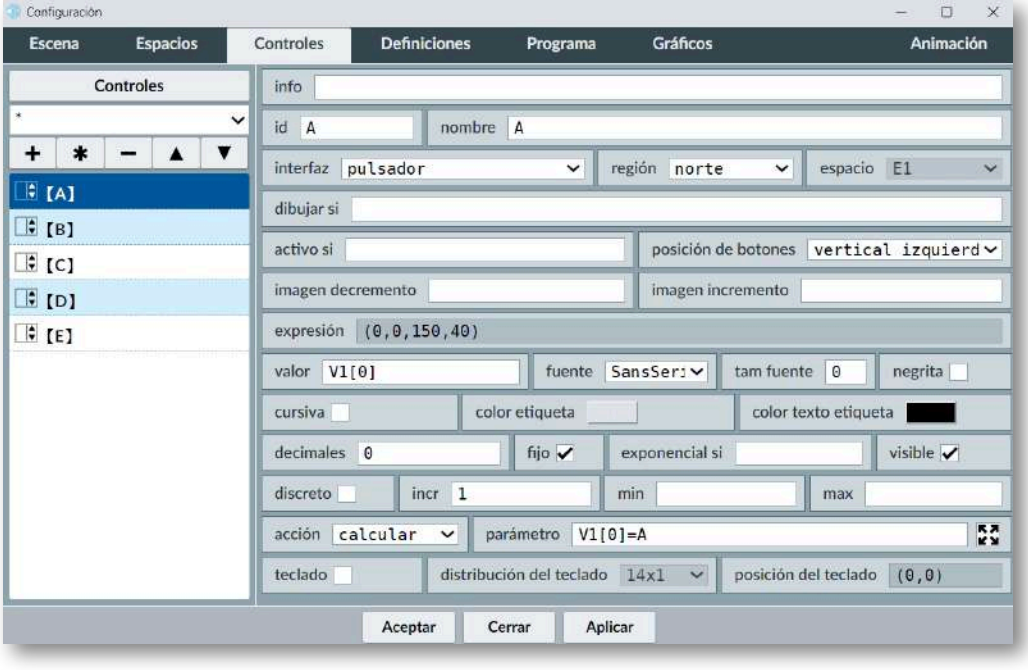


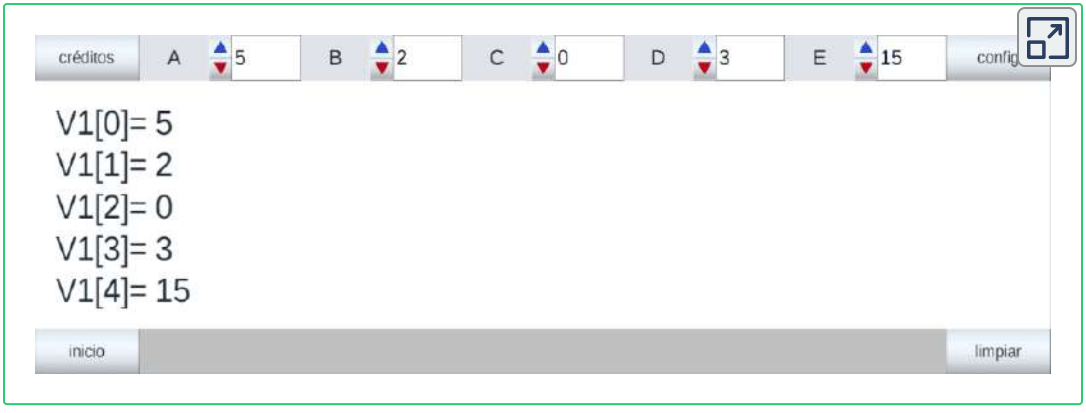
Para referirse a un elemento de un vector, se escribe su identificador seguido del índice entre corchetes. Así, el primer elemento de un vector llamado vec se expresa como vec[0], el tercero como vec[2], y así sucesivamente. El índice también puede ser una variable.

Presentamos ahora otro ejemplo en el que los datos de un vector se introducen mediante controles. Al igual que en los casos anteriores, definimos un vector con cinco elementos.



A continuación, creamos cinco controles, cada uno de los cuales asigna su valor a uno de los elementos del vector correspondiente.





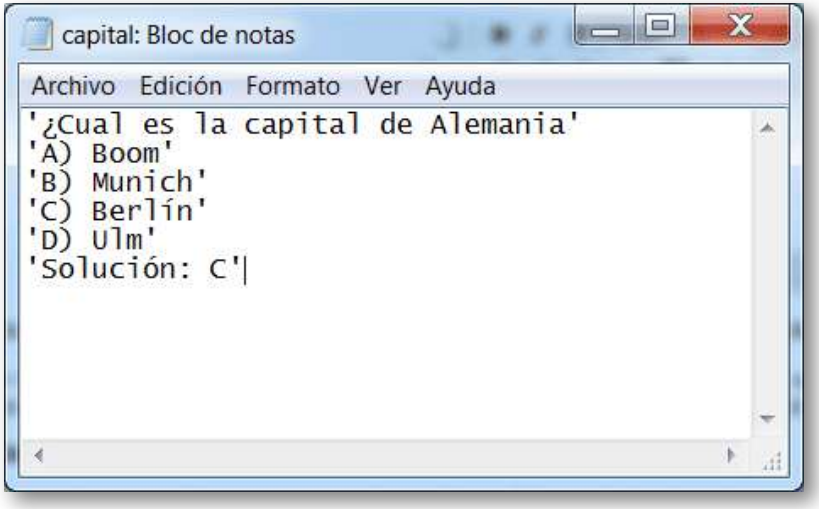
Lectura de datos de los elementos de un vector en ficheros

Los elementos de un vector pueden tomar sus valores desde un fichero de texto externo a la escena. Para ello, en el campo **archivo** se debe indicar la ruta relativa del fichero con respecto a la escena, junto con su nombre y extensión (se recomienda utilizar la extensión **.txt** y codificación UTF-8). El nombre del fichero también puede introducirse mediante una variable o un control, escribiendo su nombre entre corchetes. Por ejemplo, si la variable **fichero** tiene el valor *'ejemplo.txt'*, en el campo archivo se puede escribir **[fichero]**. Si el fichero de texto que contiene el vector no está en la misma ruta que la escena, también se deberá incorporar la ruta relativa en el campo archivo.

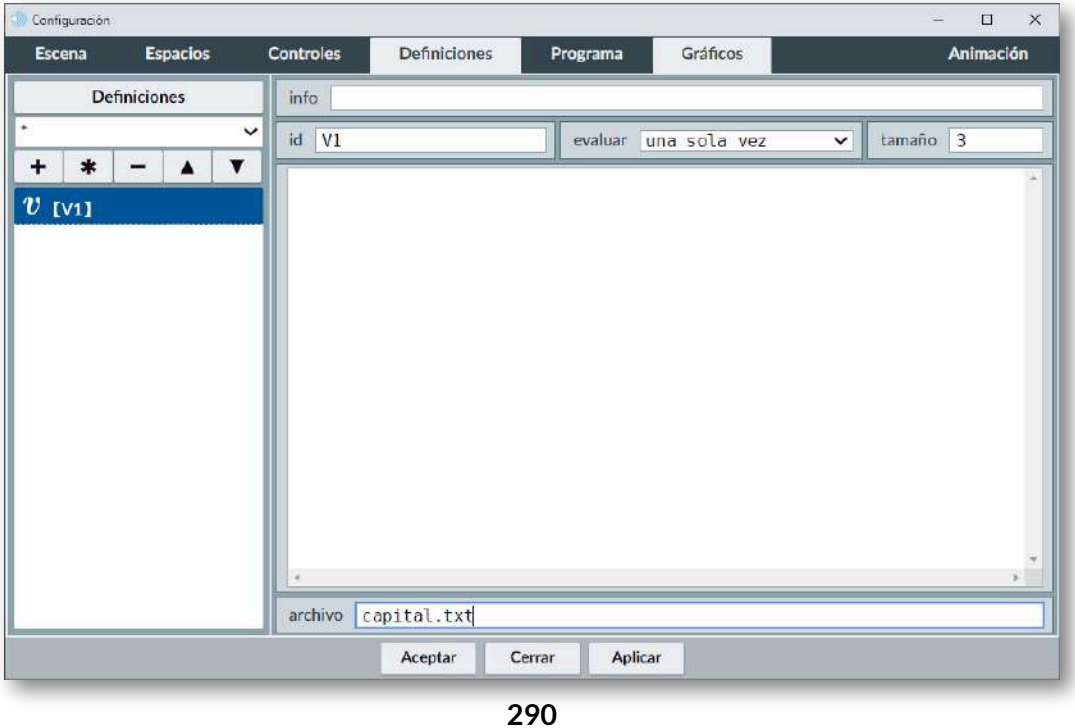
La asignación de valores se realiza de forma secuencial: cada línea del fichero corresponde a un elemento del vector. Así, la primera línea será el valor del primer componente, la segunda línea el del segundo, y así sucesivamente. Si las líneas aparecen entre comillas simples, los valores se interpretarán como texto; en caso contrario, se considerarán numéricos.

El uso de ficheros en los juegos didácticos del Proyecto AJDA resulta especialmente útil, ya que permite cargar preguntas desde distintos archivos, pudiendo seleccionar el nombre del fichero mediante un control o una variable.

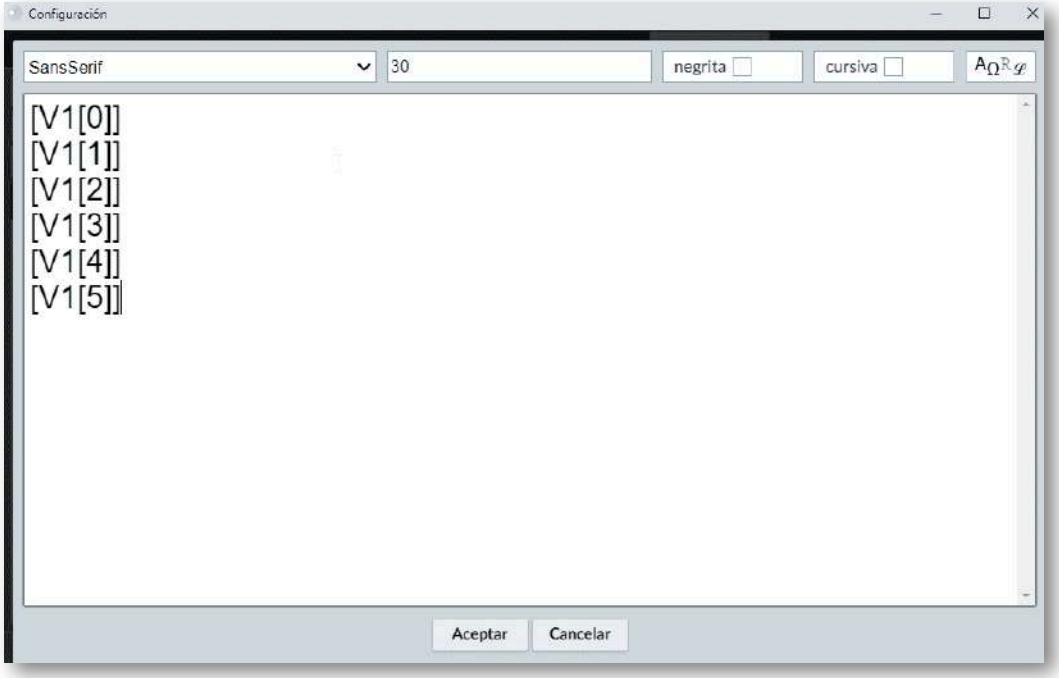
A continuación, presentamos un pequeño ejemplo. Guardamos en un fichero de texto una pregunta con cuatro opciones de respuesta y la solución. El fichero se llama **capital.txt** y su contenido es el siguiente:

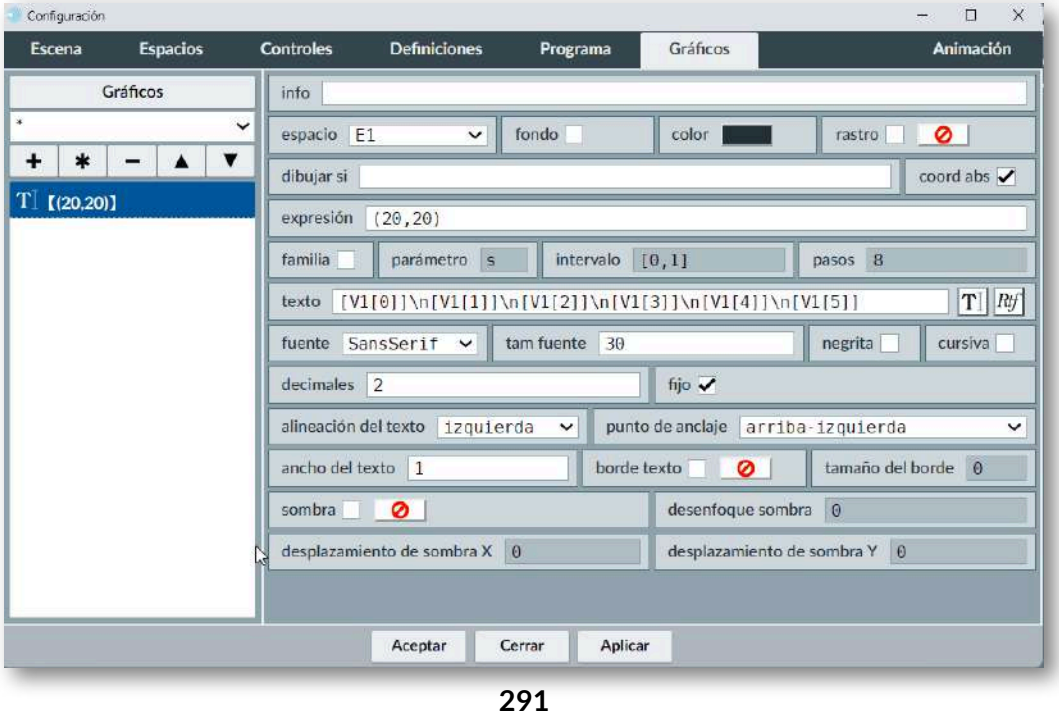


En la escena creamos un vector denominado **V1**, cuyos elementos tomarán sus valores del fichero **capital.txt**.

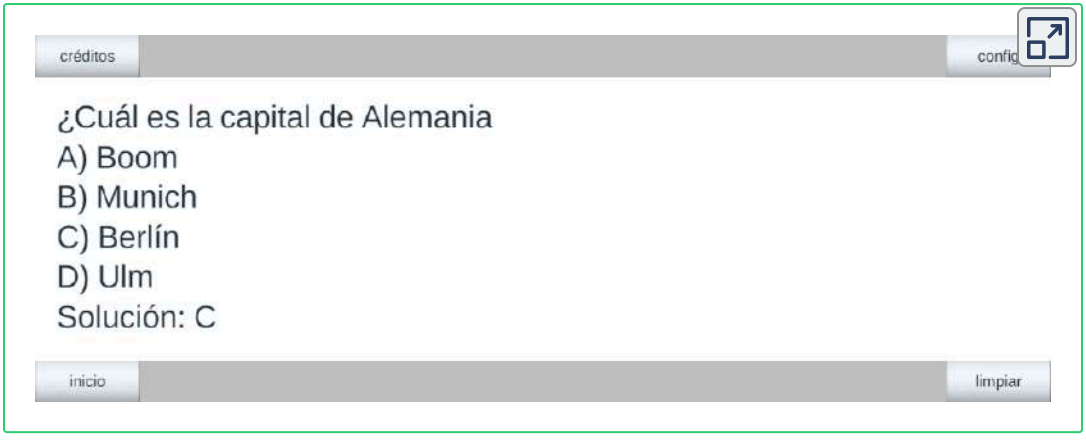


Creamos un control de texto que muestre los valores del vector: **V1[0], V1[1], V1[2], V1[3], V1[4], V1[5]**.





En la escena se visualizará el siguiente resultado:

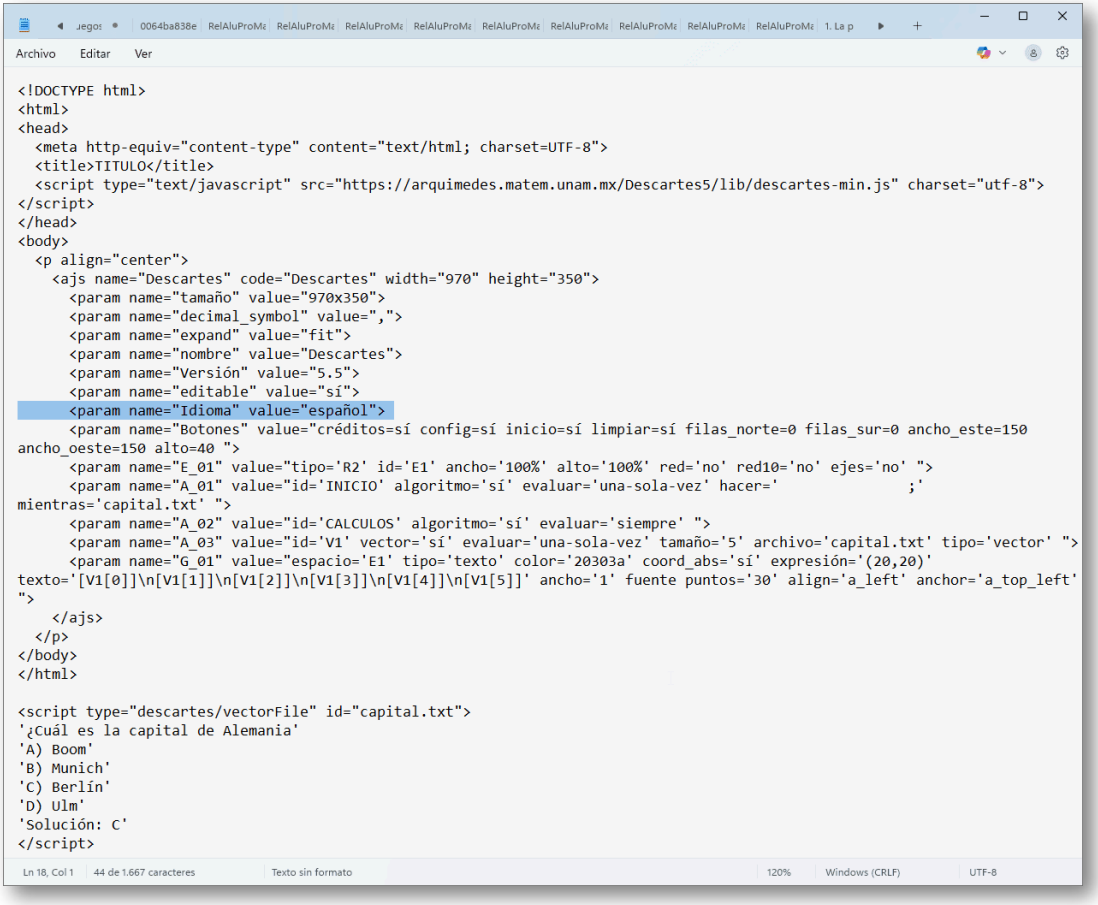


Cuando las escenas se ejecutan desde el propio ordenador (y no desde Internet), la lectura directa de ficheros externos suele estar bloqueada por motivos de seguridad del navegador. Por ello, para que se muestren correctamente los datos de un fichero, es necesario añadirlos al final del código HTML de la escena el contenido del fichero entre las etiquetas:

```
<script type="descartes/vectorFile" id="ruta relativa completa del fichero respecto de la escena">
'Dato 0 alfanumérico'
'Dato 1 alfanumérico'
'Dato 2 alfanumérico'
Dato 3 numérico
Dato 4 numérico
Dato 5 numérico
'Dato 6 alfanumérico'
...
</script>
```

De este modo, el fichero queda integrado en la propia escena. Si en el editor de Descartes está activada la opción **"vector"** dentro del submenú *Agregar al html* del menú *Opciones*, este código se incorporará automáticamente al final del archivo de la escena (de forma análoga sucede con las matrices, bibliotecas y macros).

A continuación, se muestra el código de la escena utilizada en el último ejemplo, donde se han incorporado al final del mismo los datos del vector correspondientes al fichero **capital.txt**, integrados directamente en el código. De este modo, aunque el fichero de texto original no exista como archivo independiente, la escena funcionará correctamente al estar el contenido del vector incluido en ella.



```
Archivo  Editar  Ver

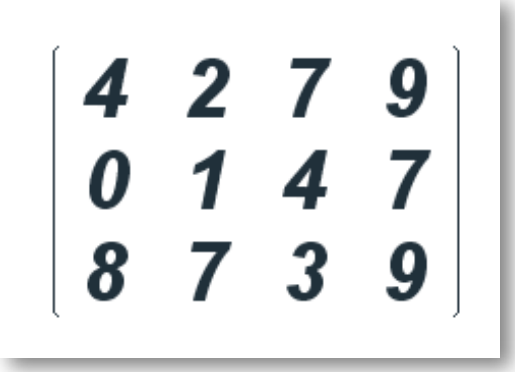
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>TITULO</title>
  <script type="text/javascript" src="https://arquimedes.matem.unam.mx/Descartes5/lib/descartes-min.js" charset="utf-8">
</script>
</head>
<body>
  <p align="center">
    <ajs name="Descartes" code="Descartes" width="970" height="350">
      <param name="tamaño" value="970x350">
      <param name="decimal_symbol" value=",">
      <param name="expand" value="fit">
      <param name="nombre" value="Descartes">
      <param name="Versión" value="5.5">
      <param name="editable" value="sí">
      <param name="Idioma" value="español">
      <param name="Botones" value="créditos=sí config=sí inicio=sí limpiar=sí filas_norte=0 filas_sur=0 ancho_este=150
ancho_oeste=150 alto=40 ">
      <param name="E_01" value="tipo='R2' id='E1' ancho='100%' alto='100%' red='no' red10='no' ejes='no' ">
      <param name="A_01" value="id='INICIO' algoritmo='sí' evaluar='una-sola-vez' hacer='                ;'
mientras='capital.txt' ">
      <param name="A_02" value="id='CALCULOS' algoritmo='sí' evaluar='siempre' ">
      <param name="A_03" value="id='V1' vector='sí' evaluar='una-sola-vez' tamaño='5' archivo='capital.txt' tipo='vector' ">
      <param name="G_01" value="espacio='E1' tipo='texto' color='20303a' coord_abs='sí' expresión='(20,20)'
texto='[V1[0]]\n[V1[1]]\n[V1[2]]\n[V1[3]]\n[V1[4]]\n[V1[5]]' ancho='1' fuente puntos='30' align='a_left' anchor='a_top_left'
">
    </ajs>
  </p>
</body>
</html>

<script type="descartes/vectorFile" id="capital.txt">
'¿Cuál es la capital de Alemania'
'A) Boom'
'B) Munich'
'C) Berlín'
'D) Ulm'
'Solución: C'
</script>

Ln 18, Col 1   44 de 1.667 caracteres   Texto sin formato   120%   Windows (CRLF)   UTF-8
```

4.10 Matrices

Una matriz es similar a un vector, pero en dos dimensiones. Puede contener números, textos, valores de variables o constantes. Se puede considerar como un conjunto de vectores: si una matriz tiene 4 columnas, cada columna se comporta como un vector, mientras que el número de filas indica cuántos elementos tiene cada vector. Por ejemplo, 3 filas implican que cada uno de los cuatro vectores tenga tres elementos.



En el ejemplo, la matriz sería equivalente a definir 4 vectores:

- Primer vector: 4, 0, 8
- Segundo vector: 3, 1, 7
- Tercer vector: 7, 4, 3
- Cuarto vector: 9, 7, 9

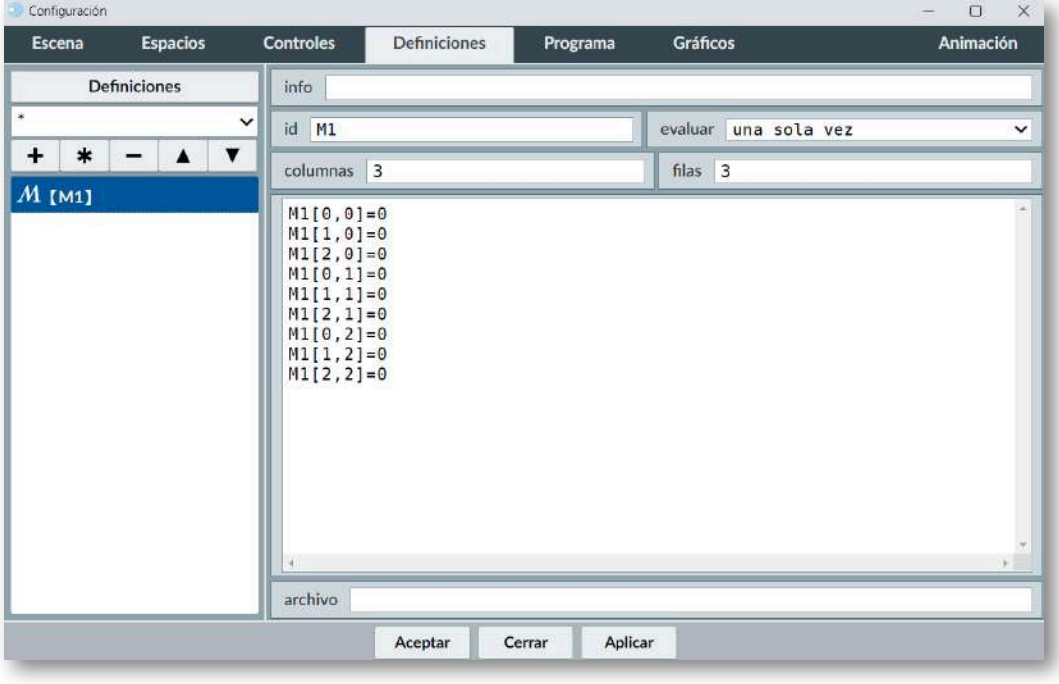
Las matrices, también llamadas arreglos, pueden imaginarse como una tabla con filas y columnas, similar a una cajonera donde cada cajón guarda un valor.

A diferencia de los vectores, para referirse a un elemento de la matriz se necesitan dos índices separados por una coma. Por ejemplo, en una matriz M, M[3,7] representa la entrada de la cuarta columna y la octava fila (recordando que los índices comienzan en cero).

De forma genérica, una matriz M tiene i columnas y j filas, y cada elemento se representa como M[i,j], estando en la columna i y la fila j.

Una matriz posee un identificador o nombre, el número de columnas y filas, y un parámetro evaluar que indica si se calcula una sola vez al inicio de la escena. El área de texto del panel de configuración sirve para asignar los valores o inicializar los elementos (también es posible usar un algoritmo para asignar valores). Por defecto, todos los elementos de una matriz se inicializan con cero, y cada valor se indica usando el nombre de la matriz seguido de los índices de columna y fila entre corchetes.

A continuación, presentamos los parámetros de la definición de una matriz, además del habitual info para comentarios:



- **id:** Campo de texto donde se introduce el identificador de la matriz. Para referirse a un elemento, se usa el identificador seguido de los índices de columna y fila entre corchetes. Ejemplo: M[1,2].

- **evaluar:** Menú con una opción: una sola vez (las instrucciones se ejecutan al cargar la escena). La actualización de la matriz suele hacerse mediante funciones.
- **columnas:** Número de columnas de la matriz.
- **filas:** Número de filas de la matriz.
- **panel de asignaciones:** Área de texto para introducir las asignaciones o instrucciones que inicializan los elementos de la matriz. Por defecto, algunas entradas se inicializan en cero. También se puede inicializar mediante una función.
- **archivo:** Ruta relativa a un archivo de texto con los datos para llenar la matriz. Los valores deben estar separados por comas o punto y coma según el signo decimal de la escena, formando un archivo CSV. Ejemplos con punto decimal:

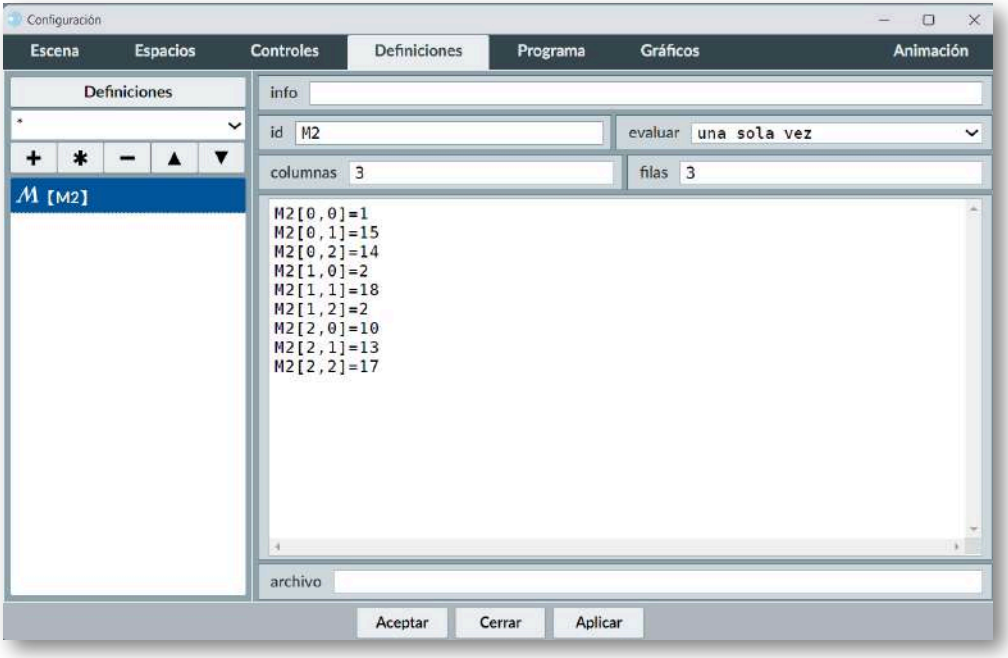
  ```
  1,2,3,4,5
  'a',b,c,d,'e'
  10.1,20.2,30.3,40.4,50.5
  ```

  Ejemplos con coma decimal:

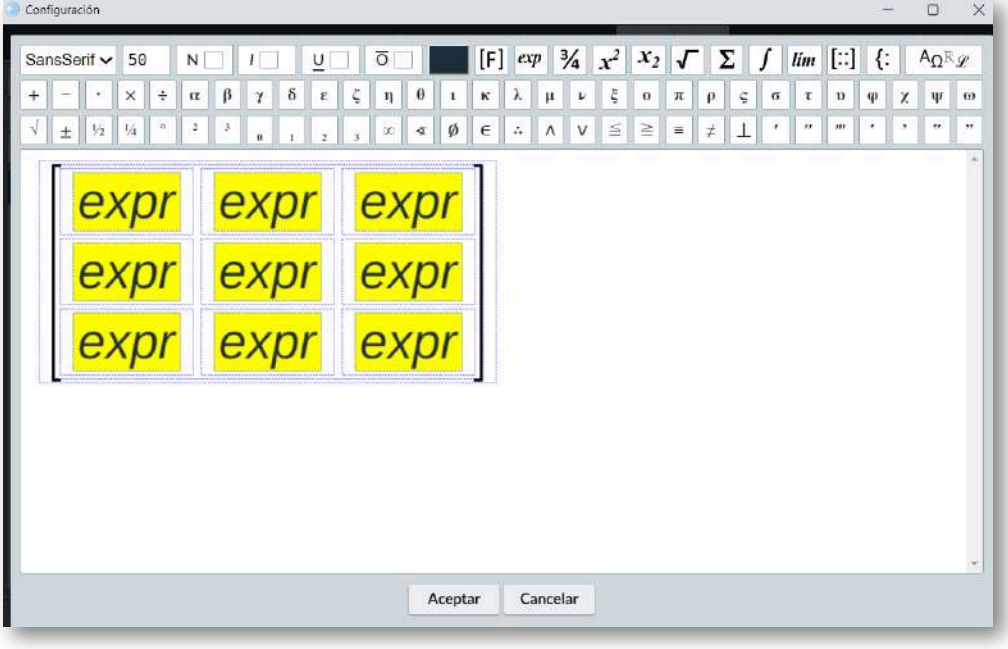
  ```
  1;2;3;4;5
  'a';b;c;d;'e'
  10,1;20,2;30,3;40,4;50,5
  ```

  Las hojas de cálculo en ocasiones exportan las cadenas de texto con comillas dobles. Sin embargo, es posible tener cadenas de texto en el archivo de texto flanqueadas por comillas sencillas o dobles, ya que internamente el programa convierte las dobles en sencillas.

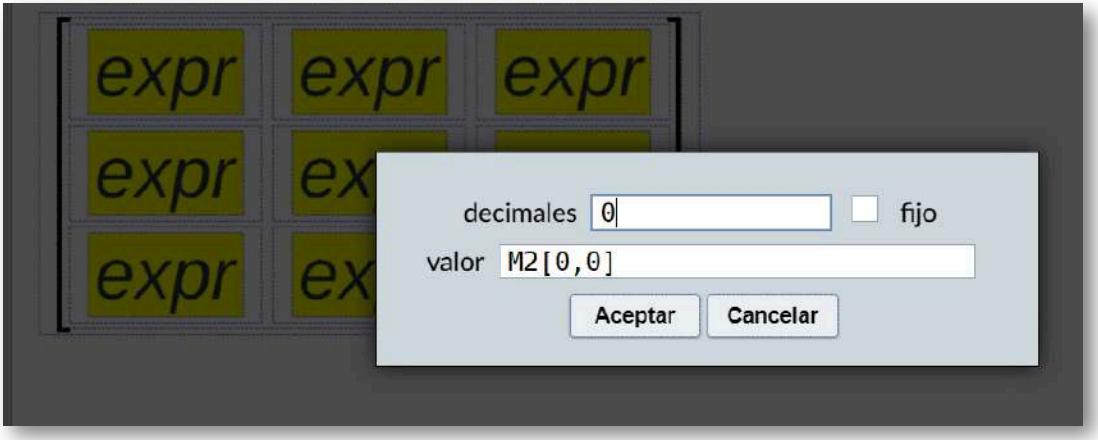
  Como ejemplo, crearemos una matriz llamada **M2**. Esta matriz estará compuesta por tres filas y tres columnas, con los siguientes valores: **Columna 1:** M2[0,0]=1; M2[0,1]=15; M2[0,2]=14. **Columna 2:** M2[1,0]=2; M2[1,1]=18; M2[1,2]=2. **Columna 3:** M2[2,0]=10; M2[2,1]=13; M2[2,2]=7.



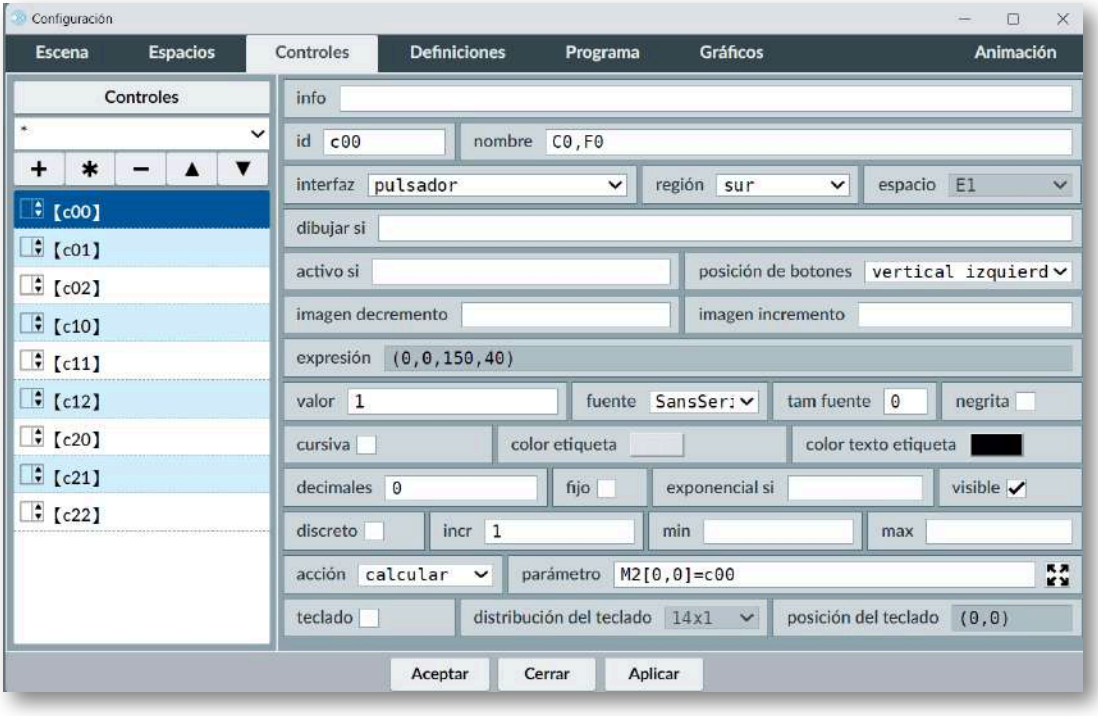
A continuación, utilizando el editor de textos avanzado, mostramos los valores de la matriz con un formato gráfico tipo matriz: M2[0,0], M2[1,0], M2[2,0], M2[0,1], M2[0,2], M2[1,2], M2[2,2].



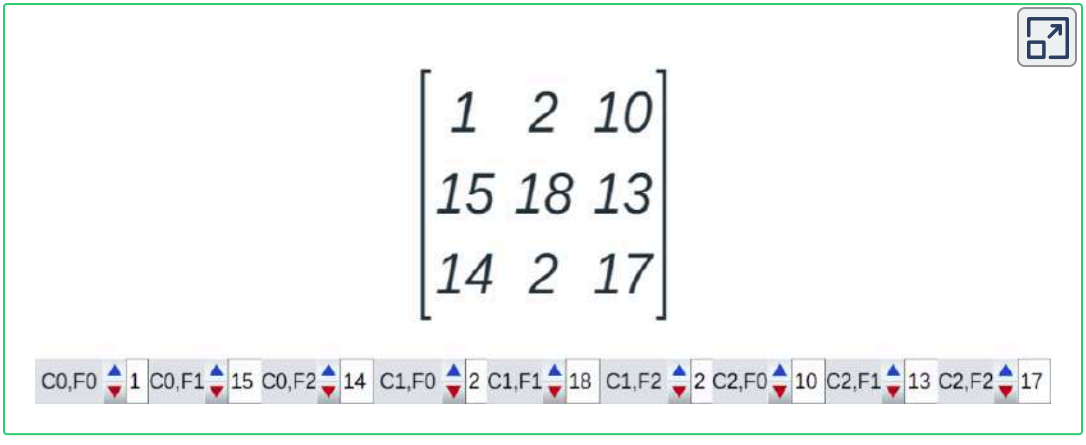
Cada elemento de la matriz, representado como *expr*, es una variable que se corresponde con cada uno de los elementos de la matriz **M2** previamente definida.



Creamos nueve controles tipo pulsador, cada uno de los cuales asignará el valor a uno de los elementos de la matriz.



La escena obtenida es la siguiente:

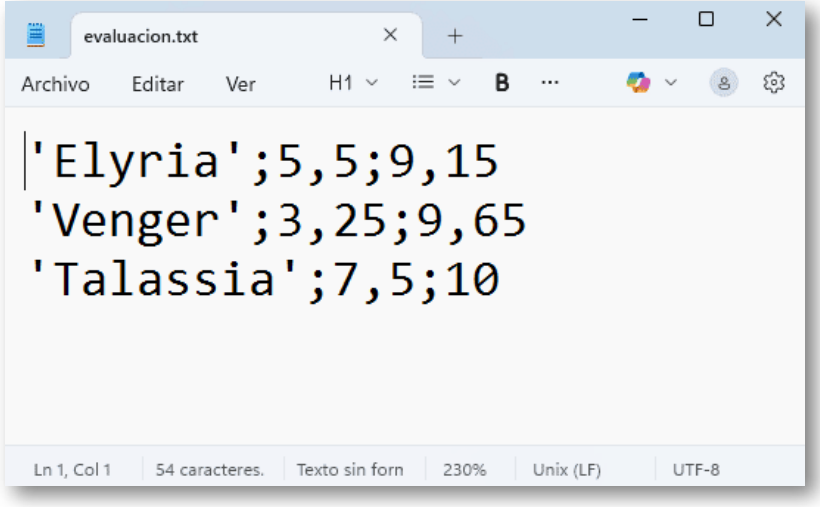


Lectura de datos de los elementos de una matriz en ficheros

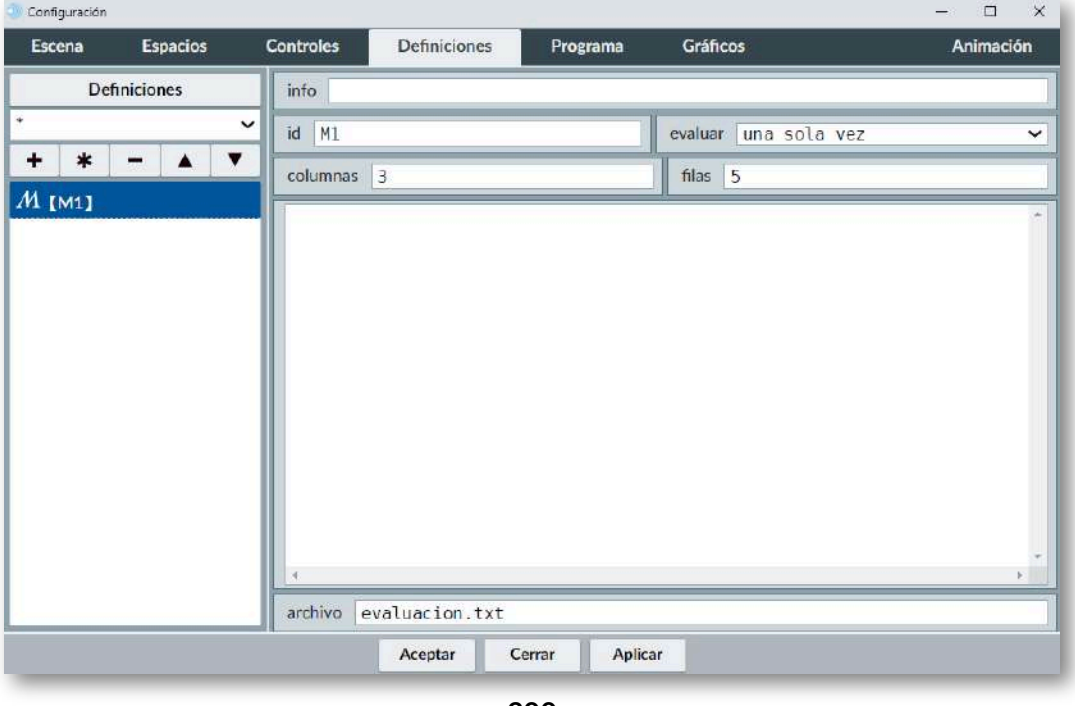
Los elementos de una matriz pueden tomar sus valores desde un fichero de texto externo a la escena. Para ello, en el campo **archivo** se debe indicar la ruta relativa del fichero con respecto a la escena, junto con su nombre y extensión (se recomienda utilizar la extensión **.txt** y codificación UTF-8). El nombre del fichero también puede introducirse mediante una variable o un control, escribiendo su nombre entre corchetes. Por ejemplo, si la variable **fichero** tiene el valor *'evaluacion.txt'*, en el campo archivo se puede escribir **[fichero]**. Si el fichero de texto que contiene el vector no está en la misma ruta que la escena, también se deberá incorporar la ruta relativa en el campo archivo.

La asignación de valores se realiza de forma secuencial: cada línea del fichero corresponde a una fila de la matriz. Los elementos de cada fila se separan mediante punto y coma cuando la cifra decimal utiliza la coma, o mediante comas cuando la cifra decimal utiliza el punto. Así, la primera línea contiene los valores de la primera fila de la matriz; la segunda línea, los de la segunda fila; y así sucesivamente. Si los datos aparecen entre comillas simples, los valores se interpretarán como texto; en caso contrario, se considerarán numéricos.

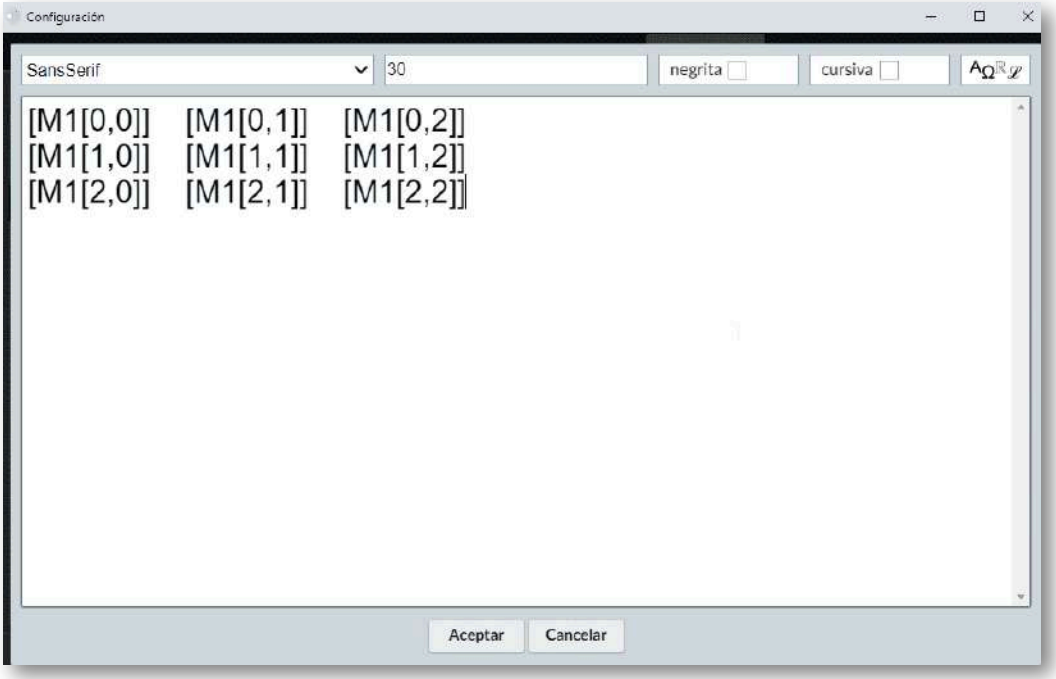
Presentamos un pequeño ejemplo. Guardamos en un fichero de texto llamado **evaluacion.txt** las calificaciones correspondientes a dos exámenes de tres alumnos. Su contenido es el siguiente:

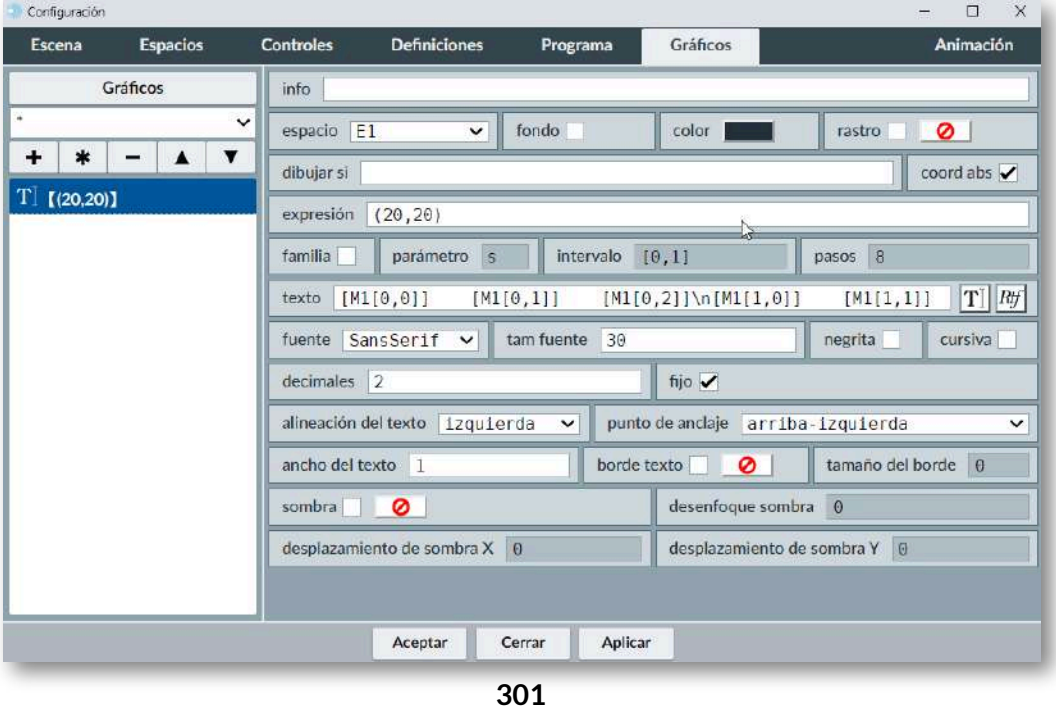


En la escena creamos una matriz denominada **M2**, cuyos elementos tomarán sus valores del fichero **evaluacion.txt**.

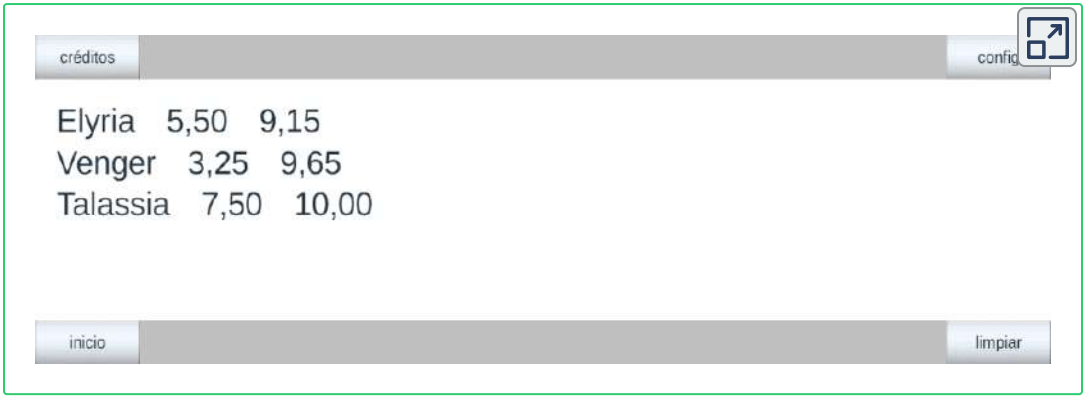


Creamos un control de texto que muestre los valores de la matriz con el formato propio de esta.





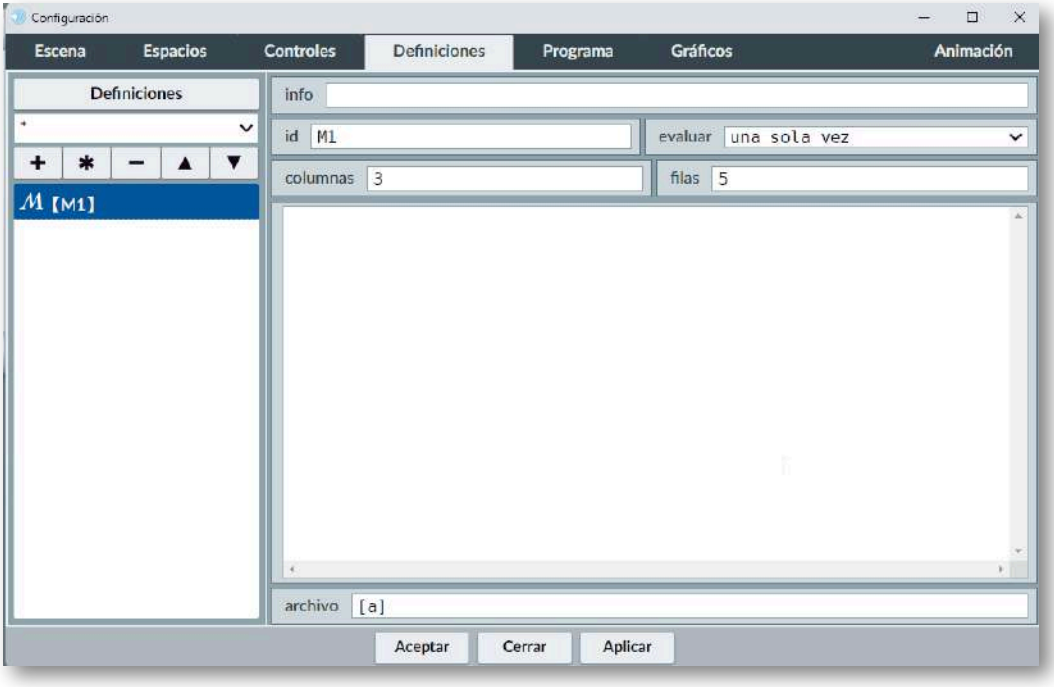
En la escena se visualizará el siguiente resultado:



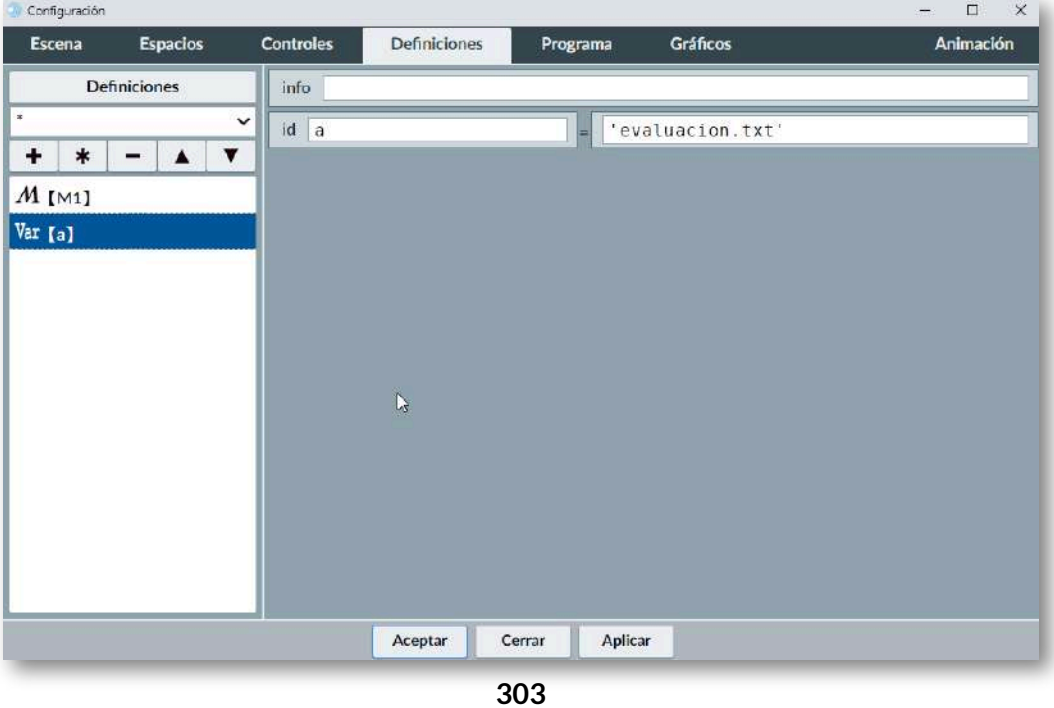
Cuando las escenas se ejecutan desde el propio ordenador (y no desde Internet), la lectura directa de ficheros externos suele estar bloqueada por motivos de seguridad del navegador. Por este motivo, para que los datos se muestren correctamente, es necesario añadir al final del código HTML de la escena el contenido del fichero entre las etiquetas correspondientes. Este proceso lo realiza automáticamente el editor de Descartes si, en las opciones, está activada la casilla **agregar al HTML**; en este caso, la opción **matriz**.

```
<script type="descartes/matrixFile" id="ruta relativa completa del fichero respecto de la escena">
'Dato 0 alfanumérico';Dato 2 numérico;Dato 3 numérico
Dato 4 numérico';Dato 1 alfanumérico';'Dato 2 alfanumérico'
Dato 5 numérico;Dato 6 numérico;Dato 7 numérico
...
</script>
```

El campo **archivo** puede introducirse mediante una variable, por ejemplo, **a**, que debe escribirse entre corchetes.

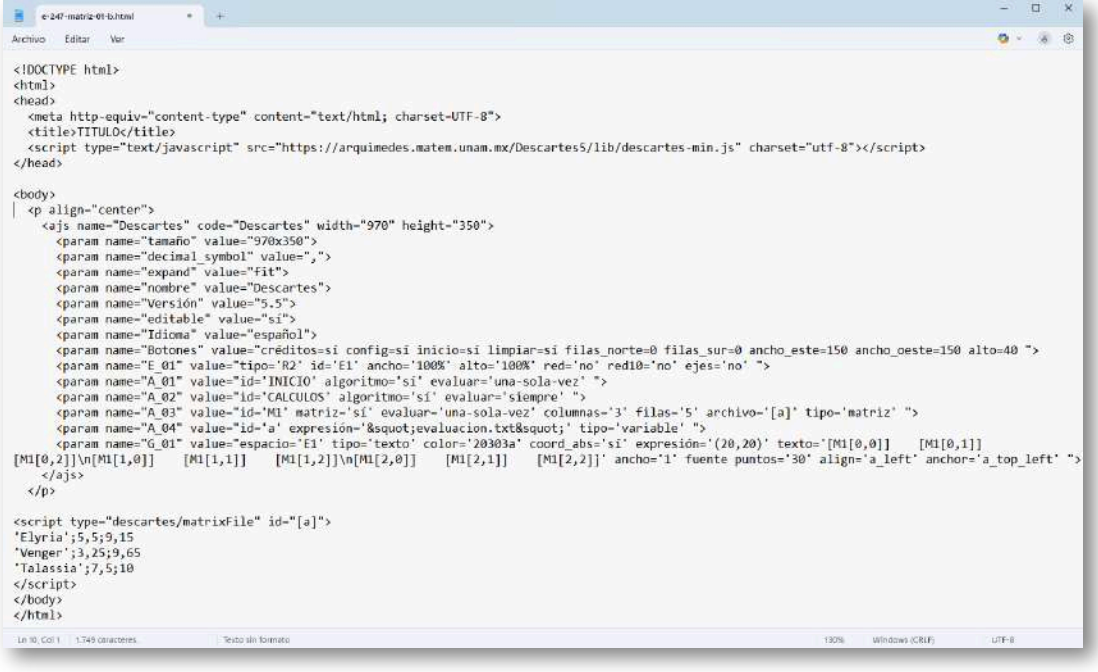


Definimos la variable **a** y le asignamos como valor el nombre del fichero (y, si es necesario, también su ruta).



De este modo, el fichero queda integrado en la propia escena. Si en el editor de Descartes está activada la opción **"matriz"** dentro del submenú *Agregar al HTML* del menú *Opciones*, este código se incorporará automáticamente al final del archivo de la escena (de forma análoga ocurre con matrices, bibliotecas y macros). Asimismo, cuando la matriz está incluida al final del fichero dentro de su propio código, la escena también genera automáticamente el archivo correspondiente.

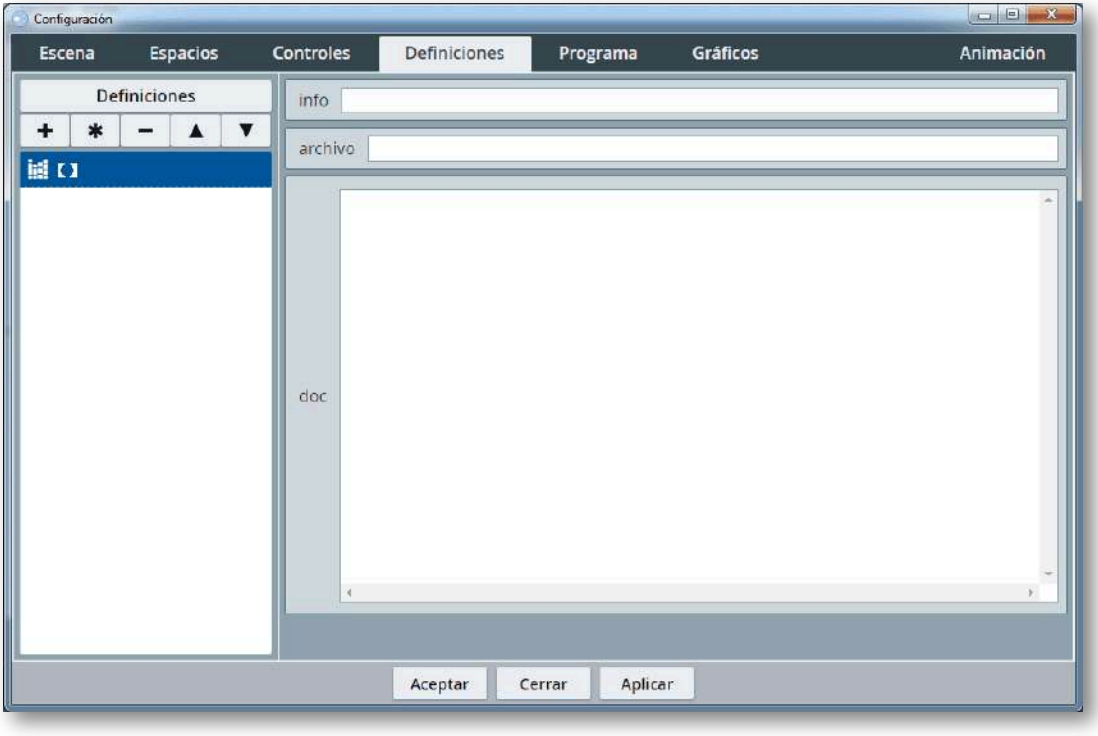
A continuación, se muestra el código de la escena utilizada en el último ejemplo, donde se han añadido al final los datos de la matriz correspondientes al fichero **evaluacion.txt**, integrados directamente en el código. De esta manera, aunque el fichero de texto original no exista como archivo independiente, la escena funcionará correctamente, ya que el contenido del vector está incluido en ella. En este caso, el nombre (y la ruta) del fichero están definidos mediante la variable **a**; por ello, es esta variable la que aparece entre corchetes en el lugar del nombre del archivo.



4.11 Biblioteca

En ocasiones, una escena en Descartes puede contener una gran cantidad de elementos en el selector Definiciones, lo que puede dificultar la búsqueda de elementos específicos. Aunque todos los elementos sean esenciales para la escena, muchos no requieren modificaciones posteriores y ocupan líneas de código innecesarias. Además, puede suceder que un conjunto de definiciones sea común a varias escenas; en estos casos, colocarlas en un archivo de biblioteca permite que sean accesibles desde distintas escenas de manera centralizada.

Por tanto, resulta útil extraer del archivo de la escena aquellas definiciones que se mantendrán fijas y ubicarlas en un bloque de texto llamado biblioteca. Los campos de configuración de este apartado son:



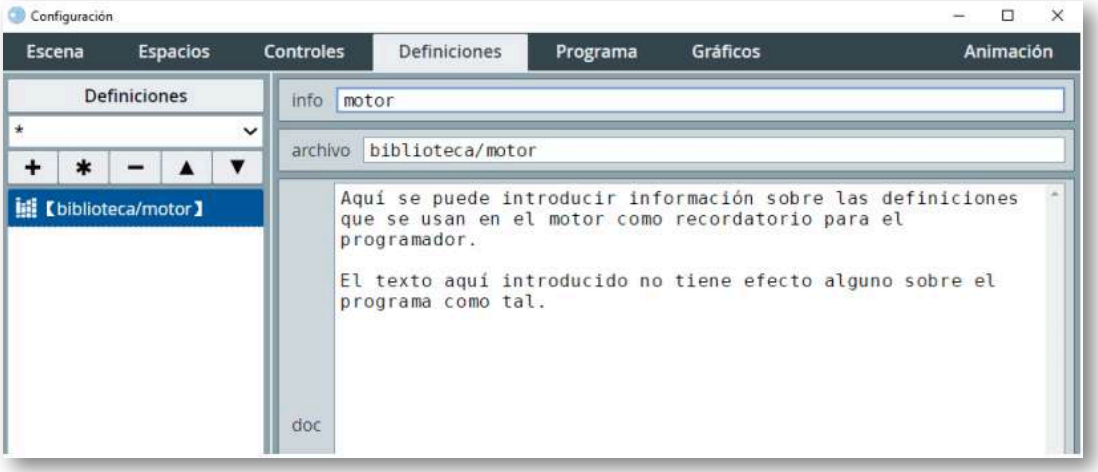
- **info:** Comentario para el programador que no se muestra en pantalla.
- **archivo:** Campo de texto donde se introduce la ruta del archivo que contendrá las definiciones a mantener fuera del código principal de la escena. La ruta será relativa a la carpeta donde se guarda el archivo HTML de la escena.
- **doc:** Panel de texto donde se pueden incluir notas o recordatorios sobre el material almacenado en la biblioteca. El contenido no es interpretado por Descartes y se considera únicamente como texto.

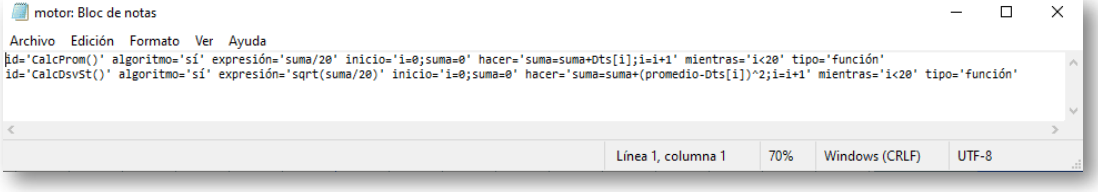
Una vez creada la biblioteca, puede cargarse y guardarse dentro del archivo HTML de la escena, y sus elementos pueden editarse directamente desde el editor de configuraciones de Descartes.

Por tanto, una **biblioteca** de Descartes es un archivo de texto que contiene definiciones de escenas de Descartes, tales como funciones, variables, vectores o matrices, que pueden reutilizarse en múltiples escenas sin necesidad de ser modificadas. De esta manera, es posible mantener un conjunto de definiciones comunes a varias escenas, evitando la repetición y facilitando la organización y mantenimiento del código. Además, las bibliotecas permiten actualizar las definiciones en un solo archivo y que los cambios se reflejen automáticamente en todas las escenas que las utilicen, lo que ahorra tiempo y reduce errores.

Presentaremos un ejemplo en el que definimos una biblioteca en un archivo de texto llamado motor, el cual contiene dos funciones: CalcProm() y CalcDsvSt(). Esta biblioteca puede ser reutilizada en diferentes escenas que necesiten acceder a dichas funciones, evitando tener que redefinirlas en cada escena individualmente.

En este ejemplo la biblioteca motor contiene dos funciones, aunque en escenas reales podría haber muchas más.





Es importante usar archivos codificados en UTF-8 para evitar errores al integrarlos en el HTML.

Variables locales con el mismo nombre en distintas funciones no generan conflictos mientras no se ejecuten simultáneamente. Para proteger variables dentro de una función, se deben declarar en el parámetro local. Además, algunas funciones dependen del orden de ejecución; por ejemplo, CalcDsvSt() debe ejecutarse después de CalcProm() para calcular correctamente la desviación estándar.

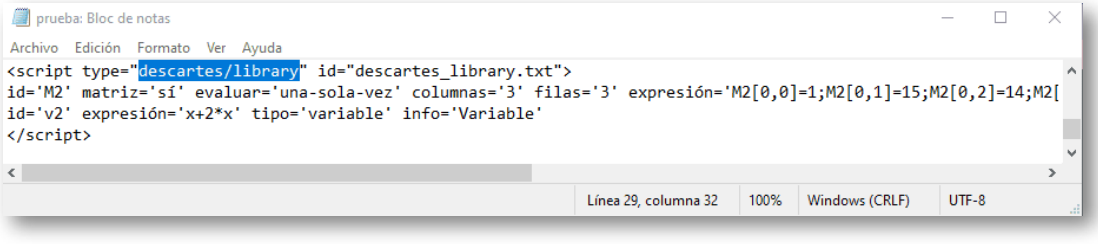
Una vez incluida la biblioteca, sus funciones se pueden revisar y editar desde el panel de Definiciones, y los cambios se guardan tanto en la escena como en el archivo de la biblioteca. El filtro de Definiciones permite mostrar solo las funciones de una biblioteca o todas las definiciones de la escena, manteniendo un orden más claro del código. Se recomienda evitar duplicar nombres de funciones entre la biblioteca y la escena para prevenir conflictos.

Cuando las escenas de Descartes se utilizan en el equipo local, para que las **bibliotecas** funcionen correctamente es necesario añadir su contenido al final del código HTML de la escena, entre las siguientes etiquetas:

```
<script type="descartes/library" id="ruta relativa completa del fichero de la biblioteca respecto de la escena">
Definición 1 (por ejemplo una variable)
Definición 2 (por ejemplo una función)
Definición 3 (por ejemplo un vector)
Definición 4 (por ejemplo una matriz)
Definición 5 (por ejemplo otra función)
Definición 6 (por ejemplo otro vector)
...
</script>
```

De este modo, la biblioteca queda integrada en la propia escena. Si en el editor de Descartes está activada la opción **"biblioteca"** dentro del submenú *Agregar al html* del menú *Opciones*, el código correspondiente se añadirá automáticamente al final del archivo de la escena.

A continuación, se muestra un ejemplo de una biblioteca incorporada al final del código de una escena (el código de la escena va antes del representado entre las etiquetas ajs).

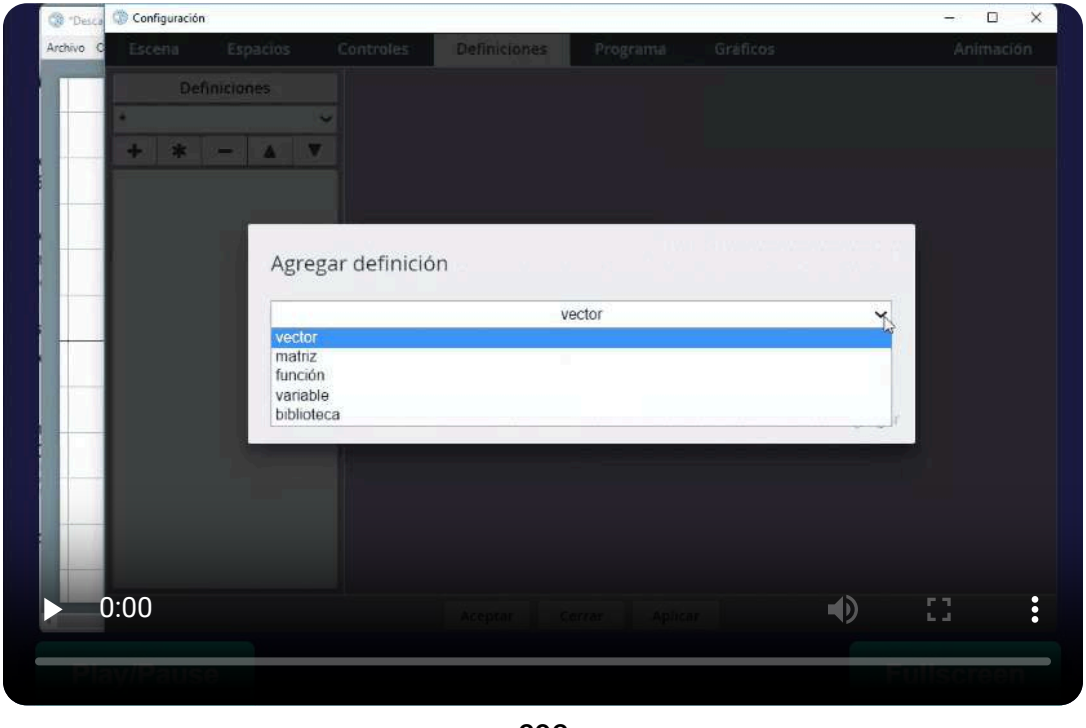


4.12 Resumen sobre definiciones

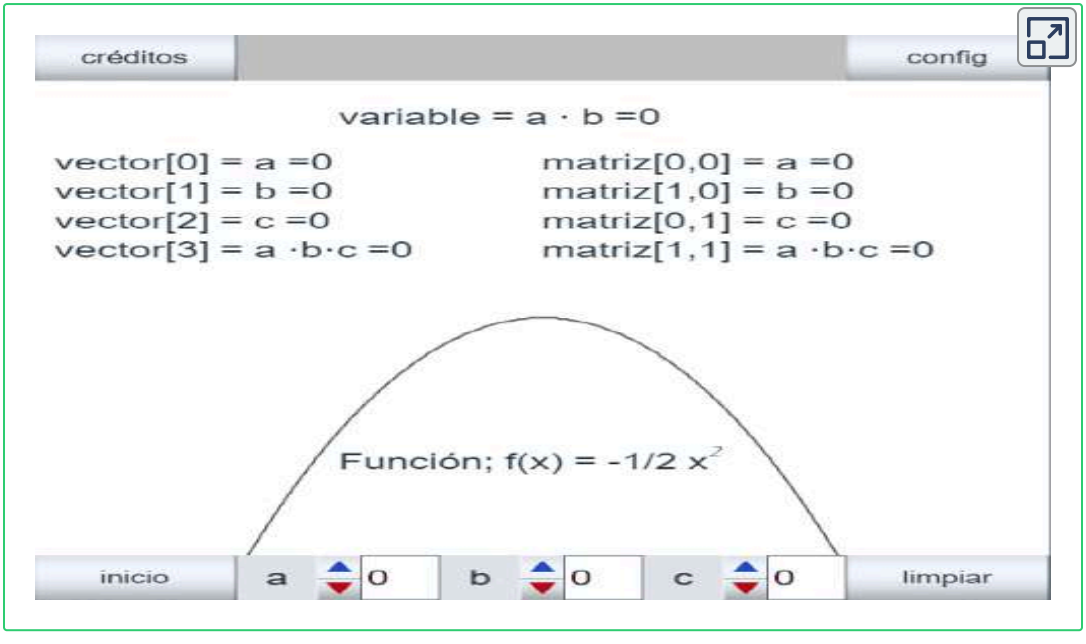
Hemos visto que las definiciones recogen buena parte de la programación interna que se desarrolla en Descartes. Incluyen variables, que pueden almacenar un valor o una expresión; vectores, entendidos como variables capaces de contener una secuencia de valores; matrices, que amplían el concepto de vector más allá de una única hilera; funciones, formadas por conjuntos de instrucciones que pueden ejecutarse cuando se desee e incluso de manera cíclica; y bibliotecas, que agrupan varias de estas definiciones con el objetivo de mantener el código del selector organizado y estructurado.

En este apartado vamos a presentar las diferentes definiciones de Descartes de forma conjunta.

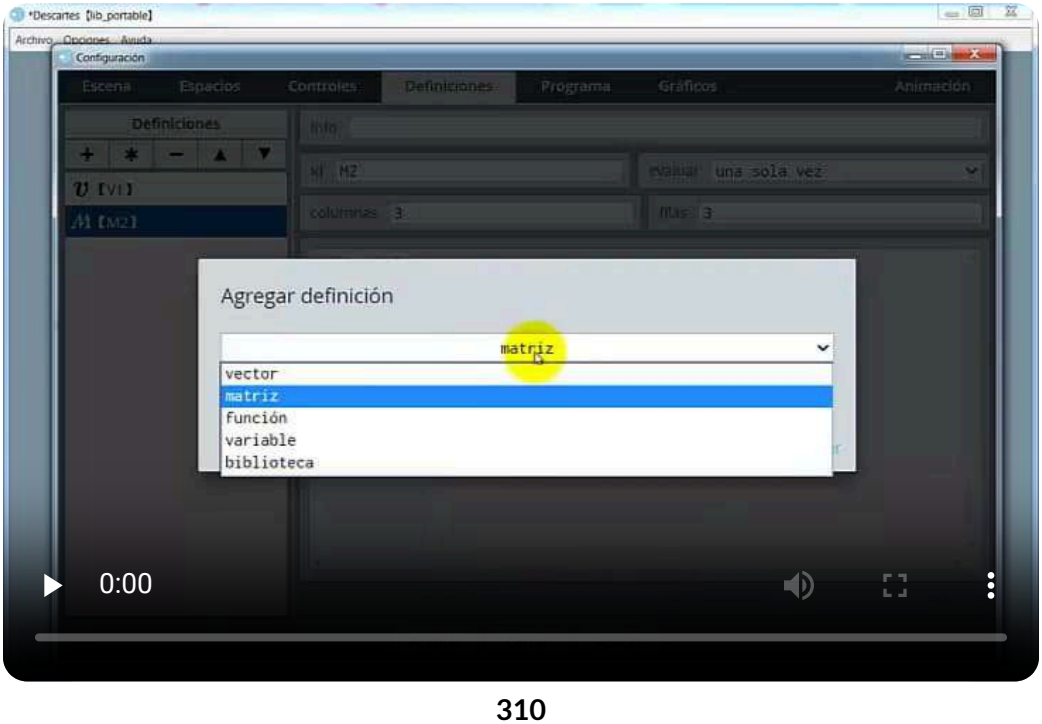
En primer lugar, presentamos un vídeo resumen de la programación con definiciones.



En segundo lugar, incluimos una escena con diferentes definiciones:



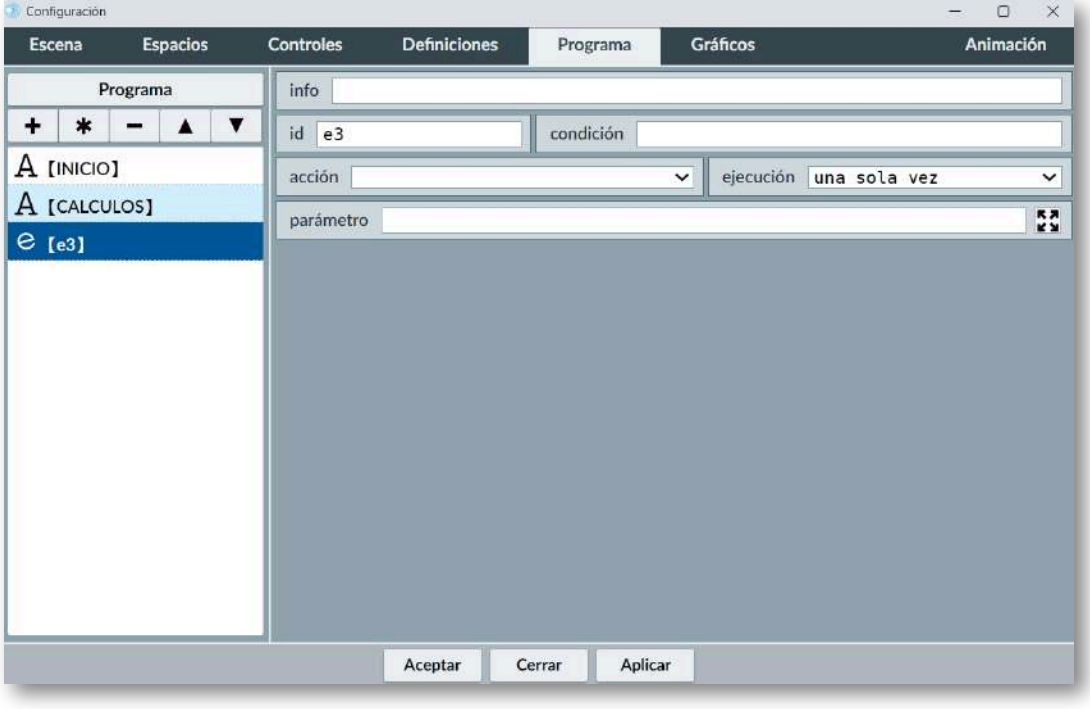
Finalmente incluimos un vídeo resumen sobre definiciones.



4.13 Eventos

Un evento es una acción, o un conjunto de acciones, que se ejecuta cuando se cumple una determinada condición. Además, es posible establecer la frecuencia con la que dichas acciones se llevarán a cabo, es decir, los eventos pueden configurarse para ejecutarse únicamente la primera vez que se cumpla la condición, cada vez que esta se verifique o de forma continua mientras la condición permanezca activa. Las acciones disponibles son las mismas que pueden ejecutarse mediante el uso de controles.

Los eventos se encuentran en el selector **Programa** y los campos que pueden definirse en ellos, además de *info*, son los siguientes:



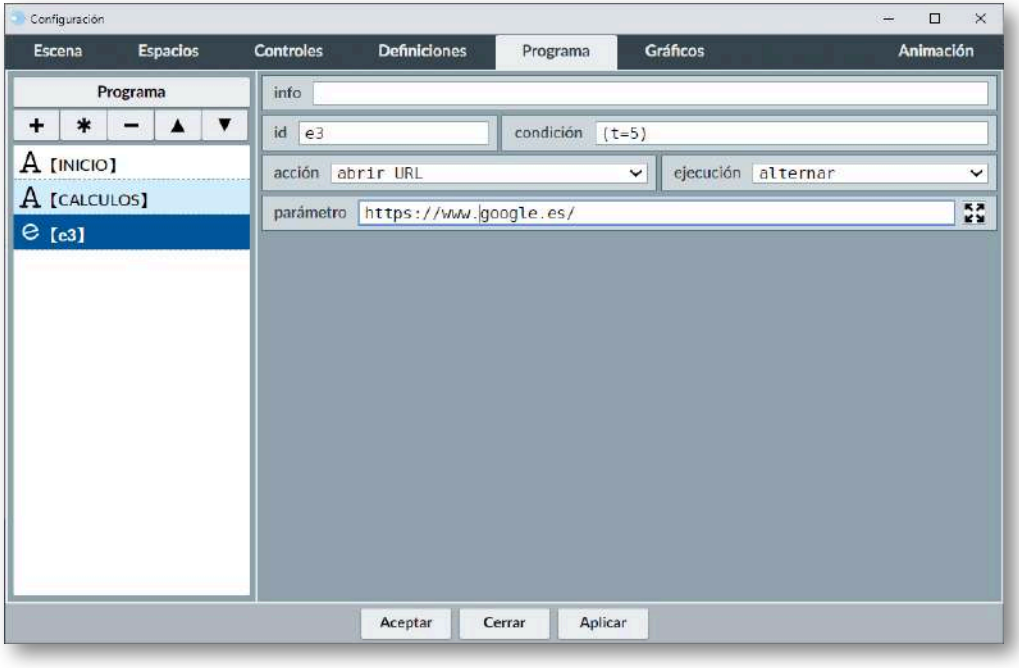
- **id:** campo de texto en el que se introduce el identificador del evento. Se utiliza principalmente para que el programador pueda localizarlo con facilidad y, por lo general, no se referencia en otras partes del programa.

- **condición:** campo de texto donde se especifica la condición que debe cumplirse para que el evento se ejecute. Cuando la condición es verdadera, el evento se activa.
- **acción:** menú en el que se selecciona la acción que se ejecutará si se cumple la condición. Es el mismo menú de acciones disponible en los controles, por lo que cualquier acción posible en un control también puede asociarse a un evento.
- **ejecución:** menú que permite determinar cuándo debe ejecutarse la acción en función del comportamiento de la condición. Sus opciones son:
    - **una sola vez:** la acción se ejecuta únicamente la primera vez que la condición pasa de falsa a verdadera.
    - **alternar:** la acción se ejecuta cada vez que la condición cambia de falsa a verdadera.
    - **siempre:** la acción se ejecuta en todo momento mientras la condición sea cierta, y no solo cuando cambia de estado.
- **parámetro:** campo de texto donde se introducen las instrucciones que se ejecutarán si la acción es *calcular*; la dirección URL si la acción es *abrir URL*; la dirección de la escena si la acción es *abrir Escena*; o la ruta del archivo si la acción es *reproducir*. Su funcionamiento es análogo al de los controles numéricos e incluye un botón que abre una ventana de edición para facilitar la introducción del contenido.

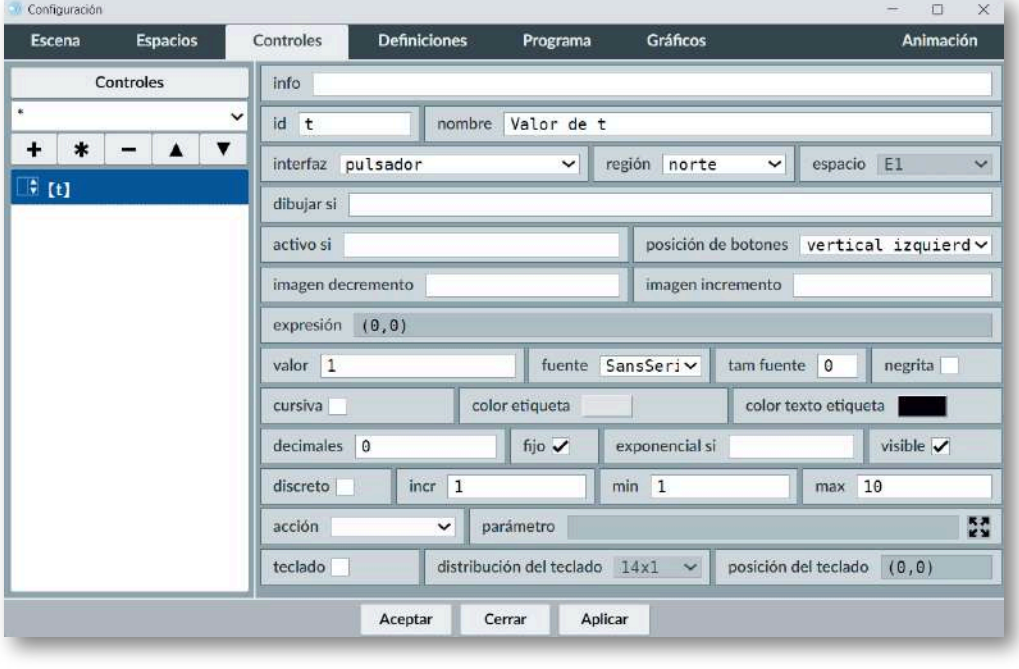
Los eventos resultan especialmente útiles cuando se desea que un botón controle una animación y, al mismo tiempo, ejecute otras instrucciones que no se quieren incluir dentro de la animación. En este caso, la variable que controla la animación puede asociarse a un evento cuya acción sea iniciar o detener dicha animación.

Pondremos un ejemplo sencillo: cuando se cumple la condición de que *t* sea igual a 5, se deberá abrir la página web del buscador **Google**. Esta acción se ejecutará cada vez que la condición se cumpla.

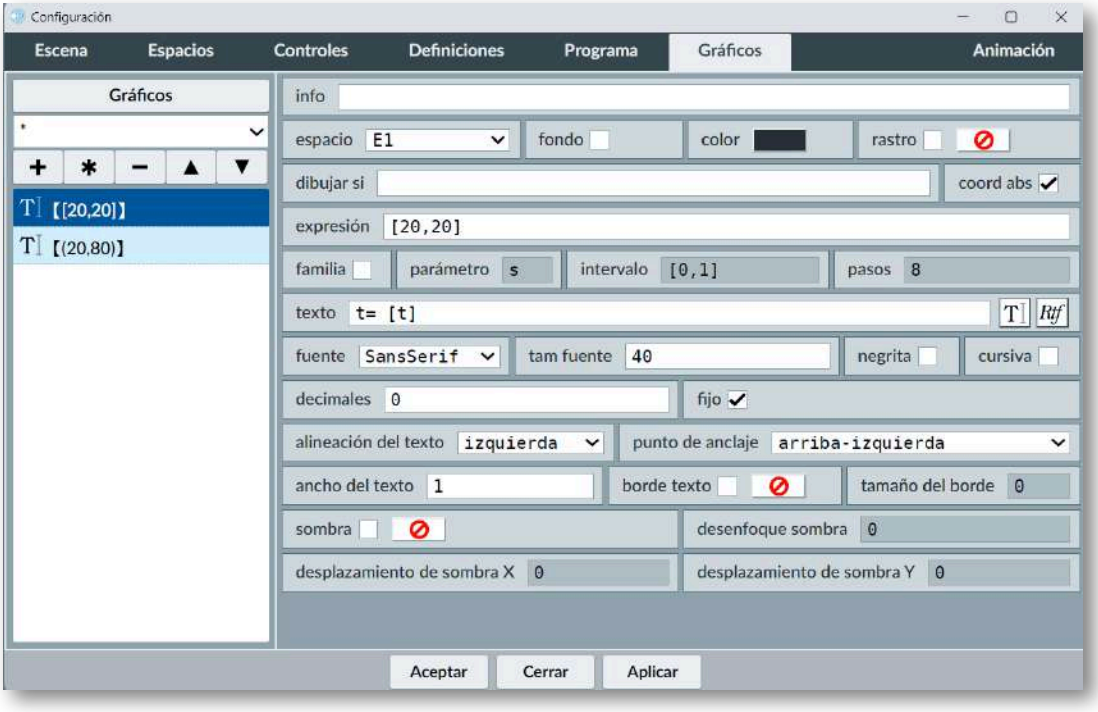
Definimos un evento que cumpla los requisitos indicados.



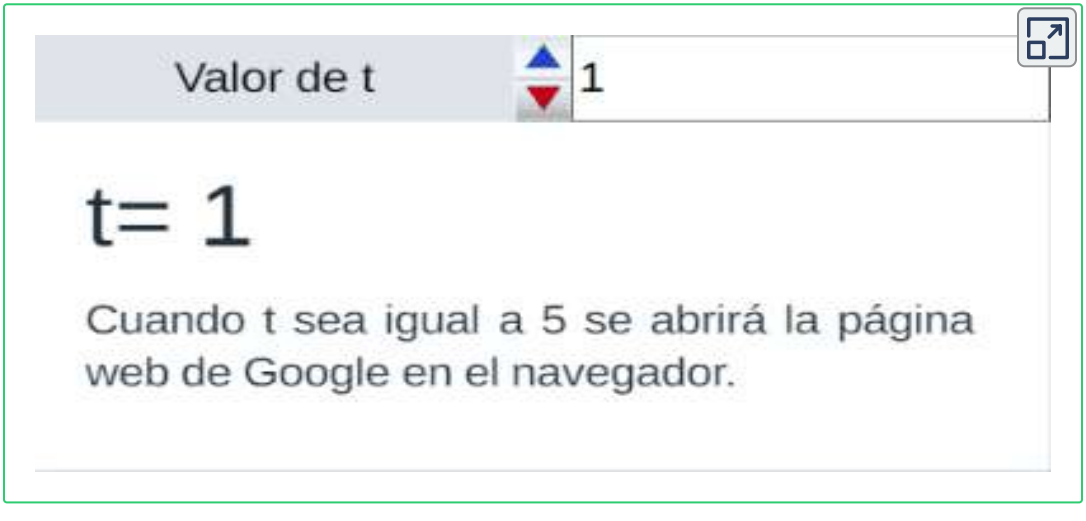
Incluimos un control tipo pulsador que asigne el valor a la variable **t**.



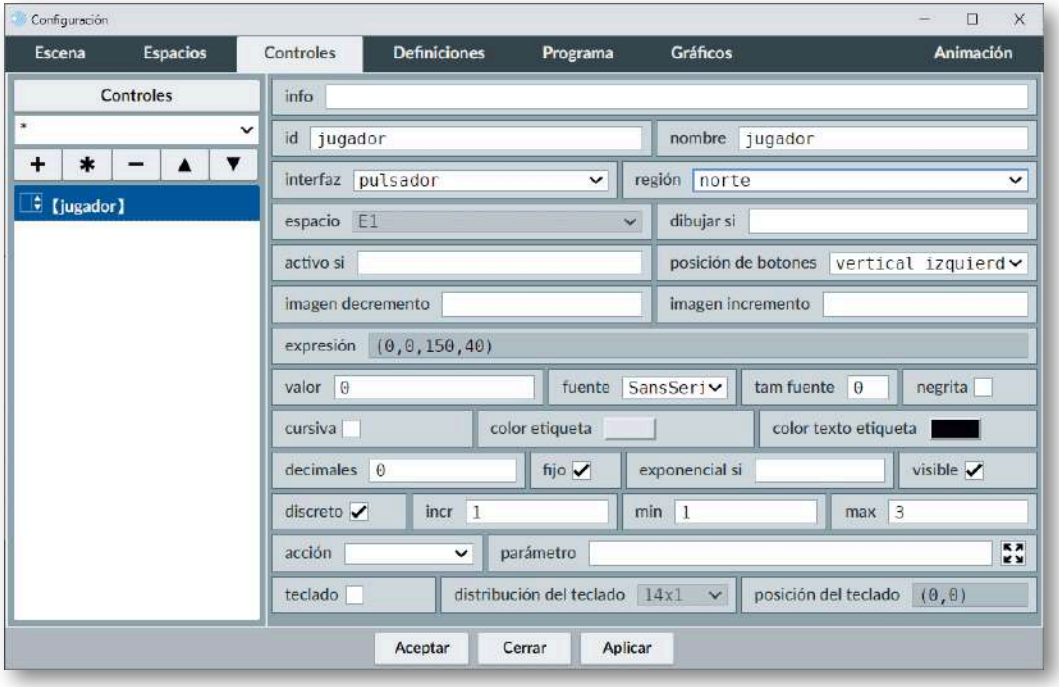
Colocamos textos en pantalla para presentar el valor de la variable **t** y para indicar la acción que se desarrollará cada vez que t tome el valor de 5.



La escena obtenida es la siguiente:



Desarrollamos otro ejemplo para ilustrar el funcionamiento de los eventos. En primer lugar, introducimos un control de tipo pulsador, al que llamaremos **jugador**, situado en la zona norte. Este control será un número entero que podrá tomar valores del uno al tres.

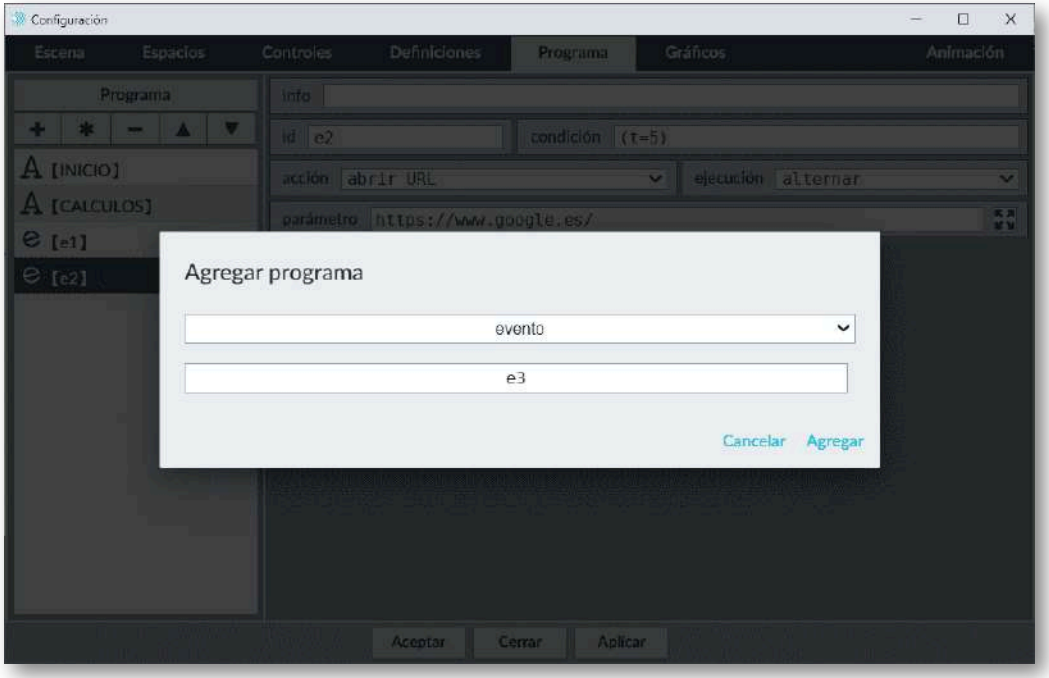


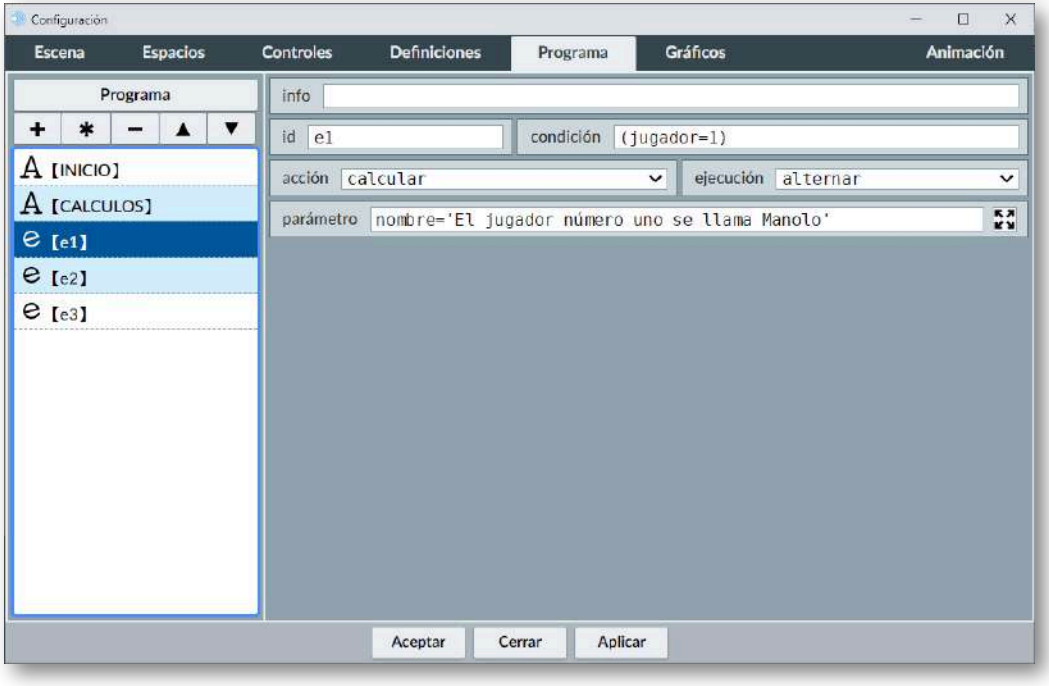
Diseñamos tres eventos que asignarán un valor a la variable **nombre** definida en el texto anterior. El primero se ejecutará cuando la variable **jugador** sea igual a 1; el segundo, cuando **jugador** tome el valor 2; y el tercero, cuando **jugador** sea igual a 3.

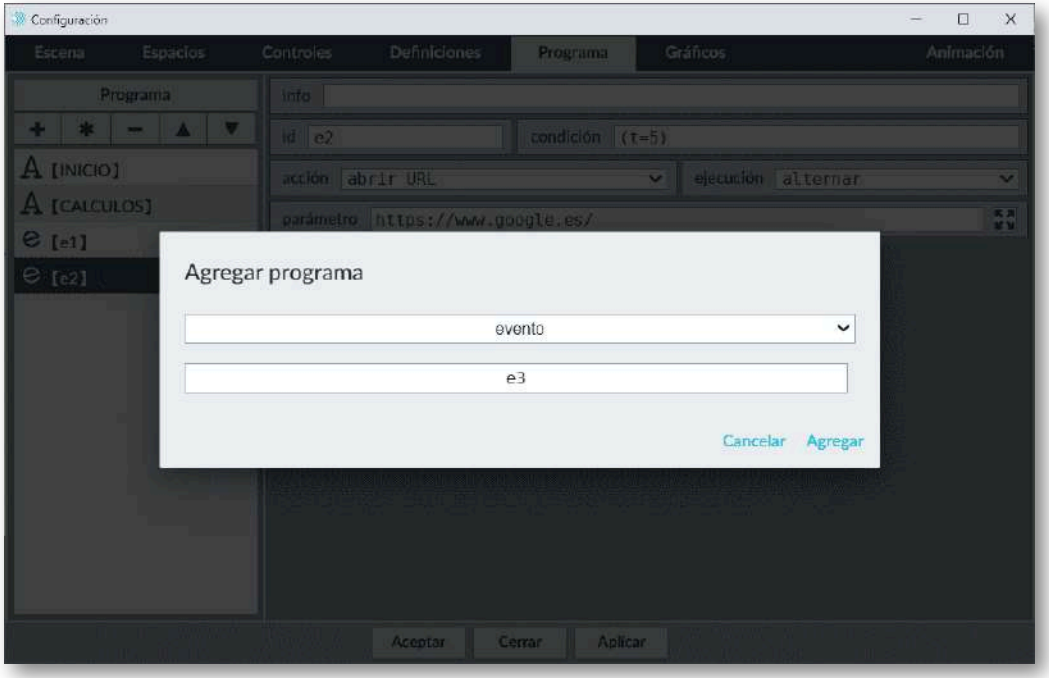
En los tres casos, la acción se configurará en modo **alternar** (es decir, se ejecutará cada vez que la condición vuelva a cumplirse después de haber dejado de hacerlo) y la acción seleccionada será **calcular**.

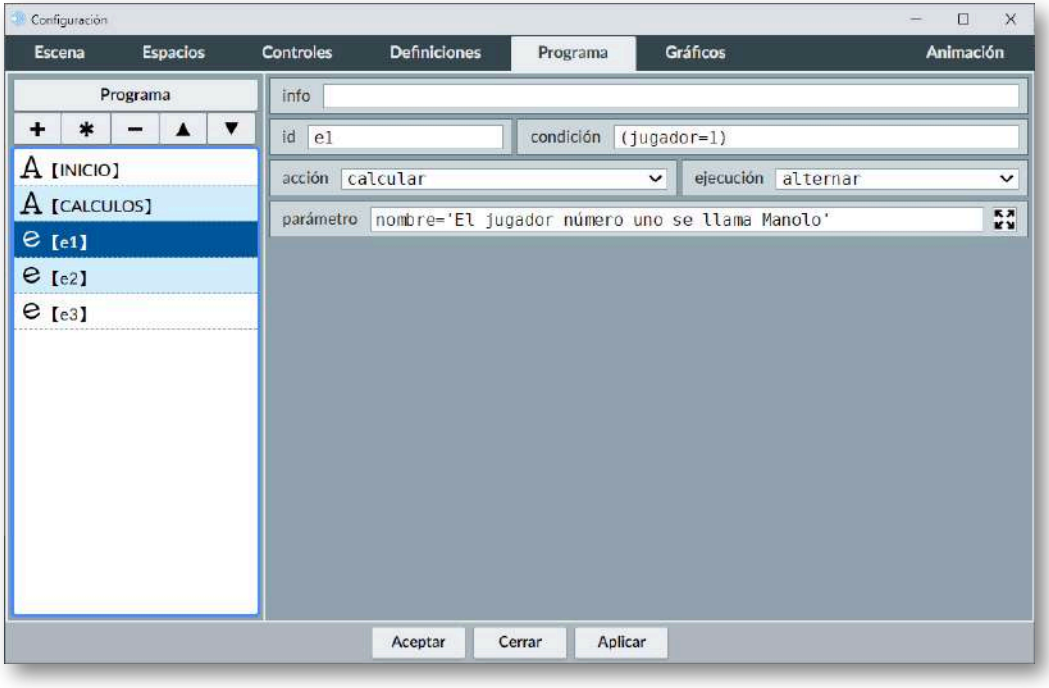
Las asignaciones serán las siguientes: en el primer evento, *nombre='El jugador número uno se llama Manolo'*; en el segundo, *nombre='El jugador número dos se llama Mary'*; y en el tercero, *nombre='El jugador número tres se llama Javi'*. Recordamos que los textos deben escribirse entre comillas simples.

Primero añadimos cada uno de los eventos y, a continuación, definimos sus parámetros.

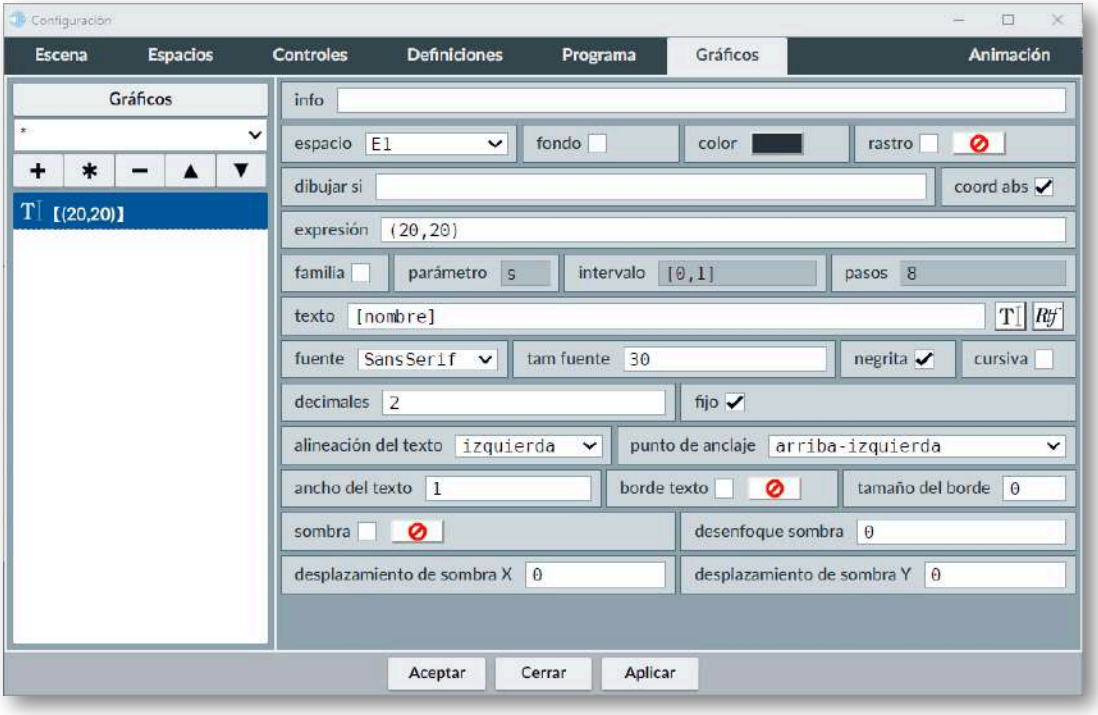




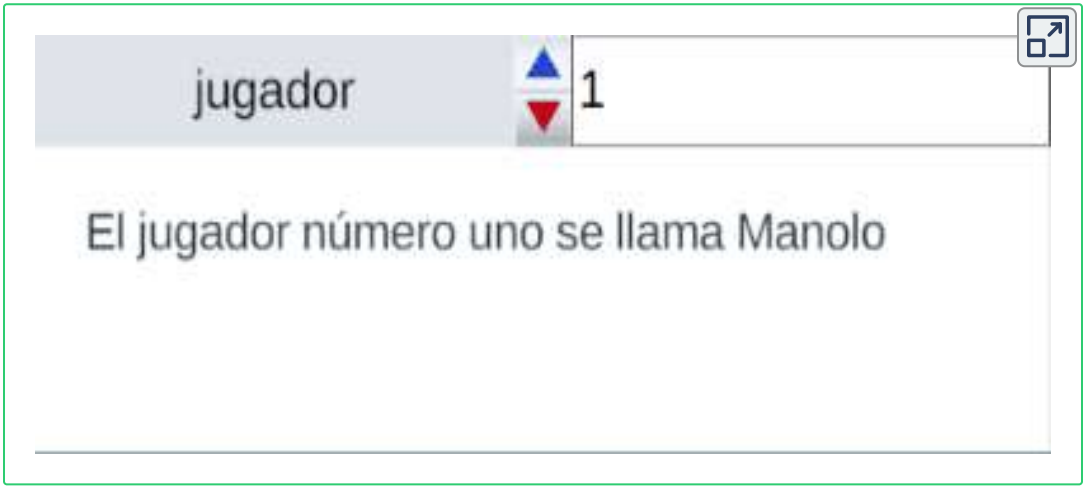




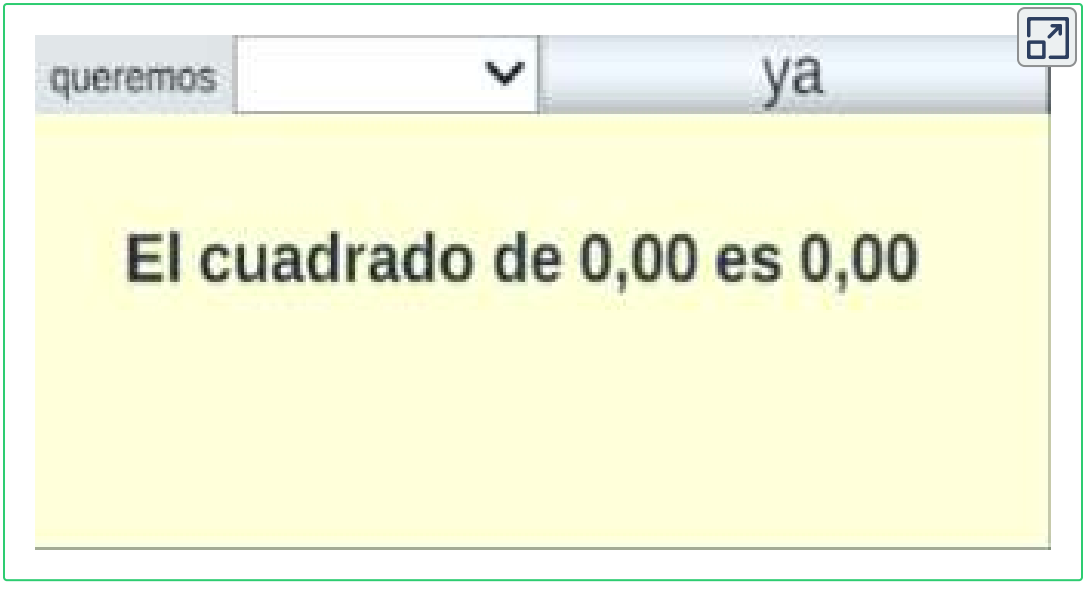
Introduciremos un texto cuyo valor será el contenido de la variable **nombre**.



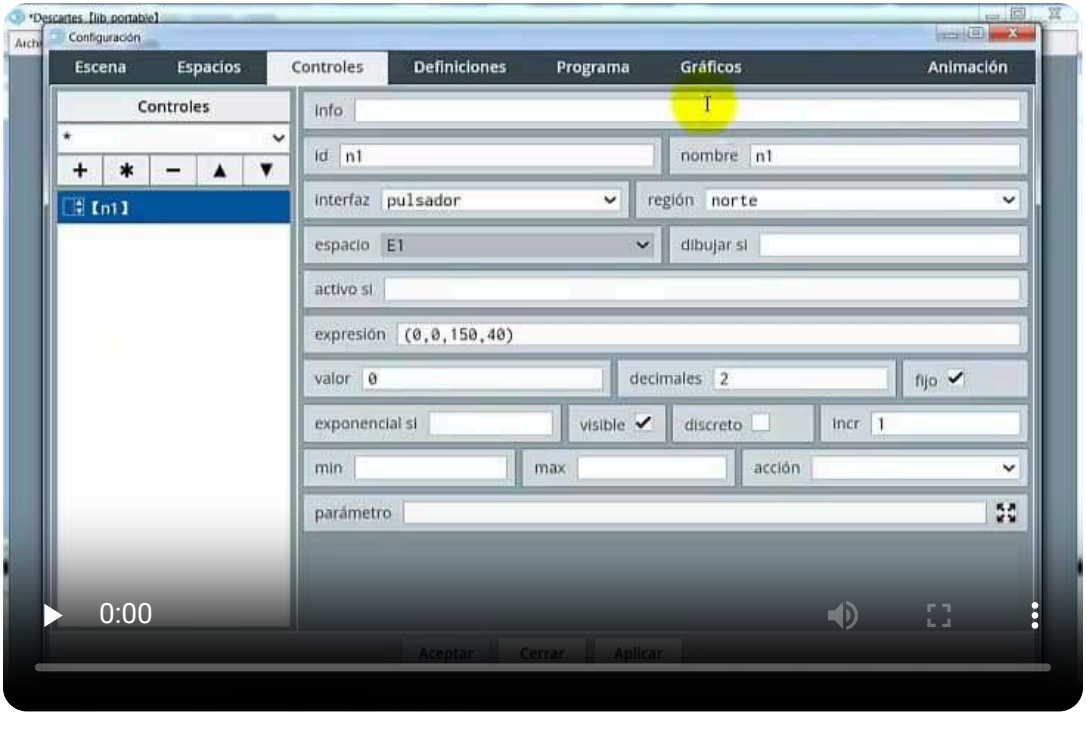
La escena resultante es la siguiente:



Incluimos una escena en la que se desarrollan distintos eventos, que pueden seleccionarse mediante un control de tipo menú y un control tipo botón.



En el siguiente vídeo se muestra cómo introducir eventos en las escenas.



4.14 Algoritmos Inicio y Cálculos

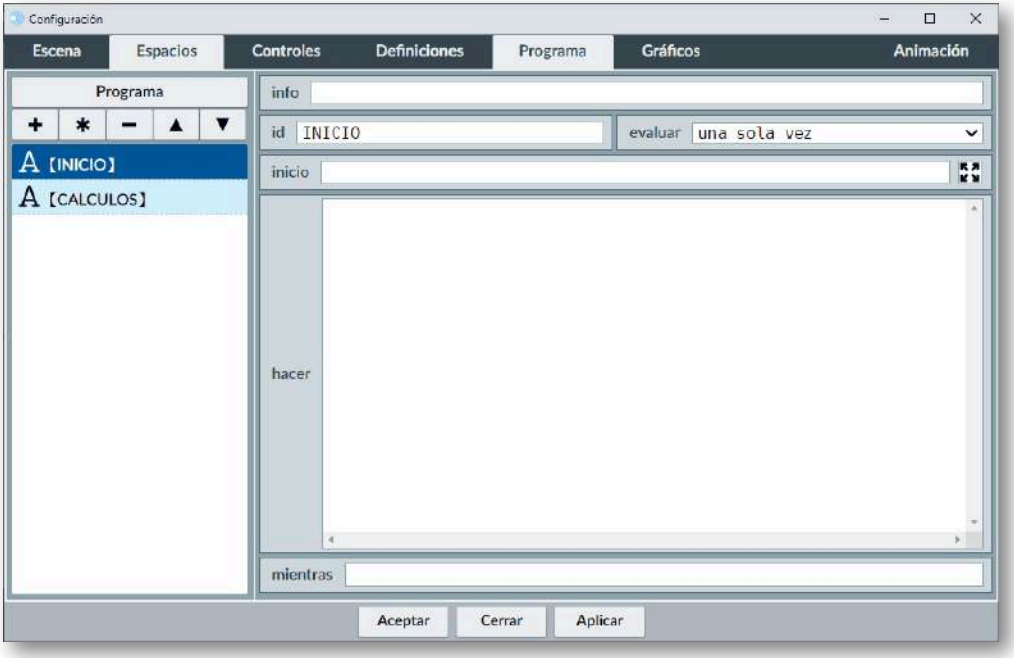
En ocasiones es necesario realizar operaciones más complejas que implican tareas iterativas o condicionales, así como múltiples acciones encadenadas: asignación de valores a variables, vectores y matrices, creación de funciones, entre otras. En definitiva, los algoritmos permiten programar procesos de cálculo de distinta complejidad. Este tipo de operaciones ya se ha abordado anteriormente en los apartados dedicados a animaciones y funciones.

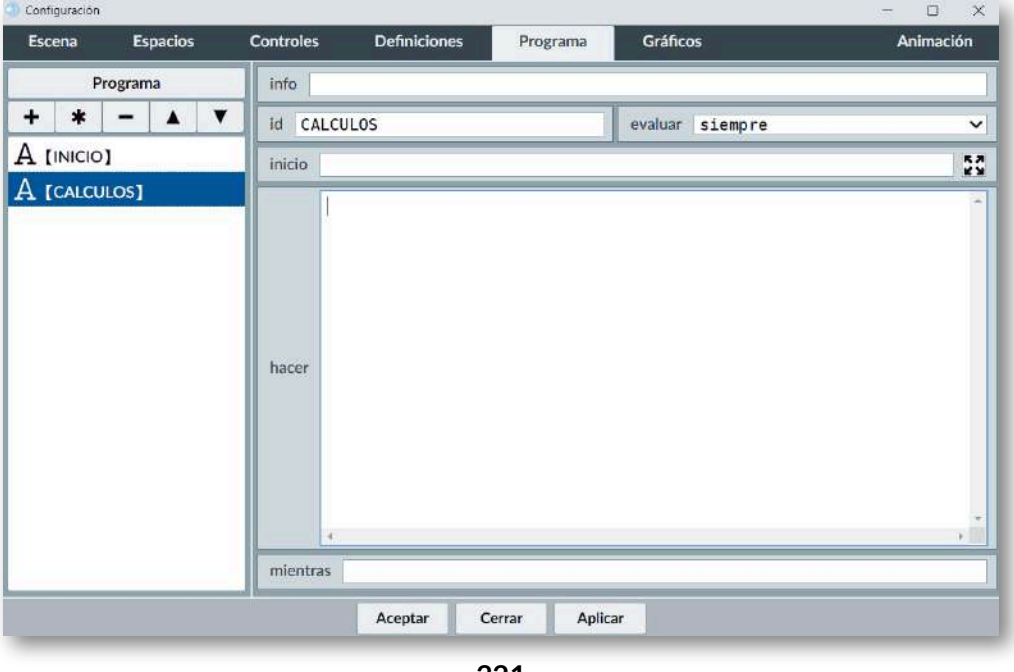
Las escenas de Descartes incluyen por defecto dos algoritmos en el selector **Programa**: **INICIO** y **CALCULOS**. La diferencia entre ambos es que **INICIO** se ejecuta únicamente al comenzar la escena, mientras que **CALCULOS** se ejecuta de manera continua. Son los algoritmos recomendados para trabajar, aunque es posible crear otros nuevos a partir de ellos mediante copia. En realidad, lo que los distingue es el parámetro *evaluar*: en **INICIO** se define por defecto como **una sola vez**, mientras que en **CALCULOS** aparece como **siempre**, aunque este campo puede modificarse. Aunque es posible cambiar esta configuración, se recomienda mantenerla en **una sola vez** para **INICIO** y en **siempre** para **CALCULOS**

Descartes no permite crear nuevos algoritmos de tipo **INICIO** o **CALCULOS**, aunque sí permite clonar los existentes o cambiarles el nombre.

El algoritmo **INICIO** se utiliza generalmente para realizar los cálculos iniciales que preparan el interactivo para su uso. Estos cálculos pueden incluir asignaciones, condicionales y llamadas a funciones. Por su parte, el algoritmo **CALCULOS** también puede contener instrucciones, asignaciones, condicionales y llamadas a funciones, y se ejecuta cada vez que el usuario realiza un cambio o acción en el interactivo.

A continuación, se muestran dos capturas de los paneles de configuración de los algoritmos **INICIO** y **CALCULOS**, y se describen sus principales parámetros (además de los comentarios, info):





- **id:** campo de texto para el nombre del algoritmo. Por defecto aparece como **INICIO** y se recomienda no modificarlo.
- **evaluar:** menú que indica cuándo deben evaluarse las instrucciones del algoritmo:
    - **una sola vez:** el algoritmo se ejecuta únicamente al inicio de la escena. Para el algoritmo **INICIO**, esta opción está seleccionada por defecto. Se recomienda mantenerla así para reducir cálculos y evitar que el interactivo se ralentice.
    - **siempre:** el algoritmo se ejecuta cada vez que el usuario realiza un cambio en el interactivo, como modificar un control. Esta opción se utiliza para algoritmos que deben ejecutarse constantemente. Se recomienda mantener estos algoritmos lo más pequeños posible, ya que al repetirse continuamente pueden ralentizar el interactivo. Esta es la opción seleccionada por defecto para el algoritmo **CÁLCULOS**
- **inicio:** campo de texto donde se incluyen asignaciones, llamadas a funciones u otras instrucciones que se realizan al inicio del algoritmo. Si un algoritmo requiere repetirse varias veces, este campo contiene las asignaciones previas a dichas repeticiones. Cada instrucción debe separarse con un punto y coma (;).
- **hacer:** panel donde se introducen las instrucciones que se ejecutarán de forma cíclica. Si el algoritmo no incluye instrucciones repetitivas, se pueden colocar aquí o en el campo **inicio** indistintamente.
- **mientras:** campo de texto donde se define una condicional que determina la repetición de las instrucciones del panel **hacer** mientras la condición sea verdadera. Cuando la condición o condiciones dejen de cumplirse el ciclo no se volverá a ejecutar y el algoritmo finalizará su ejecución. Si se deja vacío, las instrucciones se ejecutan solo una vez.

Es importante tener en cuenta que el número máximo de ciclos permitidos es de 100,000. Si la condición del campo **mientras** implicara más iteraciones, Descartes detendrá la ejecución al alcanzar este límite. Esta restricción busca evitar que usuarios con experiencia limitada en programación generen errores por ciclos infinitos.

La ejecución del algoritmo se lleva a cabo de la siguiente manera:

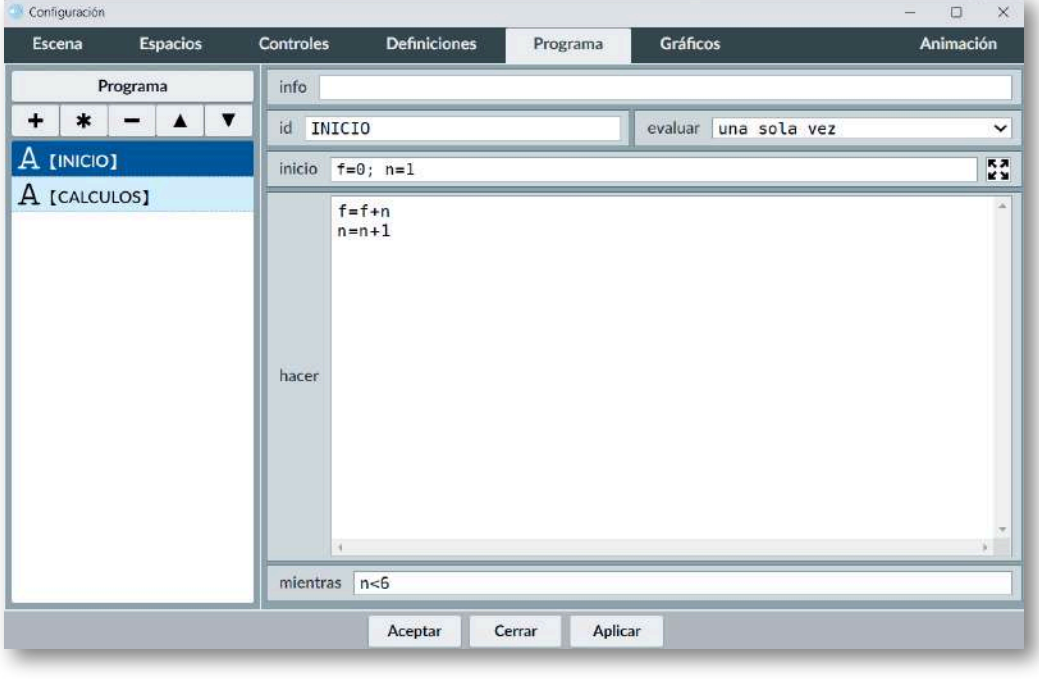
- **Paso 1:** En el campo **inicio** pueden escribirse una serie de asignaciones y llamadas a otros algoritmos, separadas por punto y coma (;).
- **Paso 2:** Se realizan las asignaciones, llamadas a otros algoritmos y operaciones indicadas en **hacer**. Este es el paso que se repite del algoritmo.
- **Paso 3:** Se comprueba si se cumple la condición (expresión booleana) contenida en **mientras**. Si la condición se cumple, se vuelve al paso 2, ejecutando de nuevo **hacer** y verificando la condición **mientras**. Este proceso continúa hasta que la condición deje de cumplirse o se alcance el límite de 100,000 repeticiones, que actúa como medida de seguridad. Si la condición **mientras** se deja en blanco, las instrucciones de **hacer** se ejecutan una sola vez.

Para realizar asignaciones condicionales se puede utilizar la construcción *(A)?a:b*, donde **A** es una expresión booleana, y **a** y **b** son los valores asignados según se cumpla o no la condición. Por ejemplo, *y=(x>0)?cos(4*pi*x):1* asigna a **y** el valor *cos(4*pi*x)* si **x>0** y el valor 1 si **x≤0**.

La combinación de los bucles **inicio** - **hacer** - **mientras**, las asignaciones condicionales *(A)?a:b* y la posibilidad de llamar algoritmos desde **CALCULOS** permite programar procesos de cálculo relativamente complejos.

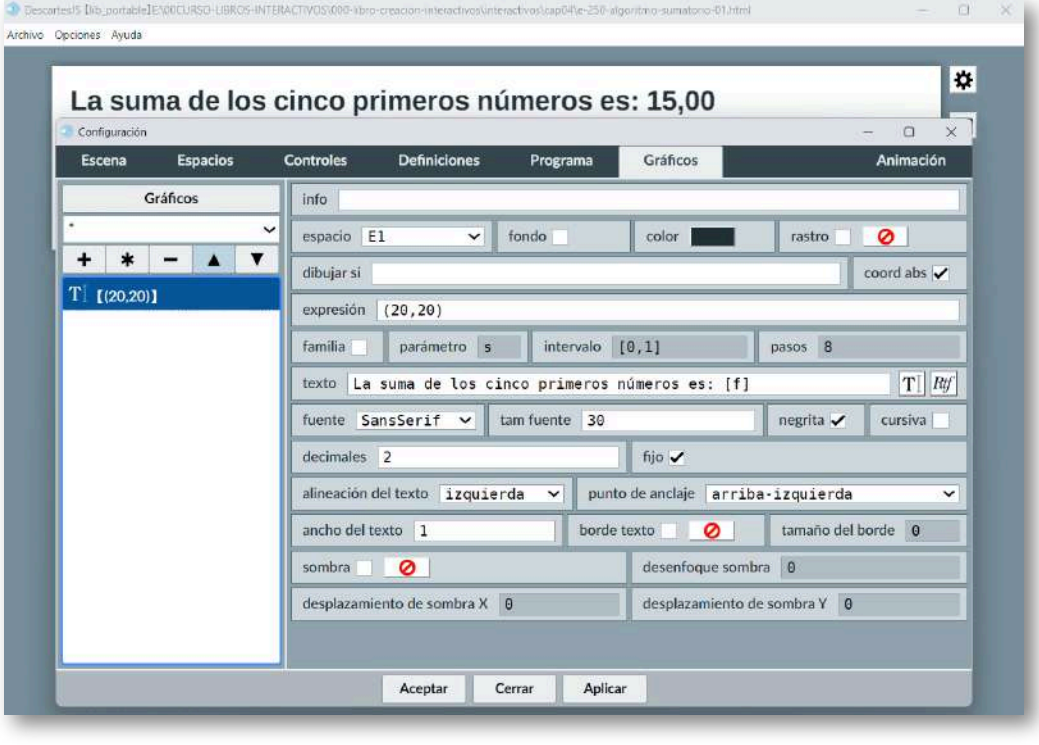
En este ejemplo, utilizamos un algoritmo para sumar los cinco primeros números enteros (x=5; en general, cualquier número sería la función sumatorio). En **inicio** se asigna *n=1* y *f=0*. En **hacer** se realizan las operaciones *f=f+n* y *n=n+1*, mientras *n<6*. Comentamos el funcionamiento ciclo a ciclo:

- Primer ciclo: f=0 y n=1 → f=0+1=1, n=1+1=2. La condición n<6 se cumple, se ejecuta otro ciclo.
- Segundo ciclo: f=1 y n=2 → f=1+2=3, n=2+1=3. La condición se cumple, se ejecuta otro ciclo.
- Tercer ciclo: f=3 y n=3 → f=3+3=6, n=3+1=4. La condición se cumple, se ejecuta otro ciclo.
- Cuarto ciclo: f=6 y n=4 → f=6+4=10, n=4+1=5. La condición se cumple, se ejecuta otro ciclo.
- Quinto ciclo: f=10 y n=5 → f=10+5=15, n=5+1=6. La condición n<6 ya no se cumple, el algoritmo se detiene. El valor final de f=15 corresponde a la suma de los cinco primeros números.

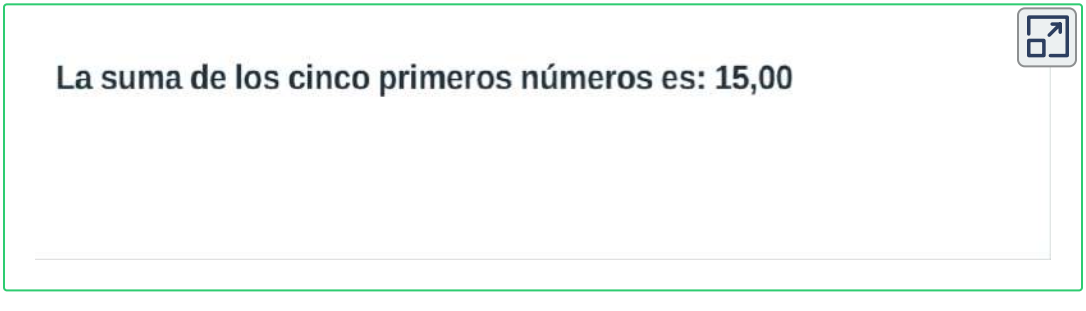


Si se tratara de un número mayor de elementos, el ciclo se repetiría más veces hasta alcanzar el total correspondiente.

Mediante un texto mostraremos el valor de **f**, que es la variable del algoritmo que suma los cinco primeros números naturales. Se puede introducir como texto sin formato, incluyendo la variable **f** entre corchetes en el campo texto: **[f]**. En nuestro ejemplo, el valor de **f** será 15.

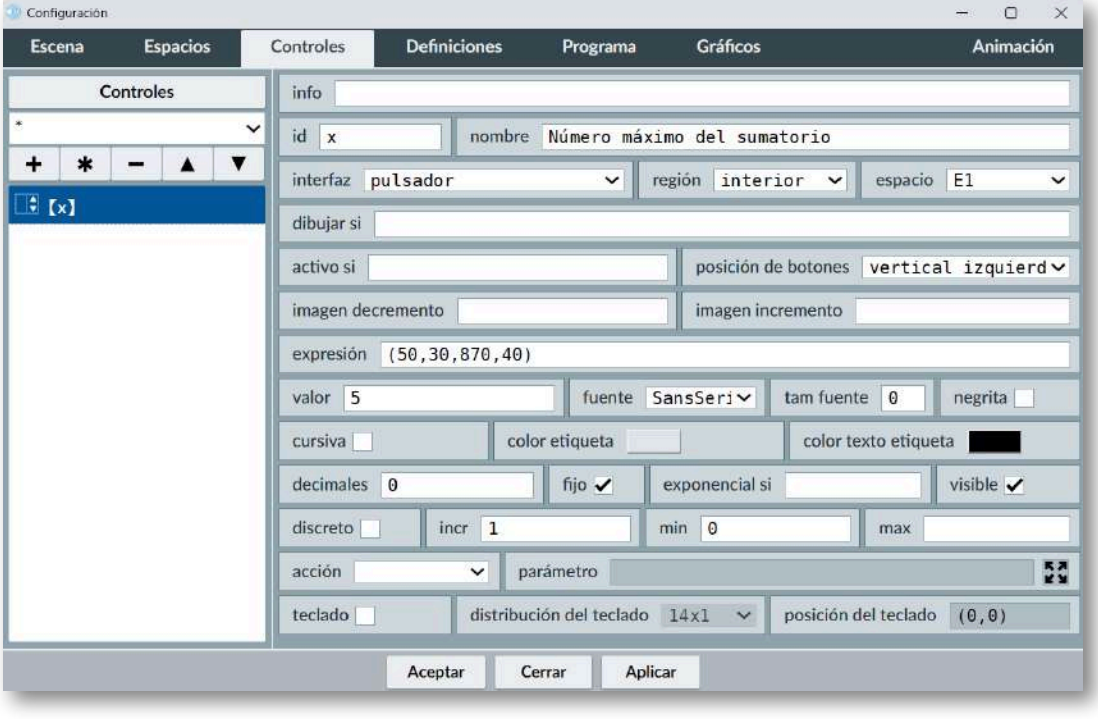


La escena sería la siguiente:

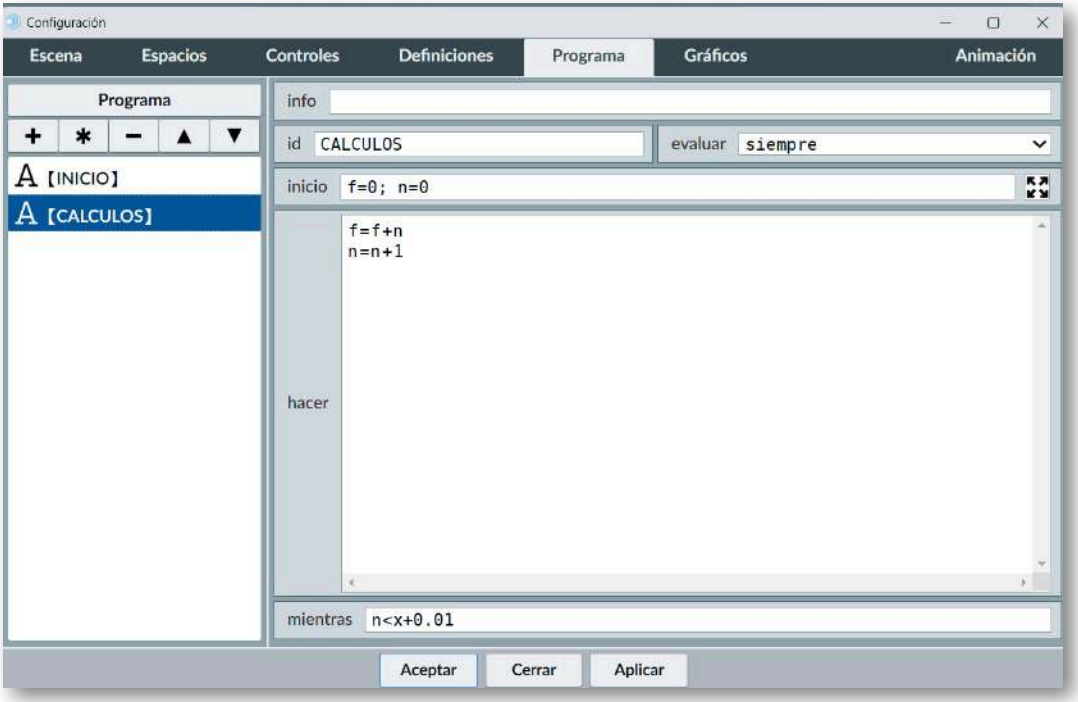


Si en lugar de calcular el sumatorio de un número concreto queremos realizarlo para cualquier número positivo introducido mediante un control, tendríamos que utilizar el algoritmo **CALCULOS**, ya que es necesario que se evalúe cada vez que se modifique el valor del control. Es decir, el parámetro *evaluar* debe estar configurado en **siempre** (esto también podría lograrse utilizando funciones).

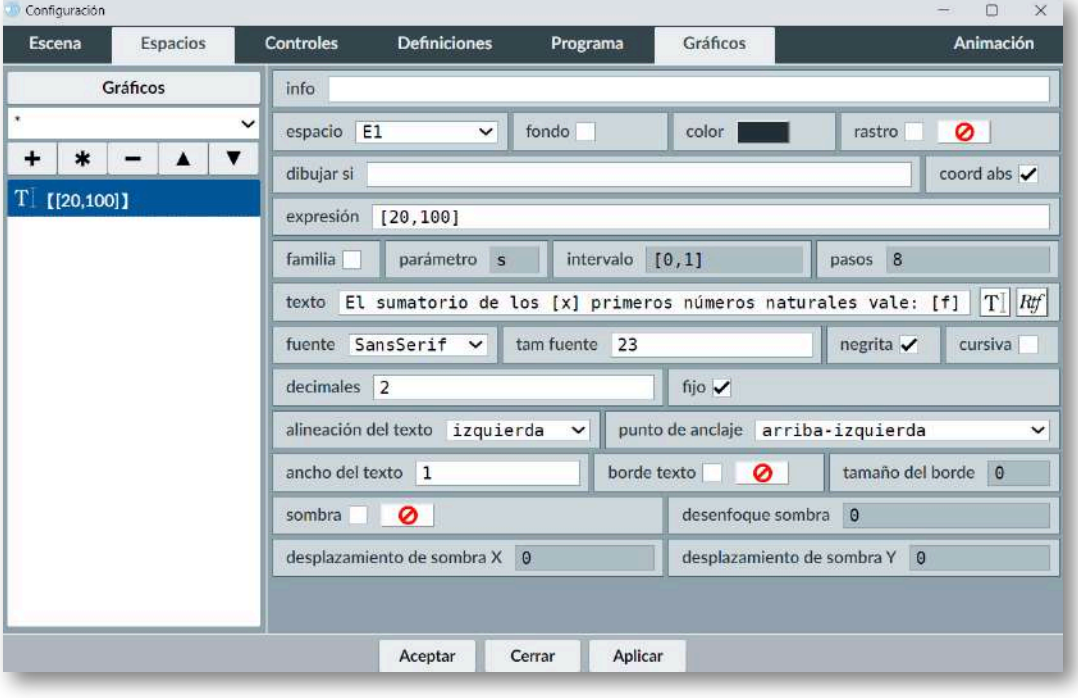
Desarrollamos el ejemplo paso a paso. En primer lugar, añadimos un control de tipo pulsador, al que identificaremos como **x**.



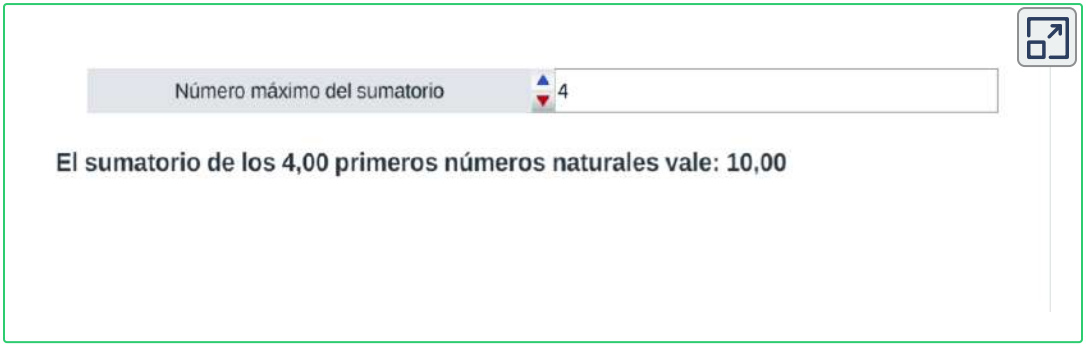
A continuación, definimos el algoritmo **CALCULOS**, que es el elemento de programación encargado de determinar el sumatorio. Como se puede observar, es muy similar al ejemplo anterior, pero en el campo **mientras** se introduce la variable **x**, cuyo valor se obtiene del control tipo pulsador. El valor de **x** indica el número de veces que se repetirá el ciclo definido en **hacer**, es decir, el número de iteraciones necesarias para calcular el sumatorio.



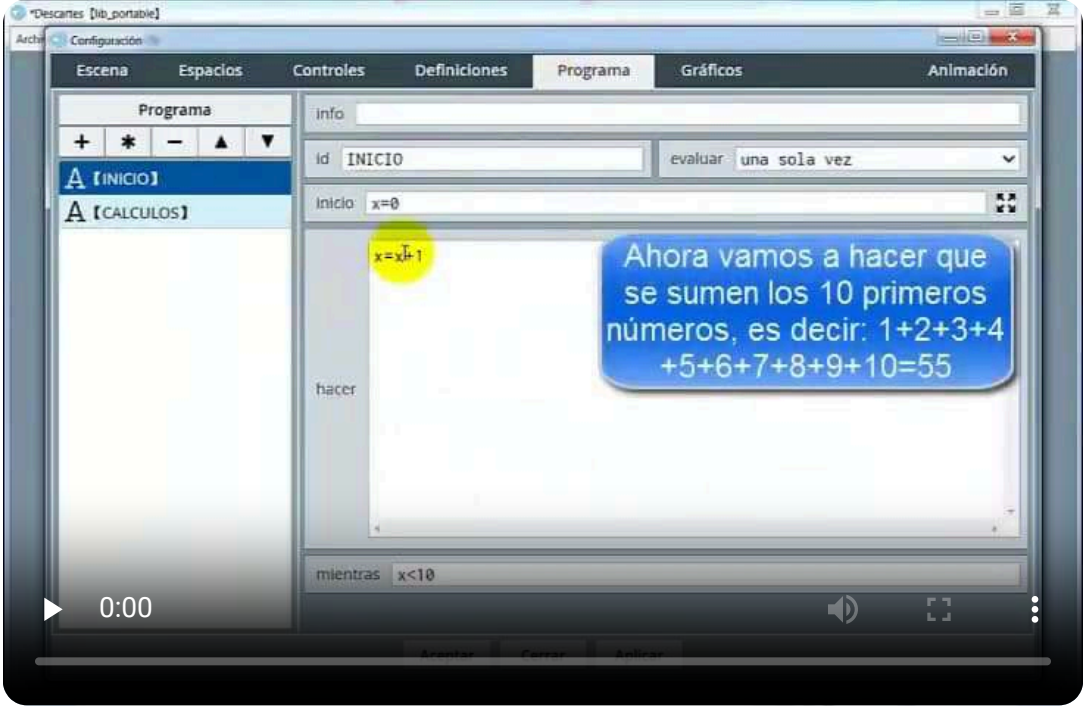
Colocamos un texto para visualizar el resultado del sumatorio.



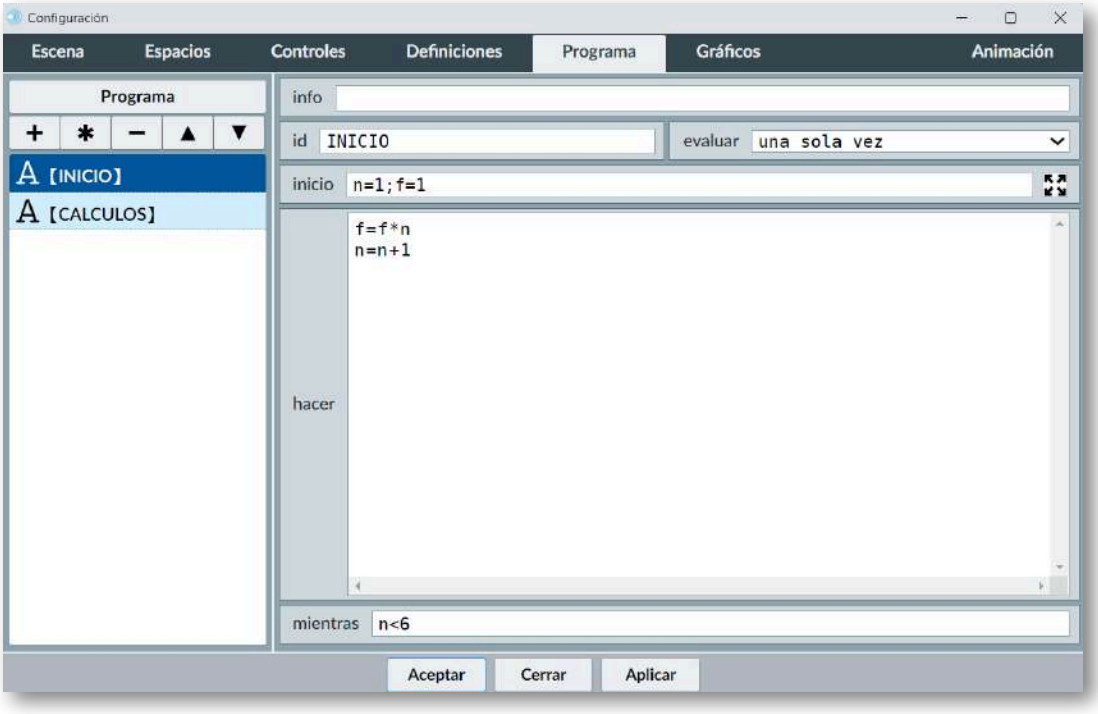
La escena sería la siguiente:



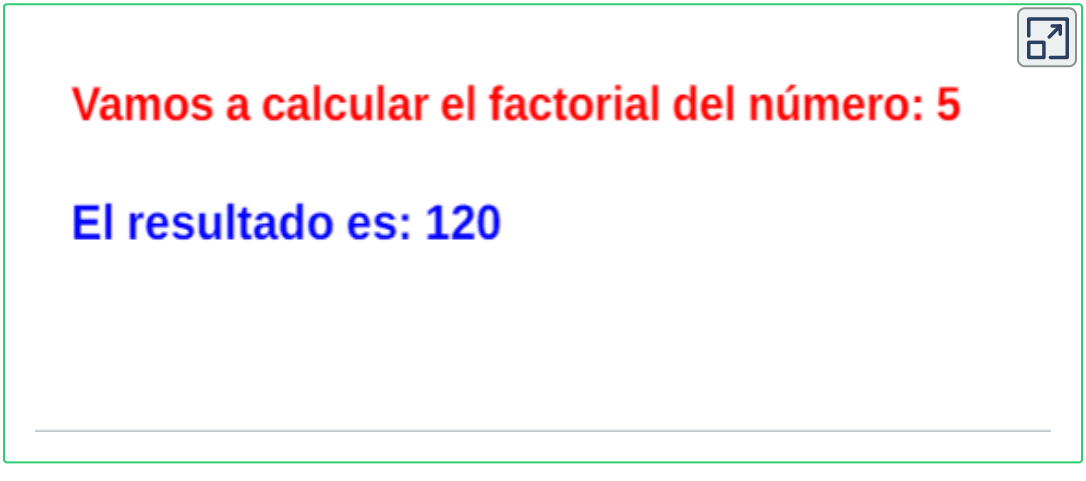
En el siguiente vídeo se presenta la realización de un algoritmo tipo sumatorio.



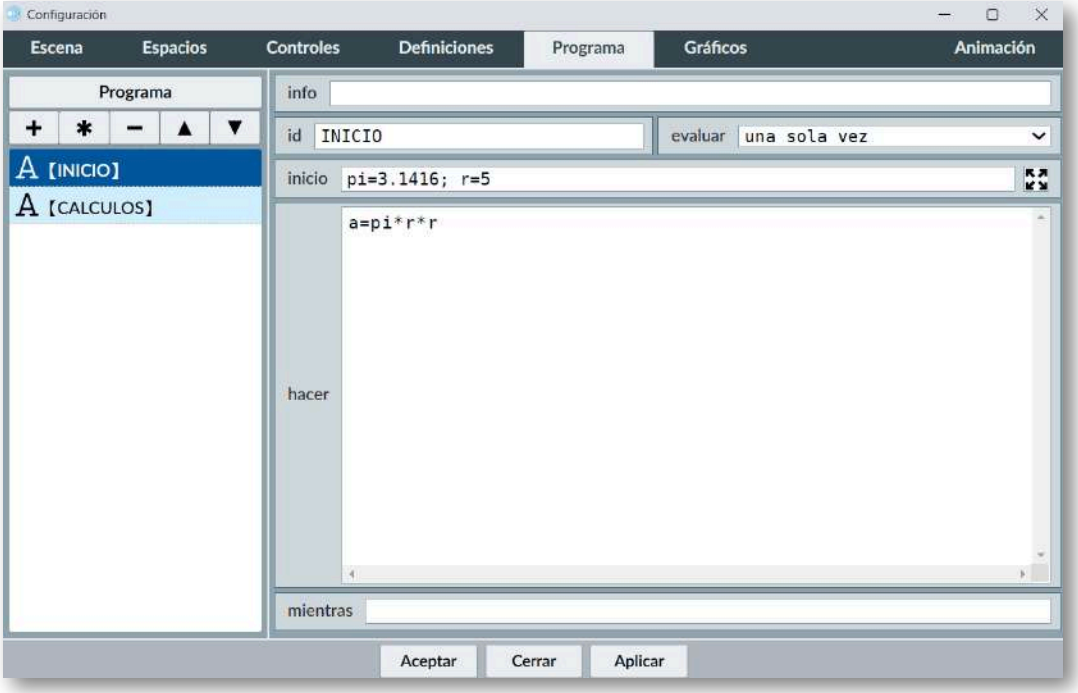
Mostramos otro pequeño ejemplo de cómo calcular el factorial del número 5. Al tratarse de un número fijo, se puede utilizar el algoritmo **INICIO** y evaluarlo únicamente una vez.

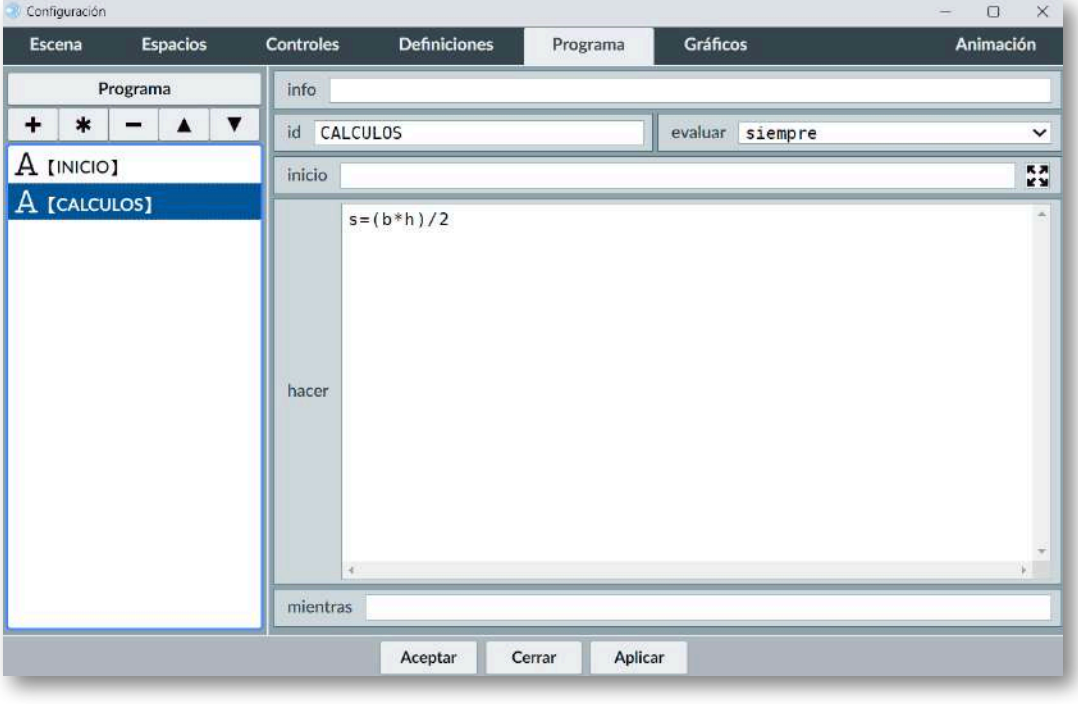


La escena resultante es la siguiente:

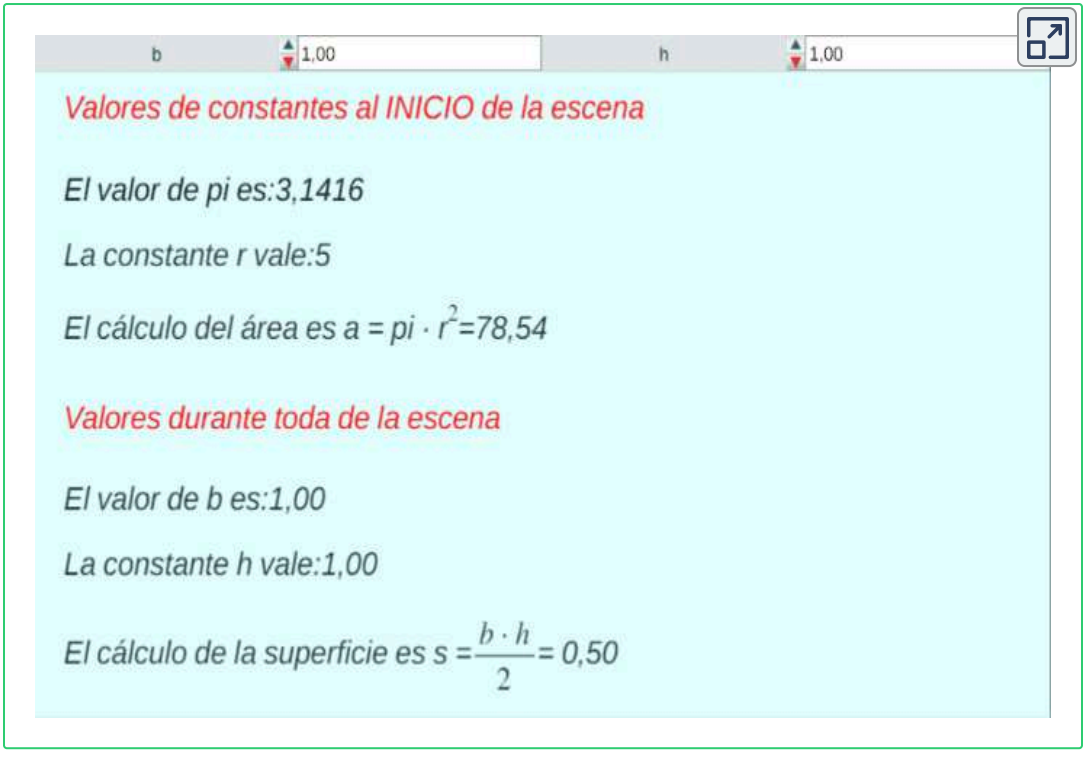


En el siguiente ejemplo se emplean ambos tipos de algoritmos para realizar distintos cálculos geométricos, tanto para el caso de que se necesite evaluar una vez como si es necesario que se evalúe siempre.





La escena es la siguiente:

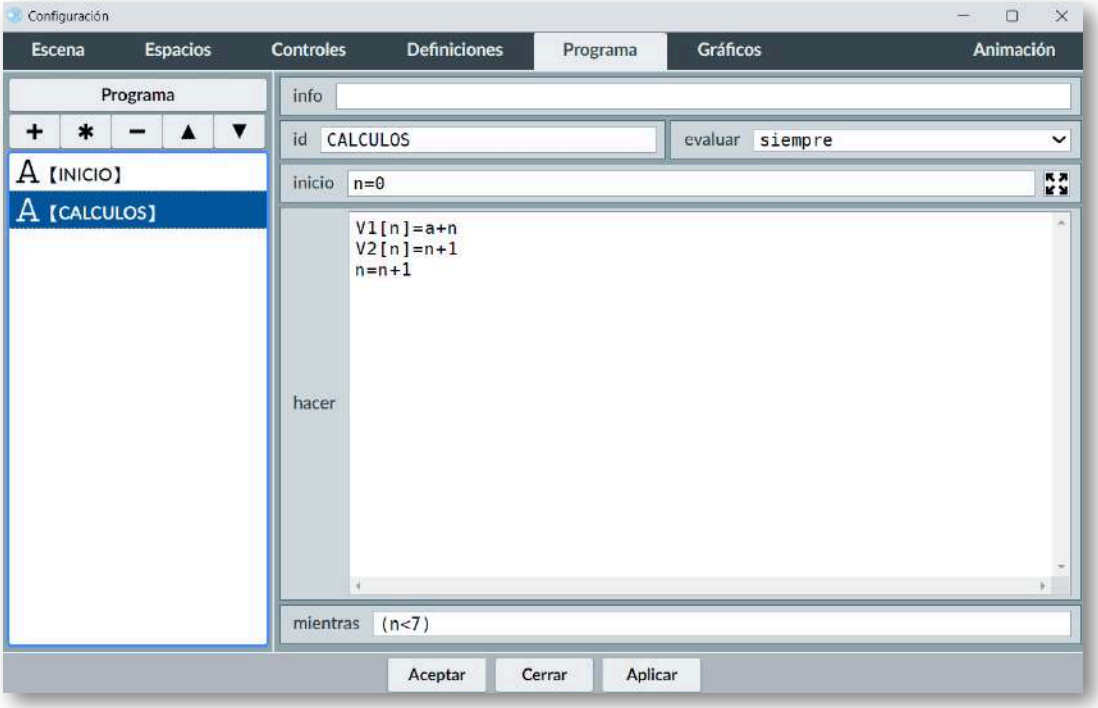


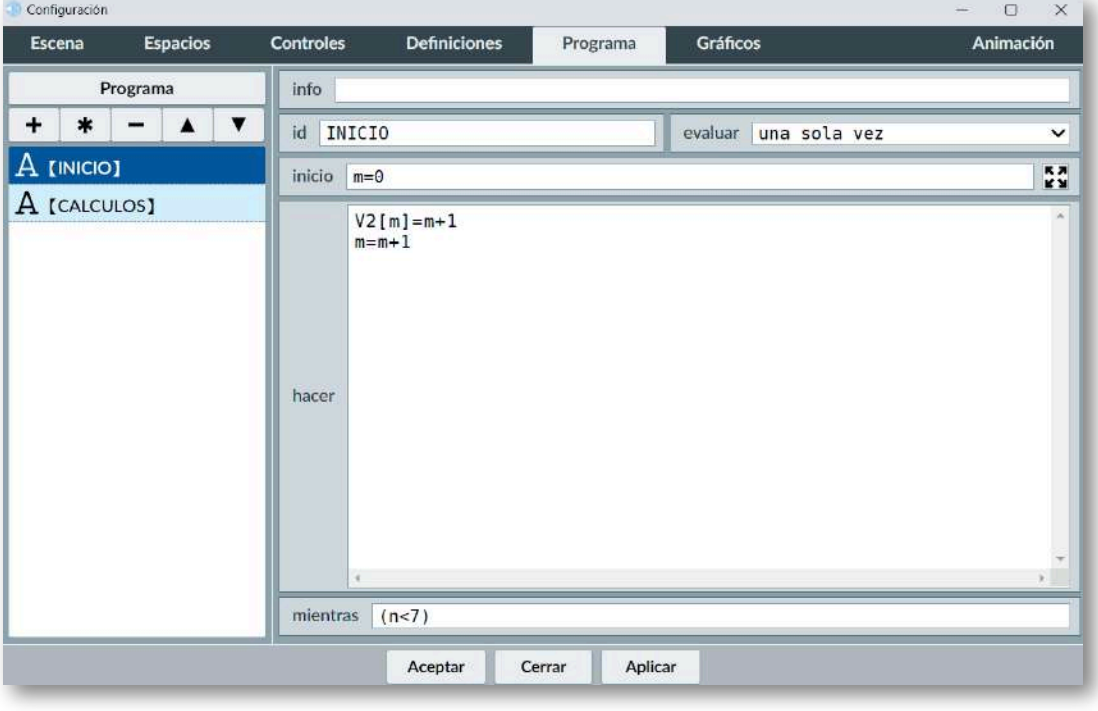
Uno de los usos habituales de los algoritmos es inicializar o asignar datos a los vectores y matrices.

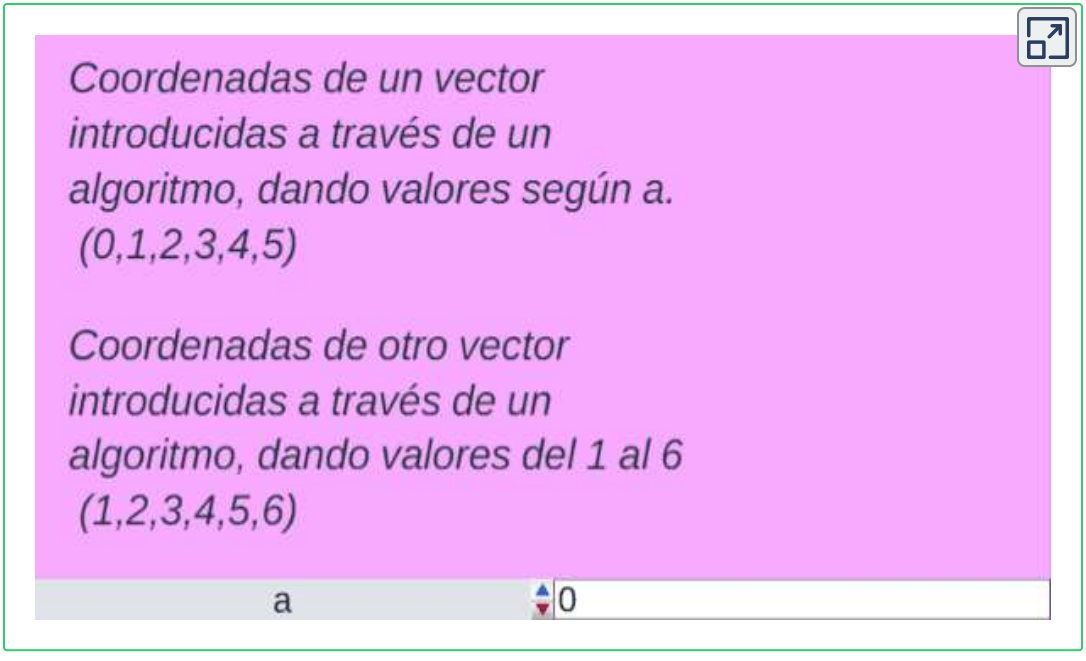
Si se necesita llenar vectores o matrices únicamente al inicio de la escena, se suele utilizar el algoritmo **INICIO**. Si sus valores cambian durante la ejecución de la escena, se puede emplear el algoritmo **CALCULOS** o utilizar los algoritmos incluidos en las funciones.

En el siguiente ejemplo asignaremos valores a dos vectores: el primero, **V1**, cuyos valores se ven afectados por una variable definida mediante un control **(a)** y se utiliza el algoritmo **CALCULOS**; y el segundo, **V2**, que se inicializa una única vez a través del algoritmo **INICIO**.

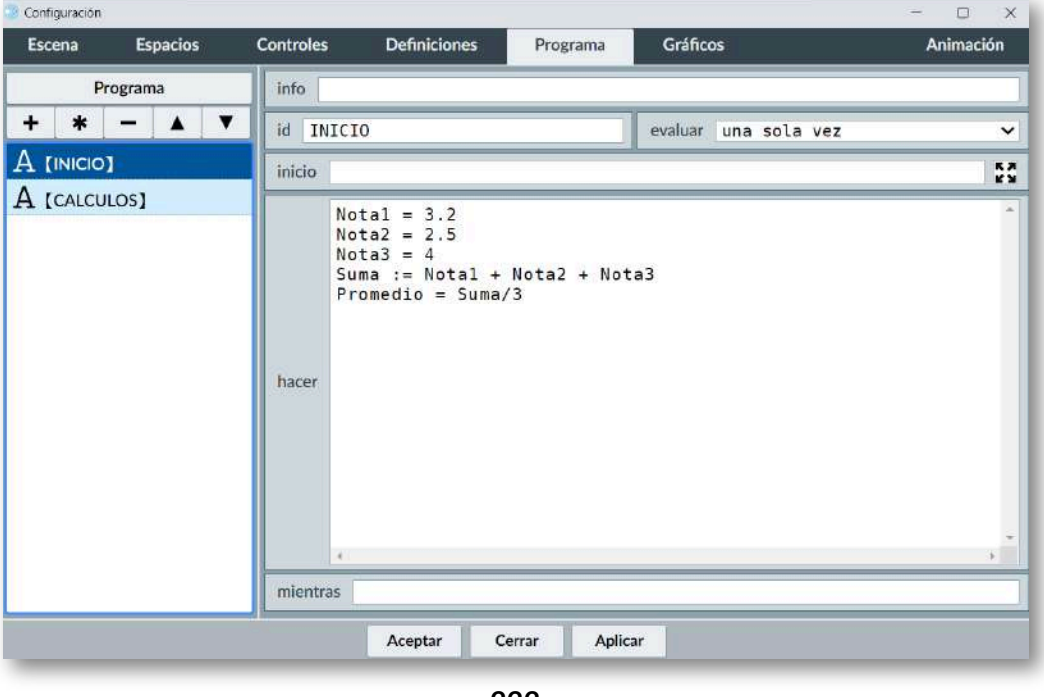
Mostramos capturas de pantalla de los algoritmos utilizados y de la escena resultante.

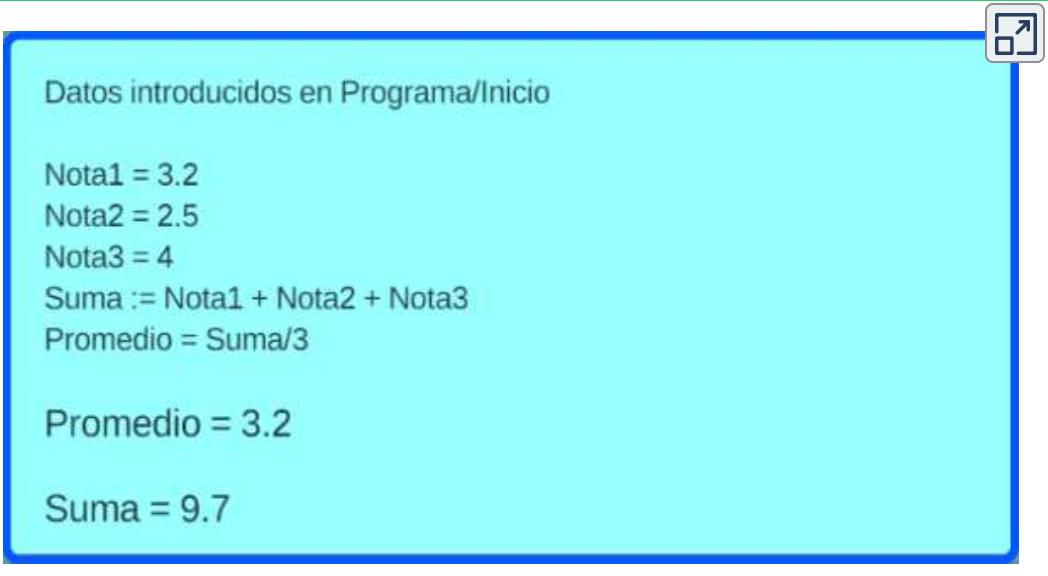

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Programa
【INICIO】
【CALCULOS】
info
id CALCULOS
evaluar siempre
inicio n=0
hacer
V1[n]=a+n
V2[n]=n+1
n=n+1
mientras (n<7)
Aceptar
Cerrar
Aplicar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Programa
【INICIO】
【CALCULOS】
info
id INICIO
evaluar una sola vez
inicio m=0
hacer
V2[m]=m+1
m=m+1
mientras (n<7)
Aceptar
Cerrar
Aplicar

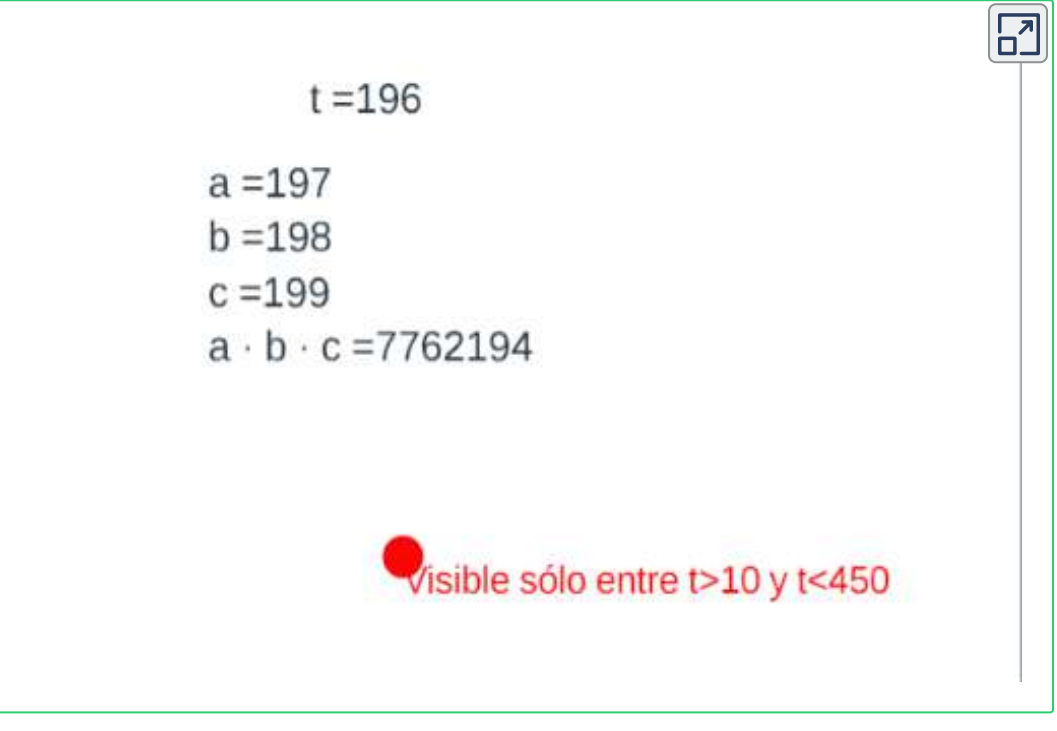


En este otro ejemplo se muestra cómo pueden definirse distintos cálculos al inicio de una escena.





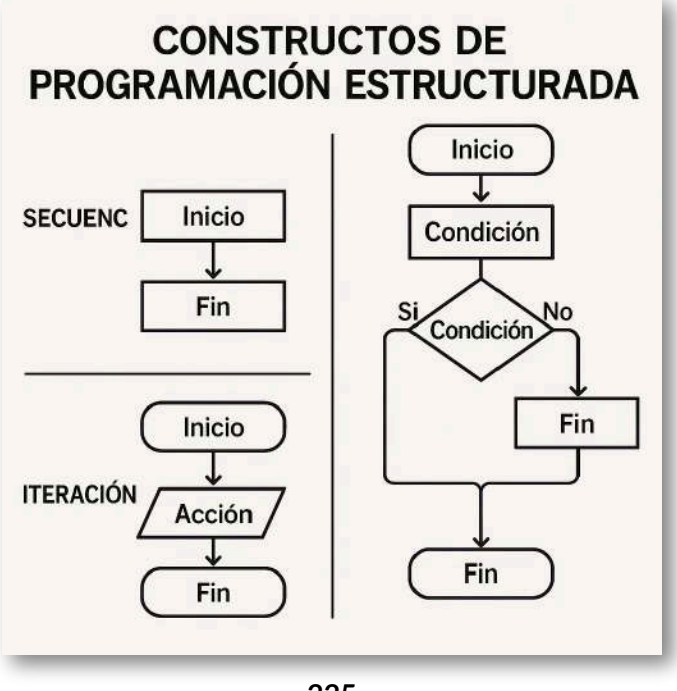
En la siguiente escena se incluyen dos algoritmos, uno de ellos corresponde al algoritmo de animación.



4.15 Algunos aspectos sobre programación

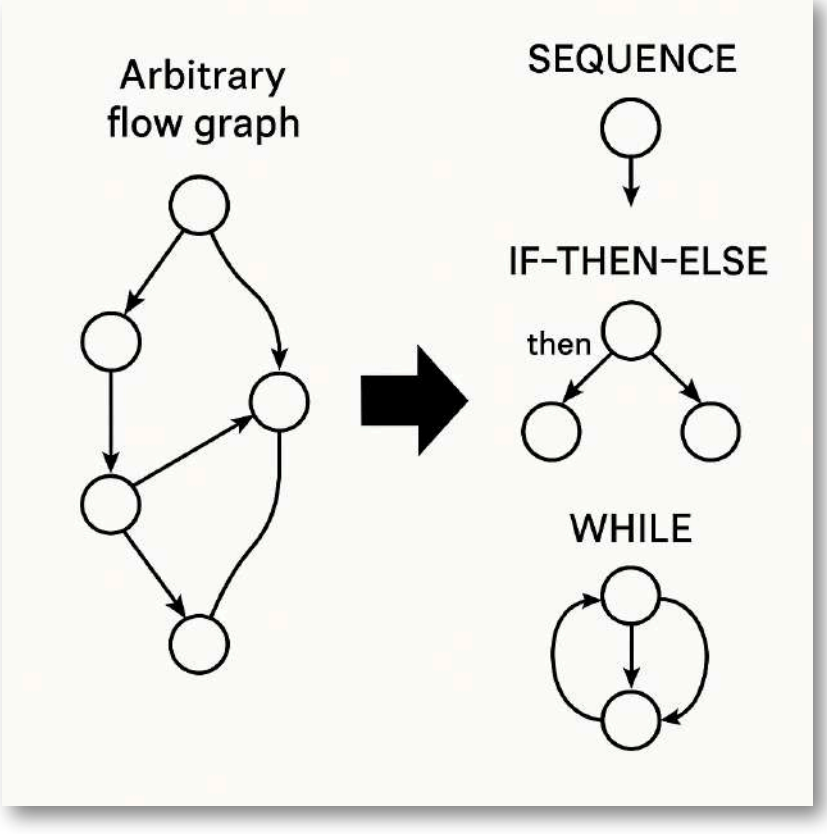
La programación de Descartes se basa en el Teorema del Programa Estructurado, también conocido como el Teorema de Böhm-Jacopini. Este establece que cualquier programa puede escribirse utilizando únicamente tres estructuras de control: **secuencia** (ejecución ordenada de instrucciones), **selección** (estructuras condicionales del tipo *if-then-else*) e **iteración** (bucles como *while*). Estas tres estructuras son suficientes para expresar cualquier función computable.

Este teorema es fundamental en la ciencia de la computación, ya que demuestra que la programación estructurada es completa; es decir, no es necesario recurrir a instrucciones de salto incondicional como *GOTO* para implementar cualquier algoritmo. Este principio impulsó mejores prácticas de programación, favoreció un código más claro y mantenible, y sentó las bases de los lenguajes de programación modernos.



La demostración del teorema consiste en probar que cualquier diagrama de flujo —es decir, cualquier representación gráfica de un programa— puede transformarse en otro equivalente que utilice únicamente las tres estructuras básicas: secuencia, selección e iteración. Para ello, se introducen variables de estado adicionales y se reorganiza el flujo de control, en un proceso conocido como *eliminación estructurada del GOTO*.

Esta demostración no solo tiene relevancia teórica, sino que también ofrece un método constructivo para convertir programas no estructurados en programas estructurados. No obstante, dicha transformación puede añadir cierta complejidad al código, lo que dio lugar a debates sobre si la programación estructurada estricta conduce siempre a programas más claros y comprensibles.



A) Algoritmos

Descripción:

El término *algoritmo* se refiere a un conjunto ordenado y finito de instrucciones que se ejecutan paso a paso para resolver un problema o realizar una tarea concreta. Los algoritmos constituyen la base de la informática, las matemáticas y numerosas disciplinas técnicas y científicas.

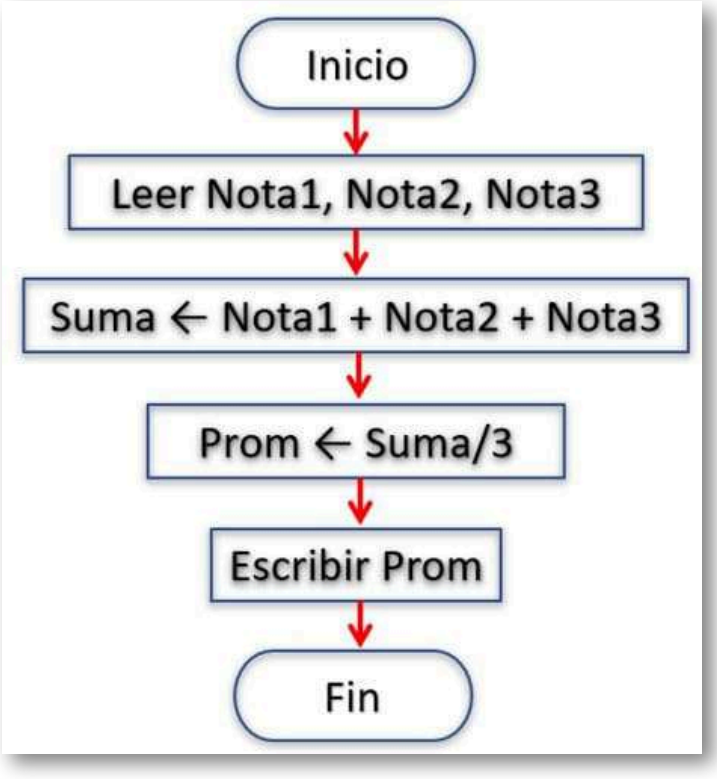
Características principales:

- **Finitud:** debe finalizar tras un número determinado y limitado de pasos.
- **Claridad:** cada instrucción ha de estar perfectamente definida y ser comprensible.
- **Entrada:** puede recibir uno o varios datos iniciales.
- **Salida:** debe producir al menos un resultado.
- **Efectividad:** cada paso debe ser realizable y ejecutable en la práctica.

Asignaciones

Una asignación en un algoritmo es la instrucción mediante la cual se otorga un valor a una variable o conjunto de ellas. Se trata de una operación fundamental en programación, ya que permite almacenar datos que podrán utilizarse posteriormente en distintos cálculos o procesos.

En pseudocódigo o en diagramas de flujo, las asignaciones suelen representarse mediante un bloque rectangular en el que se indica la operación correspondiente. Su función es establecer o actualizar el contenido de una variable dentro del desarrollo del algoritmo.



En el siguiente vídeo mostramos cómo se aplican estos conceptos de programación en el lenguaje de Descartes.



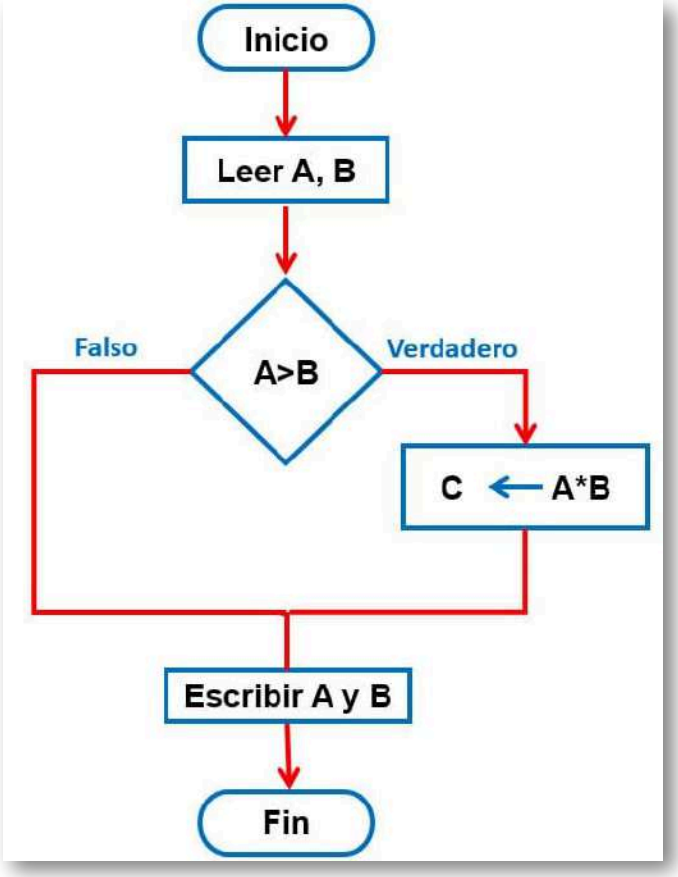
B) Estructura IF simple

Descripción:

La estructura **IF** simple es la forma más básica de control condicional en programación. Evalúa una condición y ejecuta un bloque de instrucciones únicamente cuando dicha condición es verdadera. Si no se cumple, el programa omite ese bloque y continúa con la siguiente instrucción.

Importancia:

Es esencial para implementar la lógica de decisión en cualquier programa. Permite que el comportamiento del software varíe según las circunstancias, lo que resulta imprescindible para crear aplicaciones dinámicas e interactivas.



Ejemplo:

En la imagen se leen los valores de las variables **A** y **B**. Si y solo si **A** es mayor que **B**, se asigna el producto **A*B** a la variable **C**. En el editor DescartesJS, esta operación se expresaría como:

$$C = (A>B)?\ A*B$$

C) Estructura IF-ELSE y operador ternario de JS

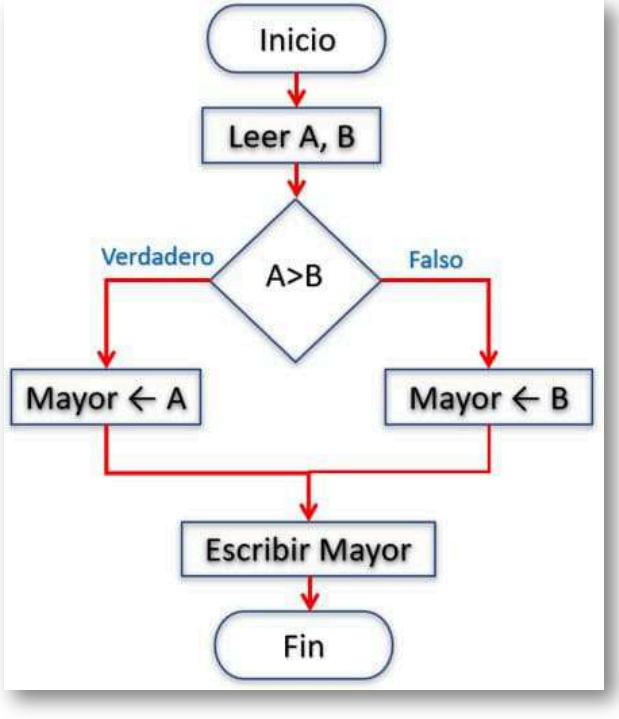
Estructura IF-ELSE

Descripción:

La estructura **IF-ELSE** amplía el IF simple al incorporar un bloque alternativo de instrucciones que se ejecuta cuando la condición resulta falsa. De este modo, se genera una bifurcación en el flujo del programa: si la condición es verdadera, se ejecuta un bloque; si es falsa, se ejecuta el otro.

Importancia:

Permite gestionar dos situaciones mutuamente excluyentes de forma clara y eficiente. Es fundamental cuando debe ejecutarse exactamente una de dos opciones posibles.



Operador ternario en JavaScript:

En el editor DescartesJS, el condicional de la imagen se codificaría de la siguiente forma:

Mayor = (A>B)? A:B

Esta expresión se conoce como **operador ternario** y responde a la sintaxis general:

condición ? valor_si_verdadero : valor_si_falso

D) Operadores de comparación:

Los operadores de comparación permiten contrastar dos valores y devuelven un resultado booleano (verdadero o falso). Entre los más habituales se encuentran: igual a (==), idéntico a (===), mayor que (>), menor que (<), mayor o igual que (>=), menor o igual que (<=) y distinto de (*!=*).

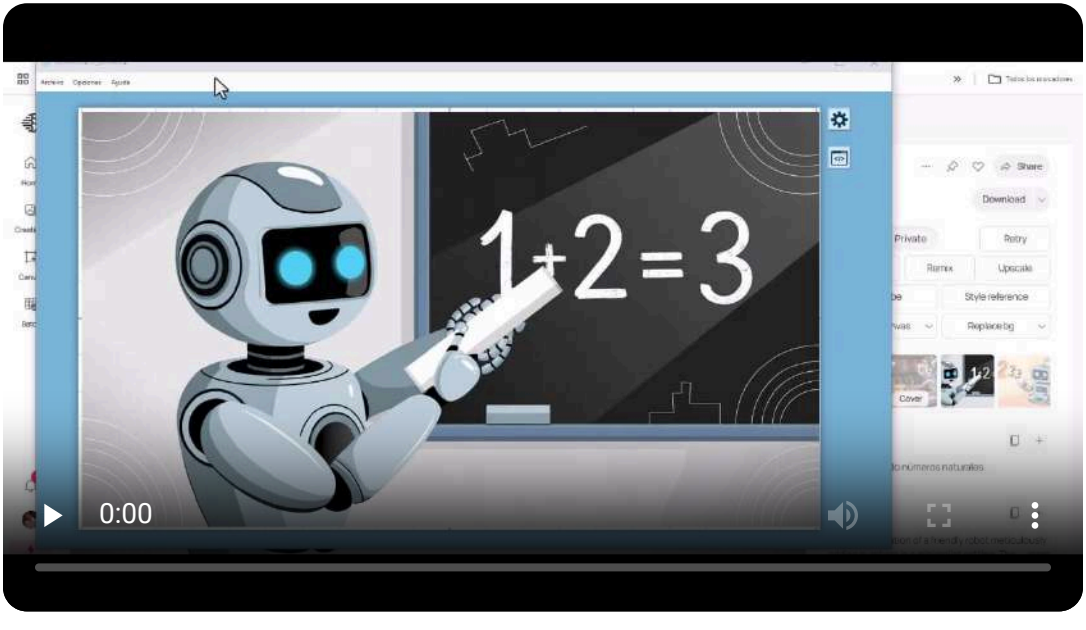
E) Operadores lógicos:

Los operadores lógicos permiten combinar varias condiciones en una sola expresión. Los principales son AND (&), OR (|) y NOT (*!*). El operador AND devuelve verdadero únicamente cuando ambas condiciones son verdaderas; OR devuelve verdadero si al menos una condición lo es; y NOT invierte el valor lógico de una condición.

Ejemplos:

c7=(A=3)&!(B=3)?1:0

Nota=(A<5)?'Suspenso':(((A>=5)&(A<7))?'Aprobado':(((A>=7)&(A<9))?'Notable':'Sobresaliente'))



En la siguiente escena se presenta un ejemplo en el que se incluyen algoritmos, incluyendo el de animación.



A la hora de programar escenas, es importante comprender cómo Descartes gestiona el orden de las actualizaciones tanto al iniciar una escena como durante la interacción del usuario con la misma.

Actualizaciones al iniciar la escena

Al cargar por primera vez la escena, se ejecuta lo siguiente en orden:

1. **Construcción interna de objetos:** se crean todos los objetos de la escena con sus valores por omisión.
2. **Fase de inicio:** una vez construidos los objetos, se actualizan en este orden:
    - Definiciones (vectores, matrices, etc.).
    - Algoritmos del programa.
    - Controles.
    - Espacios: incluye el cálculo de variables asociadas y la actualización de elementos gráficos.

3. **Fase de actualización de todos los valores:** asegura que los objetos utilicen valores actualizados en lugar de los predeterminados, lo cual es importante para la ejecución de eventos.

Cada elemento puede tener varias ejecuciones. Por ejemplo, las definiciones pueden incluir varias asignaciones de vectores y matrices, que se realizan en el orden indicado. Lo mismo aplica para los demás elementos. Esto determina cómo se muestra la escena al inicio y corresponde al resultado de pulsar **Aplicar** o **Aceptar** en el editor de configuraciones. También es útil conocer el comportamiento una vez iniciada la escena, cuando el usuario interactúa o la escena cambia, por ejemplo, debido a una animación.

Actualizaciones subsecuentes de la escena

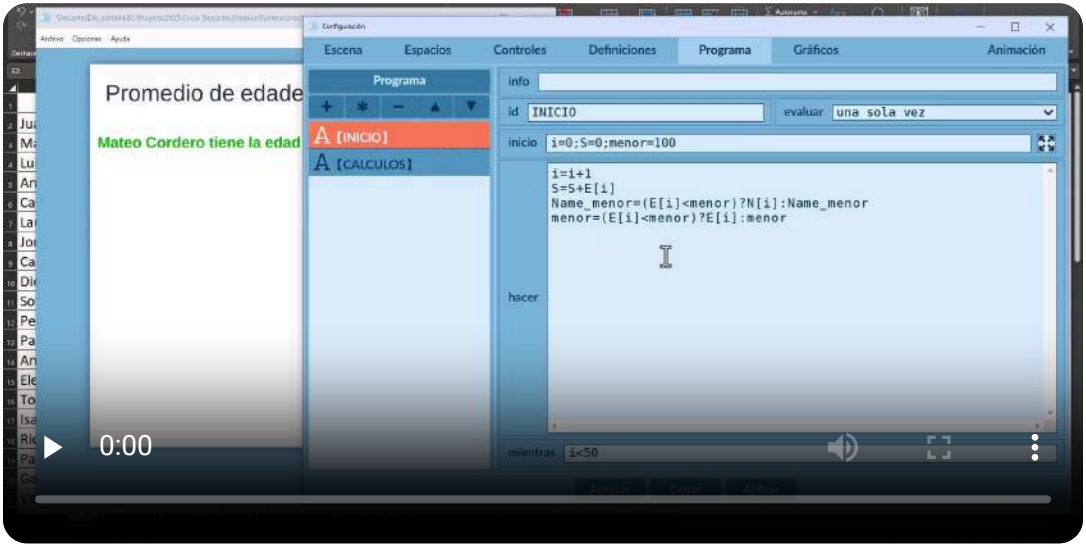
Cada interacción del usuario, o cada paso de una animación (si existe), provoca un ciclo de actualización con el siguiente orden:

1. **Actualización de auxiliares:** primero se actualizan los elementos auxiliares de las definiciones.
2. **Actualización de controles:** primera actualización de los controles.
3. **Actualización de eventos:** se procesan los eventos definidos en **Programa**.
4. **Actualización de controles:** segunda actualización de los controles para asegurar que queden correctamente sincronizados.
5. **Actualización de espacios:** se actualizan los elementos de los distintos espacios.

La segunda actualización de controles es necesaria cuando un control que aparece al final de la lista puede modificar a uno anterior; la actualización asegura que todas las modificaciones se implementen correctamente.

Como ejemplo práctico de programación avanzada, vamos a usar DescartesJS para leer datos desde una hoja de Excel. Aunque DescartesJS no lee directamente archivos Excel, sí permite trabajar con archivos en formato **txt**. Una forma sencilla de hacerlo es copiar los datos desde Excel, pegarlos en un bloc de notas y guardarlos como archivo **txt**. Luego se pueden cargar estos datos en un vector de DescartesJS. Es importante tener en cuenta que los datos tipo cadena deben ir entre **comillas simples** para que se lean correctamente.

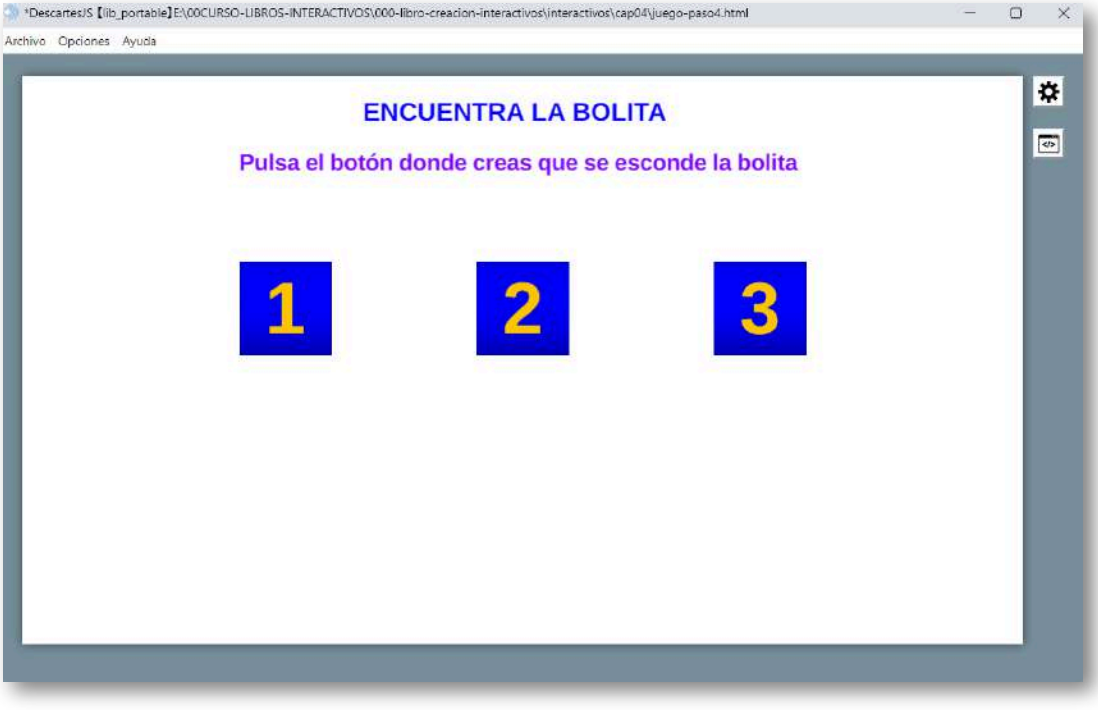
En el siguiente vídeo se muestra cómo leer 50 registros de una hoja de Excel y exportarlos a DescartesJS:



Este texto nos recuerda que la programación en DescartesJS ofrece una gran flexibilidad y creatividad: prácticamente cualquier idea puede convertirse en un ejemplo o ejercicio dentro de una escena interactiva. La reflexión principal es que, a medida que se avanza en los siguientes capítulos, se seguirá profundizando en estas herramientas y recursos, reforzando la capacidad de aplicar conceptos de programación para construir escenas más complejas y dinámicas.

4.16 Mi primer juego paso a paso. Paso 4

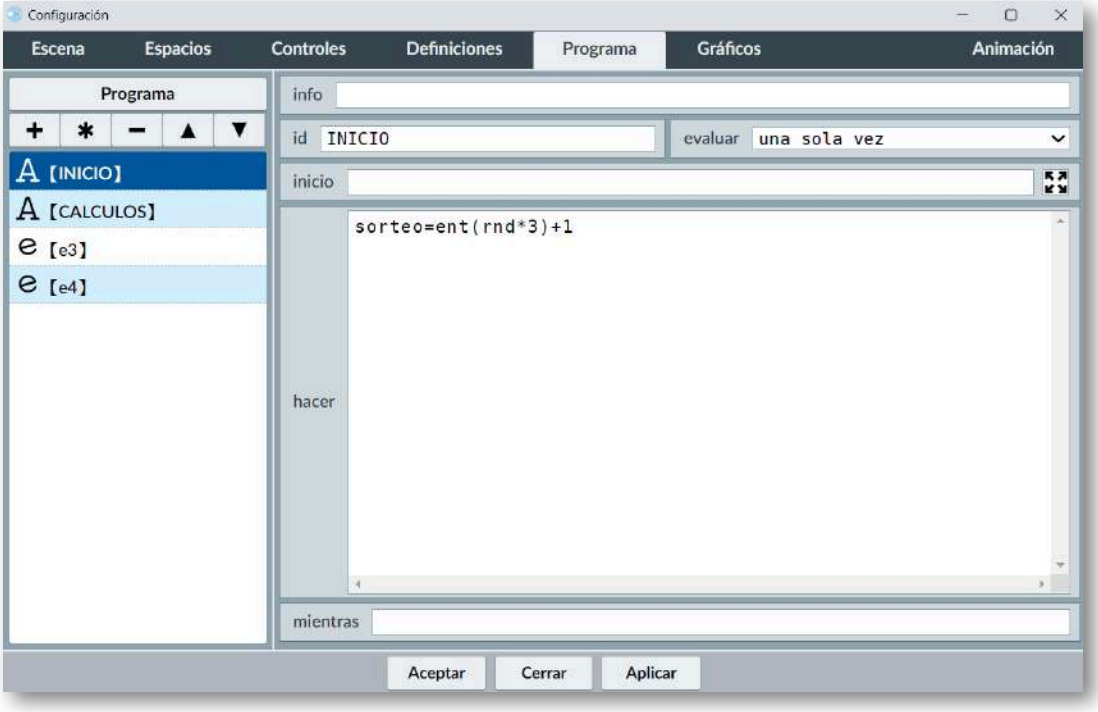
Vamos a realizar el paso 4 del juego. En primer lugar, abrimos el juego con el editor de **Descartes**, accediendo al menú *Archivo* > *Abrir* y seleccionando el archivo llamado **juego-paso3**.



Vamos a crear una variable llamada **sorteo** que generará aleatoriamente los valores 1, 2 o 3. Para ello, dentro del menú **Programa** y en el campo **hacer**, escribimos: **sorteo = ent(rnd*3) + 1**.

Explicación de la fórmula:

- **rnd** genera un número aleatorio entre 0 y 1.
- Multiplicamos **rnd** por 3 para obtener un número entre 0 y 3.
- **ent()** extrae la parte entera del número, dando 0, 1 o 2.
- Sumando 1, obtenemos los valores posibles: 1, 2 o 3.



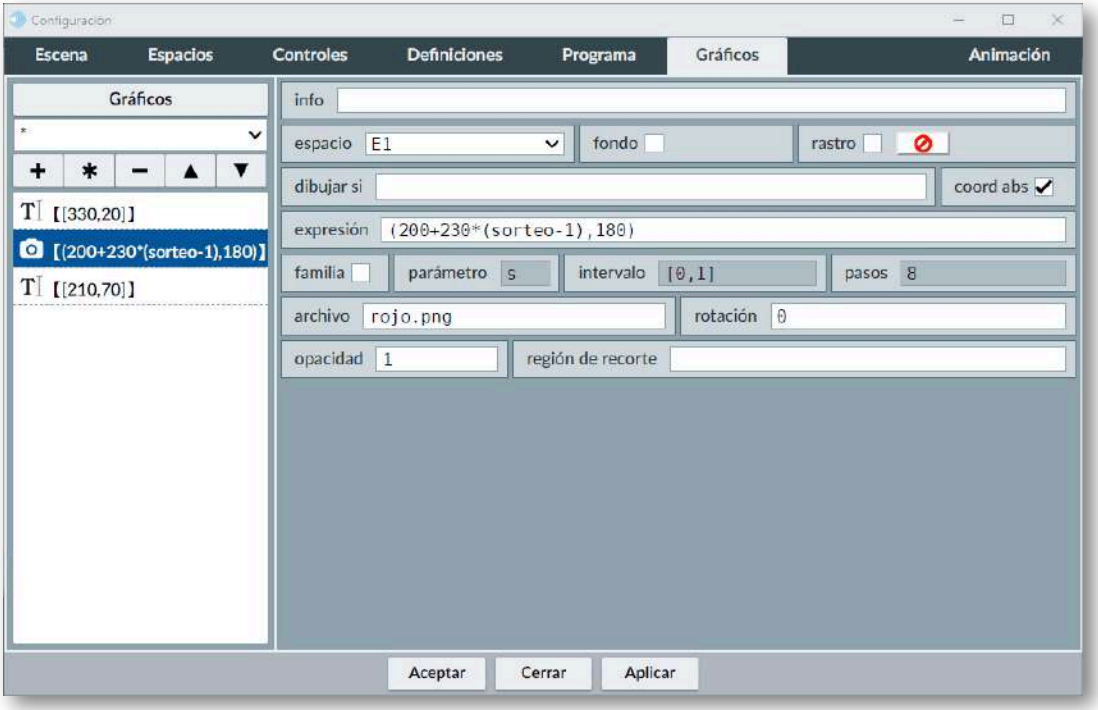
Ahora relacionaremos la posición de la bolita con el valor de **sorteo** (1, 2 o 3). Para ello, definimos la posición de la bolita en el campo *expresión* de un elemento tipo texto de la siguiente forma:

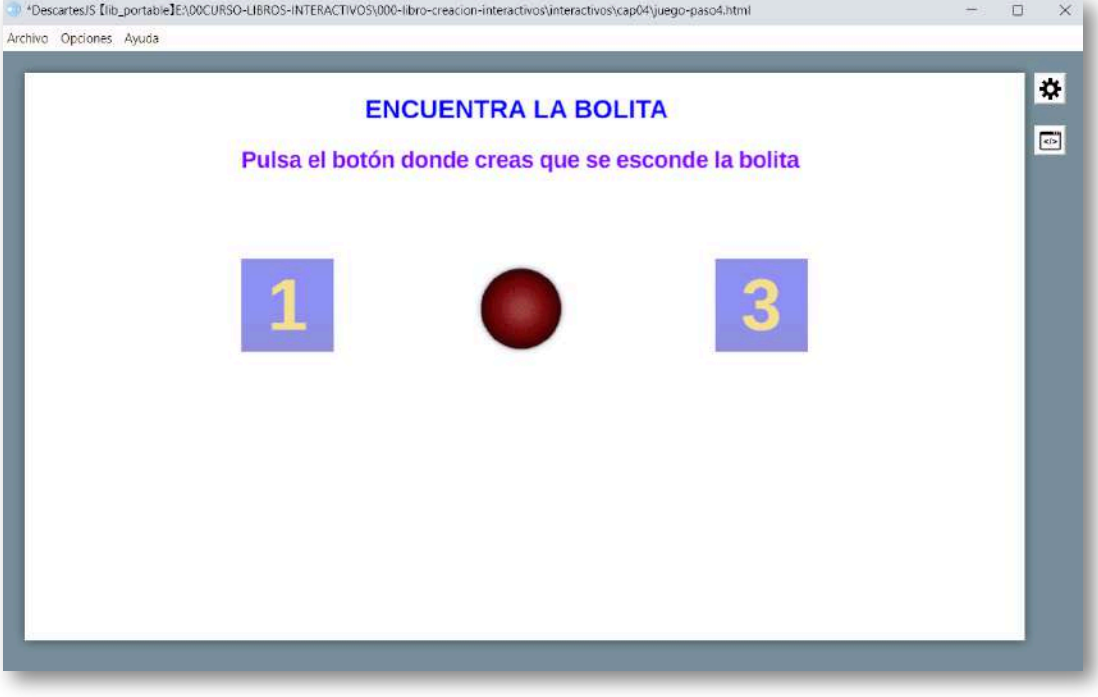
```
(x, y) = (200 + 230 * (sorteo - 1), 180)
```

De este modo:

- Si **sorteo = 1**, la bolita aparecerá detrás del control 1.
- Si **sorteo = 2**, aparecerá detrás del control 2.
- Si **sorteo = 3**, aparecerá detrás del control 3.

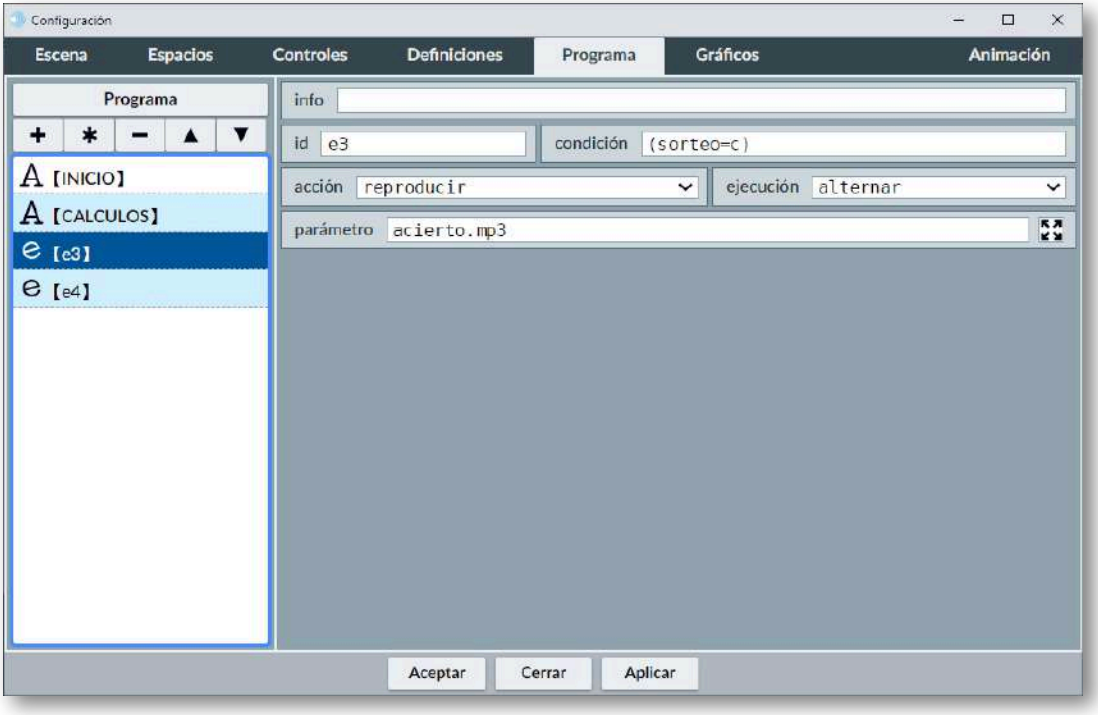
Se muestran dos capturas de pantalla: una con la acción realizada y otra con la visualización resultante de la escena. Se recuerda que el control tipo botón que es pulsado desaparece y que el resto de botones pasa a la situación de inactivo.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
[[330,20]]
[(200+230*(sorteo-1),180)]
[[210,70]]
info
espacio E1
fondo
rastro
dibujar si
coord abs
expresión (200+230*(sorteo-1),180)
familia
parámetro s
intervalo [0,1]
pasos 8
archivo rojo.png
rotación 0
opacidad 1
región de recorte
Aceptar
Cerrar
Aplicar


*DescartesJS [lib_portable]E:\00CURSO-LIBROS-INTERACTIVOS\000-libro-creacion-interactivos\interactivos\cap04\juego-paso4.html
Archivo Opciones Ayuda
ENCUENTRA LA BOLITA
Pulsa el botón donde creas que se esconde la bolita
1
3

A continuación, incorporamos los efectos sonoros mediante dos archivos de audio: uno para el acierto y otro para el fallo. Estos archivos, al igual que las imágenes, deben colocarse en la misma carpeta que el archivo de la escena.

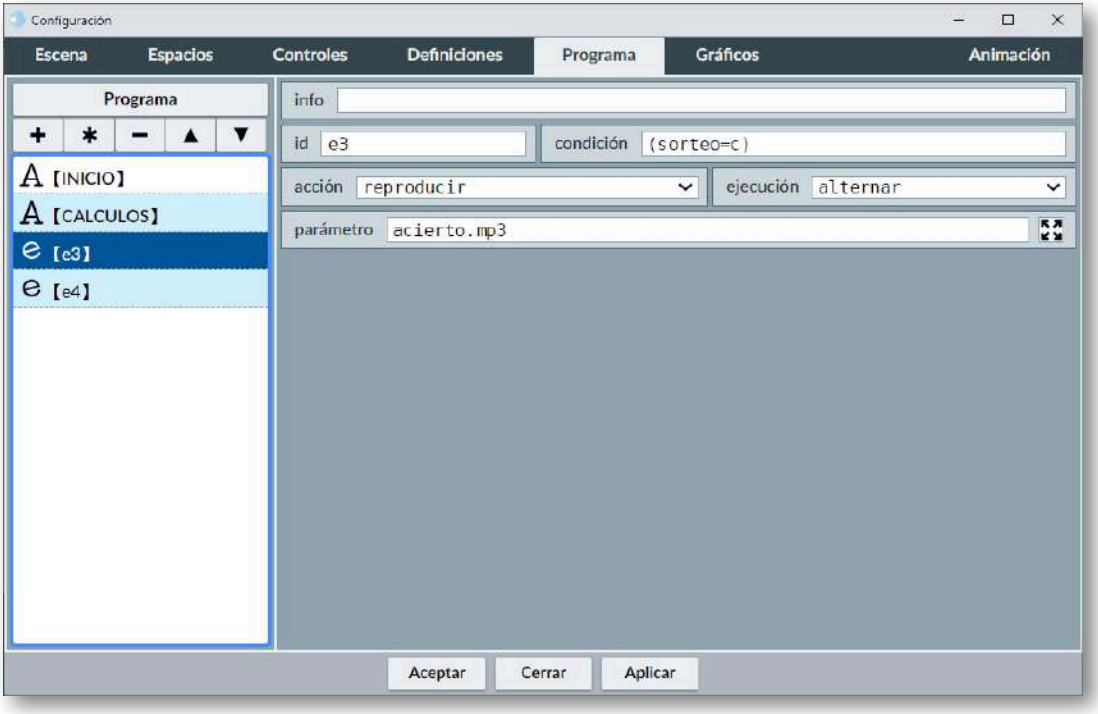
Después, creamos un evento desde el menú **Programa** (botón **+ Agregar evento**). La condición será **(sorteo = c)** y la acción consistirá en reproducir el archivo **acierto.mp3**.



Copiamos el evento anterior (botón *) y modificamos la condición a **(sorteo#c)&(c>0)**, estableciendo como acción la reproducción del archivo **fallo.mp3**, que será el sonido en caso de error.

La condición **(c>0)** indica que previamente debe haberse pulsado un control para seleccionar la bolita.

A continuación, presentamos la escena del juego en el estadío actual.

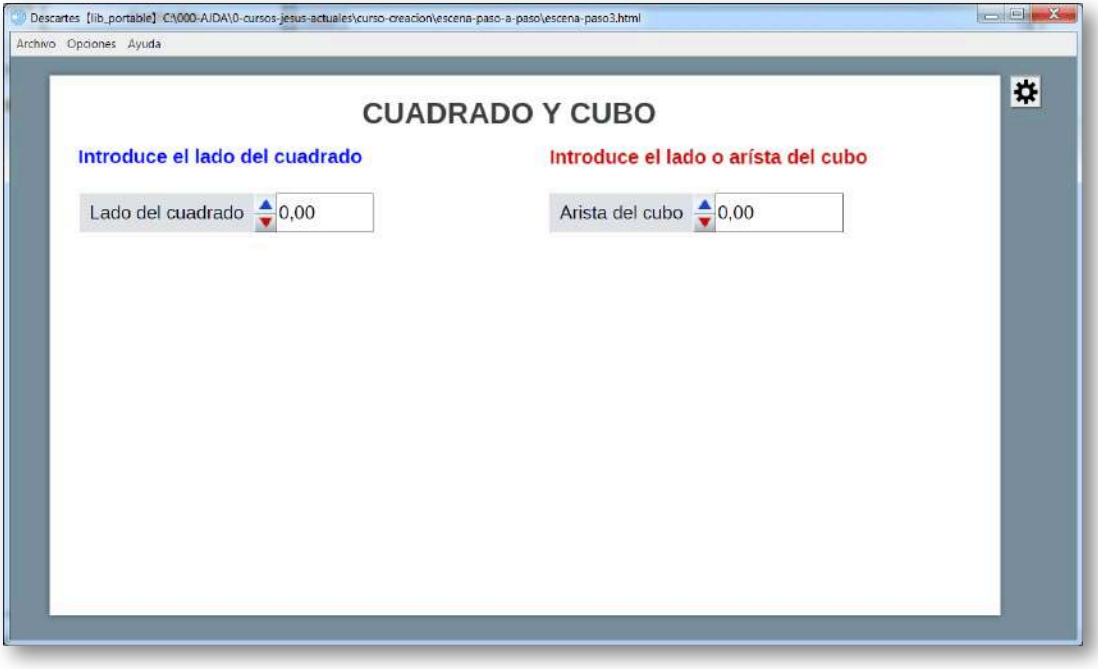




Guardamos la escena del juego con el nombre **juego-paso4**.

4.17 Mi primera escena paso a paso. Paso 4

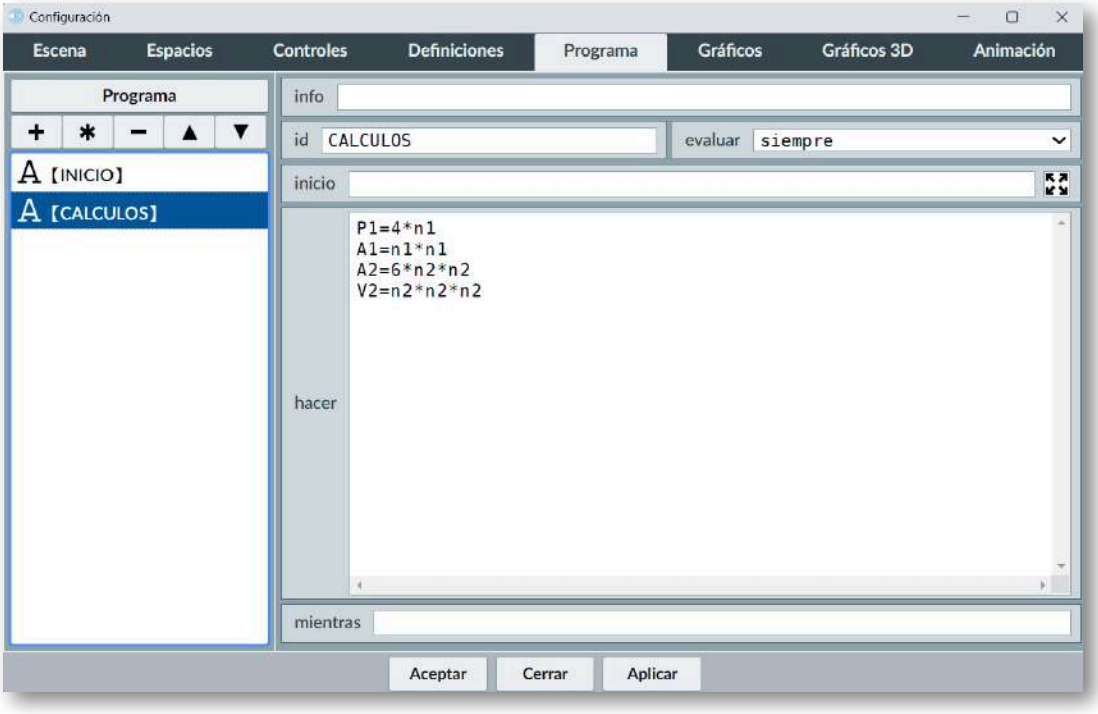
Vamos a realizar el paso 4 de la primera escena. Para comenzar, abrimos la escena con el editor de **Descartes**, accediendo al menú *Archivo* > *Abrir* y seleccionando el archivo llamado **escena-paso3**. Esto nos permitirá continuar con la configuración de los controles y variables necesarios para el desarrollo del interactivo.

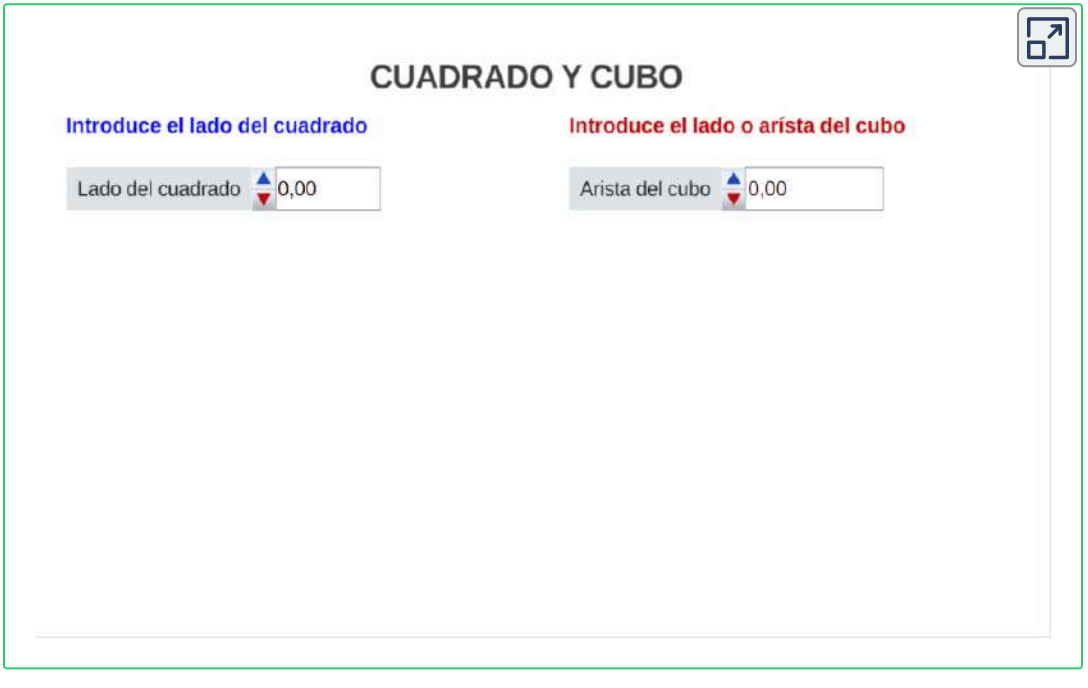


A continuación, en el menú **Programa**, dentro del algoritmo **Cálculos**, añadimos las siguientes expresiones:

```
P1 = 4*n1
A1 = n1*n1
A2 = 6*n2*n2
V2 = n2*n2*n2
```

De este modo, al modificar el valor del lado mediante los controles, se actualizarán automáticamente los valores de **P1**, **A1**, **A2** y **V2**.





Guardamos la escena con el nombre escena-paso4.



Capítulo 5

Gráficos 2D

Capítulo 5
Gráficos 2D

5.1 Introducción 355

5.2 Inclusión gráficos 2D 356

5.3 Parámetros comunes a los gráficos 360

5.4 Gráfico Ecuación 363

5.5 Gráfico Curva 370

5.6 Gráfico Sucesión 374

5.7 Gráfico Polígono 378

5.8 Gráfico Rectángulo 381

5.9 Gráfico Arco 384

5.10 Gráfico Punto 387

5.11 Gráfico Segmento 391

5.12 Gráfico Flecha 394

5.13 Gráfico Relleno 398

5.14 Gráfico Macro 401

5.15 Gráfico Imagen 420

5.16 Gráfico Texto 430

5.17 Familias en gráficos 2D 445

5.18 Mi primer juego paso a paso. Paso 5 454

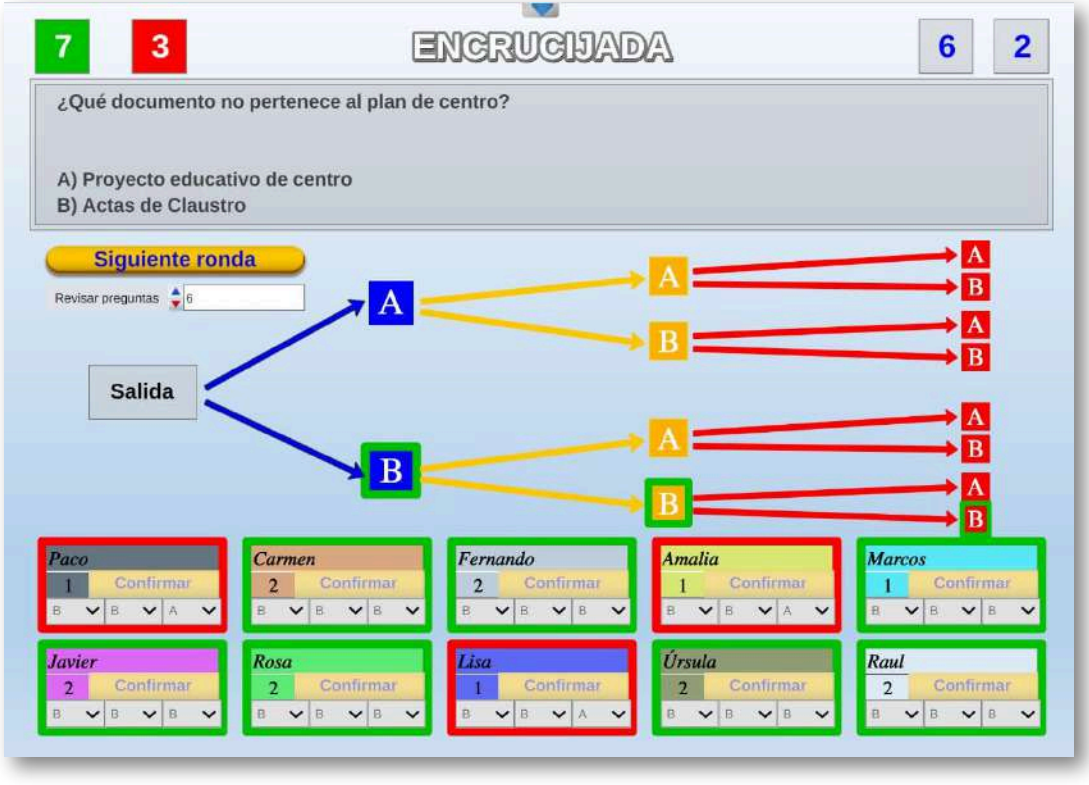
5.19 Mi primera escena paso a paso. Paso 5 460

5.1 Introducción

Nos adelantamos al selector **Gráficos 2D**, ya que los gráficos constituyen el siguiente paso natural después de definir los espacios. Estos permiten incorporar una amplia variedad de figuras y textos que enriquecen el interactivo y amplían sus posibilidades didácticas.

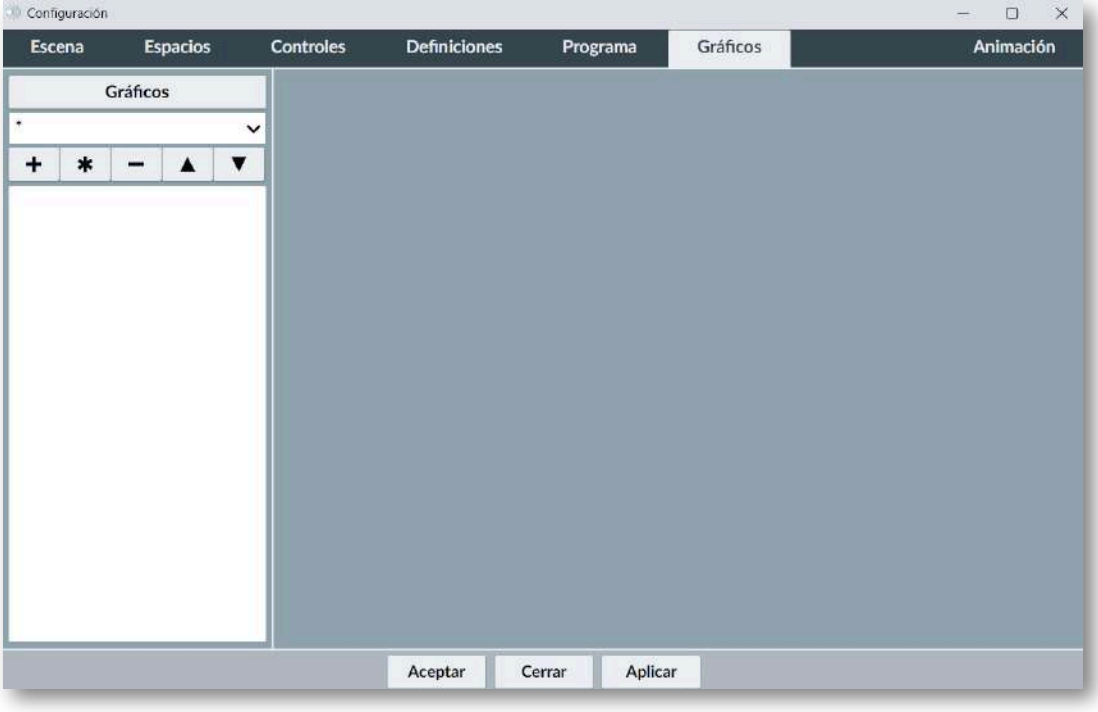
En los espacios 2D se pueden añadir distintos elementos gráficos como puntos, flechas, polígonos, arcos, rellenos, textos, imágenes, segmentos, macros, entre otros. Estos recursos se utilizan con gran frecuencia en la construcción de escenas y resultan especialmente útiles en la creación de juegos educativos, donde aportan dinamismo, claridad visual e interactividad.

Mostramos una captura de pantalla del juego AJDA *Encrucijada*, en la que se pueden apreciar múltiples elementos de **Gráficos 2D**.



5.2 Inclusión gráficos 2D

Todos los gráficos comparten un panel a la izquierda, similar al del selector **Espacio**. En este panel se muestra una lista de los gráficos existentes.



Al igual que en el selector Espacio, dispone de un botón para añadir un nuevo gráfico, otro para duplicar el gráfico seleccionado, un control para mover el gráfico hacia arriba o hacia abajo en la lista, y un botón para eliminar el gráfico seleccionado.

Los gráficos sólo pueden existir dentro de un espacio. Por ello, en la parte superior del panel izquierdo existe un menú desplegable que muestra los espacios disponibles y que funciona como filtro.

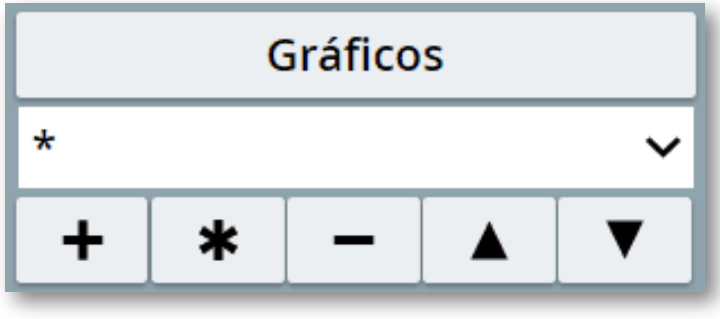
Por defecto aparece el símbolo de asterisco (*), lo que indica que se muestran todos los gráficos de todos los espacios (filtro desactivado).

Si se selecciona un espacio concreto en el menú, únicamente se visualizarán los gráficos pertenecientes a dicho espacio, lo que facilita su identificación cuando existen muchos elementos.

Cuando hay más de un gráfico en un mismo espacio y estos se intersecan, se dibujan según el orden en que aparecen en el panel izquierdo. El gráfico situado más arriba en la lista se pinta primero, y el que aparece más abajo se pinta al final, pudiendo cubrir a los que se encuentran por encima en la lista.

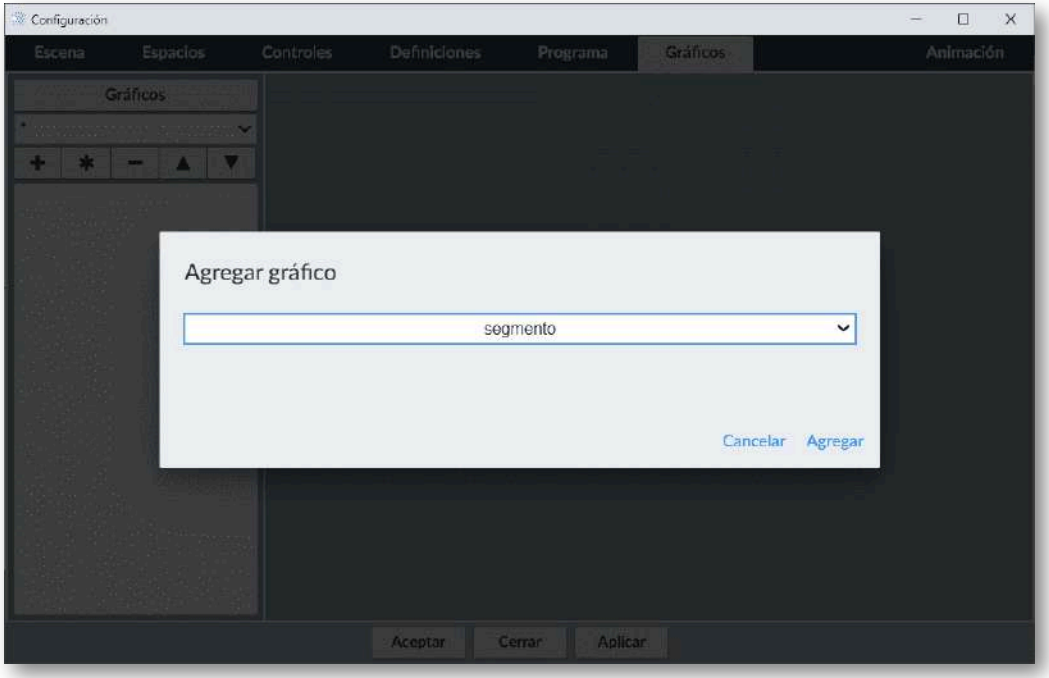
El procedimiento para introducir un elemento gráfico en una escena es el siguiente:

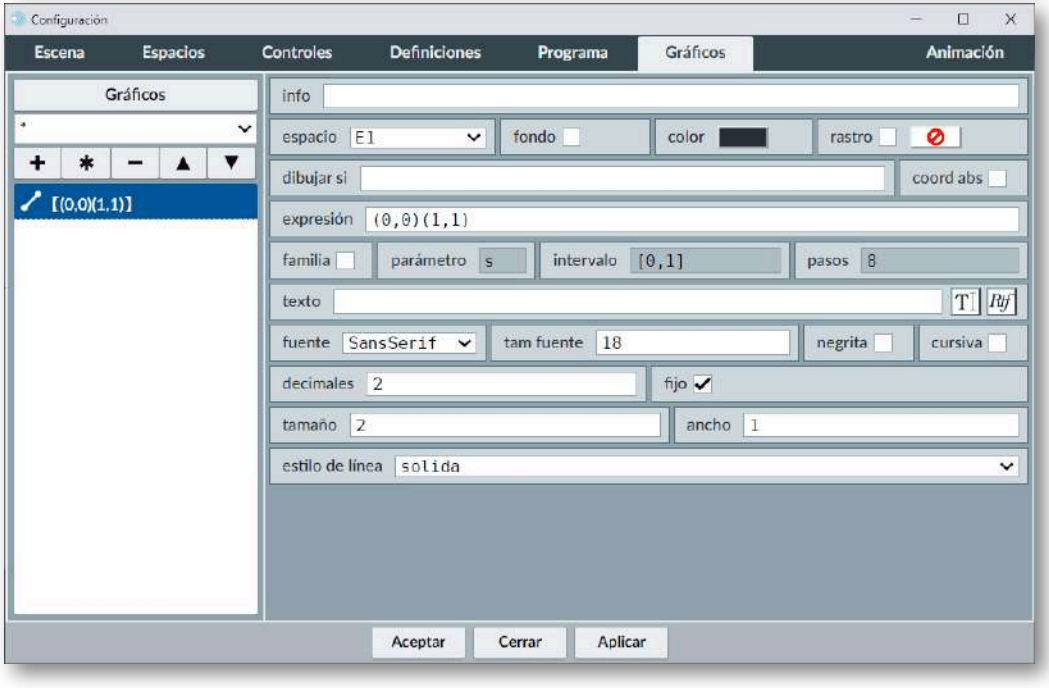
1. Acceder al panel de configuración de la escena y pulsar sobre el selector **Gráficos**.
2. Añadir un nuevo elemento desde el panel izquierdo.
    1. Pulsar el botón **+** para crear un nuevo gráfico (el botón **-** elimina el elemento seleccionado, * lo duplica y el control de desplazamiento permite subirlo o bajarlo en la lista). El botón **Gráficos** permite visualizar, editar y modificar su código.



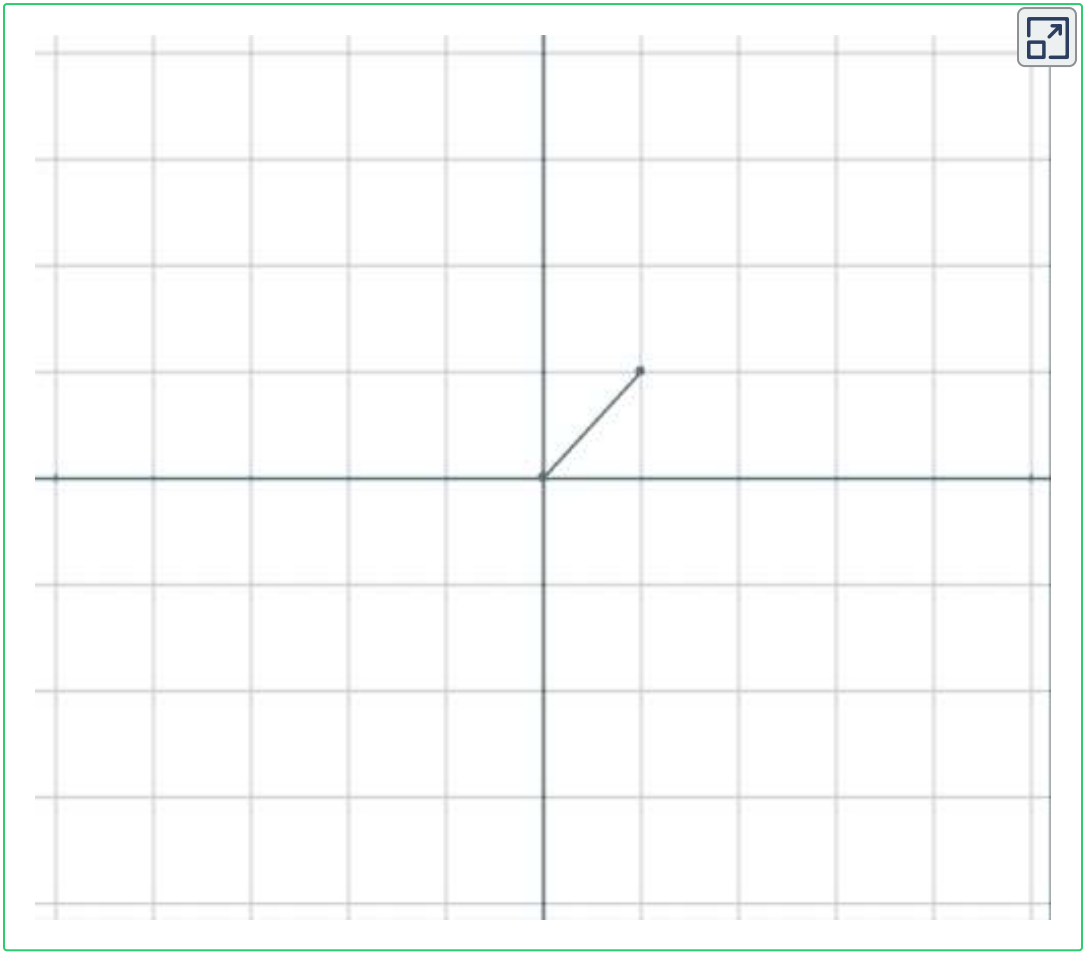
    2. Seleccionar el tipo de elemento gráfico mediante el menú desplegable correspondiente.
    3. Configurar los parámetros del elemento gráfico desde el panel de edición.

Por ejemplo, si se inserta un **segmento**, aparecerá el menú específico para su configuración en el editor.





A continuación, se muestra la escena creada.



A la izquierda se muestra la lista de todos los gráficos de la escena (menú de selección). A la derecha aparece el panel de edición, que varía según el tipo de gráfico y permite modificar todos los parámetros que lo definen.

En el menú desplegable situado encima de la etiqueta azul **Gráficos**, si se selecciona un espacio concreto, sólo se mostrarán los gráficos pertenecientes a ese espacio; si se selecciona *, se visualizarán todos los gráficos de la escena.

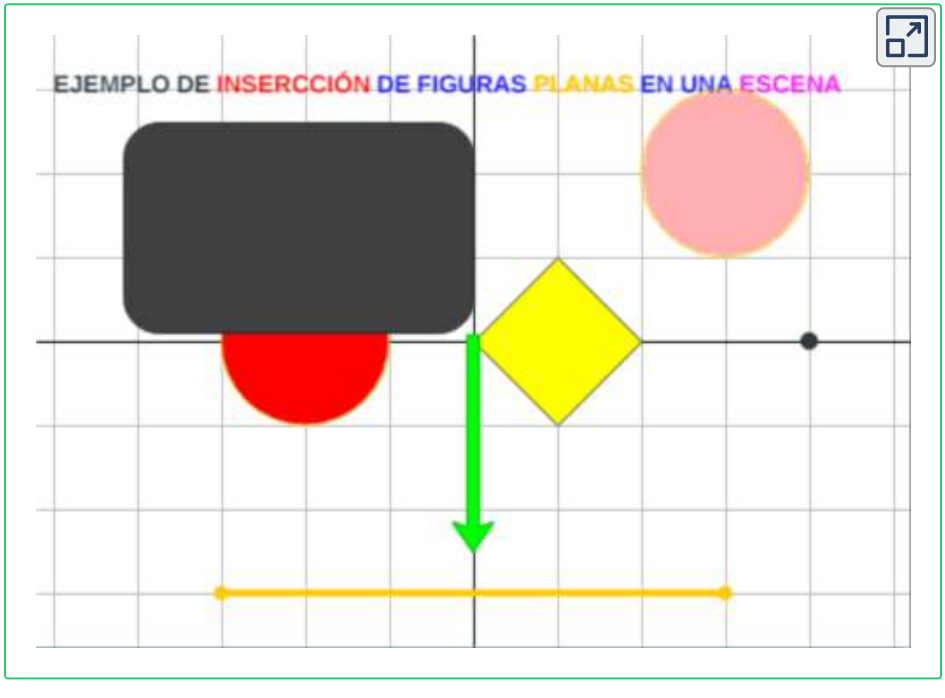
5.3 Parámetros comunes a los gráficos

A continuación, se relacionan los parámetros comunes en elementos gráficos de espacios 2D:

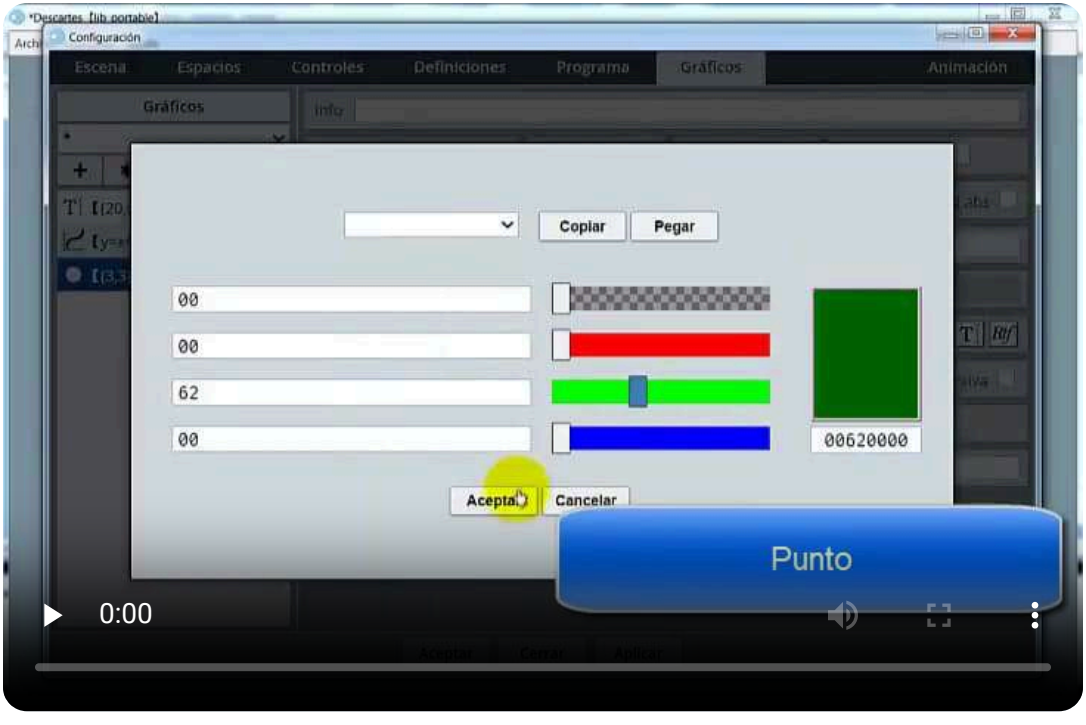
- **Menú de espacios:** Permite seleccionar a qué espacio mover el gráfico, diferente del menú de filtrado de gráficos (cada elemento gráfico solo puede pertenecer a un espacio).
- **Fondo:** Checkbox que, al estar seleccionado, dibuja el gráfico una sola vez al cargar la escena; no se actualizará con cambios posteriores en variables, salvo cambios de escala o posición del espacio.
- **Dibujar si:** Campo de texto para una condición booleana que determina si el gráfico se pinta o no.
- **Coordenadas absolutas:** Checkbox que define si el gráfico se pinta en coordenadas absolutas (origen esquina superior izquierda, pixeles) o relativas (origen plano cartesiano, unidades según escala).
- **Texto:** Campo de texto con botones T y Rtf para introducir texto simple o enriquecido cercano al gráfico.
- **Decimales:** Campo para indicar el número de decimales a mostrar en valores asociados al gráfico.
- **Fijo:** Checkbox que fuerza que siempre se muestre el número de decimales indicado, aunque no sean significativos.
- **Fuente:** Menú para seleccionar la fuente del texto: SansSerif, Serif o Monospaced.
- **Tamaño fuente:** Campo de texto para definir tamaño del texto en pixeles.
- **Negrita:** Checkbox para mostrar el texto en negritas.
- **Cursiva:** Checkbox para mostrar el texto en cursiva.
- **Ancho:** Número de píxeles de grosor del gráfico.

- **Rastro:** Checkbox con control de color que deja rastro de posiciones y formas anteriores cuando el gráfico cambia dinámicamente.
- **Familia:** Checkbox que permite repetir el trazo del gráfico varias veces según parámetros, intervalo y pasos definidos.
- **Parámetro:** Variable que varía en cada paso dentro del intervalo definido.
- **Intervalo:** Campo de texto con dos valores [inicio, fin] que define el rango de la variable del parámetro.
- **Pasos:** Número de pasos que divide el intervalo de manera equidistante para la variable del parámetro.
- **Info:** Comentario interno del elemento en la escena.
- **Estilo de línea:** Menú con opciones: sólida, punteada o discontinuas para el contorno del gráfico.

Mostramos un ejemplo de escena con diferentes gráficos 2D.



Presentamos un vídeo sobre los gráficos 2D junto con un interactivo de ejemplos de los diferentes tipos.



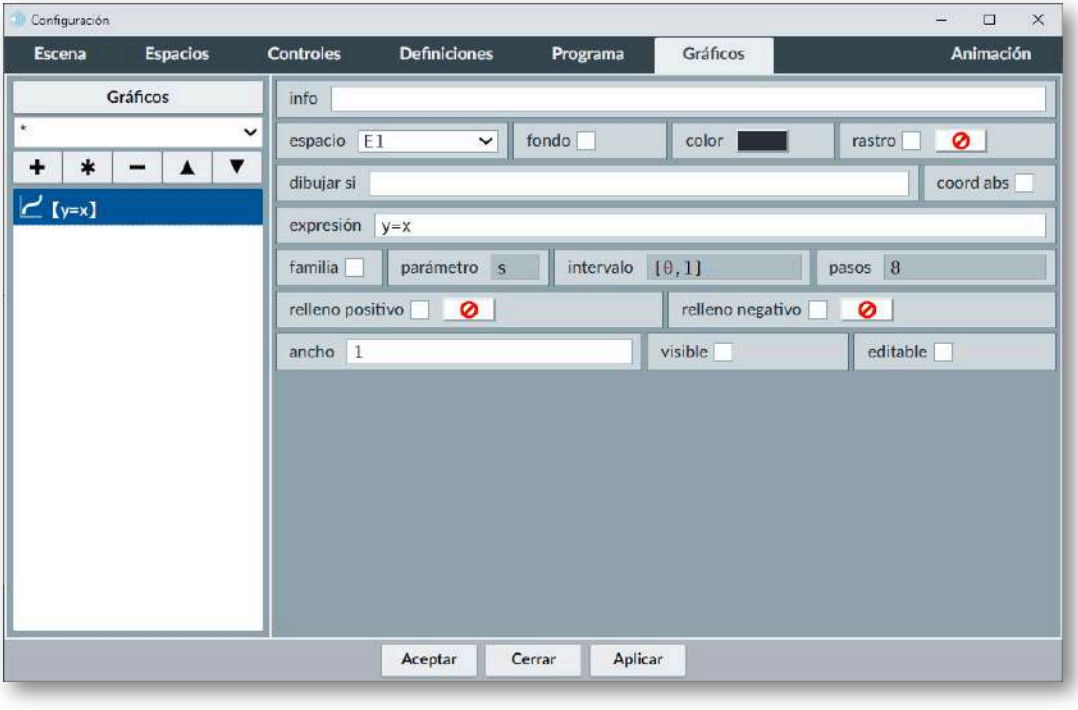


5.4 Gráfico Ecuación

A continuación, se relacionan los parámetros comunes en elementos gráficos de espacios 2D:

Una **ecuación** es un elemento gráfico bidimensional que permite representar funciones matemáticas del tipo $y = f(x)$, es decir, expresar la variable dependiente y en función de la variable independiente x, o viceversa.

El menú de edición de una ecuación presenta el siguiente aspecto:



Se describen los campos específicos de las ecuaciones:

- **Expresión:** Campo de texto donde se introduce la ecuación a graficar. Utiliza las variables x y y para referirse a las abscisas y ordenadas del plano. Puede trazarse cualquier ecuación, no necesariamente una función explícita. Por ejemplo, $y = x$, $x^2 + y^2/2 = 1$ o $x = 3$.

- **Relleno positivo:** Checkbox que permite rellenar el área entre el eje y la gráfica con el color seleccionado, cuando la gráfica queda por encima del eje correspondiente (o a la derecha, si es del tipo *x = f(y)*). El selector de color se activa sólo si esta opción está marcada.
- **Relleno negativo:** Checkbox que permite rellenar el área entre el eje y la gráfica con el color seleccionado cuando la gráfica queda por debajo del eje correspondiente (o a la izquierda, si es del tipo *x = f(y)*).
- **Visible:** Checkbox que muestra la ecuación en una barra situada en la parte inferior del interactivo, siempre que esté activado.
- **Editable:** Checkbox que sólo funciona si la opción **Visible** está marcada. Permite modificar la ecuación directamente en el interactivo, actualizando la gráfica de forma dinámica.

Existe un pequeño conjunto de **formas canónicas** para expresar ecuaciones de rectas y ciertas curvas que reciben un tratamiento especial. Su utilidad principal es que permiten que el dibujo se realice de manera mucho más rápida. Las formas canónicas que reconoce Descartes son:

y = m*x + b

x = m*y + a

A*x + B*y = C

e = sqrt((x-Fx)^2 + (y-Fy)^2)/(d + (x-Fx)*cos(t) + (y-Fy)*sen(t))

e = sqrt((x-Fx)^2 + (y-Fy)^2)/(d - ((x-Fx)*(Dx-Fx) + (y-Fy)*(Dy-Fy))/d)

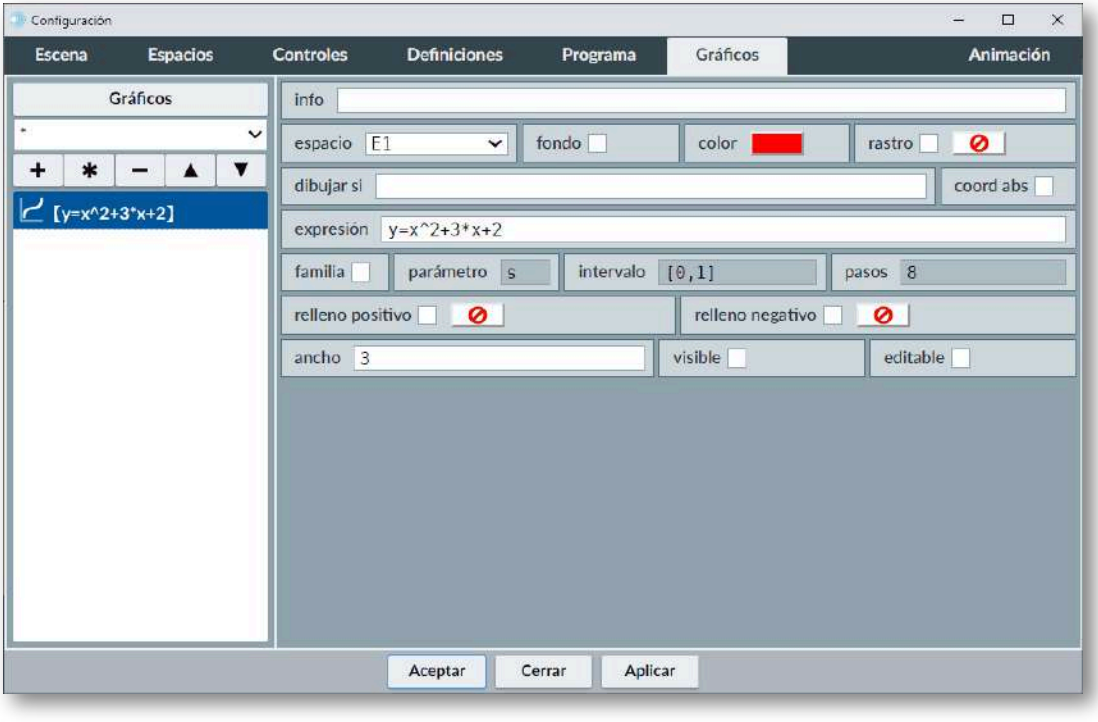
Para que el programa las reconozca y aplique el procedimiento optimizado de dibujo, deben escribirse exactamente como aparecen, sin modificar letras ni estructura. Su única ventaja radica en la rapidez con que se representan gráficamente.

Hay un detalle importante al utilizar el operador potencia (**^**) en un gráfico de tipo ecuación. Las funciones del tipo *y = x^p*, cuando *p* no es un número entero, sólo están definidas para *x > 0*.

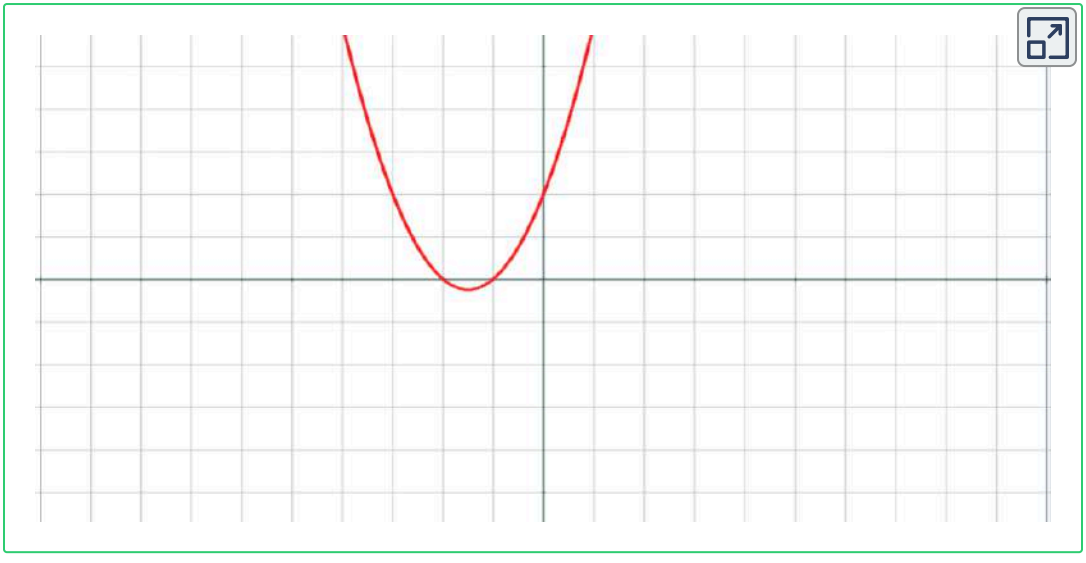
Por ejemplo, si se introduce únicamente *y = x^(3/5)*, la gráfica aparecerá sólo en la región donde *x > 0*. En cambio, si se desea obtener la gráfica completa, incluyendo los valores negativos de *x*, será necesario escribir *y = (x^3)^(1/5)*.

Este comportamiento se debe a la forma en que el programa interpreta las raíces cuando se utiliza el operador de potencia.

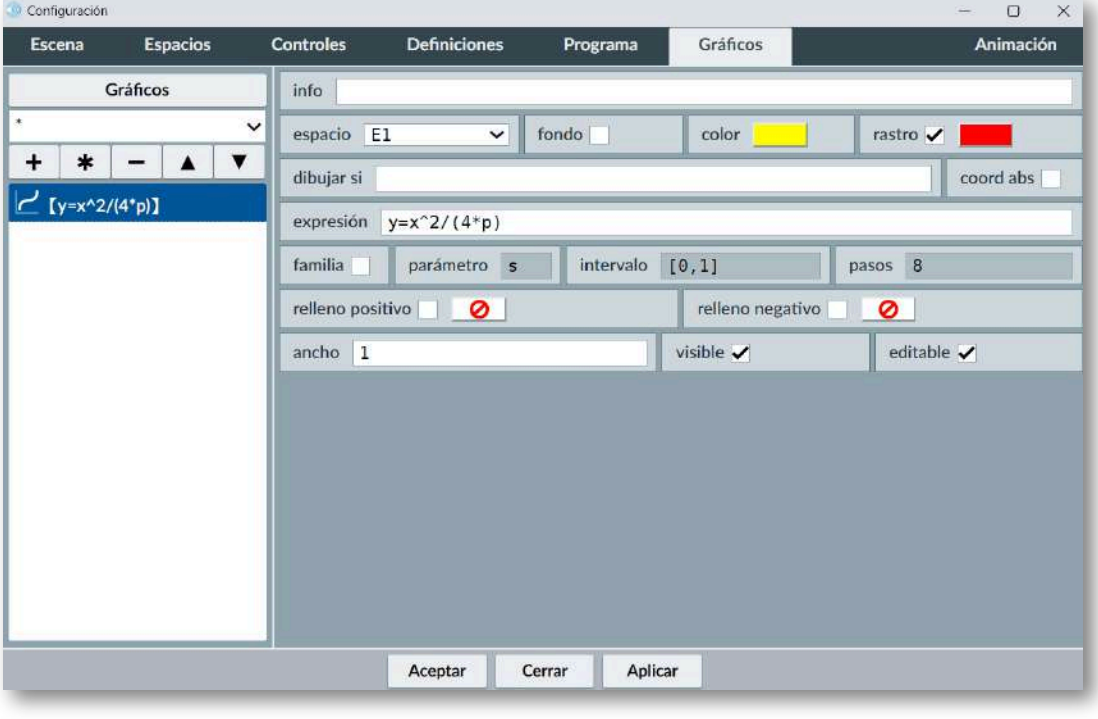
Un ejemplo sencillo de gráfico de tipo ecuación se muestra a continuación. Hemos modificado el ancho a 3 y el color del gráfico a rojo.

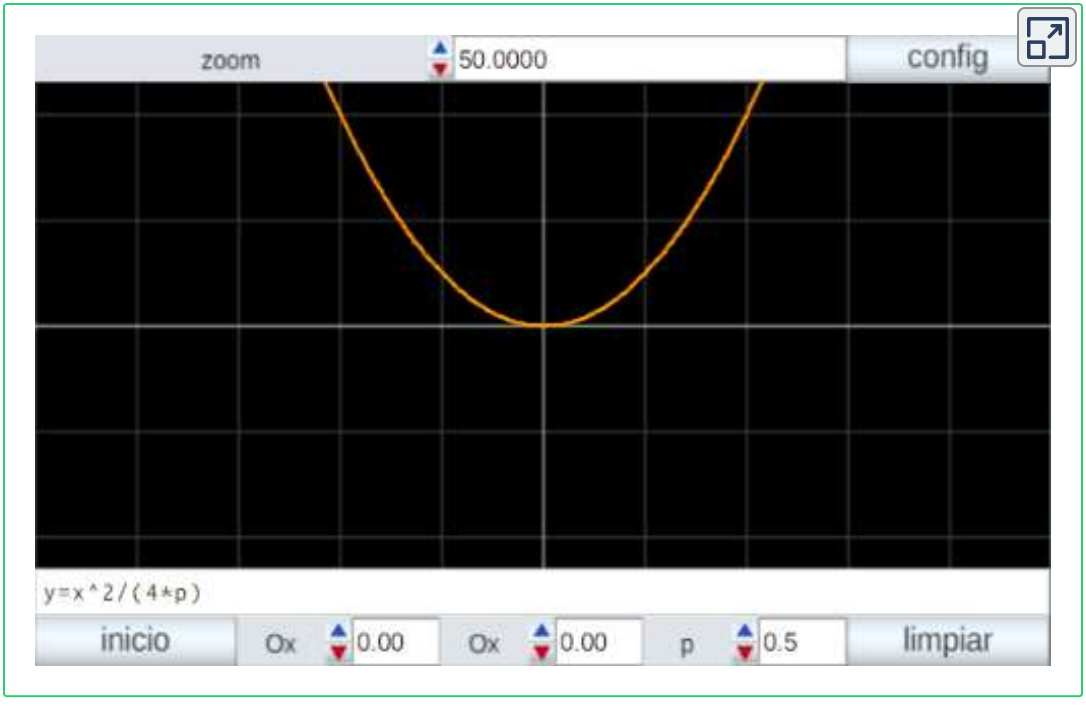


La escena que incluye esta ecuación se presenta de la siguiente forma:

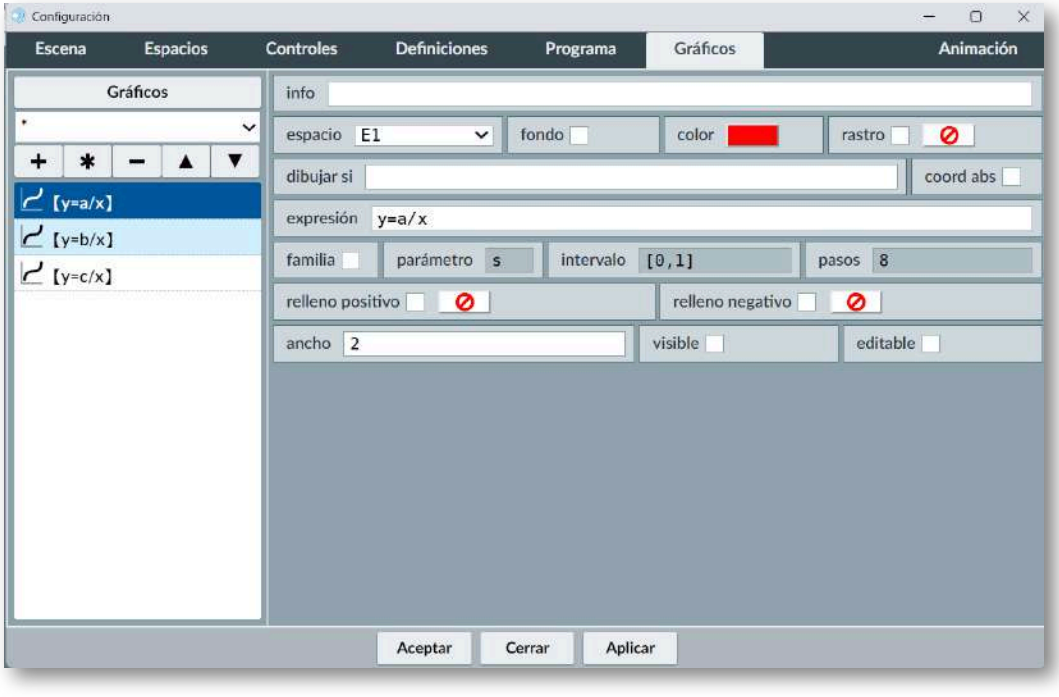


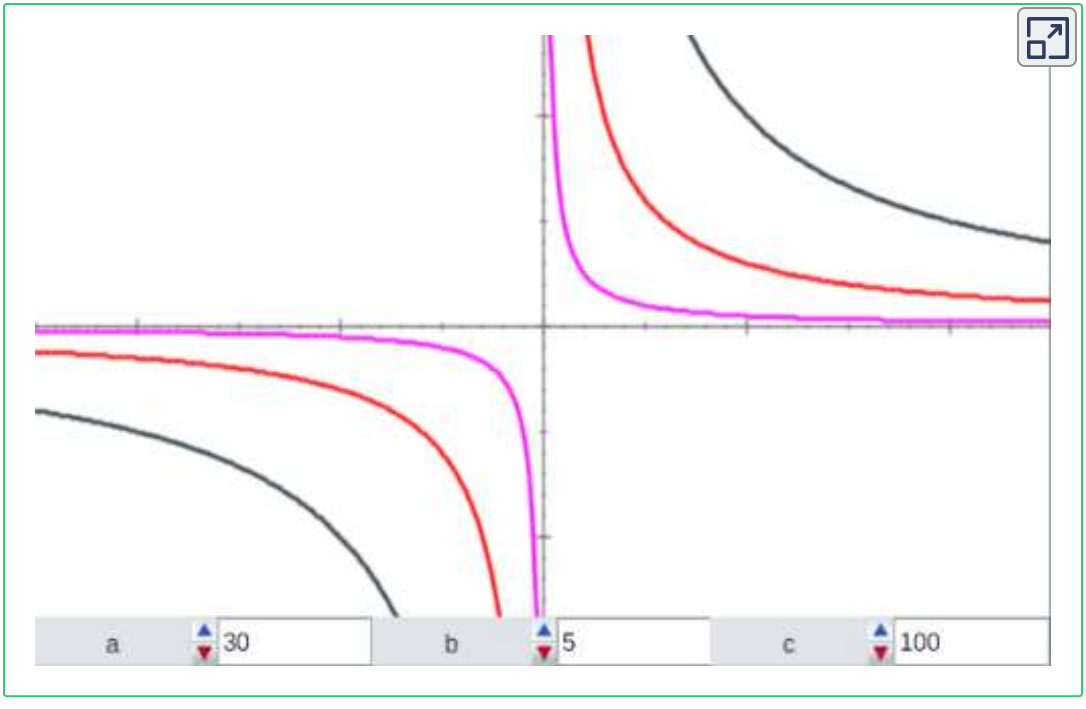
Incluimos a continuación otro ejemplo con una ecuación que incorpora un parámetro modificable mediante un control; la ecuación es visible, editable y además deja rastro.



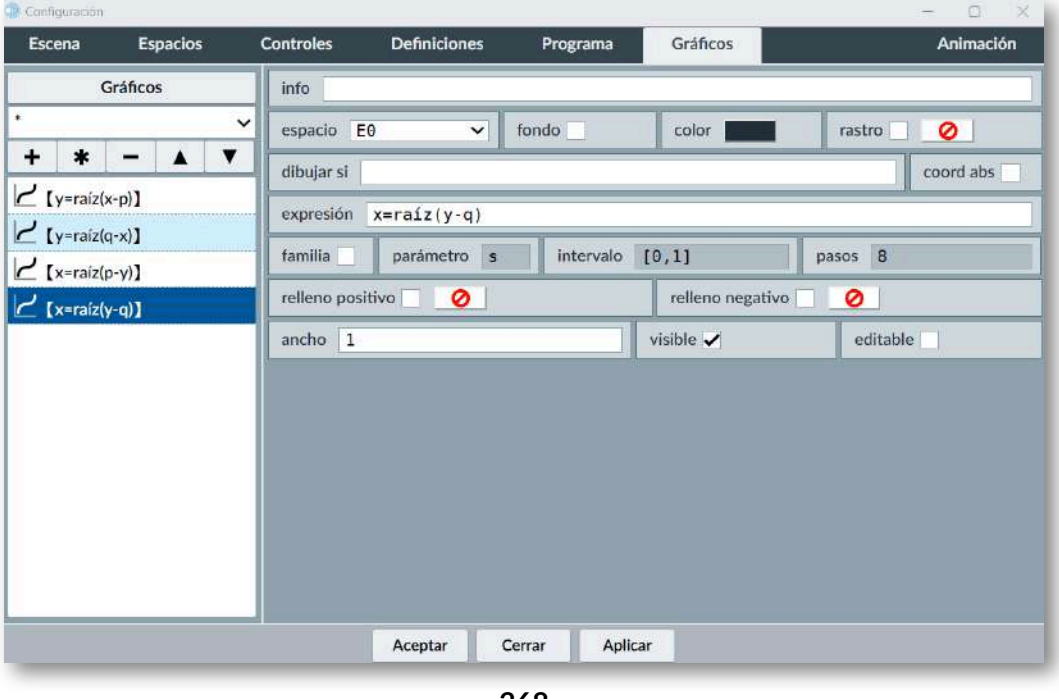


Mostramos otro ejemplo con varias ecuaciones parametrizadas.





Otro ejemplo corresponde a una función definida a trozos.





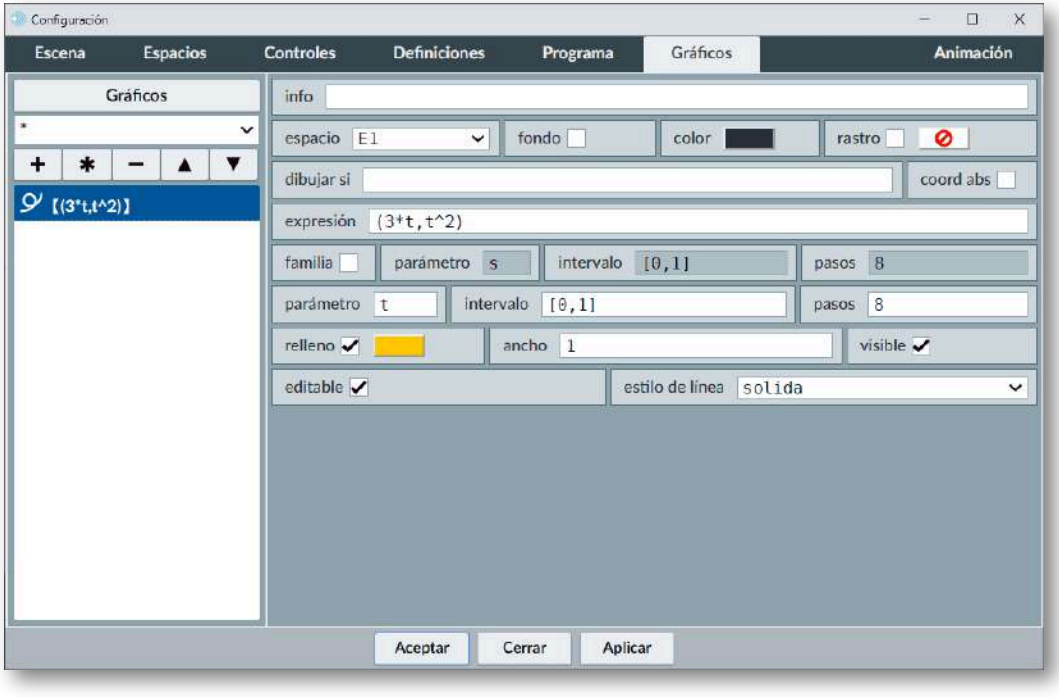
Finalmente mostramos la escena de una ecuación senoidal parametrizada.



5.5 Gráfico Curva

Este gráfico consiste en varios trazos que unen puntos determinados por un parámetro. Dicho parámetro es una variable que toma valores dentro de un intervalo dividido en un número determinado de pasos. De este modo, la representación puede entenderse como una sucesión de segmentos consecutivos unidos por sus extremos.

Una **curva** es un elemento que permite realizar representaciones bidimensionales de dos funciones en función de un parámetro común *t*. El menú de edición de una curva presenta el siguiente aspecto:

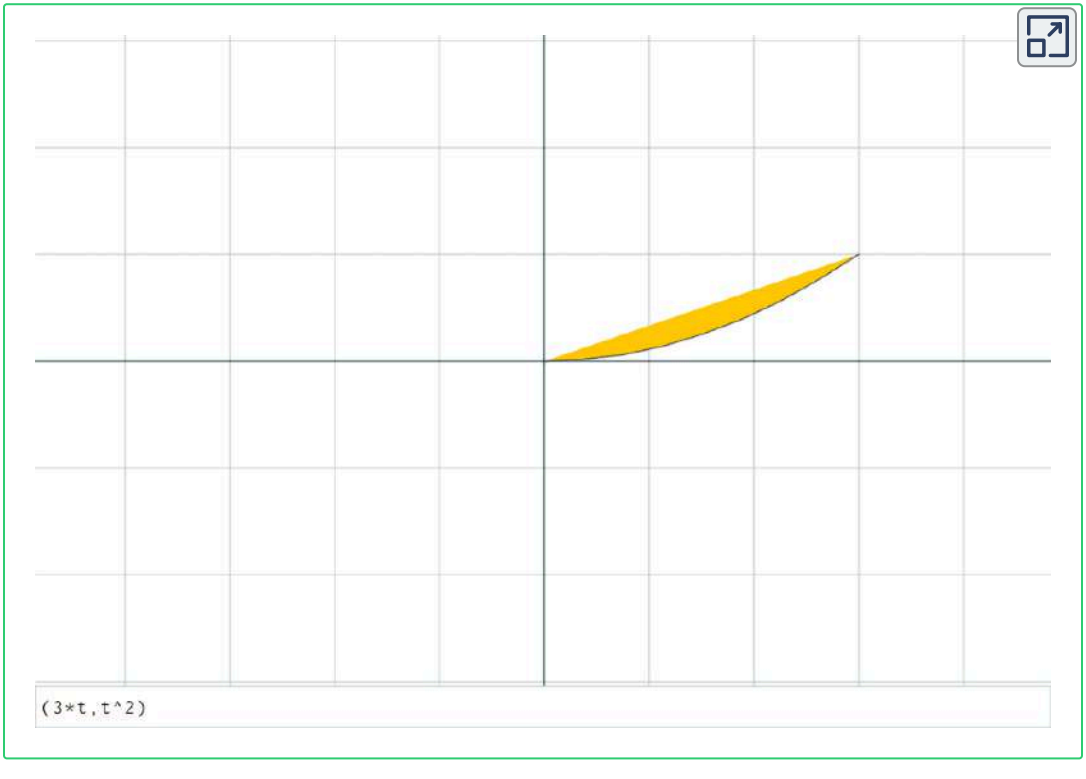


Los parámetros propios de las curvas son los siguientes:

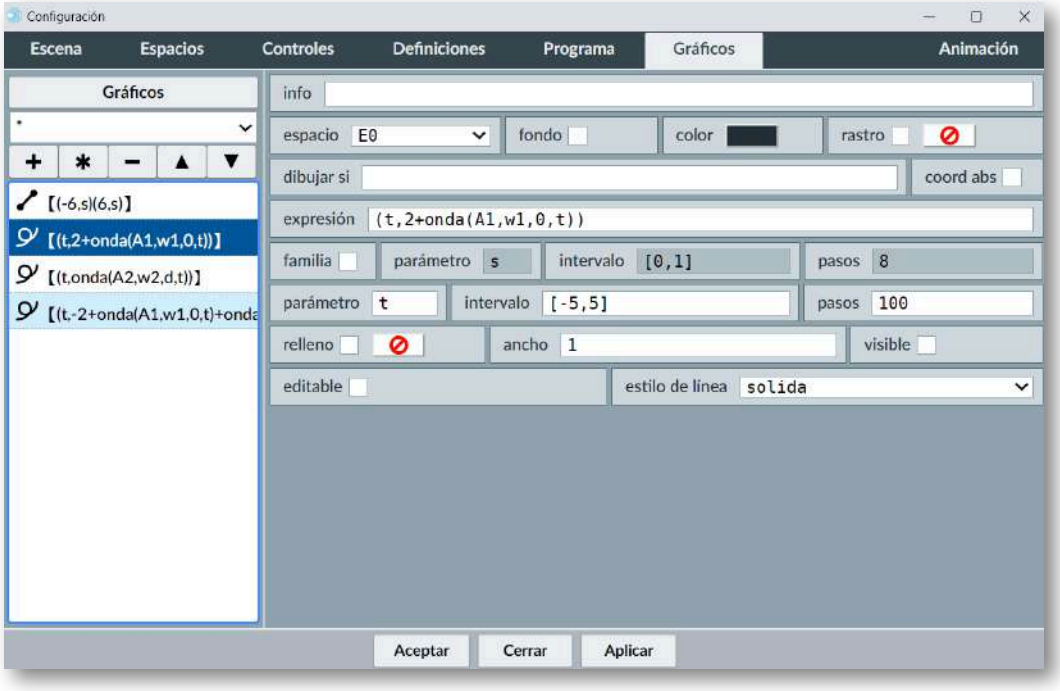
- **expresión:** Es un campo de texto donde se escribe la sucesión de puntos que representan los pares ordenados de los extremos de los segmentos que componen la curva. Para el gráfico *curva*, la expresión es del tipo (t, t), es decir, un par ordenado de funciones de la variable *t*, que es el parámetro de la curva.

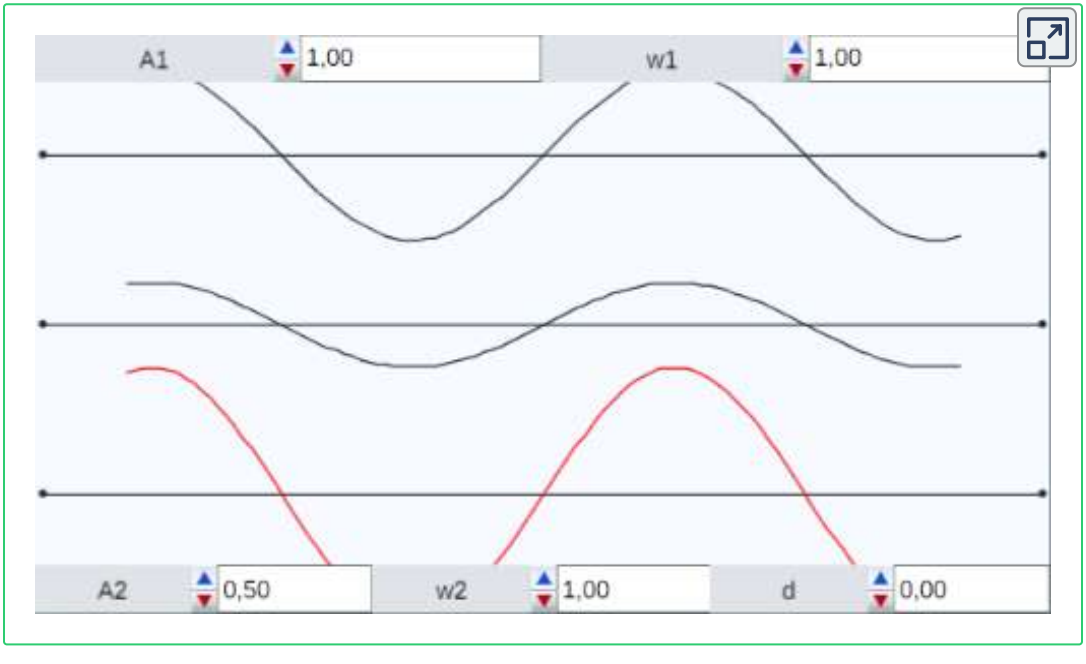
- **parámetro:** Es un campo de texto donde se introduce el nombre de la variable de la cual depende la curva. La variable por defecto es *t*. Es importante diferenciar entre el parámetro de una *familia* (por defecto *s*) y el parámetro de la curva, ya que pueden prestarse a confusión.
- **visible:** Es un checkbox que permite visualizar en el interactivo la expresión de la curva graficada si se encuentra marcado. La expresión se muestra en una barra en la parte inferior del interactivo.
- **editable:** Es un checkbox que funciona únicamente si *visible* está marcado. Permite que la curva sea modificada dinámicamente, de modo que los cambios se reflejen directamente en el gráfico.

Mostramos la escena de la curva presentada en la introducción del apartado.

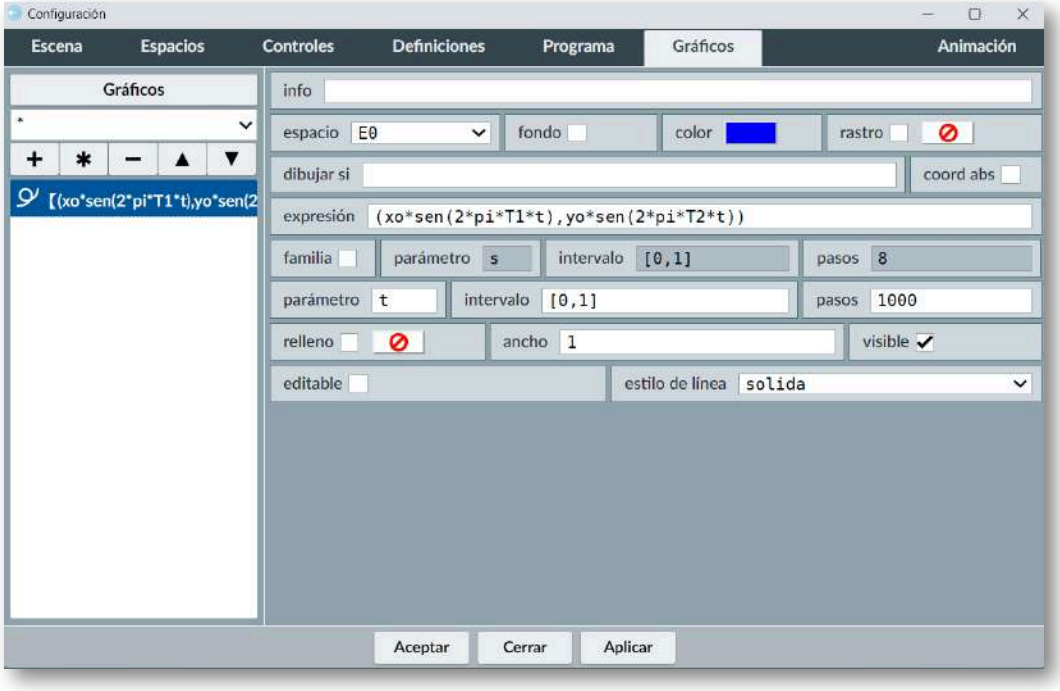


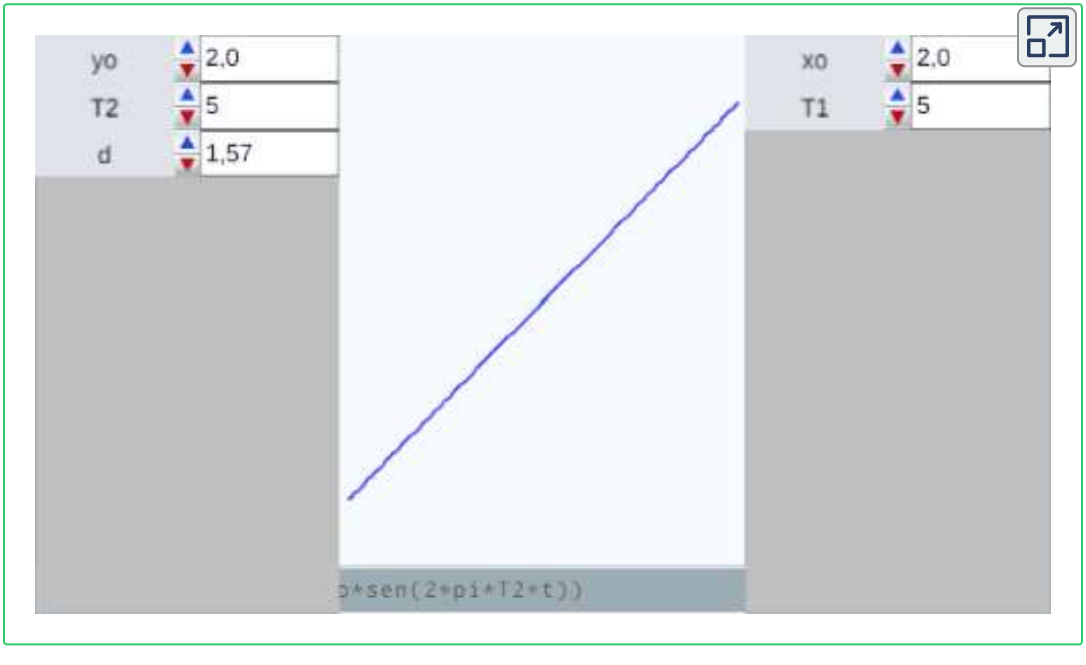
El siguiente ejemplo contiene varios gráficos tipo curva que además incluyen parámetros controlables mediante controles.





Presentamos otro ejemplo con gráficos tipo curva que mediante controles permite obtener formas geométricas llamativas.

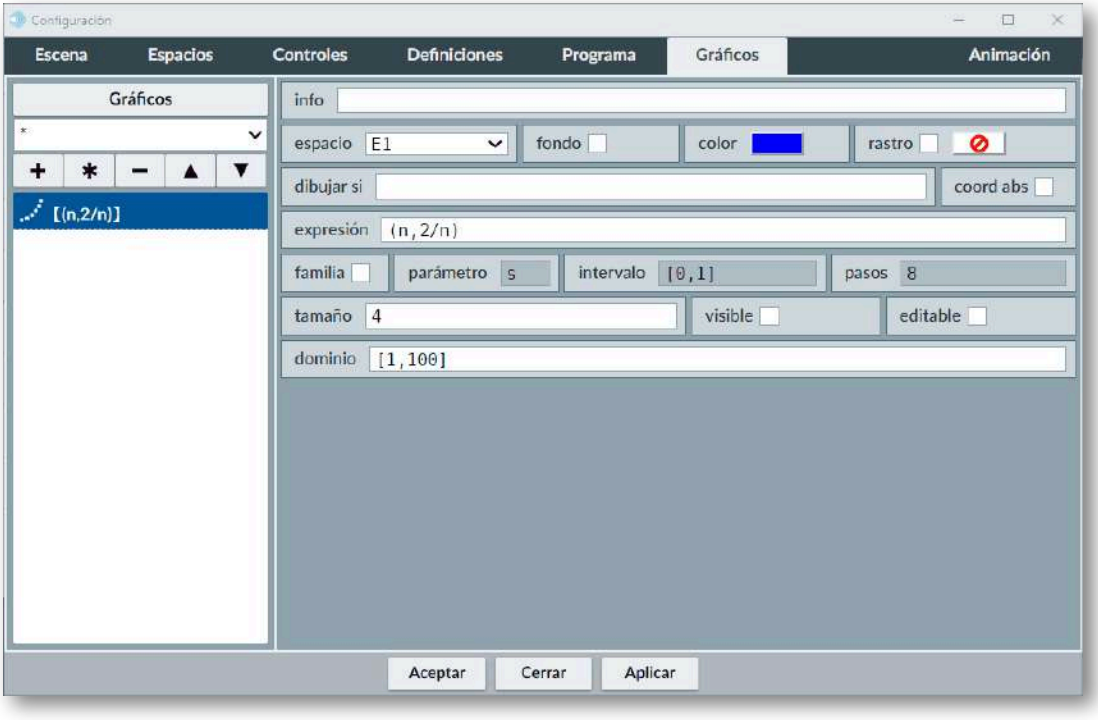




5.6 Gráfico Sucesión

Este gráfico consiste en una serie de puntos que representan una sucesión matemática. En Descartes, una sucesión requiere un parámetro entero que toma valores en los números naturales (0,1,2,3...). A partir de este parámetro se calcula la expresión para las coordenadas de las abscisas y las ordenadas, manejadas como pares ordenados.

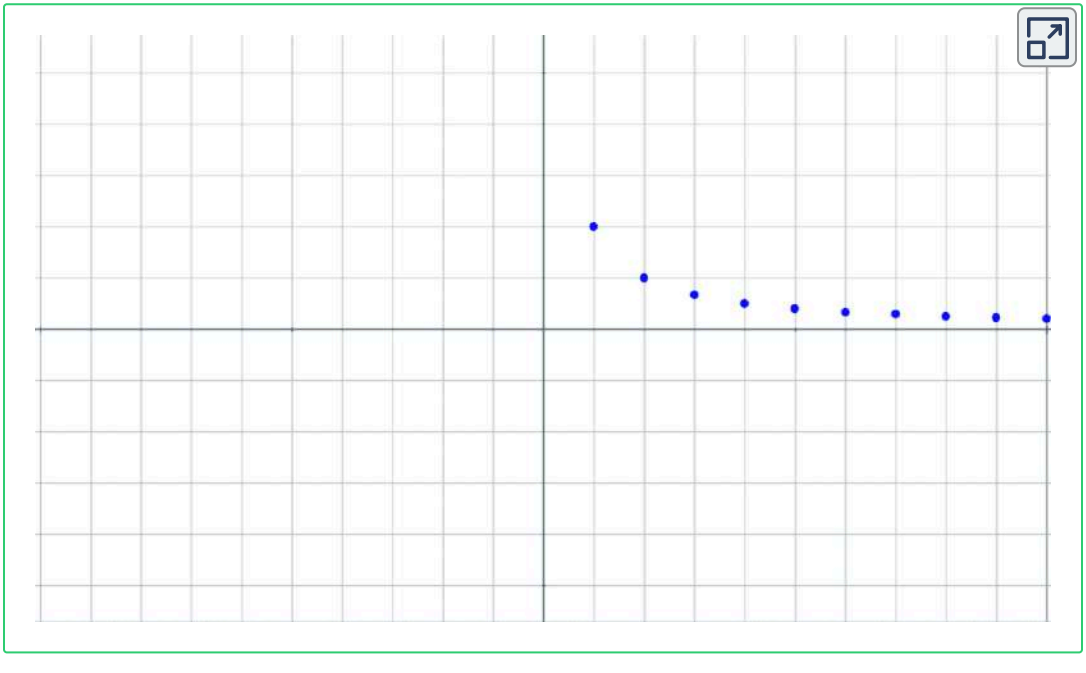
Una sucesión es una familia de puntos con parámetro entero, lo que permite la representación gráfica de funciones de manera similar a una curva, pero usando un parámetro n que solo adopta valores enteros. El panel de configuración específico del tipo sucesión tiene el siguiente aspecto:



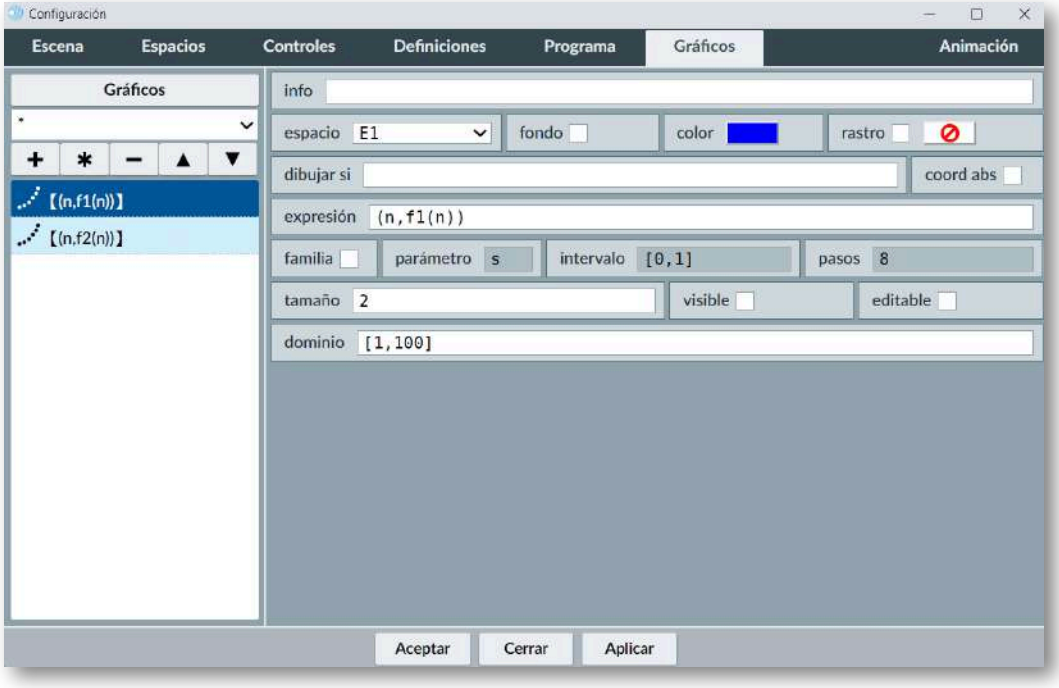
Los parámetros específicos presentes en los gráficos tipo sucesión son los siguientes:

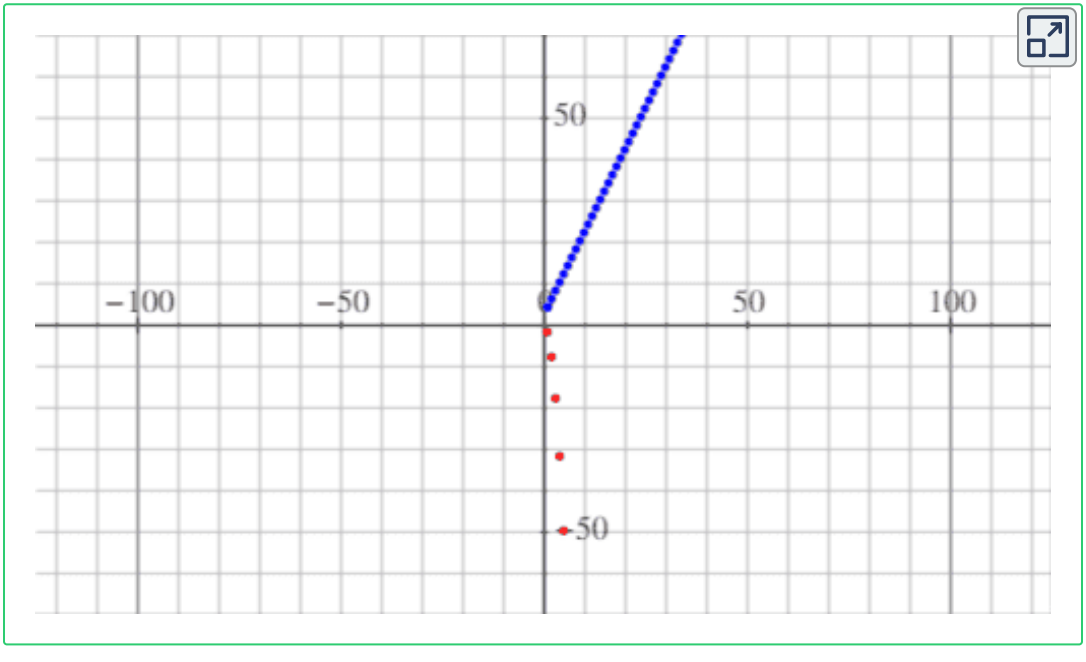
- **expresión:** campo de texto donde se introduce la expresión de la sucesión como par ordenado. Por defecto es (n,1/n), lo que genera puntos como (1,1), (2,0.5), etc. La variable n recorre los números naturales y define las coordenadas de cada punto.
- **visible:** checkbox que, al estar marcado, permite visualizar la regla de correspondencia de la sucesión dentro del interactivo como una barra en la parte inferior.
- **editable:** checkbox que, al estar marcado, permite al usuario modificar directamente en el interactivo la regla de correspondencia, haciendo que los puntos de la sucesión cambien dinámicamente.
- **dominio:** campo de texto donde se define el dominio de la sucesión. Por defecto es [1,100], indicando que n tomará valores enteros desde 1 hasta 100.

Mostramos la escena de la sucesión presentada en la introducción del apartado.

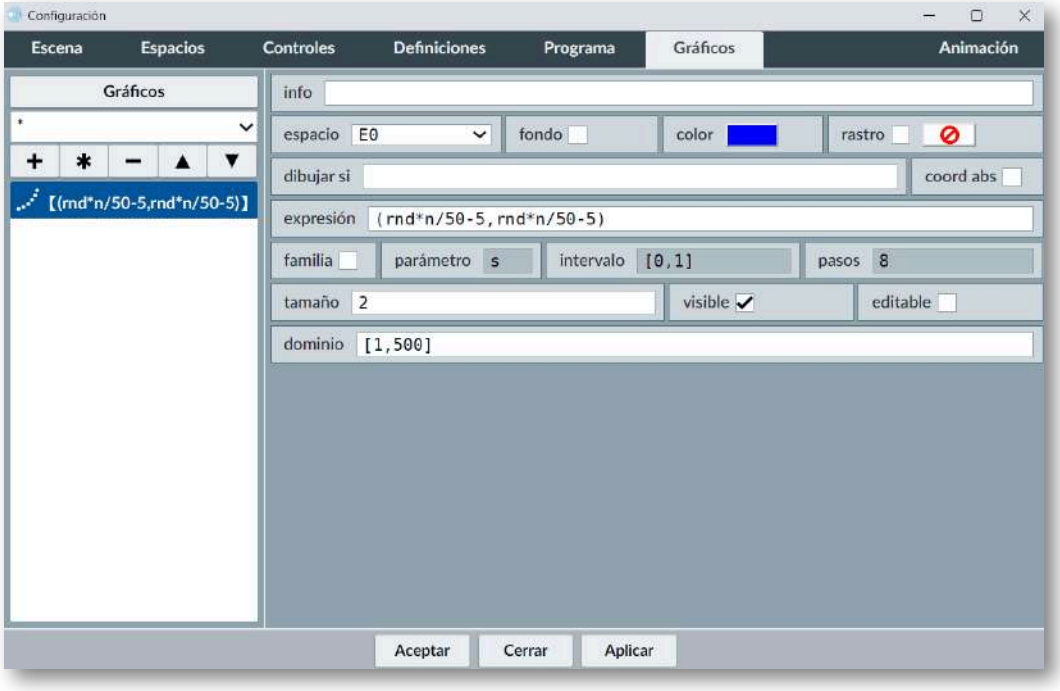


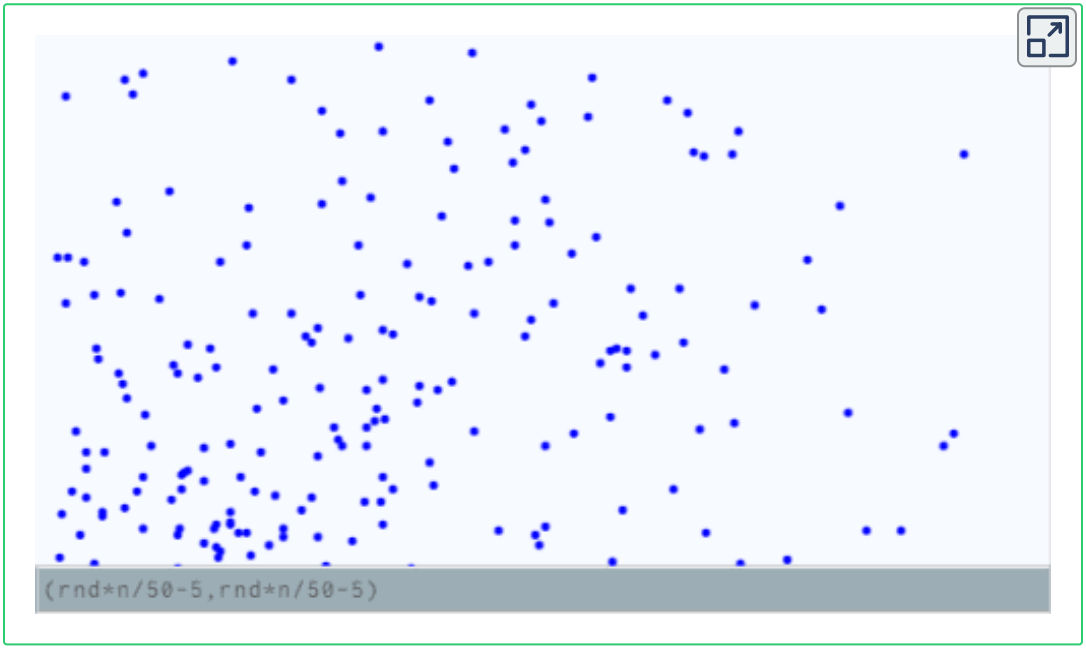
El siguiente ejemplo contiene dos sucesiones definidas mediante las funciones f1(x)= 2*x+2 y f2(x)= -2*x^2.





En este ejemplo la sucesión introduce puntos aleatorios en la escena. Cada vez que esta se actualiza, cambia la disposición de los puntos.

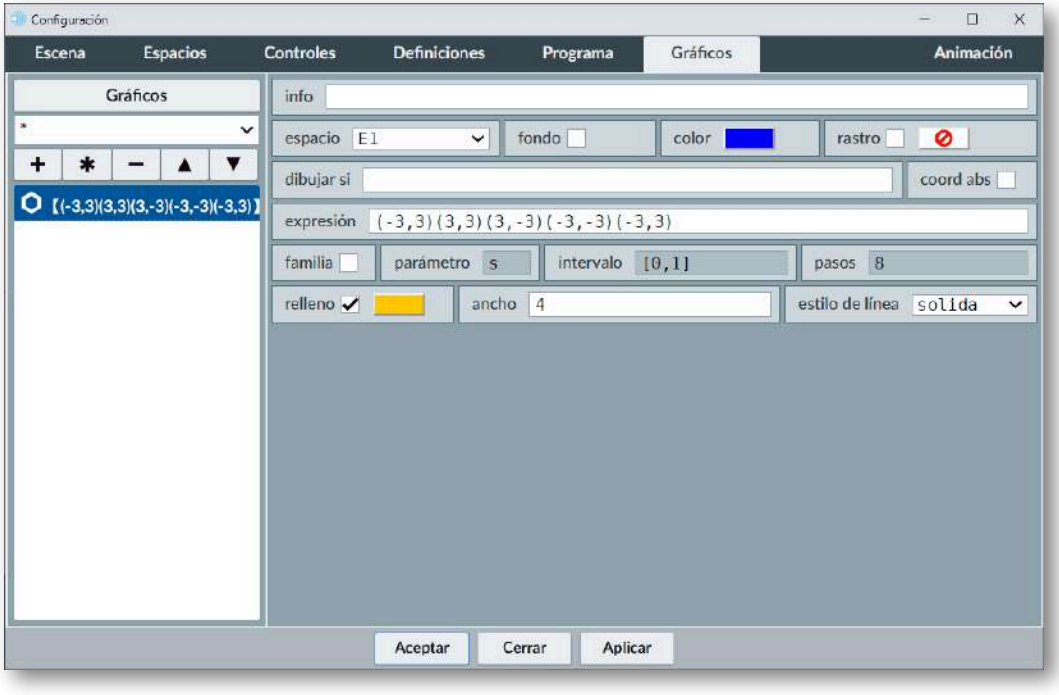




5.7 Gráfico Polígono

Este gráfico se traza a partir de coordenadas de puntos individuales que representan los vértices de un polígono. Las coordenadas de los vértices en secuencia determinan los segmentos que formarán los lados del polígono.

Un polígono es una figura geométrica bidimensional cerrada formada por una serie de lados. El panel de configuración específico del tipo polígono tiene este aspecto:

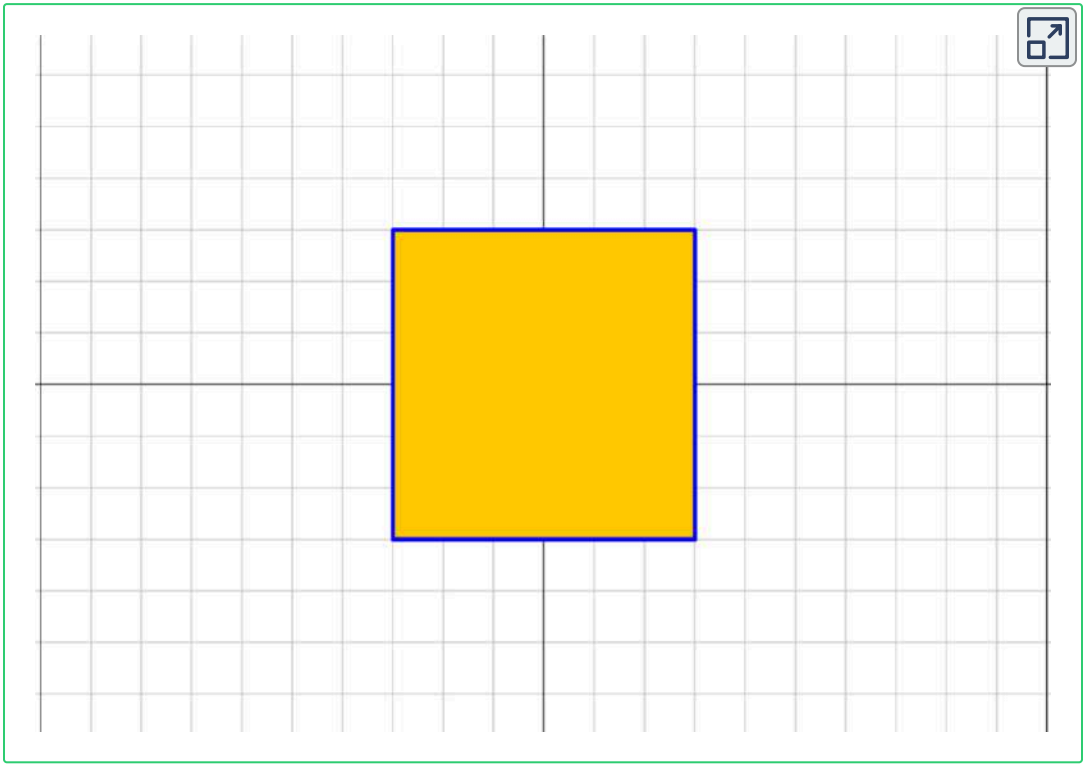


Los parámetros específicos presentes en los gráficos tipo polígono son los siguientes:

- **expresión:** campo de texto donde se introducen las coordenadas de los vértices que componen el polígono. Los lados se trazarán del primer vértice al segundo, del segundo al tercero, y así sucesivamente. Aunque se llame polígono, puede ser una figura abierta si el primer y último vértice no coinciden.

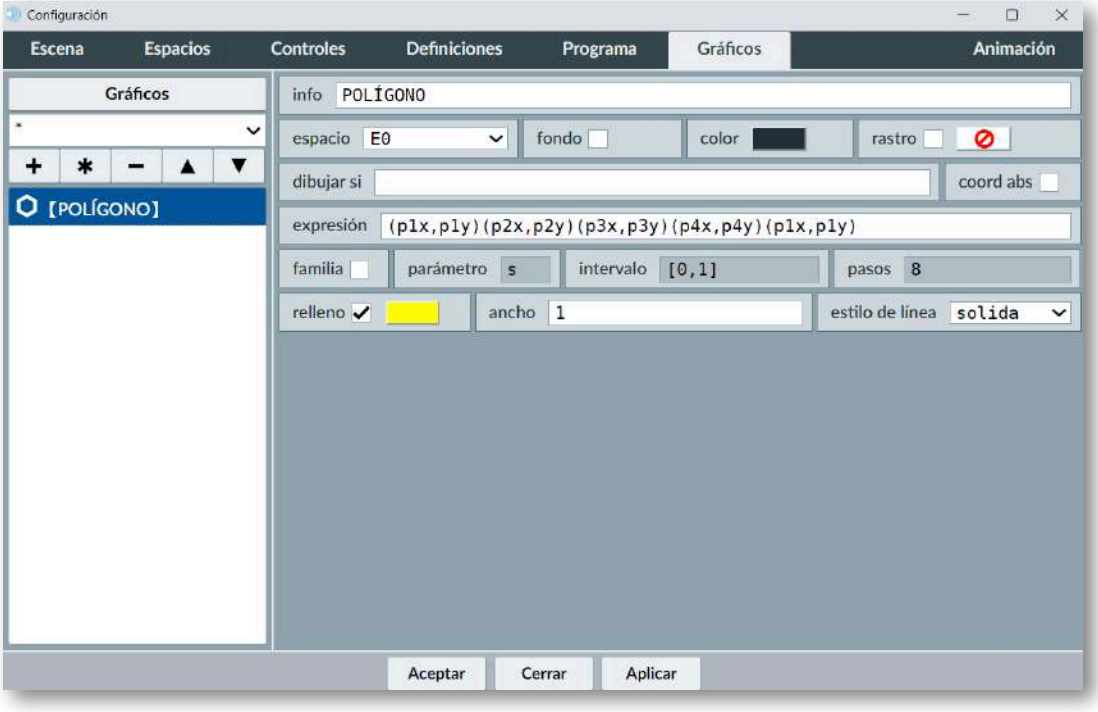
- **relleno:** checkbox que, al marcarse, hace que el interior del polígono tenga un relleno con color. El color se define con el botón de selección de color al lado del checkbox.

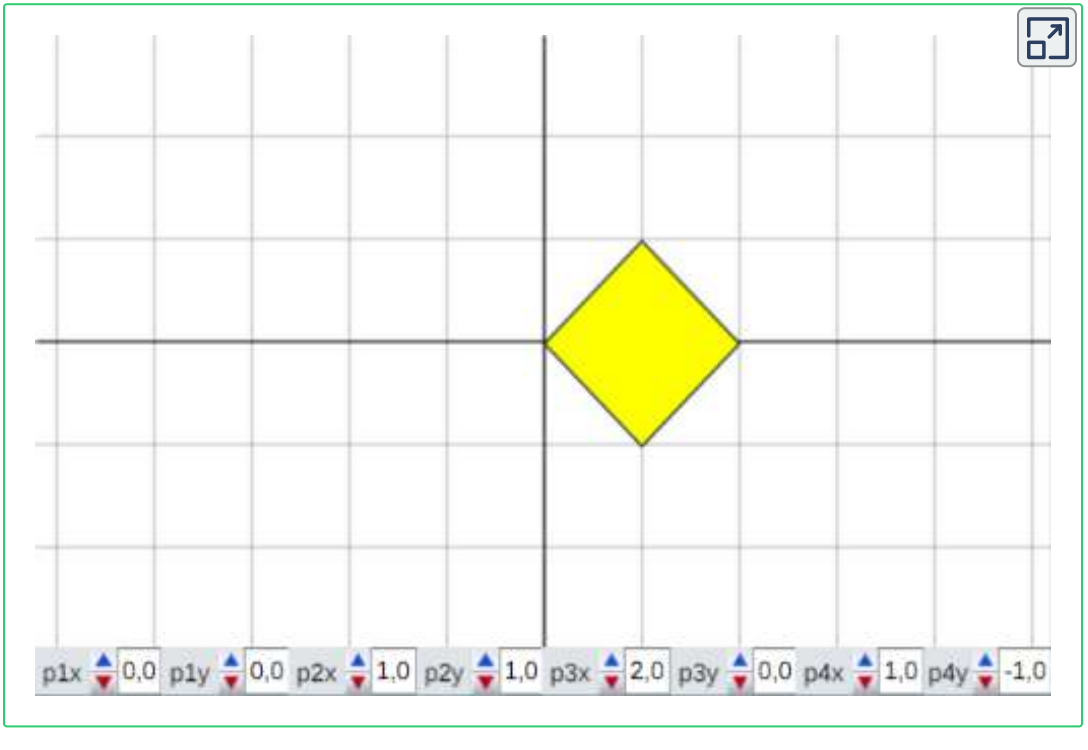
Mostramos la escena del polígono presentado en la introducción del apartado.



A continuación, incluimos un ejemplo de un polígono formado por cuatro puntos. El valor de estos puntos está definido mediante controles tipo pulsador, por lo que a través de ellos se puede modificar de forma sencilla la forma, el tamaño y la posición del polígono resultante.

Se ha utilizado color de relleno amarillo, color de borde negro, grosor 1 y estilo de línea sólida. En el campo *info* se ha añadido el comentario interno "POLÍGONO".

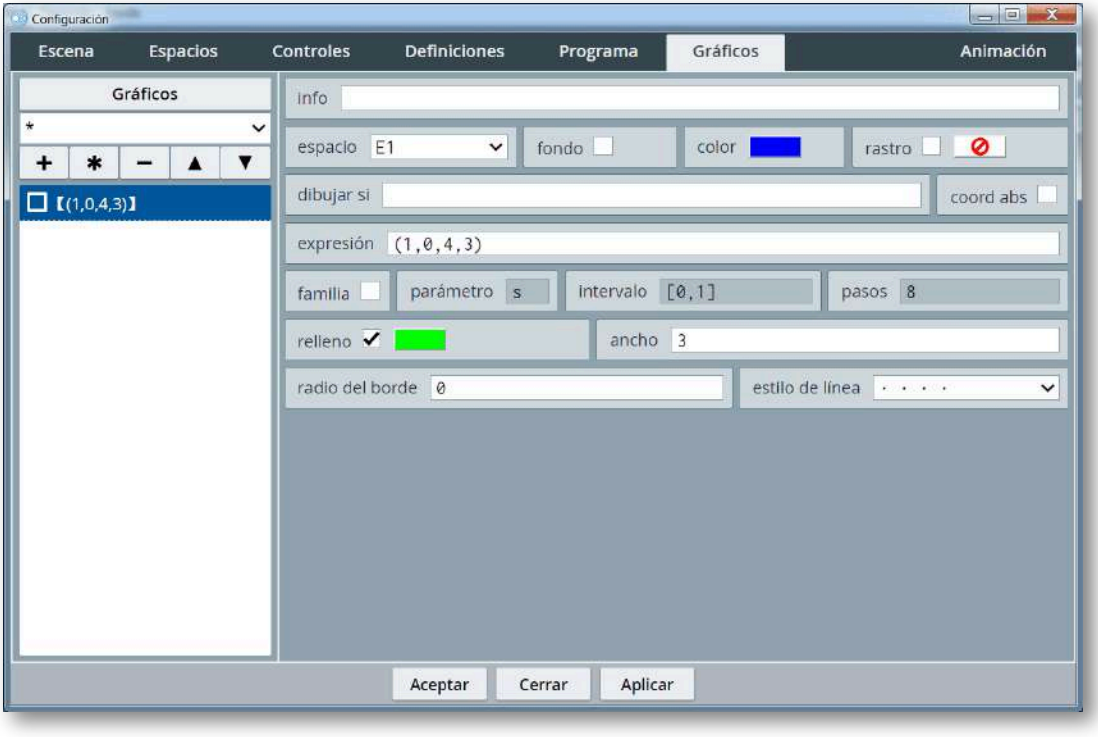

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
[POLÍGONO]
info POLÍGONO
espacio E0
fondo
color
rastro
dibujar si
coord abs
expresión (p1x,p1y)(p2x,p2y)(p3x,p3y)(p4x,p4y)(p1x,p1y)
familia
parámetro s
intervalo [0,1]
pasos 8
relleno
ancho 1
estilo de línea solida
Aceptar
Cerrar
Aplicar


p1x 0,0
p1y 0,0
p2x 1,0
p2y 1,0
p3x 2,0
p3y 0,0
p4x 1,0
p4y -1,0

5.8 Gráfico Rectángulo

Este gráfico constituye una forma abreviada y sencilla de trazar un rectángulo o un cuadrado sin necesidad de utilizar el gráfico *Polígono* para ello.

Un rectángulo es un tipo especial de polígono con cuatro lados, paralelos dos a dos. El panel de configuración específico del tipo rectángulo presenta el siguiente aspecto:



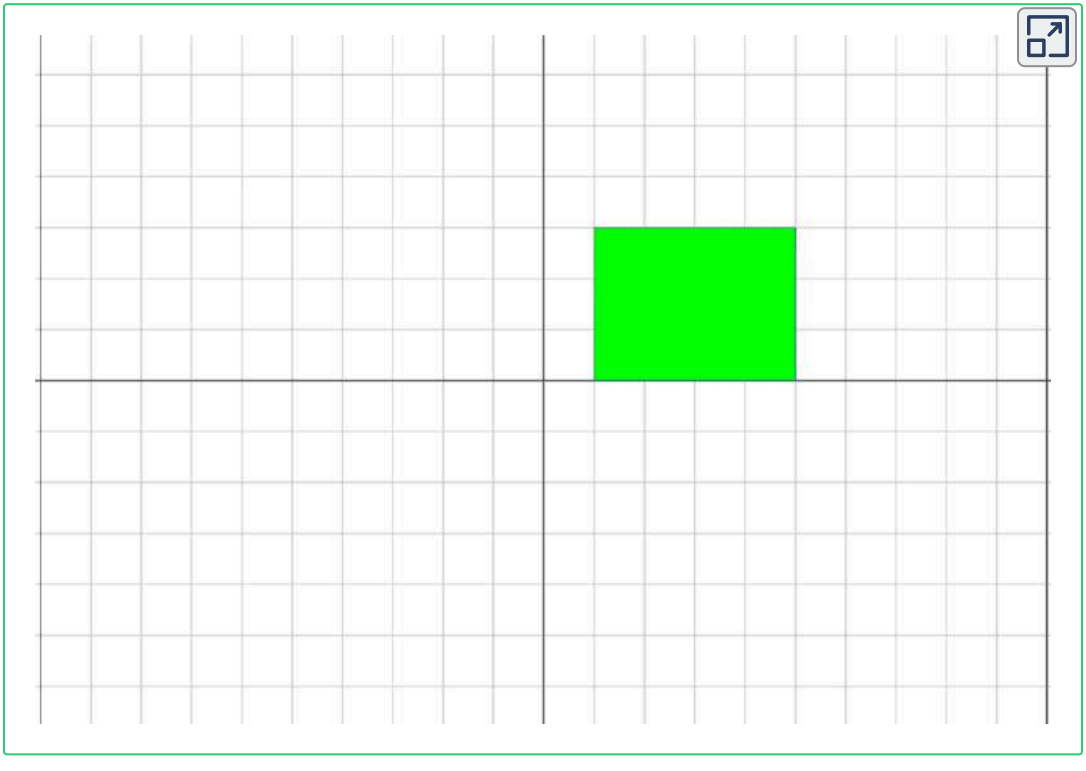
Los parámetros específicos presentes en los gráficos tipo rectángulo son los siguientes:

- **expresión:** campo de texto en el que se introducen cuatro valores entre paréntesis. Los dos primeros corresponden a un vértice del rectángulo. El tercero indica las unidades (o píxeles) hacia la derecha donde se ubicará el vértice opuesto al primero.

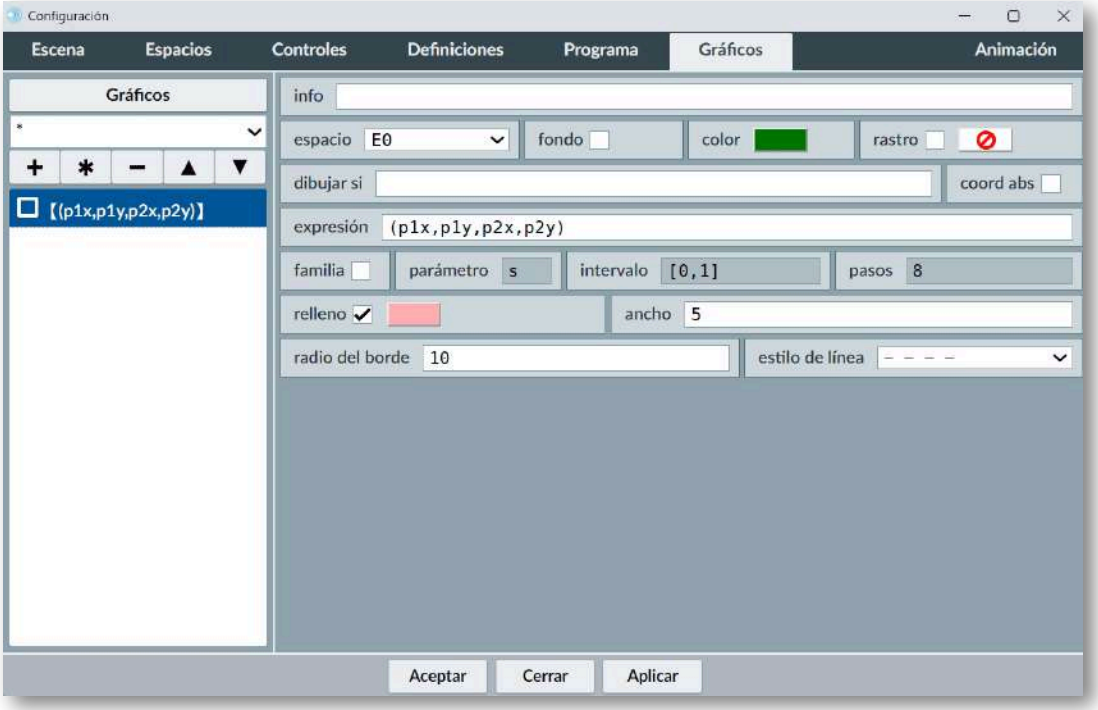
El cuarto indica las unidades (o píxeles) hacia arriba —o hacia abajo si se usan coordenadas absolutas— donde se situará dicho vértice opuesto. Es posible utilizar valores negativos en estas dos últimas entradas.

- **radio del borde:** campo de texto donde se introduce el número de píxeles correspondiente al radio del redondeo de las esquinas. Cuanto mayor sea el valor, mayor será la porción redondeada del rectángulo.

Mostramos la escena del rectángulo presentado en la introducción del apartado.



A continuación, incluimos un ejemplo de un rectángulo. El valor del origen, su anchura y su altura están definidos mediante controles tipo pulsador, por lo que a través de ellos se puede modificar de forma sencilla la forma, el tamaño y la posición del rectángulo resultante.

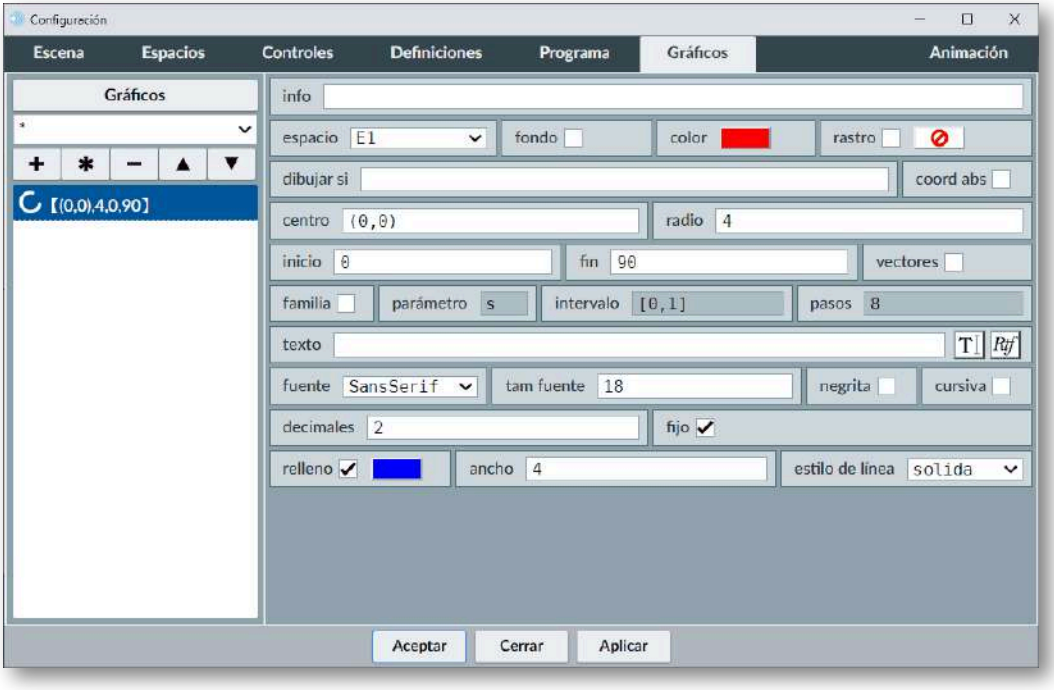

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
【(p1x,p1y,p2x,p2y)】
info
espacio E0
fondo
color
rastro
dibujar si
coord abs
expresión (p1x,p1y,p2x,p2y)
familia
parámetro s
intervalo [0,1]
pasos 8
relleno
ancho 5
radio del borde 10
estilo de línea
Aceptar
Cerrar
Aplicar


p1x 0,0
p1y 0,0
ancho 3,0
alto 2,0

5.9 Gráfico Arco

Este gráfico consiste en una parte de una circunferencia. Aunque es posible obtener el mismo resultado mediante un gráfico de tipo ecuación, este elemento permite una introducción más cómoda, ya que basta con indicar el centro, el radio y el ángulo que debe barrer el arco.

Un arco es una porción de una circunferencia. Un arco de 360º sexagesimales corresponde a una circunferencia completa. En este caso se trabaja con grados sexagesimales. El panel de configuración específico del tipo arco presenta el siguiente aspecto:

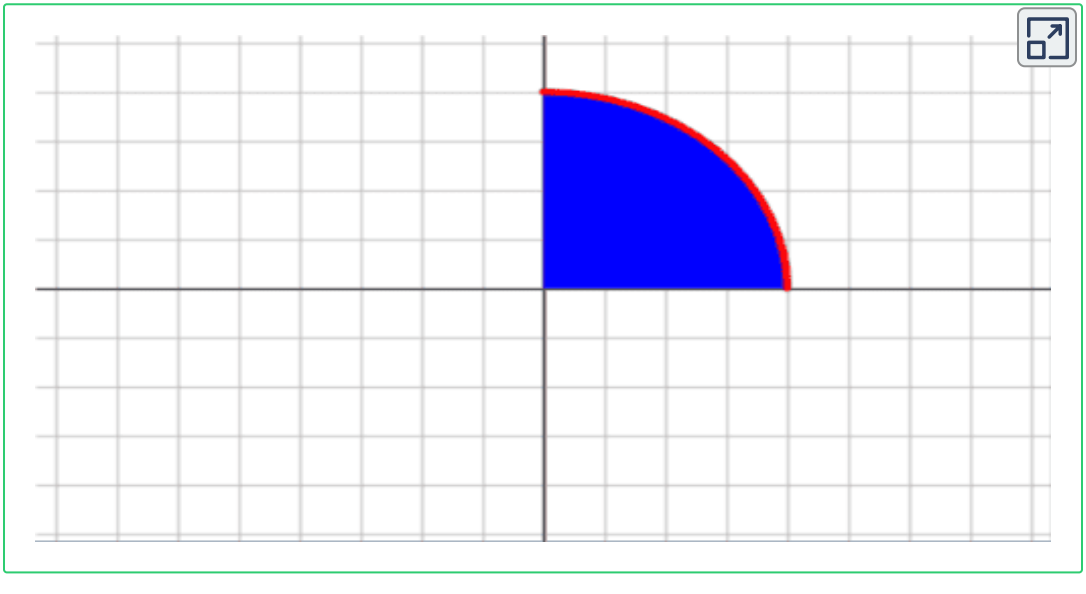


Los parámetros específicos presentes en los gráficos tipo arco son los siguientes:

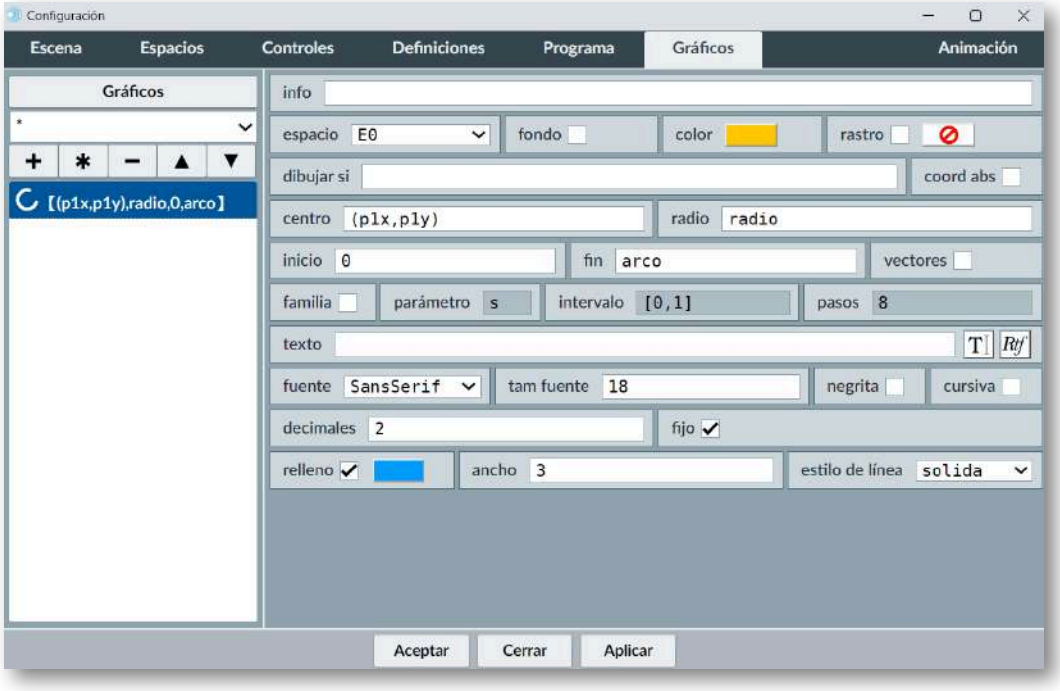
- **centro:** campo de texto donde se indican las coordenadas del centro del arco. Por ejemplo, (3,2) significa que el centro estará 3 unidades a la derecha del origen y 2 unidades hacia arriba.

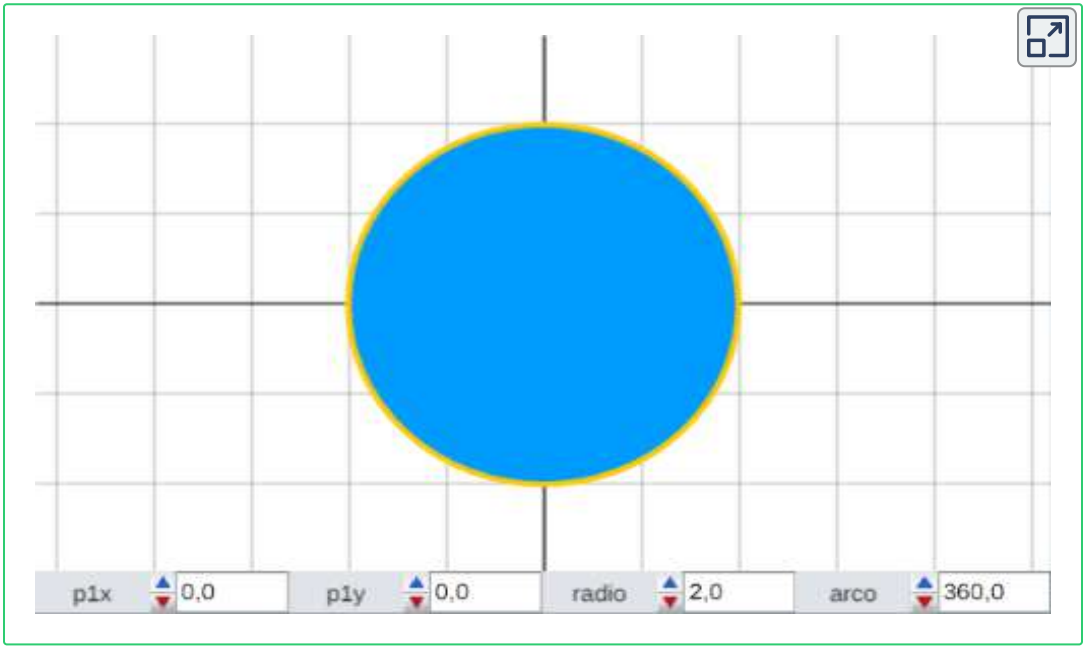
- **radio:** campo donde se especifica el valor del radio del arco.
- **inicio:** campo de texto donde se introduce el ángulo inicial, en grados, a partir del cual comienza a trazarse el arco.
- **fin:** campo de texto donde se introduce el ángulo final, en grados, hasta el cual se trazará el arco.
- **vectores:** checkbox que modifica la forma de definir el inicio y el fin. Si no está marcado, los campos *inicio* y *fin* representan ángulos en grados. Si está marcado, ambos campos reciben coordenadas de vectores que determinan los lados inicial y final del arco. Por ejemplo, puede introducirse (3,2) como vector de inicio. Estos vectores tienen como origen el centro del arco.
- **relleno:** checkbox que determina si el arco estará relleno. Incluye un control de color para seleccionar el color correspondiente. En el caso del arco, el relleno corresponde al área delimitada por los radiovectores que lo definen (dados por los ángulos o por los vectores) y el arco determinado por el radio indicado.

Vemos la escena del arco presentado en la introducción del apartado.



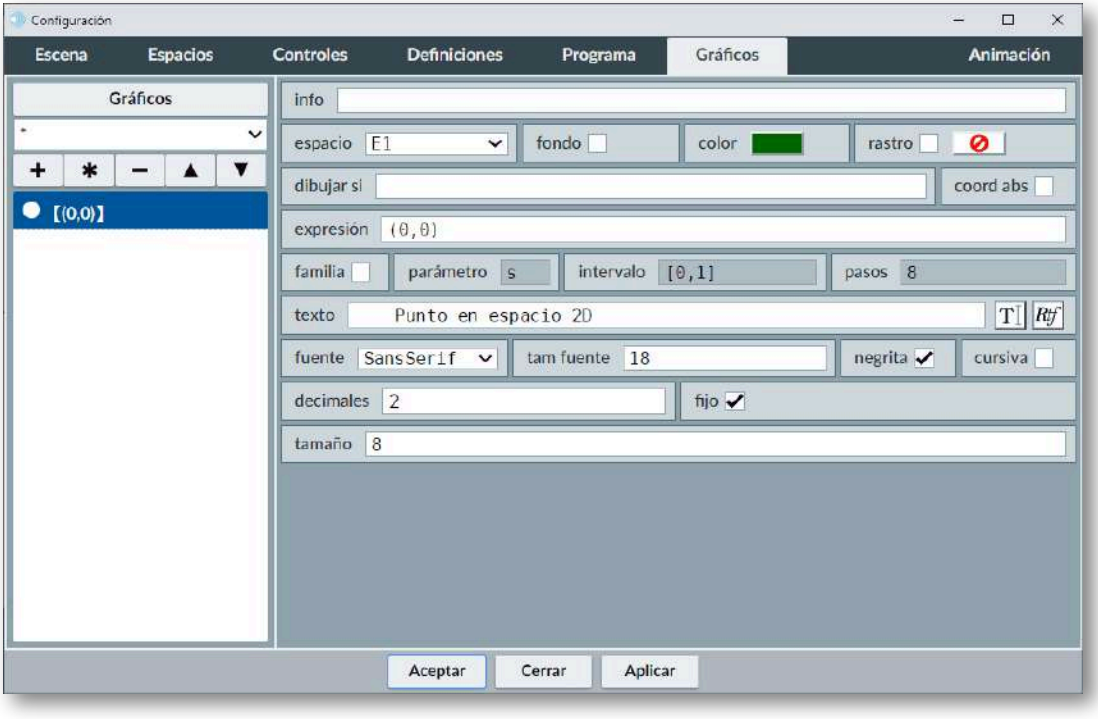
En este ejemplo, el centro, el radio y los grados del arco se definen mediante controles tipo pulsador.





5.10 Gráfico Punto

Este gráfico consiste en un único punto cuyas coordenadas se dan de forma explícita. Representa un punto en el espacio en el que se encuentre dentro de la escena. El panel de configuración específico del tipo punto tiene este aspecto:

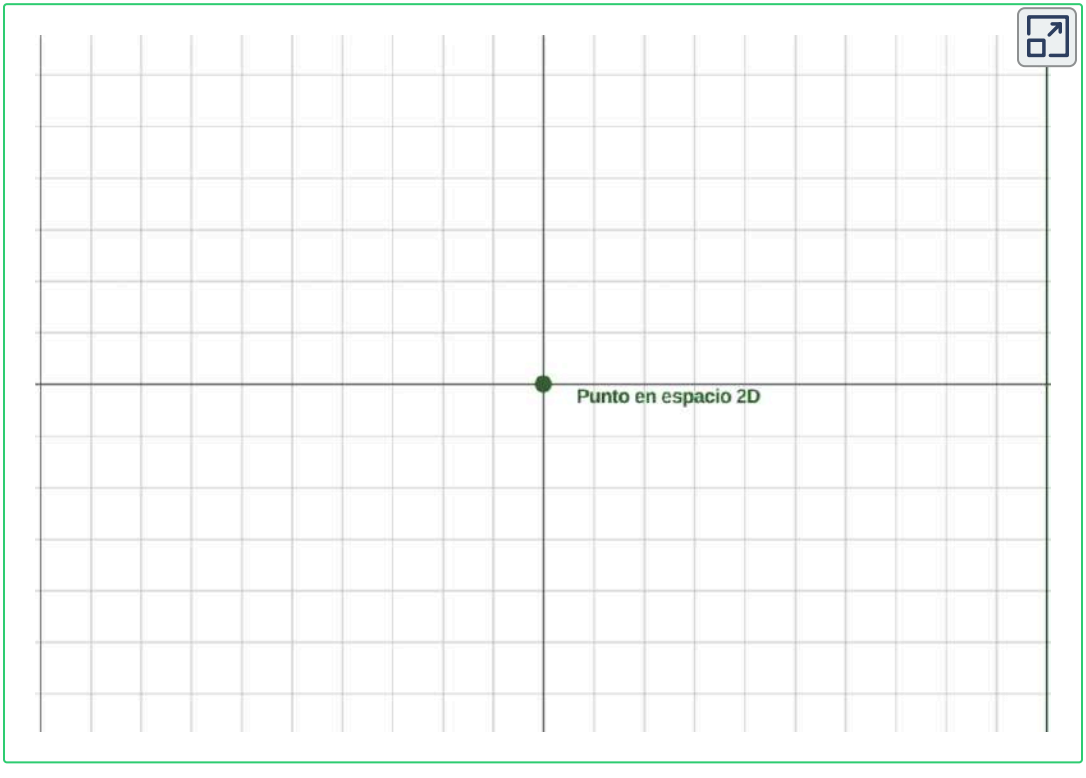


Los parámetros específicos presentes en los gráficos tipo punto son los siguientes:

- **expresión:** es un campo de texto en donde se colocan las coordenadas del único punto que pueden ser absolutas o relativas.
- **decimales:** es un campo de texto donde se introduce un número entero correspondiente al número de decimales que habrán de mostrarse en caso de que el texto contenga alguna variable cuyo valor se quiera visualizar.

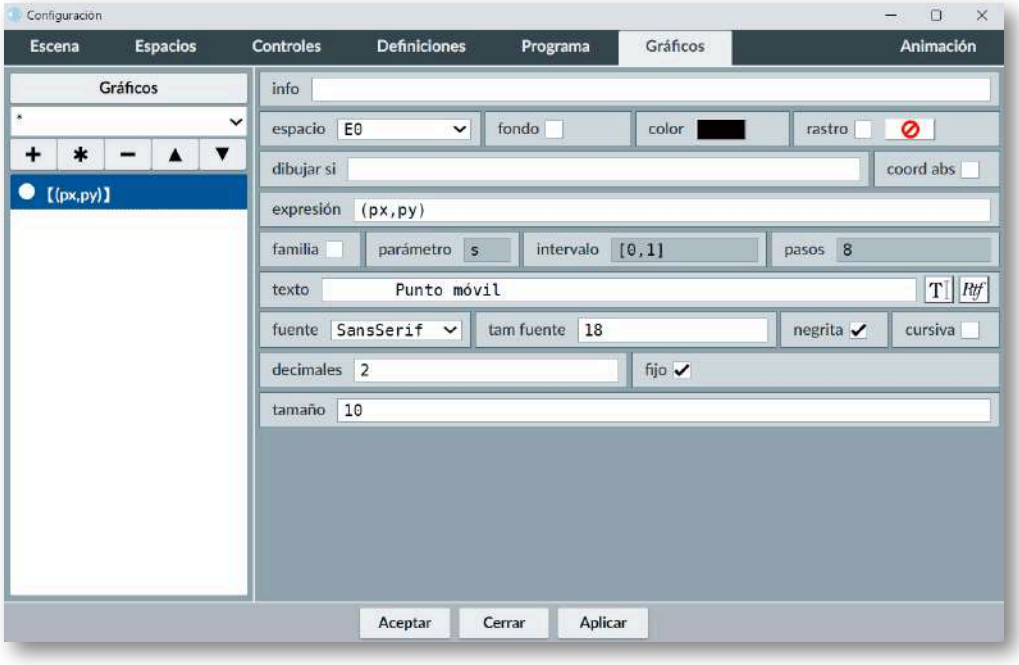
- **fijo:** es un checkbox que determina si siempre se muestran el número de decimales elegido en el control decimales, aun cuando no sean significativos. Siempre se mostrarán los decimales elegidos si este checkbox está marcado; de lo contrario, solo se mostrarán si son significativos.
- **Tamaño:** es el radio del "punto". En realidad, el programa dibuja un disco de radio tamaño. Si tamaño=0 el punto no se dibuja. Esto último puede aprovecharse para dibujar textos asociados a puntos invisibles, usando las coordenadas del espacio.

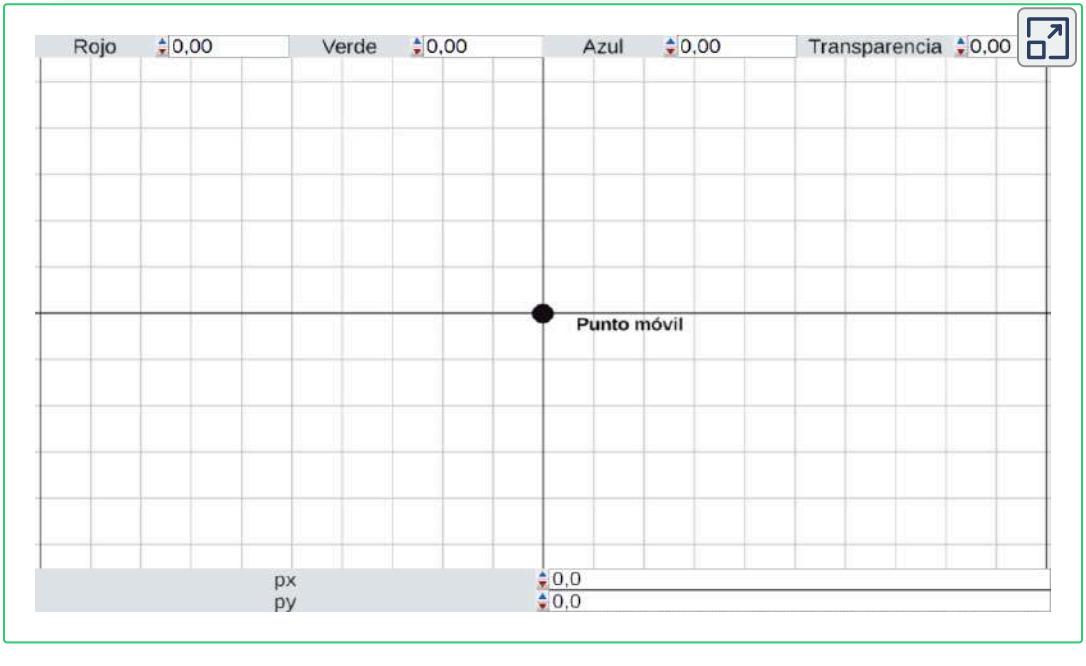
Mostramos la escena del punto presentado en la introducción del apartado.



A continuación, incluimos un ejemplo de un punto cuyas coordenadas están dadas por controles tipo pulsador, lo que permite cambiar la posición del punto dentro del espacio de la escena.

También se ha parametrizado el color y la transparencia del punto, pudiendo introducir mediante pulsadores el grado de transparencia y los valores de color en formato RGB.





En este ejemplo se muestra un punto que cambia de posición con una animación, pudiendo configurarse para que deje rastro o no.

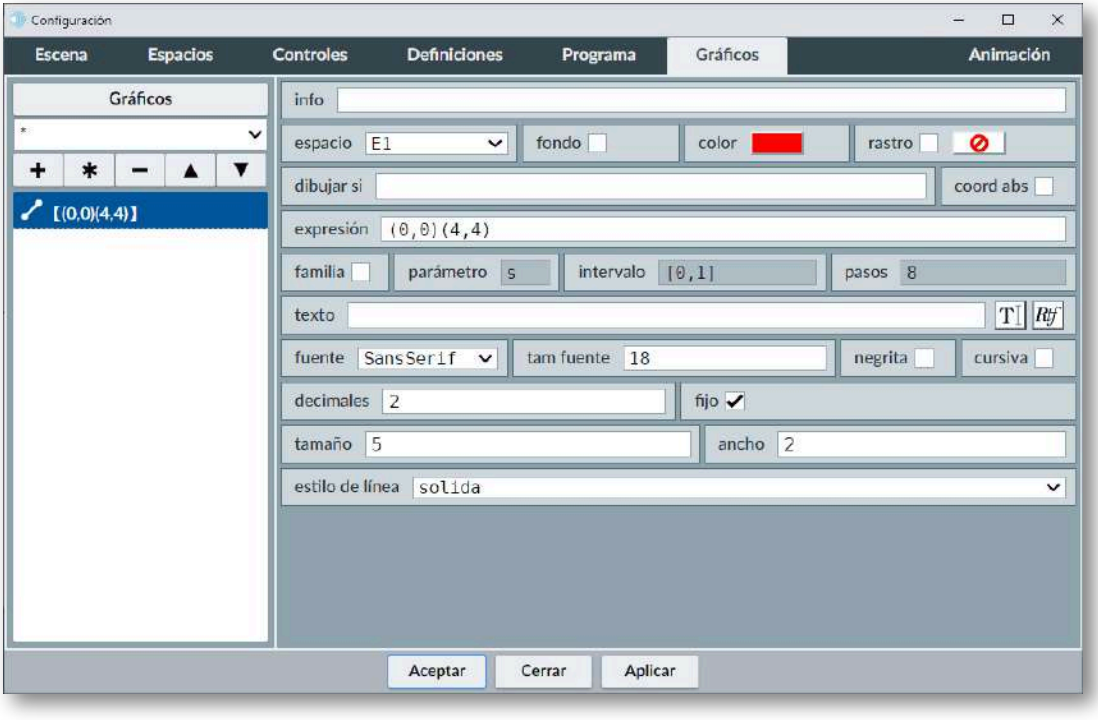




5.11 Gráfico Segmento

Este gráfico consiste en un segmento definido por las coordenadas de sus extremos. Un segmento es una línea recta limitada por dos puntos.

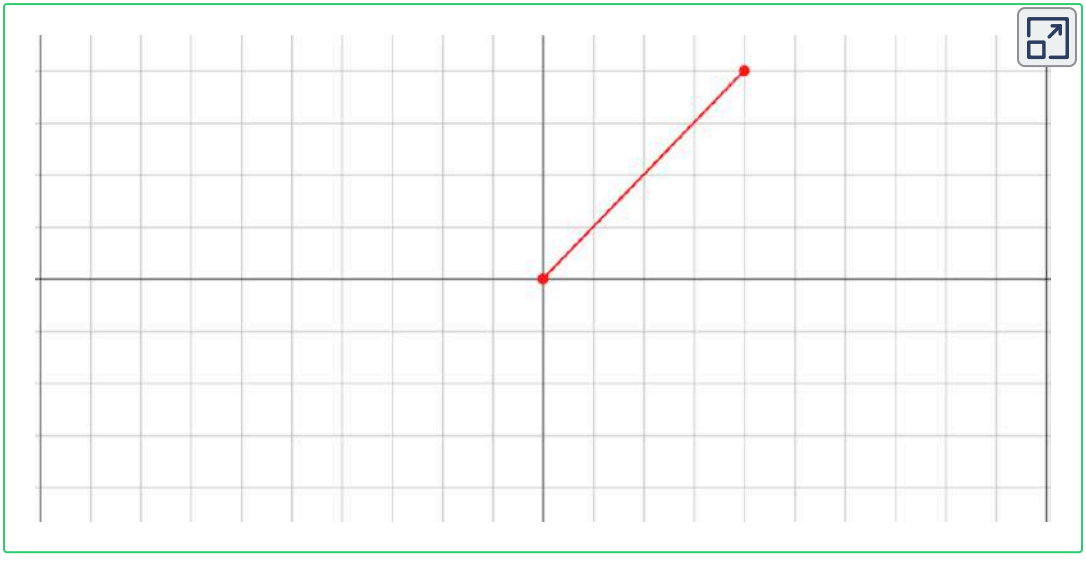
El panel de configuración específico del tipo segmento tiene este aspecto:

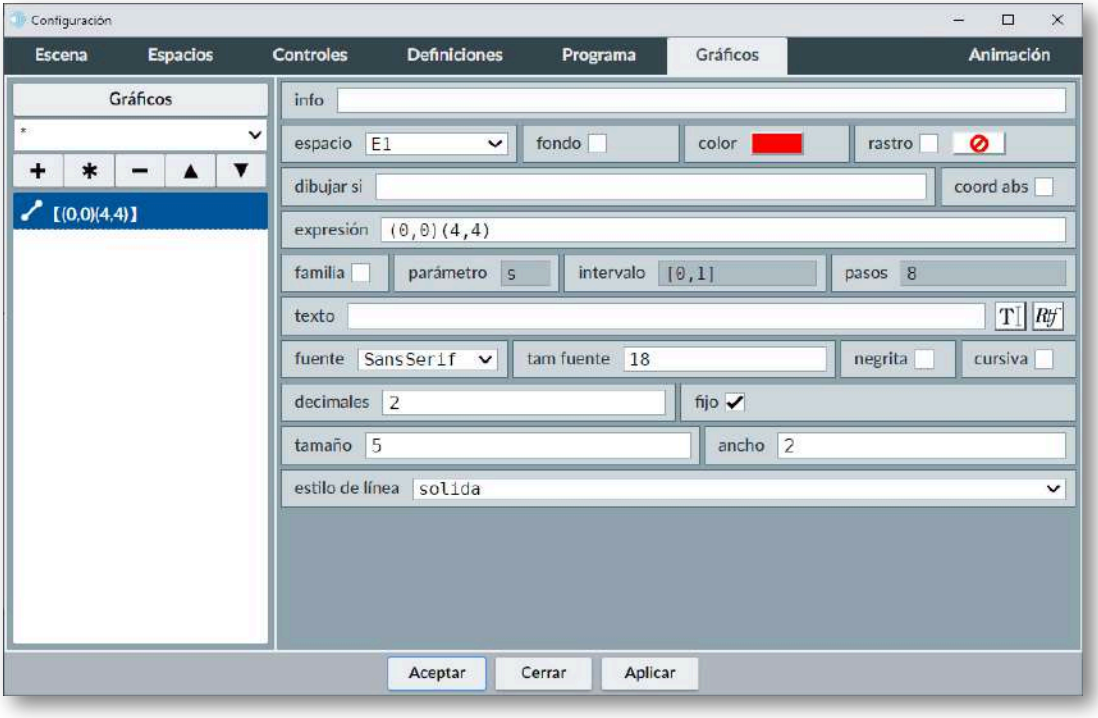


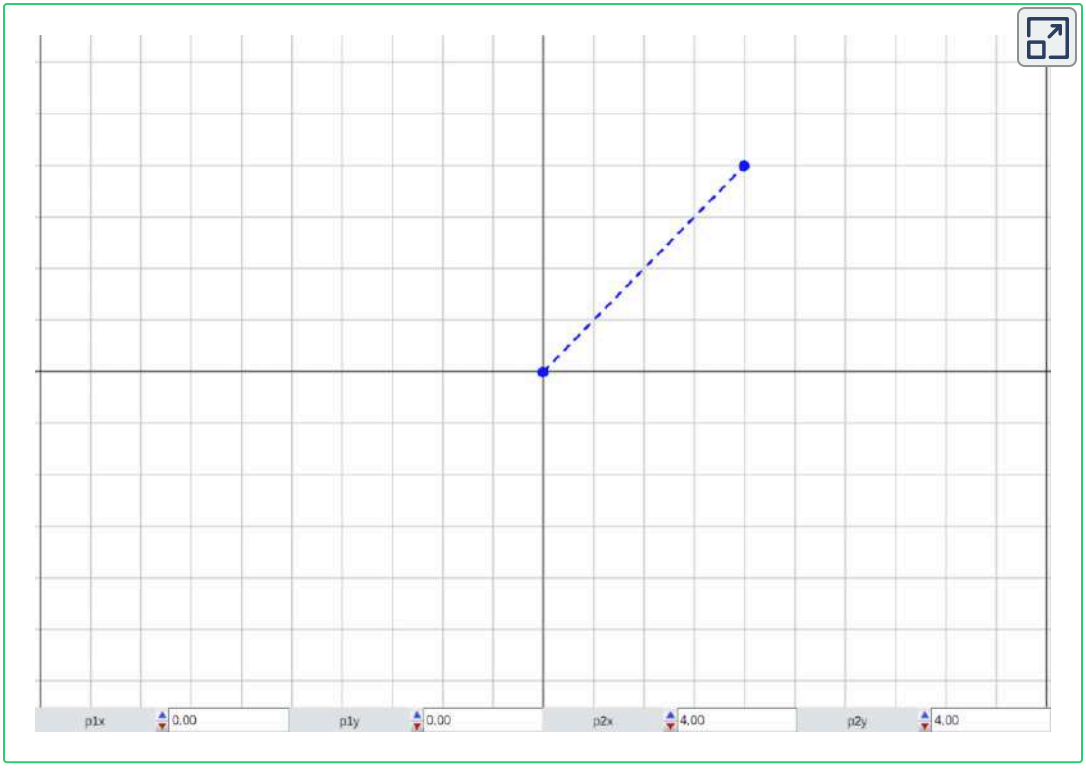
Los parámetros específicos presentes en los gráficos tipo segmento son los siguientes:

- **expresión:** es un campo de texto en el cual se introducen un par de coordenadas absolutas o relativas correspondientes a los extremos del segmento.
- **ancho:** es un campo de texto donde se introduce un número entero correspondiente al ancho en píxeles del segmento.

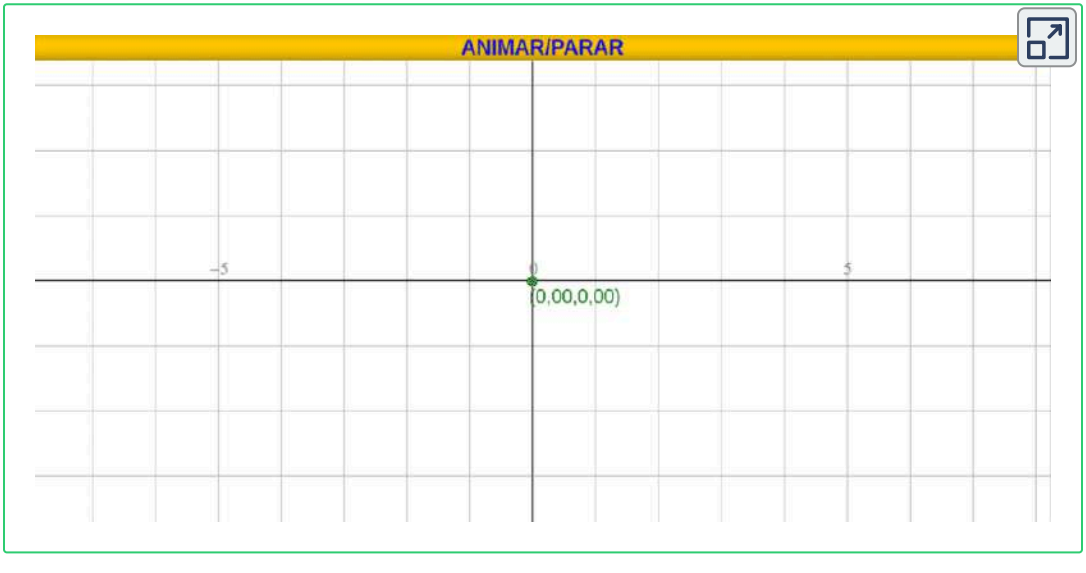
Mostramos la escena del segmento presentado en la introducción del apartado y otra similar, pero con los extremos definidos mediante controles.





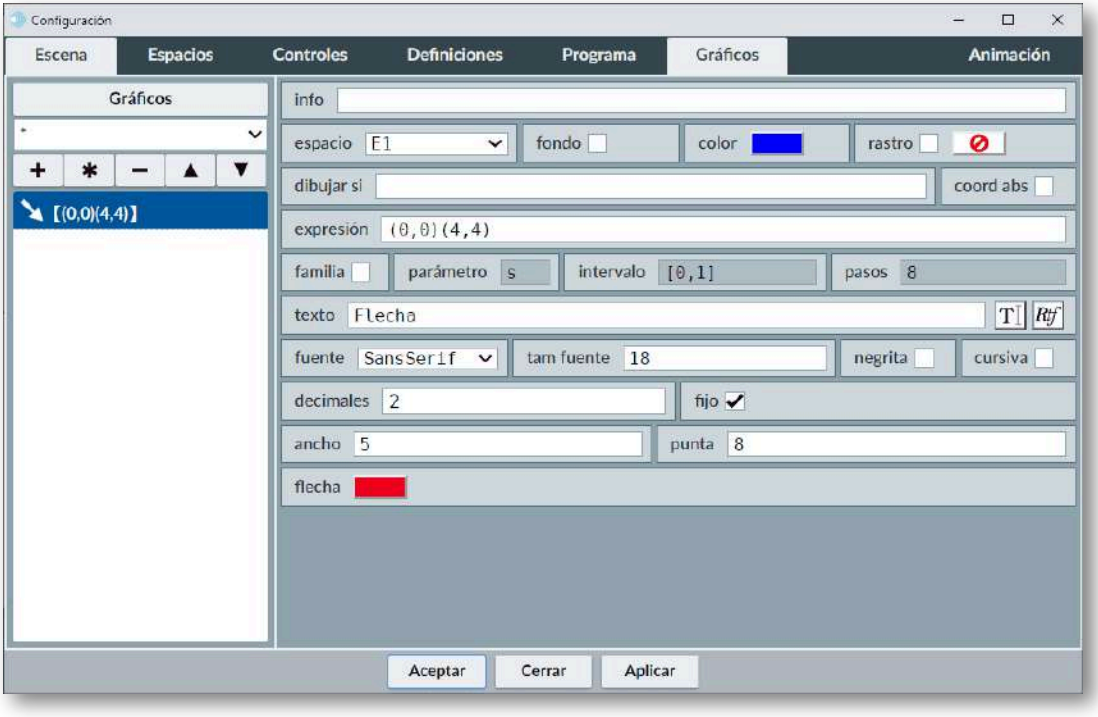


Incluimos otro ejemplo en el que, mediante una animación, uno de los extremos del segmento cambia de posición de forma aleatoria.



5.12 Gráfico Flecha

Este gráfico es muy similar al segmento, con la salvedad de que en la última coordenada indicada en su campo de texto **expresión** se dibuja una punta de flecha. Una flecha es una línea recta limitada por dos extremos, uno de los cuales termina en la punta. El panel de configuración específico del tipo flecha tiene este aspecto:

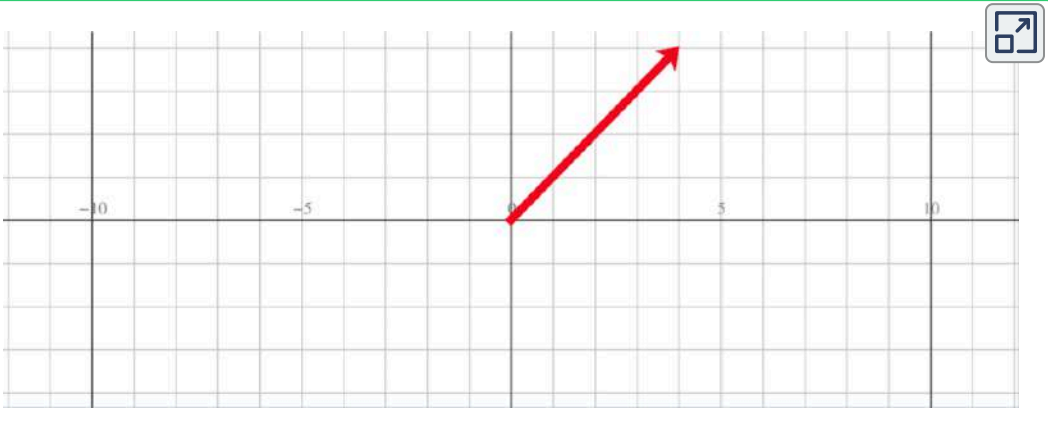


Los parámetros específicos presentes en los gráficos tipo flecha son los siguientes:

- **expresión:** es un campo de texto donde se indica el par de coordenadas de los extremos de la flecha. La última coordenada corresponde al extremo con la punta de la flecha.
- **ancho:** es un campo de texto que determina el grosor en píxeles de la flecha. En versiones actuales existen dos parámetros de ancho; solo el segundo es funcional.

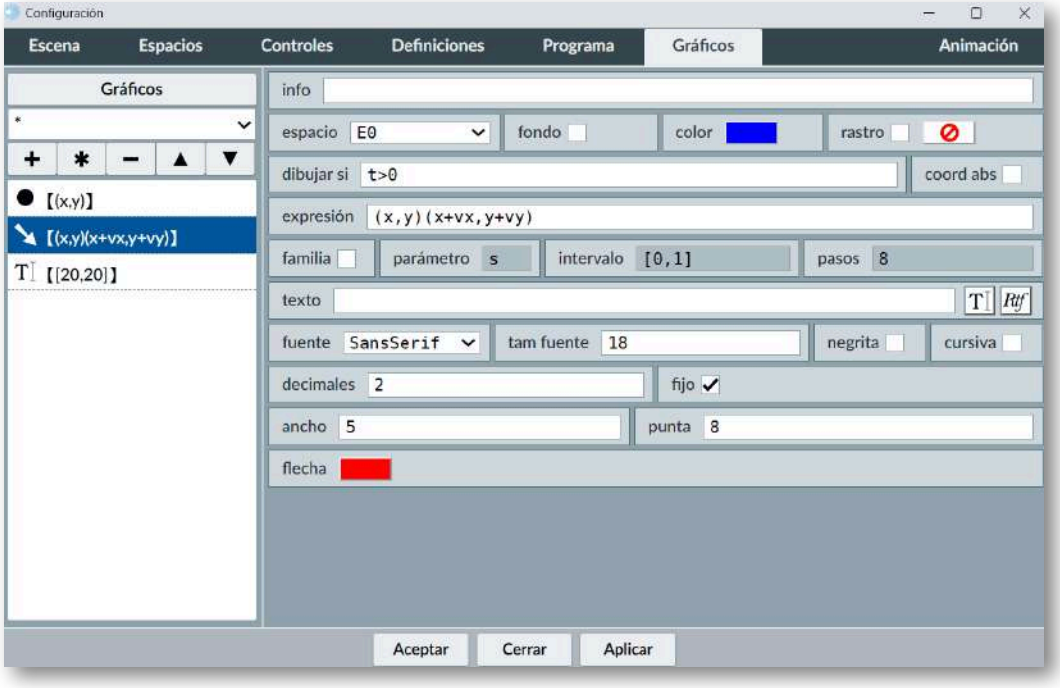
- **punta:** es un campo de texto que define el tamaño de la punta de la flecha en píxeles.
- **flecha:** es un botón que abre la herramienta de control de colores para seleccionar el color de la flecha. Controla el interior de la flecha, mientras que el otro control de color maneja el borde.

Vemos la escena de la flecha presentada en la introducción.



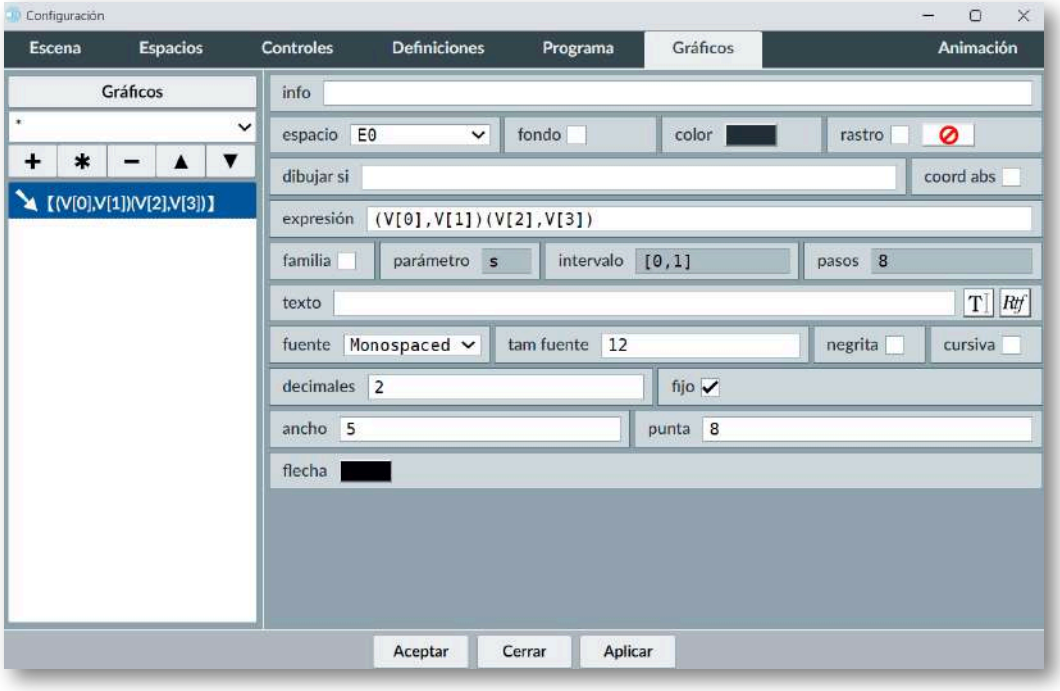


En este ejemplo se muestra un punto y una flecha animada, además de un control que permite configurar un parámetro.





En esta escena se define una flecha a través de los componentes de un vector que a su vez recogen su valor de pulsadores.



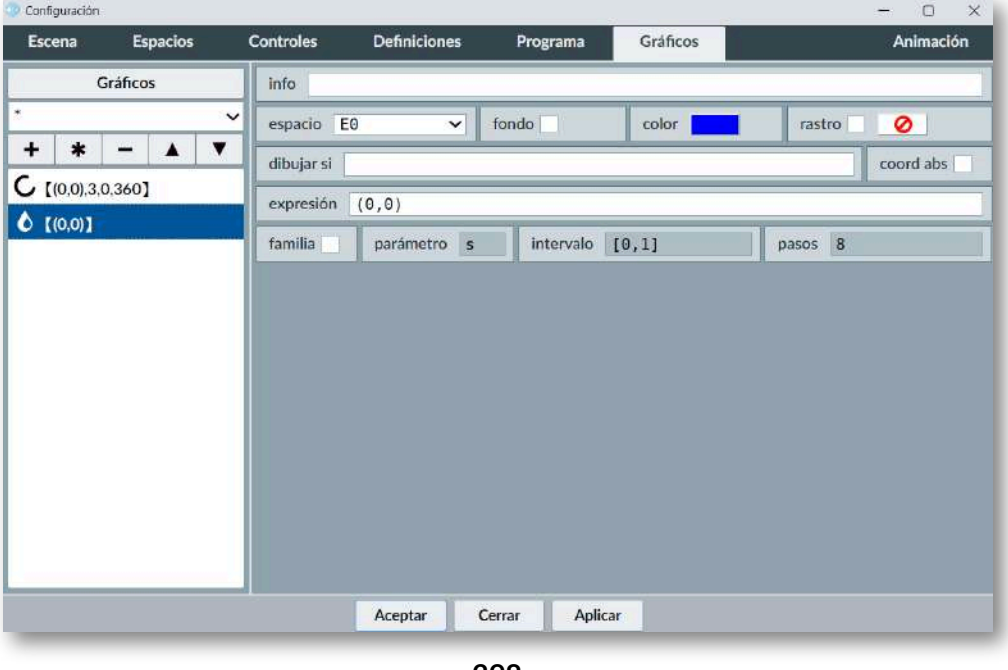


5.13 Gráfico Relleno

Este gráfico consiste en un relleno con un color elegido por el usuario. Los límites de este relleno se determinan por los bordes de otros gráficos.

Un relleno se define únicamente mediante un punto y un color. El punto actúa como semilla o punto inicial de un proceso de coloreado, que llena la región cerrada en la que se encuentra con el color seleccionado. Los gráficos de fondo no funcionan como fronteras para el relleno, a menos que el propio relleno sea de fondo. En otras palabras, el punto del relleno coloreará todo el espacio cerrado alrededor de él.

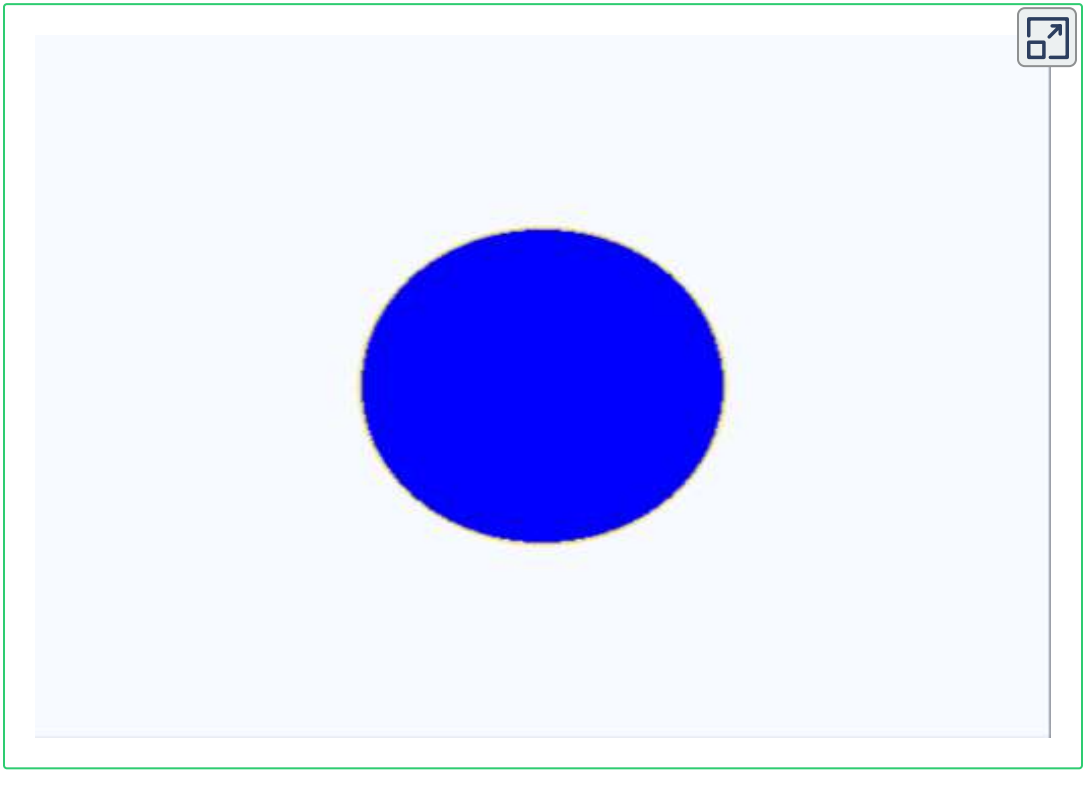
Aunque muchos gráficos tienen su propio relleno, a veces es útil colorear áreas que no están delimitadas por un solo gráfico, sino por varios. En estos casos, conviene utilizar el gráfico de tipo relleno. El panel de configuración específico para este tipo de relleno se ve de la siguiente manera:



El parámetro específico presente en los gráficos tipo relleno es el siguiente:

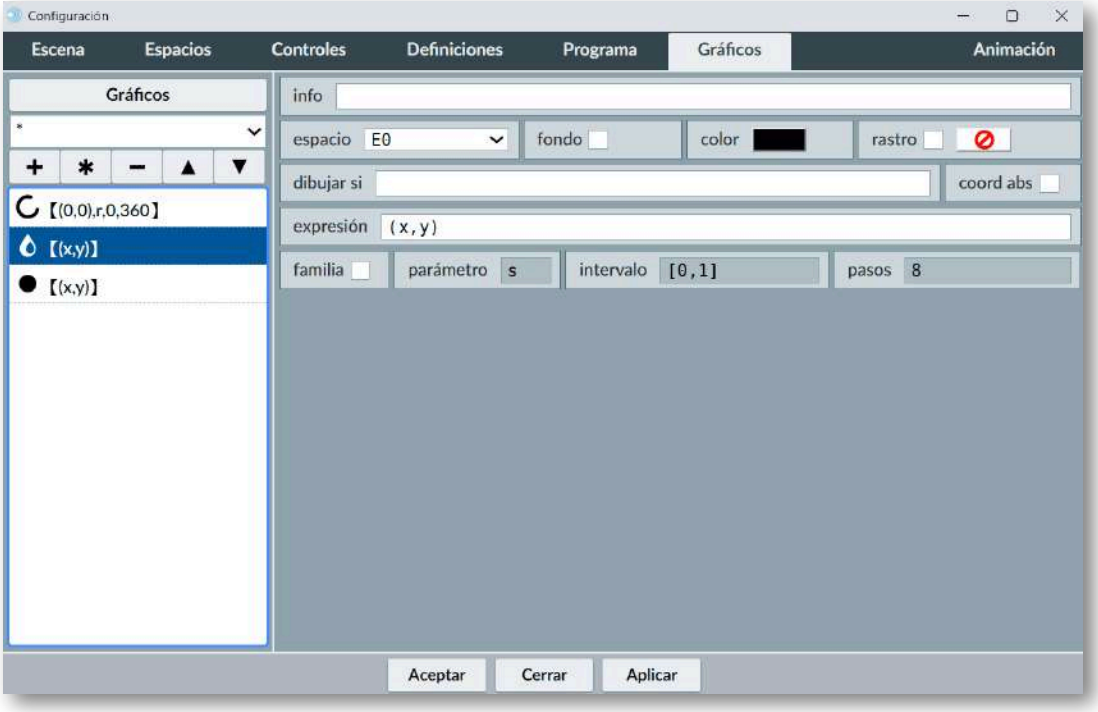
- **Expresión:** es un campo de texto donde se introduce la coordenada de un punto que debe estar dentro del área delimitada que se desea rellenar con color.

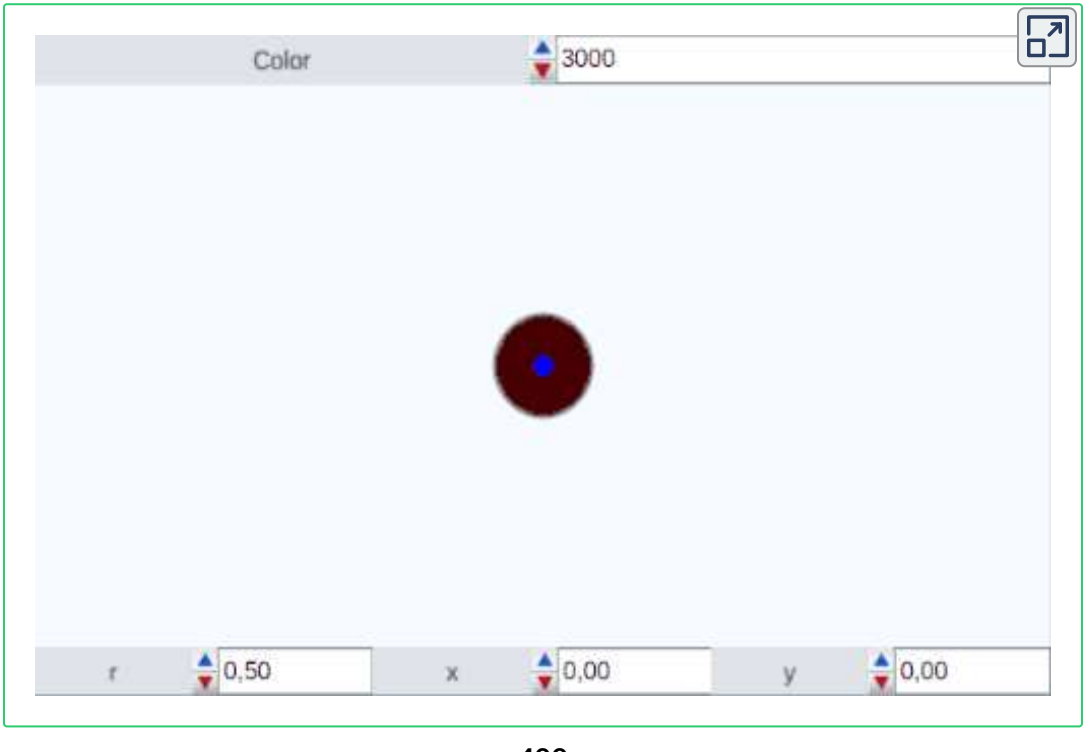
Mostramos la escena del relleno presentado en la introducción del apartado.



En la siguiente escena se muestra un ejemplo de relleno móvil cuyas coordenadas se definen mediante controles tipo pulsador. Además, se ha dibujado un círculo usando un elemento arco.

También se ha colocado un control en la zona norte que permite cambiar el color del relleno.

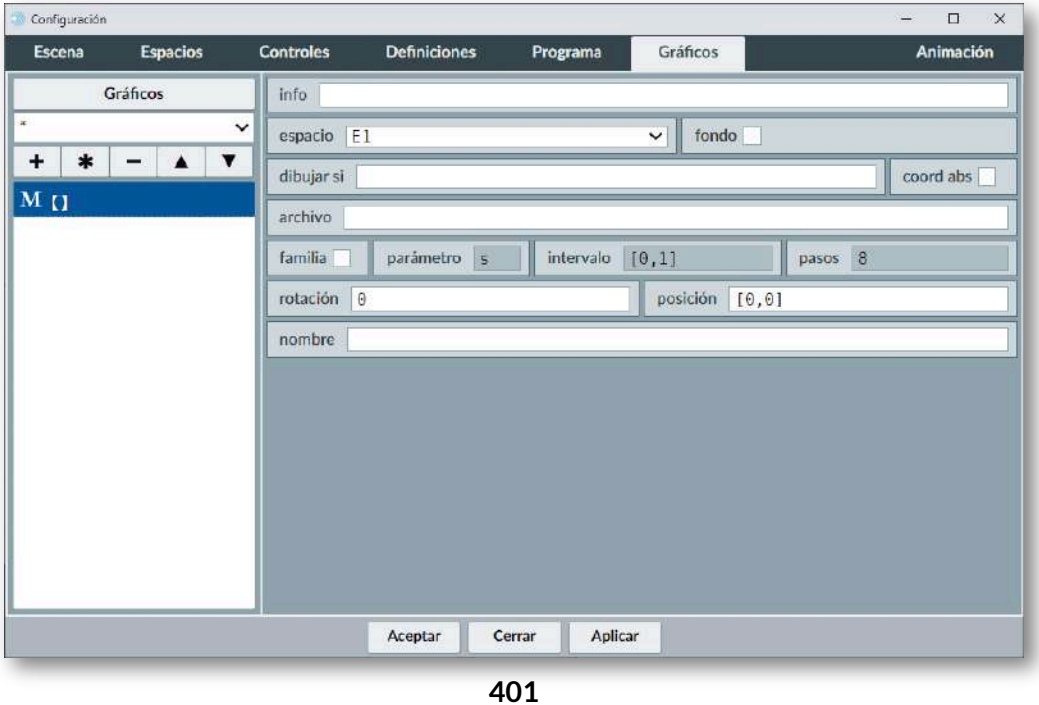

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
【(0,0),r,0,360】
【(x,y)】
【(x,y)】
info
espacio E0
fondo
color
rastro
dibujar si
coord abs
expresión (x,y)
familia
parámetro s
intervalo [0,1]
pasos 8
Aceptar
Cerrar
Aplicar


Color
3000
r
0,50
x
0,00
y
0,00

5.14 Gráfico Macro

Un macro es un archivo de texto que contiene parte del código de una escena de **Descartes** y que, al ser interpretado por Descartes, actúa como un gráfico dentro de la escena. Los macros pueden utilizarse varias veces en la misma escena y también en escenas diferentes. Por ejemplo, existe un macro que dibuja cinco círculos; cada vez que se llame al macro, dibujará cinco círculos. Así, si se desean dibujar 15 círculos, basta con colocar el macro tres veces, evitando tener que agregar 15 elementos tipo arco (círculos).

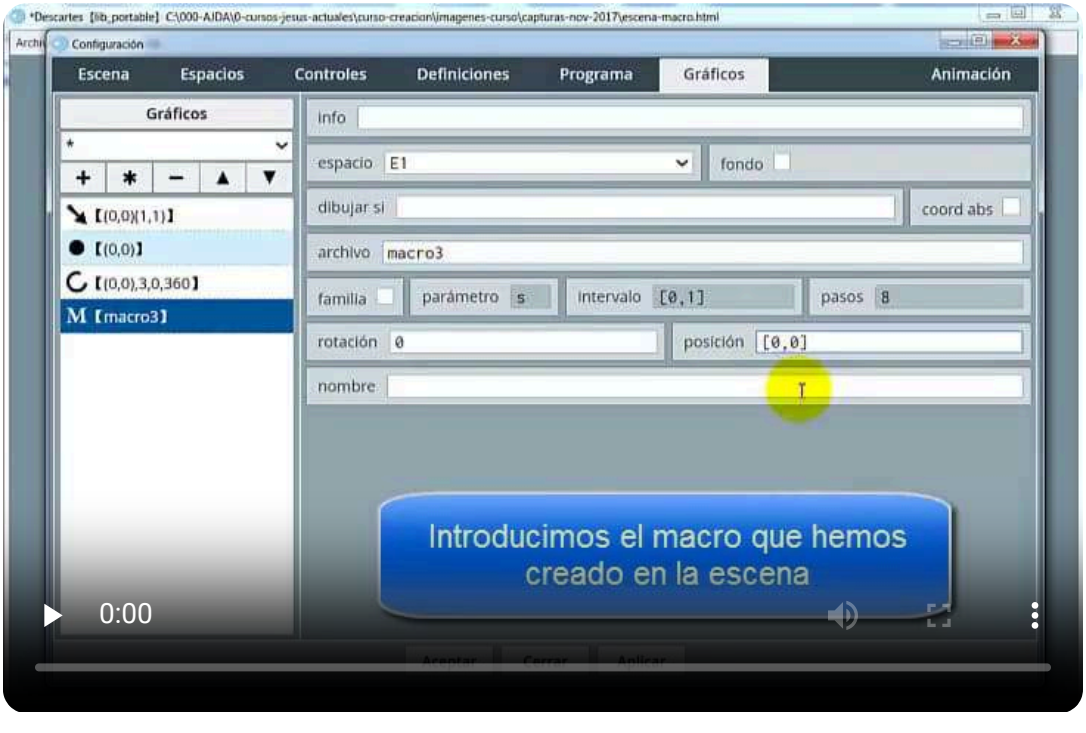
Por tanto, los macros se consideran objetos gráficos que pueden contener elementos de programación (Definiciones y Programa) y/o gráficos. Su función es permitir que todos los elementos del macro puedan ser usados desde las escenas en bloque y tantas veces como se necesite. Desde la escena que utiliza un macro, es posible modificar las propiedades o valores de sus elementos. El panel de configuración específico para un macro tiene el siguiente aspecto:



Los parámetros específicos presentes en los gráficos tipo macro son los siguientes:

- **Rotación:** es un campo de texto cuyo contenido puede ser una expresión o un número que indica el ángulo en el que se girará el macro (en sentido contrario a las agujas del reloj) al dibujarse.
- **Posición:** es un campo de texto cuyo contenido debe ser del grupo (X,Y), donde X e Y son expresiones reales que representan la abscisa y ordenada del punto inicial del macro respecto al origen de coordenadas del espacio.
- **Expresión:** ruta relativa y nombre donde se ubica el macro.
- **Nombre:** campo que permite utilizar los parámetros del macro. Para acceder a ellos, se escribe el nombre del macro seguido de un punto y luego el nombre del parámetro.

En el siguiente vídeo se ilustra la utilización de macros en escenas.



Cuando las escenas se ejecutan desde el propio ordenador (y no desde Internet), la lectura directa de ficheros externos suele estar bloqueada por motivos de seguridad del navegador. Por ello, para que se muestren correctamente los datos de un fichero (en este caso de un macro), es necesario añadirlos al final del código HTML de la escena el contenido del fichero entre las etiquetas:

```
<script type="descartes/macro" id="ruta relativa completa del fichero del macro respecto de la escena">
tipo_de_macro=R2
Idioma=español
elemento 1
elemento 2
...
</script>
```

De este modo, el macro queda integrado en la propia escena. Si en el editor de Descartes está activada la opción **"macro"** dentro del submenú *Agregar al html* del menú *Opciones*, este código se incorporará automáticamente al final del archivo de la escena (de forma análoga sucede con las matrices, bibliotecas y vectores).

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <script type="text/javascript" src="lib/descartes-min.js" charset="utf-8"></script>
  <title>TITULO</title>
</head>
<body>

    <ajs name="Descartes" code="Descartes" width="970" height="550">
        código de la escena
    </ajs>

<script type="descartes/macro" id="mm2.txt">
tipo_de_macro=R2
Idioma=español
id='INICIO' algoritmo='sí' evaluar='una-sola-vez'
id='CALCULOS' algoritmo='sí' evaluar='siempre'
tipo='texto' expresión='[20,20]' texto='Texto para un macro. El valor a: [a]' fijo='sí'
</script>

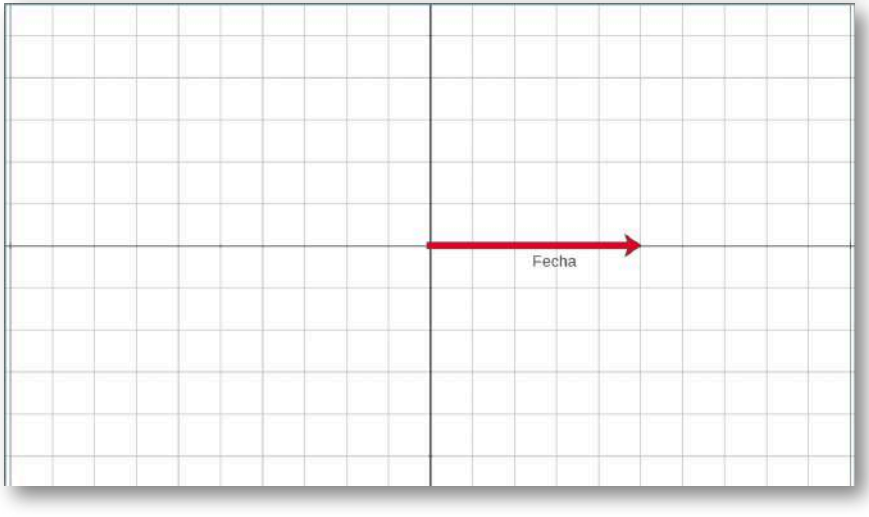
</body>
</html>
```

Creando un macro. Ejemplo 1.

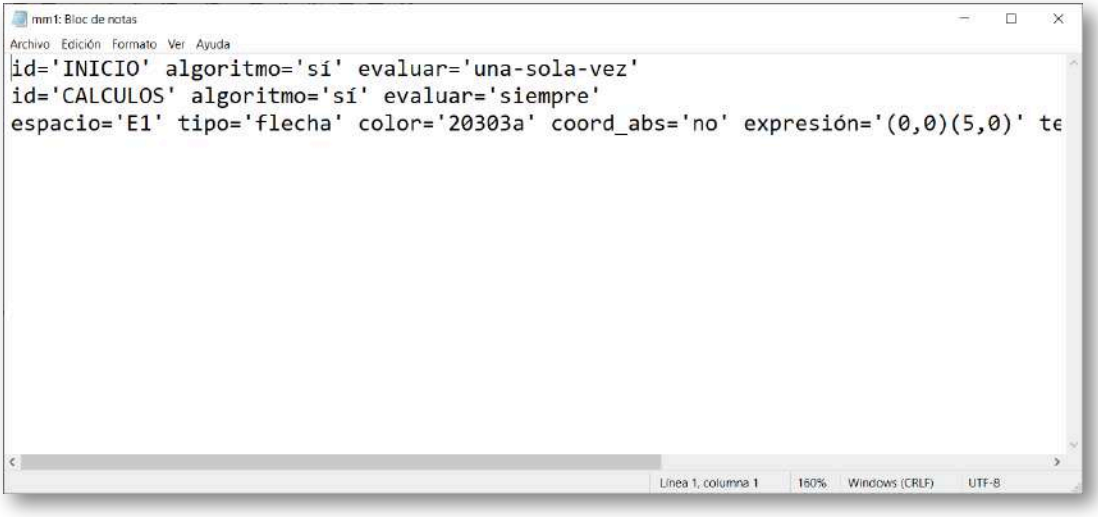
Vamos a crear un ejemplo de macro muy sencillo para entender cómo se pueden crear y utilizar estos elementos. Haremos un macro que dibuje una flecha desde el origen de coordenadas hasta el punto (0,5), en color rojo, con un ancho de 5, una punta de 10 y un texto que la acompañe con la palabra "Flecha".

- **Paso 1:** En una escena en blanco, introducimos el elemento segmento con los parámetros indicados.

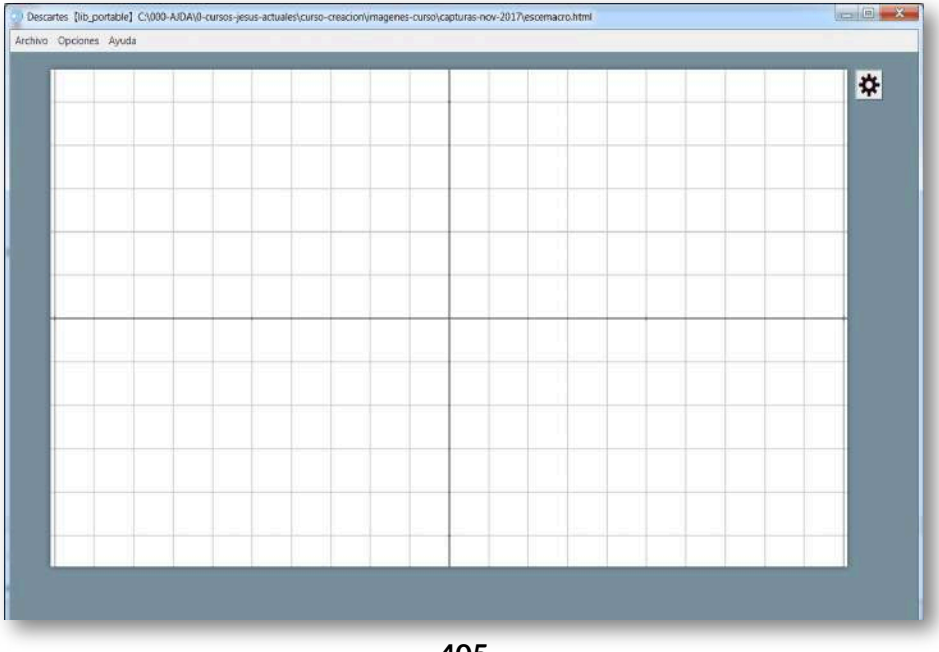




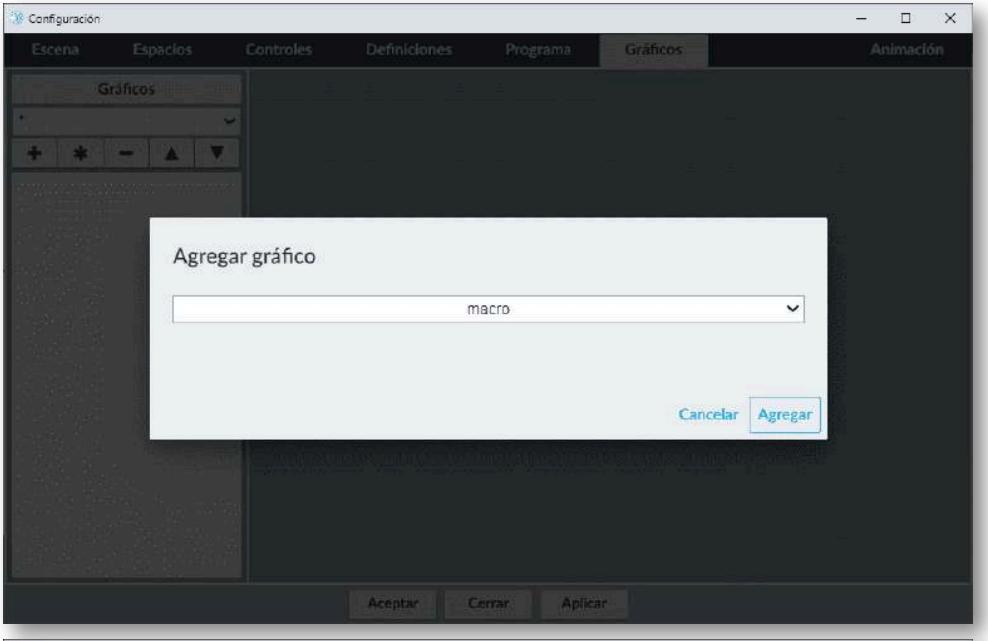
- **Paso 2:** En el editor de **Descartes**, en la opción Archivo, elegimos Exportar a macro de Descartes. Guardamos el archivo con el nombre **mm1.txt**, que contendrá el código del macro.

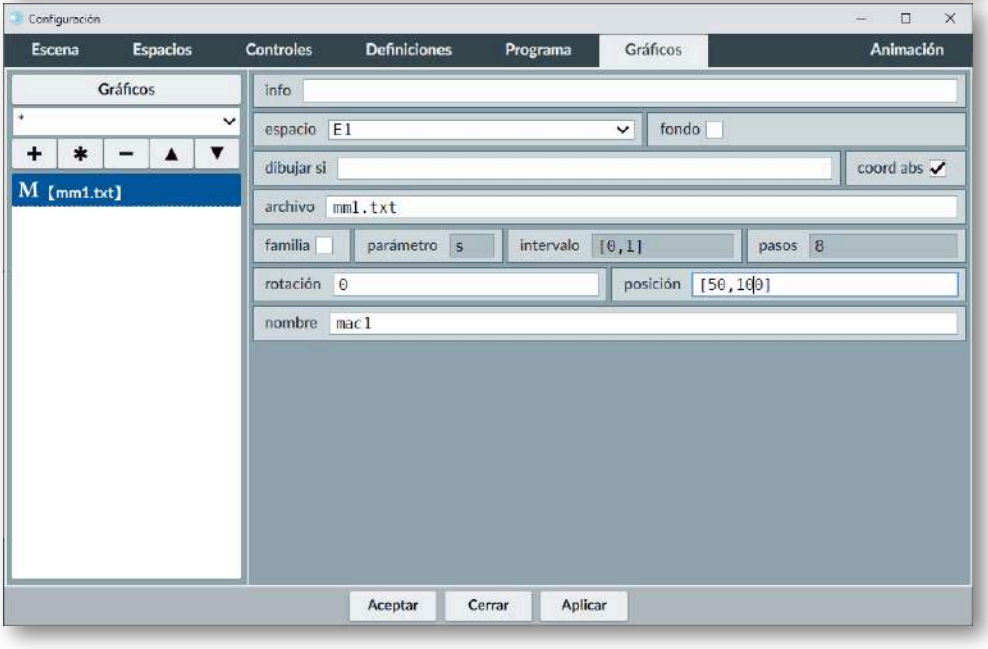


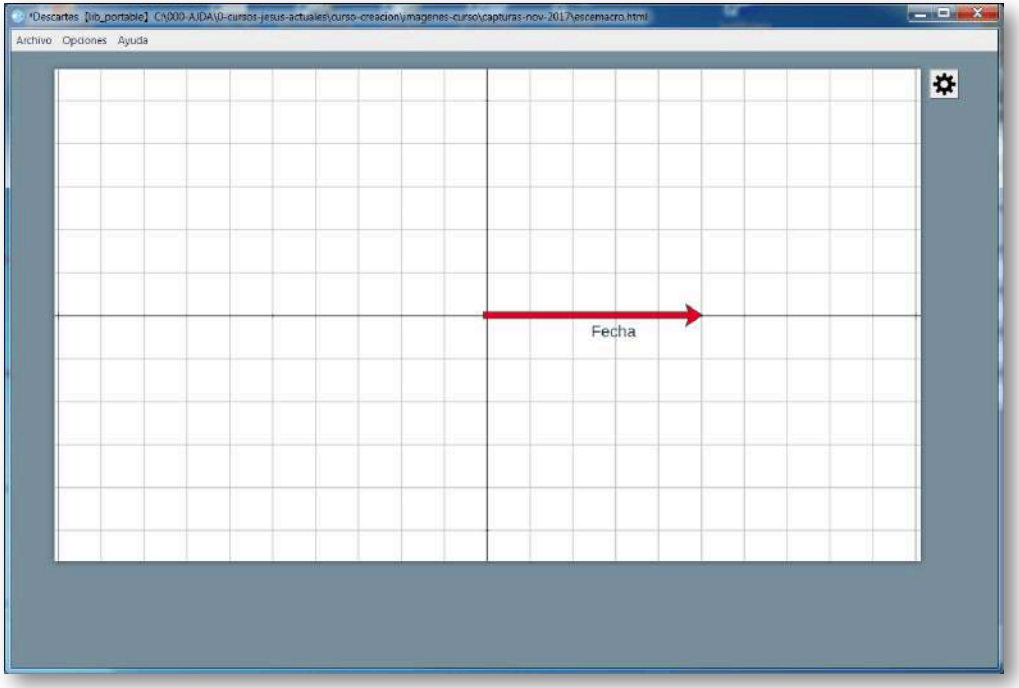
- **Paso 3:** En la misma carpeta donde guardamos el macro **mm1.txt**, creamos una escena en blanco con el gestor de escenas y la guardamos con el nombre **escemacro.html**.



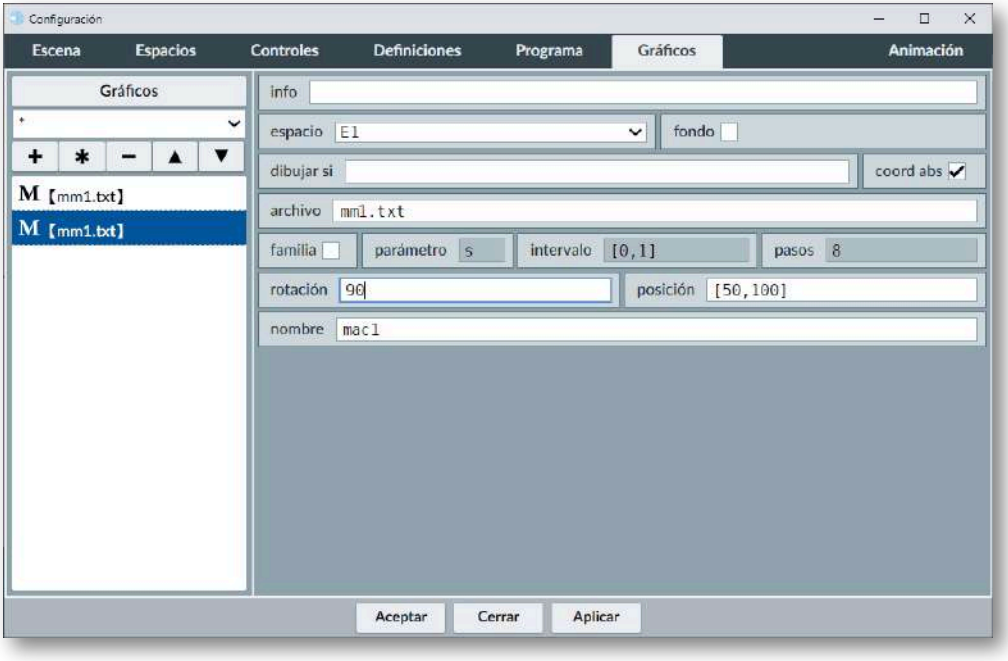
- **Paso 4:** Introducimos el macro en la escena: en el menú de la izquierda de Gráficos pulsamos **+** (Añadir) y en la ventana que aparece seleccionamos Macro. El macro será **mm1.txt** y el nombre que le daremos en la escena será **mac1**.



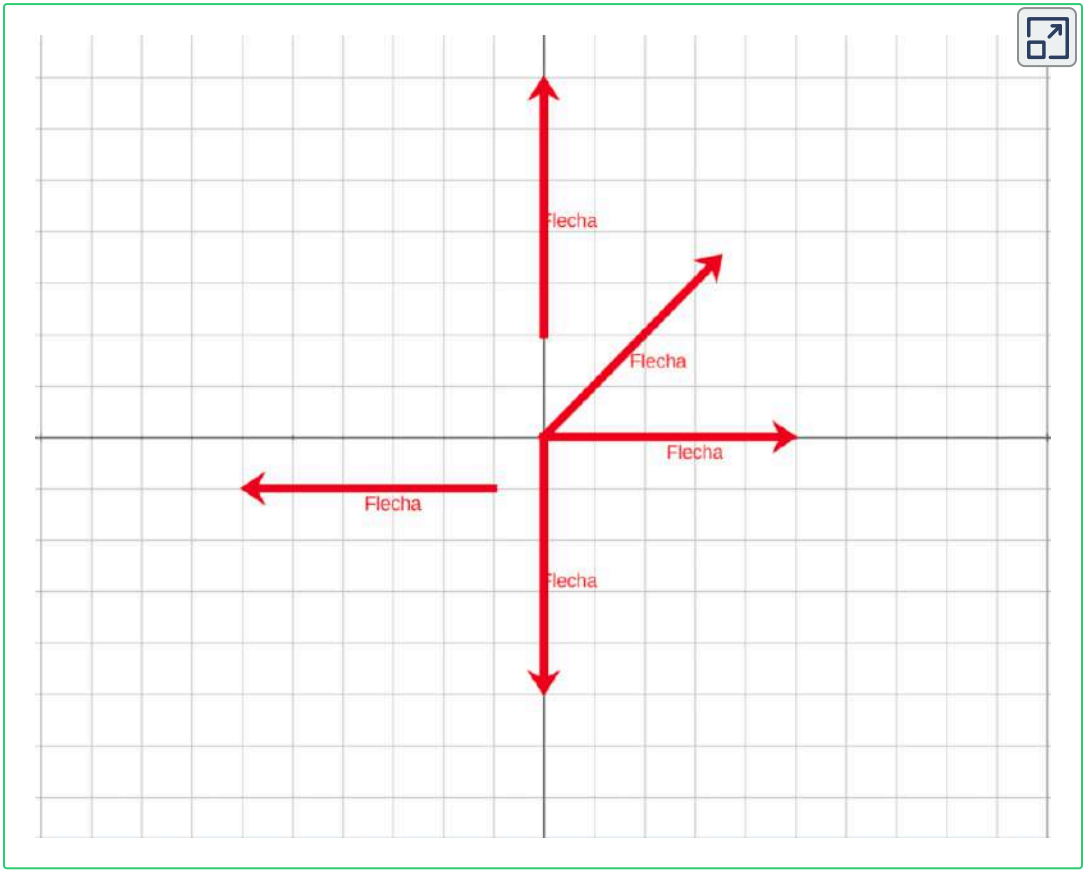




- **Paso 5:** Añadimos más macros basados en **mm1.txt**, usando diferentes posiciones y rotaciones.



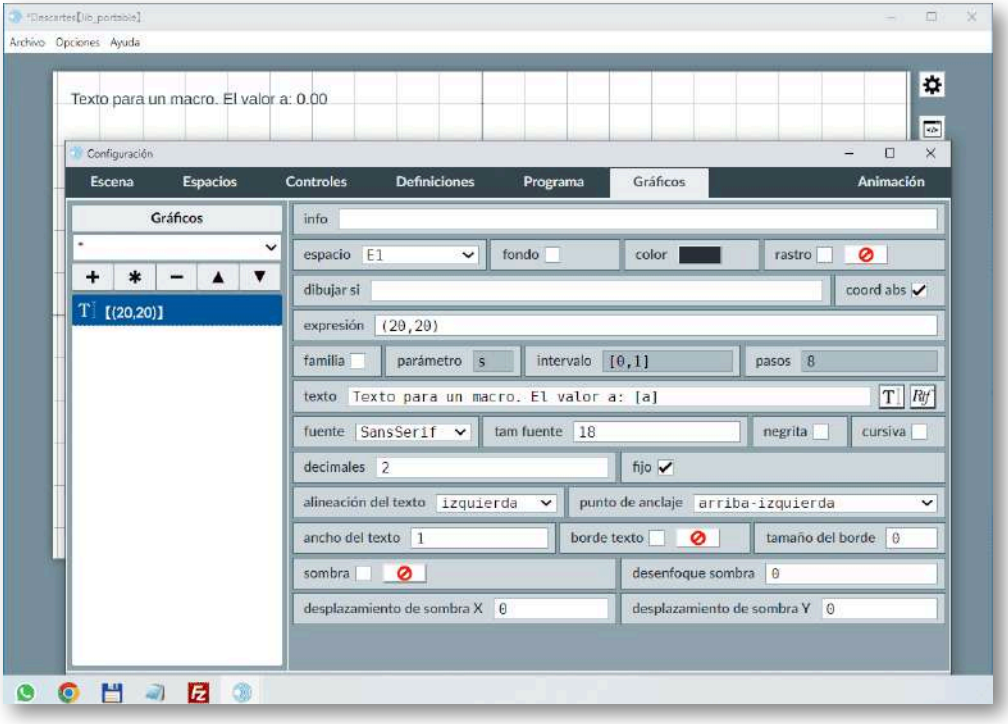
La escena obtenida, según la configuración específica que se le asigne, podría ser como la que se muestra a continuación. También podrían haberse añadido distintos macros, con diferentes disposiciones y orientaciones.



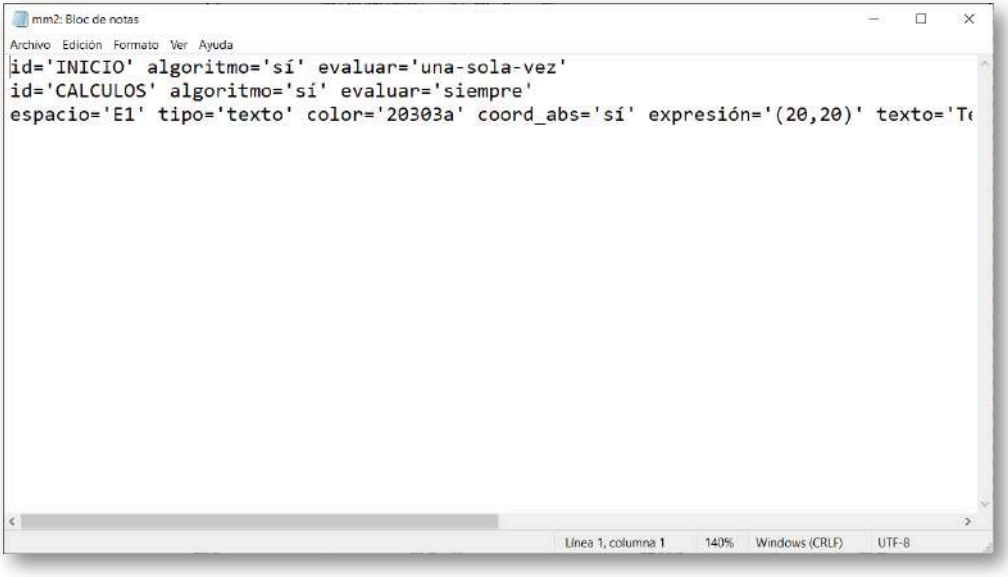
Creando un macro. Ejemplo 2.

Vamos a crear un ejemplo de macro muy sencillo que contenga una variable **a**, la cual pueda ser modificada desde la escena que llama al macro.

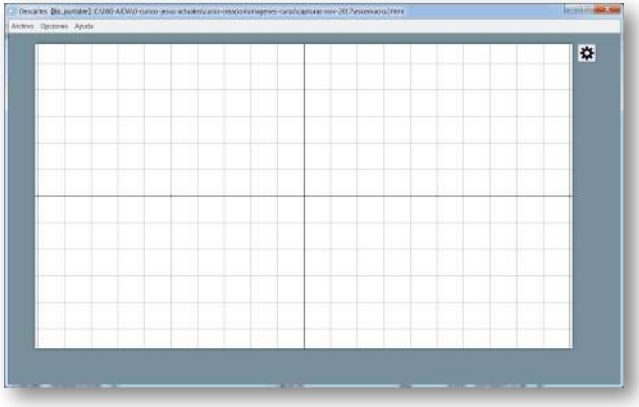
- **Paso 1:** Partimos de una escena en blanco en que introduciremos un texto con una variable tal como se indica a continuación.



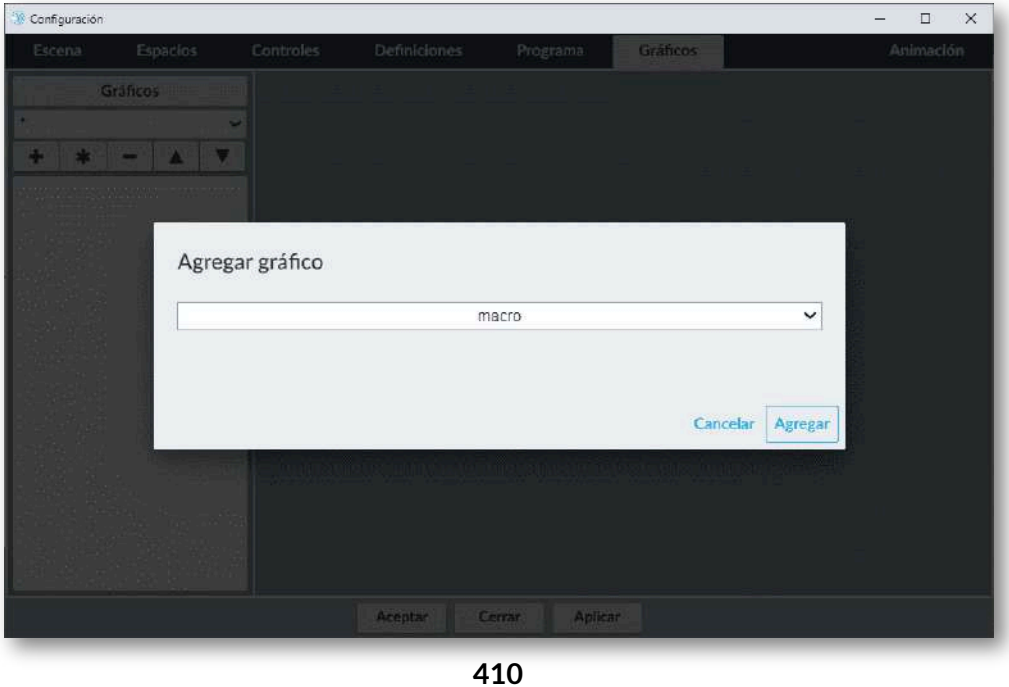
- **Paso 2:** En el editor de **Descartes**, en la opción Archivo, elegimos Exportar a macro de Descartes. Guardamos el archivo con el nombre **mm2.txt**, que contendrá el código del macro.

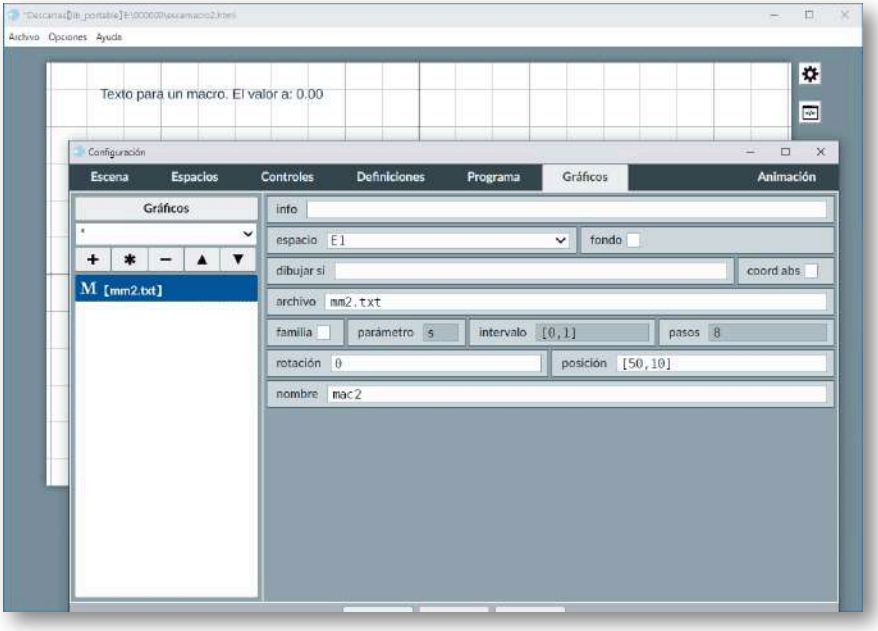


- **Paso 3:** En la misma carpeta donde guardamos el macro **mm2.txt** (fichero del bloc de notas), creamos una escena en blanco con el gestor de escenas y la guardamos con el nombre **escemacro2.html**.

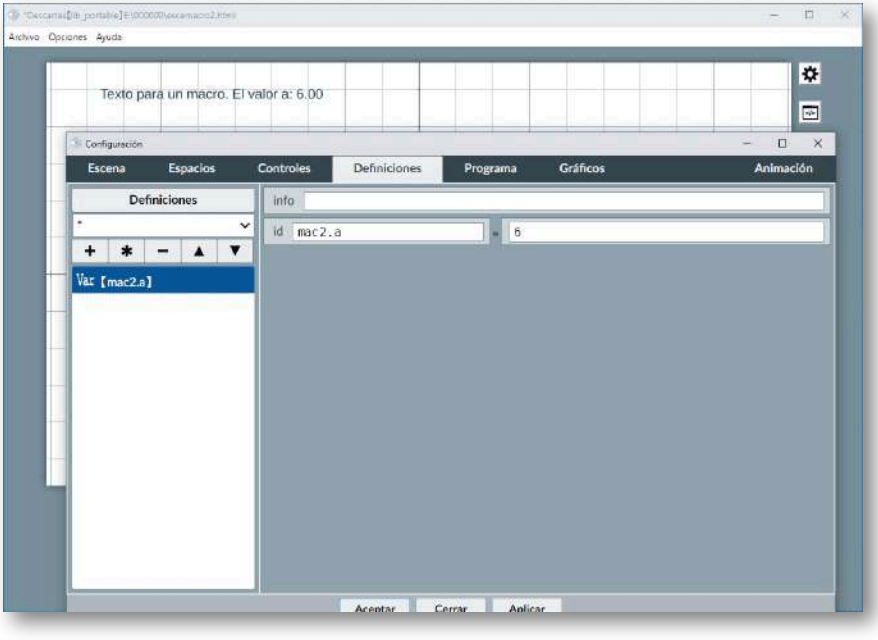


- **Paso 4:** Introducimos el macro en la escena: en el menú de la izquierda de Gráficos pulsamos **+** (Añadir) y en la ventana que aparece seleccionamos Macro. El macro será **mm2.txt** y el nombre que le daremos en la escena será **mac2**.

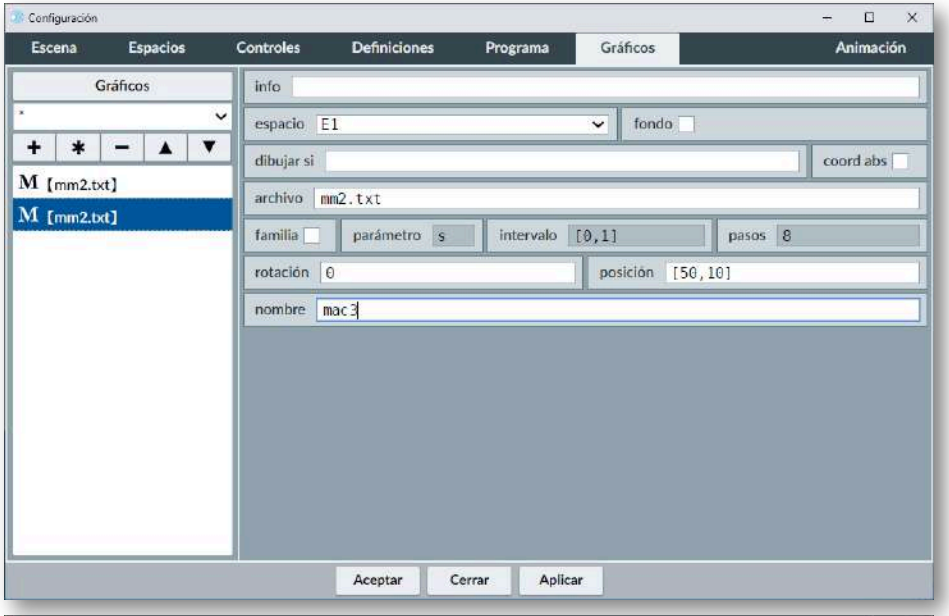


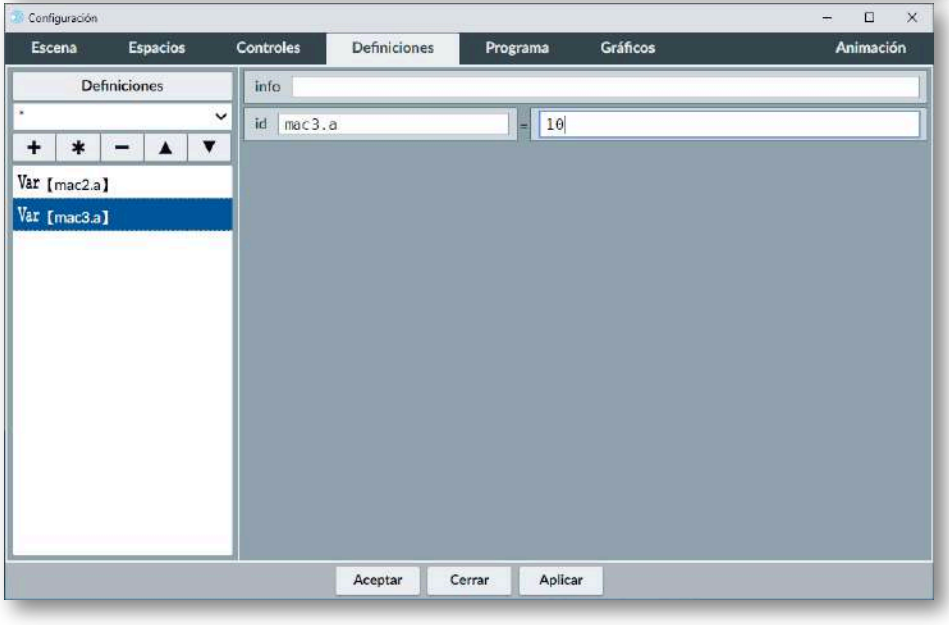


- **Paso 5:** Damos valor a la variable **a** del macro. Para ello, introducimos la variable **mac2.a** (nombre del macro.nombre de la variable) con un valor, por ejemplo 6, y veremos cómo cambia el valor de la variable en el macro.

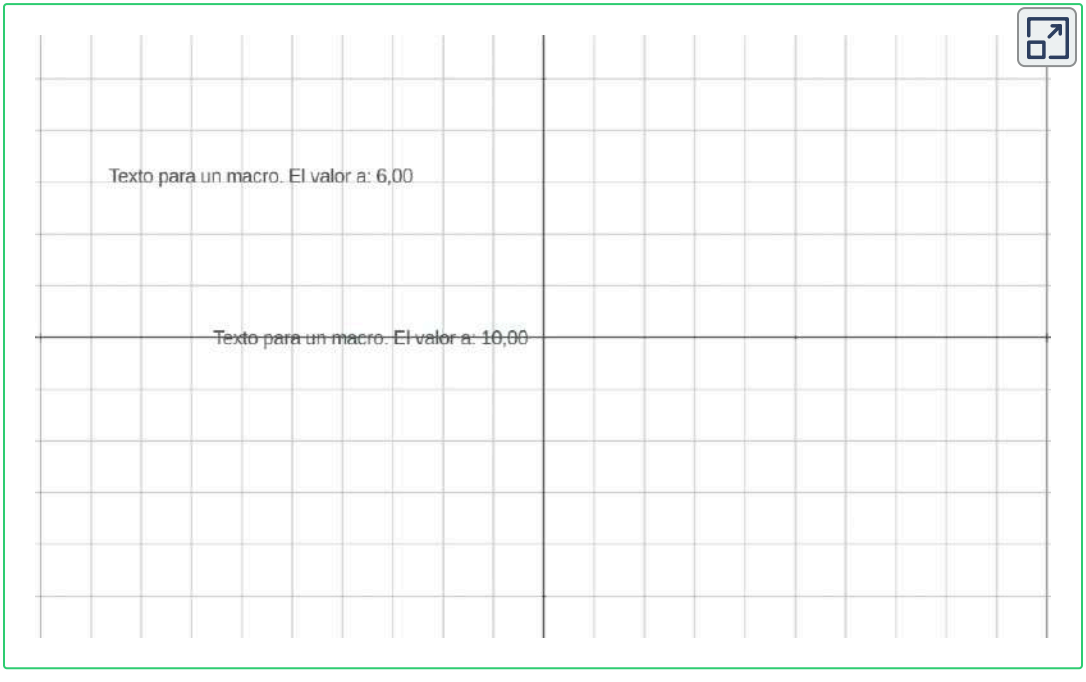


- **Paso 6:** Insertamos otro macro basado en **mm2.txt**, al que llamaremos **mac3**. Después definimos la variable **mac3.a** con un valor, por ejemplo 10, y observamos cómo **mac3.a** sólo afecta al valor de **a** en el macro **mac3**, mientras que **mac2.a** afecta únicamente a la variable **a** del macro **mac2**.





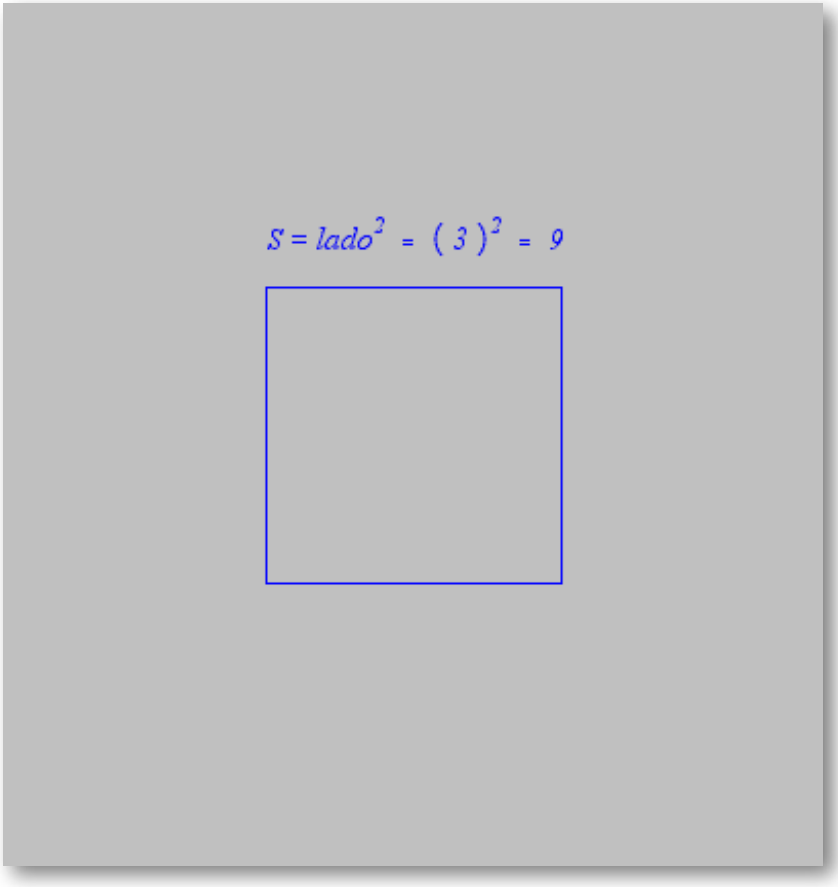
La escena obtenida, según la configuración específica que se le asigne, podría ser como la que se muestra a continuación. También podrían haberse añadido distintos macros, con diferentes disposiciones y orientaciones.



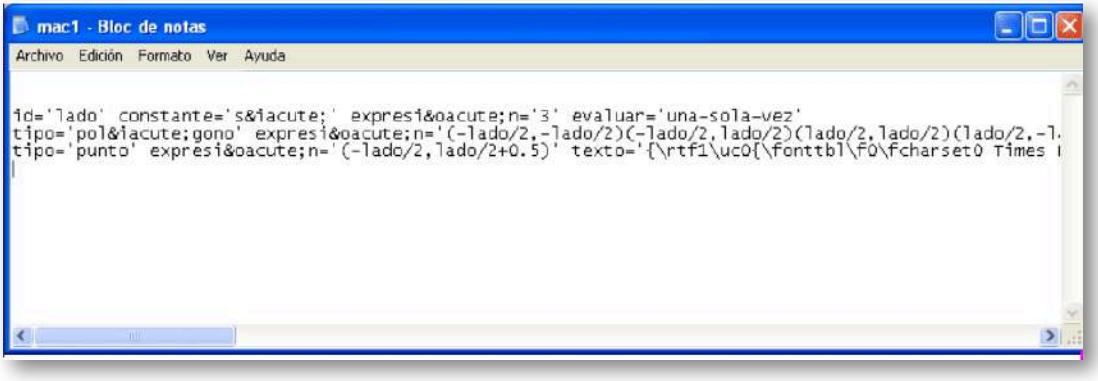
Creando un macro. Ejemplo 3.

A continuación, se describen los pasos necesarios para crear un nuevo ejemplo de escena con un macro que incluya variables modificables desde la propia escena:

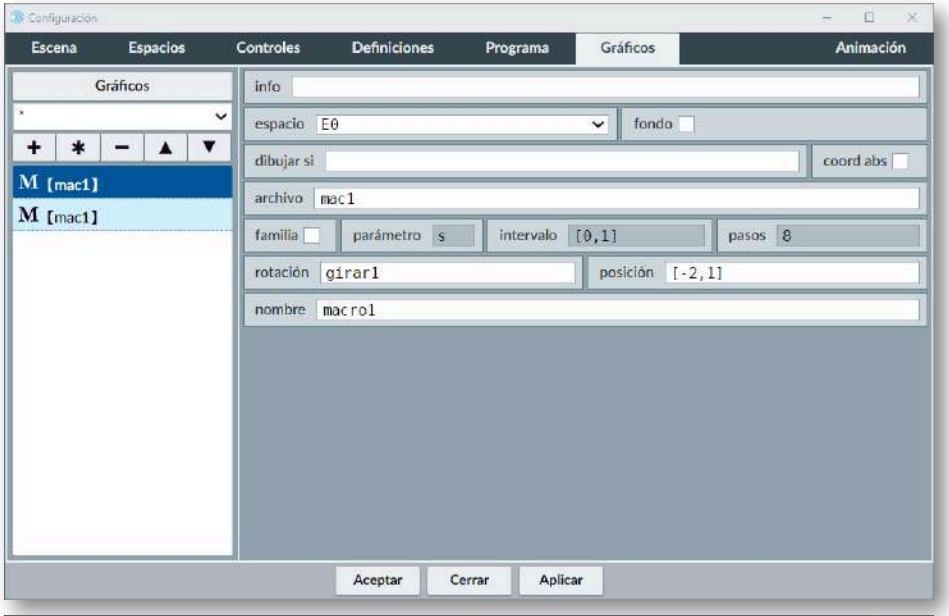
- **Paso 1.** Se crea una escena con un único espacio 2D, que puede incluir elementos gráficos 2D, definiciones y/o programa dentro de ese mismo espacio. Por ejemplo, podemos diseñar una escena en la que se dibuje un cuadrado junto con la expresión de la fórmula de su área, cuyo lado tenga inicialmente un valor de 3 unidades (un único espacio con un polígono, un texto y los elementos de programación necesarios).

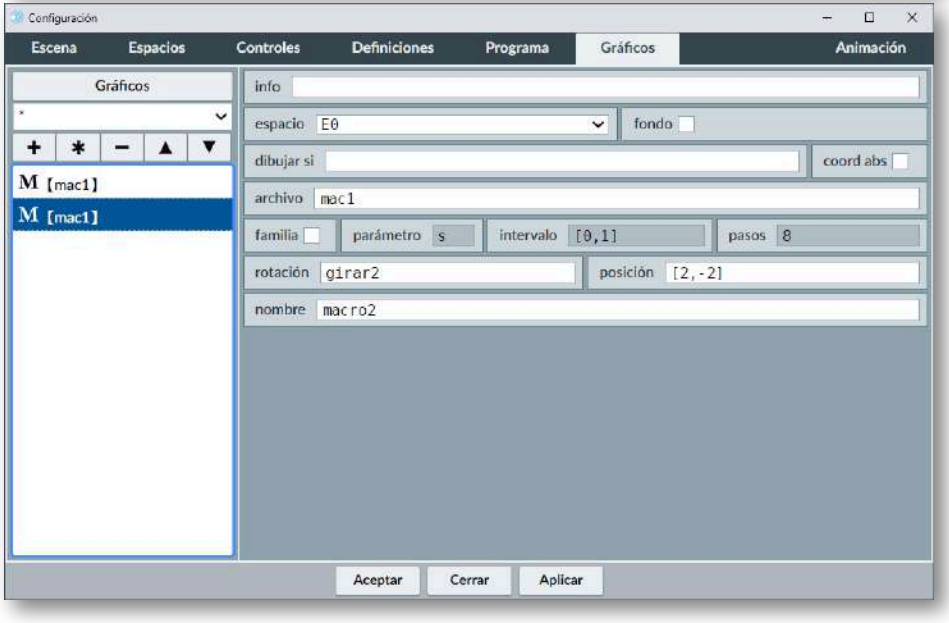


- **Paso 2.** En el editor de **Descartes**, dentro de la opción Archivo, elegimos Exportar a macro de Descartes y lo guardamos con el nombre **mac1**. El archivo generado contendrá el código del macro.

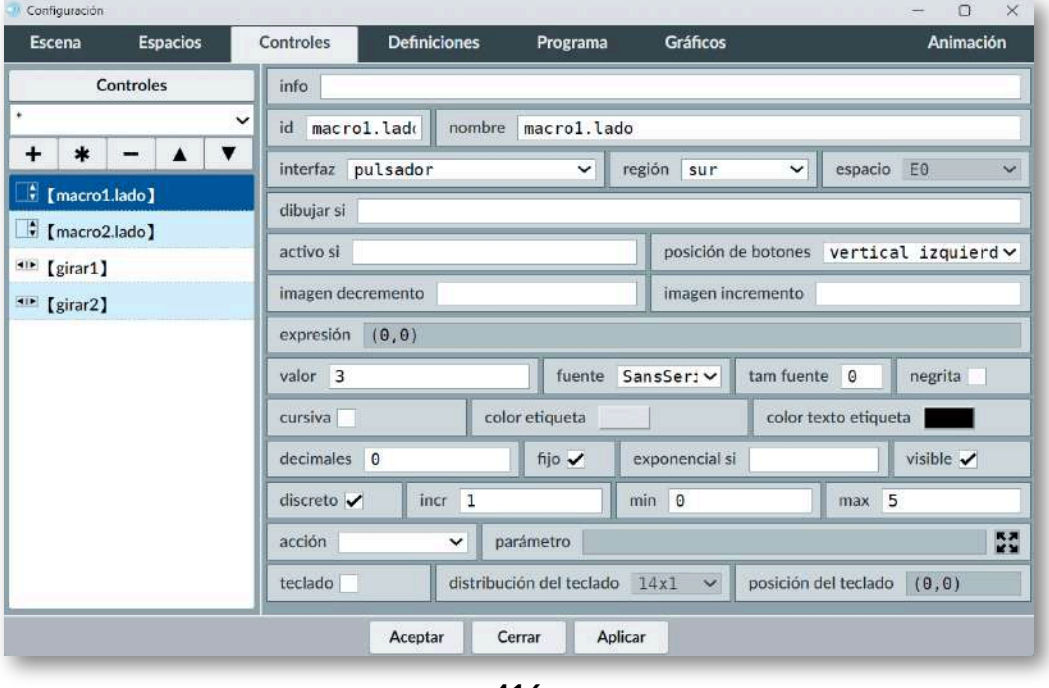


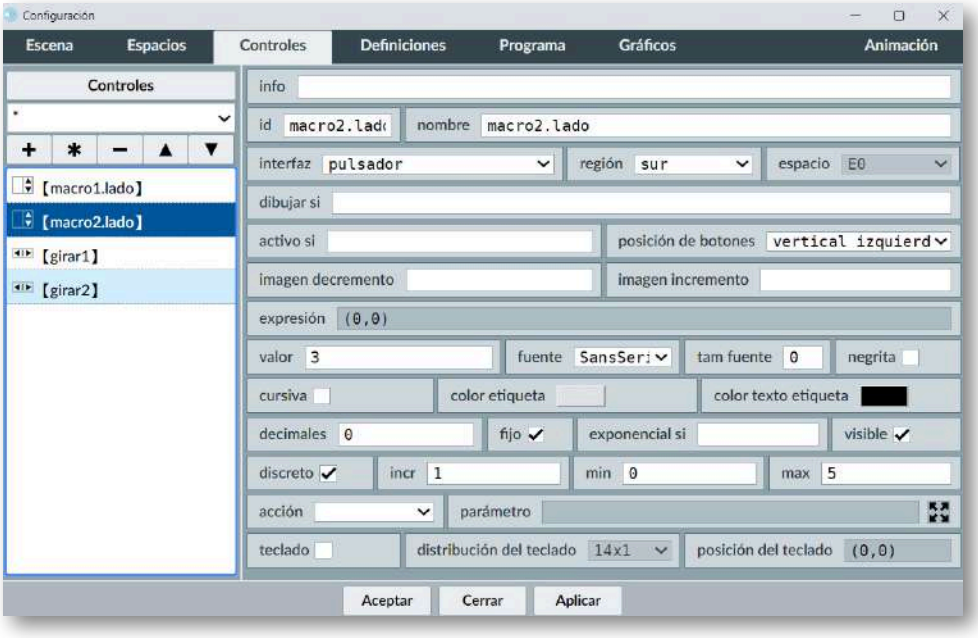
- **Paso 3.** Desde una escena nueva con un espacio 2D, el macro puede utilizarse tantas veces como se desee y es posible modificar sus parámetros. En nuestro ejemplo, insertaremos dos veces el macro **mac1** en distintas posiciones y con diferentes rotaciones.



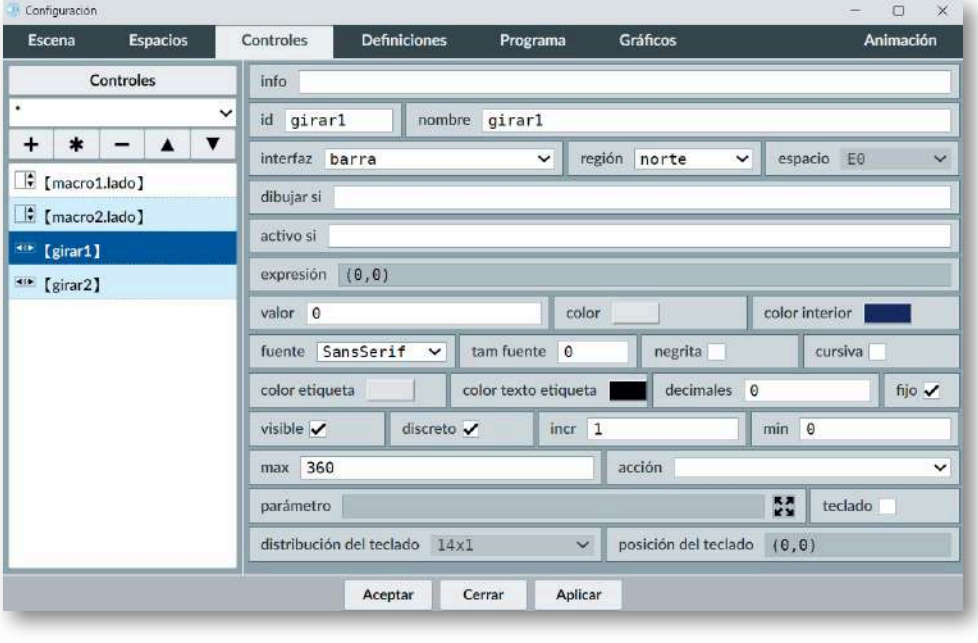


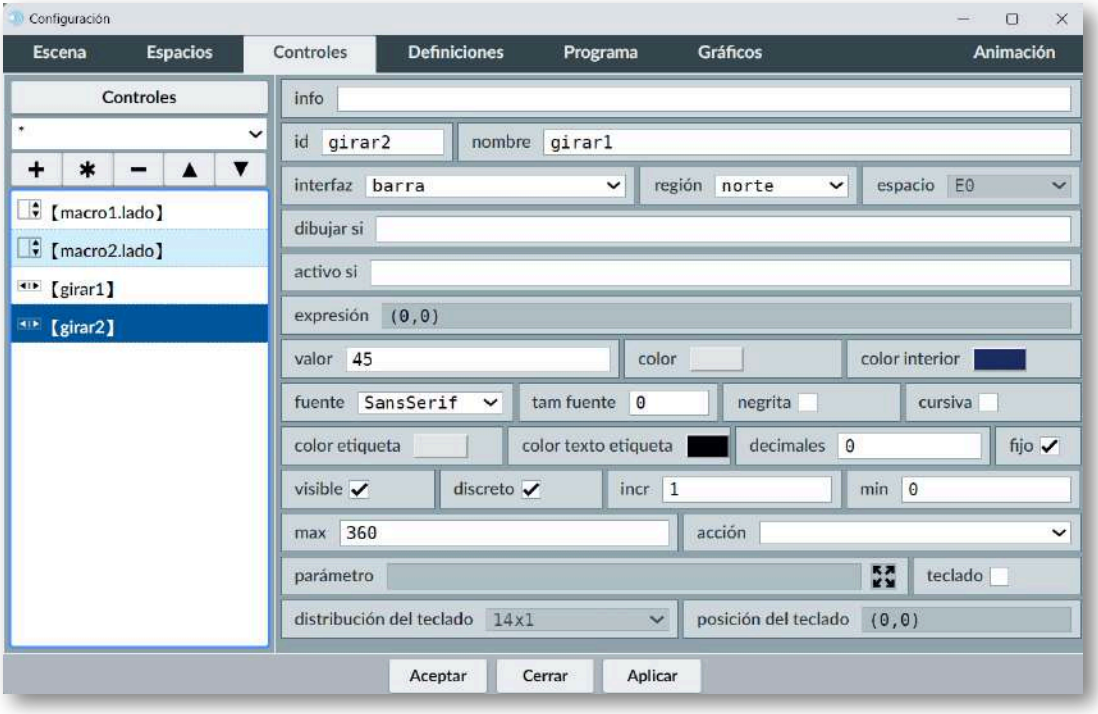
- **Paso 4.** Definiremos controles de tipo pulsador que permitan modificar la longitud del lado del cuadrado contenido en el macro.
    - El macro debe tener previamente definida la variable **lado** en su estructura, tal y como se puede observar en el código de la imagen del **Paso 2**.
    - Desde la escena, el valor de una variable del macro se modifica utilizando la estructura **nombreDelMacro.nombreDeLaVariable**.
    - En nuestro caso, emplearemos **macro1.lado** para variar el valor de **lado** en el primer macro.
    - También utilizaremos **macro2.lado** para modificar el valor de **lado** en el segundo macro.
    - De este modo, el valor de la variable **lado** en **macro1** coincidirá con el asignado a **macro1.lado** en la escena y lo mismo ocurrirá con **macro2**.





- **Paso 5.** Mediante dos controles de tipo barra de deslizamiento definiremos las variables **girar1** y **girar2**, que están asociados al campo de rotación de los macros, como se puede ver en las imágenes del **Paso 3**.



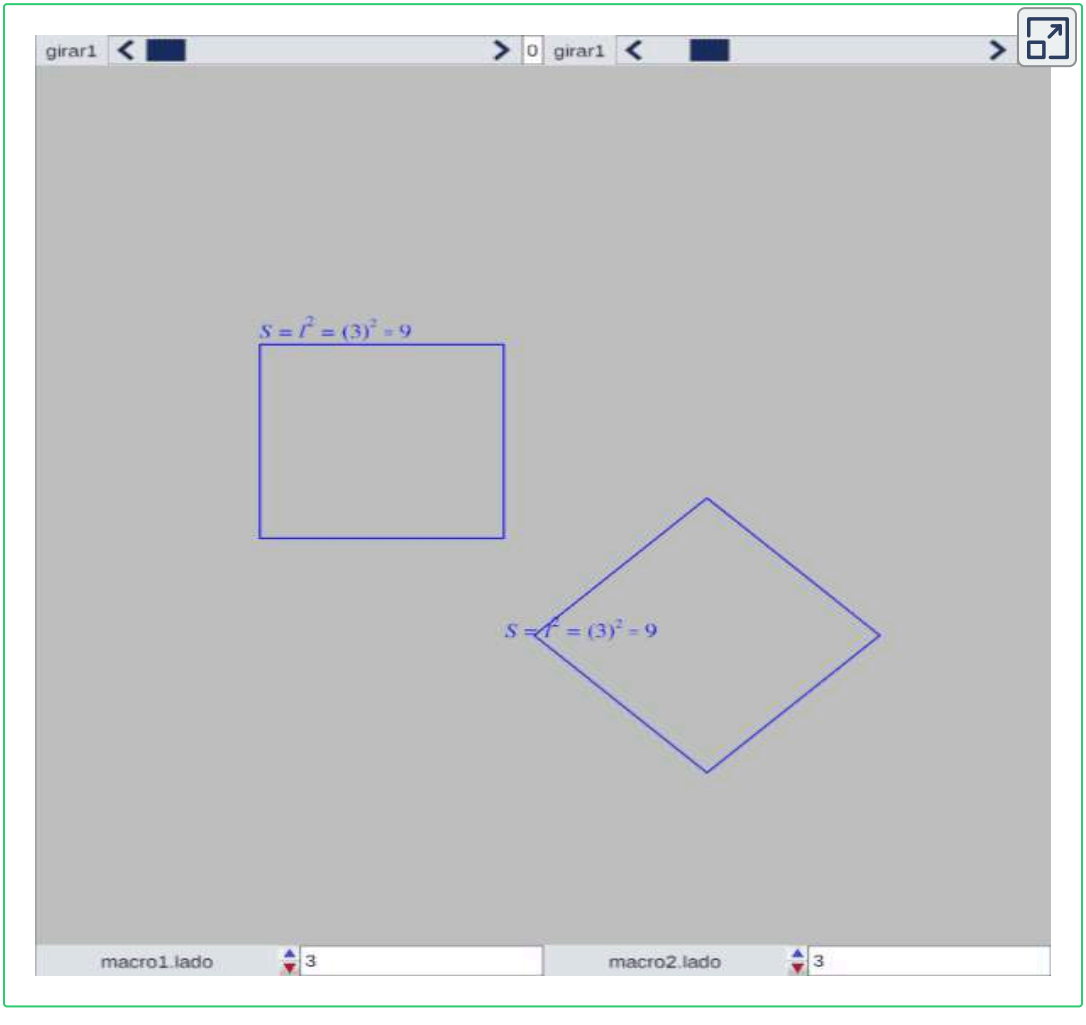


Es fundamental distinguir qué variables afectan a los macros y de qué manera lo hacen.

- En nuestro ejemplo, existen variables que afectan directamente a los datos internos del macro, como **lado**. Estas variables forman parte del propio código del macro y determinan su comportamiento interno. Para poder modificarlas desde la escena que utiliza el macro, es necesario anteponer el nombre del macro, usando la sintaxis **macro1.lado** o **macro2.lado**. De este modo, cada macro puede mantener su propio valor independiente, lo que permite que varios macros puedan coexistir en la misma escena sin interferir entre sí.
- Por otra parte, existen variables que afectan al macro de manera externa a su código, como **girar1** y **girar2**. Estas variables no forman parte del código interno del macro, sino que controlan aspectos de su presentación, como la rotación, desde fuera del macro.

- En estos casos, la variable se trata como cualquier otra variable de la escena y puede utilizarse directamente con su nombre, por ejemplo, **girar1** o **girar2**. Esto facilita la interacción del usuario y permite vincular los valores de estas variables con controles de la escena, como barras de deslizamiento, botones o entradas numéricas, para modificar la apariencia del macro en tiempo real.

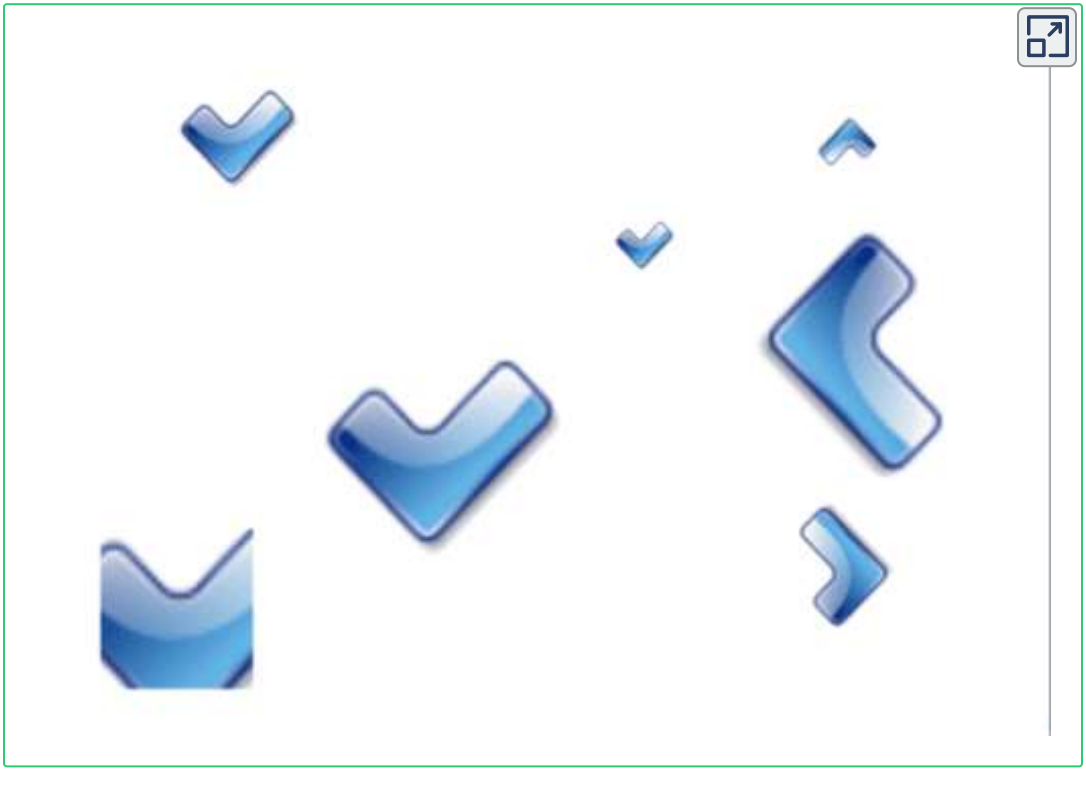
Presentamos a continuación la escena resultante del ejemplo que acabamos de desarrollar.



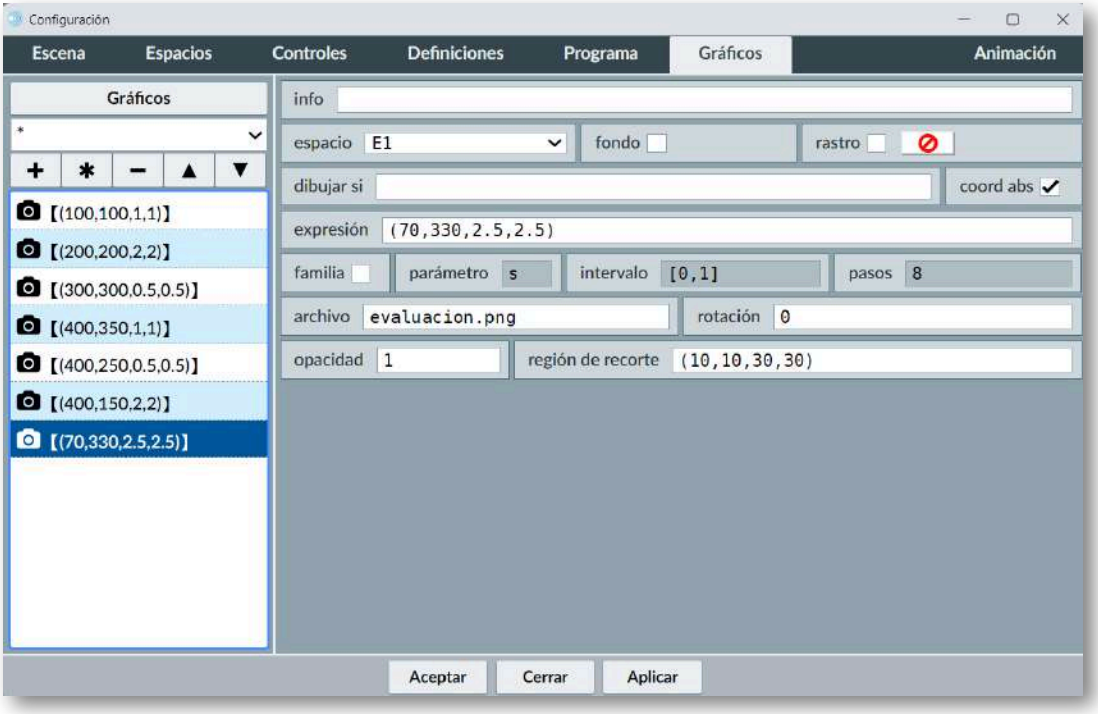
5.15 Gráfico Imagen

En el capítulo 2 analizamos cómo introducir imágenes en las escenas y que se admiten los tipos jpg, png, gif, svg y webp. Recordamos que, al agregar imágenes, el archivo correspondiente debe acompañar al de la escena, siguiendo la ruta relativa indicada en el campo **Fichero** del panel de configuración de la imagen. En este apartado abordaremos otros aspectos relevantes de las escenas.

En la siguiente escena se muestra cómo, a partir de una sola imagen, es posible posicionarla, escalarla, rotarla o recortarla de distintas maneras.



Se presenta el panel de configuración de una imagen, usando una de las imágenes introducidas en el ejemplo anterior.



Repasamos los parámetros propios de las imágenes:

- **Expresión:** campo de texto en el que se introducen las coordenadas donde se mostrará la imagen en el interactivo. Por defecto son dos coordenadas que indican la esquina superior izquierda de la imagen. También se pueden introducir cuatro entradas: la tercera y cuarta representan el factor de escala del ancho y alto de la imagen. En este caso, las dos primeras entradas pasan a indicar el centro de la imagen. Los factores de escala pueden ser negativos, invirtiendo la imagen horizontal y/o verticalmente.
- **Archivo:** campo donde se indica la ruta al archivo de la imagen. Si se encuentra en una subcarpeta, se utiliza la diagonal normal (/) para separar los niveles. También admite direcciones de imágenes alojadas en la web, siempre que la URL apunte directamente al archivo y finalice con su correspondiente extensión.

También es posible agregar una imagen directamente en el HTML mediante el siguiente código:

<script type="descartes/imagen" id="nombre_de_la_imagen">

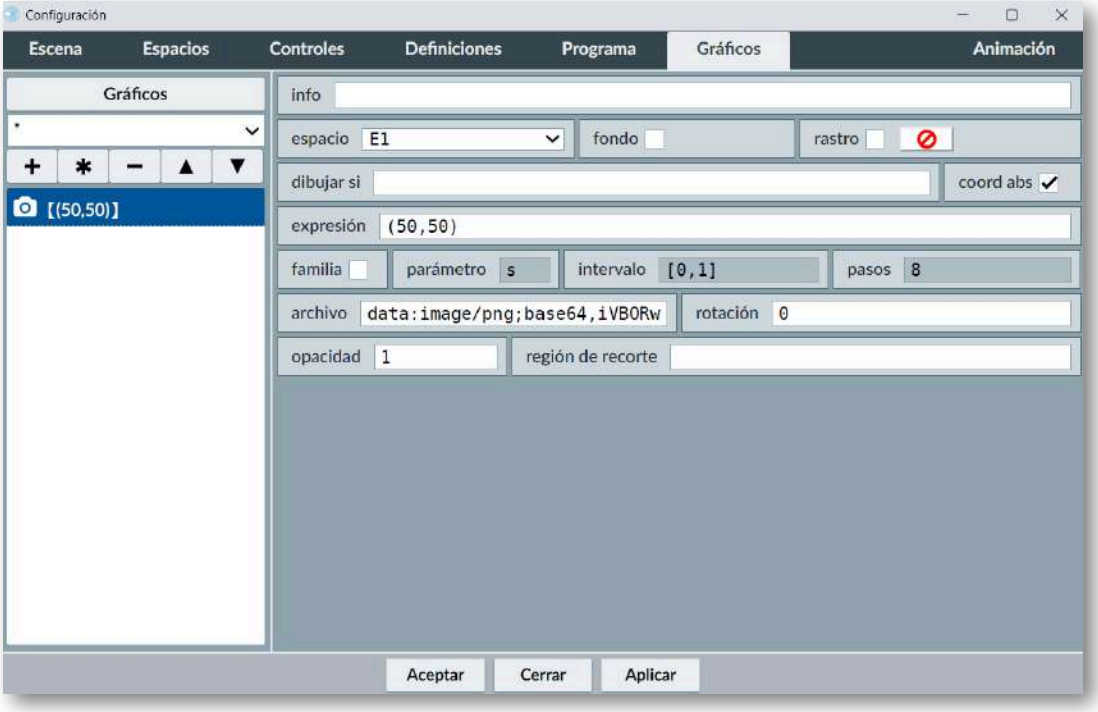
donde el nombre del id será el que se use en el campo **archivo**.

El nombre de la imagen debe incluir la extensión del archivo a partir del cual se generó el código en el script. El contenido del script debe ser la imagen codificada en **base64**, y para obtener este código a partir de un archivo de imagen se puede usar la página https://www.base64-image.de/, que genera el código listo para copiar dentro de las etiquetas `<script>` de apertura y cierre.

Este método resulta útil cuando se desea que la imagen esté incluida directamente dentro del mismo archivo HTML de la escena, evitando así la dependencia de archivos externos y facilitando la portabilidad del proyecto.

- **Rotación:** campo de texto donde se introduce el ángulo en grados de rotación de la imagen.
- **Opacidad:** campo de texto donde se introduce un valor o expresión entre 0 y 1. Un valor de 0 hace la imagen totalmente transparente y 1 la hace completamente opaca.
- **Región del recorte:** campo de texto donde se introduce una expresión de la forma **(x, y, ancho, alto)**. Esta expresión permite mostrar sólo una parte de la imagen. Las coordenadas **x** e **y** indican la esquina superior izquierda del recorte, y **ancho** y **alto** definen las dimensiones del recorte.

Presentamos un ejemplo en el que la imagen del logo de Descartes está codificada en base64, lo que permite incorporarla a la escena sin necesidad de adjuntar ningún archivo adicional. Basta con copiar la imagen en ese formato e incluir el código resultante en el campo **archivo** del elemento imagen.



Ahora guardaremos la misma imagen del logo de Descartes en formato base64 al final del código de la escena, entre las etiquetas **<script type="descartes/imagen" id="logo-descartes.png">** y su correspondiente etiqueta de cierre **</script>**.

El código de la imagen en base64 comienza con **** (si la extensión es PNG) y continúa con un código cuya longitud dependerá del tamaño de la imagen.

e-268-imagen-02.html

Archivo Editar Ver

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>TITULO</title>
  <script type="text/javascript" src="lib/descartes-min.js" charset="utf-8"></script>
</head>

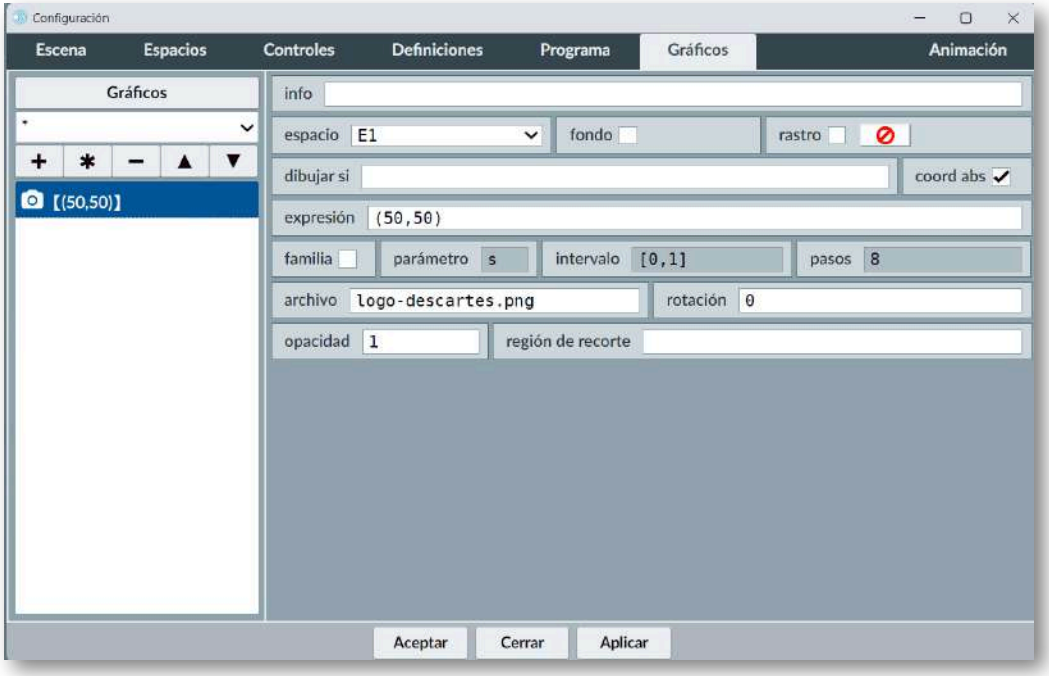
<body>

  <p align="center">
    <ajs name="Animación de un punto" code="Descartes.class" width="600" height="200">
      <param name="tamaño" value="600x200">
      <param name="decimal_symbol" value=",">
      <param name="expand" value="fit">
      <param name="nombre" value="Animación de un punto">
      <param name="Versión" value="5.5">
      <param name="editable" value="sí">
      <param name="Idioma" value="español">
      <param name="Botones" value="créditos=no config=no inicio=no limpiar=no filas_norte=0 filas_sur=0
ancho_este=150 ancho_oeste=150 alto=40 ">
      <param name="E_01" value="tipo='R2' id='E1' ancho='100%' alto='100%' red='no' red10='no' ejes='no' ">
      <param name="A_01" value="id='INICIO' algoritmo='sí' evaluar='una-sola-vez' ">
      <param name="A_02" value="id='CALCULOS' algoritmo='sí' evaluar='siempre' ">
      <param name="G_01" value="espacio='E1' tipo='imagen' coord_abs='si' expresión='(50,50)' archivo='logo-
descartes.png' ">
    </ajs>
  </p>

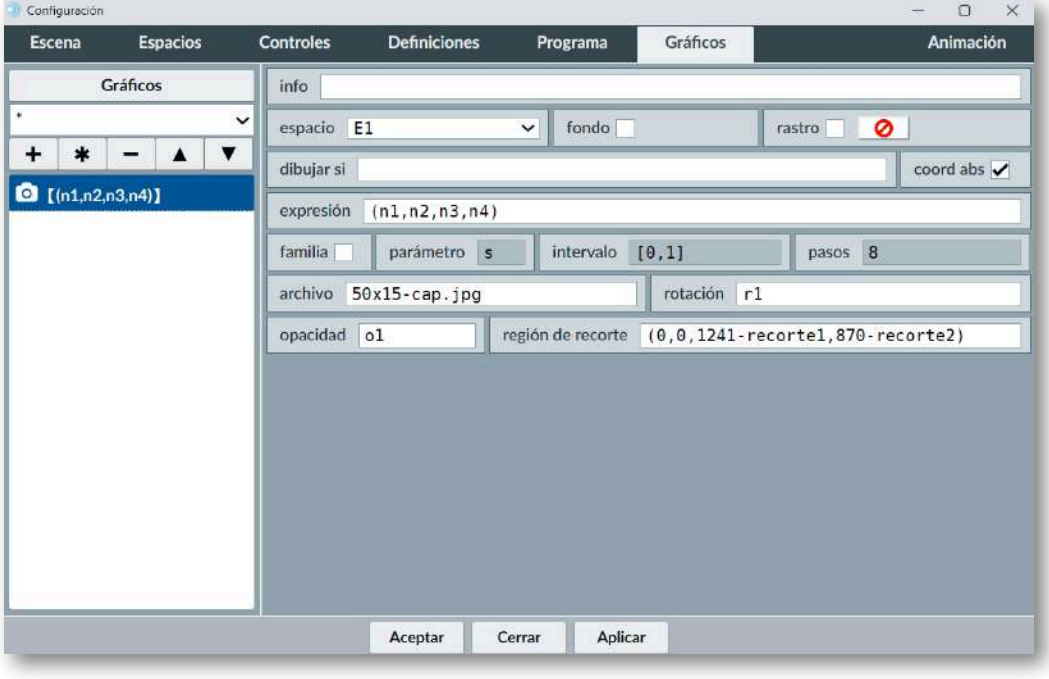
<script type="descartes/imagen" id="logo-descartes.png">

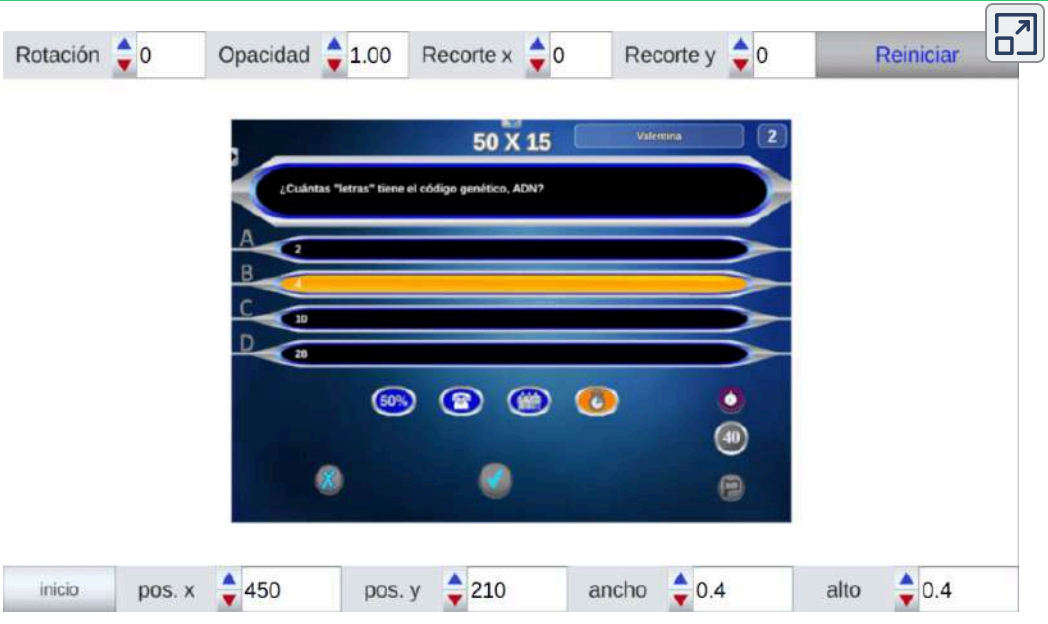
ZppciQ5kqX/4xR9BEAV63GwiswN5vj9PSMzK7OquqdbZJIZQYbT3QzQ5S0KC/f//p8X/oP/qhcLubReR62R//LIwyY/9Pjz3/j+TjF/f3//nfb7
U/r76yHX3w8ZLznf/eefbf58T5PXyz8+
8Mc90vr766H//sb674V+f/HHBV13Ni3lr4vkdft5PeXfC43780Mdvf11qct+vu/fN35L+f2zfi/6rSj+/Dv89YXciNIp3MjNrieP/G3
+uwLXn+RTr/O3OYvip/H9PMLPt9+VEJC/be+P7zH+NUB/C/IfP4V/jr79F8G3+fsO/6dY/r4/8MO//UUq/z74X4j/cmP/c0X291
+Mkdq/bOf3z3unv3d/djdzJaL1t6K+YKc/LsMbFyH372OVr8afws/t+xp89TjjJuUn7rj42mkkI/ovpJxOmuml+
33fabPEbNca3802idJr3ZsN22QpedZXetbI2PFOsrbd4M7L9uda0nff8d1vp86dT+KtlrhY4iP/5Vf47375v/kK722FKCmYVr9YsS5TXbMMZU5/
8y4Skt5v3soX4D++ftMf/1JYKtXM2xTmzgZnXD+XWCX9o7b8y7PzvsL3nxZKoZ3fCxAi7l1YTHIyEGvykmqKzaylRBW7CZqs3DzbIgOpFDss0rJ
7tdCsm+7NZ1r63mvFqullsIlEFK/eyA09RbJyLtRPy50amsVLLqXU0koPZZRZveZaaq2tCuRm85ZbabW11ttos3vPvfTaW+
```

En el campo **archivo** de la imagen se debe indicar el nombre del **id** definido en el script, en este caso **logo-descartes.png**. Es importante no olvidar la extensión (en este caso PNG) con la que se generó la imagen, ya que debe coincidir exactamente. Puede verse que la escena es la misma.

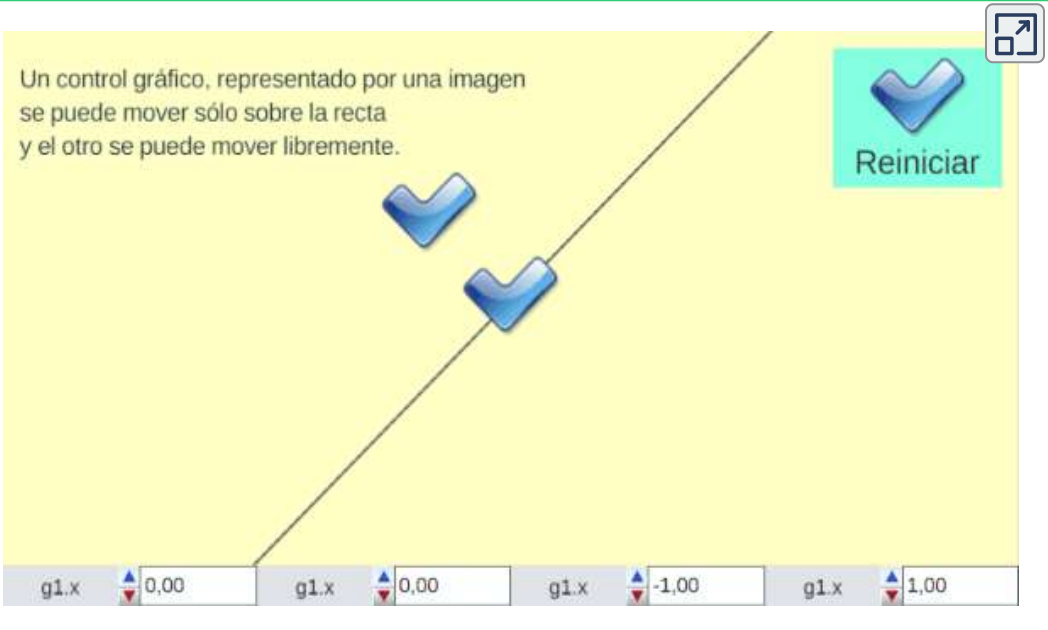


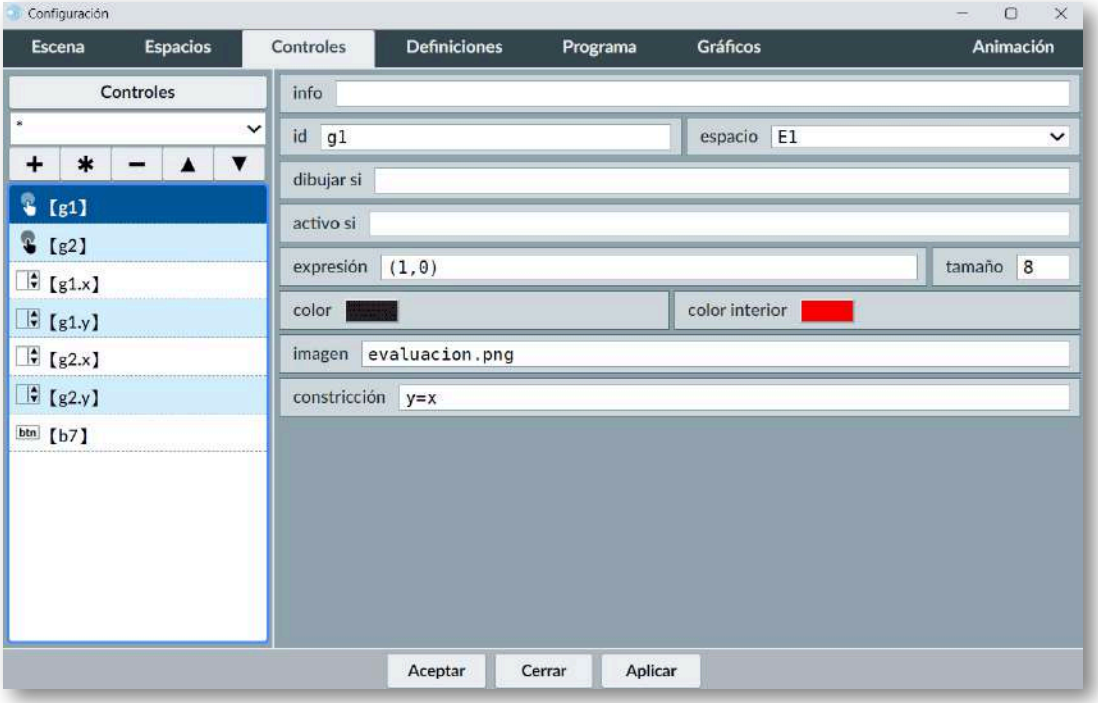
Mostramos otro ejemplo en el que los parámetros de la imagen están definidos mediante controles.

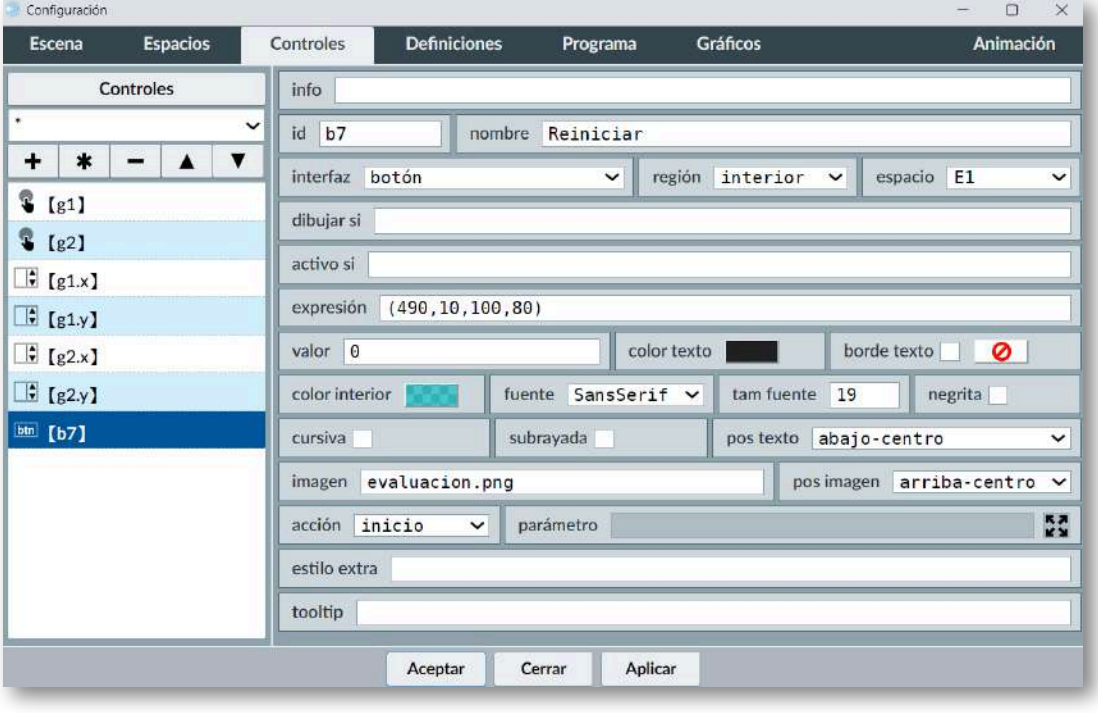




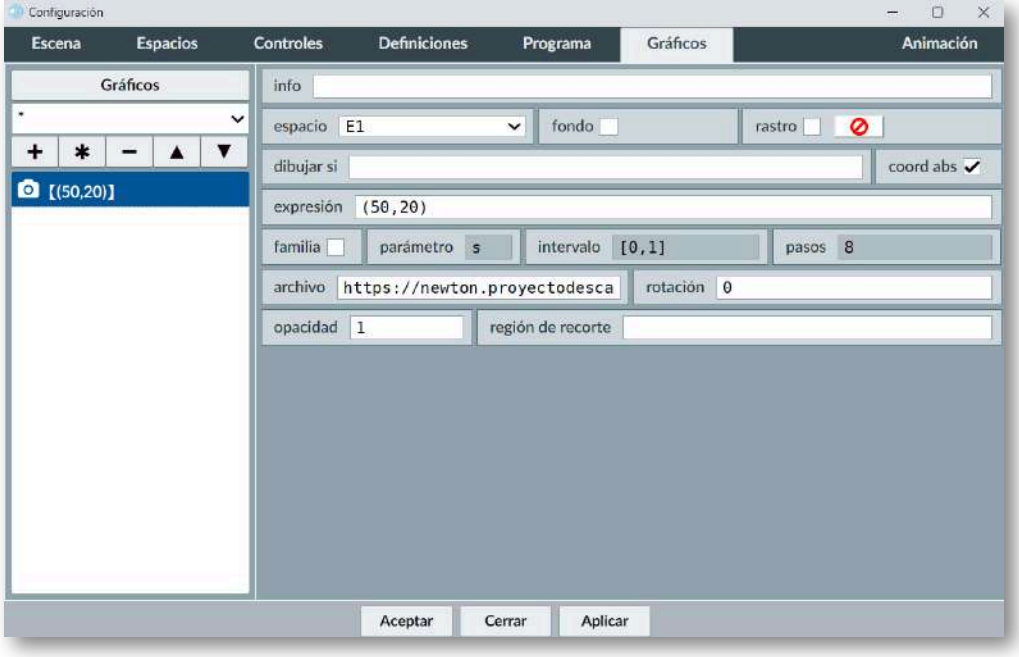
Recordamos también que las imágenes pueden formar parte tanto de los controles gráficos como de los controles de tipo botón, tal como se muestra en la siguiente escena.




Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
【g1】
【g2】
【g1.x】
【g1.y】
【g2.x】
【g2.y】
【b7】
info
id g1
espacio E1
dibujar si
activo si
expresión (1,0)
tamaño 8
color
color interior
imagen evaluacion.png
constricción y=x
Aceptar
Cerrar
Aplicar

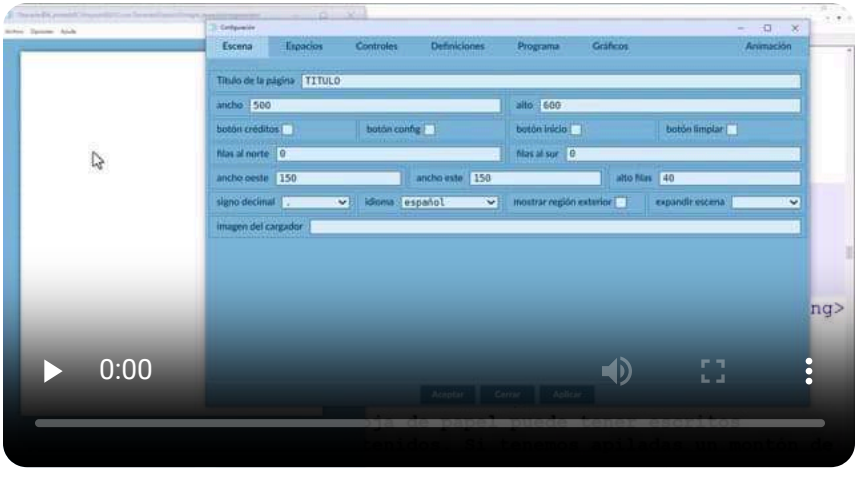

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
【g1】
【g2】
【g1.x】
【g1.y】
【g2.x】
【g2.y】
【b7】
info
id b7
nombre Reiniciar
interfaz botón
región interior
espacio E1
dibujar si
activo si
expresión (490,10,100,80)
valor 0
color texto
borde texto
color interior
fuente SansSerif
tam fuente 19
negrita
cursiva
subrayada
pos texto abajo-centro
imagen evaluacion.png
pos imagen arriba-centro
acción inicio
parámetro
estilo extra
tooltip
Aceptar
Cerrar
Aplicar

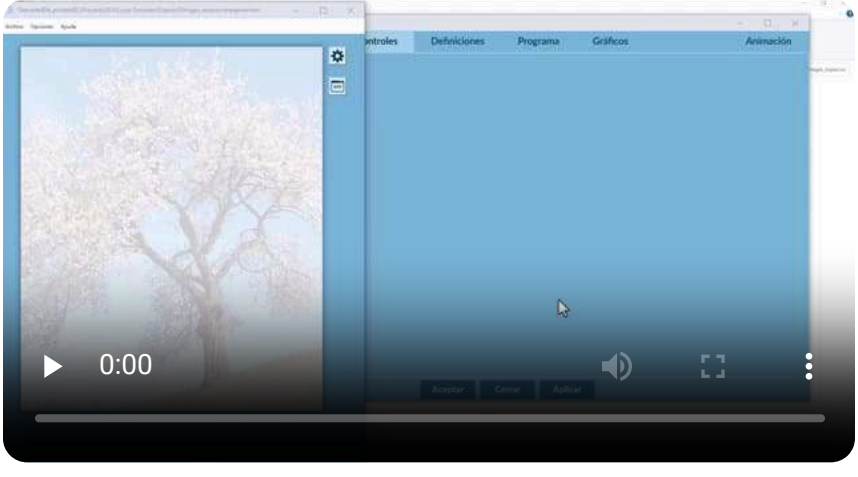
Tal como hemos comentado, las imágenes incluidas en escenas de Descartes pueden provenir de un enlace web. Dicho enlace debe incluir el nombre del archivo de la imagen y una extensión reconocida por Descartes.





A continuación, incluimos varios vídeos ilustrativos sobre el uso de imágenes y la configuración de sus parámetros en escenas.


Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Título de la página
TITULO
ancho
500
alto
600
botón créditos
botón config
botón inicio
botón limpiar
filas al norte
0
filas al sur
0
ancho oeste
150
ancho este
150
alto filas
40
signo decimal
idioma
español
mostrar región exterior
expandir escena
imagen del cargador
0:00


Definiciones
Programa
Gráficos
Animación
0:00

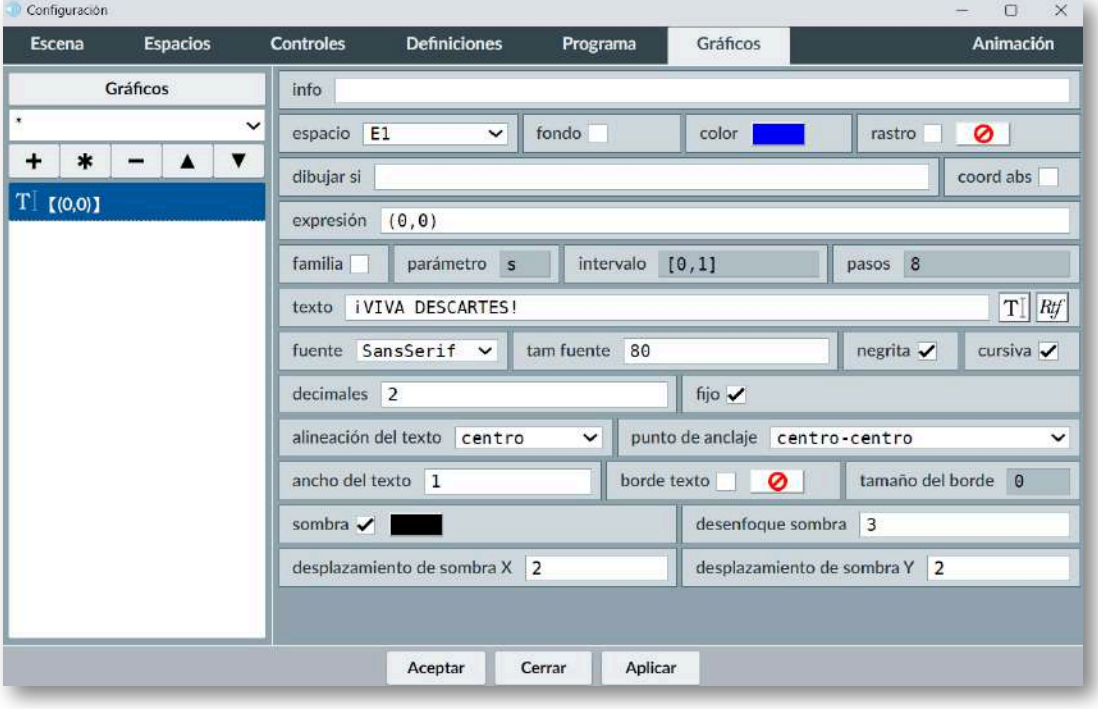

Selector gráficos
En este selector y en el espacio imagen, agregamos una imagen llamada perro.jpg (archivo) con expresión (0,0,1,1)
Este espacio lo hemos dedicado para escribir textos
0:00

5.16 Gráfico Texto

En el capítulo 2 tratamos sobre la introducción de textos en escenas. Ahora abordaremos algunos aspectos nuevos y repasaremos otros ya vistos.

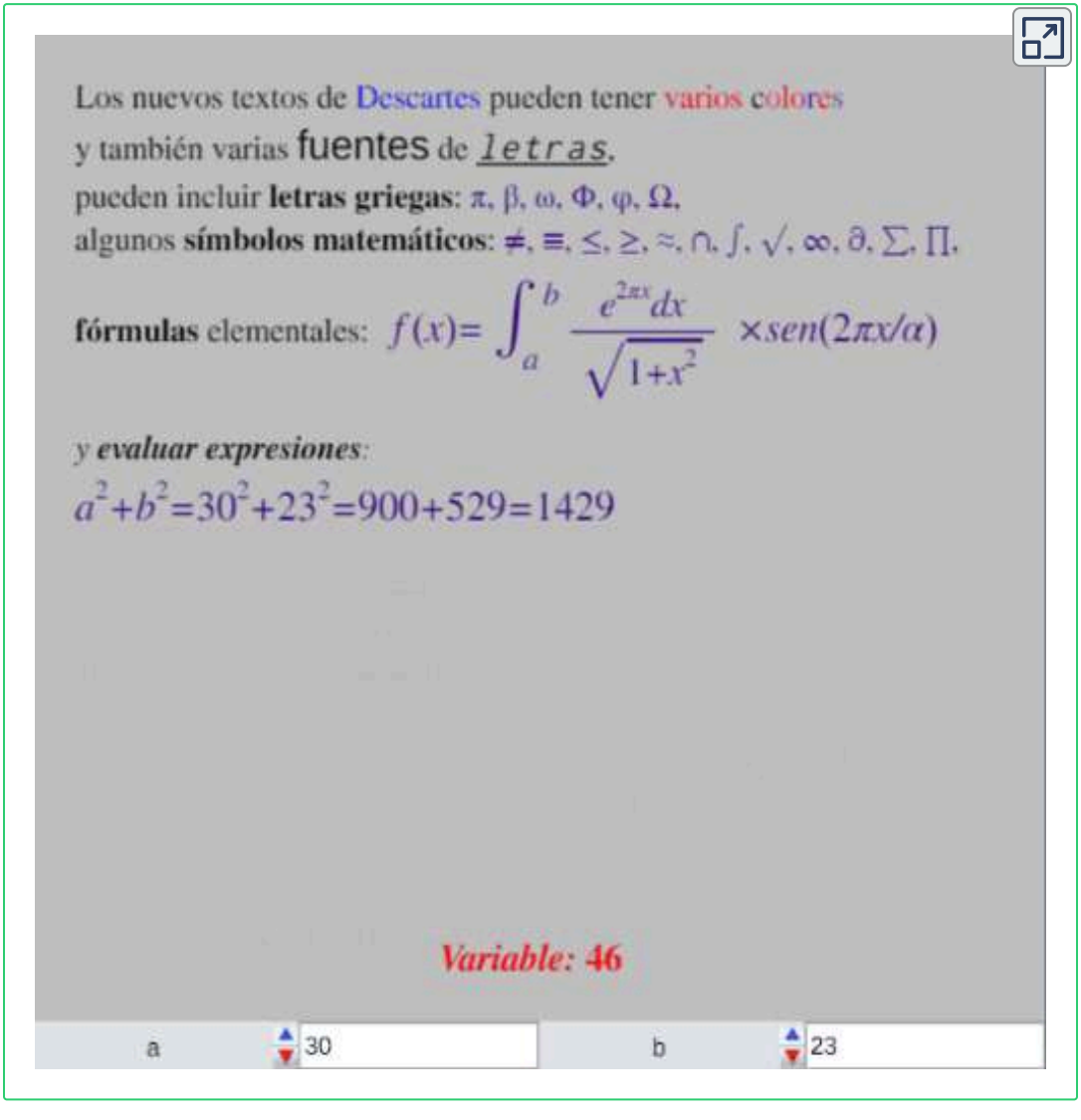
Este gráfico consiste en un texto que se puede introducir para explicar aspectos particulares de una lección. Por ejemplo, se puede usar para incluir instrucciones, explicaciones de fenómenos, preguntas, etc. En ocasiones, también puede emplearse temporalmente para mostrar datos al programador, ayudando a depurar la escena.

Como se muestra en el siguiente ejemplo, sin necesidad de utilizar un editor complejo, es posible definir texto en la escena con numerosos atributos, tales como tipo de letra, tamaño, color, negrita, cursiva, sombra, alineación, entre otros.

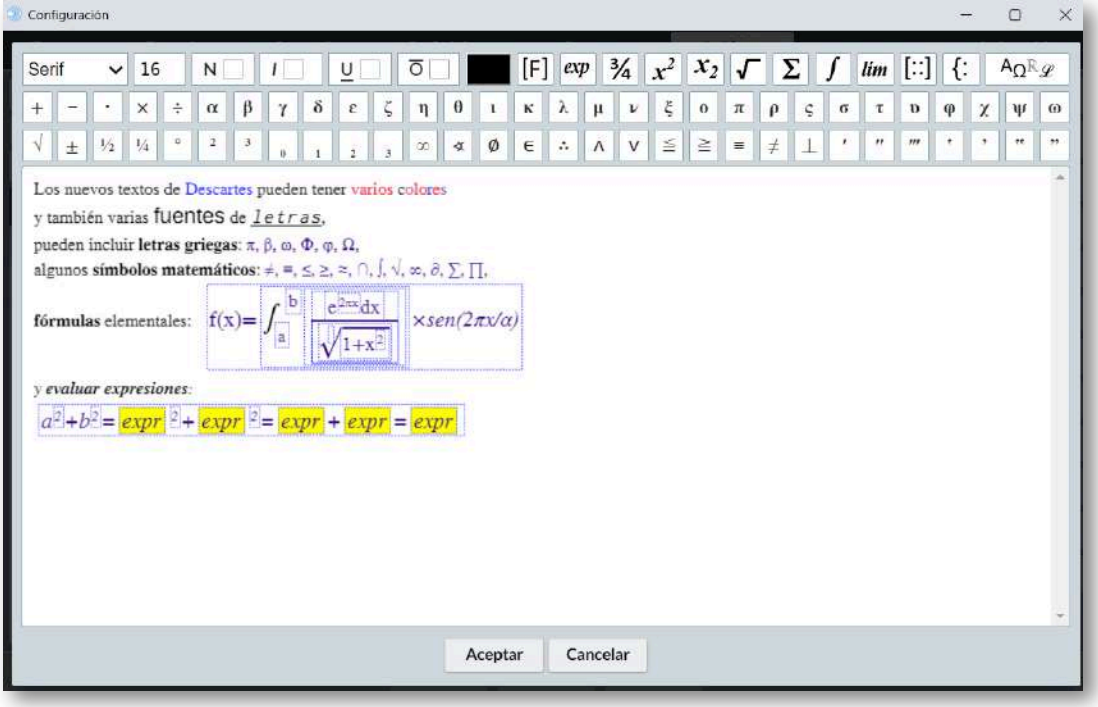




En la siguiente escena vemos textos con formatos más elaborados.



Se puede observar que el editor de textos utilizado es mucho más completo y versátil.



Recordamos y desarrollamos los principales campos de configuración del elemento tipo texto:

- **Expresión:** campo de texto donde se introduce el par de coordenadas de anclaje del texto, ya sea entre corchetes [ ] o entre paréntesis ( ). Mediante el menú **Punto de anclaje** es posible definir qué parte del texto quedará fijada a dichas coordenadas (por ejemplo, el centro, la esquina inferior derecha, etc.).
- **Alineación del texto:** menú aplicable únicamente al texto simple (no enriquecido). Cuando el contenido supera el ancho en píxeles definido en el parámetro **Ancho**, el texto se divide en varias líneas. Este menú permite decidir si dichas líneas se alinean a la izquierda, al centro o a la derecha.

- **Punto de anclaje:** determina la relación entre las coordenadas indicadas en **Expresión** y la caja invisible que contiene el texto. Por ejemplo, la opción arriba-derecha hace que la esquina superior derecha de esa caja coincida con las coordenadas especificadas.
- **Ancho del texto:** campo donde se define el ancho máximo en píxeles. En texto simple, este valor limita la longitud de las líneas antes de generar un salto automático. En texto enriquecido, los saltos de línea deben introducirse manualmente y el parámetro **Ancho** debe fijarse en 1. Si el valor es menor de 20 el campo es ignorado.
- **Borde texto:** casilla que añade un contorno al texto con el color seleccionado en el control asociado. Si está desactivada, el parámetro **Tamaño del borde** queda inhabilitado.
- **Tamaño del borde:** campo donde se especifica el grosor del borde en píxeles. Si se mantiene el valor 0 pero la opción **Borde texto** está activada, Descartes asignará automáticamente un grosor por defecto.
- **Sombra:** casilla que permite añadir una sombra al texto en el que no se ha usado el editor enriquecido, acompañada de un control para definir su color. Al activarse, se habilitan los siguientes parámetros:
    - **Desenfoque sombra:** valor numérico que controla la nitidez de la sombra. Un valor 0 produce bordes definidos; valores mayores incrementan el difuminado.
    - **Desplazamiento de sombra X:** indica, en píxeles, la distancia horizontal hacia la derecha a la que se proyecta la sombra.
    - **Desplazamiento de sombra Y:** indica, en píxeles, la distancia vertical hacia abajo a la que se proyecta la sombra.

El editor simple

Al presionar el botón **T** adyacente a un campo de tipo texto en un gráfico, se abre una ventana que permite introducir varias líneas de forma más cómoda.



En esta ventana es posible elegir entre tres fuentes distintas: **SansSerif**, **Serif** y **Monospaced** (monoespacio). La fuente seleccionada se aplica a todo el texto, ya que en el modo de texto simple no es posible combinar distintas fuentes.

También se puede seleccionar el tamaño de la fuente mediante un menú desplegable, activar **negrita** o **cursiva** mediante sus respectivos checkboxes, e insertar caracteres especiales a través de un botón con símbolos. Este botón abre una ventana adicional con múltiples caracteres organizados por conjuntos. En el panel izquierdo se puede navegar entre categorías, lo cual resulta especialmente útil para insertar operadores matemáticos y letras griegas.



Los saltos de línea se introducen simplemente presionando **INTRO**. Al aceptar los cambios, el texto se traslada al campo original junto al botón T, apareciendo en una sola línea. Los saltos de línea se representan mediante **\n**. También es posible escribir directamente **\n** en el campo de texto para forzar un salto.

Aunque el texto simple no permite un formato completamente enriquecido, admite cierto nivel de edición parcial mediante comandos similares a los de LaTeX. Estas opciones permiten cambiar color, aplicar negrita o cursiva a partes del texto, introducir subíndices y superíndices, y modificar alineaciones específicas.

Comentamos los principales aspectos del formato parcial del texto:

- **Cambio de color: \color{HEX}{texto}**
  El primer bloque indica el color en hexadecimal y el segundo el texto afectado.
  Ejemplo: **Hola, \color{ff0000}¿{cómo} estás?**

- **Negritas: \b{texto}**
  Ejemplo: **Descartes\b{JS}**
- **Cursivas: \i{texto}**
- **Lenguaje matemático:** Cuando se quiere incluir lenguaje matemático, es necesario flanquearlo entre las llaves de **\${ }**.
  - **Superíndices:** usar **^** dentro del modo matemático. Ejemplo: **\${A^{BC}}**
  - **Subíndices:** usar _ dentro del modo matemático. Ejemplo: **\${A_{BC}}**
  - **Fracciones: \frac{numerador}{denominador}**
    Ejemplo: **\${\frac{e^x}{b_2}}**
  - **Raíces: \sqrt{índice}{radicando}**
    Ejemplos:
    \${\sqrt{3}{sin(a)}}
    \${\sqrt{}{sin(a)}}
  - **Integrales: \int{inferior}{superior}{expresión}**
    Ejemplo: **\${\int{a}{b}{x^2 dx}}**
  - **Sumas: \sum{inicio}{fin}{expresión}**
    Ejemplo: **\${\sum{i=1}{n}{i^2}}**
  - **Productos: \prod{inicio}{fin}{expresión}**
    Ejemplo: **\${\prod{i=1}{n}{A_i}}**
- **Alineación centrada:** Basta incluir el texto a centrarse entre las llaves de **\c{}**. Si el parámetro ancho del texto del gráfico tipo texto se encuentra con valor 1 (es decir, que no se le pone límite horizontal al texto), entonces el texto a centrar se centrará respecto al renglón más largo presente. Si dicho parámetro tiene otro valor (por ejemplo, 500), el texto quedará centrado respecto a un renglón imaginario de 500 px de ancho. Conviene estudiar el apartado sobre ancho del texto para recordar cómo funciona.

- **Alineación a la derecha:** Su funcionamiento es análogo al del centrado de texto, sólo que alinea el texto incluido entre las llaves al margen derecho. La expresión a usar es **\r{}**. Es decir, hay que cambiar la c (de centrado) por r (de right o derecha).
- **Alineación a la izquierda:** El texto entre llaves de **\l{}** quedará alineado con el margen izquierdo del bloque de texto. La letra usada es l (por left o izquierda).
- **Alineación justificada:** El texto entre llaves de **\j{}** quedará justificado respecto al parámetro ancho de texto del gráfico tipo texto.

Las distintas instrucciones de edición pueden **anidarse**, es decir, combinarse unas dentro de otras para aplicar varios formatos simultáneamente a diferentes partes del texto.

Por ejemplo, si se desea mostrar el texto **Hola** en color azul, pero con la letra **o** en negritas, se debe introducir la siguiente expresión:

\color{0000FF}{H\b{o}la}

En este caso:

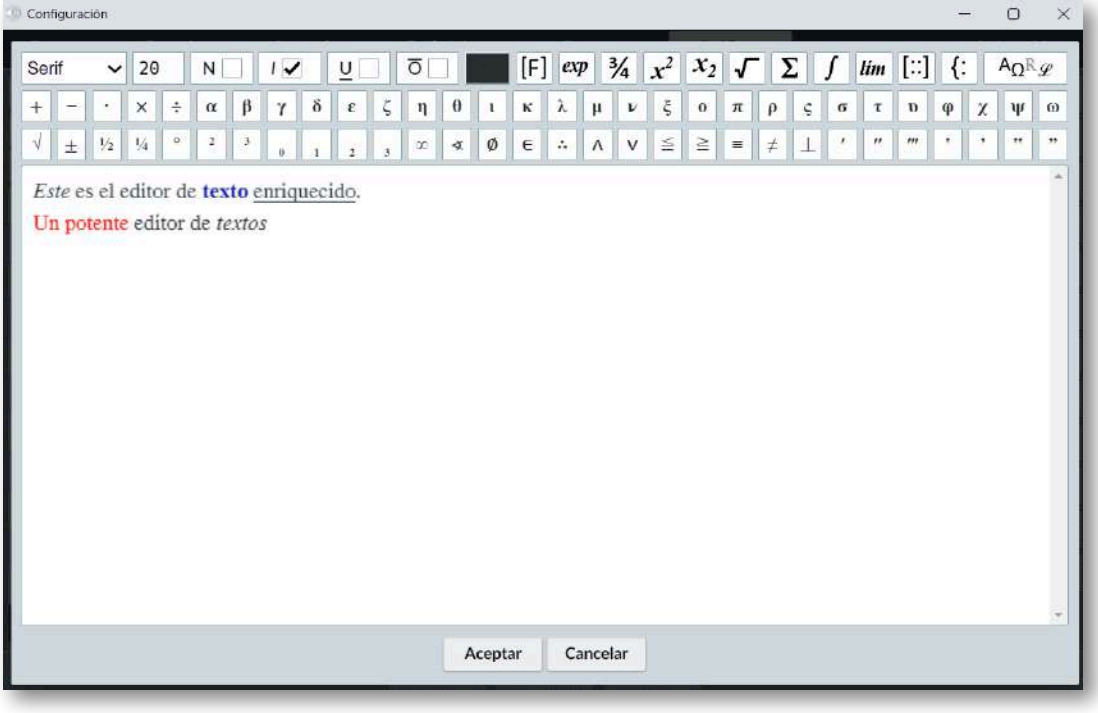
- **\color{0000FF}{...}** aplica el color azul a todo el contenido incluido entre llaves.
- **\b{o}** aplica negrita únicamente a la letra **o**, dentro del bloque azul.

El resultado final será la palabra **Hola** en azul, con la **o** resaltada en negritas.

Es preciso notar que el texto enriquecido abordado es útil únicamente para realizar pequeños cambios de edición. Para una edición verdaderamente completa de texto enriquecido, conviene estudiar el apartado que se presenta a continuación.

El editor enriquecido

Si se presiona el botón **Rtf** adyacente a un campo de introducción de texto, la ventana que se muestra para introducir texto es diferente. Esta ventana cuenta con numerosos botones en la parte superior, los cuales permiten aplicar distintos tipos de formato al texto, así como insertar diversos símbolos matemáticos.



En esta ventana se puede introducir texto de manera similar a la ventana de texto simple. Sin embargo, es posible seleccionar parte del texto y aplicarle cambios de formato únicamente a esa sección. Se pueden usar el menú para el tipo de fuente, el menú para el tamaño y los checkboxes de cursiva y negrita para modificar la selección. Cuando se coloca el cursor al final del texto y se continúa escribiendo, el formato del nuevo texto puede diferir del texto previo. Para evitar esto, es recomendable preparar primero el texto en la ventana de texto simple y luego editarlo en la ventana de texto enriquecido.

Además de los controles básicos, la ventana incluye funciones más avanzadas:

- **Botón Subrayar:** coloca una línea bajo el texto seleccionado.
- **Botón Sobrerayar:** coloca una línea sobre el texto seleccionado.
- **Botón Color:** muestra el color actual del texto y abre la herramienta de edición de color al presionarlo. Permite utilizar distintos colores para distintos elementos.
- **Botón [F]:** inserta una caja de fórmula para escribir texto en lenguaje matemático. Atajo: **CTRL + F**.
- **Botón Expresión:** permite insertar expresiones evaluables con variables o constantes. Atajo: **CTRL + E**.
- **Botón Fracción:** inserta fracciones editables. Atajo: **CTRL + /** o **CTRL + F** dentro de un cuadro de fórmula.
- **Botón Potencia:** agrega base y exponente editable. Atajo: **CTRL + ↑**.
- **Botón Subíndice:** agrega subíndice editable. Atajo: **CTRL + ↓**.
- **Botón Raíz:** inserta un radical editable. Atajo: **CTRL + R**.
- **Botón Suma:** inserta un símbolo de sumatoria con límites editables. Atajo: **CTRL + S**.
- **Botón Integral:** inserta un símbolo de integral con límites editables.
- **Botón Límite:** agrega un símbolo de límite con argumentos editables. Atajo: **CTRL + L**.
- **Botón Matriz:** permite crear y editar matrices, con opción a corchetes, paréntesis o llaves, definiendo filas y columnas.
- **Botón Definición por partes:** inserta una función definida por partes con un número determinado de entradas.
- **Botón Tabla:** abre una ventana de caracteres especiales similar a la del texto simple.

Cuando se inserta una forma de texto enriquecido, esta aparece con marcadores de posición. Por ejemplo, una integral tiene tres: el límite inferior, el superior y el del integrando. El cursor puede colocarse en cualquiera de estos marcadores con el mouse o navegando con las teclas de flechas del teclado.

El control de tamaño en la ventana de introducción de texto enriquecido es un campo de texto en lugar de un menú, y permite elegir tamaños entre 8 y 200.

Se incluyen dos renglones para introducir una gran variedad de caracteres matemáticos y letras griegas. Para insertar letras griegas en mayúscula, se mantiene presionada la tecla SHIFT mientras se pulsa la letra deseada.

Al agregar un símbolo dentro de un cuadro de fórmula, es posible insertar texto matemático a la izquierda del mismo. Los símbolos pueden eliminarse pulsando la tecla de retroceso con el cursor a su derecha. Es importante asegurarse de estar dentro del cuadro de fórmula para borrar únicamente el carácter deseado. Para añadir una línea nueva después de una fórmula, se debe mover el cursor fuera del cuadro de fórmula y presionar INTRO.

Se pueden combinar expresiones para crear fórmulas complejas, como fracciones dentro de fracciones, o radicales dentro de fracciones que a su vez tienen radicales. Los tamaños de los textos se ajustan automáticamente según el nivel jerárquico dentro de la fórmula. Una mejora reciente permite aplicar distintos estilos (colores, fuentes, etc.) dentro de una misma caja de fórmula.

Al pulsar el botón Aceptar, la ventana se cierra y el editor muestra el texto editado. El campo de texto principal contiene caracteres codificados que permiten que el texto enriquecido se visualice correctamente al presionar Aplicar.

El teclado virtual

Aunque el teclado virtual sirve para introducir texto, su configuración se realiza desde los elementos de tipo **Control**, ya que permite al usuario ingresar texto en los campos de estos controles.

Para evitar que el teclado nativo de dispositivos móviles interfiera al pulsar un campo de texto en un control, se puede usar el teclado virtual de DescartesJS. Los teclados nativos a veces ocultan parte del interactivo o reducen su tamaño para mostrarse completos, afectando la escala de la escena.

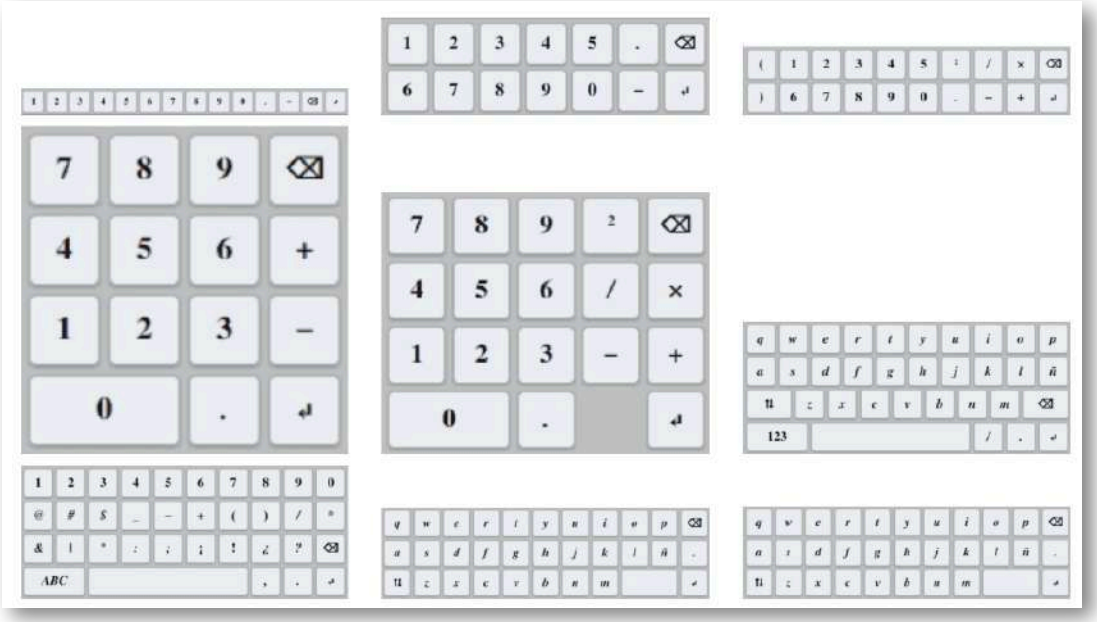
El teclado virtual aparece dentro del interactivo, en la posición que el usuario indique, y solo se activa si está marcada la casilla **teclado** en el campo de texto del control correspondiente.

El tipo de teclado a seleccionar depende del uso del campo de texto:

- Si se requiere que el usuario ingrese únicamente números, se puede usar un teclado 7 × 2 o 14 × 1.
- Para operaciones básicas de suma y resta, el teclado 4 × 4 es suficiente.
- Si se desea permitir más operaciones, se puede usar un teclado 5 × 4.
- Para ingresar texto como nombres, puede usarse un teclado 10 × 4_al f a.

De esta forma, el usuario dispone de un teclado adaptado al tamaño y funcionalidad de la escena, evitando que el teclado nativo afecte la interacción.

Al pulsar una tecla en el teclado virtual, el carácter se introduce en la posición actual del cursor dentro del campo de texto. Para cerrar el teclado y aplicar los cambios al campo, se debe pulsar la tecla de retorno.



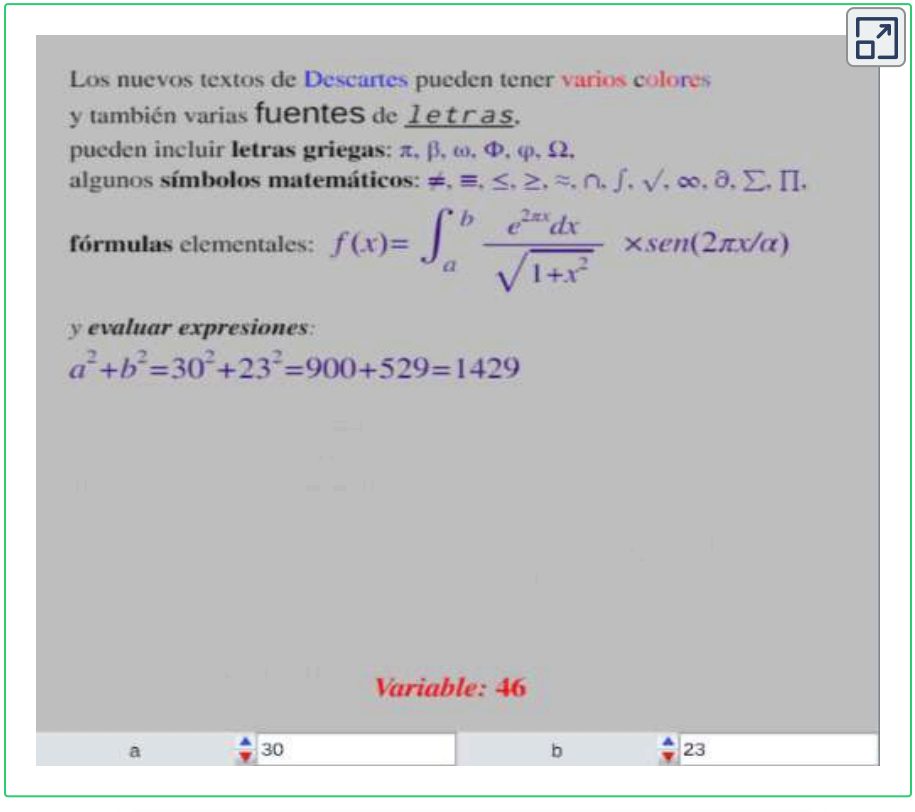
El teclado solo se muestra cuando el usuario pulsa en un campo de texto de un control que tenga activada la funcionalidad de teclado virtual. Al salir del campo de texto, el teclado desaparece. Mientras el teclado está activo, la escena se vuelve ligeramente más opaca, indicando que no se puede interactuar con otros elementos de la escena.

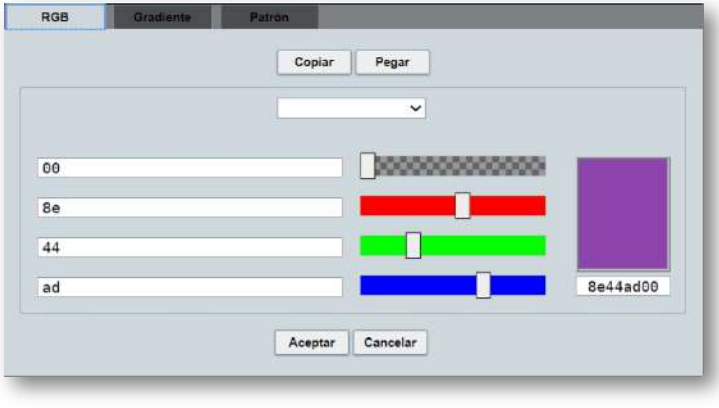
Algunos caracteres del teclado incluyen opciones adicionales no visibles directamente. Por ejemplo, si se pulsa y mantiene una letra que puede acentuarse, encima del teclado aparecerán teclas adicionales con las posibles variantes. De forma similar, al pulsar el punto decimal se muestran teclas con símbolos numéricos adicionales que no están visibles de forma predeterminada.

Para que estas funciones se desplieguen correctamente, conviene dejar un espacio libre sobre el teclado. Por ejemplo, establecer el parámetro de posición del teclado en **(100,50)** permite un colchón de 50 px para mostrar los símbolos adicionales sin que queden fuera del área visible.

Aclaraciones sobre configuración de colores.

Vamos a abordar algunos aspectos relacionados con el uso del color en Descartes, presente tanto en los textos como en la mayoría de los elementos gráficos. En la siguiente escena se muestra una simulación del panel de configuración de color de dichos elementos.





Algunos aspectos a tener en cuenta sobre la introducción de color en Descartes son los siguientes:

- El color puede definirse mediante el modelo **RGB** junto con un valor de **transparencia**, tal y como se muestra en la escena anterior y en la imagen anterior.
- Si se introduce manualmente en un campo de texto, puede expresarse en **formato hexadecimal (base 16)** o en **formato decimal**.
- **Formato hexadecimal:** cada componente (rojo, verde, azul y transparencia) admite 256 valores posibles, desde **00** (mínimo) hasta **ff** (máximo). Las letras pueden escribirse en mayúsculas o minúsculas. Al introducir un valor hexadecimal y pulsar INTRO, las barras de color y el cuadro de vista previa se actualizan automáticamente. Del mismo modo, al mover las barras, el valor hexadecimal se modifica en tiempo real.
- **Formato decimal:** cada componente toma valores entre **0** y **1**. Este formato resulta especialmente útil cuando el color depende de una variable, permitiendo efectos dinámicos. Sin embargo, en este caso las barras y la vista previa no se actualizan automáticamente; el color se aplicará correctamente al pulsar **Aplicar**.
- Si se introducen valores decimales fuera del rango 0–1, el sistema no podrá interpretarlos y el color mostrado será negro.
- **Conversión entre formatos:** para convertir un valor hexadecimal a decimal, primero se transforma a base 10 (por ejemplo, 9a equivale a 154) y después se divide entre 256, obteniendo un valor comprendido entre 0 y 1 (en este caso, 0.6015625).
- El panel de vista previa del color refleja correctamente los cambios cuando se trabaja en formato hexadecimal; en formato decimal, la actualización visual puede no coincidir hasta aplicar los cambios.

5.17 Familias en gráficos 2D

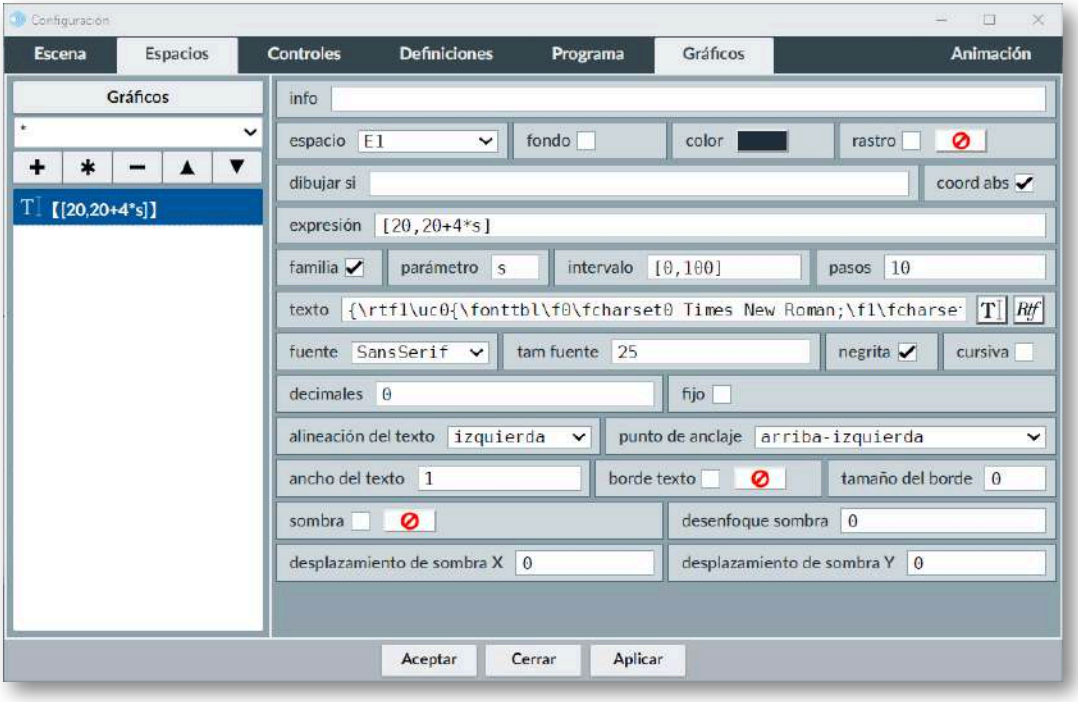
Los elementos gráficos 2D cuentan con la opción de activar el campo **Familia**. Su finalidad es que un mismo elemento se repita en la escena varias veces de una forma clónica, en las posiciones y con los valores deseados. Las opciones configurables para la familia de un elemento son:

- **Intervalo:** define el límite inferior y superior entre los que se establecerán los valores de repetición. El parámetro asociado a la familia tomará valores desde el límite inferior hasta el superior.
- **Pasos:** número de pasos uniformes que se darán para recorrer el intervalo desde el límite inferior hasta el superior.
- **Parámetro:** variable que tomará los diferentes valores de cada paso del recorrido. Se recomienda usar un nombre corto (por defecto, la letra **s**).

Por ejemplo, si queremos que en un intervalo de 0 a 100, en 11 pasos, el parámetro **s** tome los valores correspondientes, configuramos de la siguiente forma:

- **Intervalo:** [0,100]
- **Pasos:** 10 (lo que genera 11 valores uniformes desde 0 hasta 100)
- **Valores del parámetro s:** 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Supongamos que tenemos un texto que dice: "El valor del parámetro es: s". Para configurar el elemento texto con la familia, activamos el campo **Familia** con el intervalo [0,100], 10 pasos y el parámetro **s**. La posición del texto se define en función de **s** como **[20, 20 + 4*s]**, de modo que cada repetición se ubique en la posición correspondiente según el valor del parámetro.

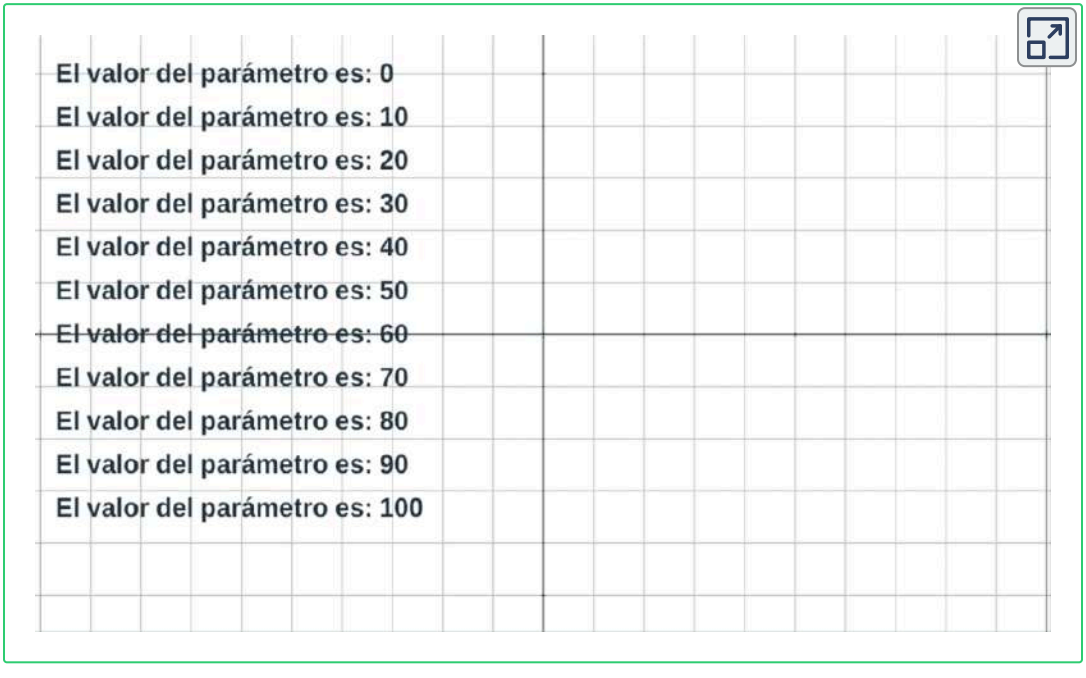


Con la configuración de familia aplicada al elemento texto, obtenemos un conjunto de 11 repeticiones, correspondientes a cada valor que adopta el parámetro **s** en el intervalo seleccionado. Cada repetición se ubica verticalmente según la expresión **20 + 4*s**, lo que produce las siguientes posiciones:

- Posición 20 → s = 0
- Posición 60 → s = 10
- Posición 100 → s = 20
- Posición 140 → s = 30
- Posición 180 → s = 40
- Posición 220 → s = 50
- Posición 260 → s = 60
- Posición 300 → s = 70
- Posición 340 → s = 80
- Posición 380 → s = 90
- Posición 420 → s = 100

Además, cada elemento de texto incluye dentro de su contenido el valor correspondiente del parámetro **s**, de modo que el texto refleja dinámicamente el valor que representa en esa posición. Esto permite visualizar claramente cómo varía el parámetro a lo largo del intervalo definido.

El resultado final es un conjunto de 11 elementos de texto alineados verticalmente, cada uno mostrando su valor de **s**, tal como se ilustra en la escena a continuación.



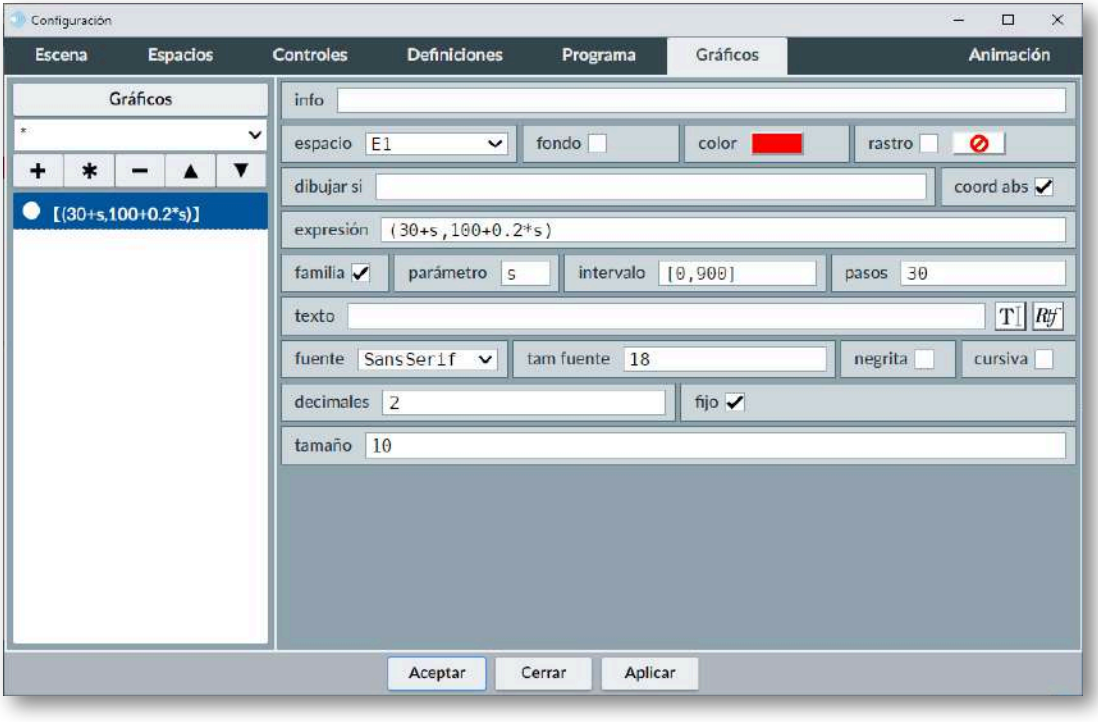
Consideremos un ejemplo sencillo: tenemos un elemento gráfico (2d) tipo punto de color rojo ubicado inicialmente en la posición absoluta **(30,100)** y cuyo de tamaño es de **10**.

Queremos que este punto se repita varias veces utilizando la opción de **familia**. Además, queremos establecer donde se colocarán los puntos clonados respecto al "original".

Configuramos la familia del siguiente modo:

- **Intervalo:** [0, 900] → el parámetro recorrerá valores desde 0 hasta 900.
- **Pasos:** 30 → se generarán 30 pasos uniformes dentro del intervalo.
- **Parámetro:** variable que tomará cada valor del recorrido (por defecto **s**).
- **Localización:** se define en función del parámetro **s** con la expresión que determina la posición del punto en cada repetición.

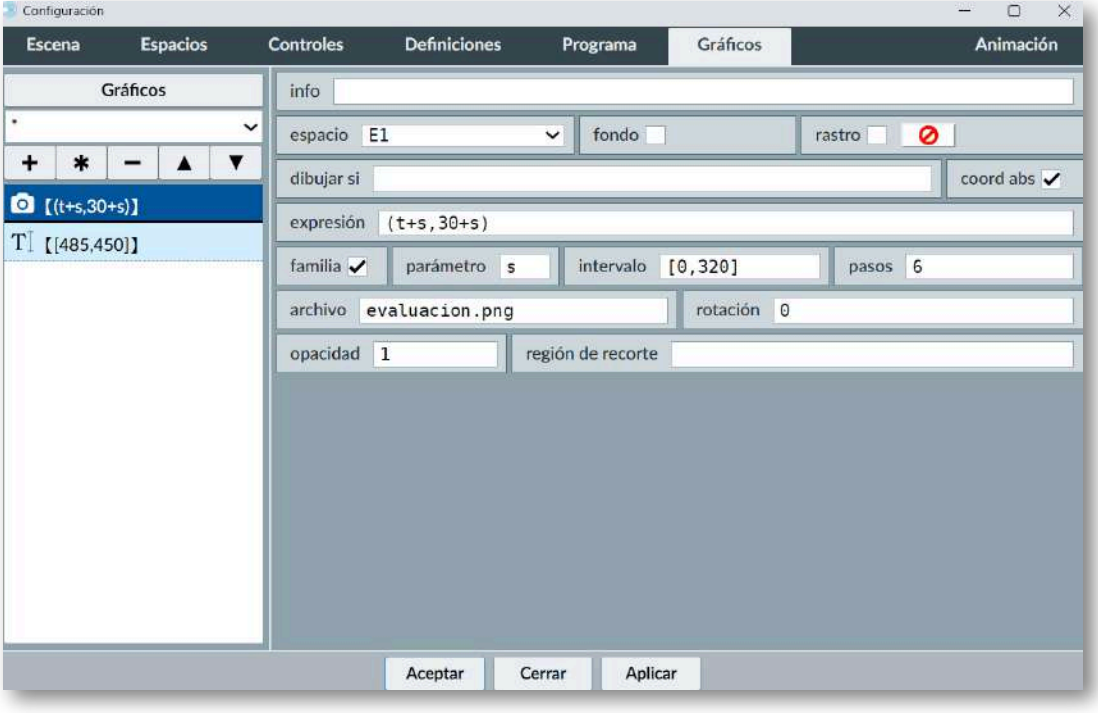
De esta manera, el punto rojo se repetirá 31 veces (incluyendo los extremos) a lo largo del intervalo, generando un patrón uniforme y escalonado según la función de posición definida.



La escena quedará de la siguiente forma:

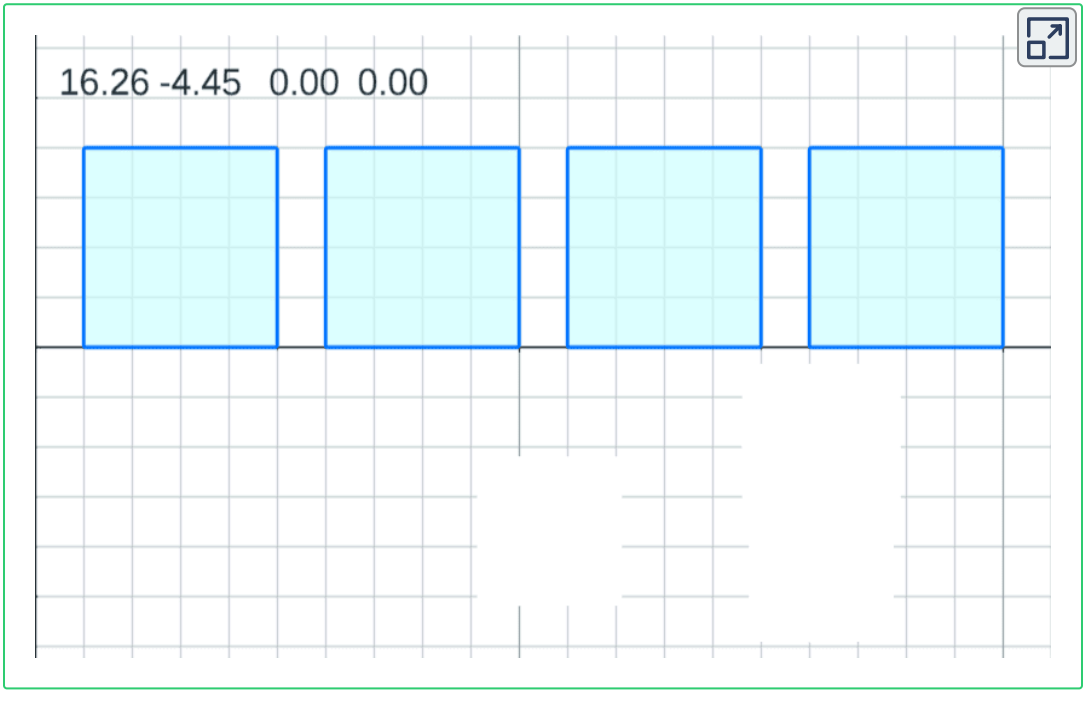


Ahora pondremos un ejemplo de una familia "animada":

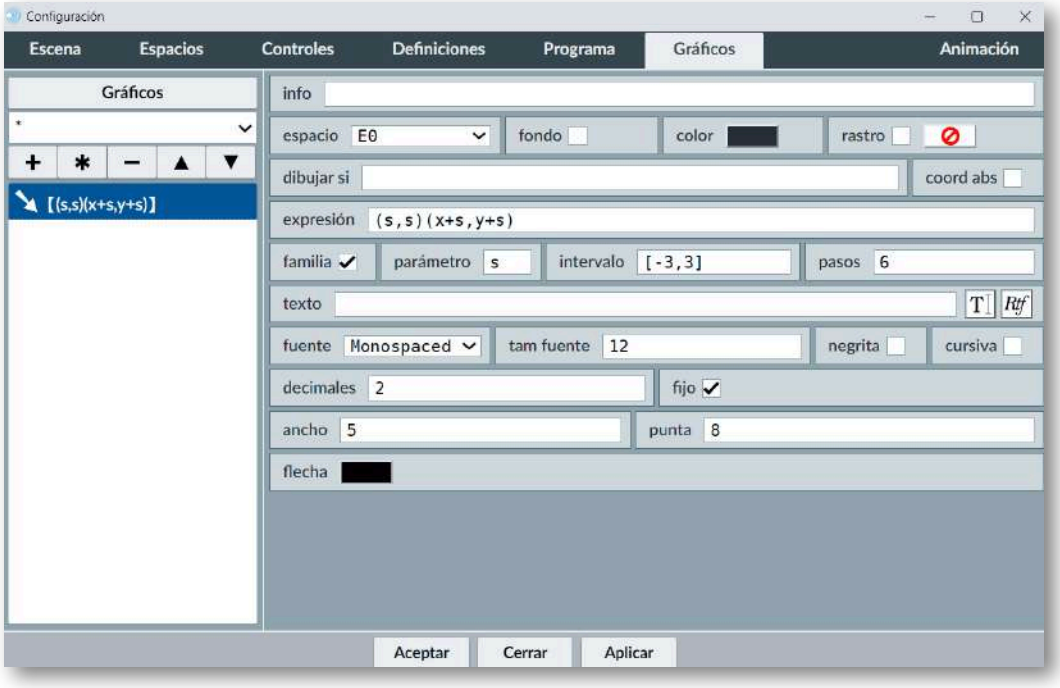


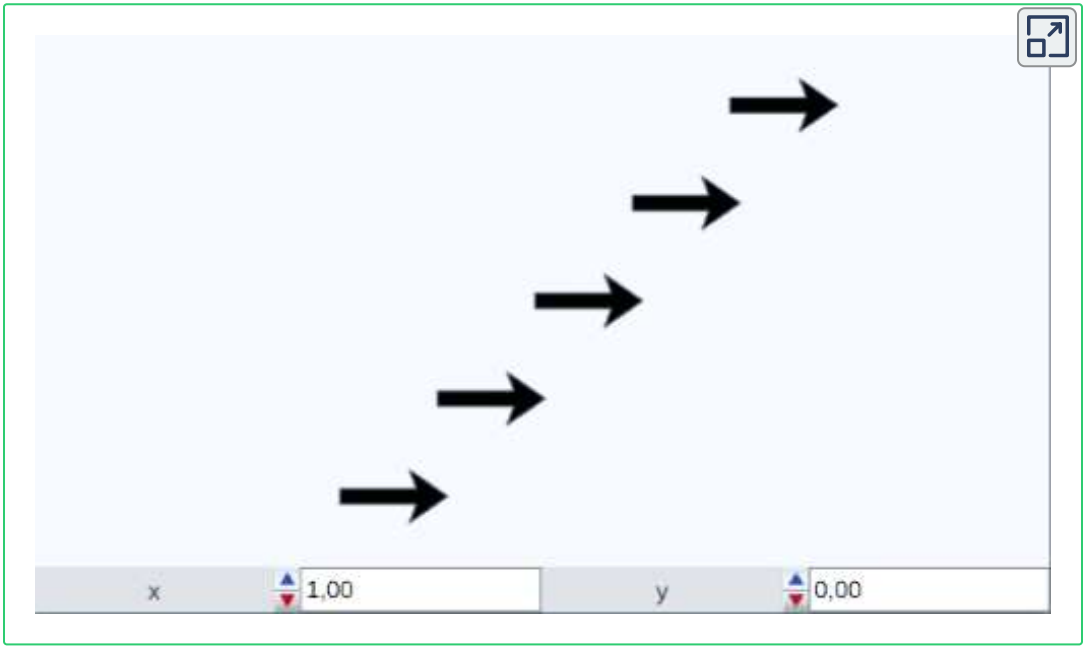


Mostramos a continuación ejemplos más elaborados de escenas con familias.

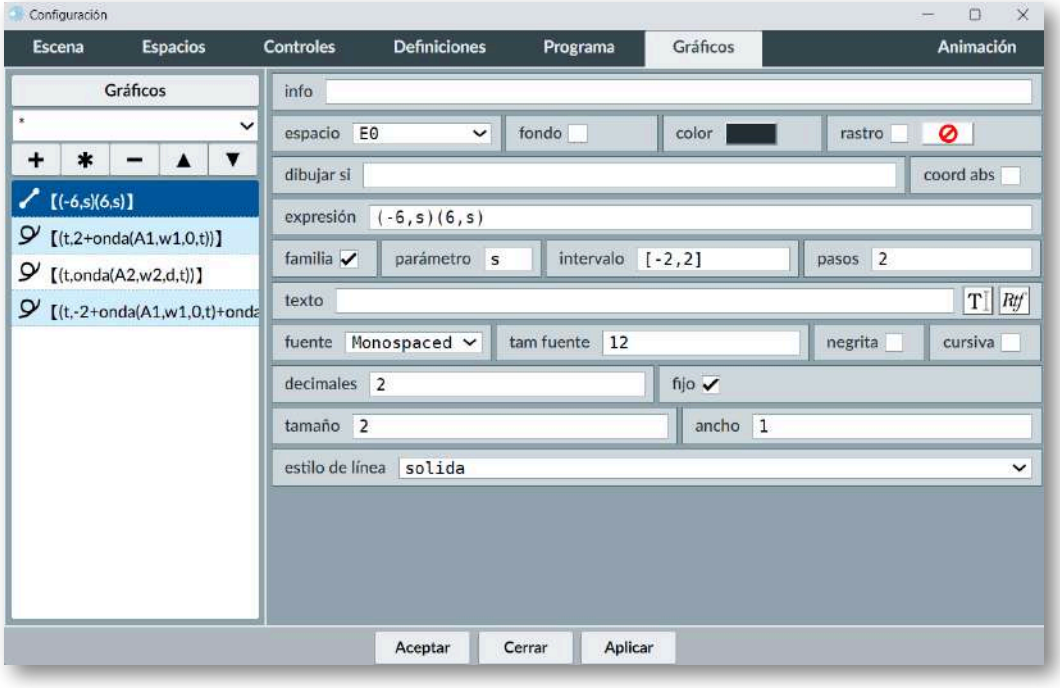


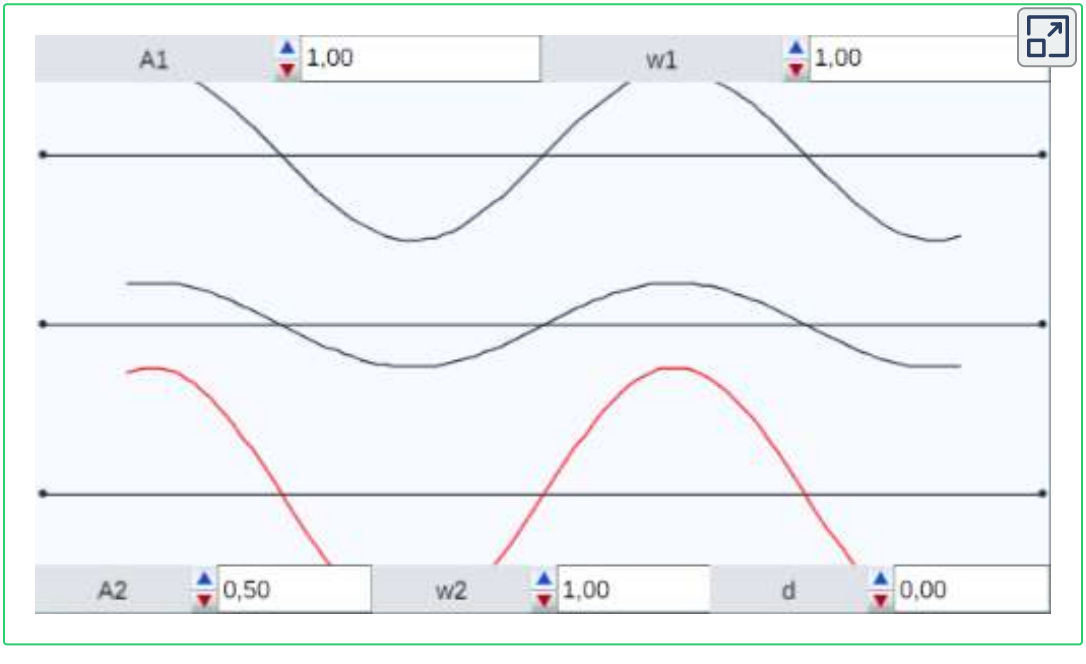
Incluimos un ejemplo de una familia de flechas cuyas puntas están dadas por pulsadores.

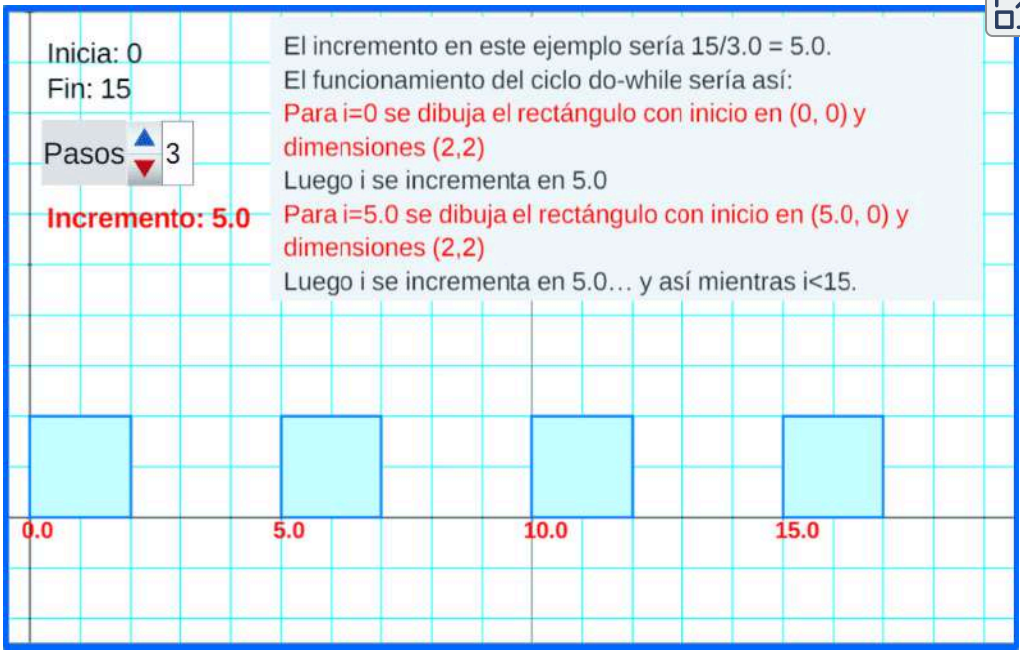


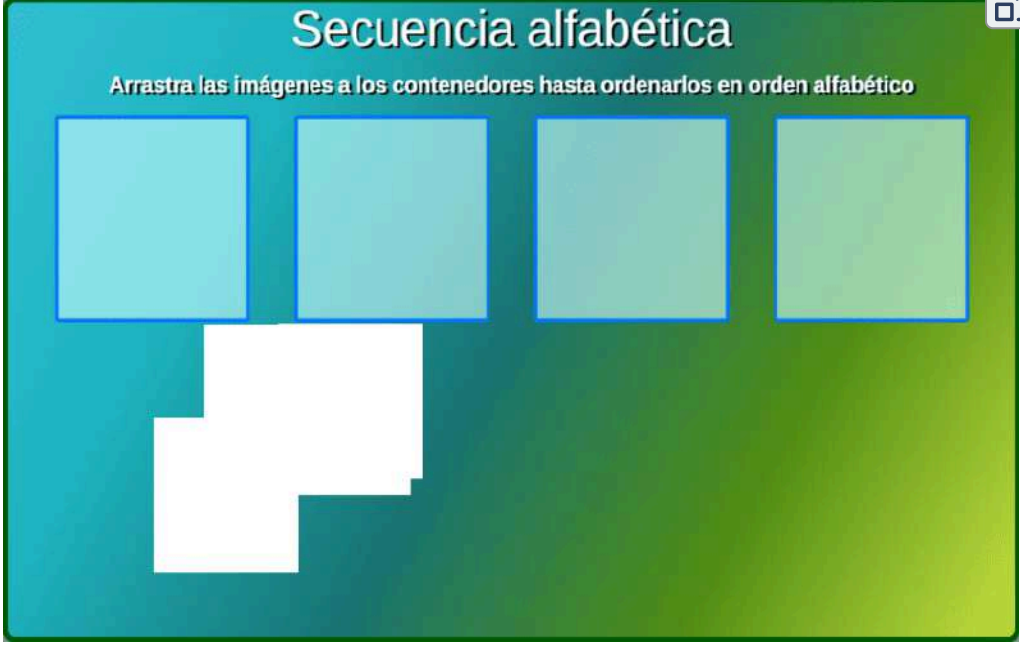


Mostramos otro ejemplo de una familia de segmentos sobre los que se dibujan curvas parametrizadas.



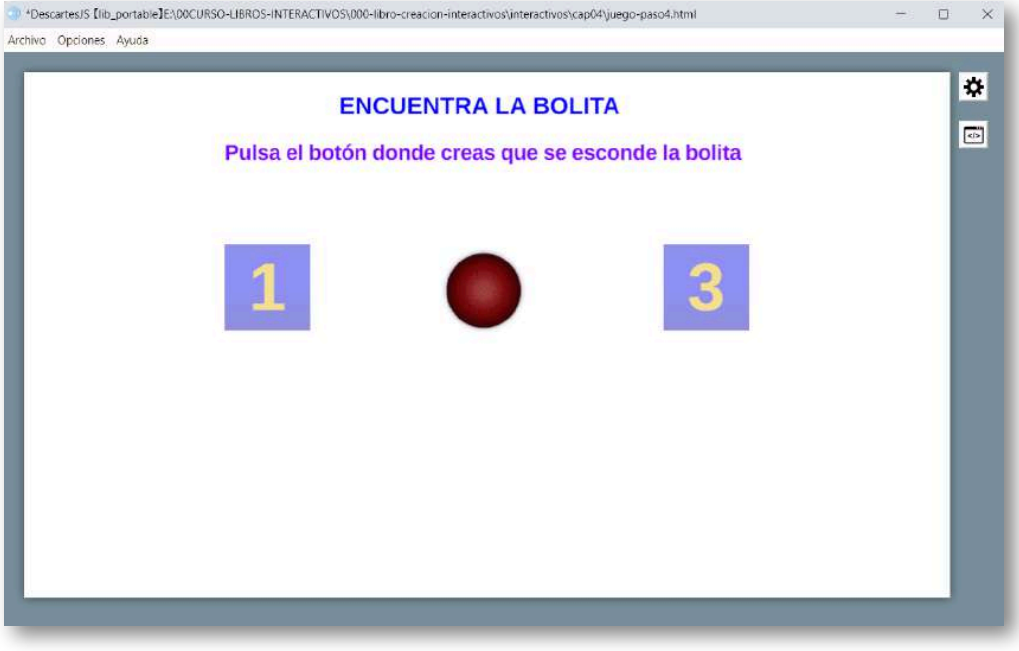



Inicia: 0
Fin: 15
Pasos 3
Incremento: 5.0
El incremento en este ejemplo sería 15/3.0 = 5.0.
El funcionamiento del ciclo do-while sería así:
Para i=0 se dibuja el rectángulo con inicio en (0, 0) y dimensiones (2,2)
Luego i se incrementa en 5.0
Para i=5.0 se dibuja el rectángulo con inicio en (5.0, 0) y dimensiones (2,2)
Luego i se incrementa en 5.0… y así mientras i<15.
0.0
5.0
10.0
15.0


Secuencia alfabética
Arrastra las imágenes a los contenedores hasta ordenarlos en orden alfabético

5.18 Mi primer juego paso a paso. Paso 5

Vamos a realizar el paso 5 del juego. En primer lugar, abrimos el juego con el editor de **Descartes**, accediendo al menú *Archivo* > *Abrir* y seleccionando el archivo llamado **juego-paso4**.

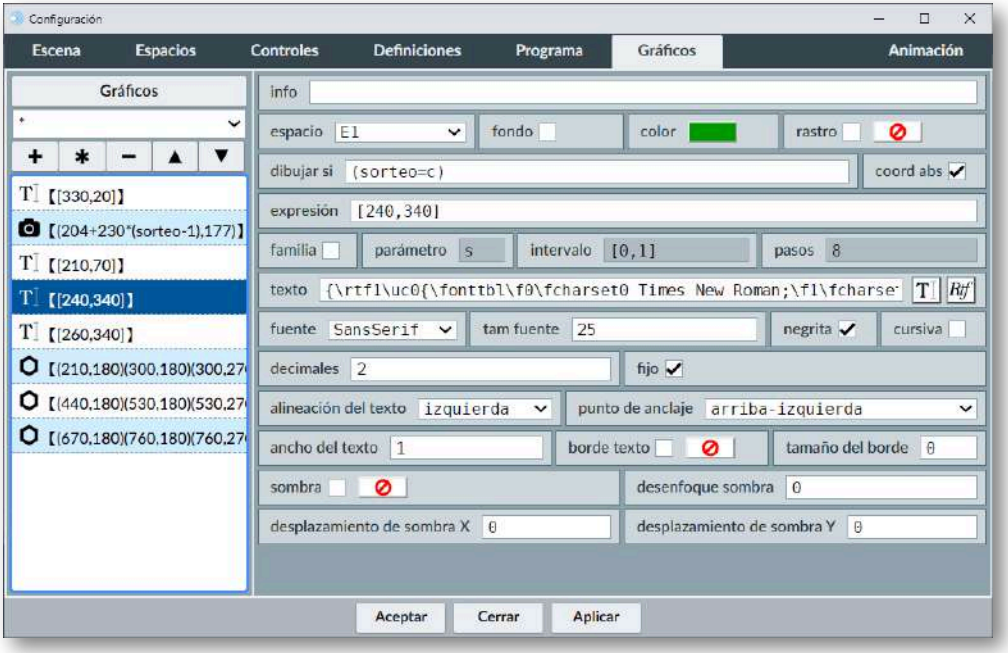


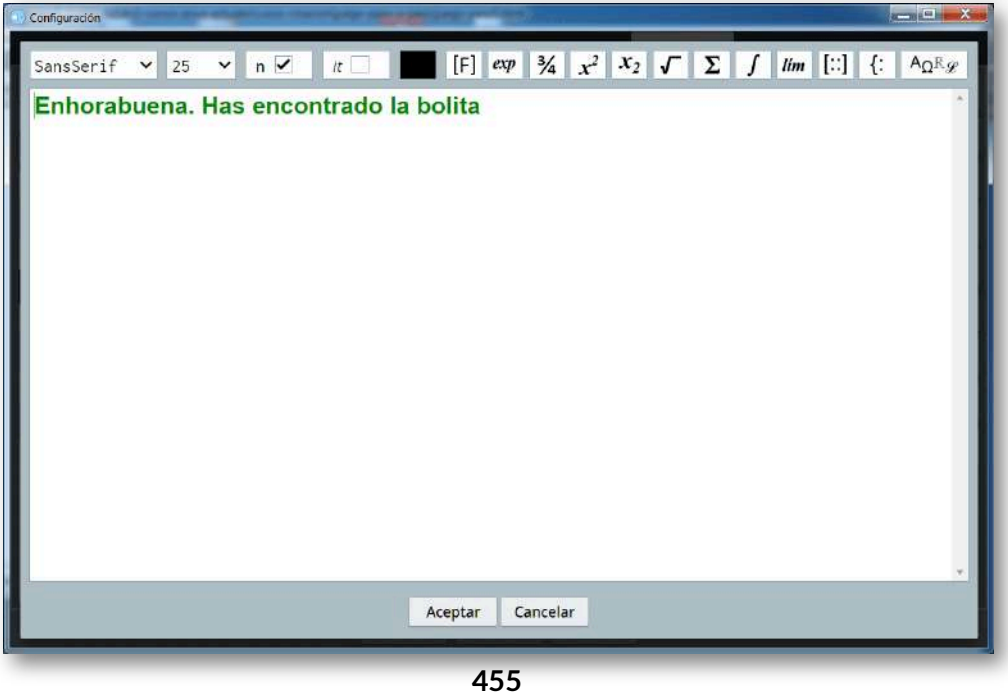
A continuación, vamos a agregar un elemento de **texto** mediante el menú **Gráficos** y el botón **+ Agregar texto**. Este texto será de color verde y se colocará justo debajo de la ubicación de la bolita, junto a los controles que se dibujarán cuando la bolita aparezca.

Configuraremos el texto con las siguientes características:

- **Condición "dibujar-si":** `(sorteo = c)` → el texto solo se mostrará cuando se cumpla esta condición.
- **Contenido del texto:** "Enhorabuena. Has encontrado la bolita".
- **Color:** verde, para resaltar el mensaje.
- **Posición:** debajo de la bolita, de modo que se visualice claramente al cumplirse la condición.

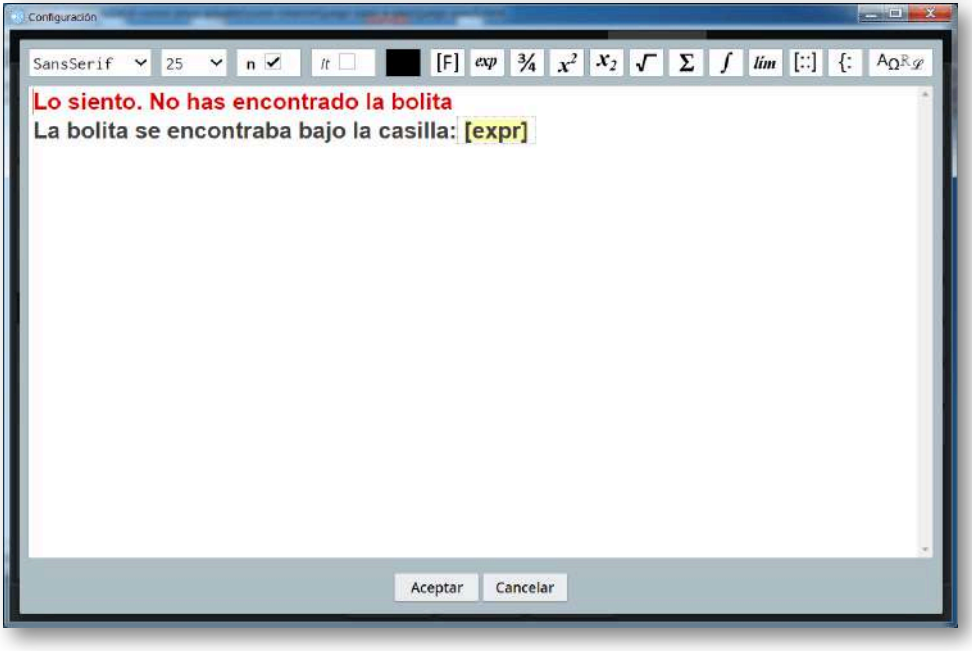
Con esta configuración, cada vez que la bolita coincida con el valor del sorteo, aparecerá el mensaje de felicitación en la escena, proporcionando retroalimentación visual inmediata al usuario.



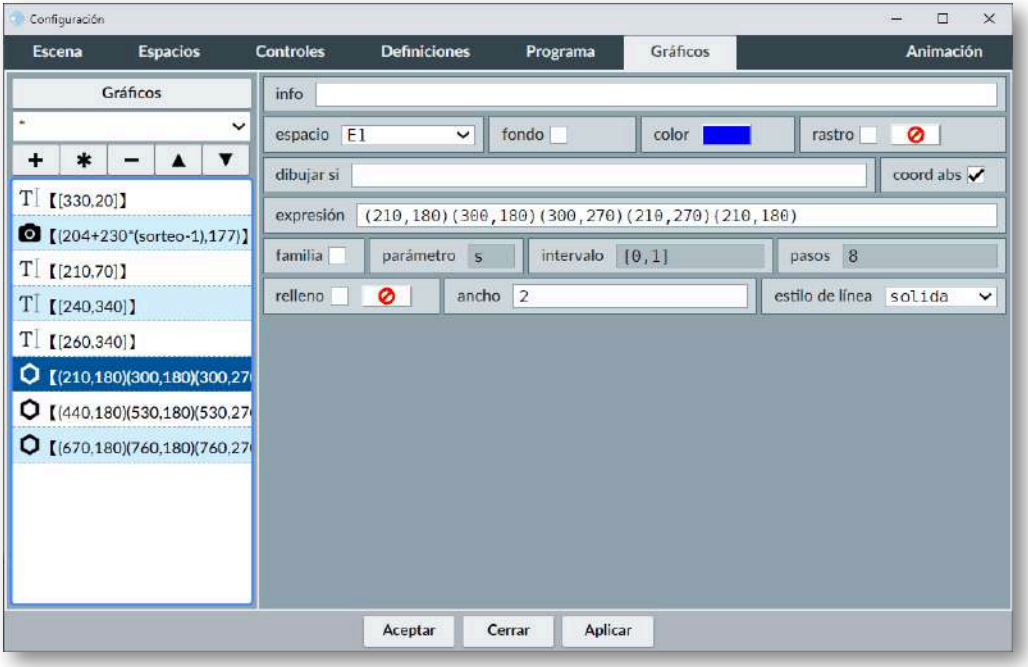


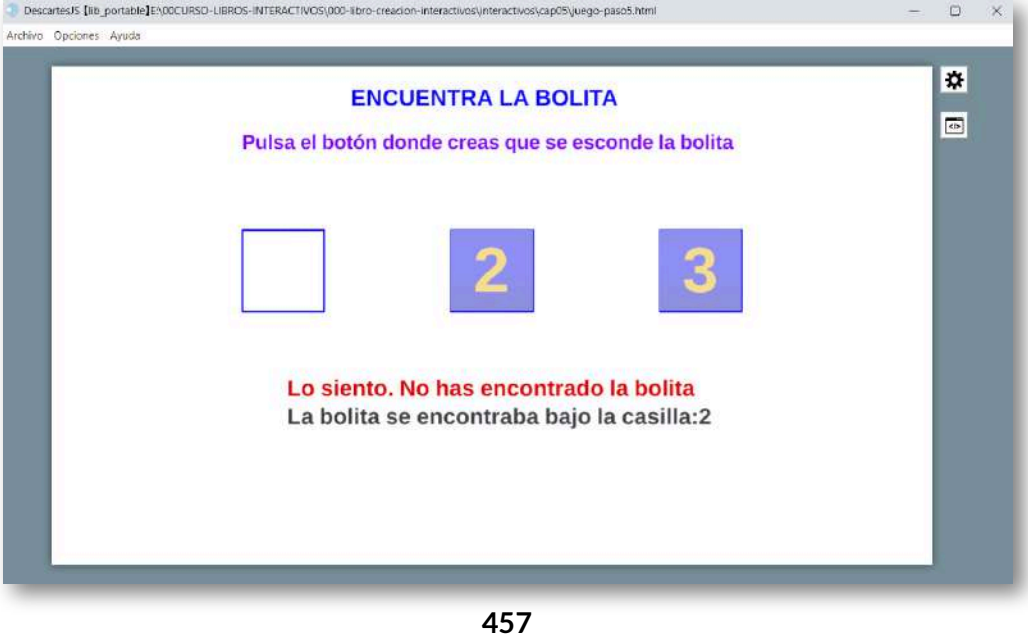
Agregamos un texto rojo que aparece si no se encuentra la bolita, con la condición (sorteo ≠ c) y c > 0, mostrando: "Lo siento. No has encontrado la bolita", seguido de la indicación de la solución: "La bolita se encontraba bajo la casilla:".



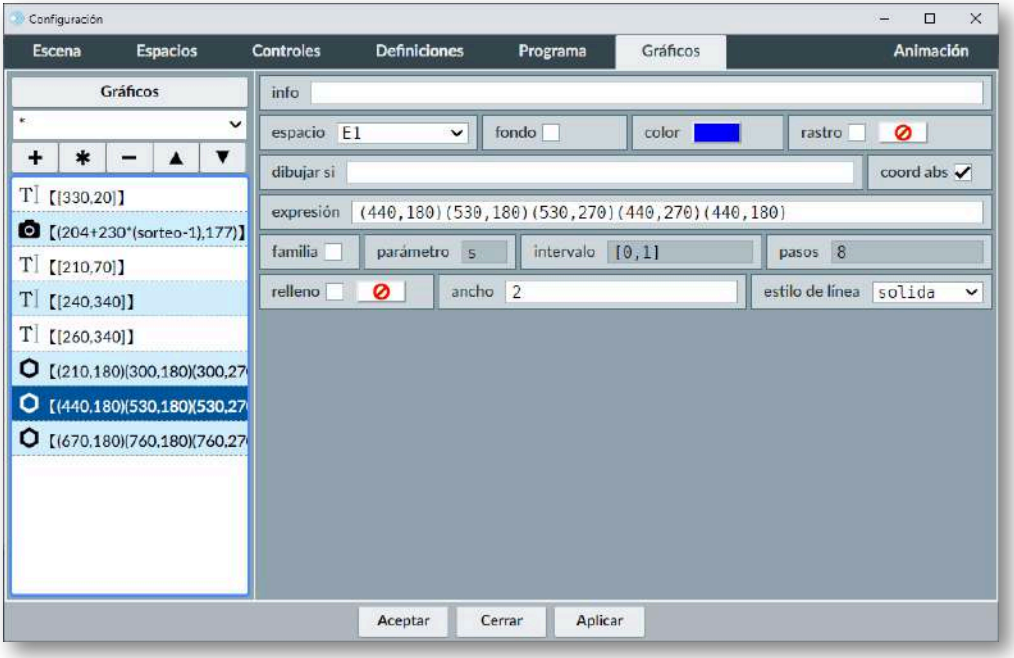


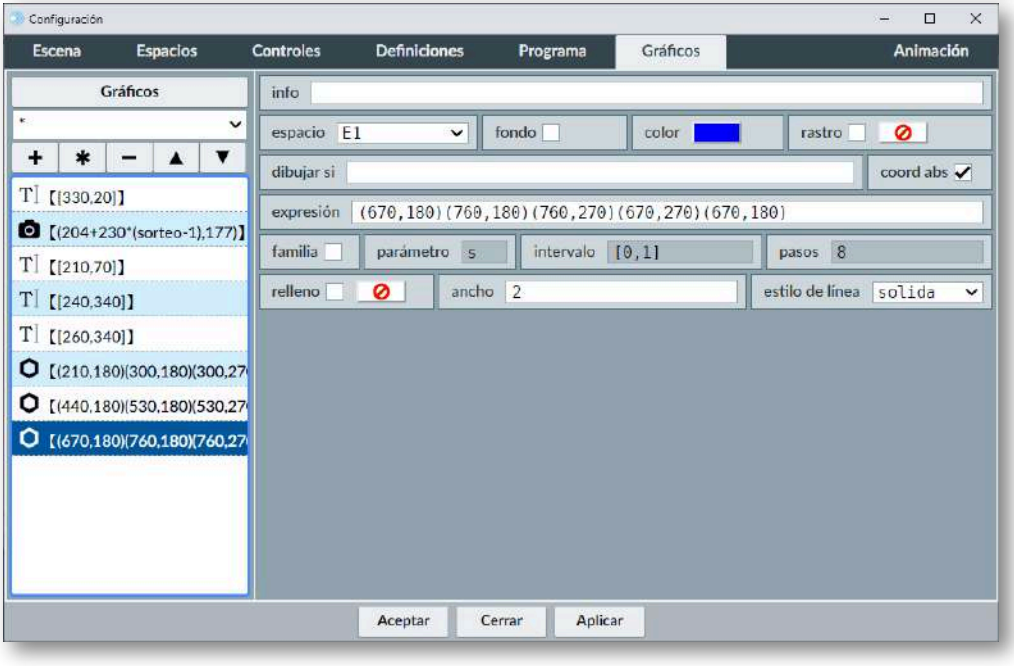
Ahora introducimos un recuadro azul que cubre las tres posibles posiciones de la casilla. Utilizando coordenadas absolutas y un ancho de línea de 2, se dibuja conectando los puntos: (210,180), (300,180), (300,270), (210,270) y cerrando en (210,180).



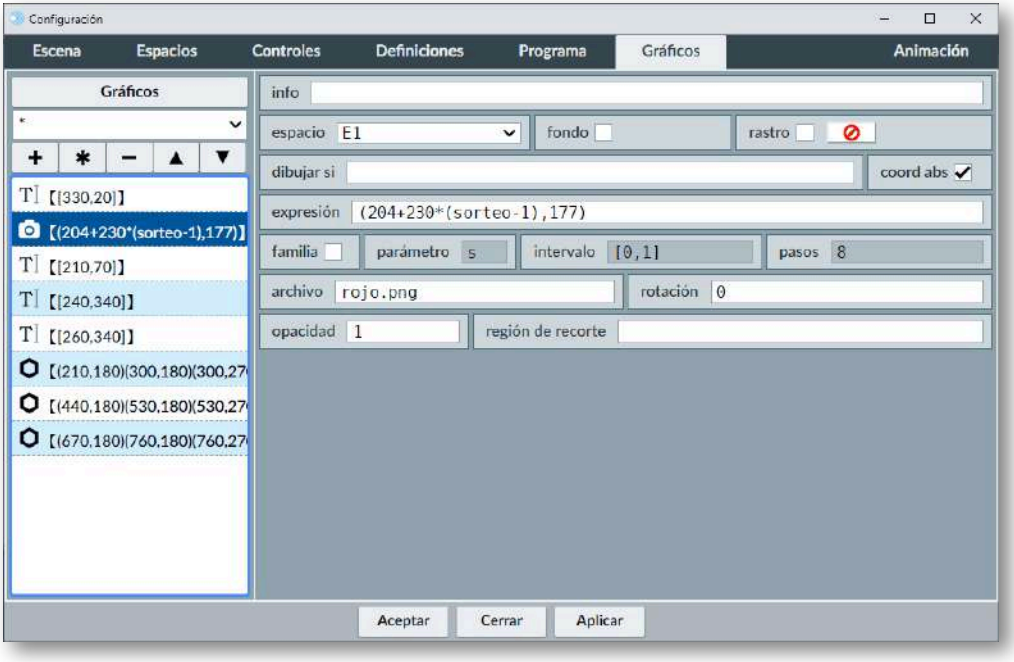


Los otros dos recuadros se crean copiando el primero y desplazando cada copia 230 unidades hacia la derecha respecto al anterior, manteniendo el mismo tamaño y color.





Se ajusta la posición de la imagen de la bolita para centrarla en el recuadro correspondiente, mediante (204 + 230 × (sorteo - 1), 177).

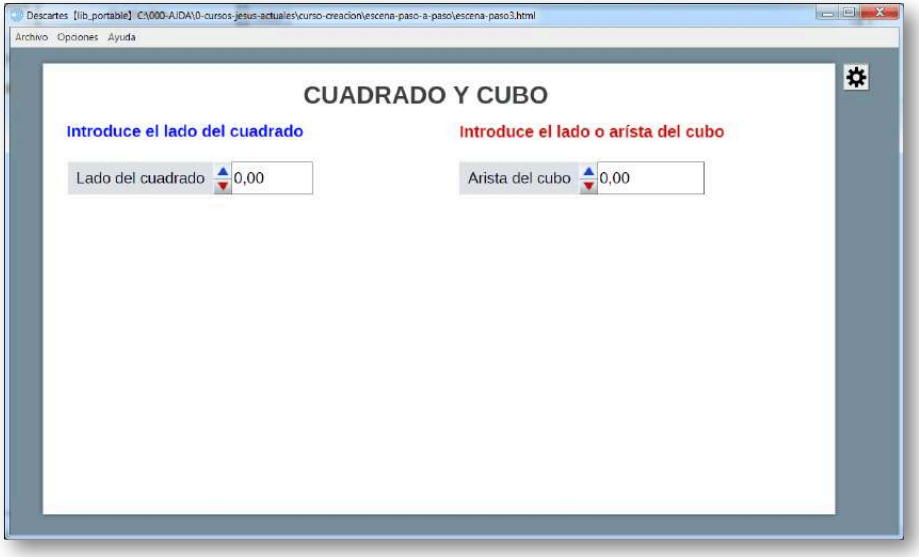




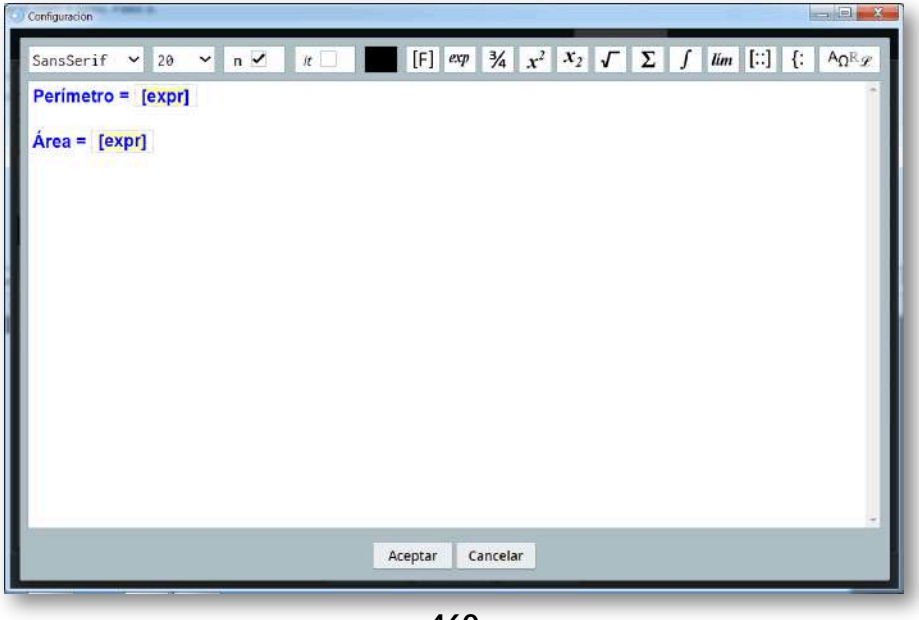
Guardamos la escena del juego con el nombre **juego-paso5**.

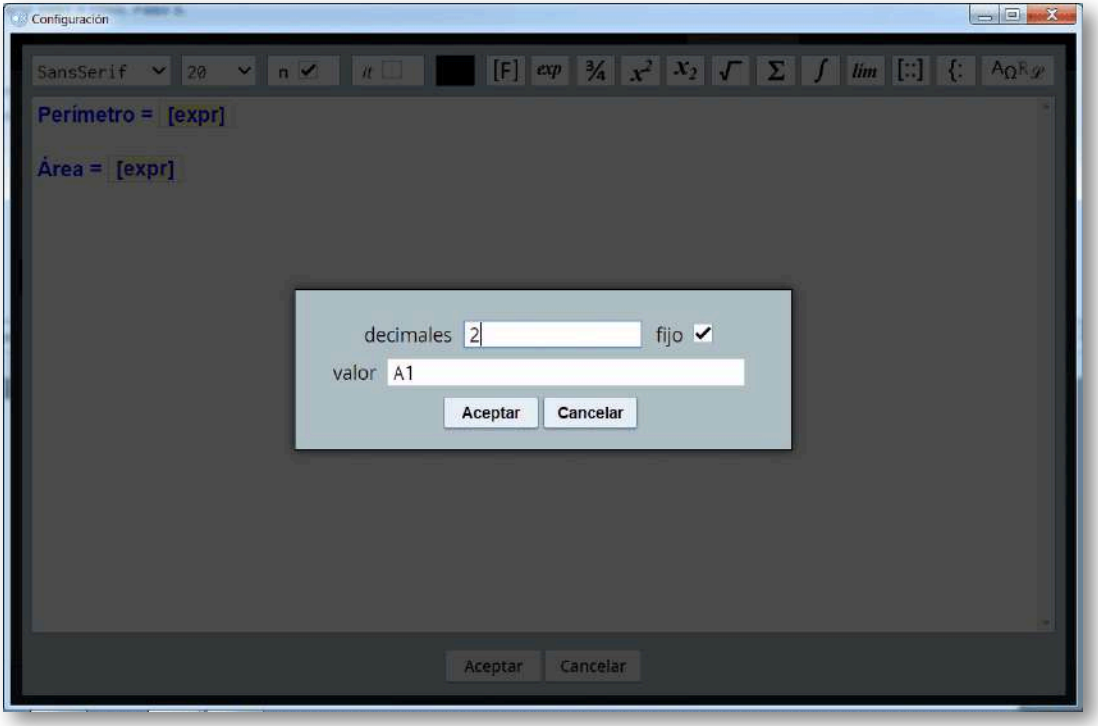
5.19 Mi primera escena paso a paso. Paso 5

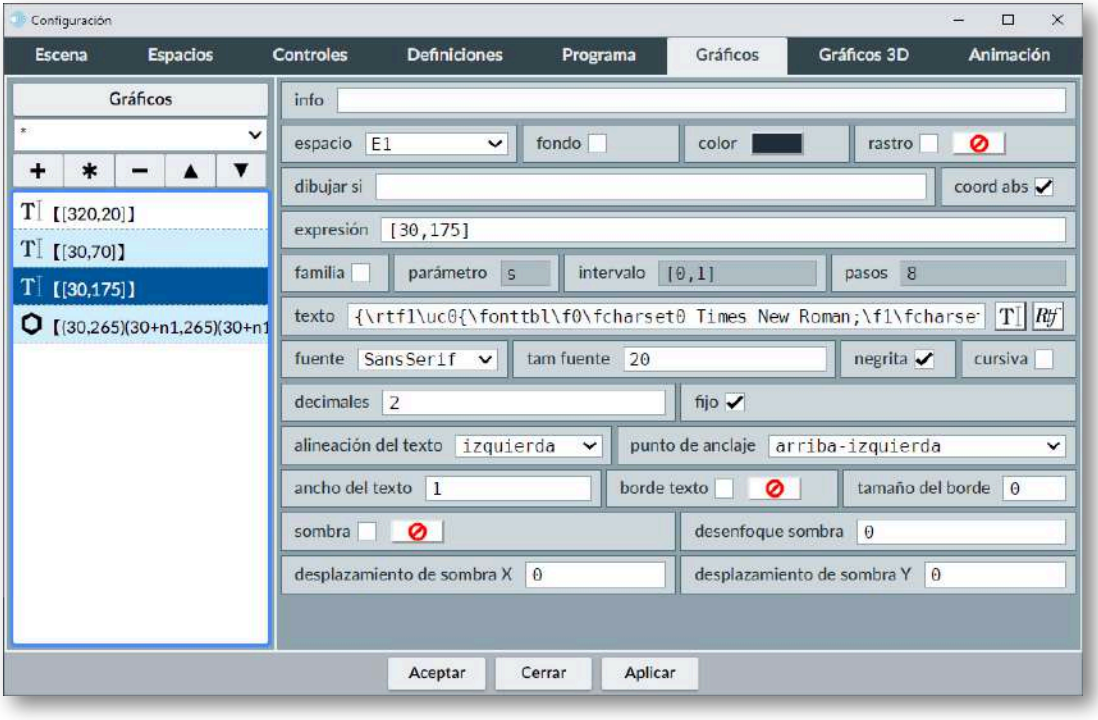
Vamos a realizar el paso 5 de la primera escena. Para comenzar, abrimos la escena **escena-paso4** con el editor de **Descartes**.

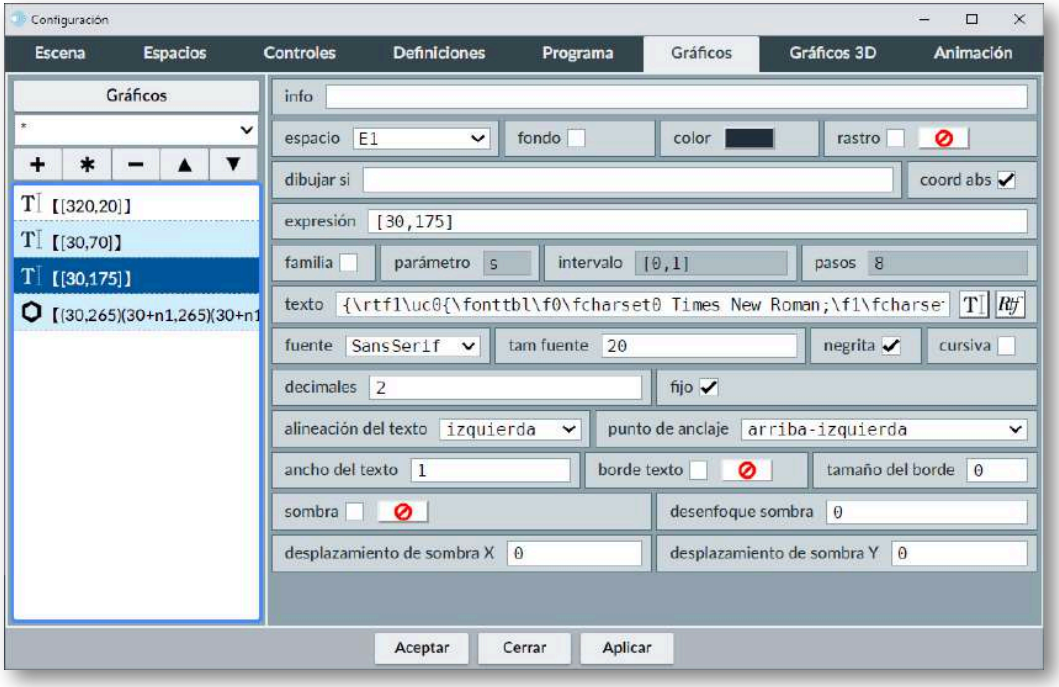


A continuación, insertaremos un texto con formato en el espacio 2D que muestre el valor del perímetro y del área del cuadrado.




Configuración
SansSerif
20
n
Perímetro = [expr]
Área = [expr]
decimales 2
fijo
valor A1
Aceptar
Cancelar
Aceptar
Cancelar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Gráficos 3D
Animación
Gráficos
[320,20]
[30,70]
[30,175]
(30,265)(30+n1,265)(30+n1
info
espacio E1
fondo
color
rastro
dibujar si
coord abs
expresión [30,175]
familia
parámetro s
intervalo [0,1]
pasos 8
texto {\rtf1\uc0{\fonttbl\f0\fcharset0 Times New Roman;\f1\fcharse
fuente SansSerif
tam fuente 20
negrita
cursiva
decimales 2
fijo
alineación del texto izquierda
punto de anclaje arriba-izquierda
ancho del texto 1
borde texto
tamaño del borde 0
sombra
desenfoque sombra 0
desplazamiento de sombra X 0
desplazamiento de sombra Y 0
Aceptar
Cerrar
Aplicar



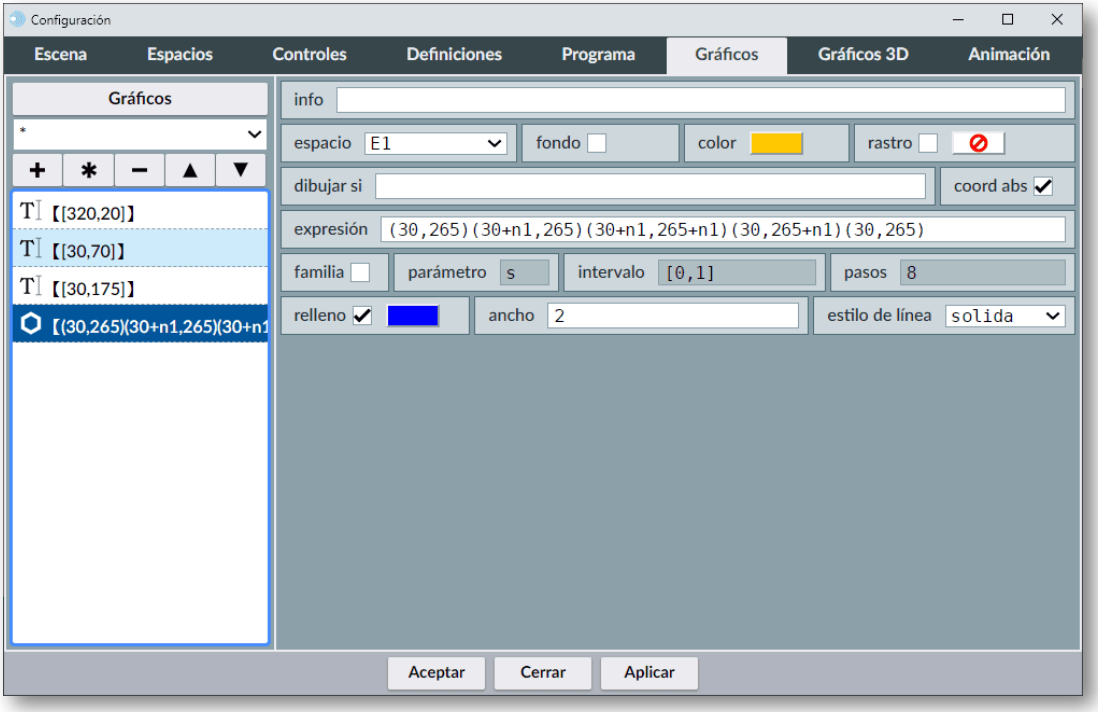
A continuación, se dibuja un cuadrado en el espacio 2D utilizando coordenadas absolutas. La longitud de su lado depende del valor de la variable **n1**, por lo que sus dimensiones variarán dinámicamente.

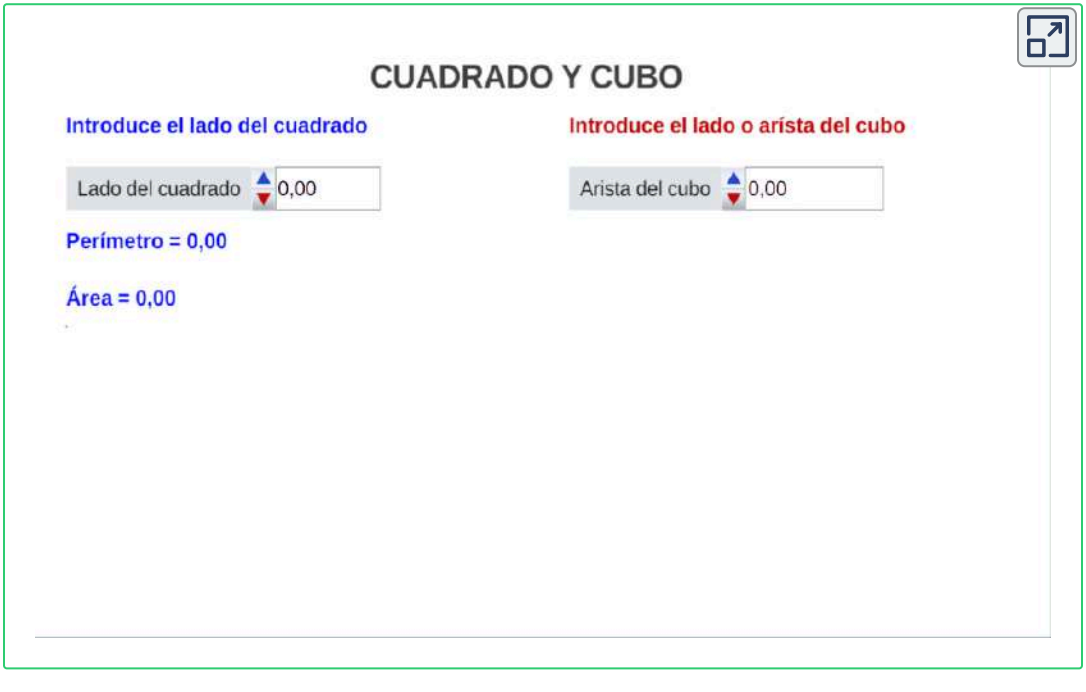
Los vértices del cuadrado se definen en el siguiente orden:

- (30, 265) → vértice inferior izquierdo.
- (30 + n1, 265) → vértice inferior derecho.
- (30 + n1, 265 + n1) → vértice superior derecho.
- (30, 265 + n1) → vértice superior izquierdo.
- (30, 265) → punto final que cierra la figura.

Como puede observarse, **n1** interviene tanto en la coordenada horizontal como en la vertical, garantizando que ancho y altura sean iguales y, por tanto, que la figura sea un cuadrado. Si **n1** aumenta, el cuadrado crece proporcionalmente; si disminuye, se reduce.

El cuadrado tendrá el **borde naranja** y el **interior azul**, lo que facilita su identificación visual dentro de la escena.





Guardamos la escena con el nombre escena-paso5.



Capítulo 6

Gráficos 3D

Capítulo 6
Gráficos 3D

6.1 Introducción 467

6.2 Inclusión gráficos 3D 469

6.3 Parámetros comunes a los gráficos 3D 473

6.4 Gráfico 3D Punto 477

6.5 Gráfico 3D Segmento 481

6.6 Gráfico 3D Polígono 484

6.7 Gráfico 3D Polígono regular 487

6.8 Gráfico 3D Triángulo 490

6.9 Gráfico 3D Curva 493

6.10 Gráfico 3D Cara 496

6.11 Gráfico 3D Superficie 499

6.12 Gráficos 3D. Cuerpos geométricos 502

6.13 Gráfico 3D Texto 518

6.14 Gráfico 3D Macro 521

6.15 Familias en gráficos 3D 533

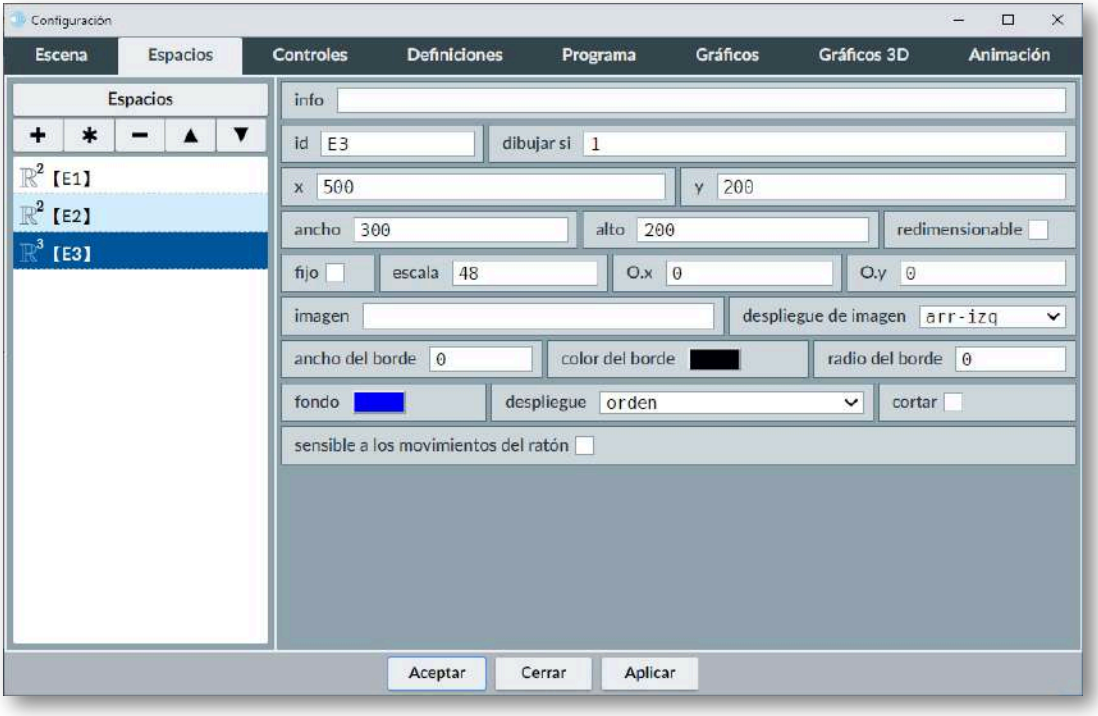
6.16 Mi primer juego paso a paso. Paso 6 537

6.17 Mi primera escena paso a paso. Paso 6 544

6.1 Introducción

Uno de los tipos de espacios que se pueden incorporar en una escena son los espacios tridimensionales o espacios 3D. Al crear un espacio 3D, aparece un nuevo apartado en el panel de configuración de la escena, se trata del selector denominado **Gráficos 3D** (si no hubiera ningún espacio 2D creado en la escena no se mostraría el selector **Gráficos**).

Para que este apartado se muestre correctamente, normalmente es necesario pulsar el botón **Aceptar** o **Aplicar** en el panel de configuración de la escena.

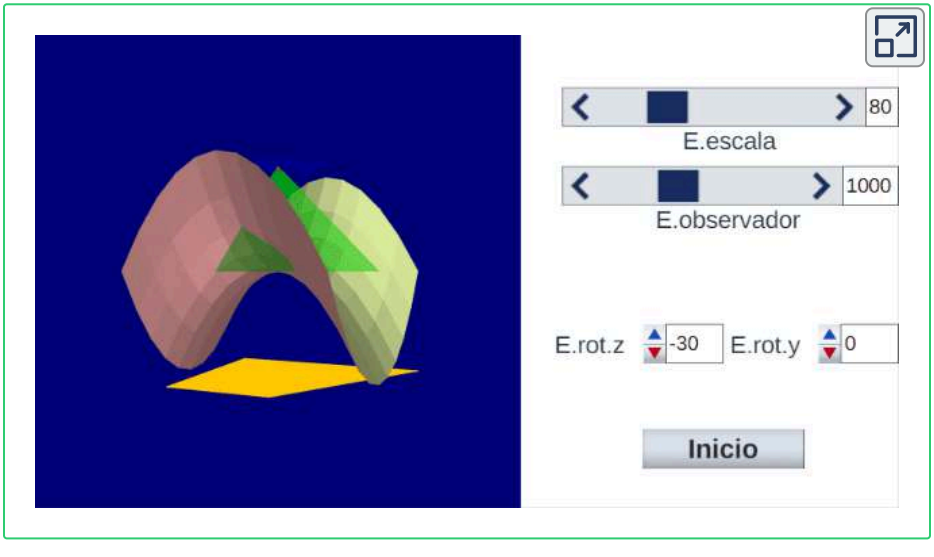


En los **gráficos 3D** es posible incorporar distintos elementos tridimensionales, como cuerpos geométricos, superficies, caras, polígonos regulares e irregulares, macros, textos, curvas, y puntos, entre otros, que lógicamente solo se representan en espacios tridimensionales o 3D.

Este capítulo explica cómo añadir y configurar elementos en espacios tridimensionales, llamados **gráficos 3D**. Los parámetros y el procedimiento de inserción son muy similares a los de un espacio 2D.

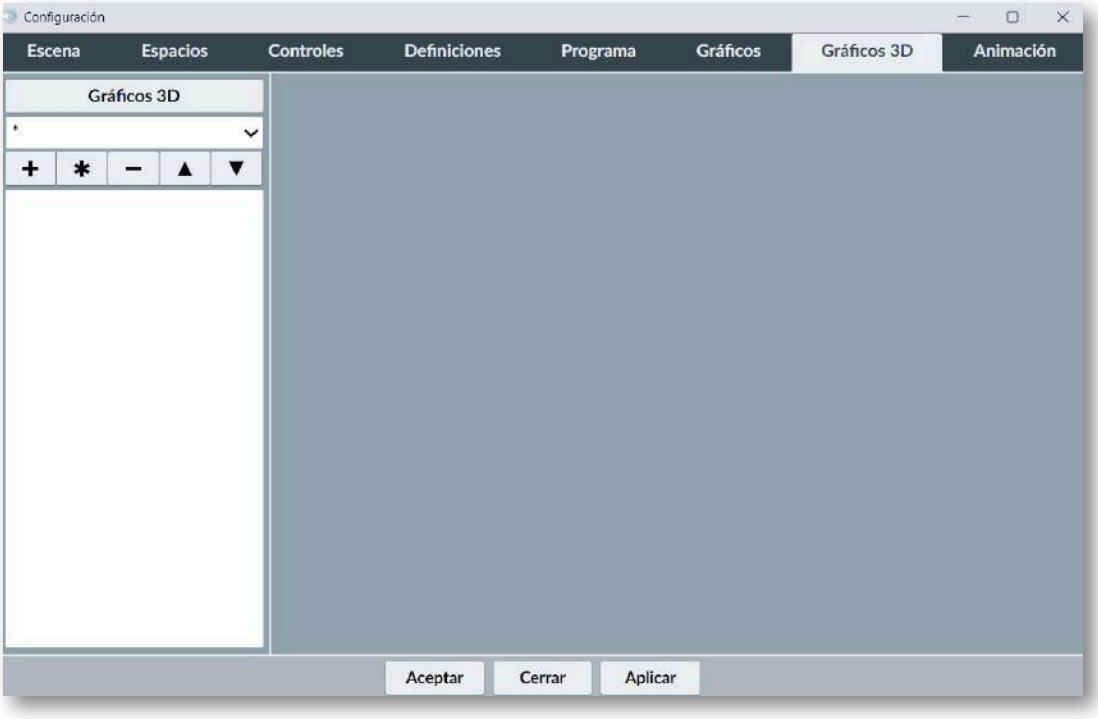
Mostramos una captura del juego **Proyecto AJDA** "Ancho y estrecho" y una escena con espacios y elementos 3D.





6.2 Inclusión gráficos 3D

Todos los gráficos comparten un panel a la izquierda, similar al del selector **Espacio**. En este panel se muestra una lista de los gráficos existentes.



Al igual que en el selector Espacio o Gráficos, dispone de un botón para añadir un nuevo gráfico, otro para duplicar el gráfico seleccionado, un control para mover el gráfico hacia arriba o hacia abajo en la lista, y un botón para eliminar el gráfico seleccionado.

Los gráficos 3D sólo pueden existir dentro de un espacio 3D. Por ello, en la parte superior del panel izquierdo existe un menú desplegable que muestra los espacios 3D disponibles y que funciona como filtro.

Por defecto aparece el símbolo de asterisco (*), lo que indica que se muestran todos los gráficos 3D de todos los espacios 3D.

Si se selecciona un espacio 3D concreto en el menú, únicamente se visualizarán los gráficos 3D pertenecientes a dicho espacio, lo que facilita su identificación cuando existen muchos elementos.

Cuando hay más de un gráfico 3D en un mismo espacio 3D y estos se intersecan, se dibujan según el orden en que aparecen en el panel izquierdo. El gráfico 3D situado más arriba en la lista se pinta primero, y el que aparece más abajo se pinta al final, pudiendo cubrir a los que se encuentran por encima en la lista.

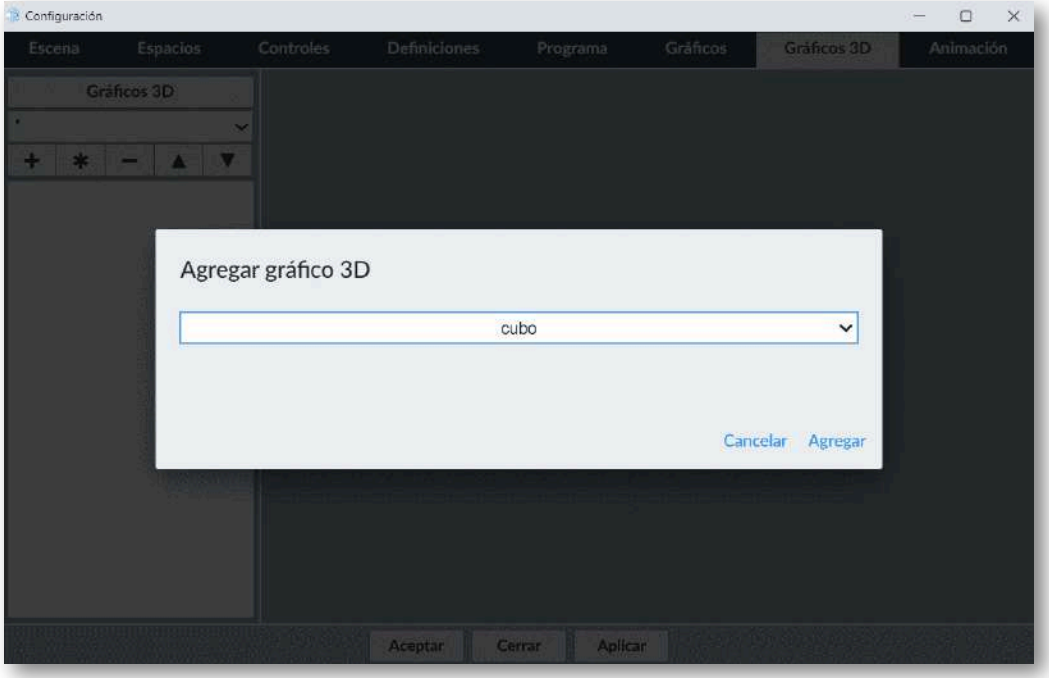
El procedimiento para introducir un elemento gráfico 3D en una escena es el siguiente:

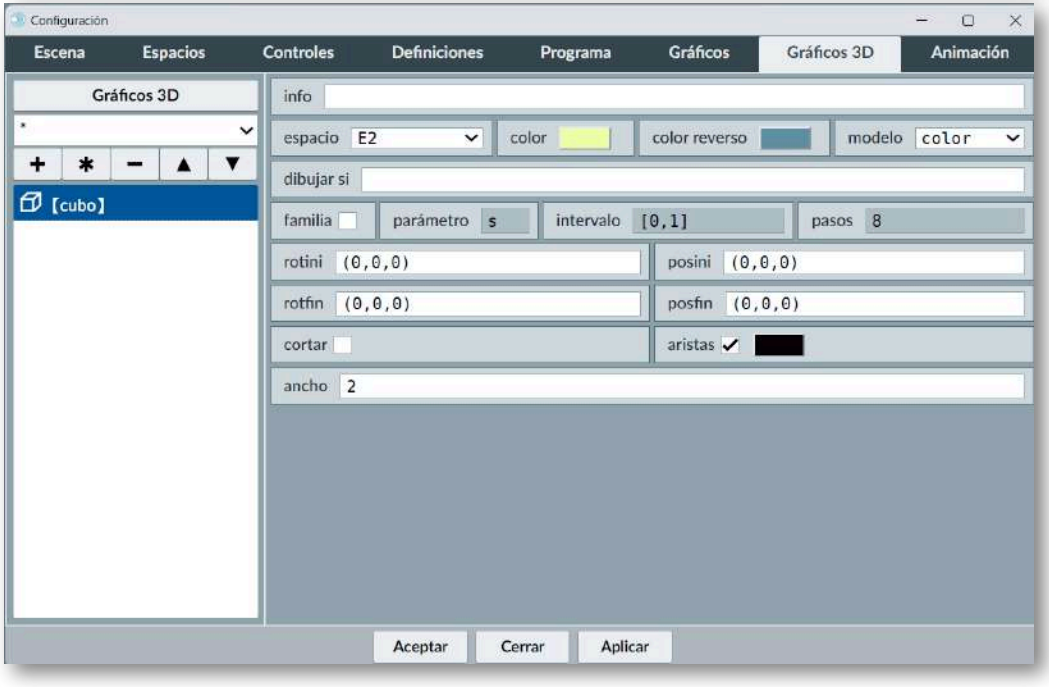
1. Acceder al panel de configuración de la escena y pulsar sobre el selector **Gráficos 3D**.
2. Añadir un nuevo elemento desde el panel izquierdo.
    1. Pulsar el botón **+** para crear un nuevo gráfico 3D (el botón **-** elimina el elemento seleccionado, * lo duplica y el control de desplazamiento permite subirlo o bajarlo en la lista). El botón **Gráficos 3D** permite visualizar, editar y modificar su código.



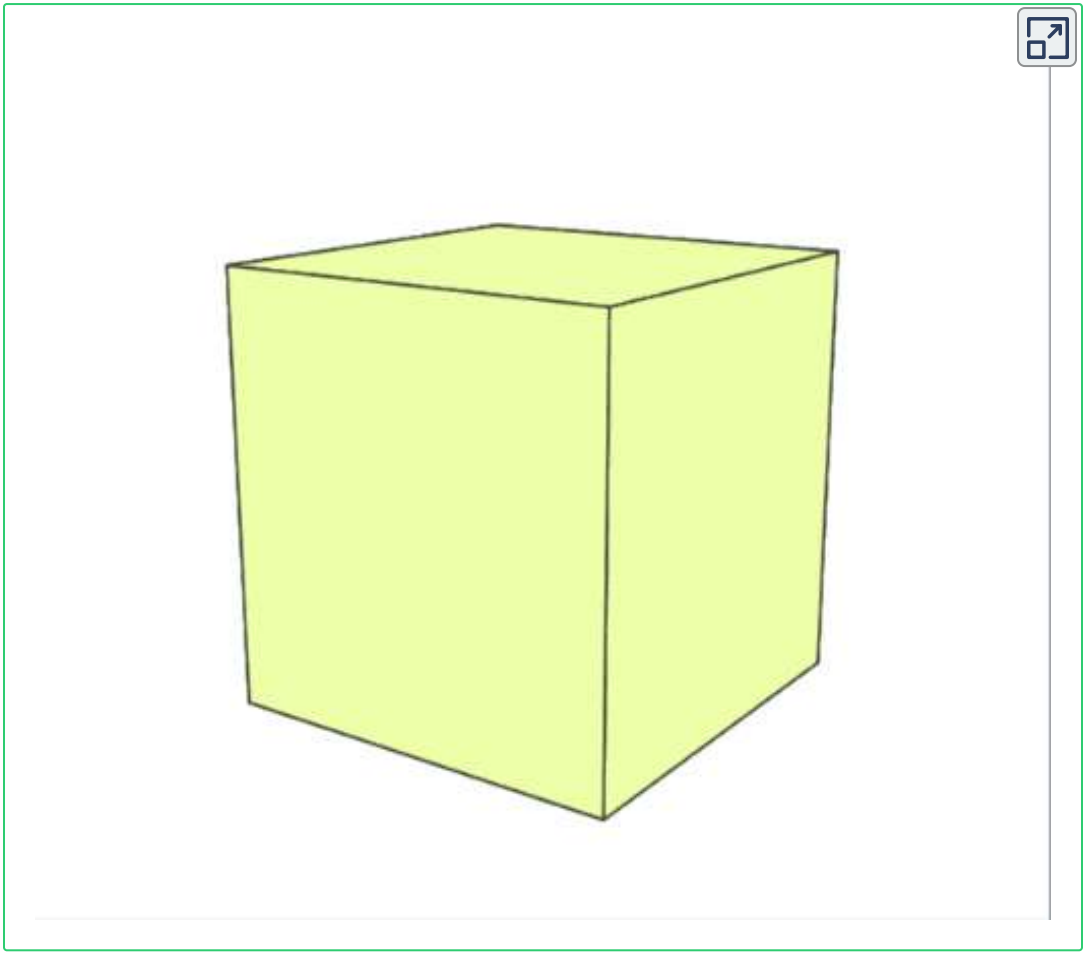
    2. Seleccionar el tipo de elemento gráfico 3D mediante el menú desplegable correspondiente.
    3. Configurar los parámetros del elemento gráfico 3D desde el panel de edición.

Por ejemplo, si se inserta un **cubo**, aparecerá el menú específico para su configuración en el editor.





A continuación, se muestra la escena creada.



A la izquierda se muestra la lista de todos los gráficos 3D de la escena (menú de selección). A la derecha aparece el panel de edición, que varía según el tipo de gráfico 3D y permite modificar todos los parámetros que lo definen.

En el menú desplegable situado encima de la etiqueta azul **Gráficos 3D**, si se selecciona un espacio concreto, sólo se mostrarán los gráficos pertenecientes a ese espacio; si se selecciona *, se visualizarán todos los gráficos 3D de la escena.

6.3 Parámetros comunes a los gráficos 3D

A continuación, se describen los principales parámetros comunes de los elementos gráficos 3D en los espacios tridimensionales de escenas de Descartes:

- **Info:** Comentario para el programador, no visible en escena.
- **Color reverso:** Botón que lanza la herramienta de control de colores para definir el color de la cara posterior del gráfico 3D. Para más información, consulte el apartado sobre la herramienta de control de colores.
- **rotini:** Campo de texto donde se introduce la rotación inicial como un vector de tres entradas [x, y, z]. La primera entrada rota respecto al eje X, la segunda respecto al eje Y y la tercera respecto al eje Z. También se pueden usar ángulos de Euler, colocando la palabra Euler antes del vector, para rotaciones en ejes modificados.
- **posini:** Campo de texto que define la traslación inicial del gráfico mediante un vector de tres entradas [x, y, z]. Esta traslación se aplica después de la rotación inicial.
- **rotfin:** Funciona como rotini pero se aplica después de la traslación inicial definida por posini, tomando como referencia el origen.
- **posfin:** Funciona como posini, pero se aplica después de la rotación final (rotfin).
- **Cortar:** Checkbox que permite mostrar intersecciones entre gráficos. Es útil para destacar solapamientos, aunque puede ralentizar la escena si hay muchos objetos.
- **Aristas:** Checkbox que hace que se dibujen las aristas del gráfico 3D (caras, polígonos o superficies). Incluye un botón para seleccionar el color de las aristas mediante la herramienta de control de color.

- **Modelo:** Menú con cuatro opciones:
  - **Color:** Pinta el gráfico con un color fijo.
  - **Luz:** Añade sensación de iluminación, creando variaciones de brillo según la orientación.
  - **Metal:** Similar a luz, pero aumenta el contraste de los brillos, dando efecto metálico.
  - **Alambre:** Solo traza las aristas con el color seleccionado, sin sombreado en las caras.
- **Nu:** Número de divisiones del parámetro u para gráficos dependientes de un parámetro.
- **Nv:** Número de divisiones del parámetro v, aplicable a gráficos que requieren dos parámetros, como superficies.
- **Ancho:** Campo de texto que define el ancho de ciertos gráficos, como el elipsoide.
- **Largo:** Campo de texto que define el largo de ciertos gráficos, como el elipsoide.

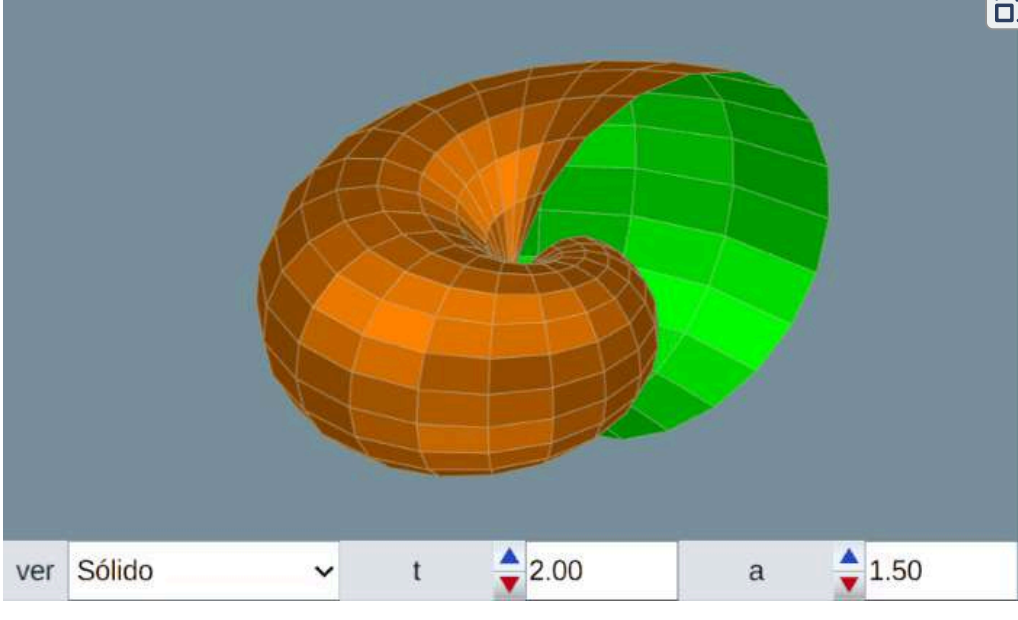
Recordemos que los gráficos 3D disponen de variables que permiten conocer y controlar los ángulos de giro del espacio, como **<Espacio>.rot.x** y **<Espacio>.rot.y**, donde **<Espacio>** es el nombre del espacio.

Además, existen las variables **<Espacio>.escala** y **<Espacio>.observador**, que permiten controlar la escala del espacio y la distancia aparente al observador, respectivamente, ambas medidas en píxeles.

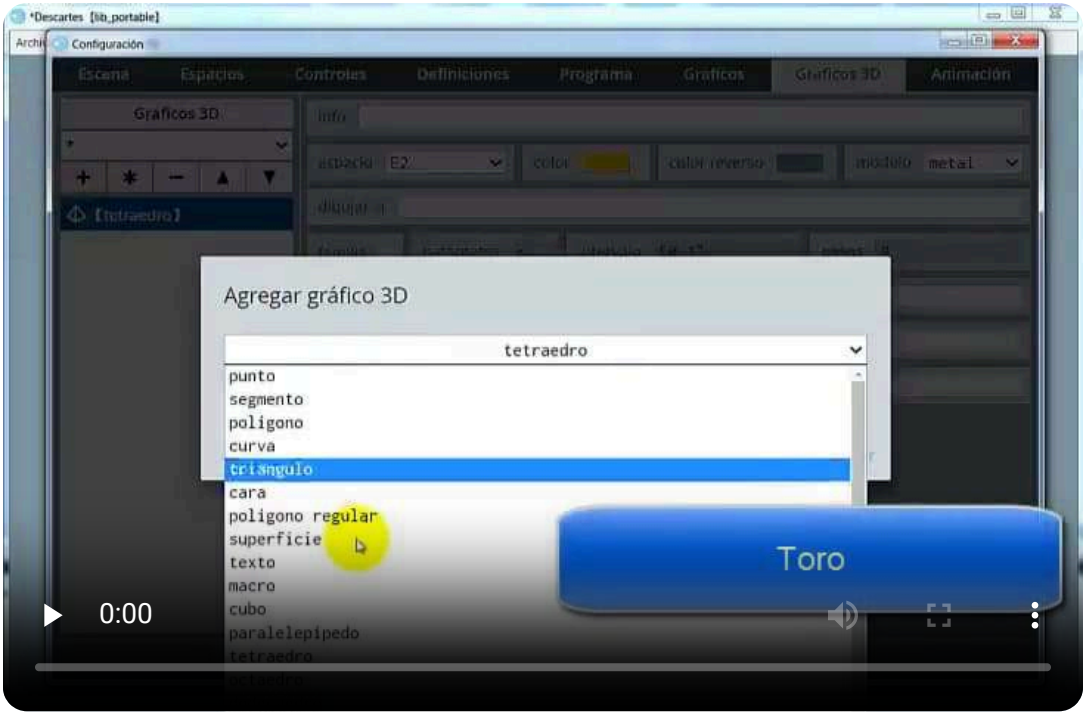
Estas variables se utilizan, con nombres externos distintos, en la escena de los poliedros. La siguiente escena ilustra su uso: si el usuario cambia la escala o gira el espacio arrastrando el ratón, los controles numéricos se actualizan automáticamente con los nuevos valores de escala y rotación.

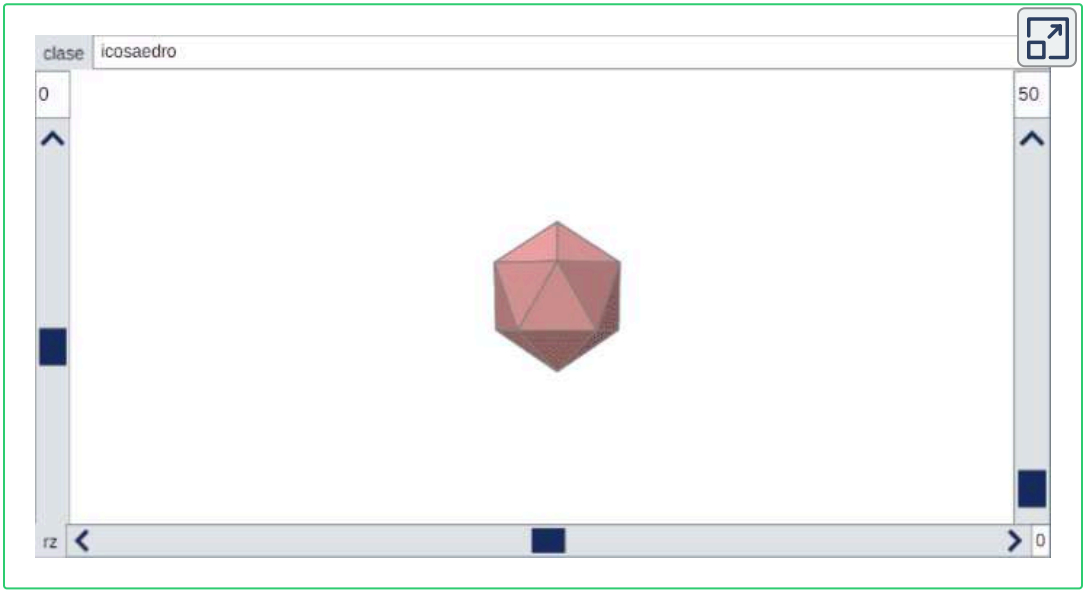
En gráficos 3D, al igual que en cualquier espacio, es posible configurar el fondo para que sea transparente. Mostramos ejemplos de escenas con diferentes gráficos 3D, la segunda con fondo transparente.





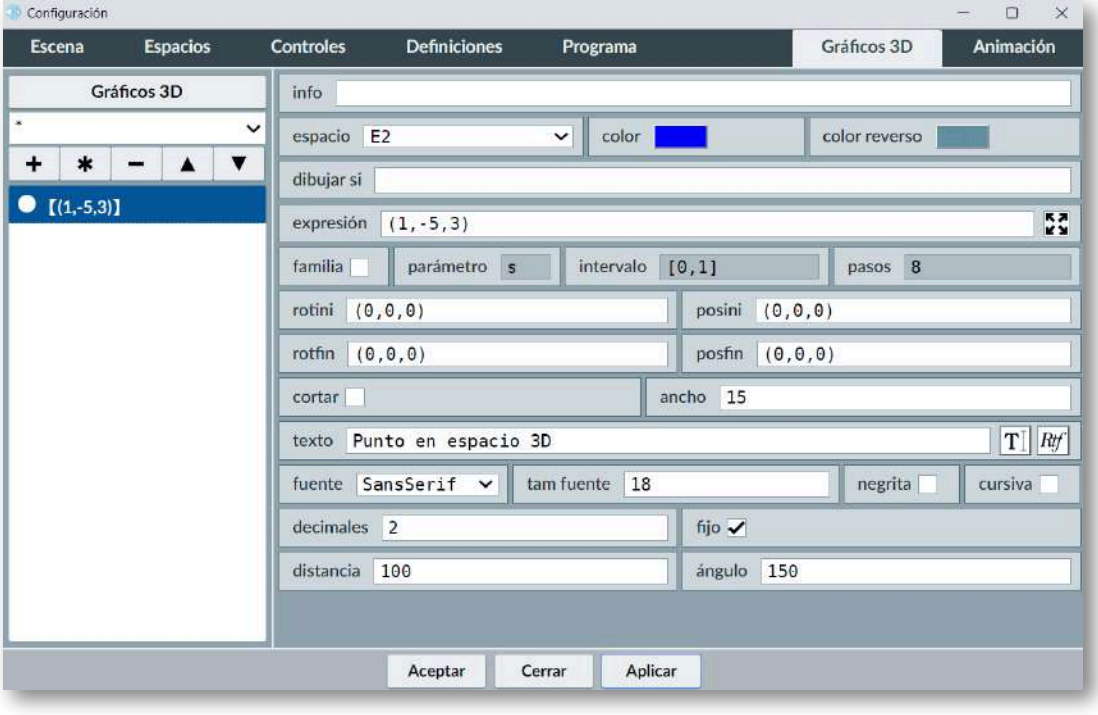
Presentamos un vídeo sobre gráficos 3D acompañado de una escena con ejemplos de los distintos cuerpos geométricos tridimensionales.





6.4 Gráfico 3D Punto

Es un gráfico que se inserta únicamente en un espacio tridimensional. Funciona de forma análoga al punto en dos dimensiones, con la diferencia de que el campo **expresión** contiene tres coordenadas: la primera corresponde al eje x, la segunda al eje y y la tercera al eje z.

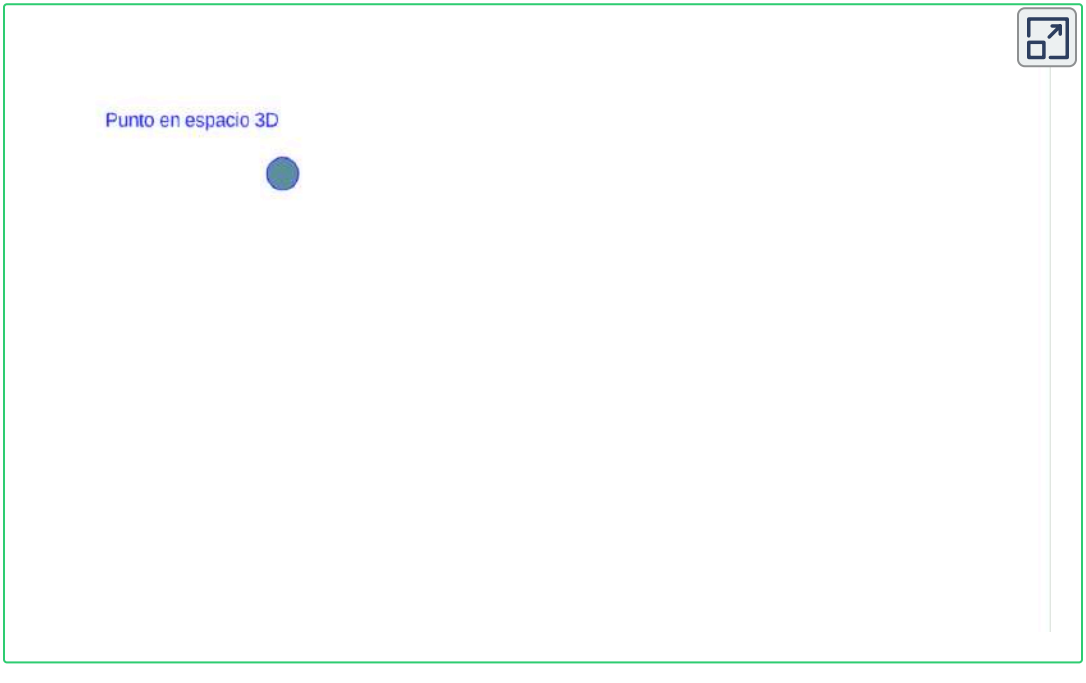


Parámetros característicos presentes en los gráficos 3D tipo punto son los siguientes:

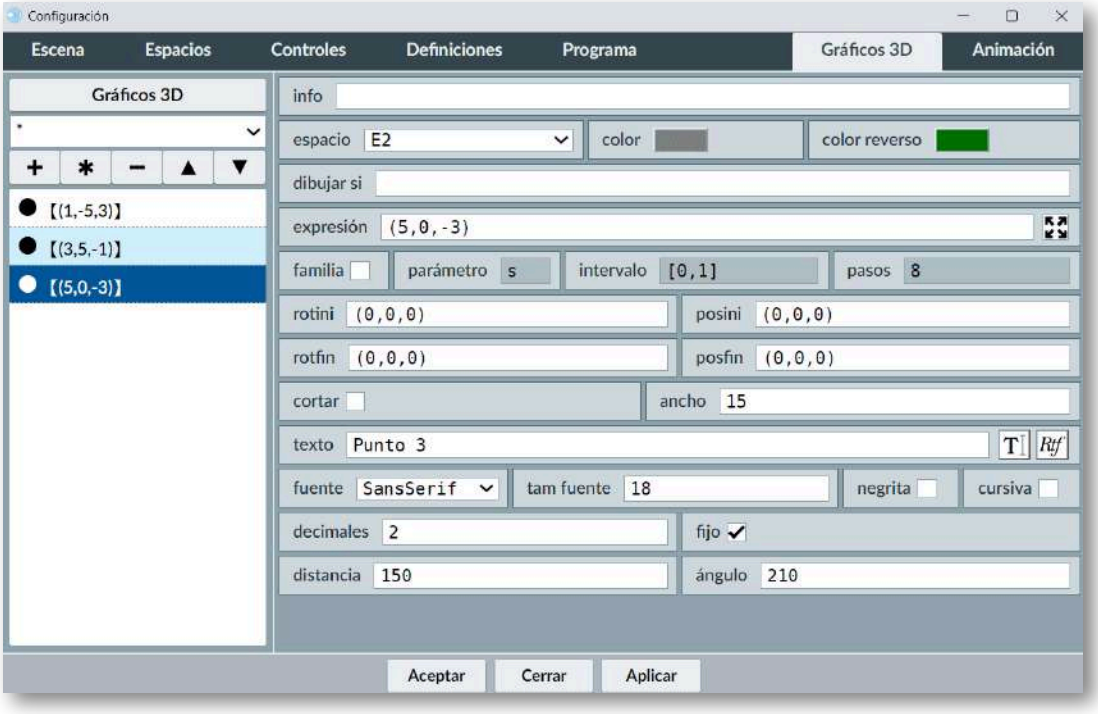
- **expresión:** es un campo de texto en donde se colocan las coordenadas 3D del único. Estas coordenadas deben ser relativas.
- **decimales:** es un campo de texto donde se introduce un número entero correspondiente al número de decimales que habrán de mostrarse en caso de que el texto contenga alguna variable cuyo valor se quiera visualizar.

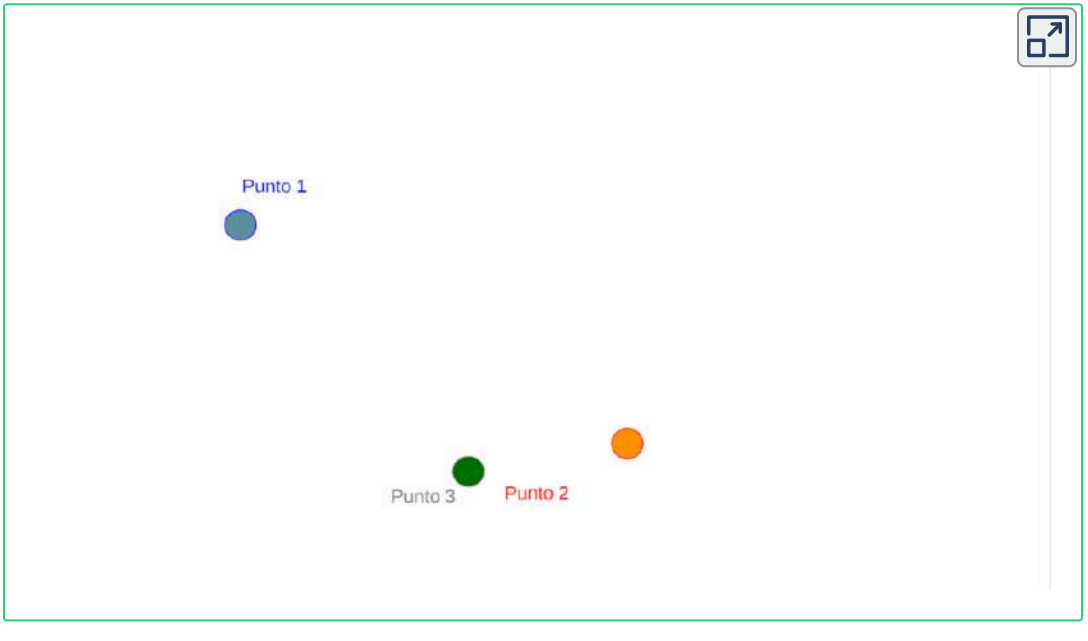
- **fijo:** es un checkbox que determina si siempre se muestran el número de decimales elegido en el control decimales, aun cuando no sean significativos. Siempre se mostrarán los decimales elegidos si este checkbox está marcado; de lo contrario, solo se mostrarán si son significativos.
- **Ancho:** tamaño del punto en píxeles. El valor por defecto es 1. Si se pone cero, el punto desaparece.
- **Distancia:** Campo de texto que afecta al texto asociado al punto. Indica la separación, en píxeles, entre el punto y su texto.
- **Ángulo:** Campo de texto que determina la dirección (en grados) en la que se coloca el texto respecto al punto. Tomando como referencia las coordenadas del punto, el texto se sitúa a la distancia indicada por distancia y con la orientación definida por ángulo.

Mostramos la escena del punto en espacio 3D presentado en la introducción del apartado.

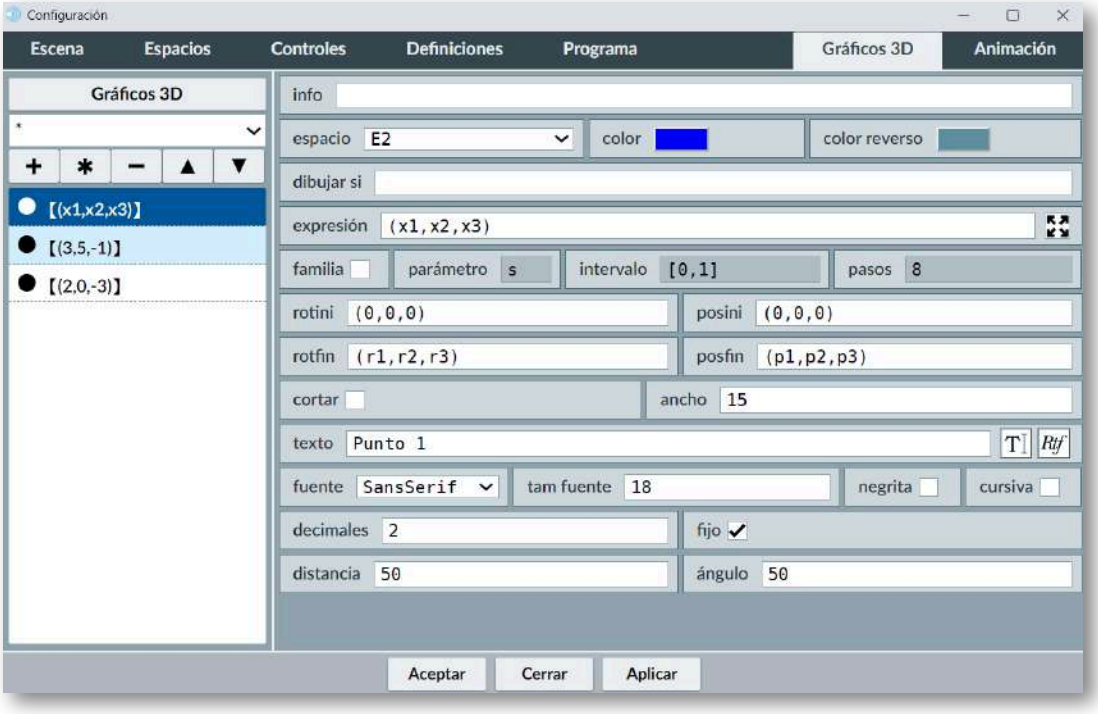


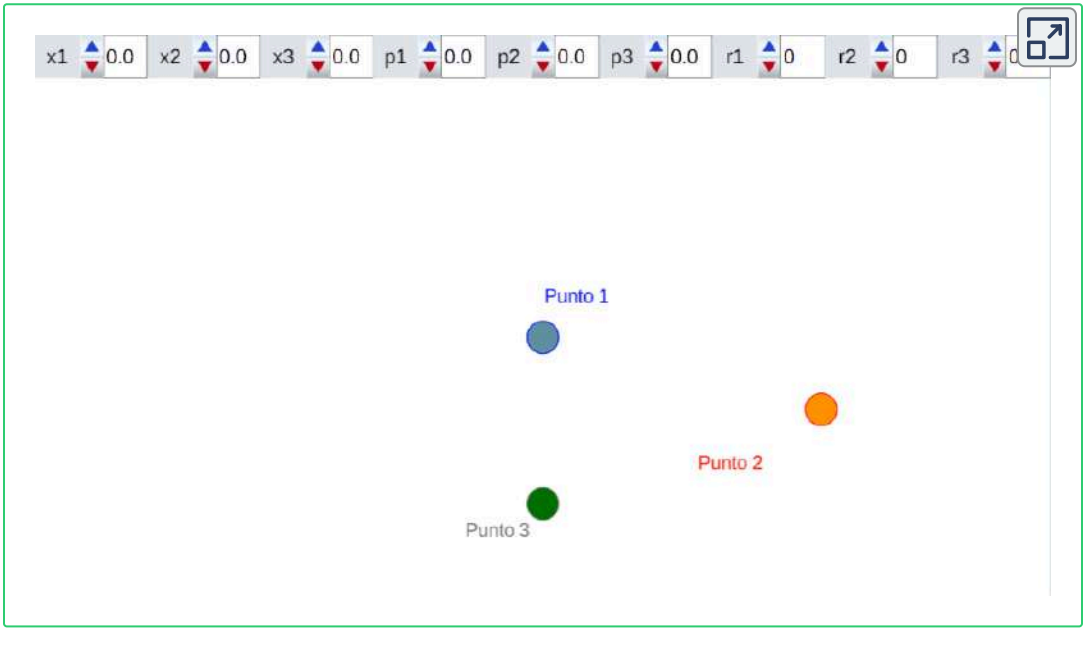
Incluimos un ejemplo de una escena con tres puntos. Al mover la escena puede apreciarse la tridimensionalidad del espacio.





En el siguiente ejemplo hemos introducido pulsadores para controlar la posición, el desplazamiento y la rotación de uno de los puntos.

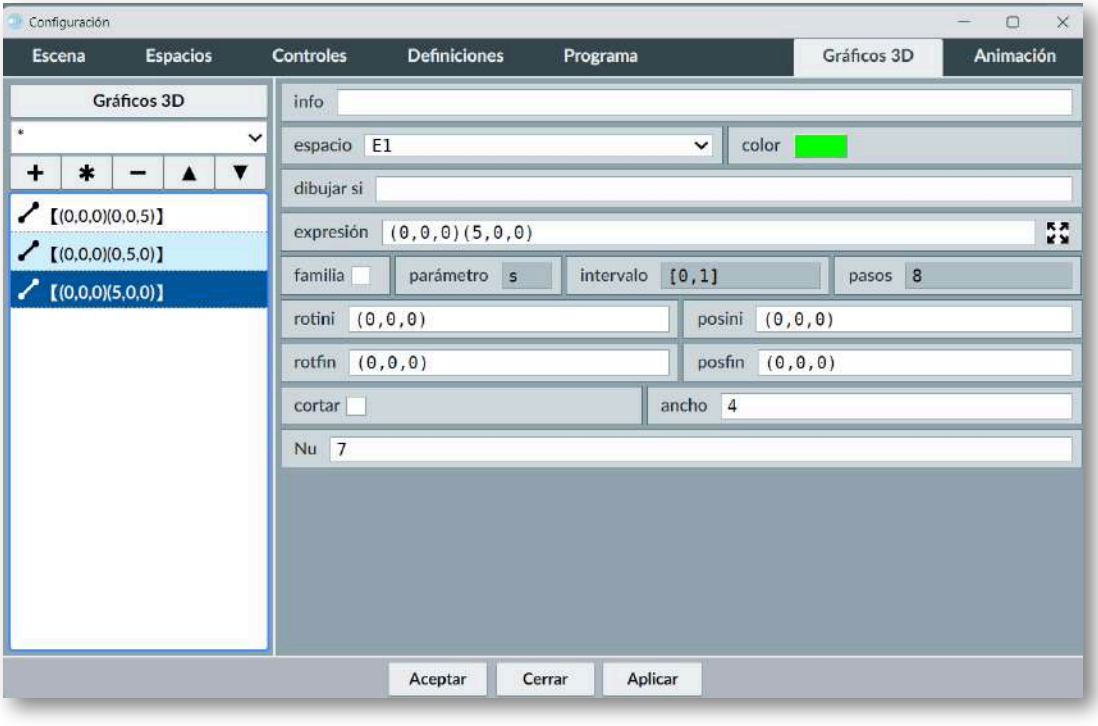




6.5 Gráfico 3D Segmento

Es equivalente al gráfico segmento en dos dimensiones, con la diferencia de que en el campo **expresión** las coordenadas de inicio y fin contienen tres entradas, una por cada eje del espacio tridimensional.

Un segmento es una línea recta limitada por dos puntos. El panel de configuración específico del tipo segmento 3D presenta el siguiente aspecto:

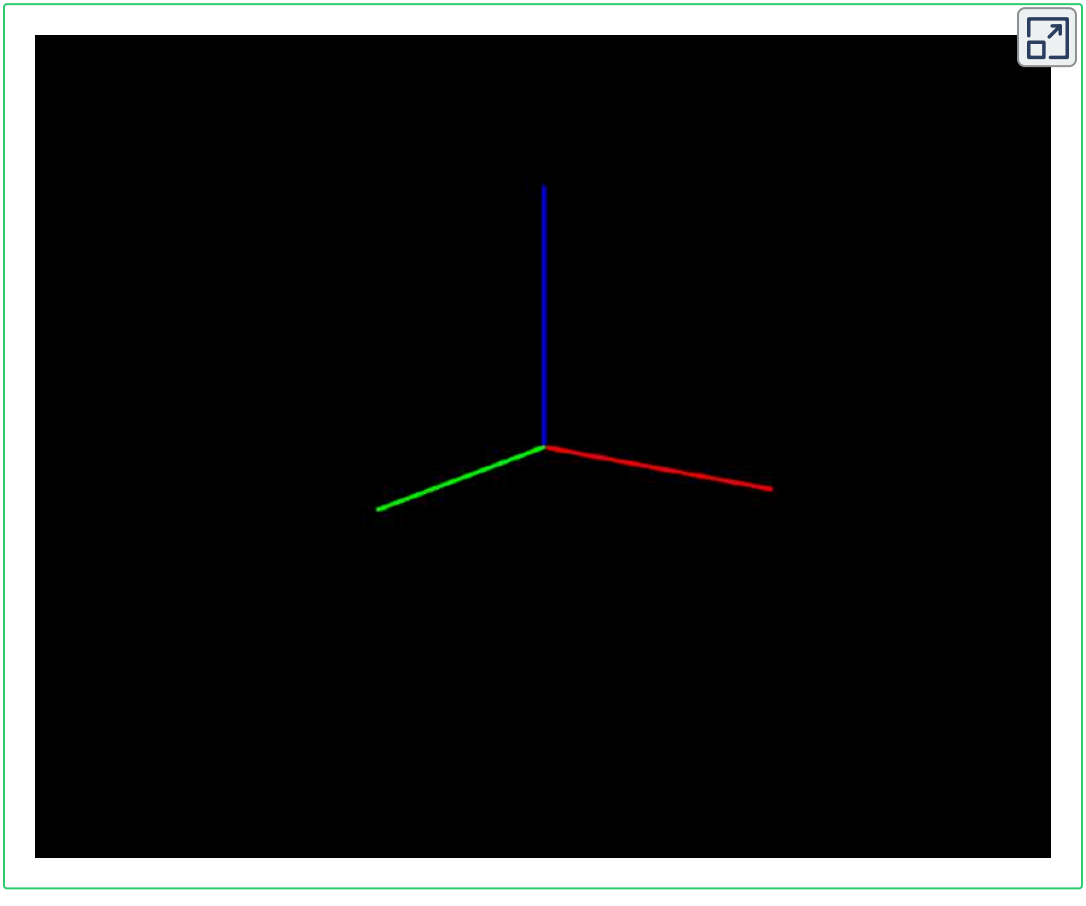


Parámetros característicos presentes en los gráficos 3D tipo segmento son los siguientes:

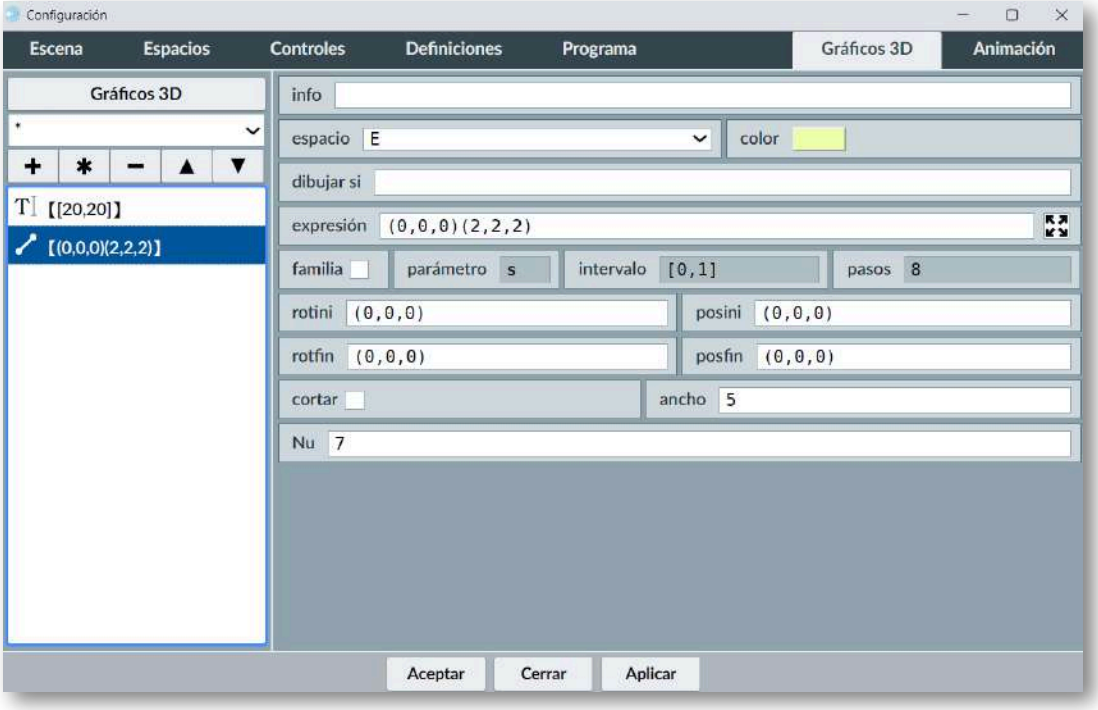
- **expresión:** Es un campo de texto. Su contenido debe tener la forma: $(X_1,Y_1,Z_1)(X_2,Y_2,Z_2)$ donde X_1, Y_1, Z_1, X_2, Y_2, Z_2, son expresiones numéricas que representan las coordenadas de los extremos del segmento.

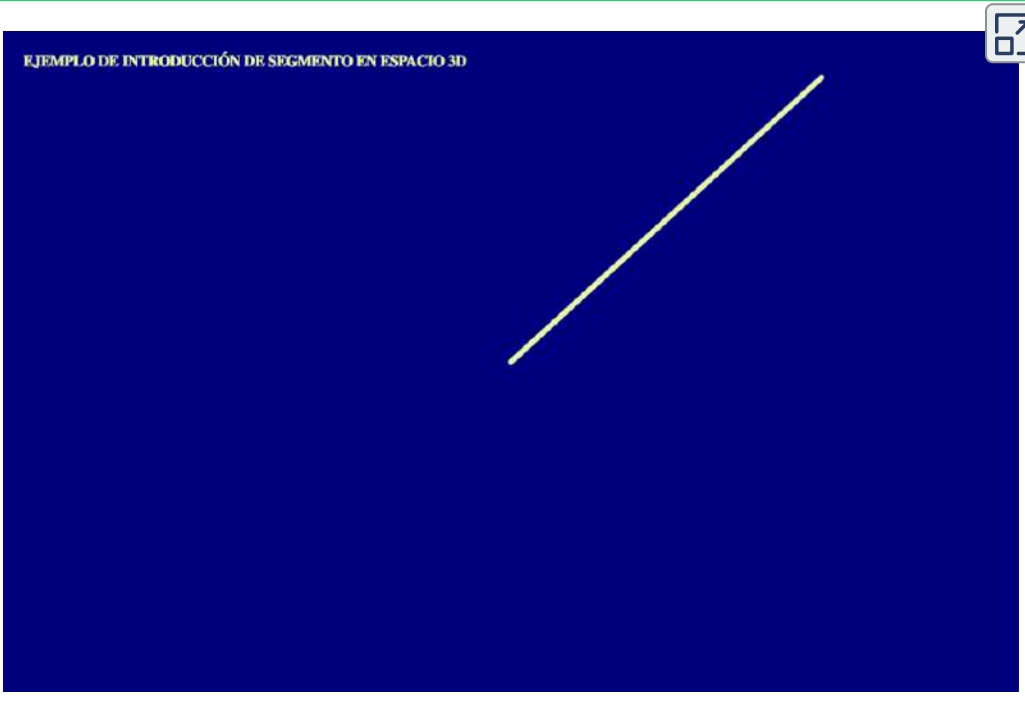
- **ancho:** es un campo de texto donde se introduce un número entero correspondiente al ancho en píxeles del segmento.
- **Nu:** es el número de intervalos en los que se parte el segmento que se va a dibujar. Basta indicar 1 para su representación. Si se indica un número mayor representará el segmento como una poligonal de segmentos.

Presentamos la escena que contiene los tres segmentos ejemplificados en la introducción del apartado.



Incluimos también un ejemplo de un segmento animado en un espacio 3D.

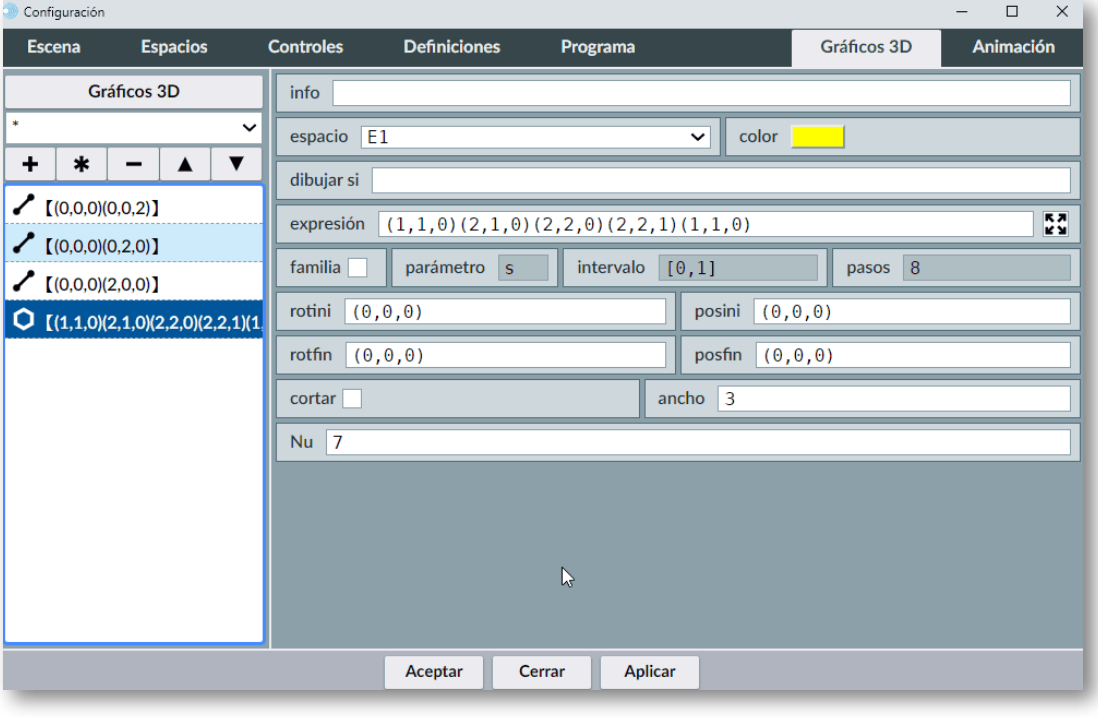

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Gráficos 3D
【[20,20]】
【(0,0,0)(2,2,2)】
info
espacio E
color
dibujar si
expresión (0,0,0)(2,2,2)
familia
parámetro s
intervalo [0,1]
pasos 8
rotini (0,0,0)
posini (0,0,0)
rotfin (0,0,0)
posfin (0,0,0)
cortar
ancho 5
Nu 7
Aceptar
Cerrar
Aplicar


EJEMPLO DE INTRODUCCIÓN DE SEGMENTO EN ESPACIO 3D

6.6 Gráfico 3D Polígono

Es similar al gráfico polígono en dos dimensiones, con la diferencia de que en el campo **expresión** las coordenadas de cada vértice contienen tres entradas, una por cada eje del espacio tridimensional. Otra diferencia es que no admiten relleno.

Los polígonos en 3D no admiten relleno. Están formados por un conjunto de lados o segmentos unidos entre sí dentro del espacio tridimensional.

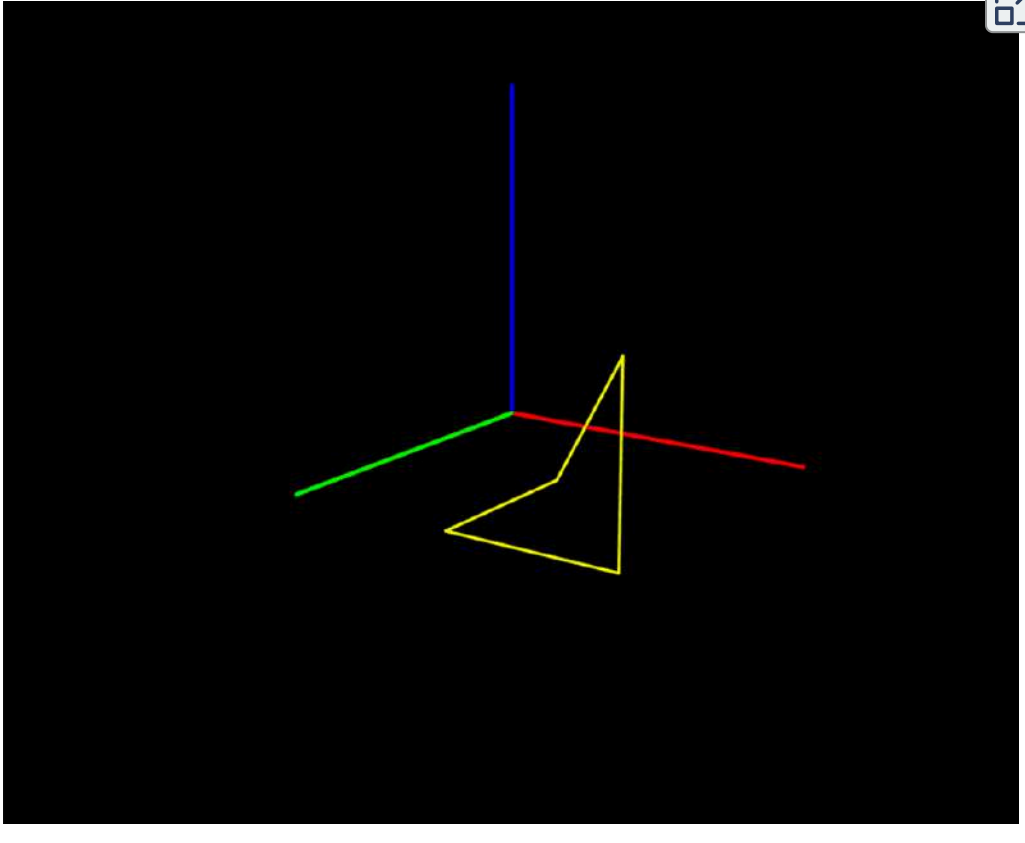


Parámetros característicos presentes en los gráficos 3D tipo polígono son los siguientes:

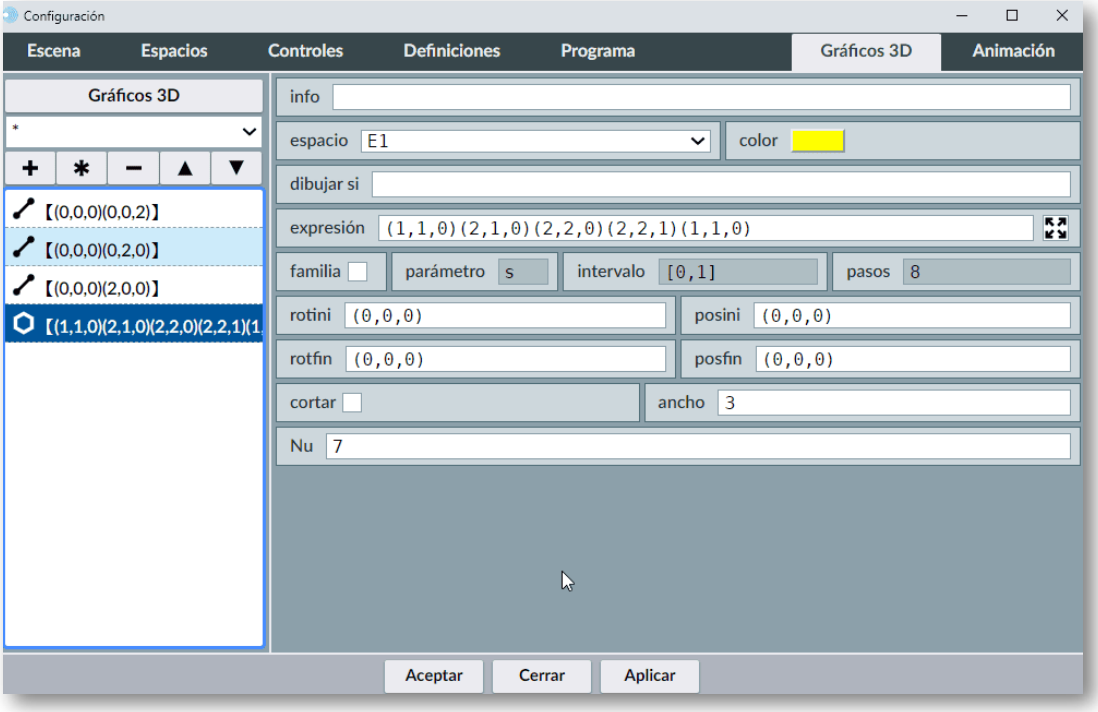
- **Expresión:** Campo de texto cuyo contenido debe tener la forma $(X_1,Y_1,Z_1)(X_2,Y_2,Z_2)...(X_n,Y_n,Z_n)$, donde cada terna (X_i,Y_i,Z_i) representa las coordenadas numéricas de un vértice de la poligonal en el espacio 3D.

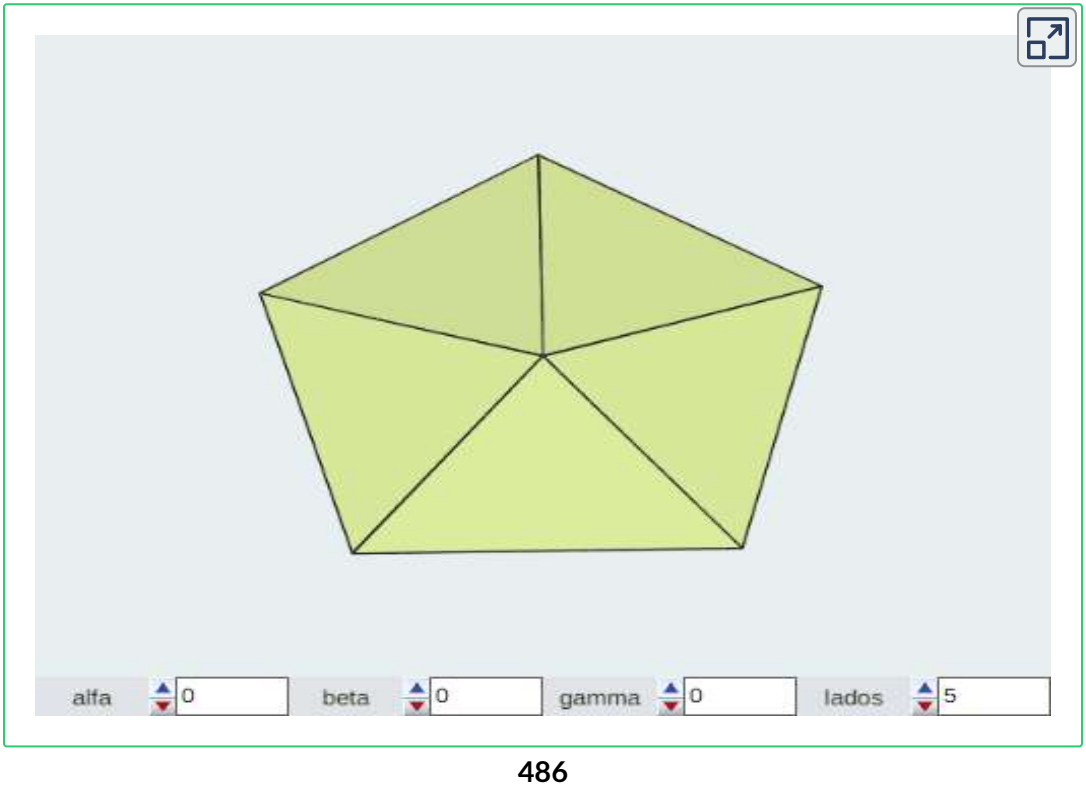
- **Nu:** Campo de texto que define en cuántos segmentos se subdivide cada lado de la poligonal. En la mayoría de los casos es suficiente establecer **Nu = 1**.
- **Ancho:** Indica el grosor de los segmentos que forman el polígono.

Presentamos la escena que contiene el polígono ejemplificado en la introducción del apartado; además, se han dejado tres segmentos como referencia.



Incluimos un ejemplo en el que la posición y la rotación del polígono se controlan mediante pulsadores.

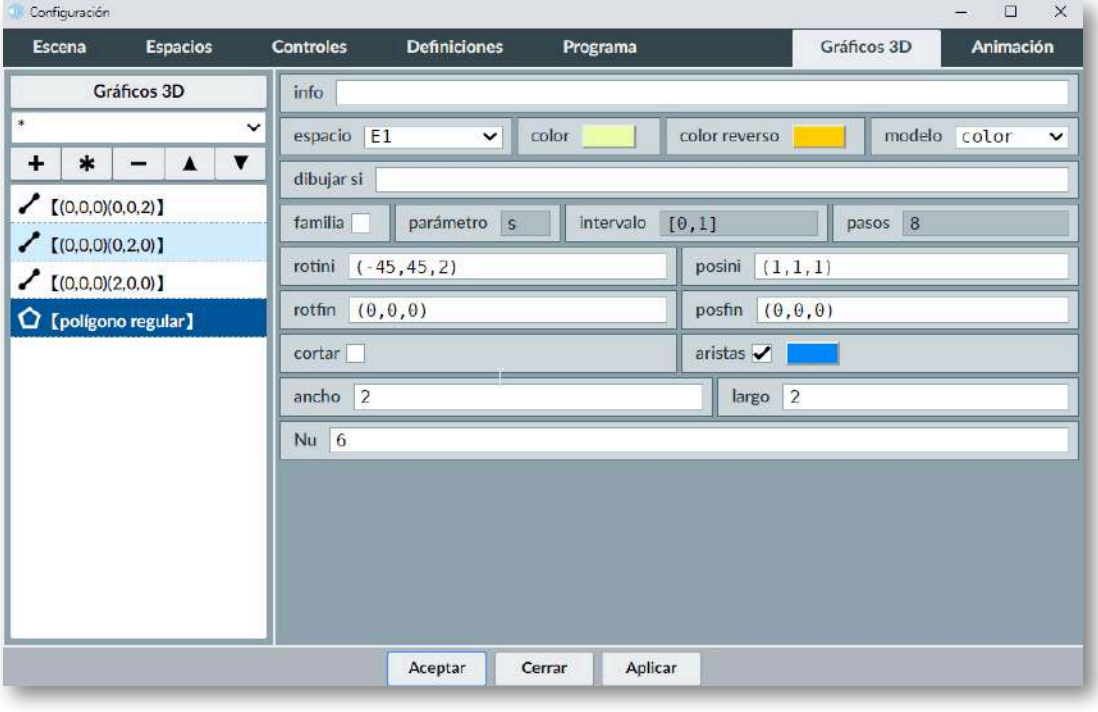

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Gráficos 3D
*
【(0,0,0)(0,0,2)】
【(0,0,0)(0,2,0)】
【(0,0,0)(2,0,0)】
【(1,1,0)(2,1,0)(2,2,0)(2,2,1)(1
info
espacio E1
color
dibujar si
expresión (1,1,0)(2,1,0)(2,2,0)(2,2,1)(1,1,0)
familia
parámetro s
intervalo [0,1]
pasos 8
rotini (0,0,0)
posini (0,0,0)
rotfin (0,0,0)
posfin (0,0,0)
cortar
ancho 3
Nu 7
Aceptar
Cerrar
Aplicar


alfa 0
beta 0
gamma 0
lados 5

6.7 Gráfico 3D Polígono regular

En los espacios 3D se pueden introducir polígonos regulares (figuras planas con todos sus lados iguales), indicando el número de lados deseado, como cuadrados, pentágonos o hexágonos.

Este gráfico permite definir un polígono regular mediante el parámetro **Nu**, que determina su número de lados. Además, admite rotaciones y traslaciones iniciales y finales, tal como se describe en el apartado de parámetros comunes de los gráficos 3D.



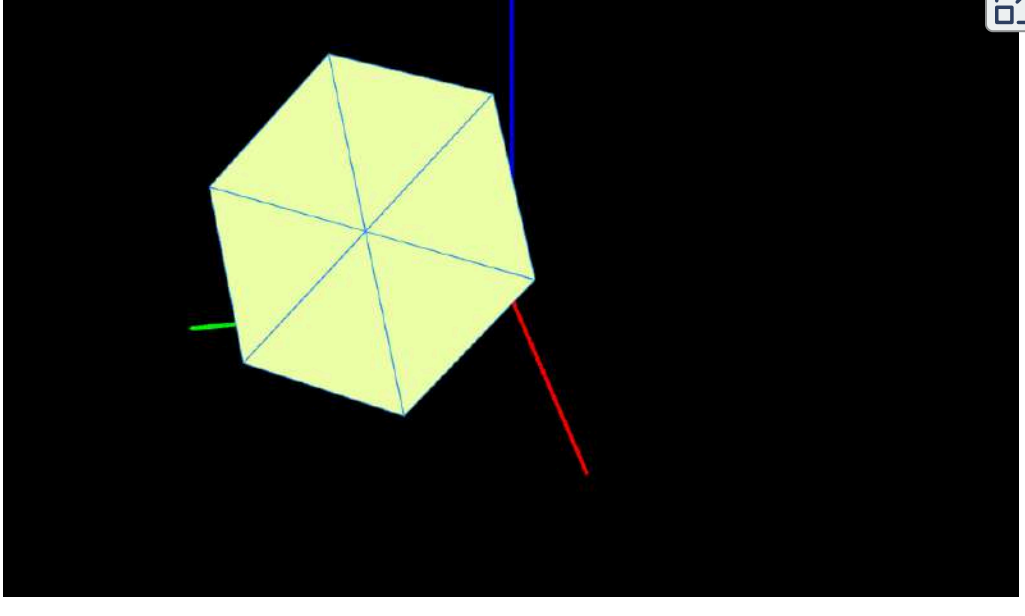
Parámetros característicos presentes en los gráficos 3D tipo polígono regular son los siguientes:

- **Expresión:** Campo de texto utilizado únicamente como identificador durante la edición. El número de lados lo determina **Nu**. El radio del polígono regular es la mitad de **ancho** (siempre que **ancho = largo**).

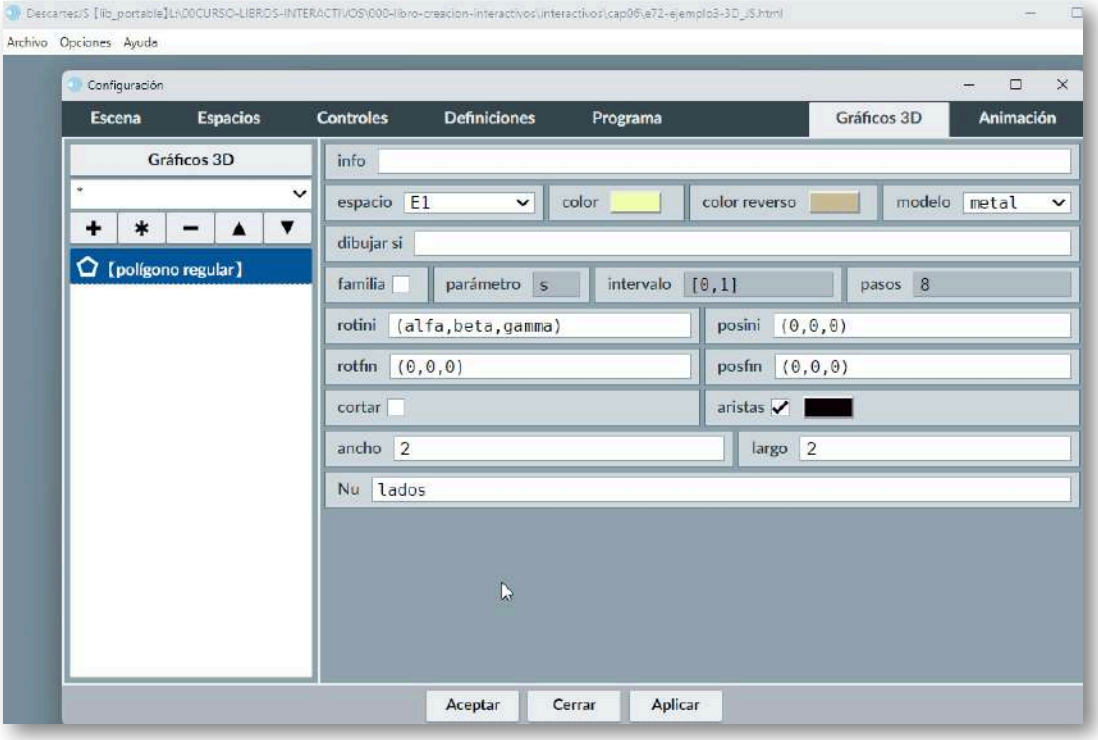
Si **ancho** y **largo** son distintos, el polígono deja de ser regular y se convierte en una deformación de uno regular.

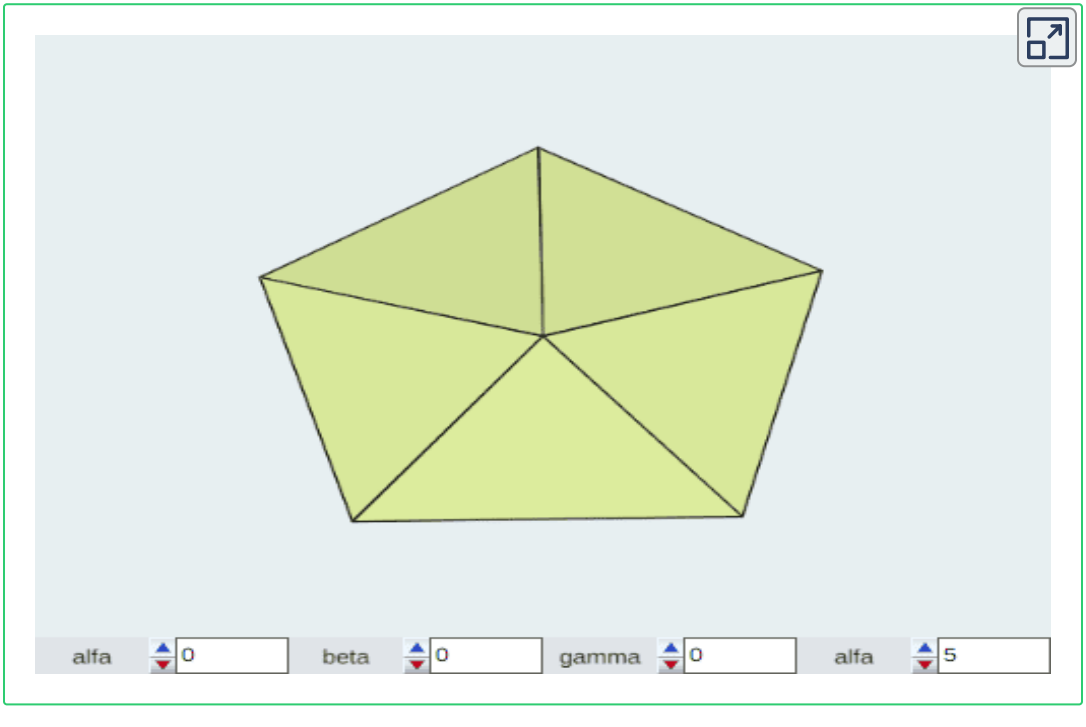
- **Nu:** Campo de texto que indica el número de lados del polígono regular.
- **Ancho:** Define la mitad del radio del polígono regular inscrito en su esfera circunscrita. Si **ancho ≠ largo**, el polígono se deforma.
- **Largo:** Define también la mitad del radio de la esfera circunscrita. Para que el polígono sea regular debe cumplirse **ancho = largo**; en caso contrario, se obtiene una deformación.

Mostramos la escena que contiene el polígono regular ejemplificado en la introducción del apartado; además, se han dejado tres segmentos como referencia.



En el siguiente ejemplo hemos introducido un polígono regular cuyo número de lados puede modificarse mediante un control de tipo pulsador. Además, la escena dispone de controles que permiten rotar la figura geométrica en distintas direcciones.

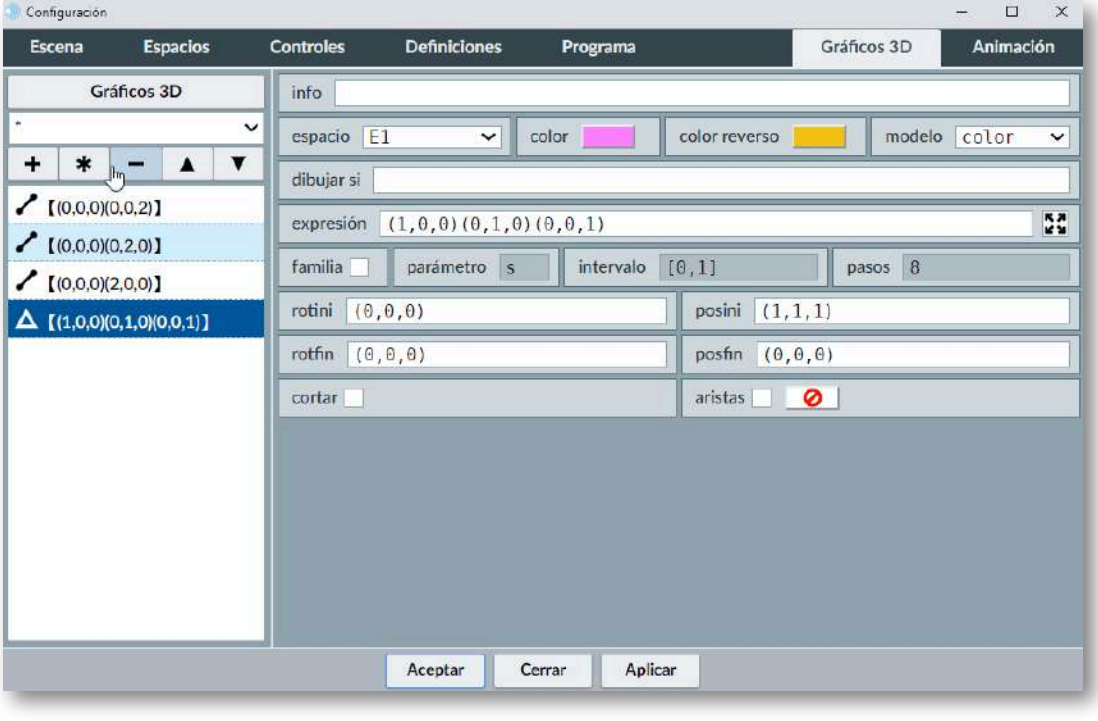

Descartes/S [lib_portable]\L:\00CURSO-LIBROS-INTERACTIVOS\000-libro-creacion-interactivos\interactivos\cap06\e72-ejemplo3-3D_JS.html
Archivo Opciones Ayuda
Configuración
Escena Espacios Controles Definiciones Programa Gráficos 3D Animación
Gráficos 3D
[polígono regular]
info
espacio E1 color color reverso modelo metal
dibujar si
familia parámetro s intervalo [0,1] pasos 8
rotini (alfa,beta,gamma) posini (0,0,0)
rotfin (0,0,0) posfin (0,0,0)
cortar aristas
ancho 2 largo 2
Nu lados
Aceptar Cerrar Aplicar


alfa 0 beta 0 gamma 0 alfa 5

6.8 Gráfico 3D Triángulo

Los triángulos en 3D son superficies planas con dos caras (anverso y reverso), tres vértices y tres lados. Requieren, además, la especificación del modelo de coloreado para determinar cómo se representan visualmente.

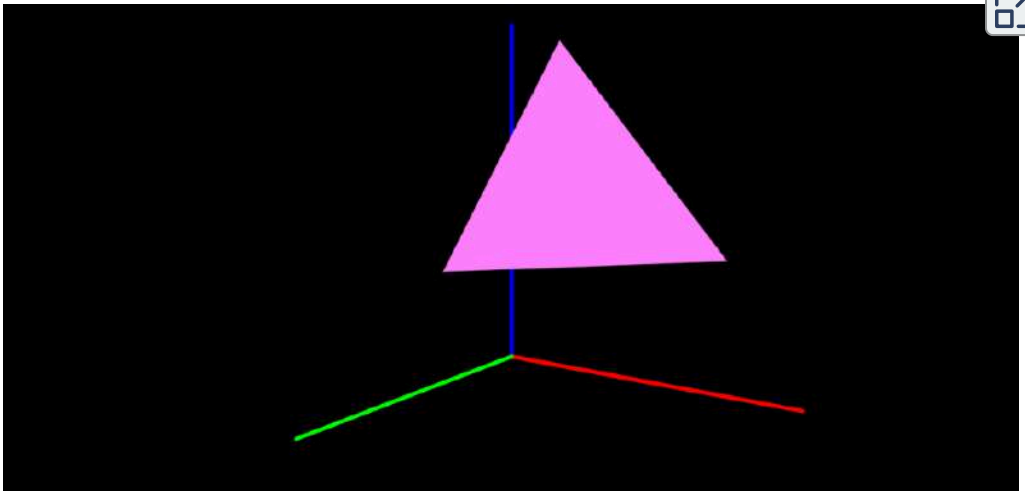
El panel de configuración de los triángulos en espacios 3D tiene el siguiente aspecto:



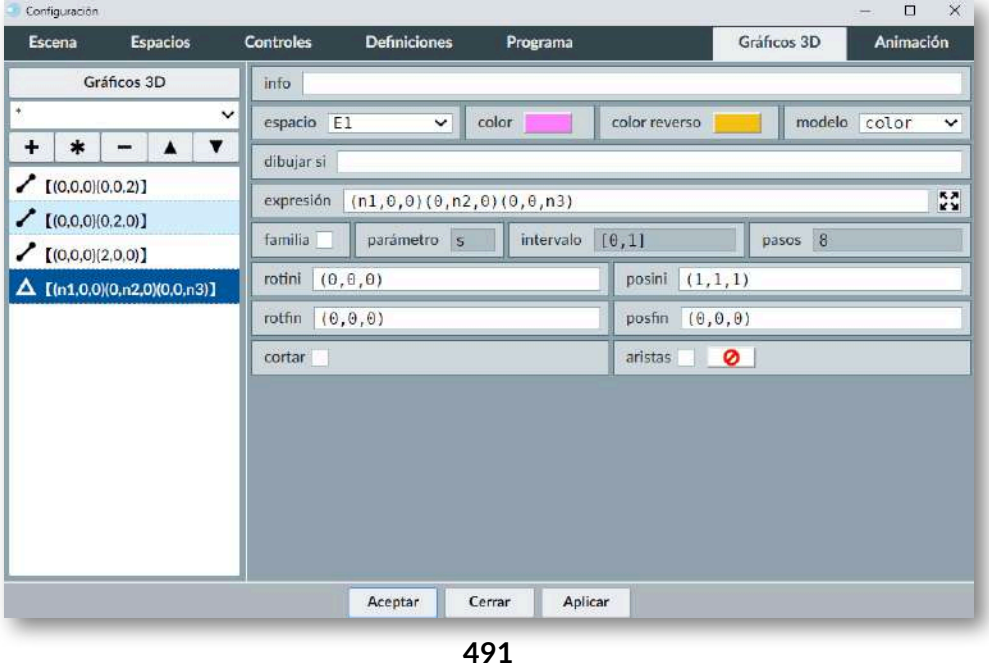
Parámetro característico presente en los gráficos 3D tipo triángulo es el siguiente:

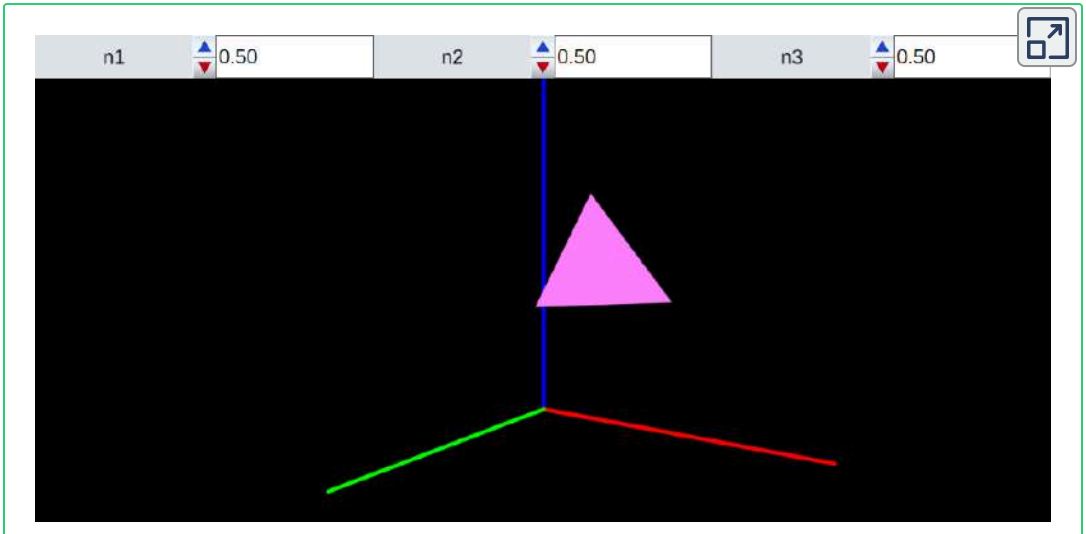
- **Expresión:** Campo de texto cuyo contenido debe ser las coordenadas de tres puntos del espacio, con la forma (X_1,Y_1,Z_1) $(X_2,Y_2,Z_2)(X_3,Y_3,Z_3)$, donde X_i, Y_i, Z_i (para i = 1, 2, 3) pueden ser expresiones numéricas. Cada terna representa un vértice del triángulo en el espacio tridimensional.

Incluimos la escena que contiene el triángulo ejemplificado en la introducción del apartado; además, se han dejado tres segmentos como referencia.

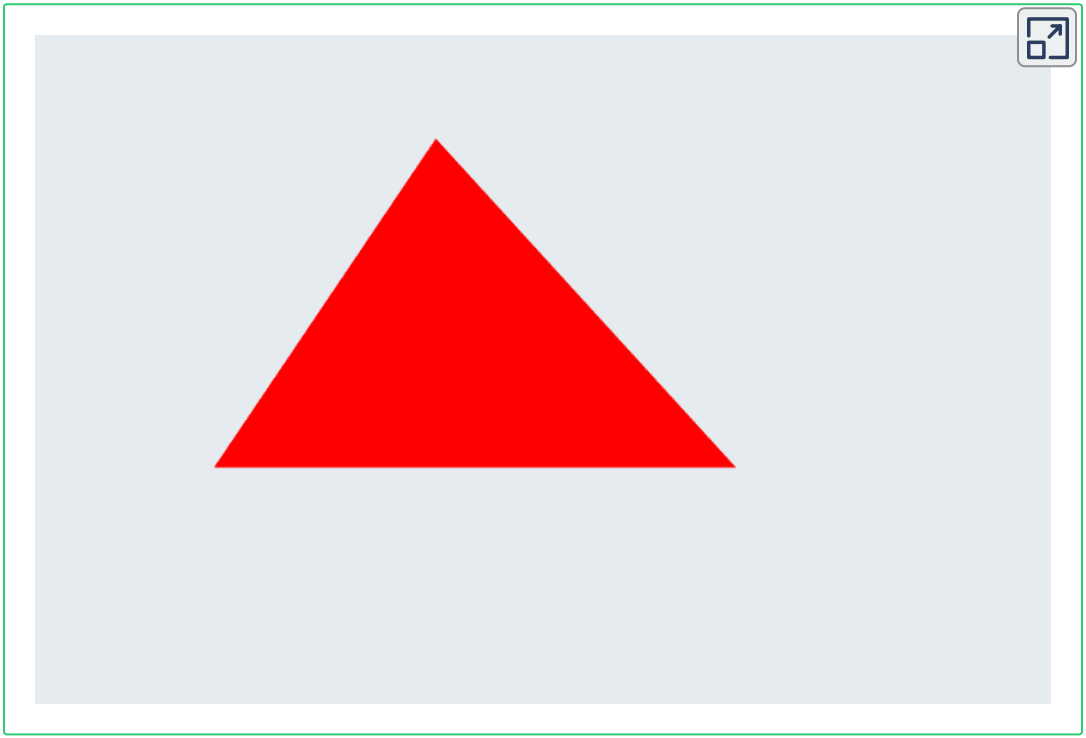


En el siguiente ejemplo mostramos un triángulo en el que podemos controlar la longitud de sus lados mediante pulsadores.





Incluimos otra escena en la que se ha animado el movimiento de un triángulo, permitiendo apreciar su geometría al realizar un giro completo de 360º.

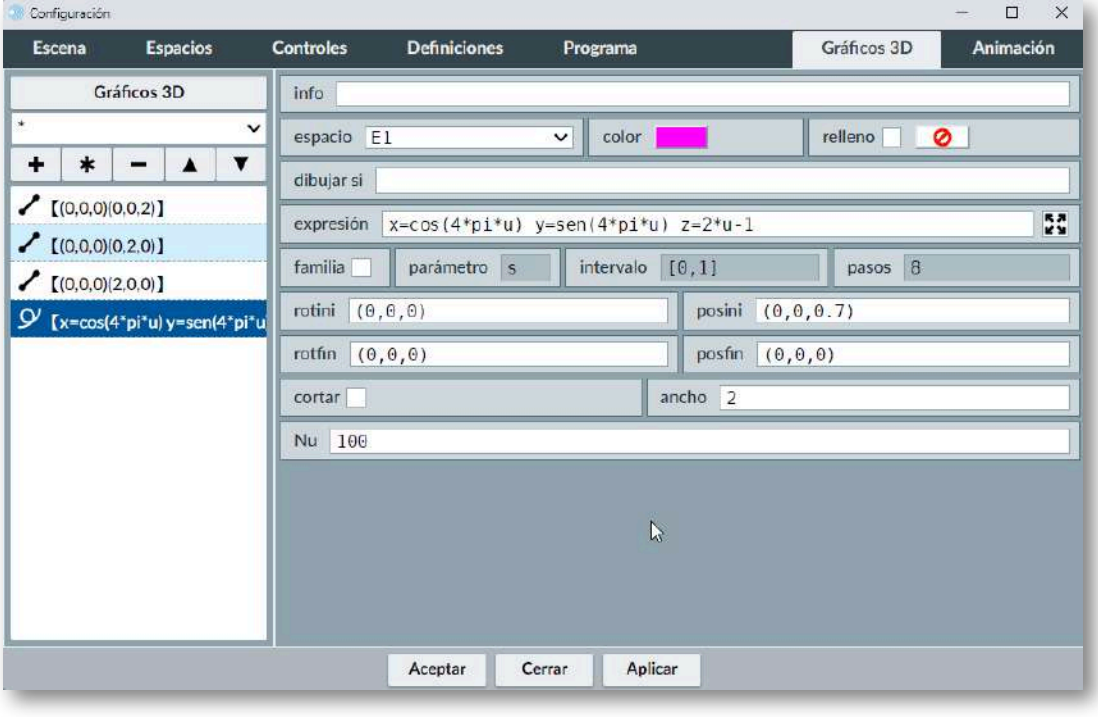


6.9 Gráfico 3D Curva

Es posible incorporar curvas en 3D, es decir, representar una función bidimensional dentro de un espacio tridimensional. Las coordenadas de cada punto de la curva se expresan en función de un parámetro **u**: x = x(u), y = y(u), z = z(u).

Es similar al gráfico curva en dos dimensiones, con la diferencia de que en el campo **expresión** se deben indicar tres expresiones —una para x, otra para y y otra para z— separadas por espacios. Estas expresiones dependen del parámetro **u**, que toma valores continuos en el intervalo [0,1].

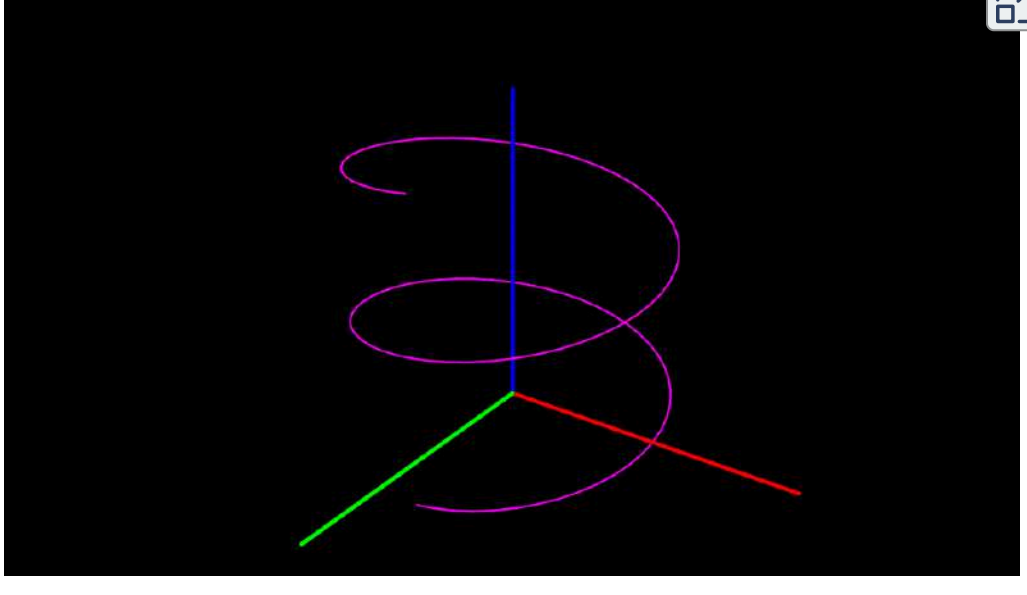
El panel de configuración de las curvas en espacios 3D tiene el siguiente aspecto:



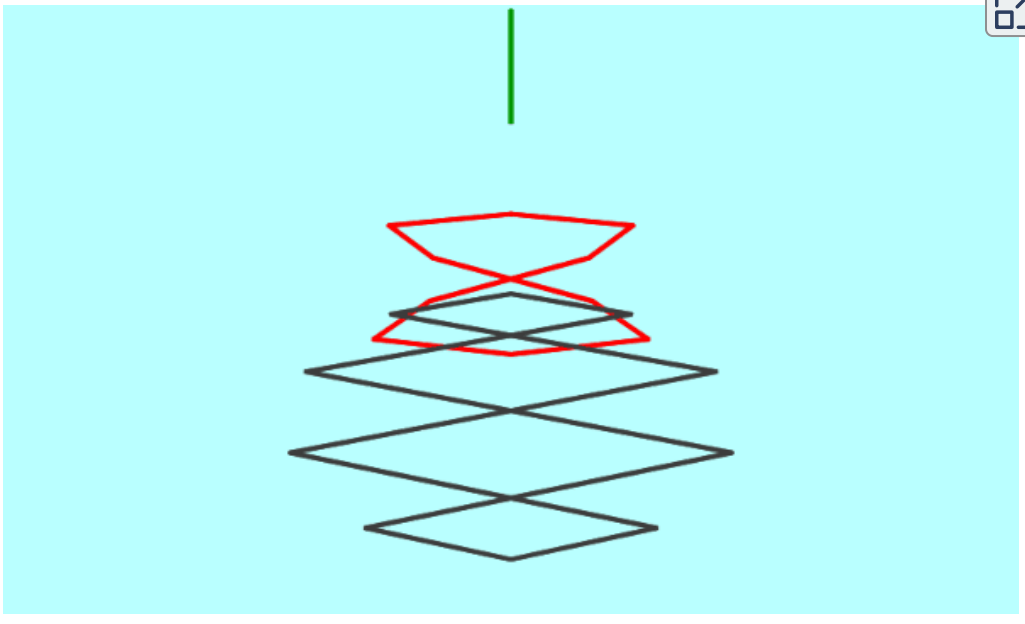
Parámetros característicos presentes en los gráficos 3D tipo curva son los siguientes:

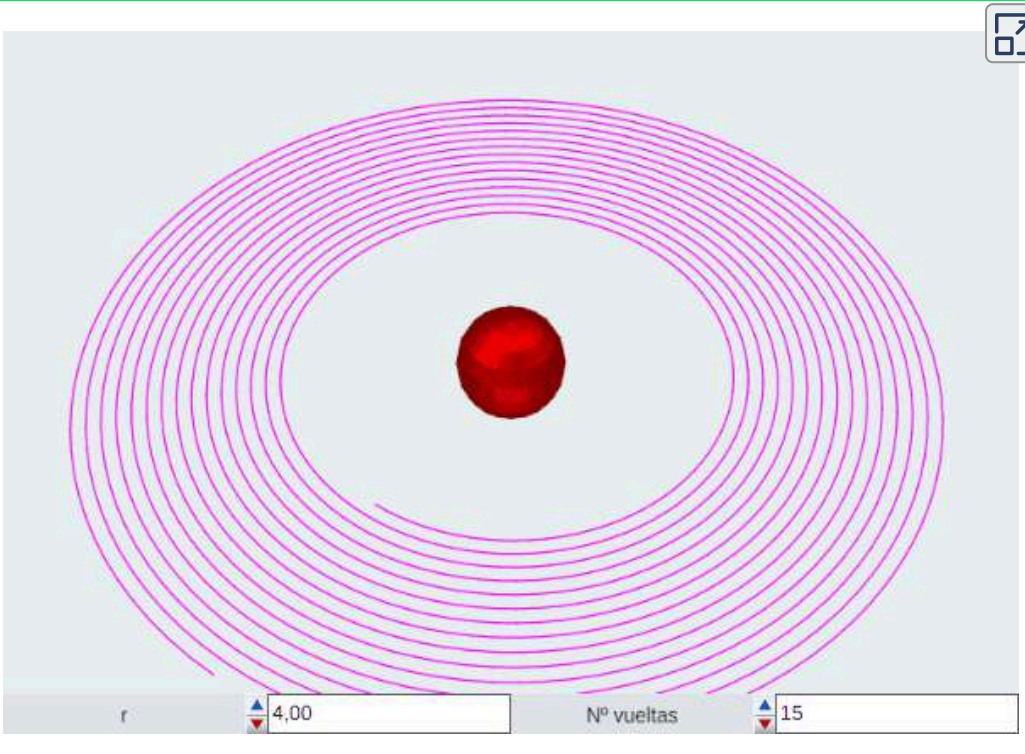
- **Expresión:** Campo de texto cuyo contenido debe tener la forma x = X(u) y = Y(u) z = Z(u), donde X, Y y Z son expresiones numéricas dependientes del parámetro **u**. La curva se dibuja como una poligonal de **Nu** lados, con **Nu + 1** vértices dados por los puntos ($X_{(i/Nu)}$, $Y_{(i/Nu)}$, $Z_{(i/Nu)}$) para i = 0, ..., Nu. Antes de definir x, y y z pueden declararse variables intermedias que se utilicen únicamente en los cálculos necesarios para representar la curva.
- **Nu:** Campo de texto que indica el número de intervalos en los que se divide el segmento unitario [0,1] para trazar la curva. Dado que las curvas se definen paramétricamente, siempre se utiliza **u** como parámetro.
- **Ancho:** Grosor de los segmentos de la curva.

Incluimos la escena que contiene la curva ejemplificada en la introducción del apartado; además, se han dejado tres segmentos como referencia.



Mostramos dos ejemplos más de curvas en espacios 3D.




r
4,00
Nº vueltas
15

6.10 Gráfico 3D Cara

Las caras son superficies bidimensionales dentro de espacios tridimensionales.

El panel de configuración de una cara permite:

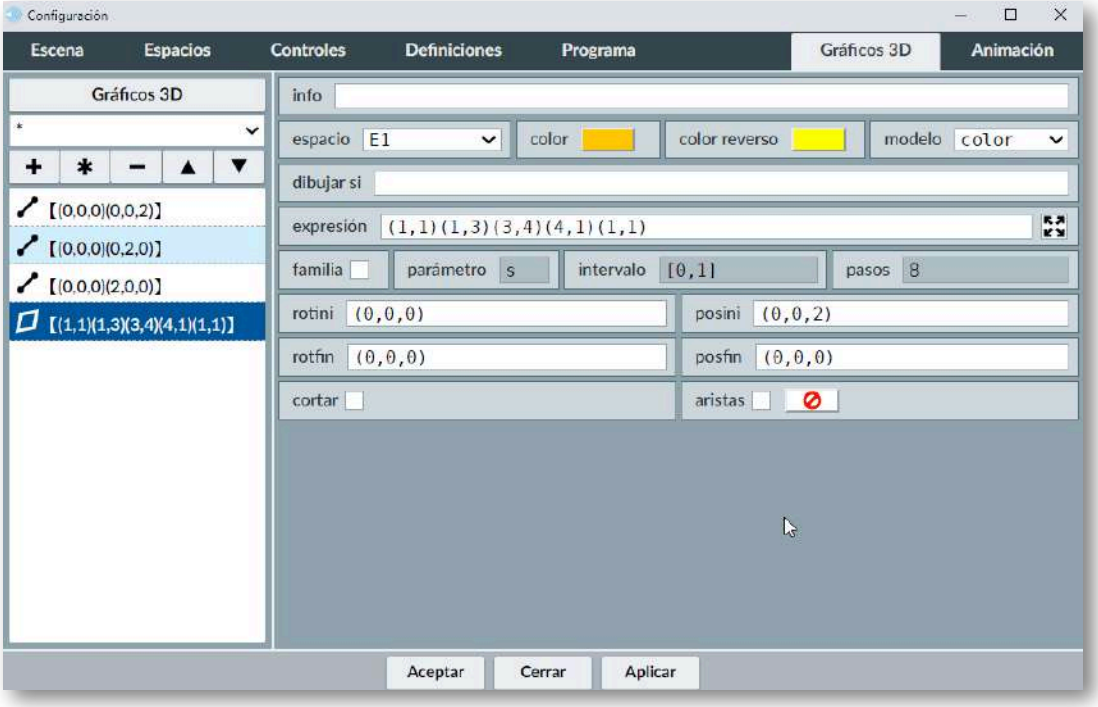
- Definir los vértices de la cara en el campo **expresión**. Por defecto, cada vértice se introduce como un par ordenado (x, y), correspondiente a dos dimensiones.
- Orientar la cara tridimensionalmente mediante los campos **rotini**, **posini**, **rotfin** y **posfin**. Para más detalles sobre estos campos, consulte el apartado sobre controles comunes a gráficos 3D.
- Opcionalmente, introducir los vértices directamente en coordenadas tridimensionales (x, y, z). Esto permite omitir la manipulación de los campos de rotación y traslación, simplificando la orientación de la cara.

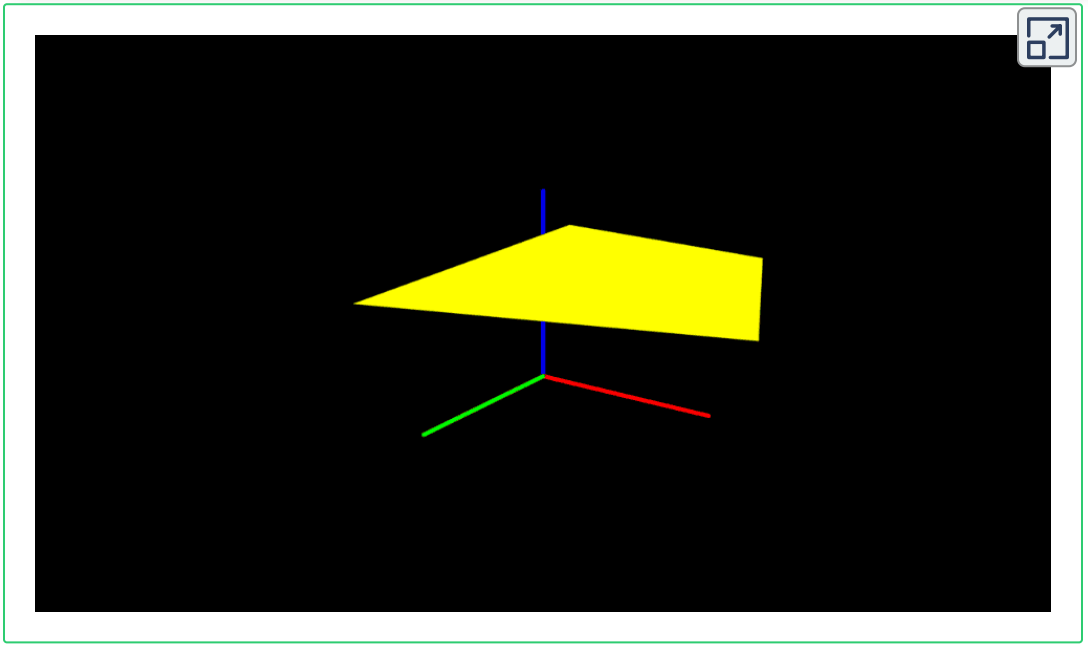
Parámetro característico presente en los gráficos 3D tipo cara es el siguiente:

- **Expresión:** Es un campo de texto donde se introducen las coordenadas de un polígono en un plano 3D, con la forma $(X_1,Y_1,Z_1)(X_2,Y_2,Z_2)...(X_n,Y_n,Z_n)$, donde X_i, Y_i, Z_i para i=1,...,n son expresiones numéricas que representan las coordenadas de cada vértice del polígono.

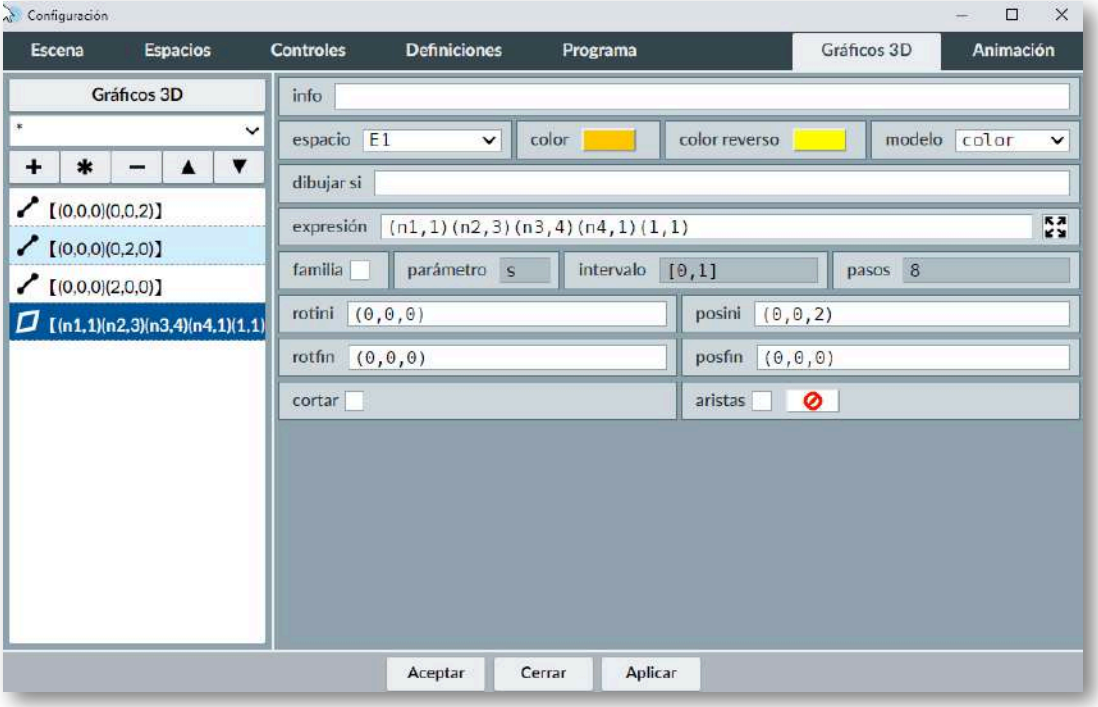
Es importante que los vértices definidos en 3D sean coplanares; de lo contrario, la cara no se dibujará correctamente, ya que dejaría de ser una superficie plana.

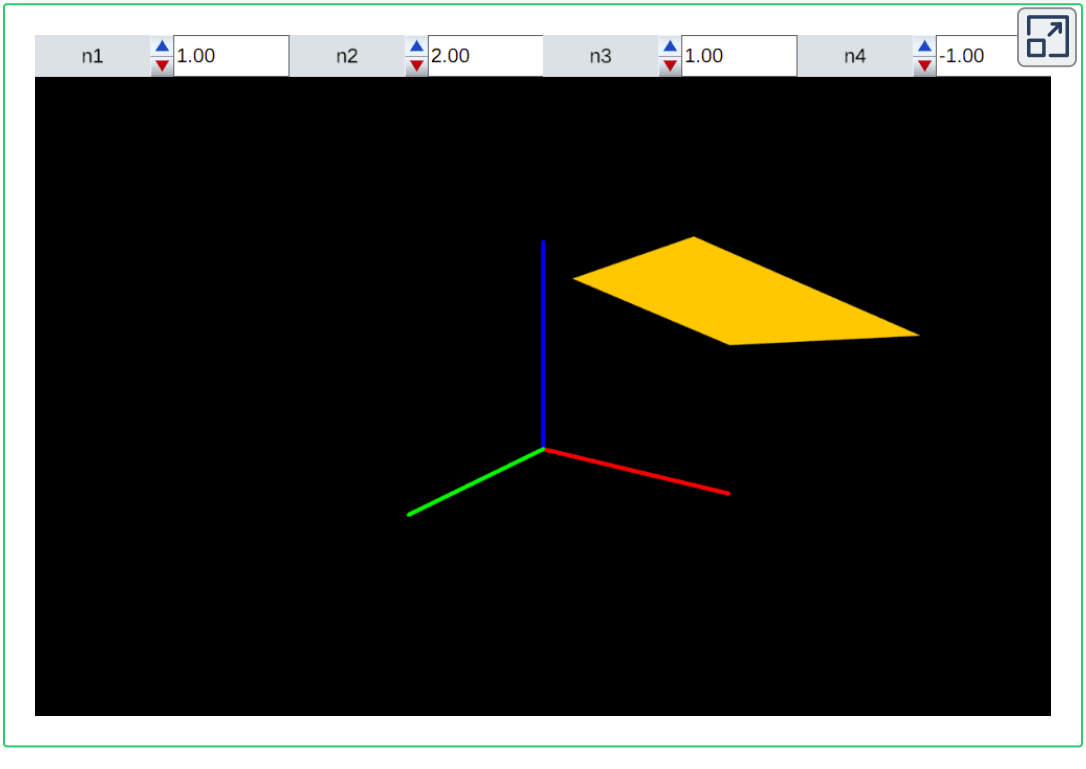
Mostramos una escena con una cara definida en el mismo plano para garantizar su correcto funcionamiento; además, se han incluido tres segmentos como referencia.





En el siguiente ejemplo mostramos una cara en el que podemos controlar sus dimensiones mediante pulsadores.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Gráficos 3D
[(0,0,0)(0,0,2)]
[(0,0,0)(0,2,0)]
[(0,0,0)(2,0,0)]
[(n1,1)(n2,3)(n3,4)(n4,1)(1,1)
info
espacio E1
color
color reverso
modelo color
dibujar si
expresión (n1,1)(n2,3)(n3,4)(n4,1)(1,1)
familia
parámetro s
intervalo [0,1]
pasos 8
rotini (0,0,0)
posini (0,0,2)
rotfin (0,0,0)
posfin (0,0,0)
cortar
aristas
Aceptar
Cerrar
Aplicar

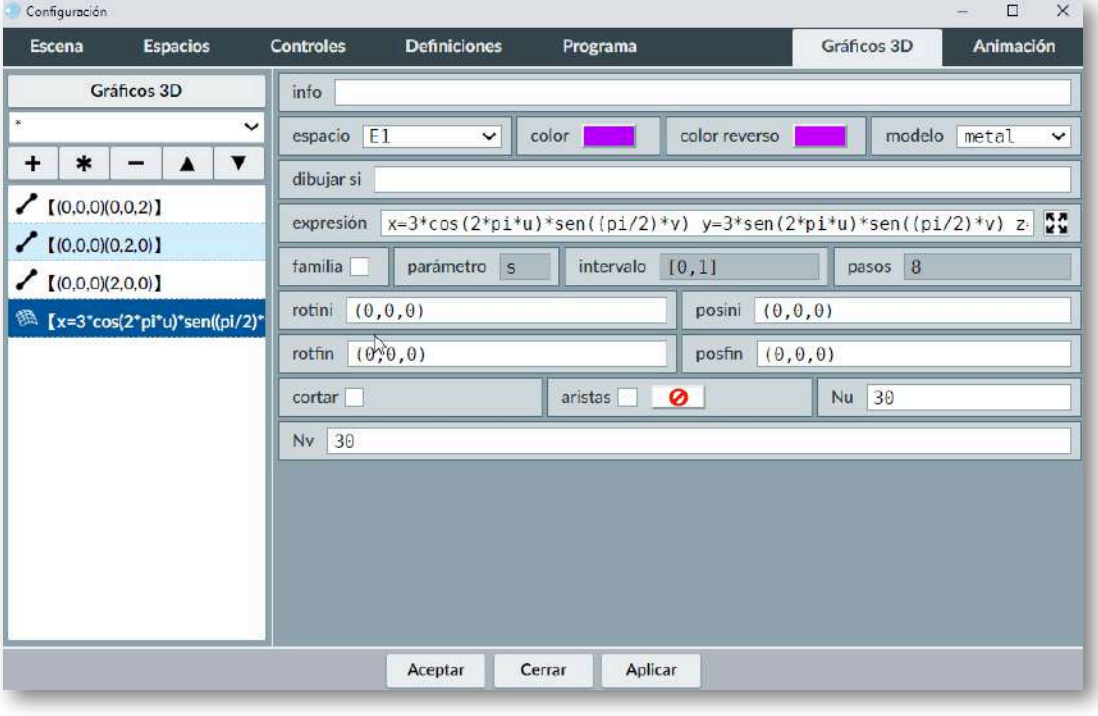

n1 1.00
n2 2.00
n3 1.00
n4 -1.00

6.11 Gráfico 3D Superficie

Las superficies son extensiones bidimensionales en espacios tridimensionales, definidas de manera paramétrica mediante dos parámetros, u y v, que varían entre 0 y 1. Cada punto de la superficie se calcula como función de u y v, y el número de particiones de estos parámetros se establece en los campos Nu y Nv, respectivamente.

El campo expresión del gráfico debe incluir una expresión para x(u,v), otra para y(u,v) y otra para z(u,v), separadas por espacios (expresiones parametrizadas).

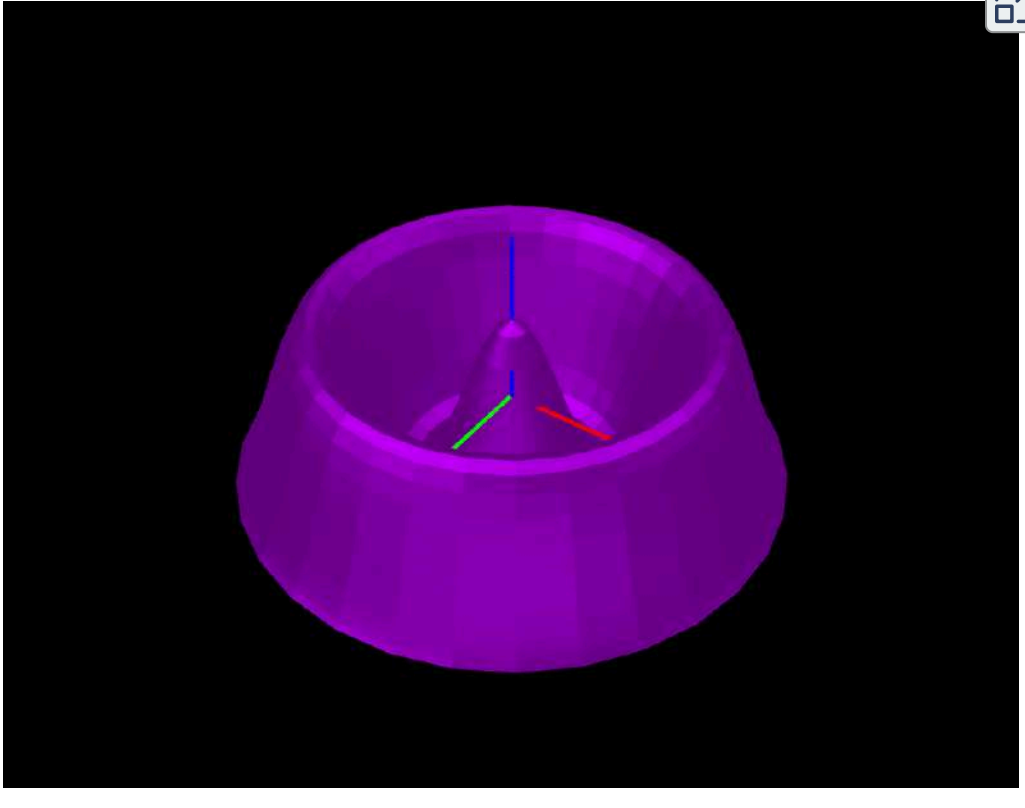
El panel de configuración de las superficies en espacios 3D tiene el siguiente aspecto:



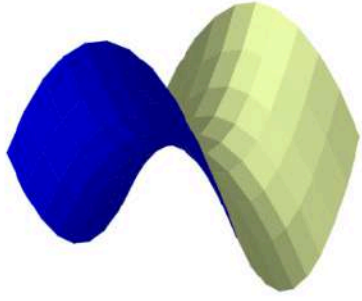
Parámetro característico presente en los gráficos 3D tipo superficie es el siguiente:

- **Expresión:** Este gráfico permite definir una superficie a partir de dos parámetros, u y v, que toman valores continuos entre 0 y 1. El número de particiones de estos parámetros se establece mediante los campos Nu y Nv. En el campo expresión se deben introducir las expresiones para x(u,v), y(u,v) y z(u,v), separadas por espacios.

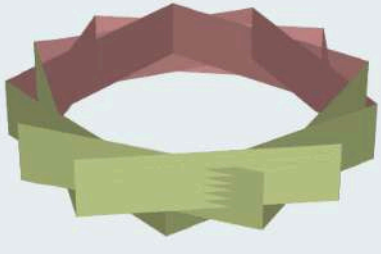
Incluimos la escena que contiene la superficie ejemplificada en la introducción del apartado; además, se han dejado tres segmentos como referencia.



Mostramos dos ejemplos de superficies en espacios 3D, en los que se han incorporado pulsadores que permiten interactuar con ellas.



a 0 b 0 c 0

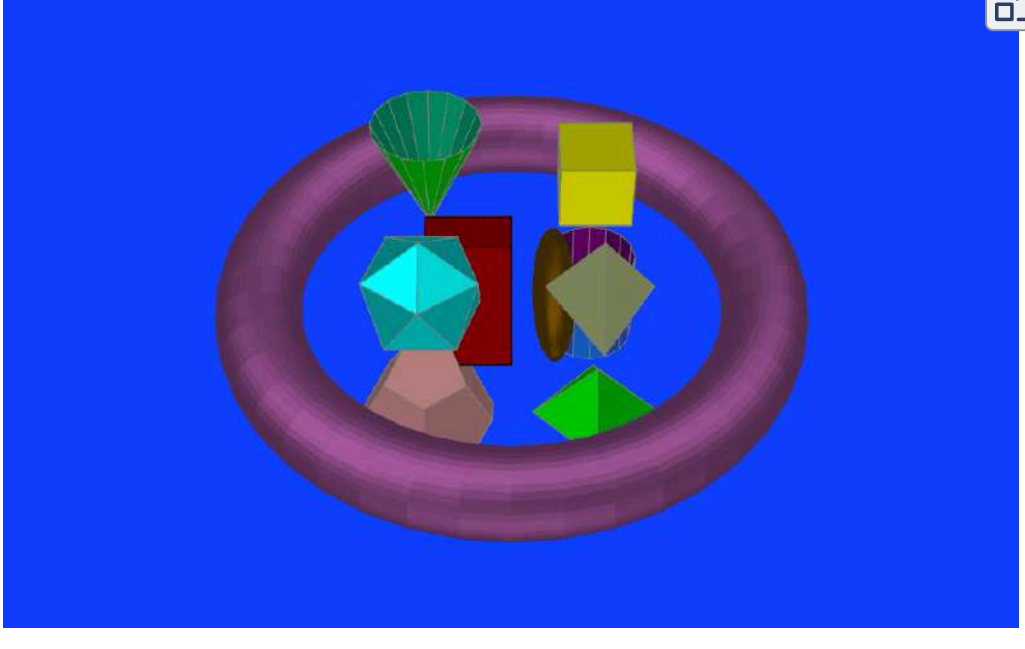


Ángulo 0 tamaño 3,00

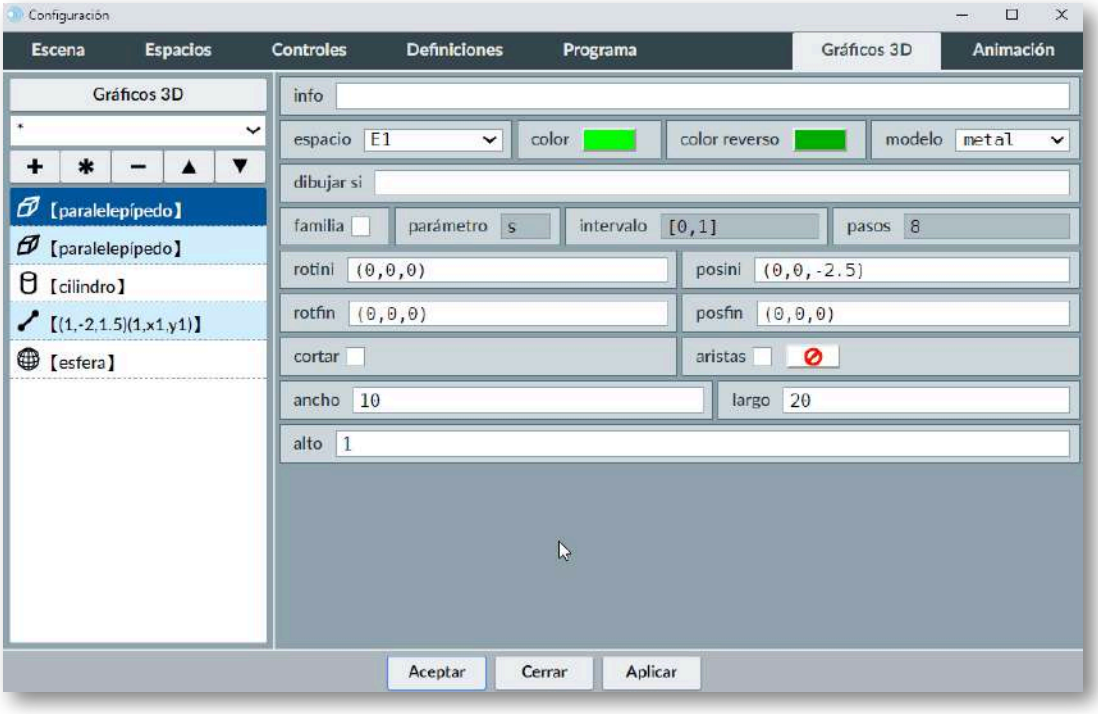
6.12 Gráficos 3D. Cuerpos geométricos

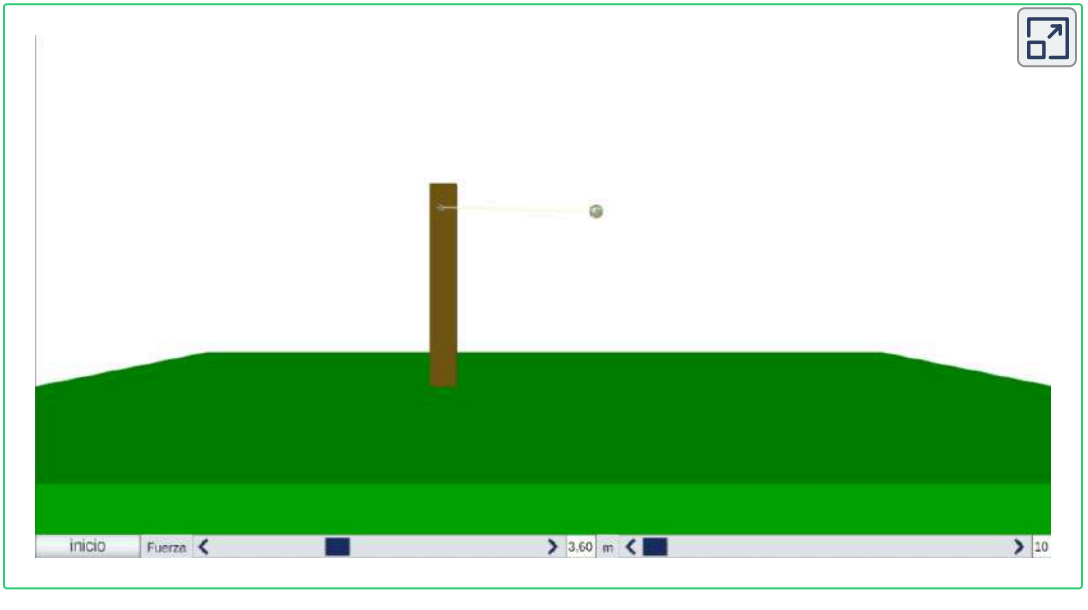
El manejo de curvas, superficies, polígonos y caras permite dibujar prácticamente cualquier figura geométrica. Sin embargo, en las escenas también es posible insertar figuras tridimensionales predefinidas. Estas figuras dependen de parámetros como ancho, largo, alto, Nu y Nv, y su campo expresión funciona únicamente como identificador durante la edición.

En la siguiente imagen se muestra una escena con diversas figuras geométricas tridimensionales disponibles actualmente en DescartesJS.



En el siguiente ejemplo se muestra cómo construir una realidad física a partir de unos pocos cuerpos geométricos colocados adecuadamente, a los cuales se les ha dotado de movilidad mediante controles.

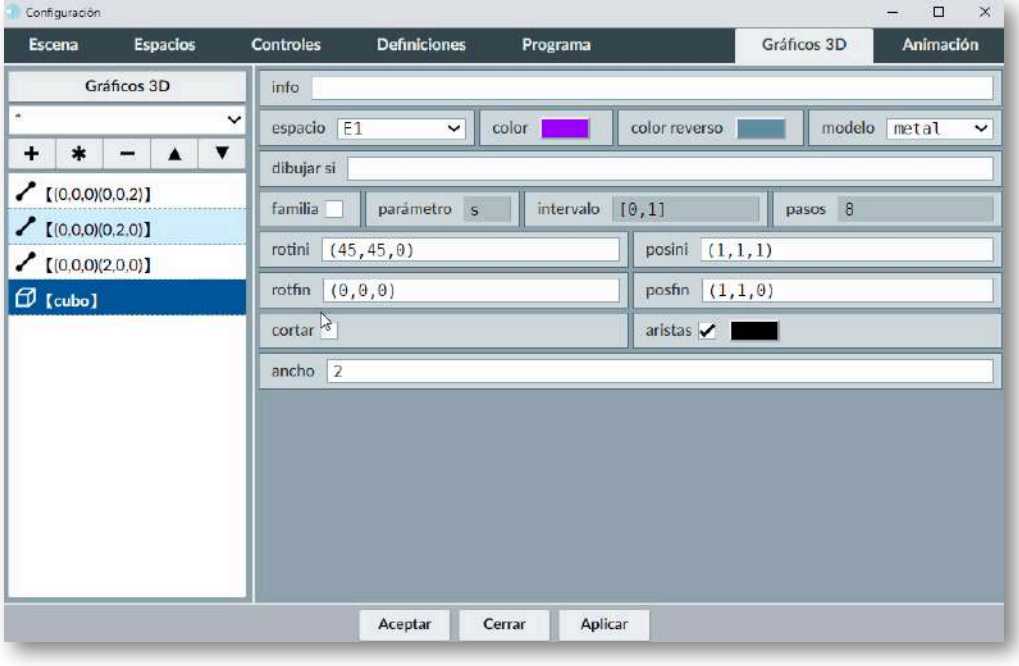


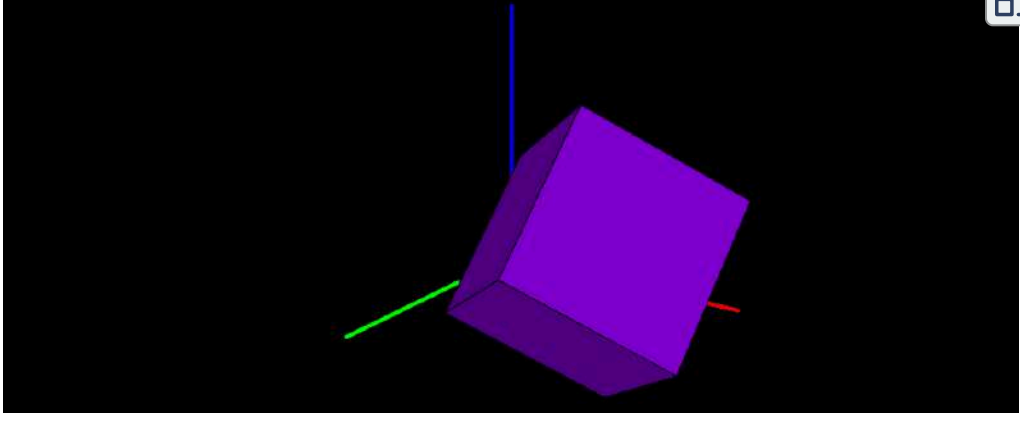


Mostramos a continuación los distintos cuerpos geométricos tridimensionales disponibles.

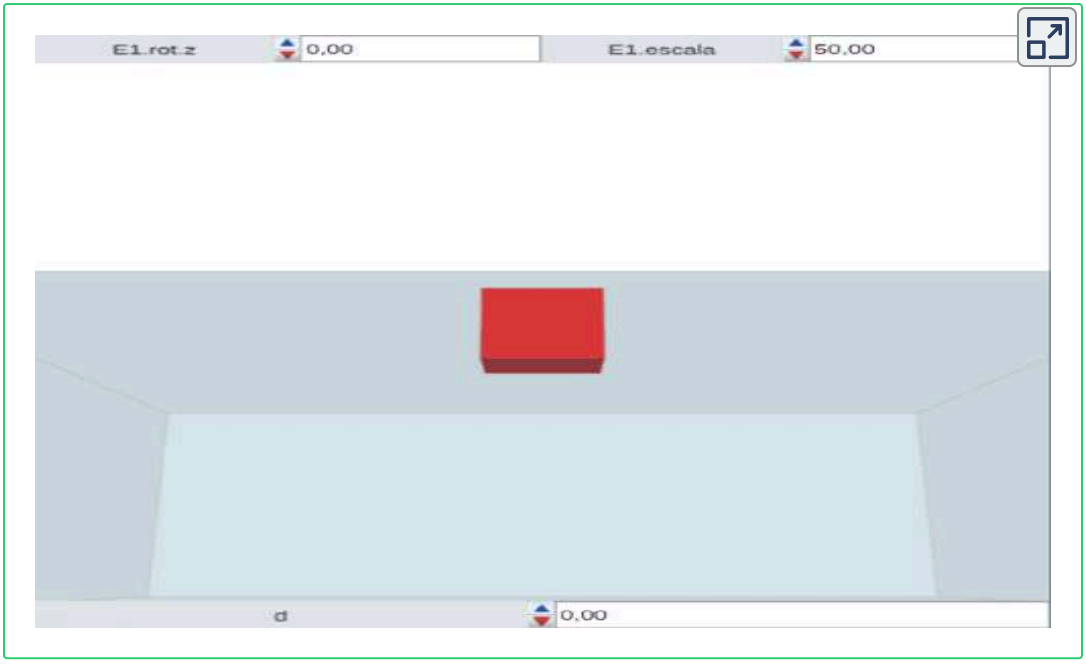
Gráfico 3D. Cubo

Es un gráfico en el que se define un campo **ancho** correspondiente a la longitud de la diagonal del cubo. El lado del cubo se obtiene dividiendo esta diagonal entre √3. Además, se puede reorientar y trasladar utilizando los campos **rotini**, **posini**, **rotfin** y **posfin**, tal como se describe en el apartado sobre controles comunes a los gráficos 3D.





Incluimos otra escena de ejemplo y un vídeo explicativo.



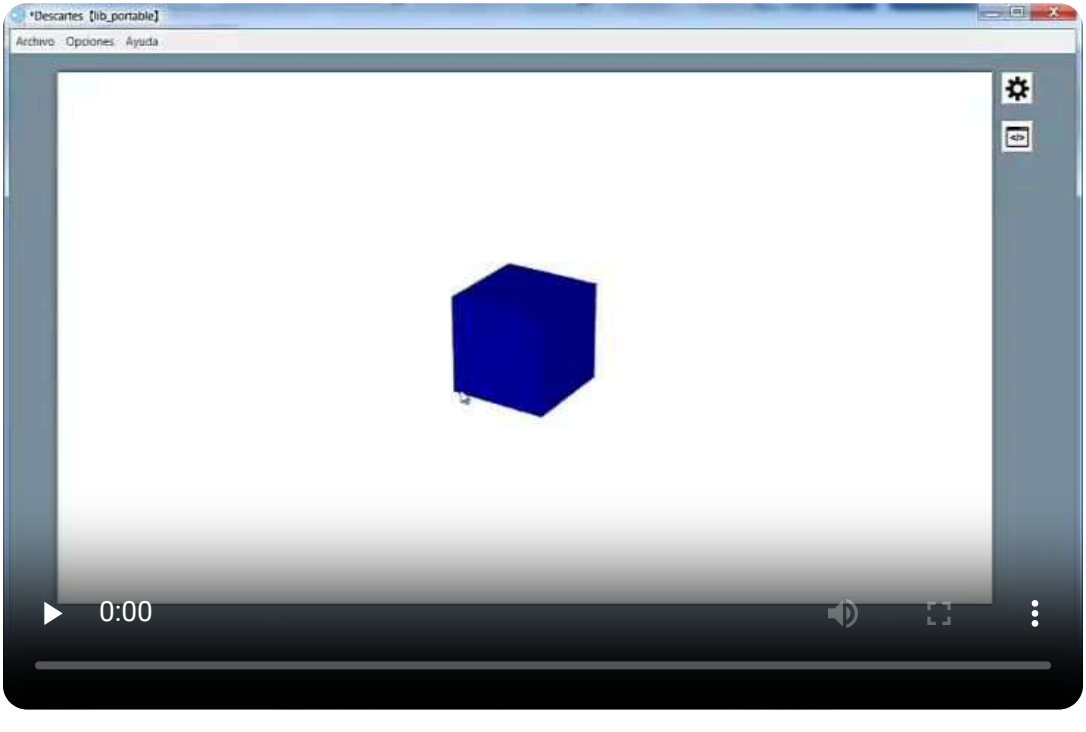
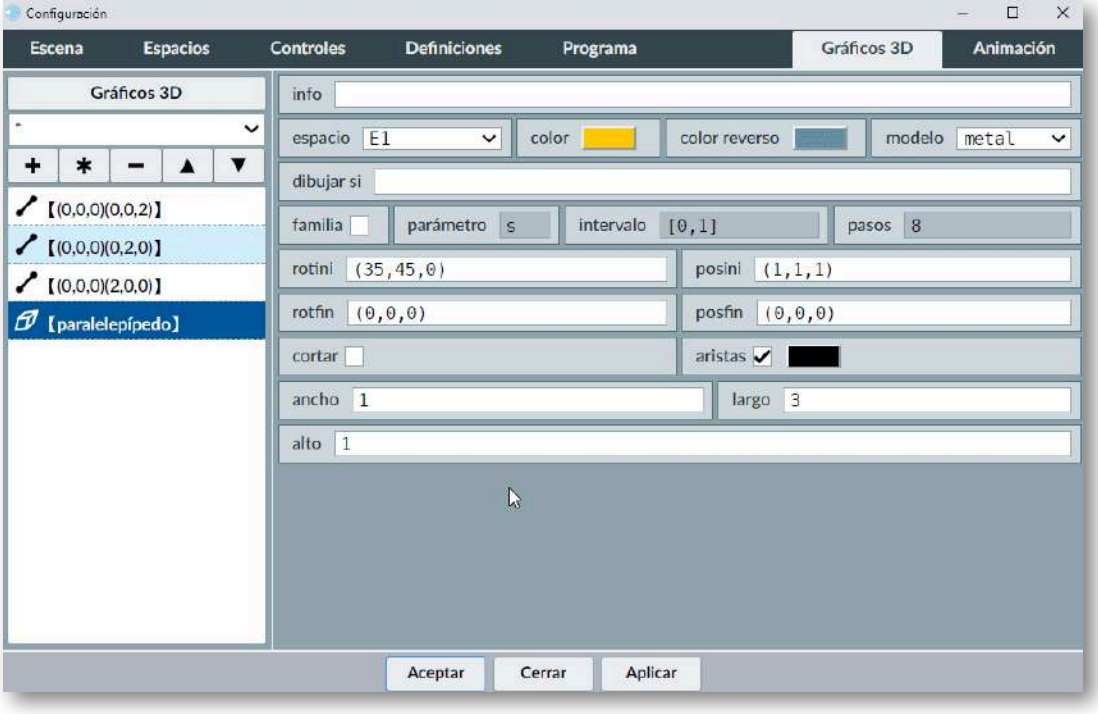


Gráfico 3D Paralelepípedo

Se trata de una figura geométrica tridimensional cuyo gráfico es similar al cubo, pero en este caso se definen tres dimensiones independientes: **ancho**, **largo** y **alto**.



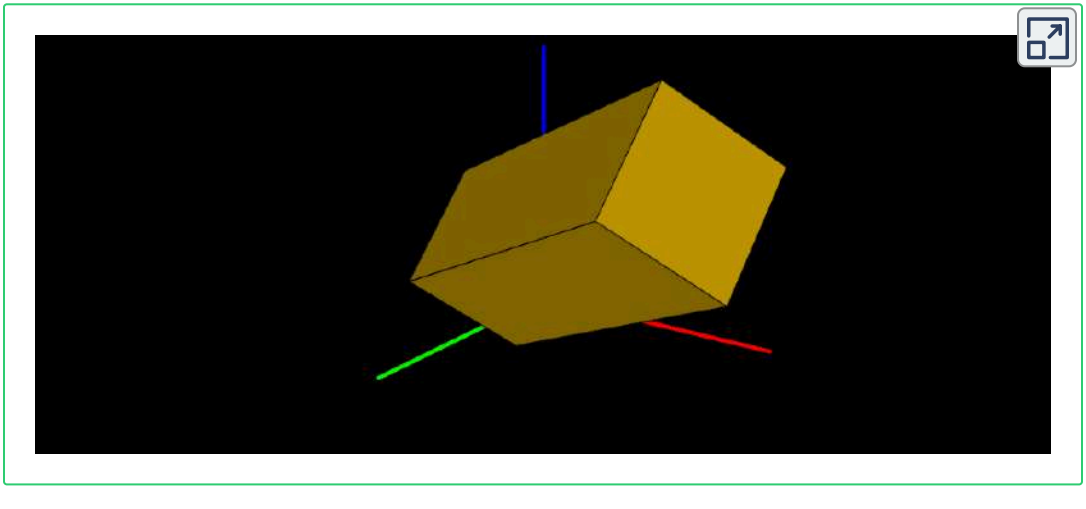
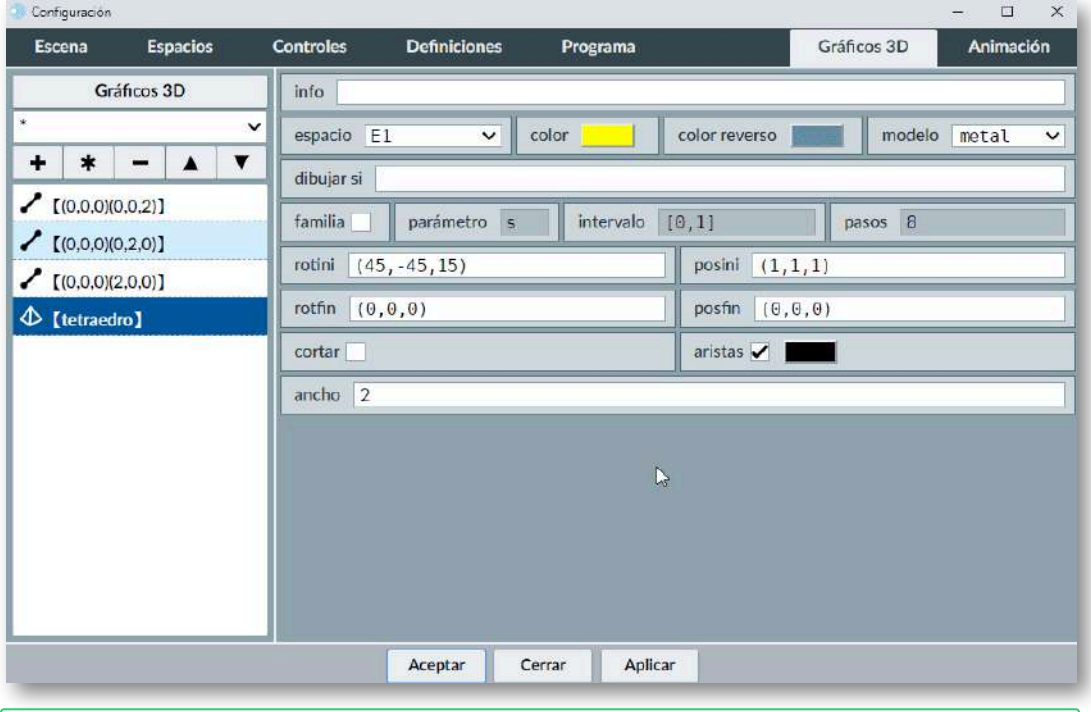


Gráfico 3D Tetraedro

Un tetraedro es un poliedro regular (sólido platónico) formado por cuatro caras que son triángulos equiláteros, seis aristas y cuatro vértices. Es la pirámide triangular más simple y convexa, con tres caras concurriendo en cada vértice.



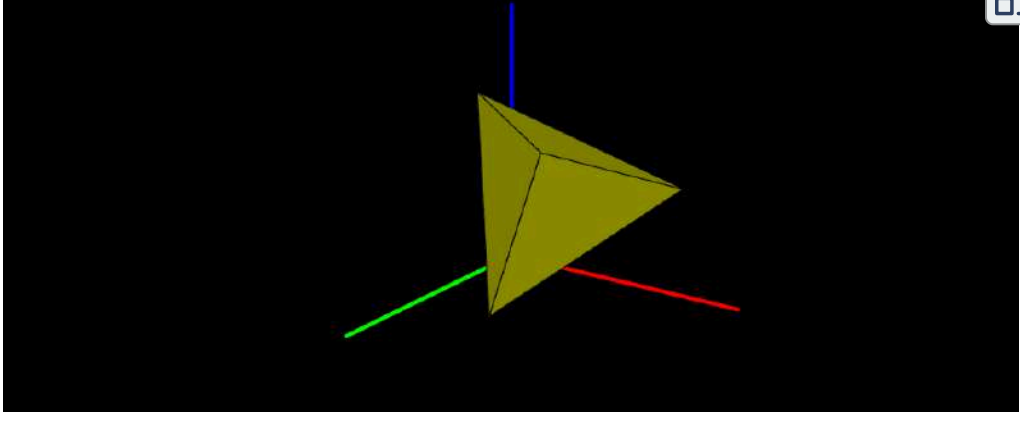
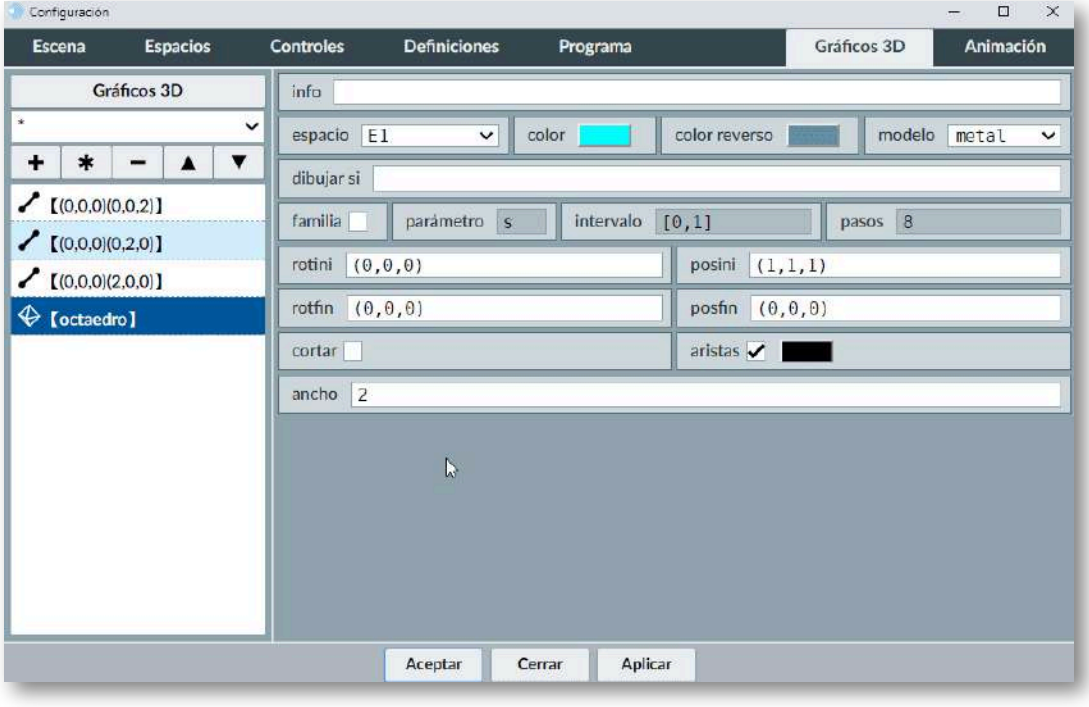
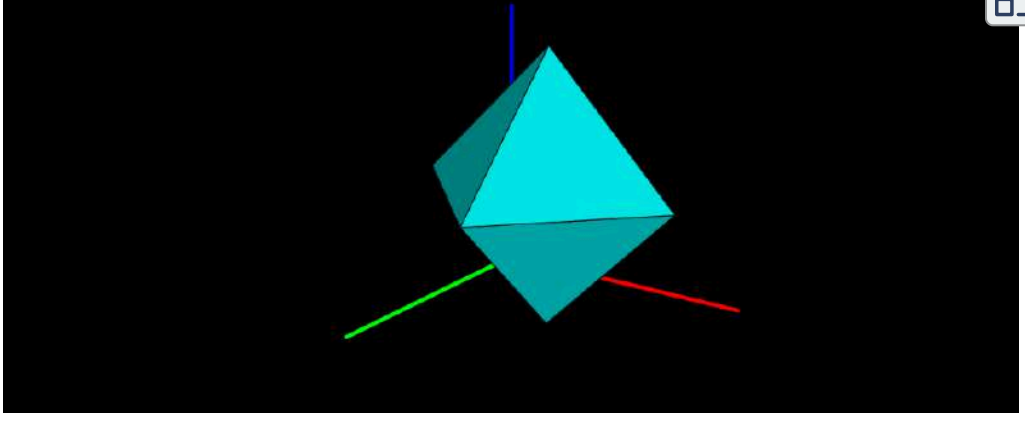


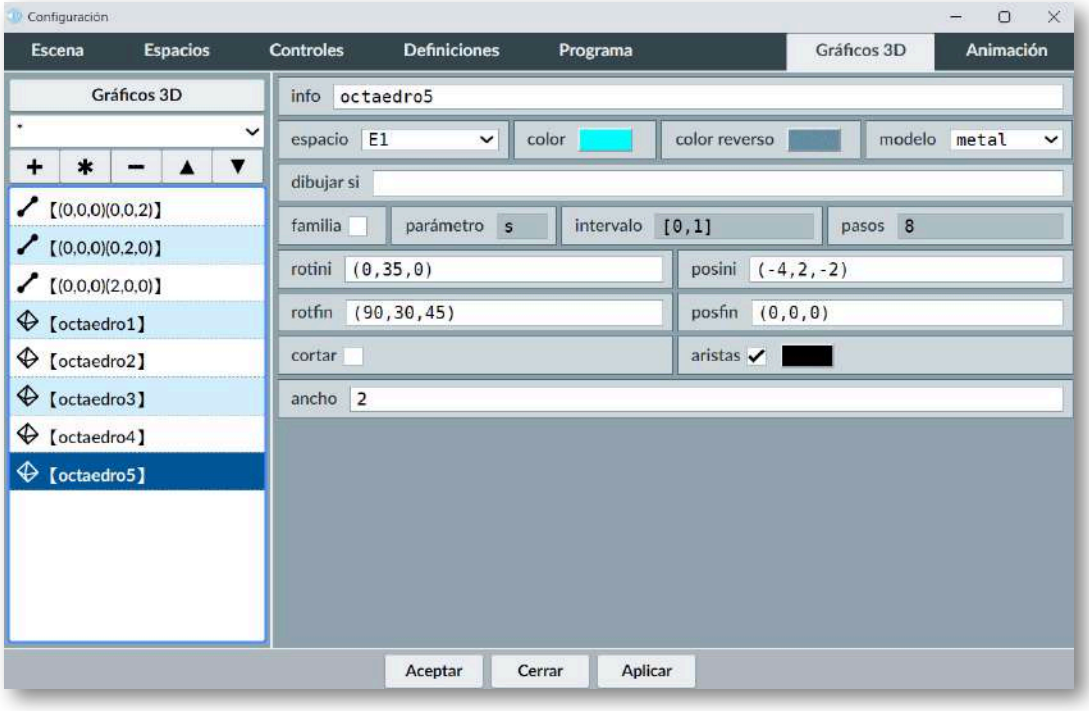
Gráfico 3D Octaedro

Un octaedro es un poliedro de ocho caras, siendo el octaedro regular un sólido platónico convexo formado por ocho triángulos equiláteros iguales, seis vértices y doce aristas. Sus tres diagonales se cortan en sus puntos medios.





Mostramos un ejemplo con varios octaedros ubicados en distintas posiciones y orientaciones.



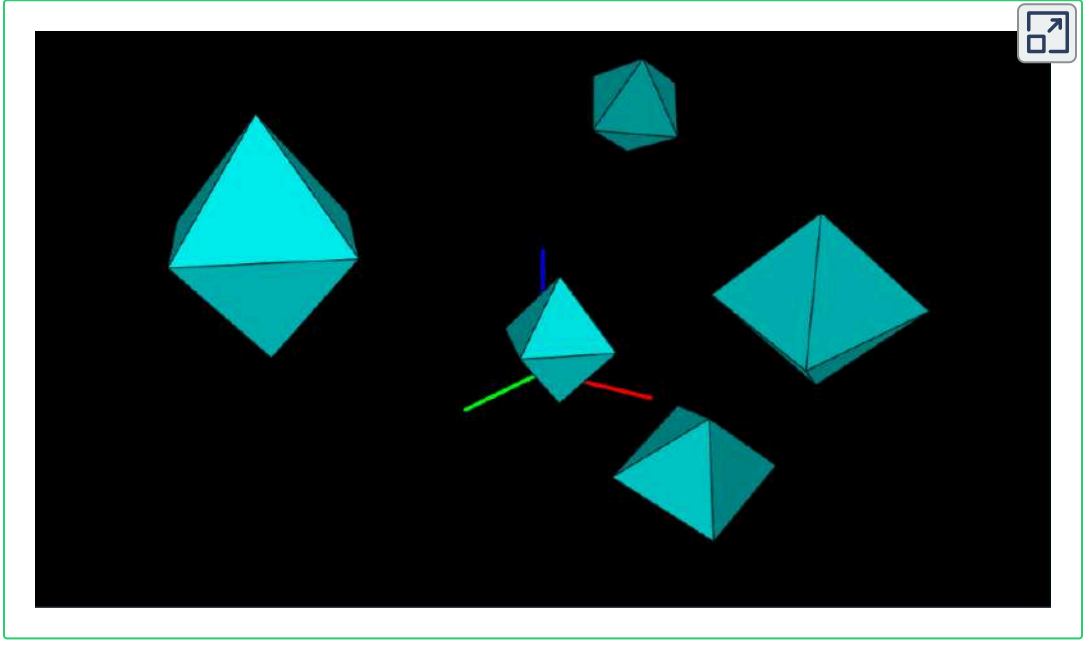
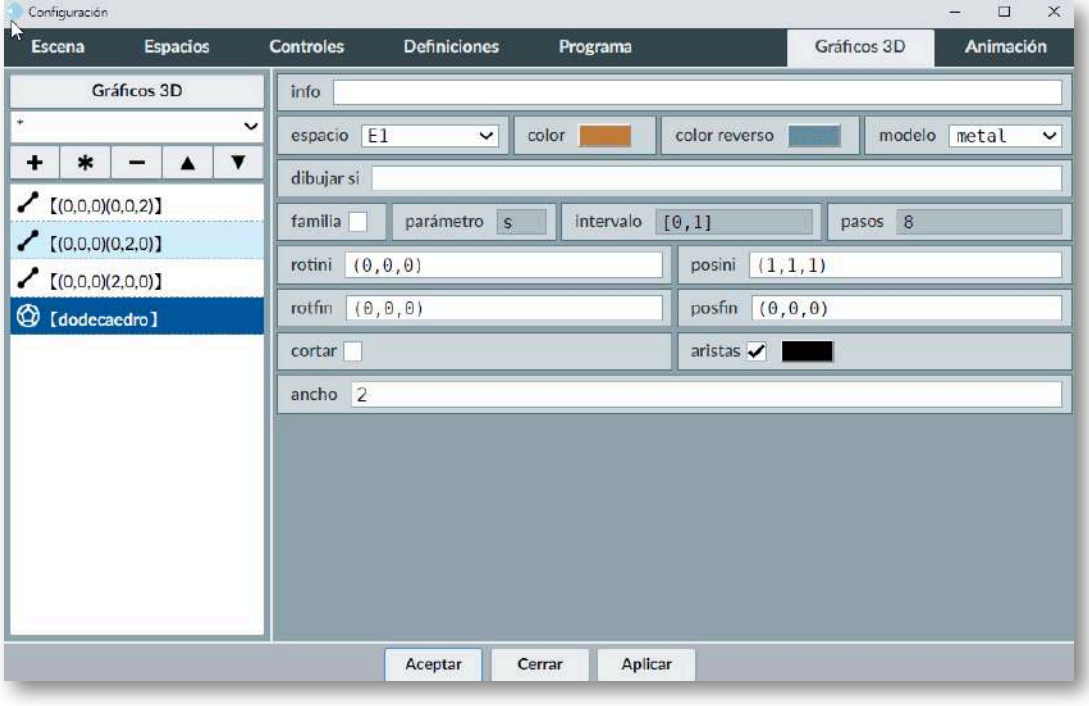


Gráfico 3D Dodecaedro

Un dodecaedro es un poliedro regular convexo compuesto por 12 caras pentagonales iguales, 30 aristas y 20 vértices, siendo uno de los cinco sólidos platónicos. Se caracteriza por su simetría, donde tres caras se encuentran en cada vértice.



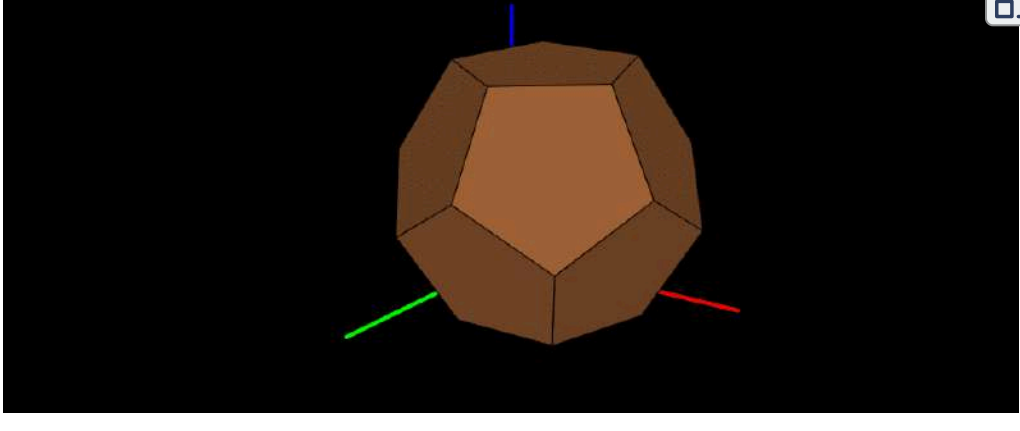
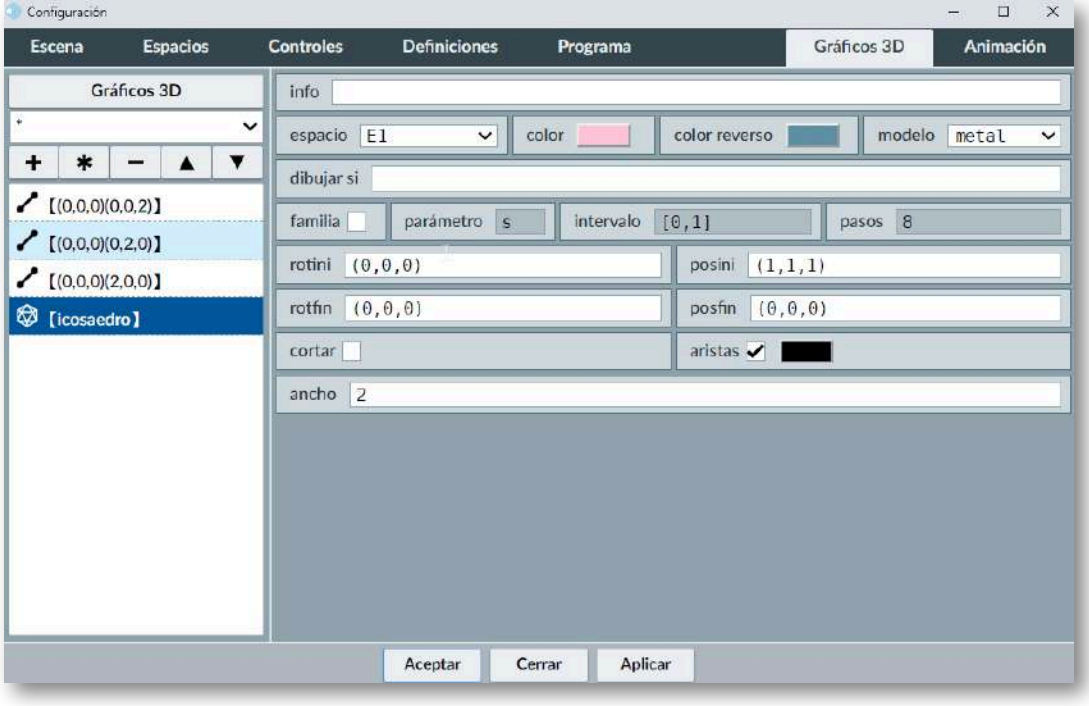


Gráfico 3D Icosaedro

El icosaedro es un poliedro regular de 20 caras formadas por triángulos equiláteros iguales, siendo uno de los cinco sólidos platónicos. Posee 12 vértices, 30 aristas y es convexo, caracterizándose por una alta simetría.



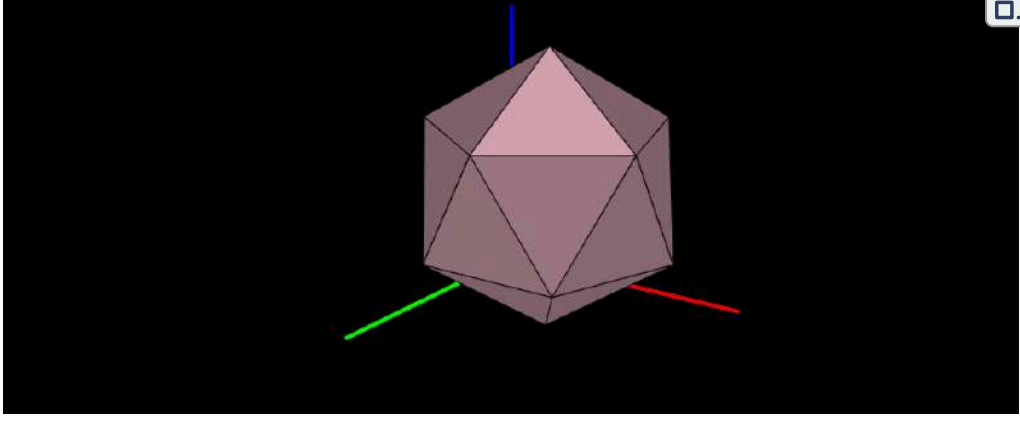
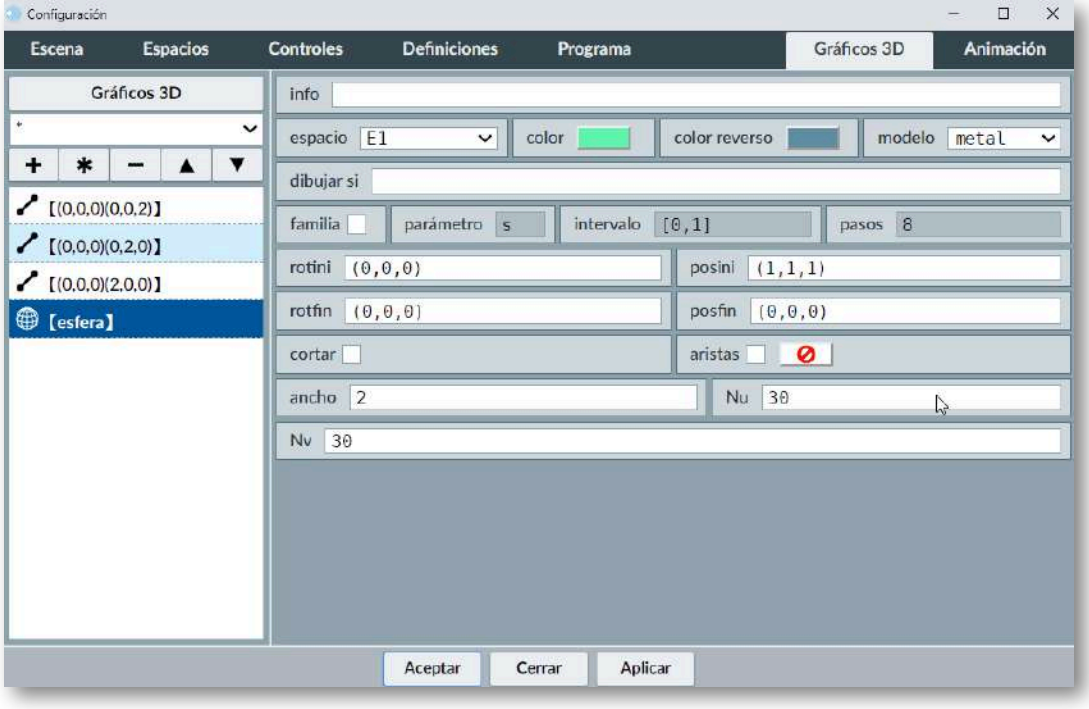
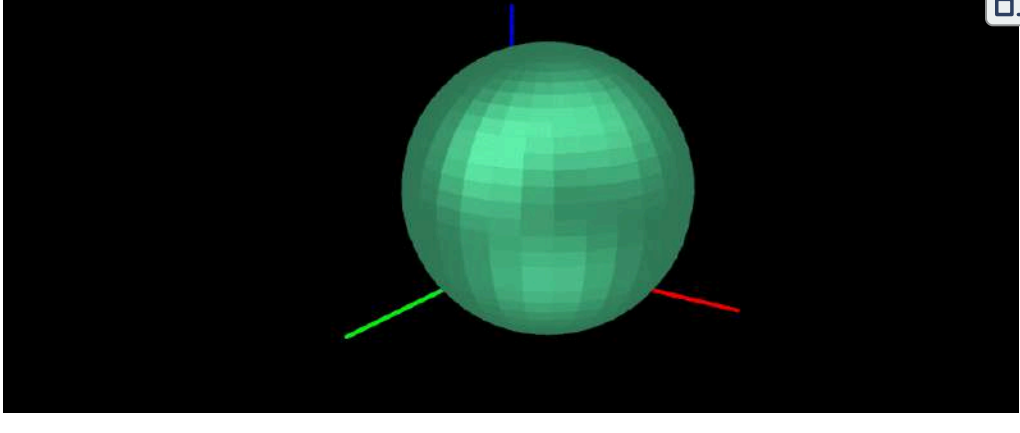


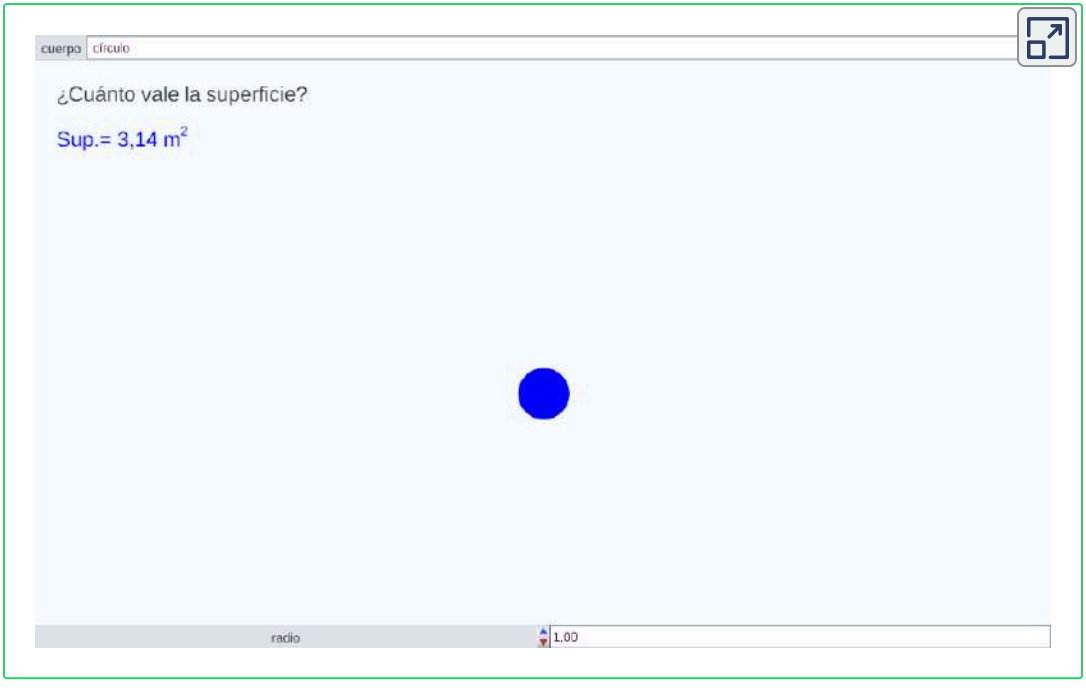
Gráfico 3D Esfera

Una esfera geométrica es un cuerpo sólido tridimensional perfectamente simétrico, formado por todos los puntos en el espacio que equidistan de un punto interior fijo llamado centro. Esta distancia constante se conoce como radio.





Incluimos un par de escenas como ejemplos de uso de esferas.



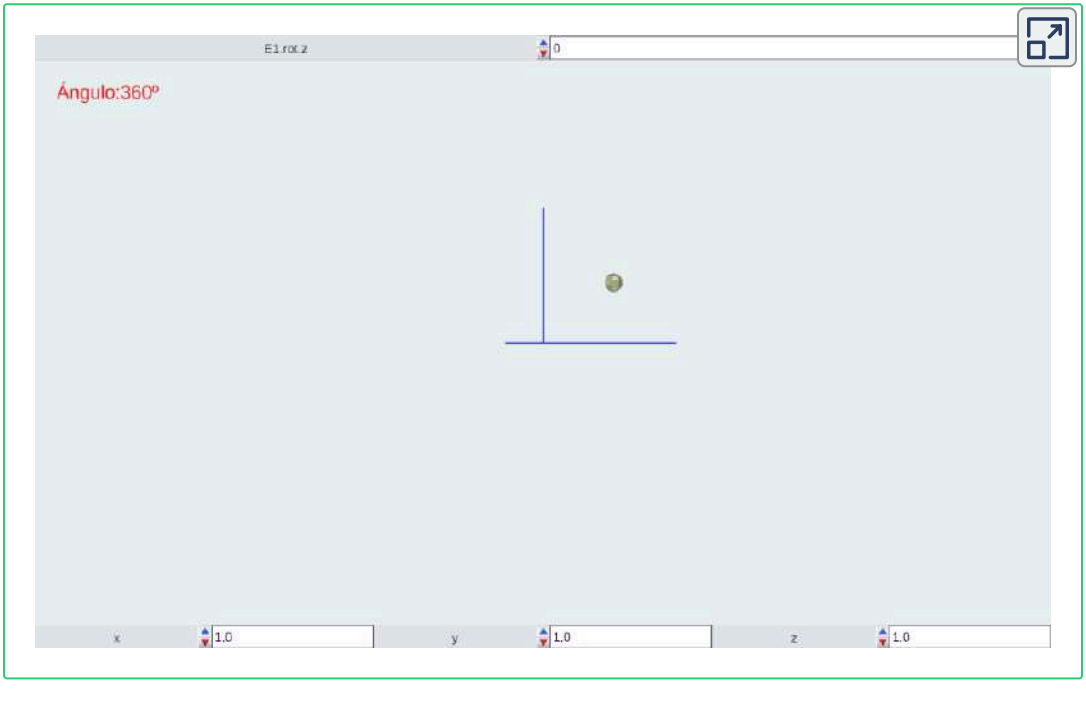
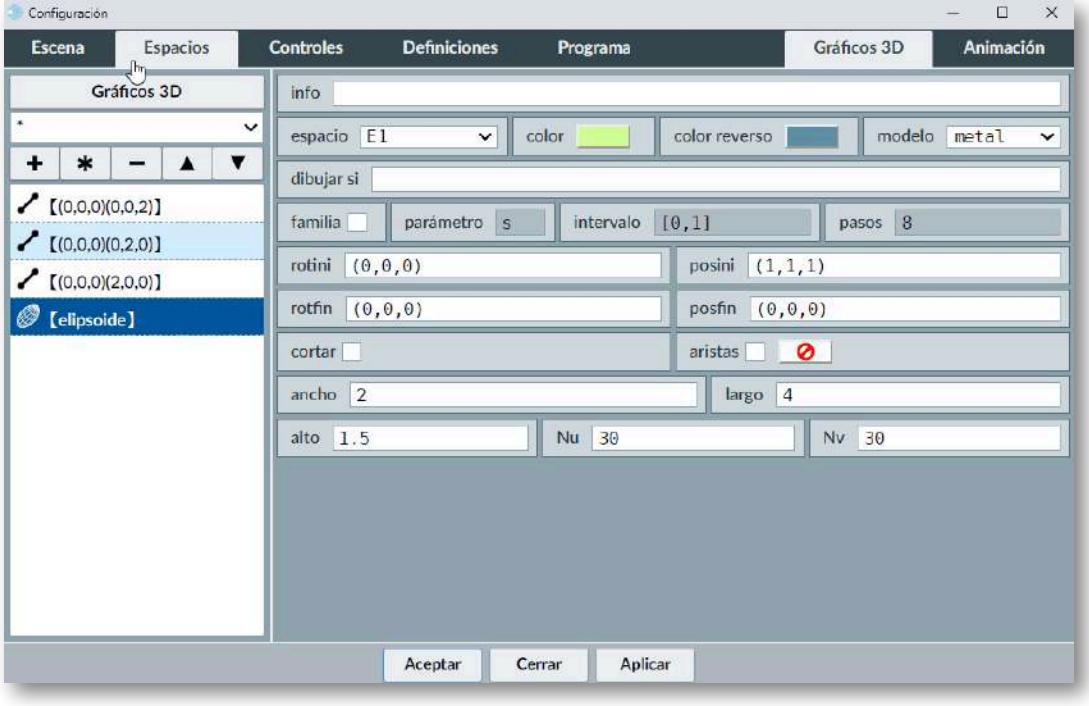


Gráfico 3D Elipsoide

Un elipsoide geométrico es una superficie curva cerrada en tres dimensiones cuyas secciones planas son todas elipses o círculos. Es la generalización tridimensional de una elipse y funciona como una esfera deformada o achatada.



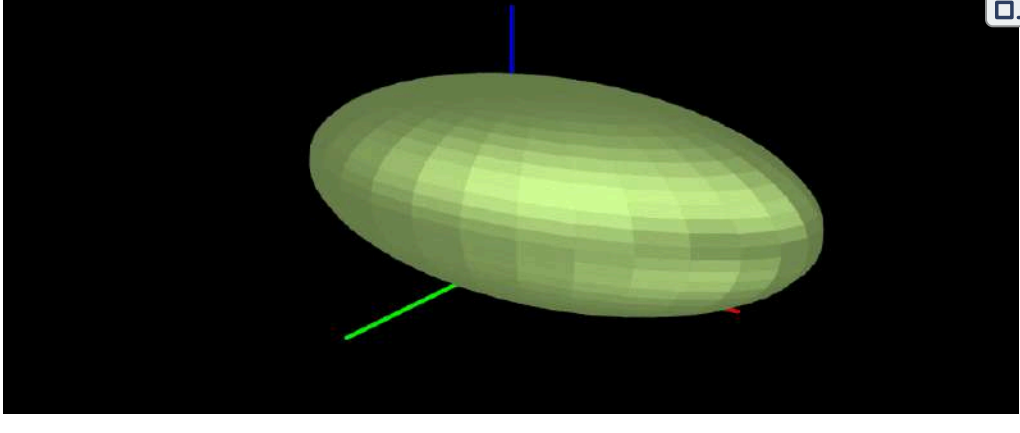
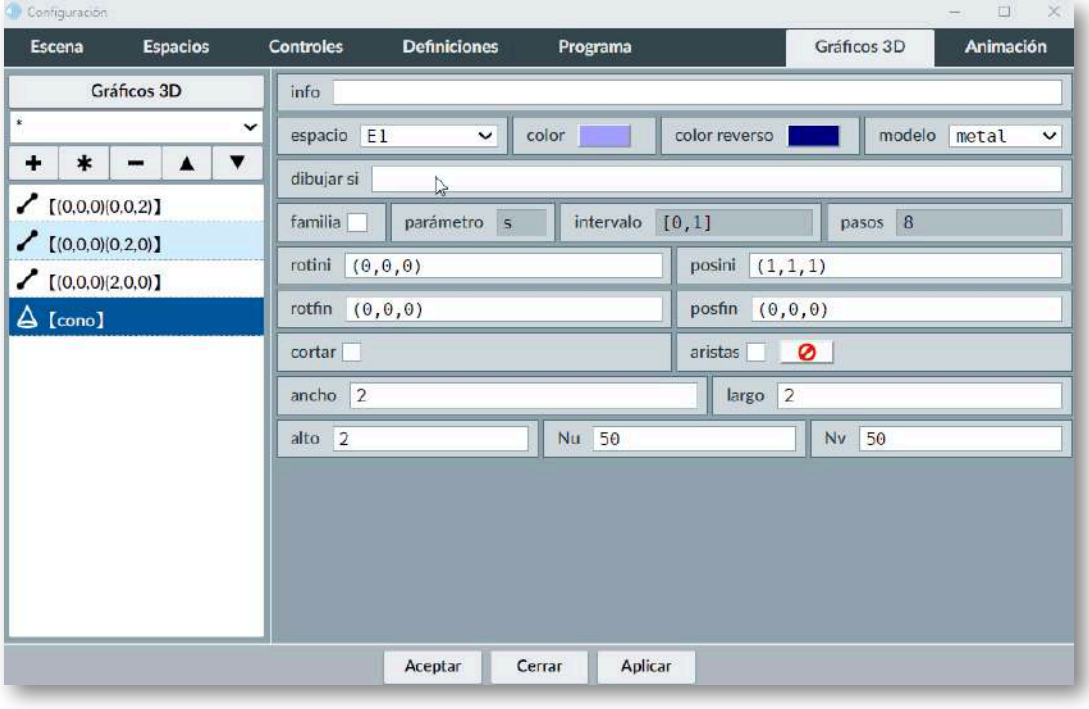


Gráfico 3D Cono

Cuerpo de revolución tridimensional formado por el giro de un triángulo rectángulo alrededor de uno de sus catetos. Consta de una base circular, una superficie lateral curva que termina en un punto llamado vértice y se caracteriza por su radio, altura y generatriz.



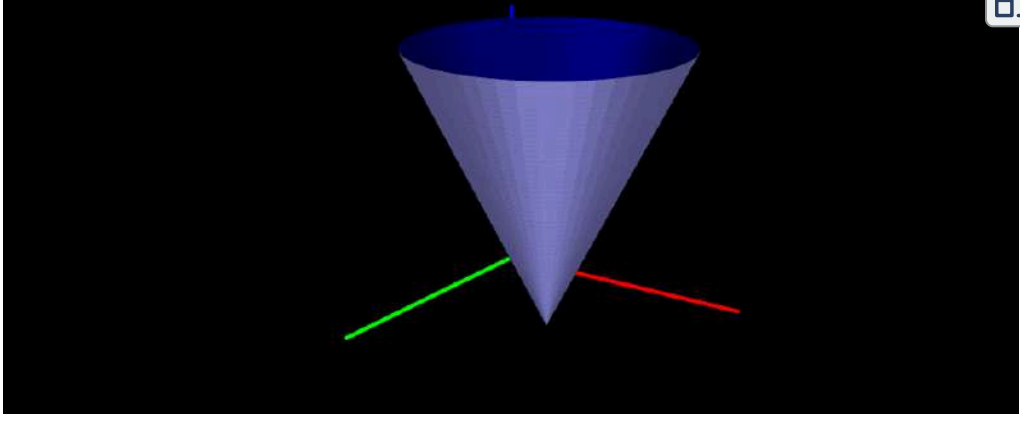
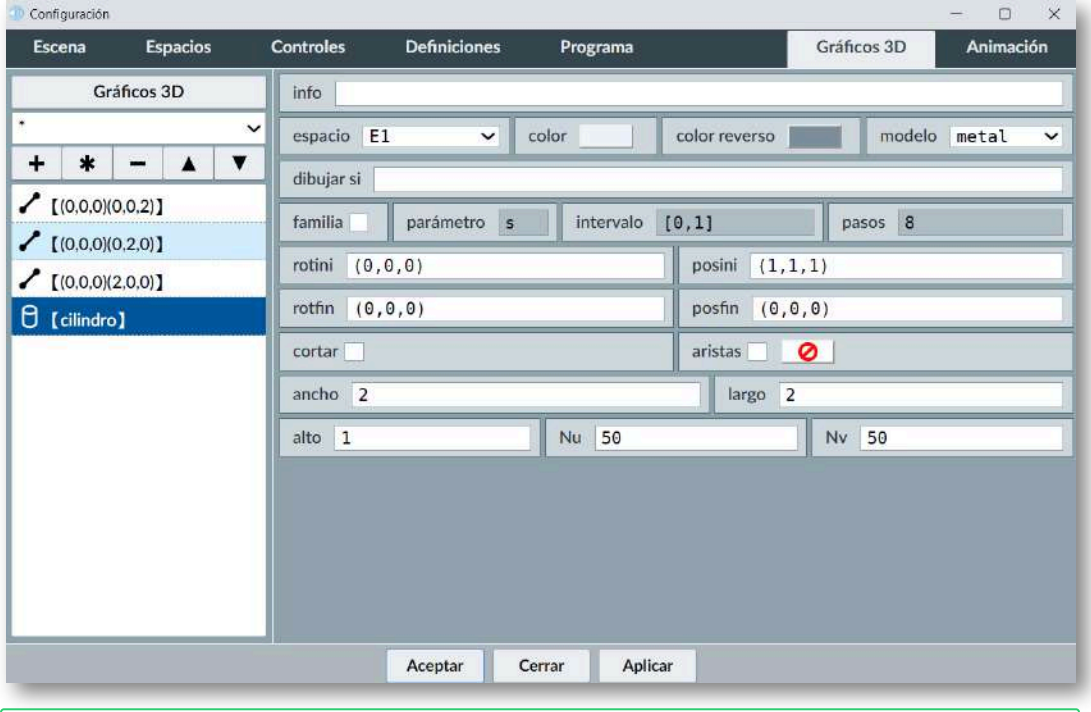


Gráfico 3D Cilindro

Cuerpo geométrico tridimensional de revolución generado por la rotación de un rectángulo alrededor de uno de sus lados. Se compone de dos bases circulares paralelas unidas por una superficie lateral curva. Sus elementos clave son la altura y el radio de la base.



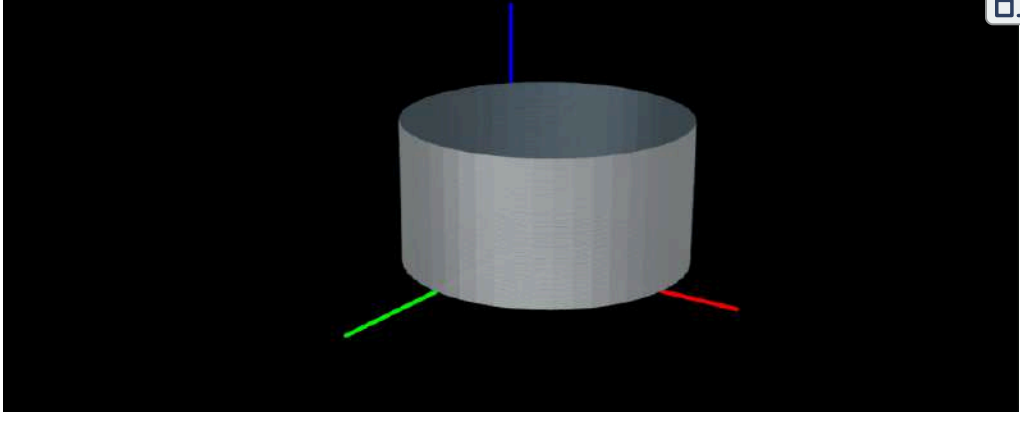
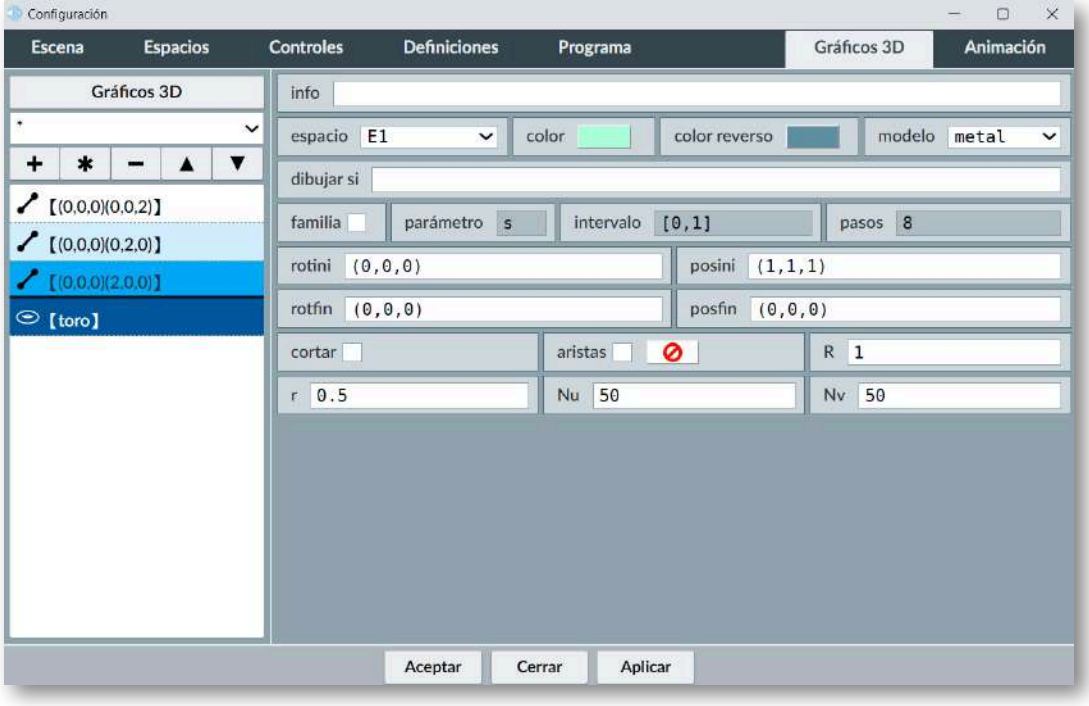
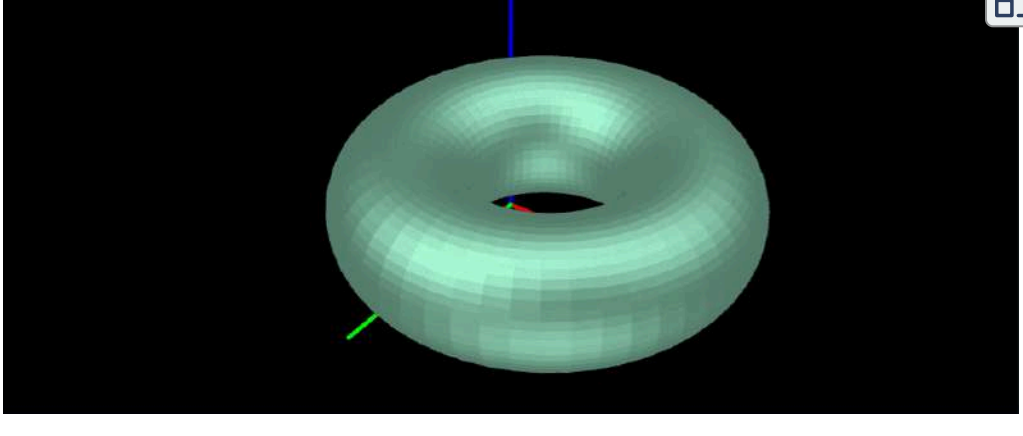


Gráfico 3D Toro

Un toro o toroide es una figura geométrica tridimensional con forma de anillo, dona o llanta, generada al hacer girar un círculo (o curva cerrada para el toroide) alrededor de un eje externo que no lo atraviesa.



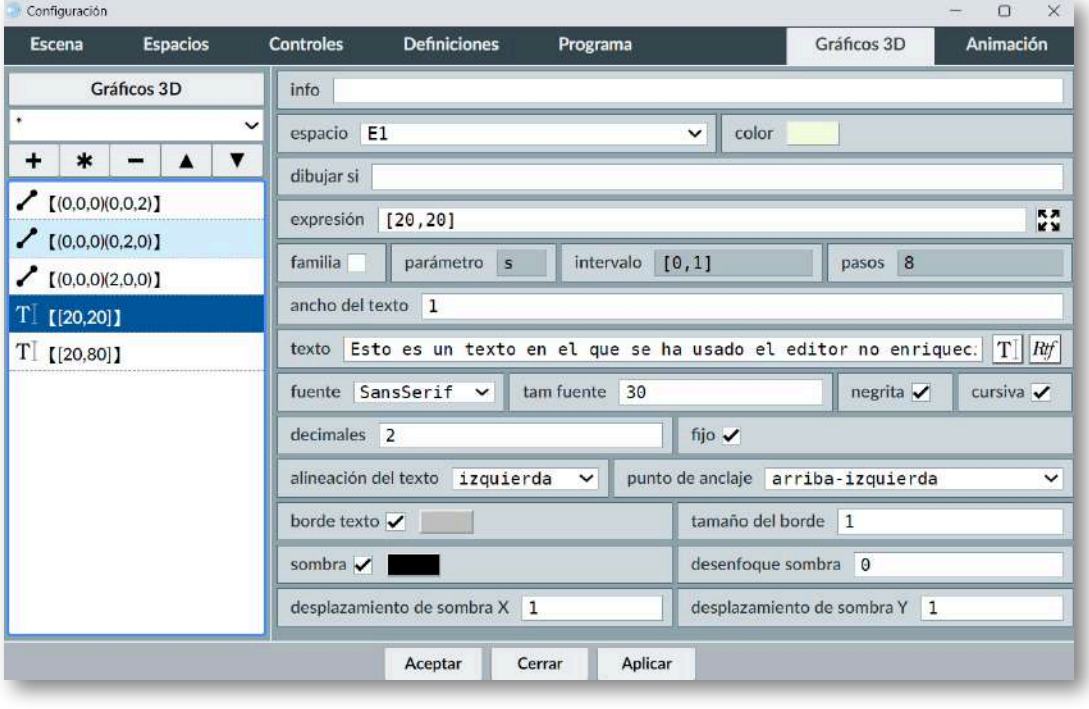


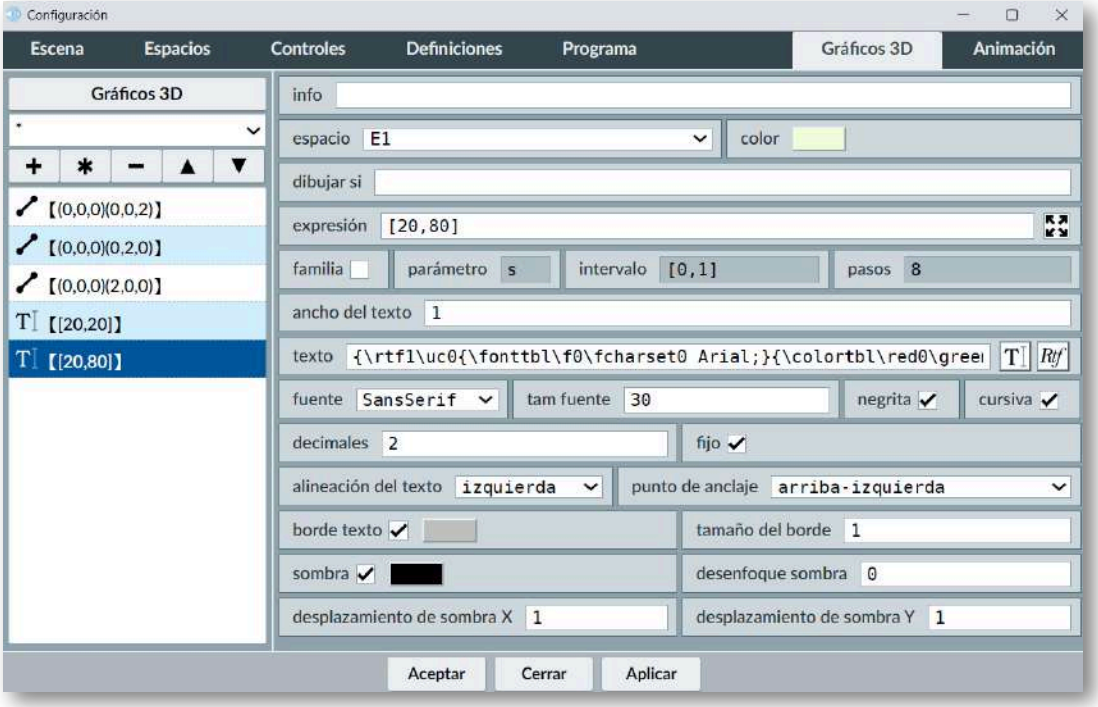
6.13 Gráfico 3D Texto

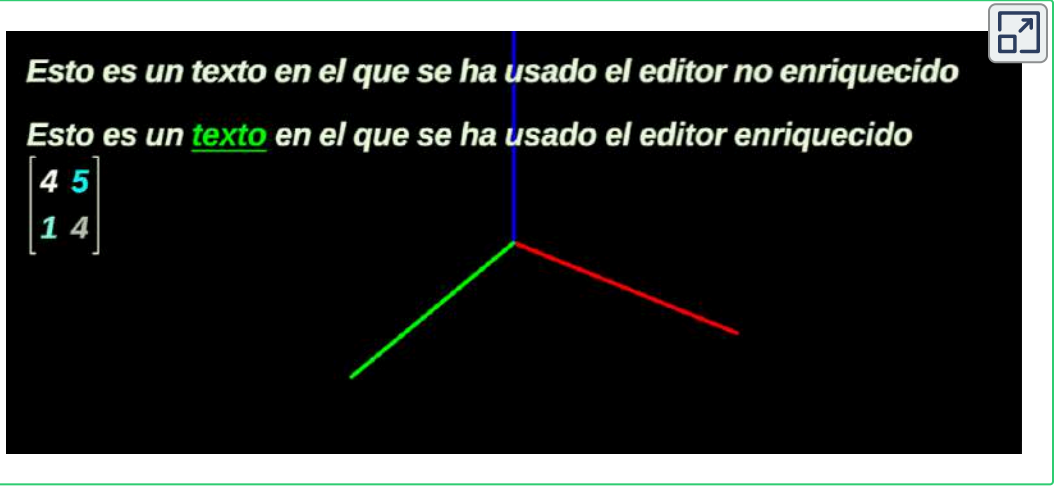
En los espacios 3D es posible introducir textos. Sus características son, en esencia, las mismas que las descritas para textos en espacios 2D. En realidad, los textos en 3D se comportan como elementos planos insertados dentro del espacio tridimensional, de manera similar a los subtítulos de una película o de un vídeo.

El panel de configuración de un texto en un espacio tridimensional es equivalente al del gráfico de texto en dos dimensiones, con algunas diferencias importantes: en 3D el texto no permite generar **rastros** y únicamente puede utilizarse en **coordenadas absolutas**. Por este motivo, las casillas de verificación correspondientes a rastro y coord. abs. no aparecen en la configuración de textos en espacios 3D.

A continuación, se muestran dos capturas del panel de configuración de texto en un espacio tridimensional.







Como hemos visto, también es posible utilizar elementos gráficos de tipo **punto** para introducir textos en espacios 3D. El texto queda asociado al punto y se posiciona respecto a sus coordenadas; si el tamaño del punto es cero, este no se dibuja, pero el texto sí se muestra correctamente en la escena.

Mostramos a continuación una escena más elaborada en la que los rótulos han sido creados con elementos gráficos 3D de tipo **texto**, lo que permite integrar información descriptiva directamente dentro del espacio tridimensional.



6.14 Gráfico 3D Macro

Los **macros en 3D** son objetos gráficos que contienen elementos de programación y/o gráficos, y permiten reutilizar conjuntos completos de elementos dentro de distintas escenas.

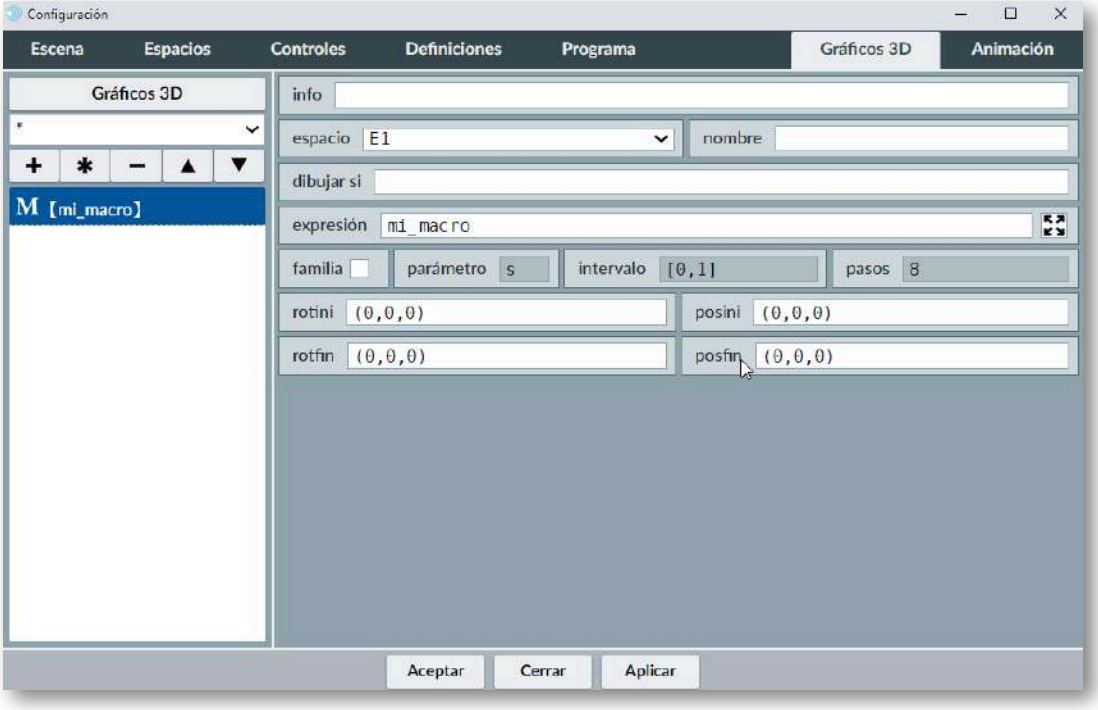
Su función principal es permitir que todos los elementos incluidos en el macro se incorporen en bloque y puedan utilizarse tantas veces como se desee dentro de una escena. Además, los macros ofrecen interesantes posibilidades cuando se animan o cuando se aplican controles sobre ellos.

Desde la escena que utiliza un macro es posible modificar las propiedades o valores de los elementos que lo componen. Por ejemplo, si se crea un macro que contiene un segmento, desde una escena se podrá llamar a ese macro e insertar tantos segmentos como se quiera, pudiendo además modificar parámetros como la longitud, el color o el grosor del segmento.

Su funcionamiento es similar al del gráfico **macro** en espacios bidimensionales. La principal diferencia es que, en los macros 3D, la ruta al archivo del macro se introduce en un campo de texto denominado **expresión**, en lugar del campo **archivo**. Este campo de texto debe contener la ruta relativa y el nombre del archivo que contiene los datos del macro.

Es importante tener en cuenta que, para utilizar macros en tres dimensiones, la escena (o archivo HTML) que los hereda debe contener previamente un **espacio 3D con el mismo nombre** que el utilizado en la escena original en la que se crearon los elementos del macro.

A continuación, presentamos el panel de configuración de un elemento gráfico 3D de tipo **macro**.



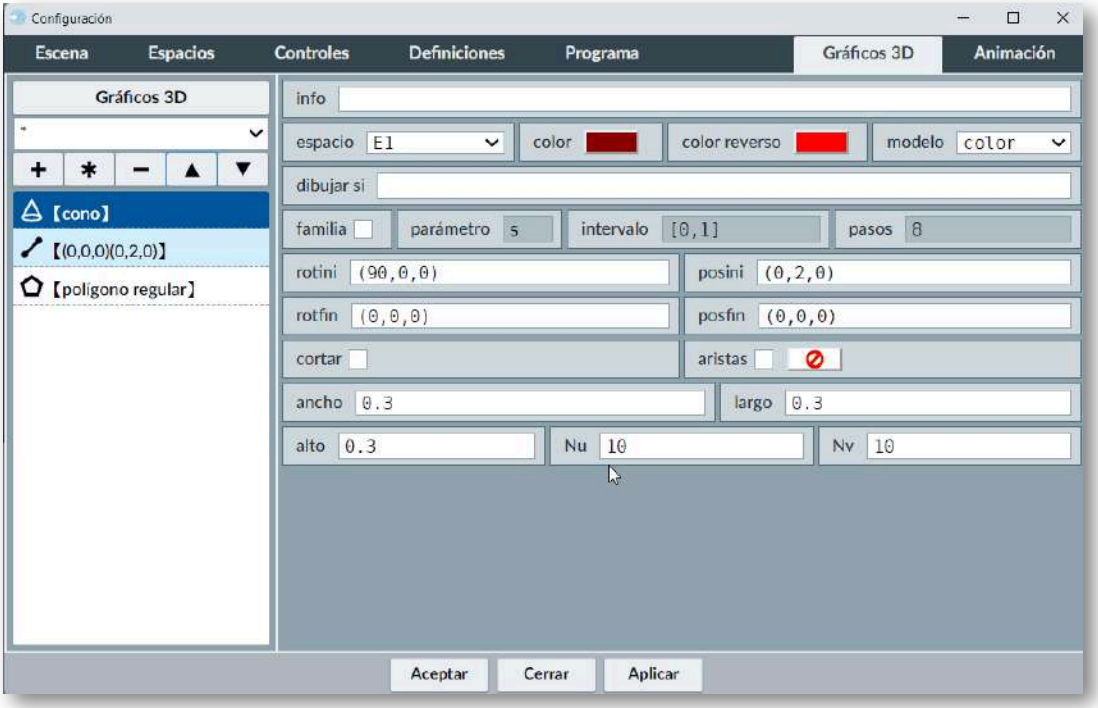
La forma de definir nuevos macros 3D es análoga a la explicada para los macros en 2D y se resume en los siguientes pasos:

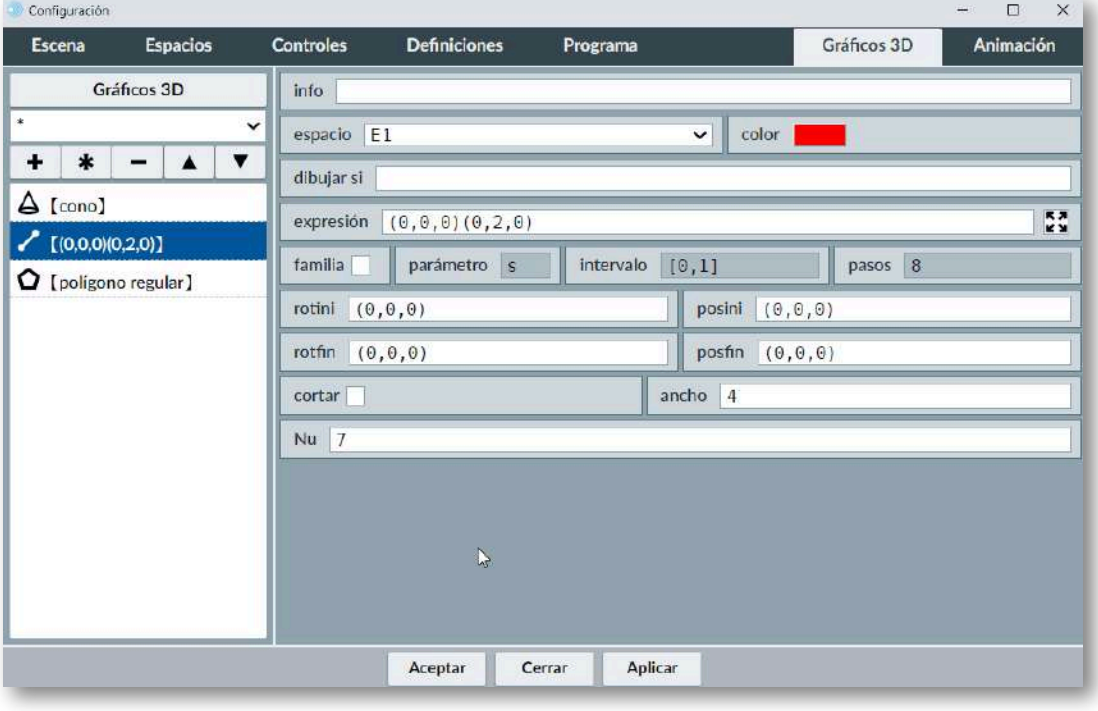
a) Se crea una escena con un único espacio 3D que puede contener elementos gráficos 3D y elementos de programación. Por ejemplo, una flecha roja en un espacio tridimensional.

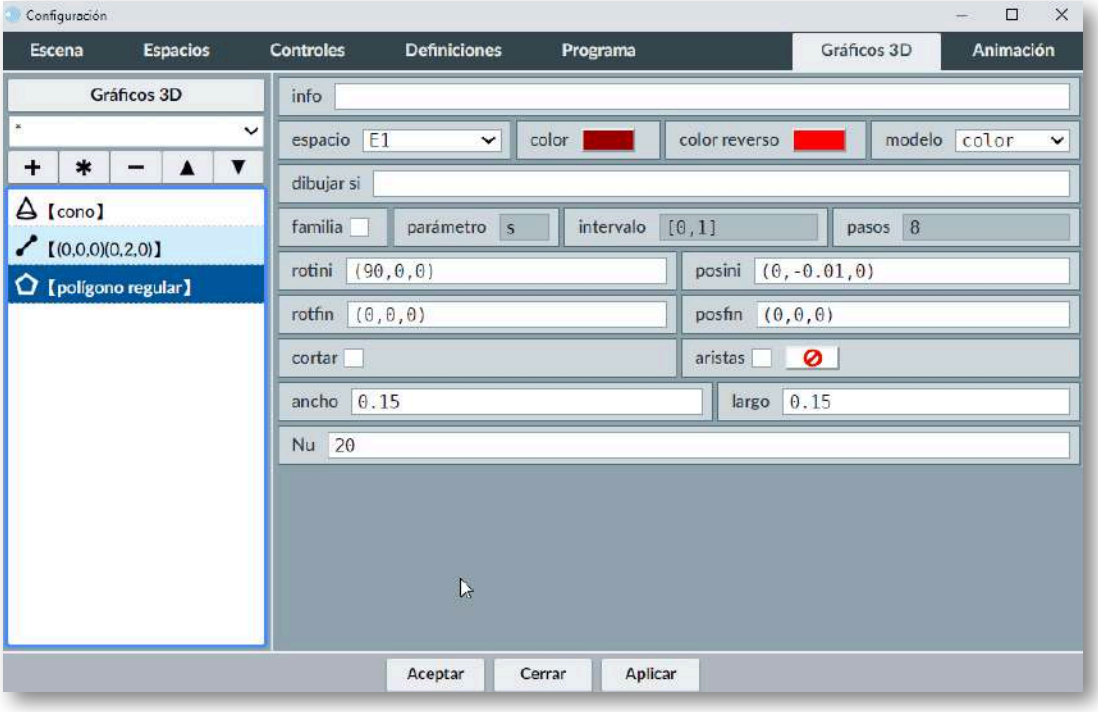
Para ello creamos un espacio 3D y, dentro de este, incorporamos tres elementos gráficos 3D:

- Un cono que actuará como la punta de la flecha.
- Un segmento que funcionará como el eje o cuerpo de la flecha.
- Un polígono regular con un número elevado de lados que simulará la parte trasera de la flecha.

Mostramos una captura de pantalla de los distintos elementos introducidos y de la escena final resultante.

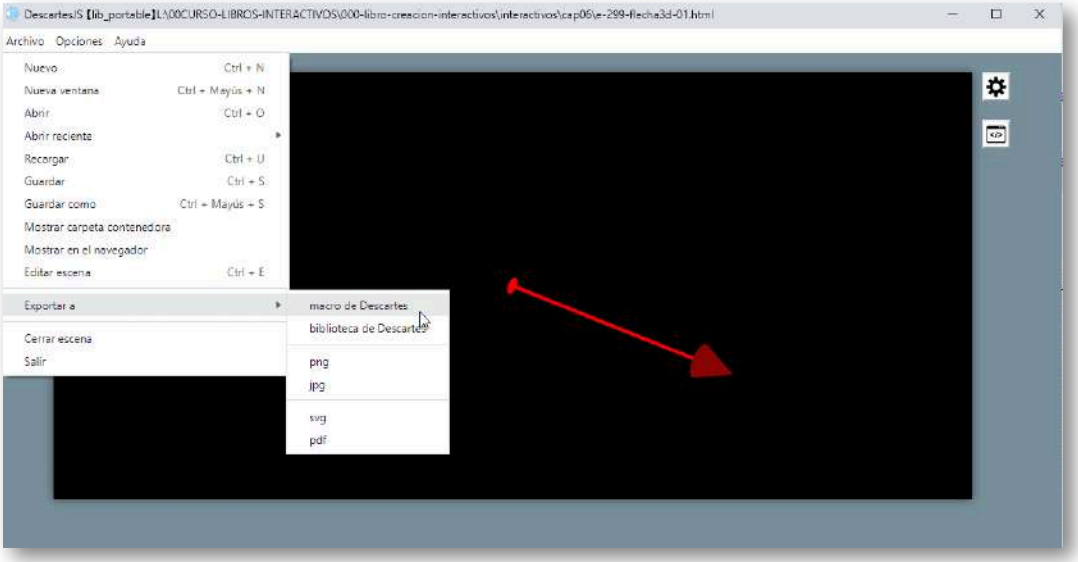

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Gráficos 3D
【cono】
【(0,0,0)(0,2,0)】
【polígono regular】
info
espacio E1
color
color reverso
modelo color
dibujar si
familia
parámetro s
intervalo [0,1]
pasos 8
rotini (90,0,0)
posini (0,2,0)
rotfin (0,0,0)
posfin (0,0,0)
cortar
aristas
ancho 0.3
largo 0.3
alto 0.3
Nu 10
Nv 10
Aceptar
Cerrar
Aplicar

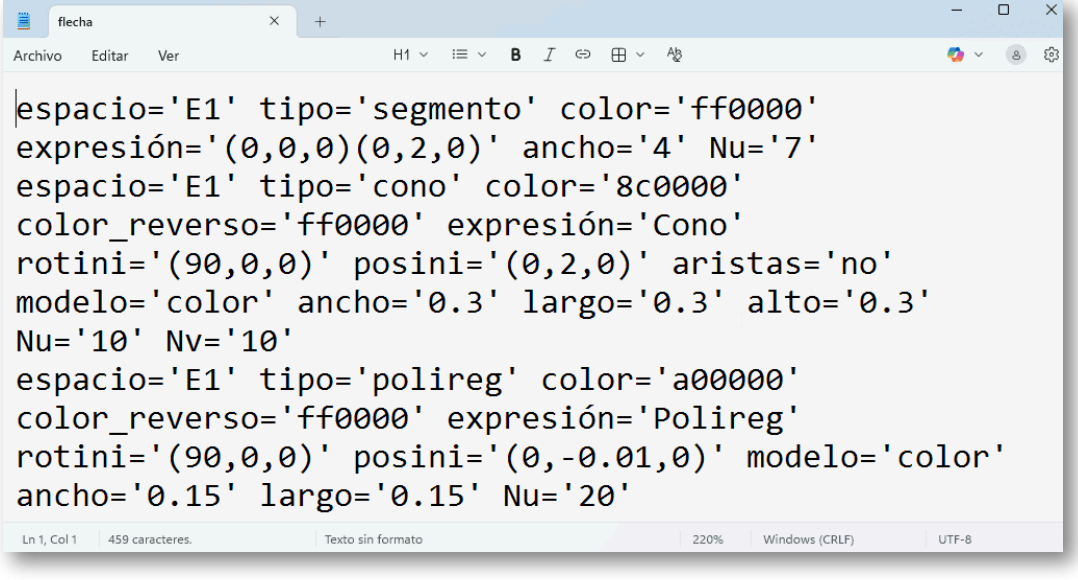

Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Gráficos 3D
【cono】
【(0,0,0)(0,2,0)】
【polígono regular】
info
espacio E1
color
dibujar si
expresión (0,0,0)(0,2,0)
familia
parámetro s
intervalo [0,1]
pasos 8
rotini (0,0,0)
posini (0,0,0)
rotfin (0,0,0)
posfin (0,0,0)
cortar
ancho 4
Nu 7
Aceptar
Cerrar
Aplicar





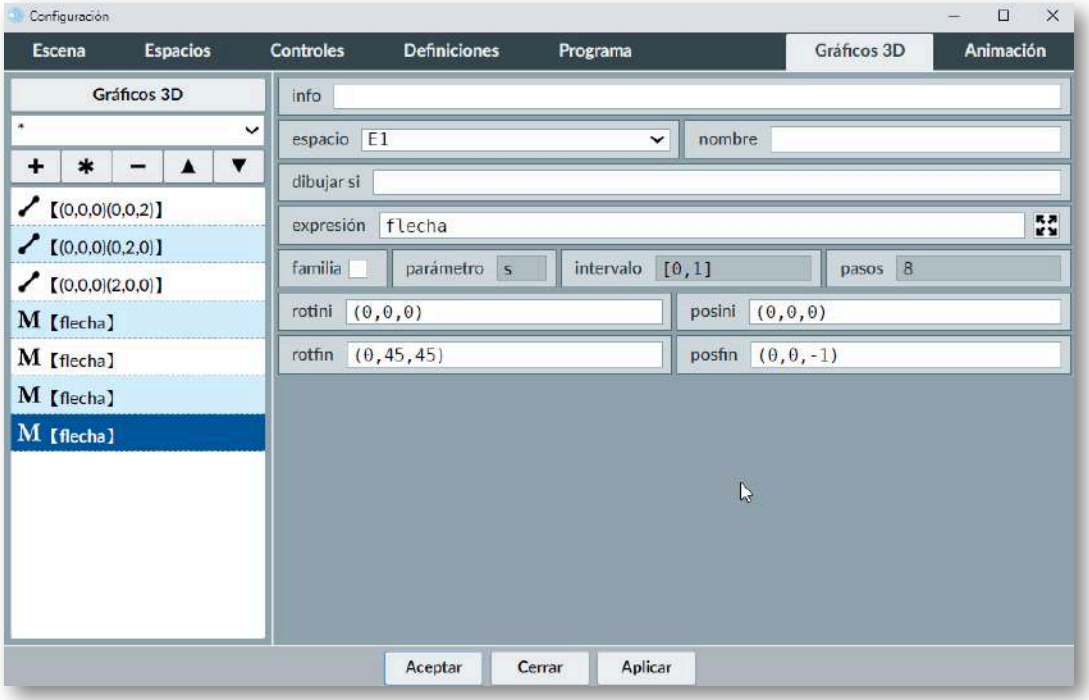
b) En el editor de Descartes, dentro del menú **Archivo**, se selecciona la opción **Exportar como macro de Descartes**. En nuestro ejemplo se guarda con el nombre **flecha**. A continuación, se muestra la forma de grabarlo y su código.

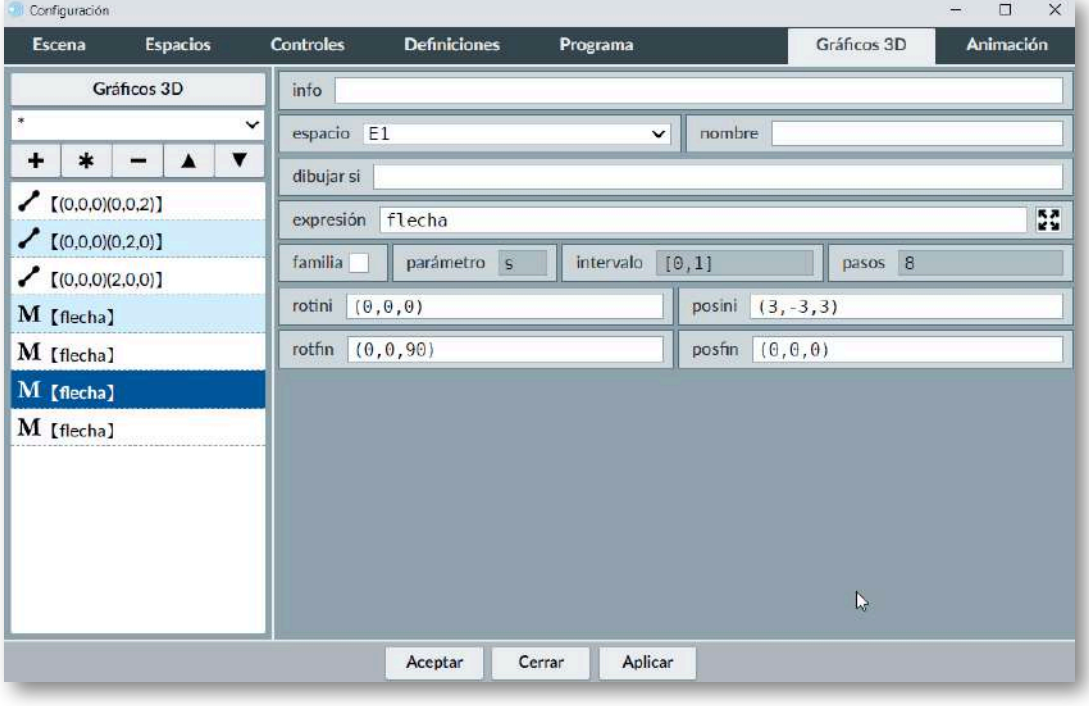




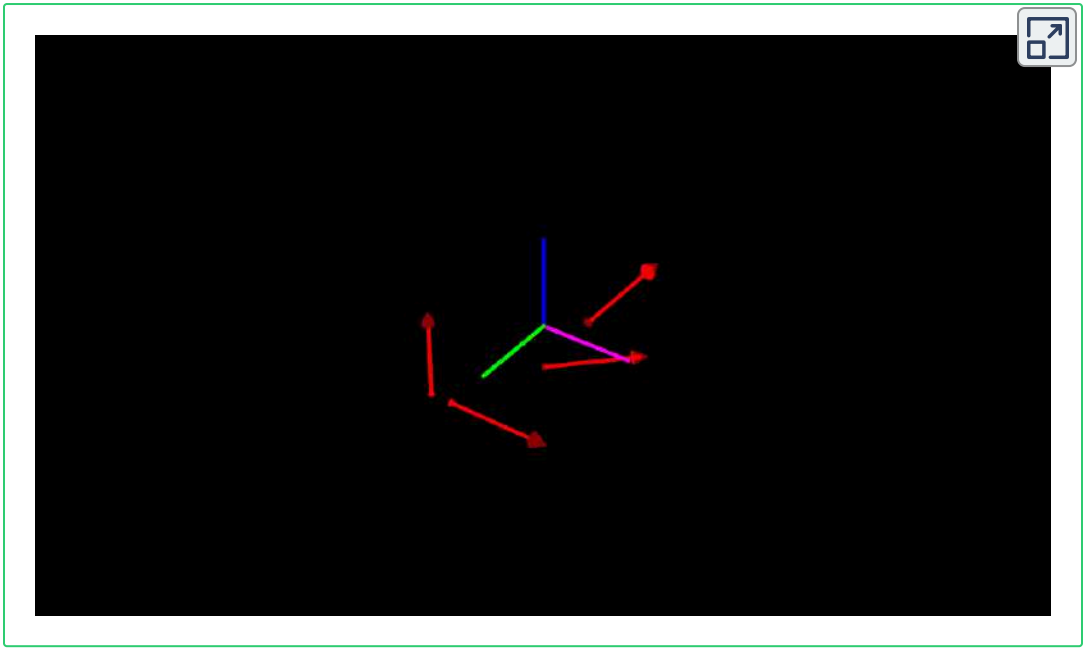
c) Desde una escena nueva que contenga un espacio 3D, el macro puede utilizarse tantas veces como se desee y es posible modificar sus parámetros.

En nuestro ejemplo, crearemos una escena con un espacio gráfico 3D en la que se incorporarán cuatro flechas utilizando el macro definido, situadas en distintas posiciones y con diferentes rotaciones.

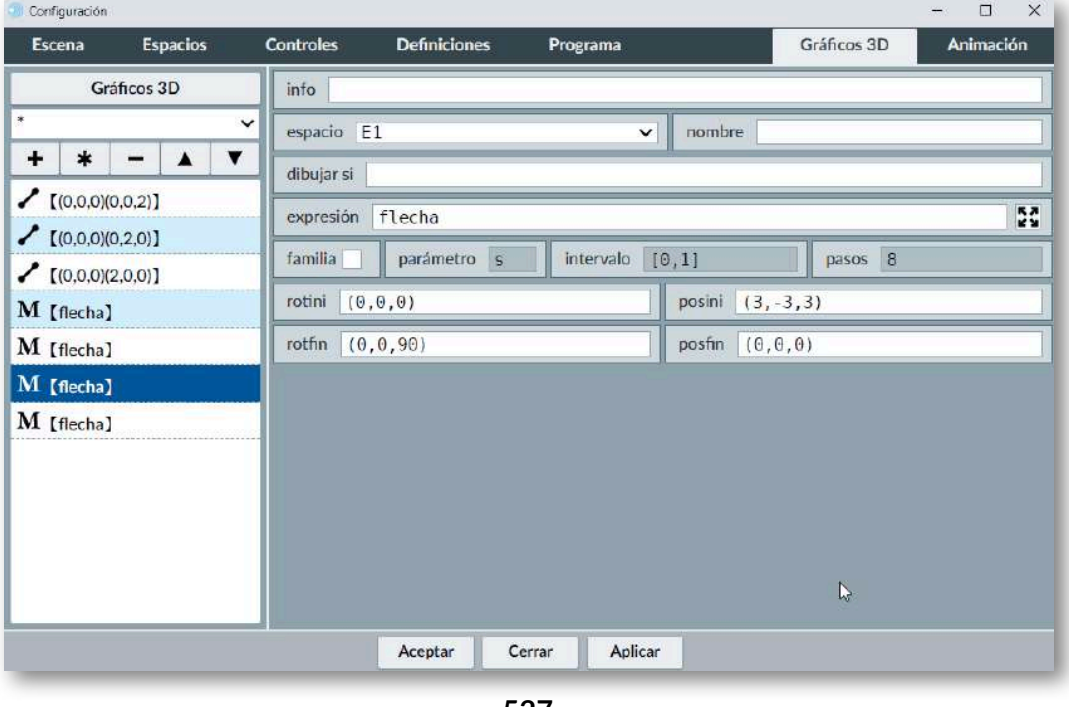


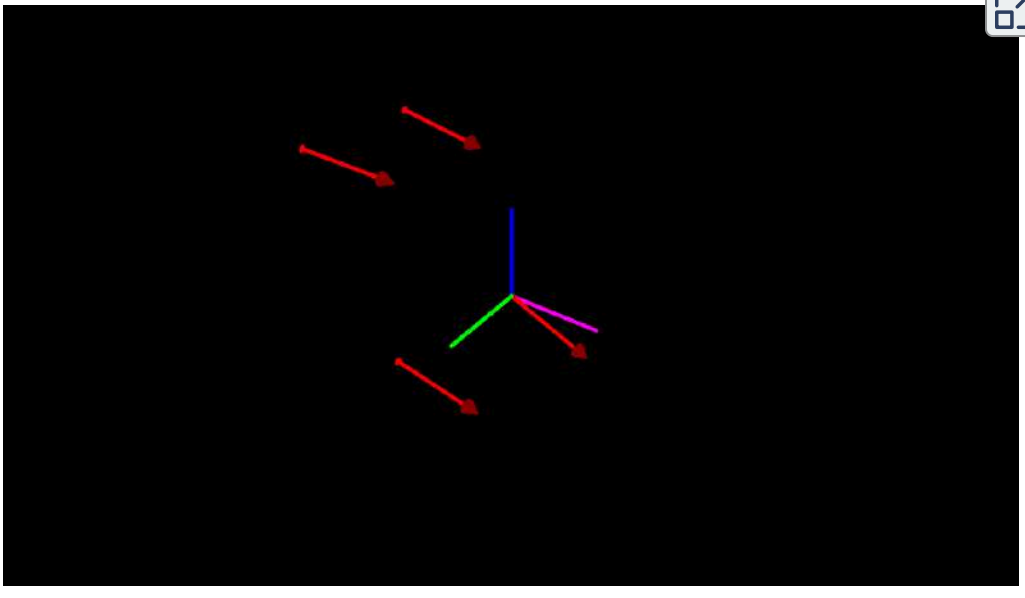


La escena resultante presenta el siguiente aspecto.



Ahora, mediante una animación, vamos a cambiar la posición y la rotación de las flechas.





Para modificar el valor de una variable de un macro desde la escena que lo llama, se procede de forma similar a los macros 2D:

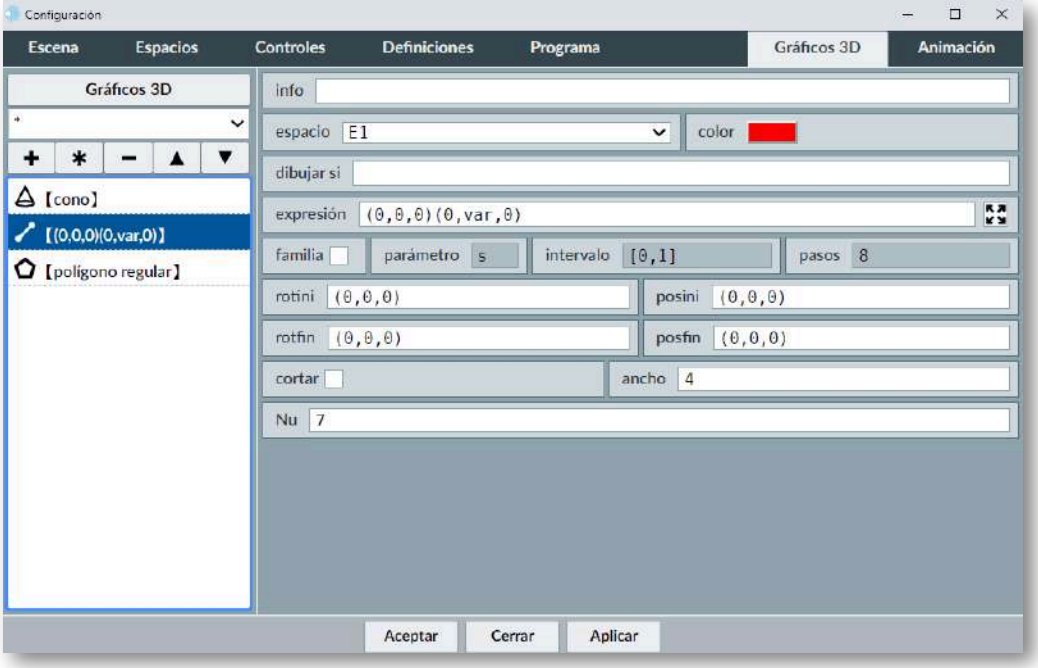
El macro debe tener definida una variable, por ejemplo, **var**. En este caso, **var** está asociada a la longitud de la flecha y afecta tanto a la longitud del segmento como a la posición del cono (punta de la flecha). Llamaremos a este macro **flecha2**

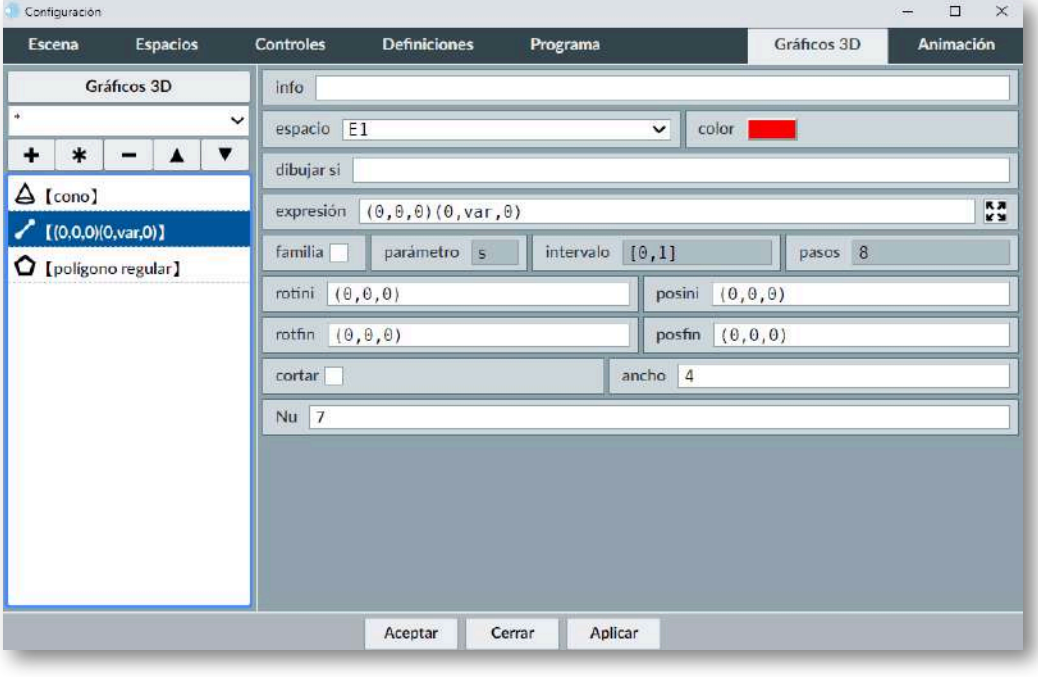
flecha2

Archivo Editar Ver H1

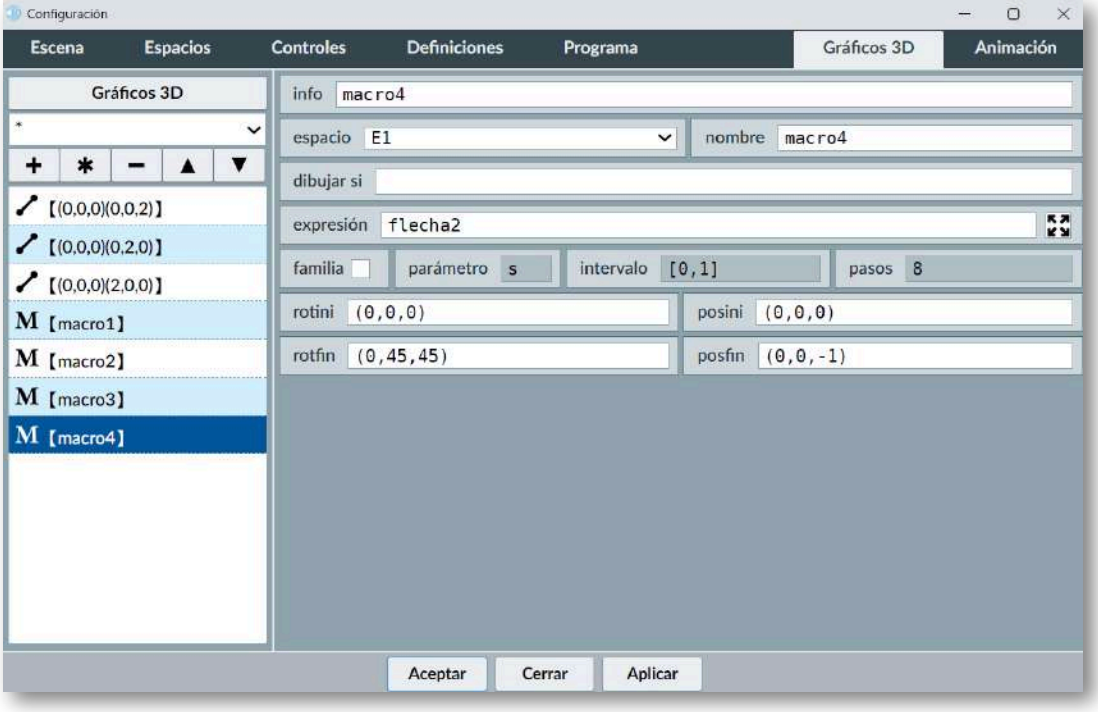
```
id='INICIO' algoritmo='sí' evaluar='una-sola-vez' inicio='E1.rot.z=325;
E1.rot.x=60; E1.rot.y=35; var=2'
id='CALCULOS' algoritmo='sí' evaluar='siempre'
espacio='E1' tipo='cono' color='8c0000' color_reverso='ff0000'
expresión='Cono' rotini='(90,0,0)' posini='(0,var,0)' modelo='color'
ancho='0.3' largo='0.3' alto='0.3' Nu='10' Nv='10'
espacio='E1' tipo='segmento' color='ff0000' expresión='(0,0,0)(0,var,0)'
ancho='4' Nu='7'
espacio='E1' tipo='polireg' color='a00000' color_reverso='ff0000'
expresión='Polireg' rotini='(90,0,0)' posini='(0,-0.01,0)'
modelo='color' ancho='0.15' largo='0.15' Nu='20'
```

En las siguientes capturas se muestra cómo se ha definido la variable **var** dentro del macro.





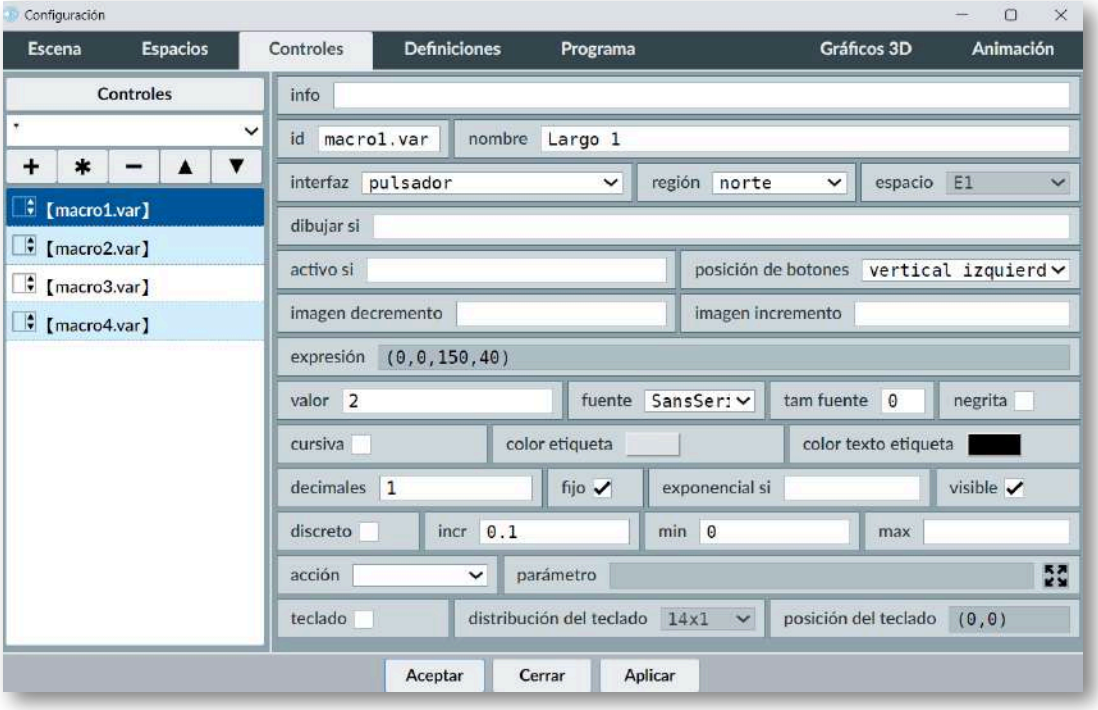
Desde la escena en la que se va a introducir el macro **flecha2** se modifica la variable del macro usando la notación **nombreDelMacro.nombreDeLaVariable**. Por ejemplo, **macro1.var** para cambiar el valor de var en el macro al que se le ha dado el nombre de macro 1, **macro2.var** para el denominado macro 2 y análogamente se procede para los macros 3 y 4.

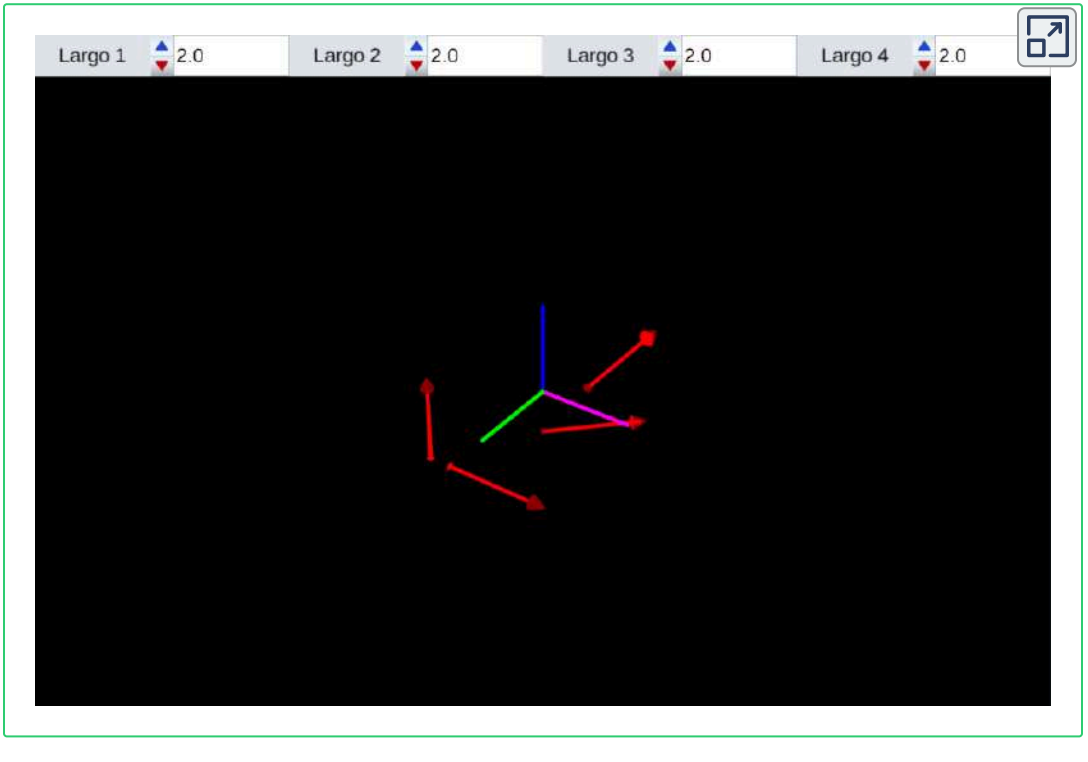


De esta forma, el valor de la variable **var** de **macro1** será igual al de la variable correspondiente en la escena que lo llama y lo mismo se aplica a los demás macros.

Vamos a crear cuatro controles con **id** macro1.var, macro2.var, macro3.var y macro4.var, de manera que cada control permita modificar la longitud de la flecha correspondiente definida por su macro respectivo.

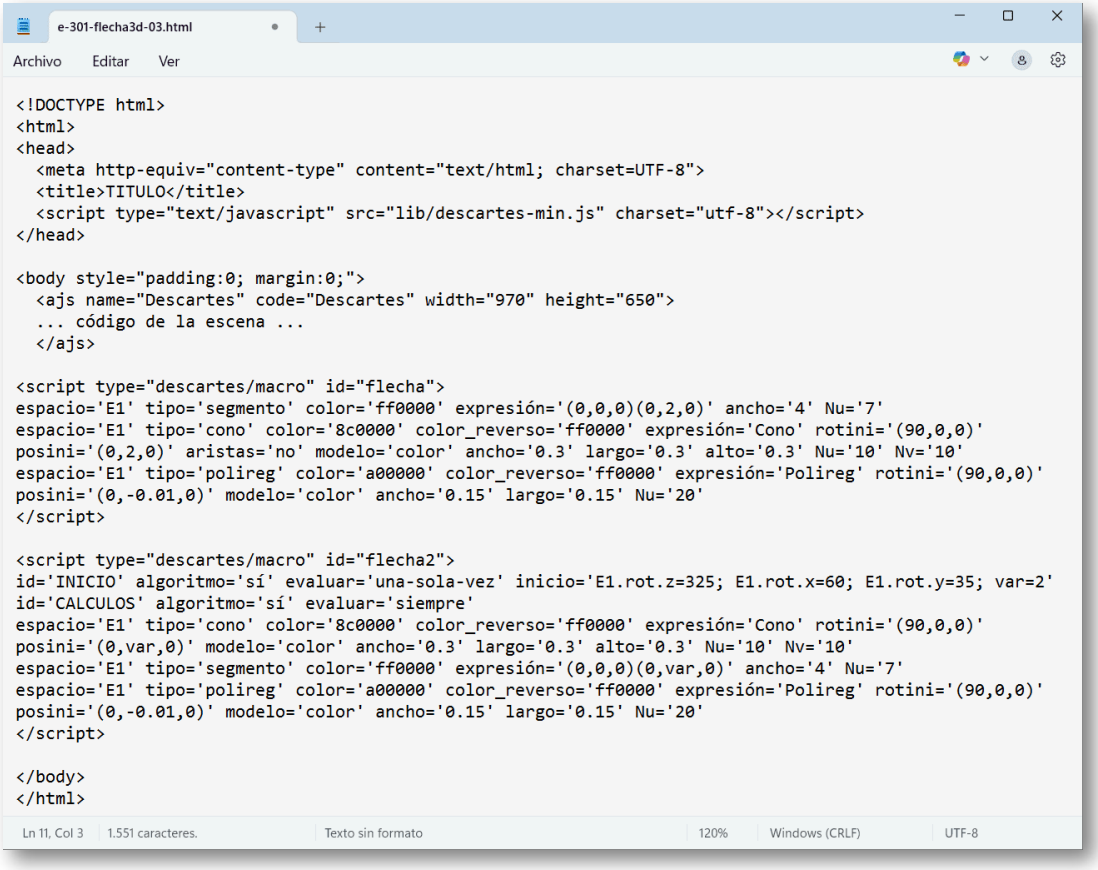
El resultado será una escena en la que se puede modificar la longitud de las diferentes flechas introducidas mediante macros.


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos 3D
Animación
Controles
[macro1.var]
[macro2.var]
[macro3.var]
[macro4.var]
info
id macro1.var
nombre Largo 1
interfaz pulsador
región norte
espacio E1
dibujar si
activo si
posición de botones vertical izquierd
imagen decremento
imagen incremento
expresión (0,0,150,40)
valor 2
fuente SansSer:
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 1
fijo
exponencial si
visible
discreto
incr 0.1
min 0
max
acción
parámetro
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar


Largo 1 2.0
Largo 2 2.0
Largo 3 2.0
Largo 4 2.0

Tal y como se explicó para los macros en gráficos 2D, es conveniente añadir el código de un macro 3D al final del código HTML de la escena, colocándolo entre las etiquetas correspondientes (`<script type="descartes/macro" id="ruta relativa completa del fichero del macro respecto de la escena">` y `</script>`) para que el navegador lo reconozca e interprete correctamente cuando la escena se rueda en local.

De este modo, el macro queda integrado en la escena. Si en el editor de **Descartes** está activada la opción **"macro"** dentro del submenú *Agregar al HTML* del menú *Opciones*, este código se incorporará automáticamente al final del archivo de la escena (de forma análoga ocurre con matrices, bibliotecas y vectores).



6.15 Familias en gráficos 3D

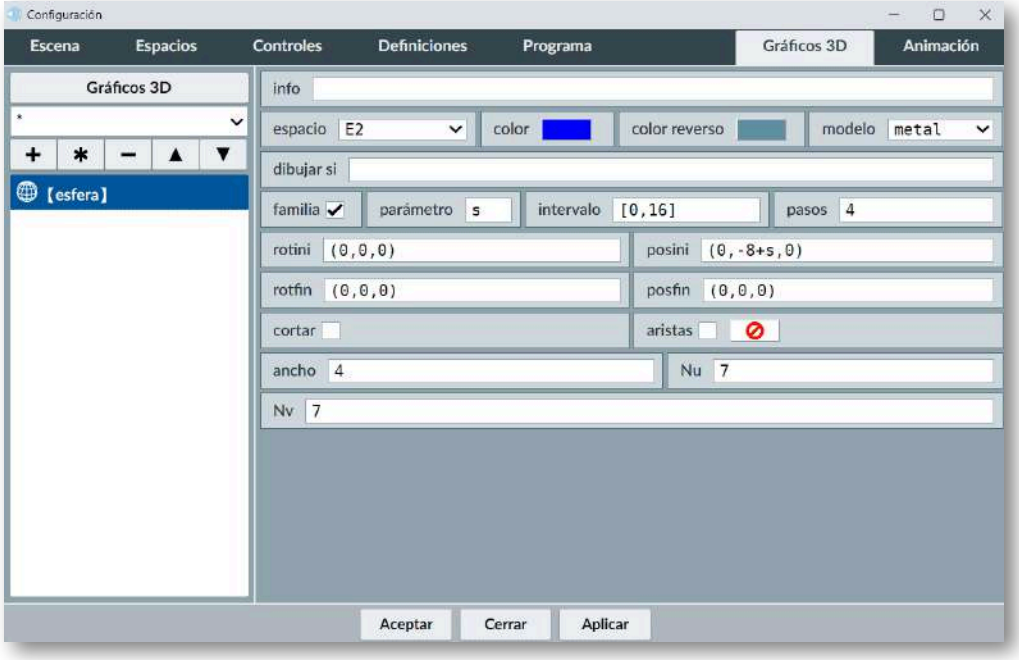
En los gráficos 3D, al igual que en los 2D, los elementos poseen la opción de activar el campo **familia**. Esta opción permite que un mismo elemento se repita automáticamente en la escena un número determinado de veces, ubicándose en posiciones definidas de manera sistemática según un parámetro. Es una herramienta especialmente útil para generar patrones o secuencias de elementos sin necesidad de multiplicarlos manualmente.

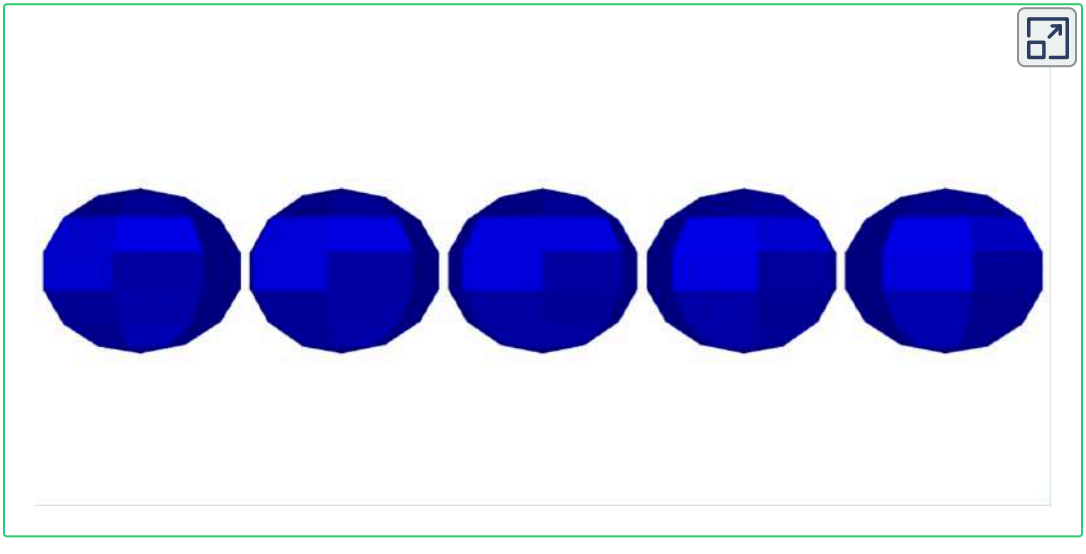
Los parámetros que definen una **familia** de gráficos 3D son los siguientes:

- **Intervalo:** define los límites inferior y superior dentro de los cuales se asignarán los valores de repetición del elemento en cuestión.
- **Pasos:** indica cuántos incrementos se realizarán para recorrer uniformemente el intervalo. El número total de elementos generados será igual a Pasos + 1.
- **Parámetro:** variable que tomará los valores correspondientes a cada paso del intervalo. Se recomienda un nombre corto, como **s**, para facilitar su uso en las expresiones de posición u otras propiedades del elemento.

Para ilustrarlo, vamos a desarrollar un ejemplo sencillo:

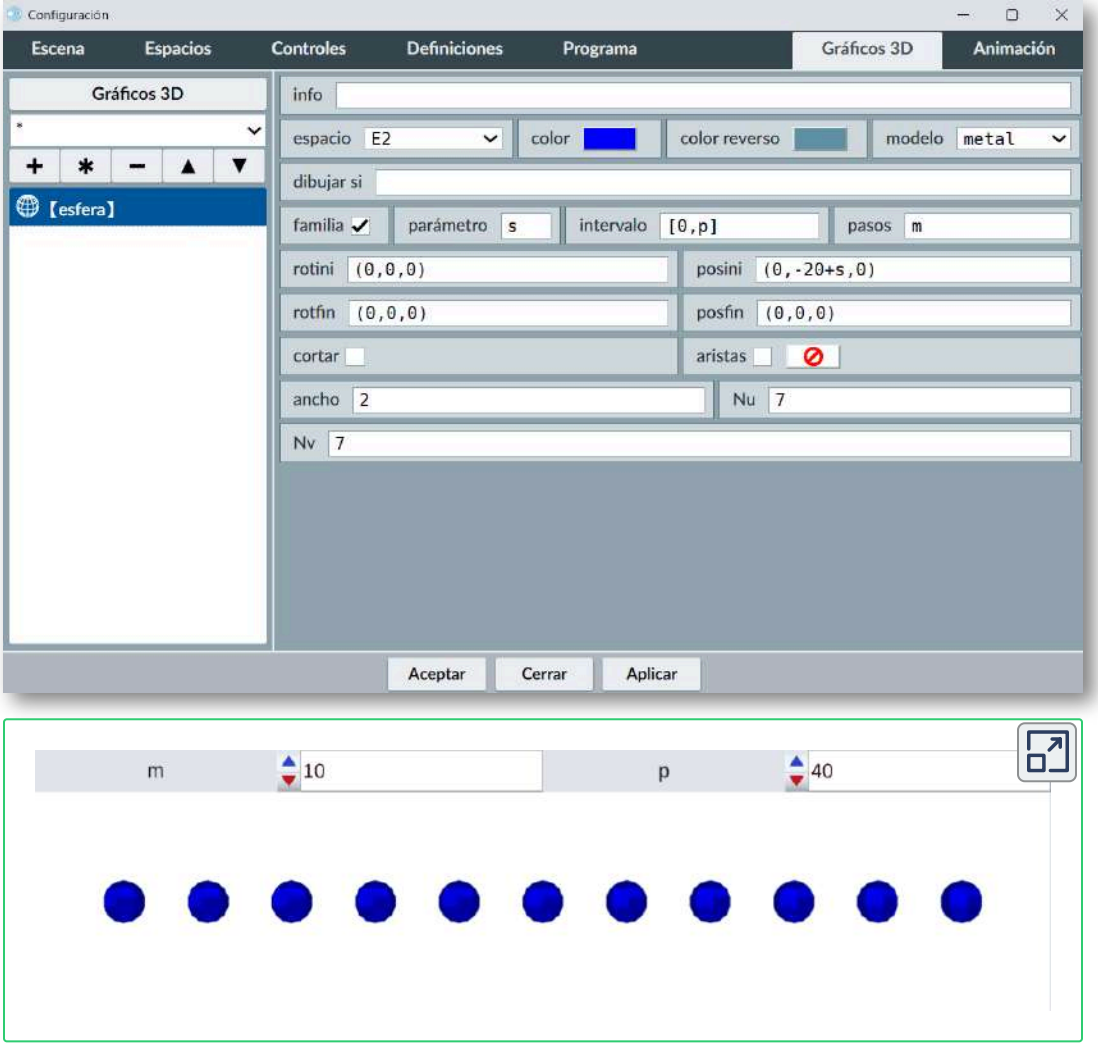
- Queremos dibujar una **esfera** cinco veces en un espacio 3D.
- Definimos el **intervalo** de 0 a 16.
- Establecemos **4 pasos**, de manera que el parámetro **s** tomará los valores 0, 4, 8, 12 y 16.
- La **posición inicial** de cada esfera se define como **posini(0, -8+s, 0)**, lo que permite que la ubicación horizontal dependa del valor del parámetro **s**.





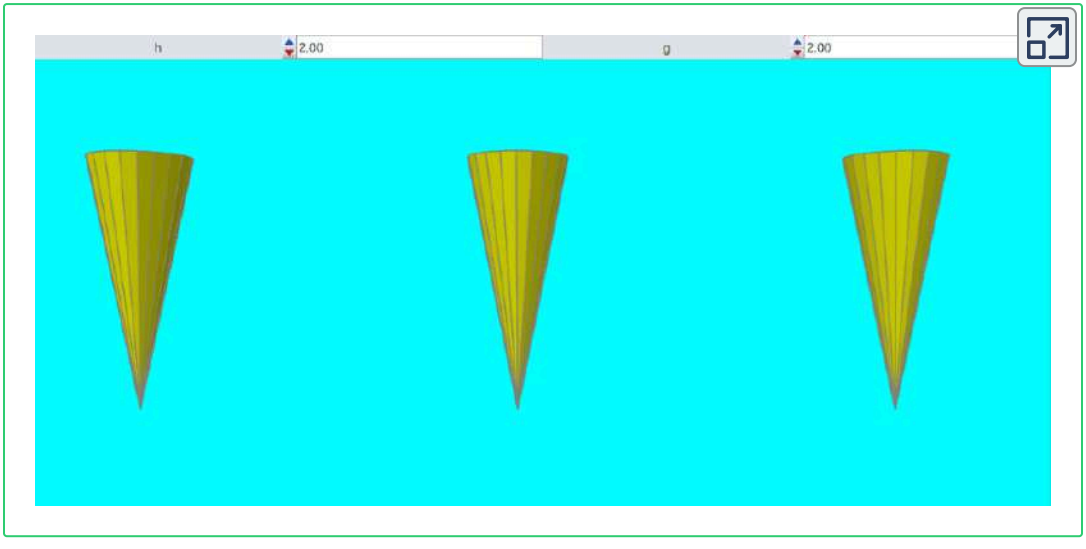
De este modo, se obtiene un total de cinco esferas, ubicadas en posiciones horizontales: -8, -4, 0, 4 y 8, correspondientes a los cinco valores del parámetro **s**. Cada esfera se dibuja automáticamente sin necesidad de replicarla manualmente, logrando un patrón ordenado y controlado mediante la familia.

El **número de pasos** y el **intervalo** pueden parametrizarse mediante una variable, lo que permite ajustar dinámicamente la distribución de los elementos en la familia según distintos valores, como se muestra en la escena siguiente.



Este método se puede aplicar a cualquier elemento gráfico 3D, permitiendo controlar posiciones, dimensiones u otras propiedades en función de un parámetro, facilitando la creación de estructuras repetitivas o secuencias complejas en la escena.

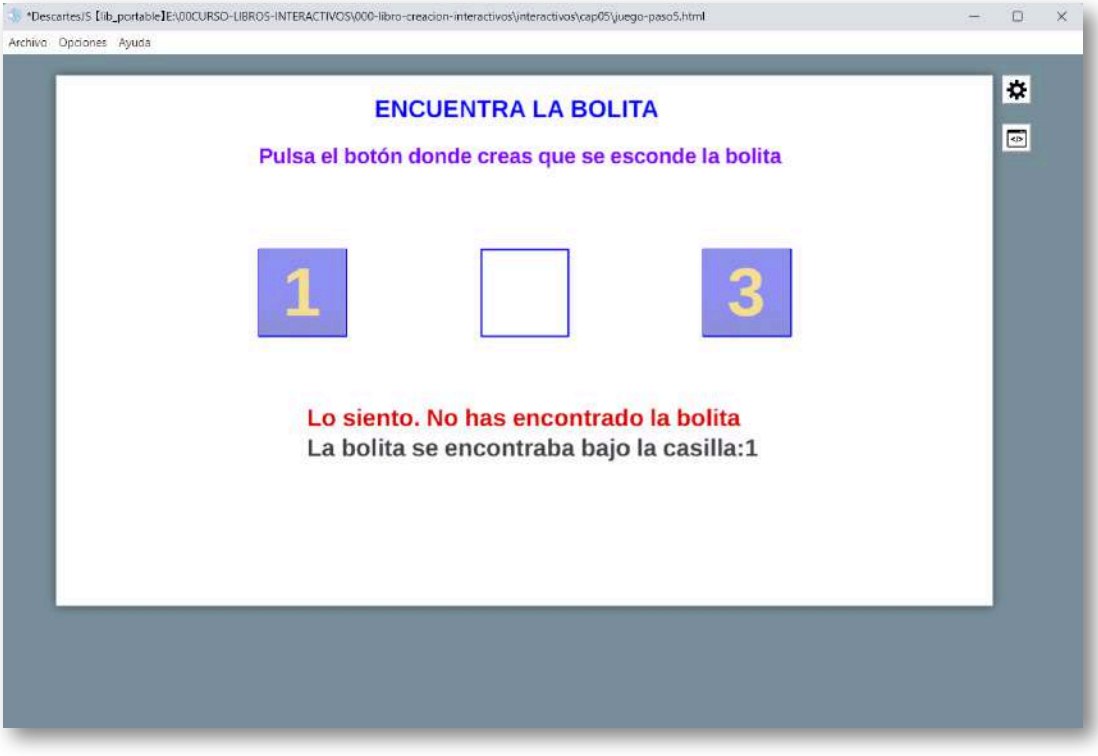
Presentamos un nuevo ejemplo de escena con familias de gráficos 3D, acompañado de un vídeo que ilustra cómo trabajar con este tipo de familias.





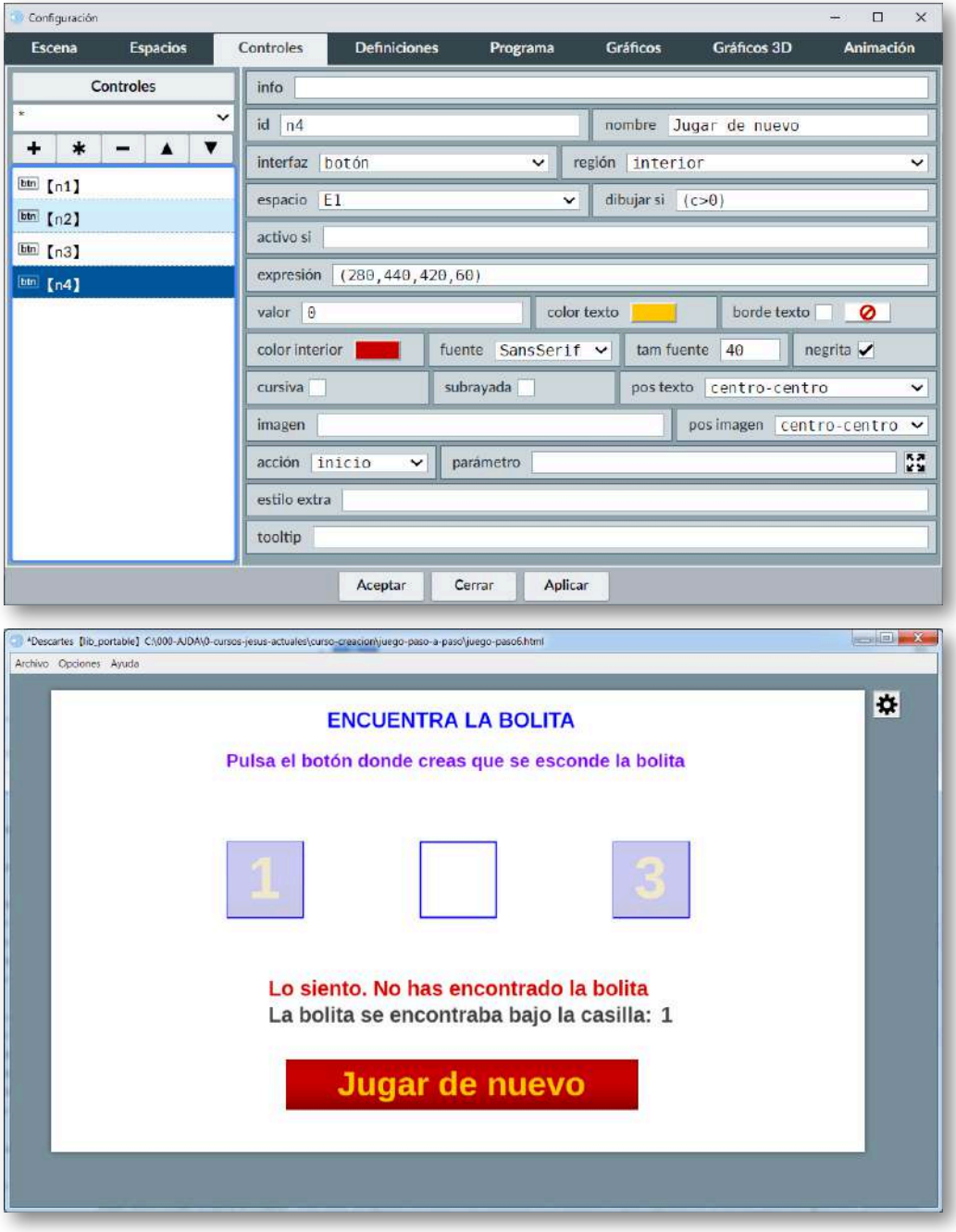
6.16 Mi primer juego paso a paso. Paso 6

Vamos a realizar el **paso 6** del juego, que corresponde al último paso. Para comenzar, abrimos el juego con el editor de **Descartes**. Para ello accedemos al menú *Archivo* > *Abrir* y seleccionamos el archivo llamado **juego-paso5**.



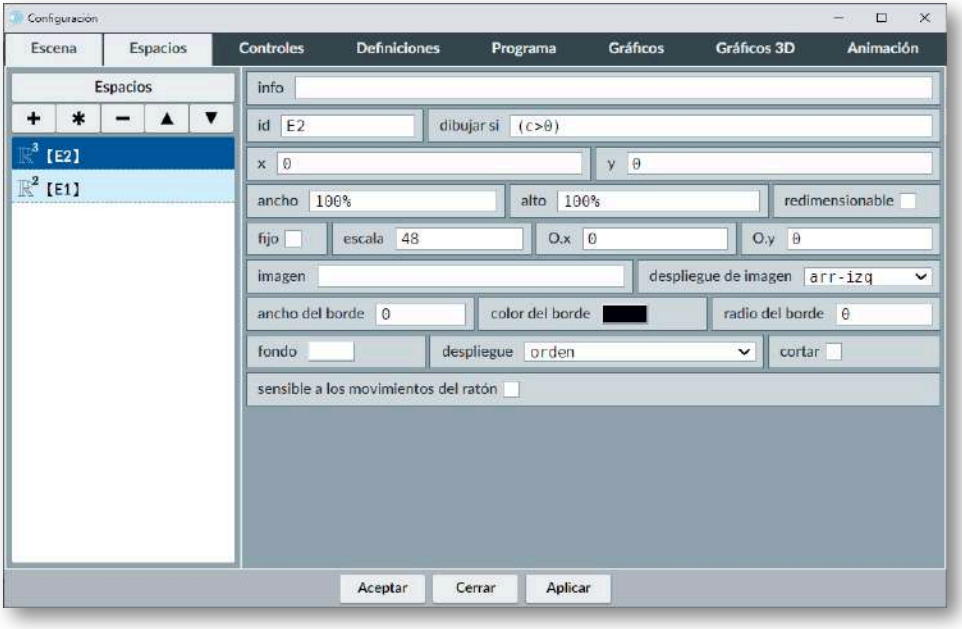
A continuación, añadiremos un control que permita reiniciar la escena. Este control será **interior** y mostrará el texto **"Jugar de nuevo"**. Además, solo aparecerá cuando la partida haya finalizado, por lo que en la condición **dibujar-si** se establecerá **(c>0)**.

El control será de tipo **botón**, con color interior **rojo** y color de texto **naranja**. El tamaño de la fuente será **40**, de tipo **SansSerif**; el texto aparecerá en **negrita** y centrado dentro del botón. Su identificador (**id**) será **n4**.

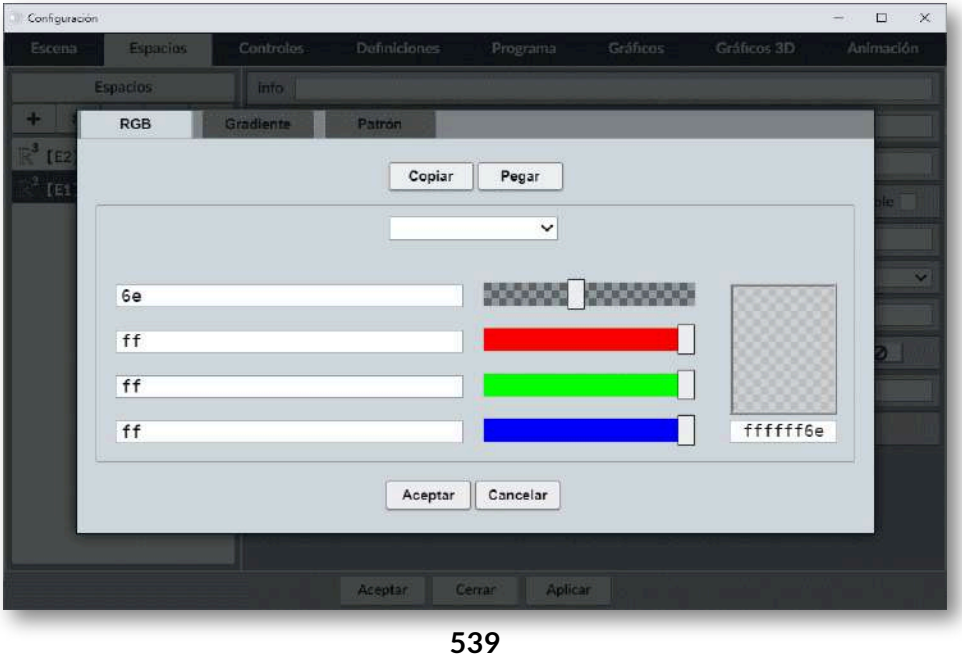


Seguidamente, accedemos al menú **Espacios** y creamos un espacio **3D** (botón **+ agregar espacio R3**) que ocupe toda la escena, con **fondo blanco** y condición **dibujar-si (c>0)**.

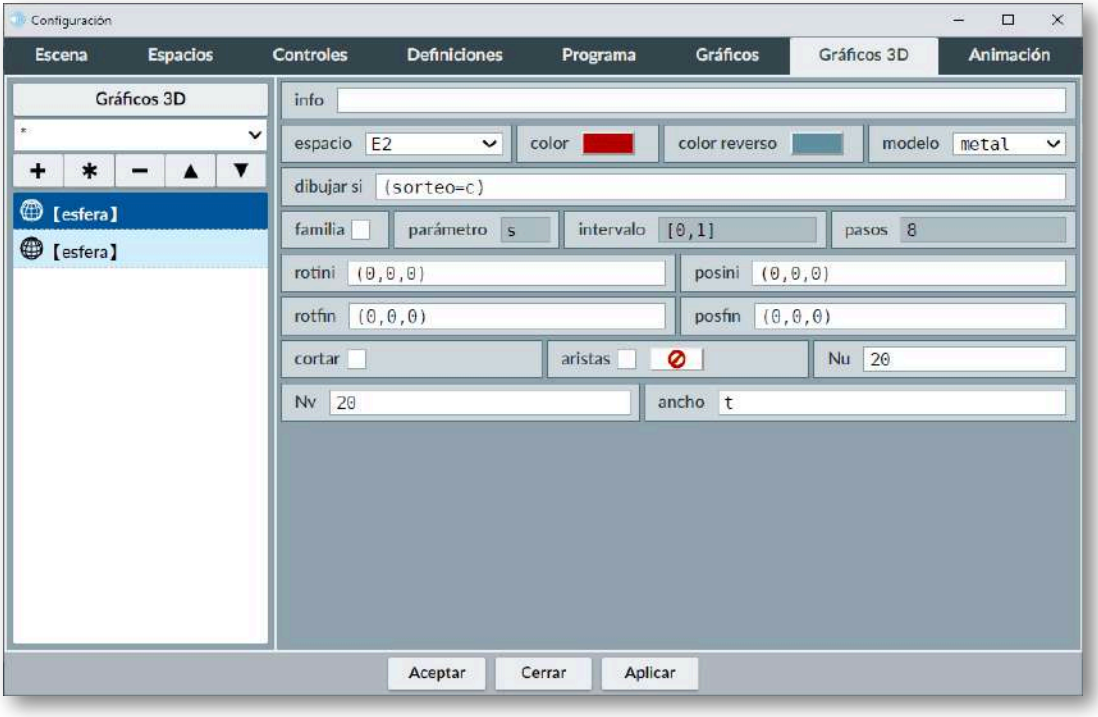
Lo colocamos arriba en la lista de espacios para que quede por debajo del espacio **2D** que ya teníamos creado.



Al espacio **2D**, que queda encima, le asignaremos un **fondo semitransparente** para poder visualizar el espacio **3D** situado detrás.

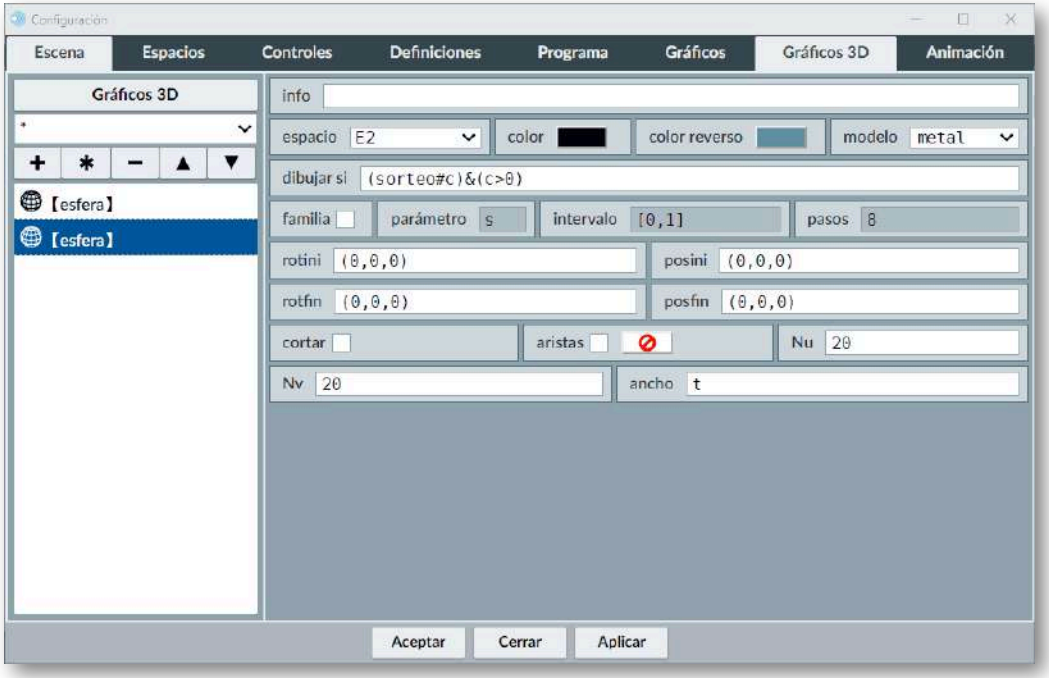


Vamos a incluir una esfera de color rojo desde el menú **gráficos 3D**, configurando **ancho = t**, **Nu = 20** y **Nv = 20**, y estableciendo que se dibuje cuando **(sorteo=c)**. El parámetro **t** controlará el tamaño de la esfera y aumentará mediante el menú de **animación**. De este modo, cuando se produzca un acierto, la esfera aparecerá y crecerá gradualmente.

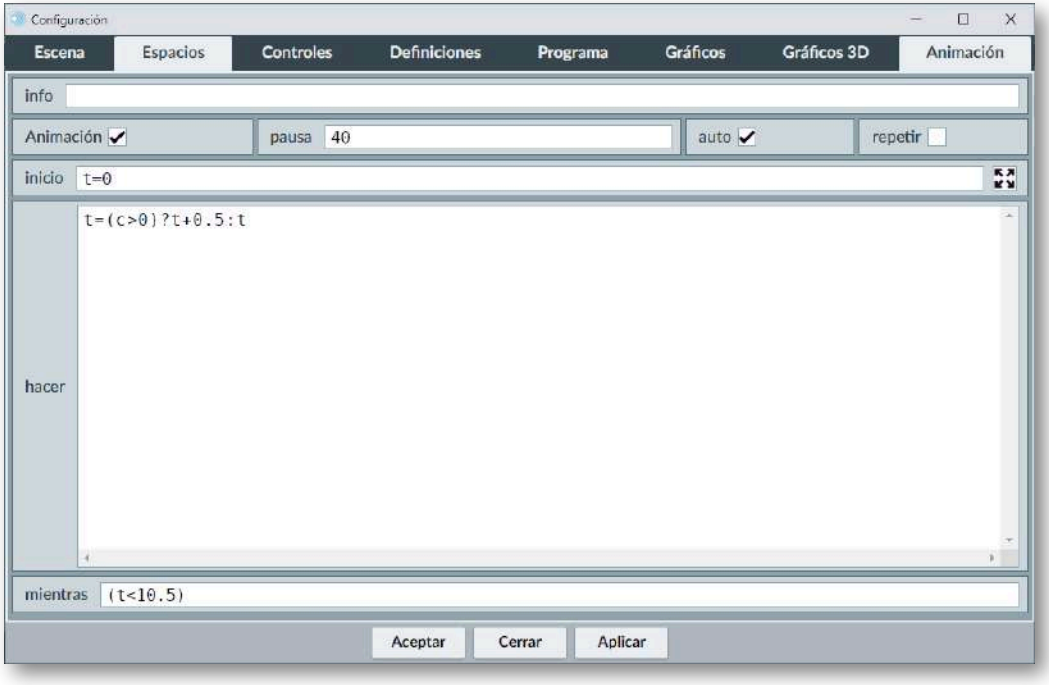


Análogamente, crearemos una esfera de color **negro** desde el menú **gráficos 3D**, configurando **ancho = t**, **Nu = 20** y **Nv = 20**, y estableciendo que se dibuje cuando **(sorteo#c)&(c>0)**. El parámetro **t** controlará el tamaño de la esfera y aumentará mediante el menú de **animación**. De este modo, cuando se produzca un fallo, la esfera aparecerá y crecerá gradualmente.

No se modificará ni la posición ni la rotación de la esfera; el modelo de dibujado se establecerá como **tipo metal** y el **color de reverso** será indiferente, dado que en este caso no será visible.



En el menú de **animación** configuraremos que la variable **t** aumente de **0.5** en **0.5** cuando **(c>0)**, manteniéndose activa mientras **(t<10.5)**.



El resultado se verá como en la primera imagen si hay acierto y como en la segunda imagen si no lo hay.





Finalmente, hemos completado nuestro primer juego totalmente funcional, en el que se han empleado de manera práctica los elementos fundamentales de **DescartesJS**, incluyendo gráficos 2D y 3D, animaciones, configuración, elementos de programación y controles interactivos.

A continuación, guardamos la escena final del juego con el nombre **juego-paso6** para conservar el resultado final obtenido.

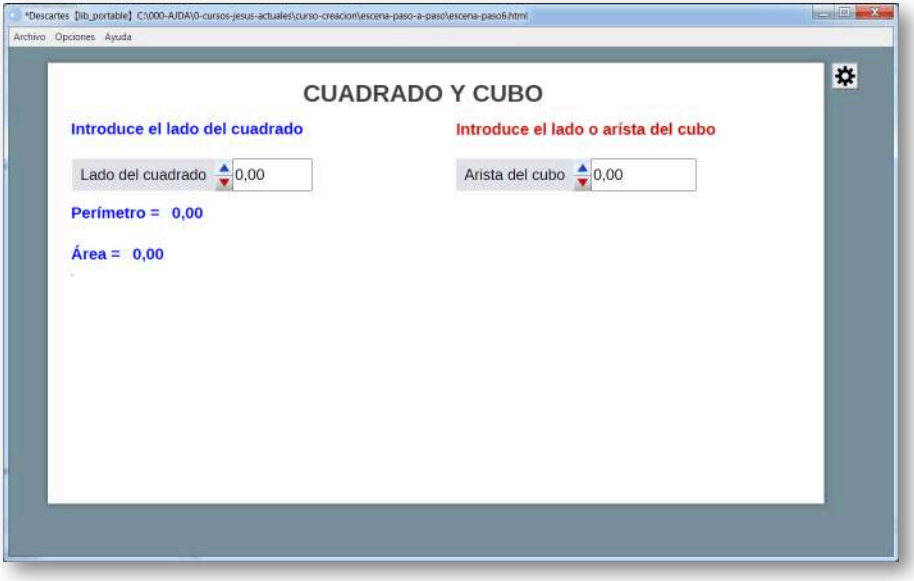


Este juego es completamente editable y puede modificarse o mejorarse según los deseos y necesidades del creador, permitiendo ajustar gráficos, animaciones, controles y mecánicas para adaptarlo a diferentes objetivos o aumentar su complejidad y funcionalidad.

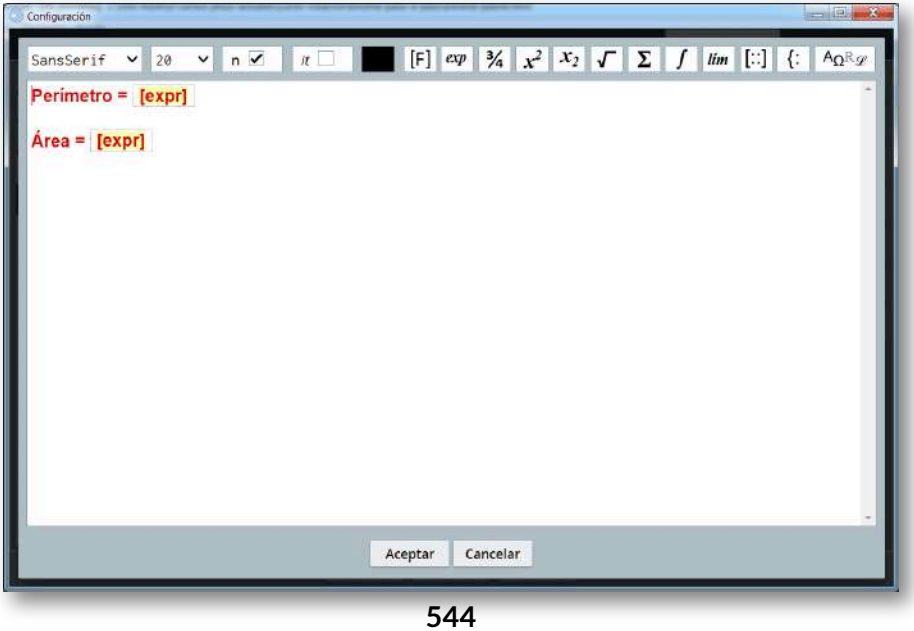
Con "Mi primer juego paso a paso" hemos completado un proyecto interactivo sencillo, pero totalmente funcional, diseñado para servir como guía, ejemplo práctico y base de aprendizaje para futuros desarrollos más complejos.

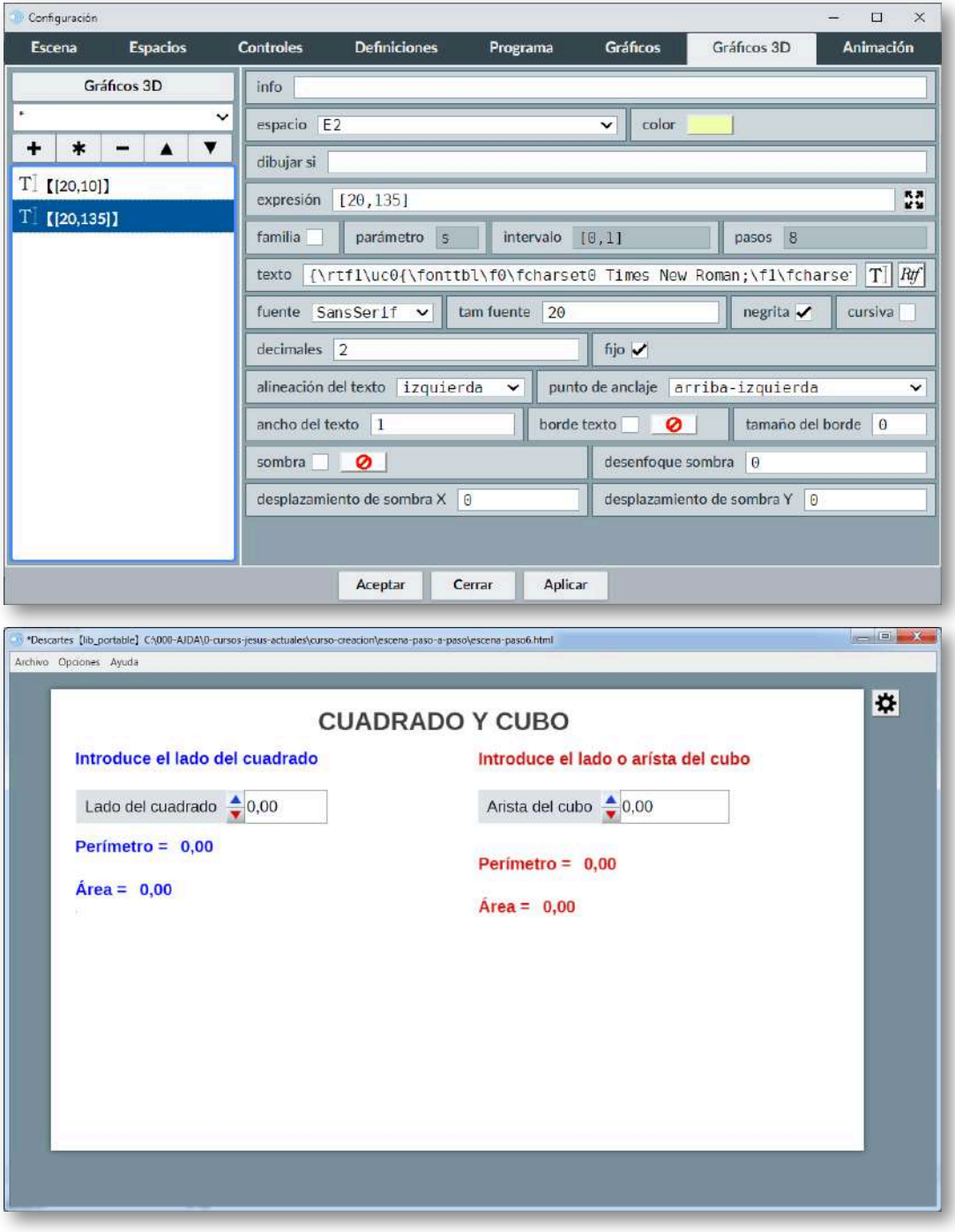
6.17 Mi primera escena paso a paso. Paso 6

Vamos a realizar el paso 6 de la primera escena. Para comenzar, abrimos la escena **escena-paso5** con el editor de **Descartes**.



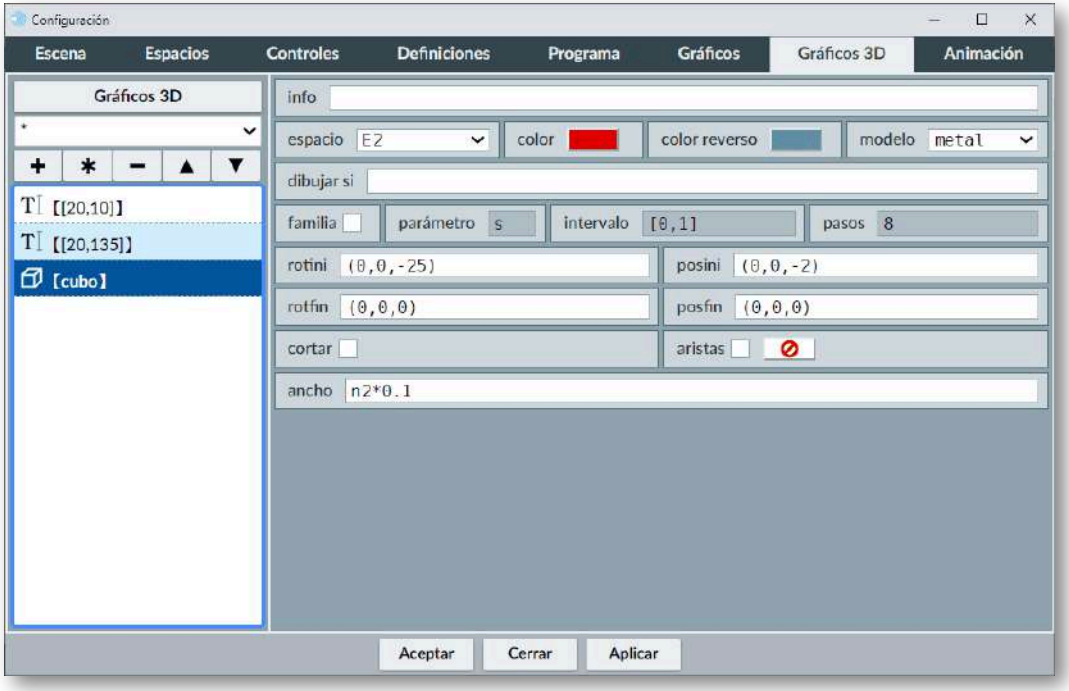
A continuación, añadiremos un texto con el editor de texto enriquecido que muestre el área y volumen del cubo en 3D.





Seguidamente, dibujaremos un cubo de color rojo y estilo metal en el espacio 3D.

Sus dimensiones estarán en función de la arista del cubo, es decir, del valor de **n2**, de la siguiente forma:





Finalmente, hemos completado nuestra primera escena totalmente funcional, en la que se han utilizado de manera práctica los elementos fundamentales de **DescartesJS**, incluyendo gráficos 2D y 3D, animaciones, configuración, elementos de programación y controles interactivos.

A continuación, guardamos la escena final con el nombre **escena-paso6** para conservar el resultado obtenido.



Esta escena es completamente editable y puede modificarse o mejorarse según los deseos y necesidades del creador, permitiendo ajustar gráficos, animaciones y controles para adaptarla a diferentes objetivos o aumentar su complejidad y funcionalidad.

Con "Mi primera escena paso a paso" hemos completado un proyecto interactivo sencillo, pero totalmente funcional, diseñado para servir como guía, ejemplo práctico y base de aprendizaje para futuros desarrollos más complejos.



Capítulo 7

Proyectos con escenas de Descartes

Capítulo 7
Proyectos con escenas de Descartes

7.1 Introducción 551

7.2 Proyecto enfocado hacia la realización de un juego didáctico. Parte de diseño web 552

7.3 Ejemplo de web basada en un juego 555

7.3.1 Diseño y realización de la escena del juego 555

7.3.2 Índice 566

7.3.3 Introducción-presentación del juego 567

7.3.4 Instrucciones 568

7.3.5 Documentación complementaria 569

7.3.6 Integración del juego didáctico en la web 571

7.3.7 Otras posibles aportaciones o variaciones 572

7.3.8 Web del juego completa 573

7.4 Proyecto enfocado hacia la realización de un material didáctico. Parte de diseño web 579

7.5 Ejemplo de web basada en un material didáctico 582

7.5.1 Diseño y realización de las escenas de un material didáctico 582

7.5.2 Introducción-presentación 589

7.5.3 Índice 590

7.5.4 Contenidos 591

7.5.5 Actividades 591

7.5.6 Integración de escenas en la web 592

7.5.7 Web de factores de conversión completa 594

7.6 Publicación de escenas 598

7.7 Recomendaciones para la puesta en práctica 599

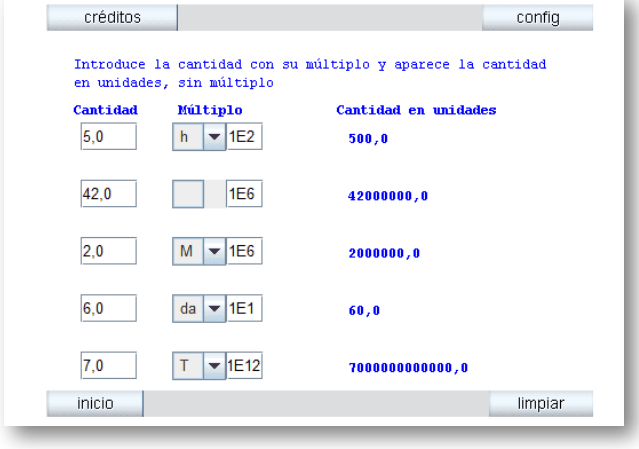
7.8 Ideas y ejemplos sobre proyectos con escenas de Descartes 601

7.1 Introducción

En este último capítulo del libro interactivo trataremos como desarrollar proyectos basados en la utilización de escenas y juegos interactivos. También trataremos aspectos relacionados con la publicación de los materiales elaborados y de su puesta en práctica en el aula.

Por tanto, abordaremos cómo elaborar proyectos educativos orientados a la creación y puesta en práctica en el aula de un juego o material didáctico basado en escenas o animaciones interactivas, como las que se muestran a continuación:





7.2 Proyecto enfocado hacia la realización de un juego didáctico. Parte de diseño web.

La utilización de actividades educativas basadas en juegos didácticos en el aula se ha mostrado como una herramienta docente eficaz, con claros efectos positivos cuando su planteamiento, enfoque y ejecución son adecuados.

En este apartado nos centraremos en los aspectos fundamentales a tener en cuenta al desarrollar la parte del proyecto que consiste en crear una web centrada en un juego didáctico.

La página de referencia sobre juegos didácticos basados en escenas interactivas de **Descartes** se encuentra en la web del Proyecto "Aplicación de juegos didácticos en el aula" (AJDA). En ella se puede acceder a una gran variedad de juegos, materiales relacionados, cursos de formación, generadores y conversores de ficheros de contenido, libros interactivos, vídeo-tutoriales, ejemplos, guías, artículos, enlaces y descargas de materiales. Estos recursos pueden servir como guía o ayuda al elaborar la web de un juego didáctico.



El diseño de la web de un juego se compone de dos partes diferenciadas: la escena o escenas interactivas del propio juego y los elementos complementarios o anexos que acompañan al juego en la web.

Los pasos o aspectos a considerar al diseñar la escena de un juego son los siguientes:

- Definir la idea o diseño previo del juego, detallando el concepto general, el nombre o título, los aspectos básicos de su mecánica y funcionamiento, así como su apariencia y estética general para orientar de manera clara el desarrollo de la escena.
- Identificar los elementos necesarios para la escena del juego (controles, textos, gráficos, espacios, elementos de programación, etc.) y analizar la viabilidad de implementar la idea en la escena.
- Planificar el desarrollo temporal del juego, considerando fases, orden de los eventos y la forma de finalizar el juego.
- Introducir los distintos elementos en el editor de escenas de forma estructurada y coherente.
- Probar el funcionamiento de la escena a medida que se desarrolla, realizando correcciones y mejoras necesarias para optimizar su operación y presentación.
- Una vez que la escena funciona correctamente, evaluar posibles mejoras o elementos que incrementen su calidad y operatividad.
- Estudiar la posibilidad de crear variantes o modalidades distintas del juego para generar dinámicas diferentes dentro del mismo.
- Finalmente, insertar la escena o escenas optimizadas en un lugar adecuado de la página web donde se presentará el juego (página del proyecto).

En cuanto a los elementos complementarios o anexos de la web del juego se recomiendan los siguientes:

- **Presentación e introducción:** incluir el título del juego, autoría, descripción introductoria, licencia o copyright. Se pueden añadir carátulas, imágenes de portada, logos, animaciones, vídeos o presentaciones relacionadas con la temática del juego.
- **Instrucciones o reglas:** redactarlas de forma clara, sencilla y concisa, indicando objetivos del juego, mecánica, proclamación de ganadores y finalización de la partida.
- **Documentos complementarios:** hojas para respuestas, registro de resultados, plantillas, documentos organizativos, elementos narrativos, premios, cartas, insignias, decoración, etc.
- **Escena o escenas del juego:** las escenas se deben integrar en la web, ya sea abiertas directamente o mediante enlaces.
- **Otros elementos opcionales:** cualquier recurso que el autor/a considere útil, como introducción y puesta en escena, música de fondo, diplomas de premios, atrezo, accesorios, entre otros.

Todos estos elementos de la web deben estar integrados de manera coherente. Para ello se recomienda:

- Usar un estilo común y consistente, que abarque tipografía, colores, imágenes, logos y símbolos.
- Definir las distintas páginas que se utilizarán. Lo más habitual es distribuir el proyecto en varias páginas relacionadas mediante un índice, hipervínculos y elementos de navegación. También es posible concentrar todos los elementos en una sola página, lo que evita menús y vínculos, aunque puede generar una página demasiado larga si no se diseña cuidadosamente.
- Ordenar contenidos y páginas de forma lógica y coherente.
- Colocar adecuadamente los elementos de navegación, especialmente los enlaces.

7.3 Ejemplo de web basada en un juego

Para ilustrar cómo puede plantearse y desarrollarse una web basada en juegos didácticos, se presentará un ejemplo completo en el que se mostrará el diseño y la creación de las escenas de un juego, así como la elaboración de la web y de los elementos complementarios en los que se integrarán dichas escenas.

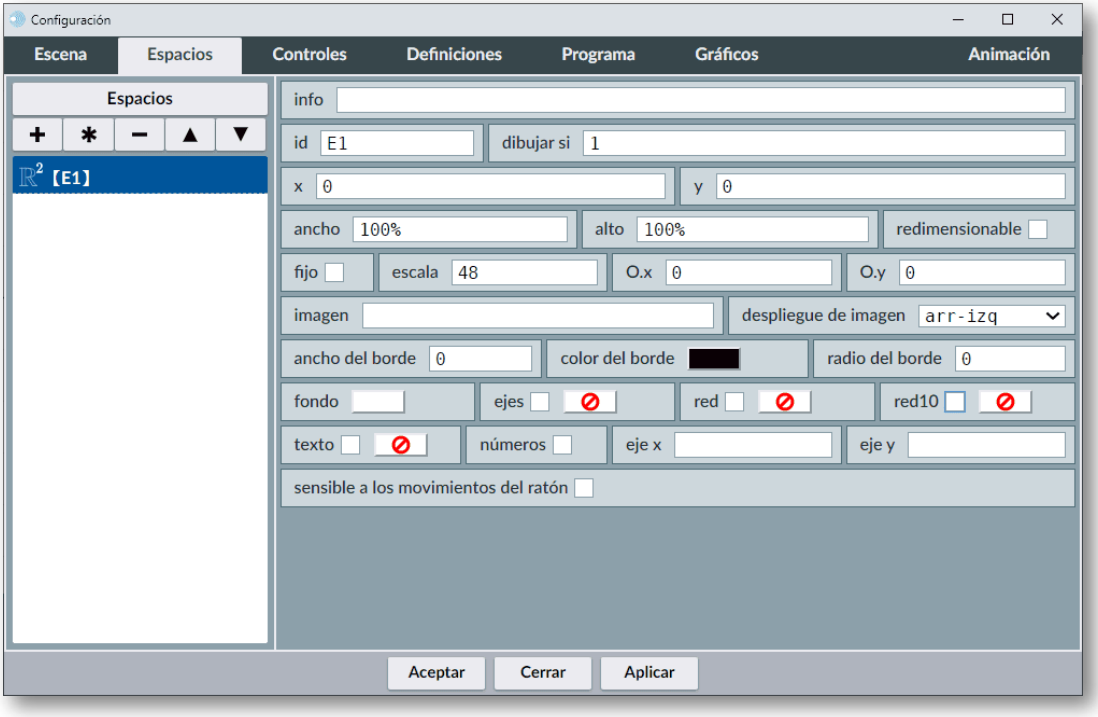
7.3.1 Diseño y realización de la escena del juego

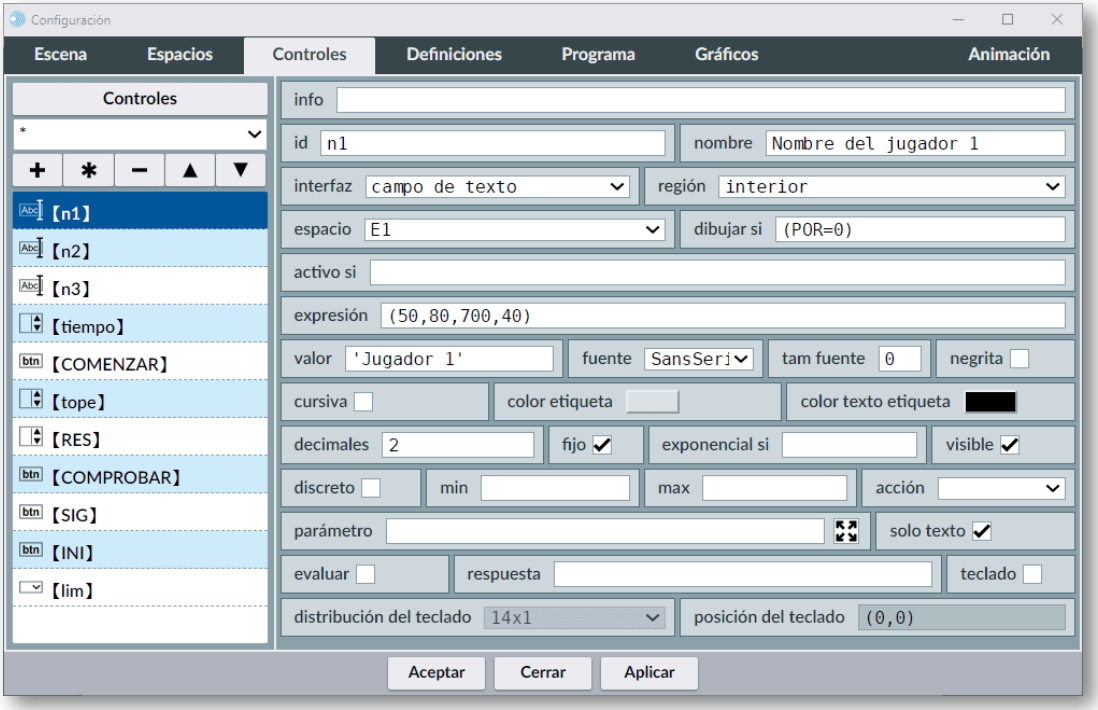
Vamos a diseñar y desarrollar la escena de un juego titulado **"Número secreto"**. A continuación, se describen los principales aspectos considerados para su realización.

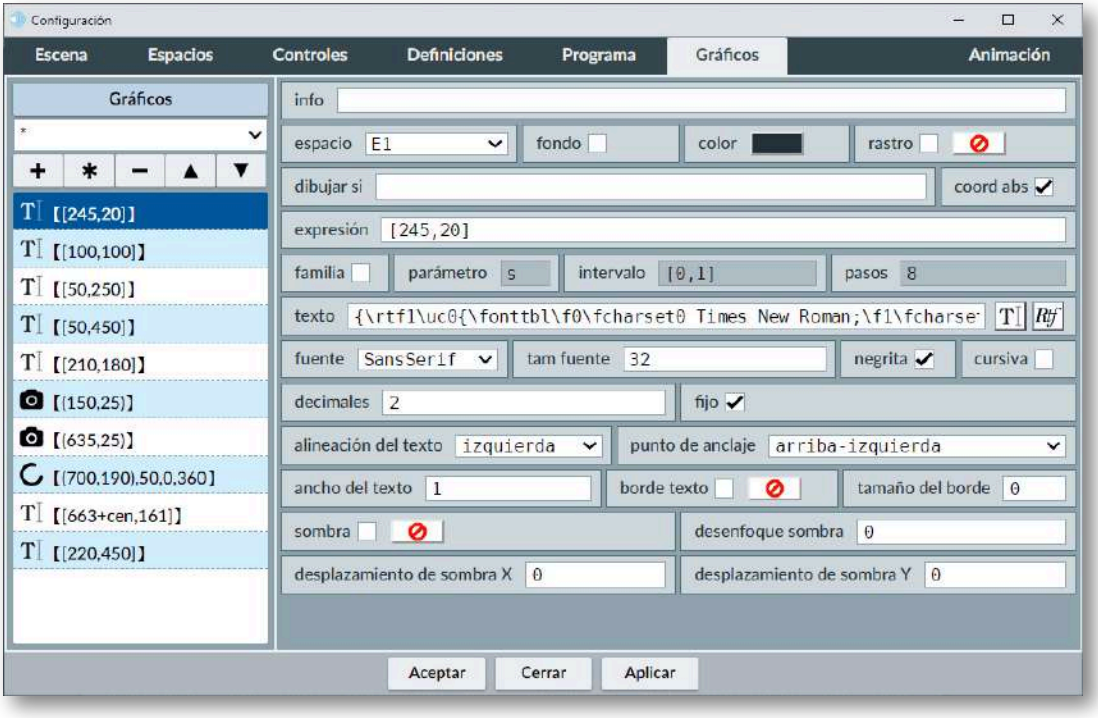
- **Idea o diseño previo.** En el juego participarán tres jugadores con el objetivo de acertar un número secreto elegido aleatoriamente entre cero y una cifra máxima que se establecerá inicialmente como parámetro. Los jugadores participarán por turnos proponiendo una cifra. Tras cada intento se comprobará si el participante ha acertado el número secreto, ganando la partida en tal caso. Si no acierta, se mostrará como pista si el número secreto es mayor o menor que la cifra indicada, pasando entonces el turno al siguiente jugador hasta que alguno acierte el número y gane la partida.
- **Elementos necesarios.** Un espacio 2D que ocupe el 100% de la escena (sin ejes, sin bordes y ni numeración), textos (título del juego, nombre de los jugadores, número secreto, pistas, ganador...), controles numéricos de tipo pulsador, botones y campos de texto (para introducir el nombre de los jugadores, el número máximo, botones de inicio, comprobación, siguiente turno, reinicio...), operaciones de cálculo, comparación y condicionales (obtención del número secreto, comparación de cifras, control de turnos...), así como posibles elementos de sonido e imágenes decorativas.

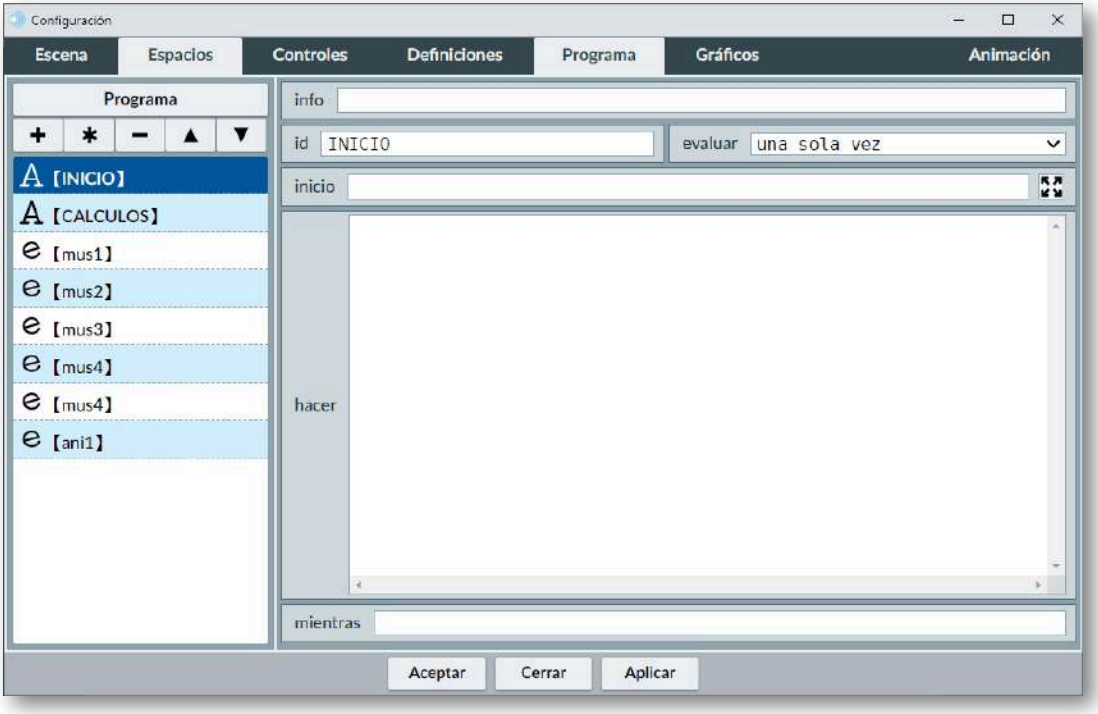
- **Desarrollo temporal.** En primer lugar, se introducen los nombres de los jugadores y la cifra máxima que puede tomar el número secreto. A continuación, los participantes intervienen por turnos, introduciendo una cifra y comprobando si coincide con el número secreto. Si un jugador acierta, gana la partida y el juego finaliza. En caso contrario, se muestra una pista indicando si la cifra introducida es mayor o menor que el número secreto y el turno pasa al siguiente jugador, repitiéndose el proceso hasta que alguien acierte.
- **Elaboración del juego en el editor de escenas.** Finalmente se construye la escena incorporando el espacio, los controles, los gráficos y los elementos de programación necesarios (definiciones y programa).

Incluimos a continuación capturas de pantalla con la configuración de los principales elementos utilizados en la elaboración del juego didáctico.




Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
【n1】
【n2】
【n3】
【tiempo】
【COMENZAR】
【tope】
【RES】
【COMPROBAR】
【SIG】
【INI】
【lim】
info
id n1
nombre Nombre del jugador 1
interfaz campo de texto
región interior
espacio E1
dibujar si (POR=0)
activo si
expresión (50,80,700,40)
valor 'Jugador 1'
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 2
fijo
exponencial si
visible
discreto
min
max
acción
parámetro
solo texto
evaluar
respuesta
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Gráficos
【[245,20]】
【[100,100]】
【[50,250]】
【[50,450]】
【[210,180]】
【(150,25)】
【(635,25)】
【(700,190),50,0,360】
【[663+cen,161]】
【[220,450]】
info
espacio E1
fondo
color
rastro
dibujar si
coord abs
expresión [245,20]
familia
parámetro s
intervalo [0,1]
pasos 8
texto {\rtf1\uc0{\fonttbl\f0\fcharset0 Times New Roman;\f1\fcharse
fuente SansSerif
tam fuente 32
negrita
cursiva
decimales 2
fijo
alineación del texto izquierda
punto de anclaje arriba-izquierda
ancho del texto 1
borde texto
tamaño del borde 0
sombra
desenfoque sombra 0
desplazamiento de sombra X 0
desplazamiento de sombra Y 0
Aceptar
Cerrar
Aplicar

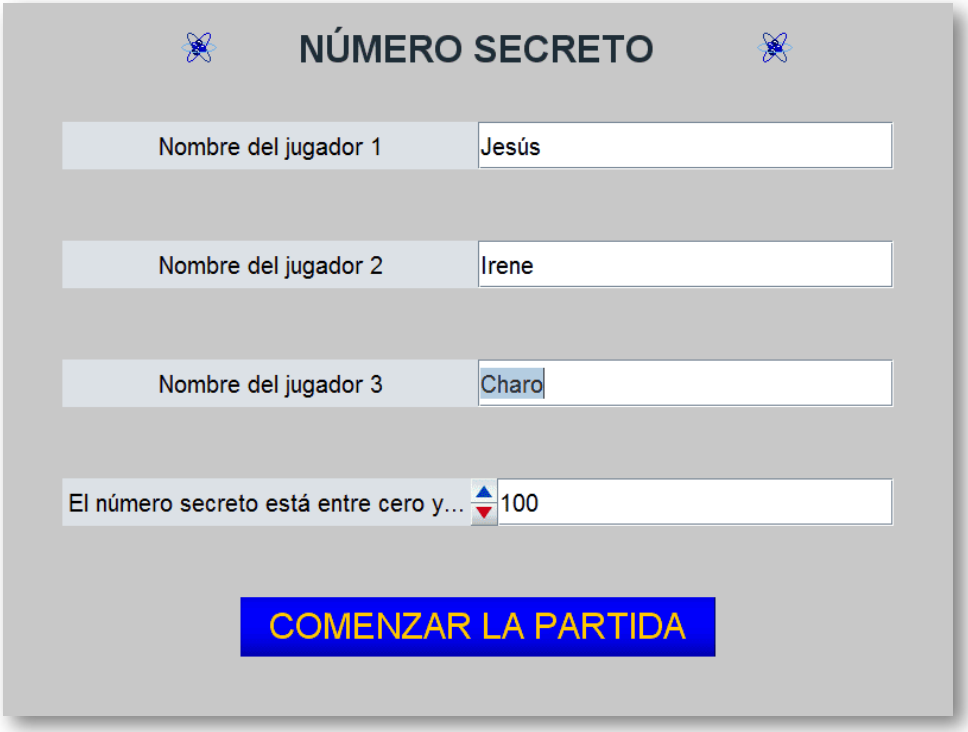


- **Probamos, corregimos y mejoramos** la escena mediante distintas pruebas de funcionamiento, introduciendo los cambios y ajustes necesarios para optimizar su operatividad y presentación.

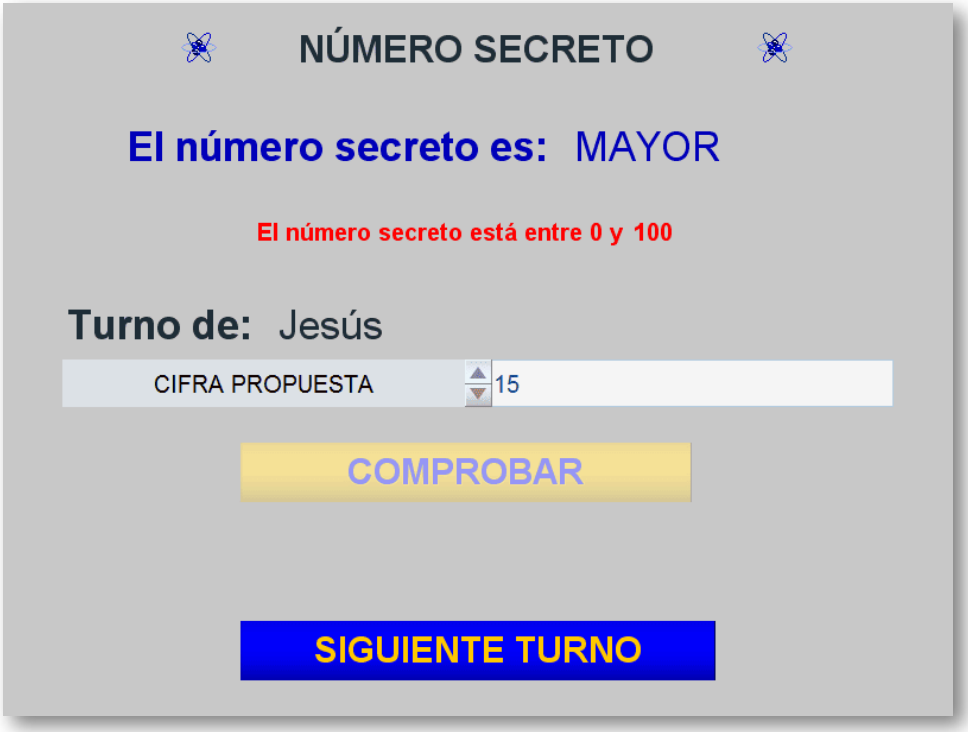
  Es habitual que aparezcan errores o comportamientos no deseados durante la elaboración de la escena, por lo que resulta fundamental realizar varias pruebas hasta lograr el funcionamiento previsto.

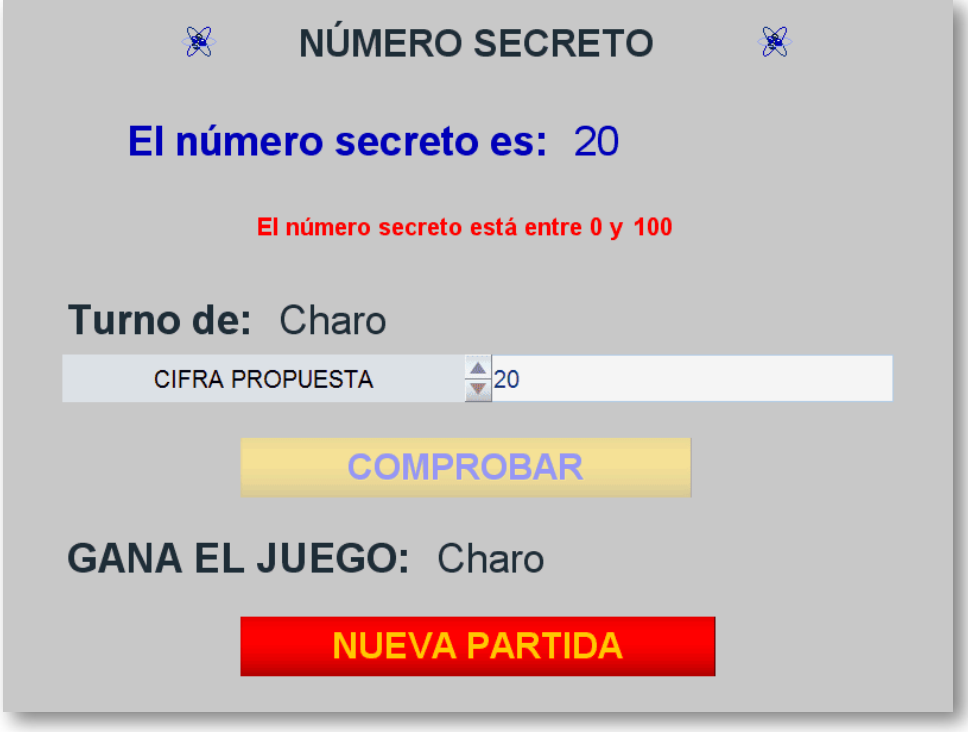
  A continuación, se muestran varias capturas de pantalla del desarrollo de una partida del juego elaborado, con el fin de ilustrar su funcionamiento.

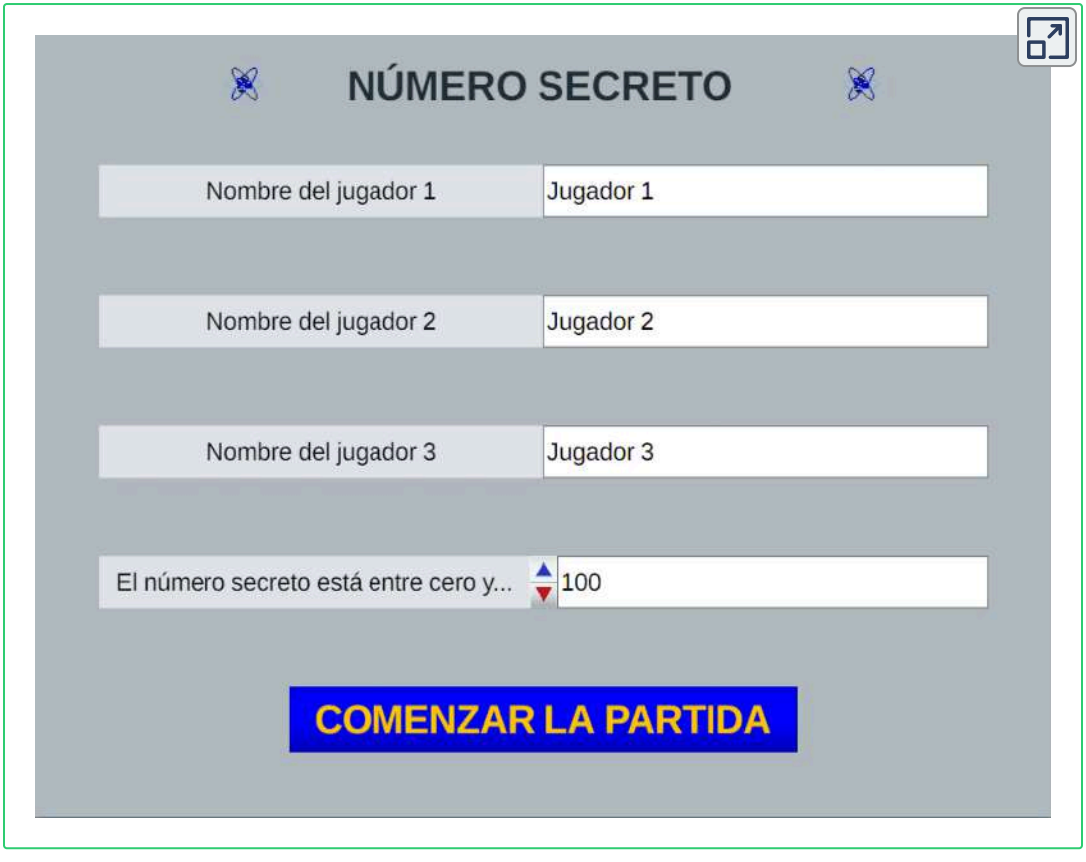
  Finalmente, se incluye la escena interactiva completa del juego didáctico que hemos creado para poder probarlo y utilizarlo directamente.


NÚMERO SECRETO
Nombre del jugador 1
Jesús
Nombre del jugador 2
Irene
Nombre del jugador 3
Charo
El número secreto está entre cero y...
100
COMENZAR LA PARTIDA


NÚMERO SECRETO
El número secreto es:
El número secreto está entre 0 y 100
Turno de: Jesús
CIFRA PROPUESTA
15
COMPROBAR

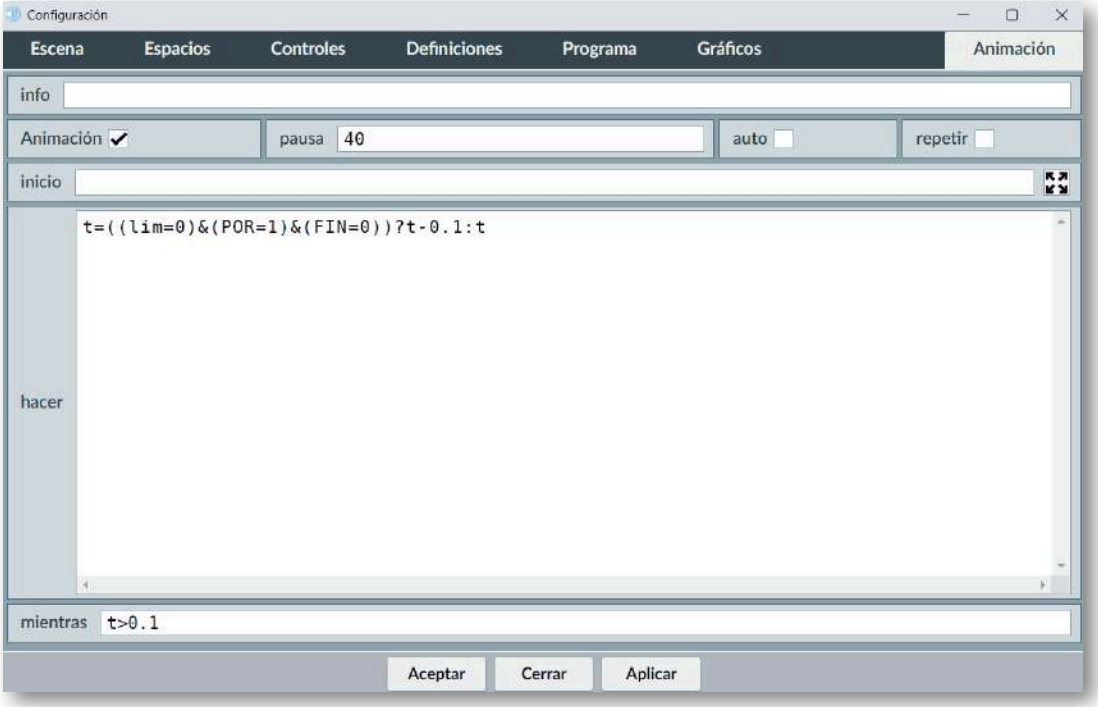

NÚMERO SECRETO
El número secreto es: MAYOR
El número secreto está entre 0 y 100
Turno de: Jesús
CIFRA PROPUESTA
15
COMPROBAR
SIGUIENTE TURNO

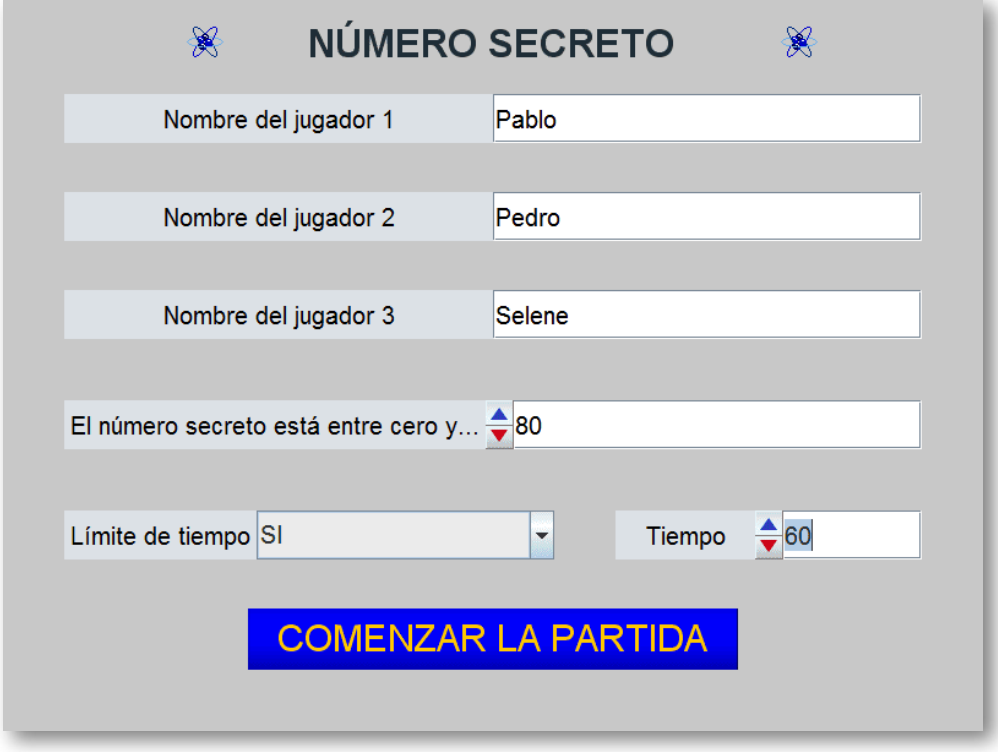

NÚMERO SECRETO
El número secreto es: 20
El número secreto está entre 0 y 100
Turno de: Charo
CIFRA PROPUESTA
20
COMPROBAR
GANA EL JUEGO: Charo
NUEVA PARTIDA

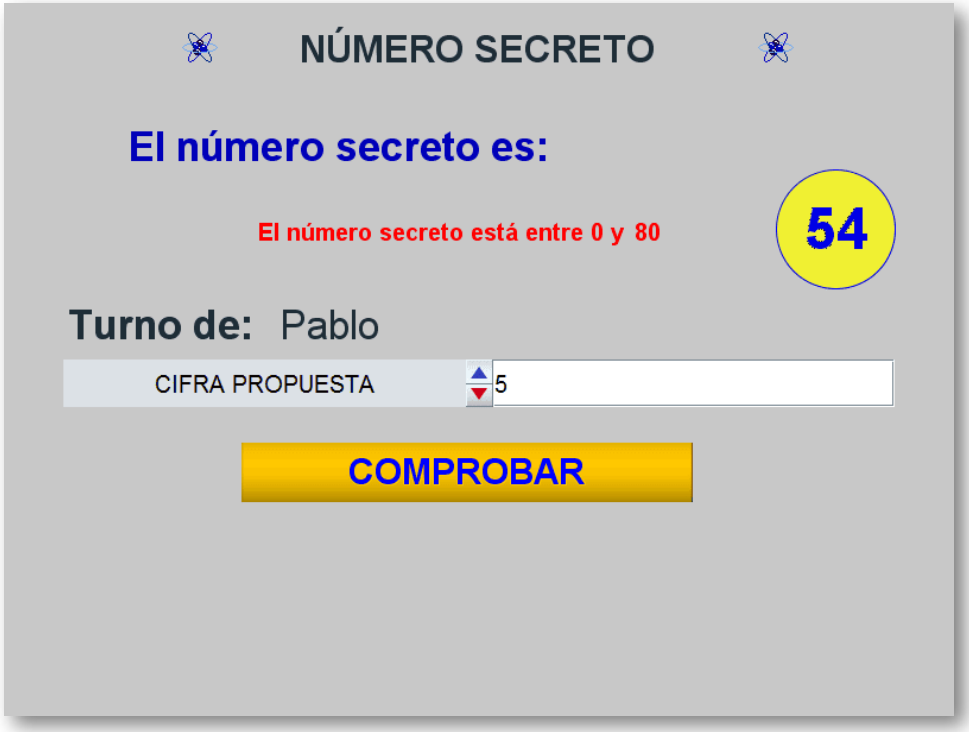


- Estudiar la posibilidad de realizar una **variante del juego**, si es posible, introduciendo mejoras o nuevos elementos. En el ejemplo que estamos tratando se añade la opción de establecer un tiempo límite, configurable como parámetro inicial, para encontrar el número secreto. Si ningún jugador lo descubre dentro del tiempo establecido, la partida finaliza sin ganador. Esta variante requiere incorporar nuevos controles, como un menú, además de nuevos gráficos, elementos de programación y la necesidad de que la escena esté animada mediante el algoritmo del selector **Animación**.

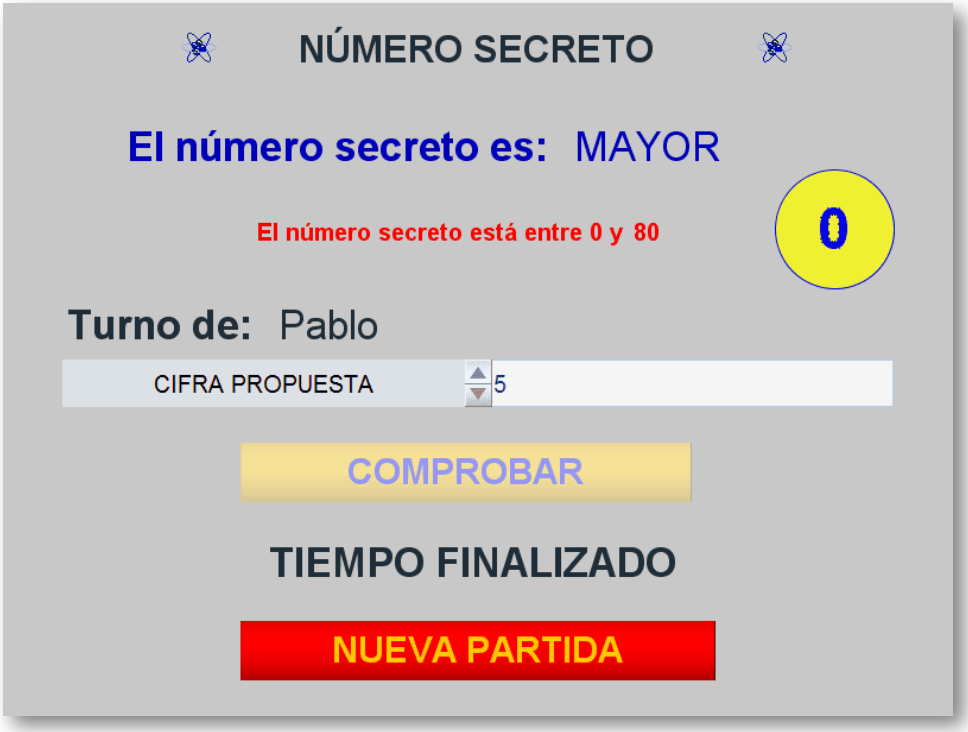
A continuación, se muestran capturas de pantalla del menú de animación y diversas imágenes del desarrollo de una partida del juego y finalmente la escena interactiva del mismo:


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
info
Animación
pausa
40
auto
repetir
inicio
hacer
t=((lim=0)&(POR=1)&(FIN=0))?t-0.1:t
mientras
t>0.1
Aceptar
Cerrar
Aplicar

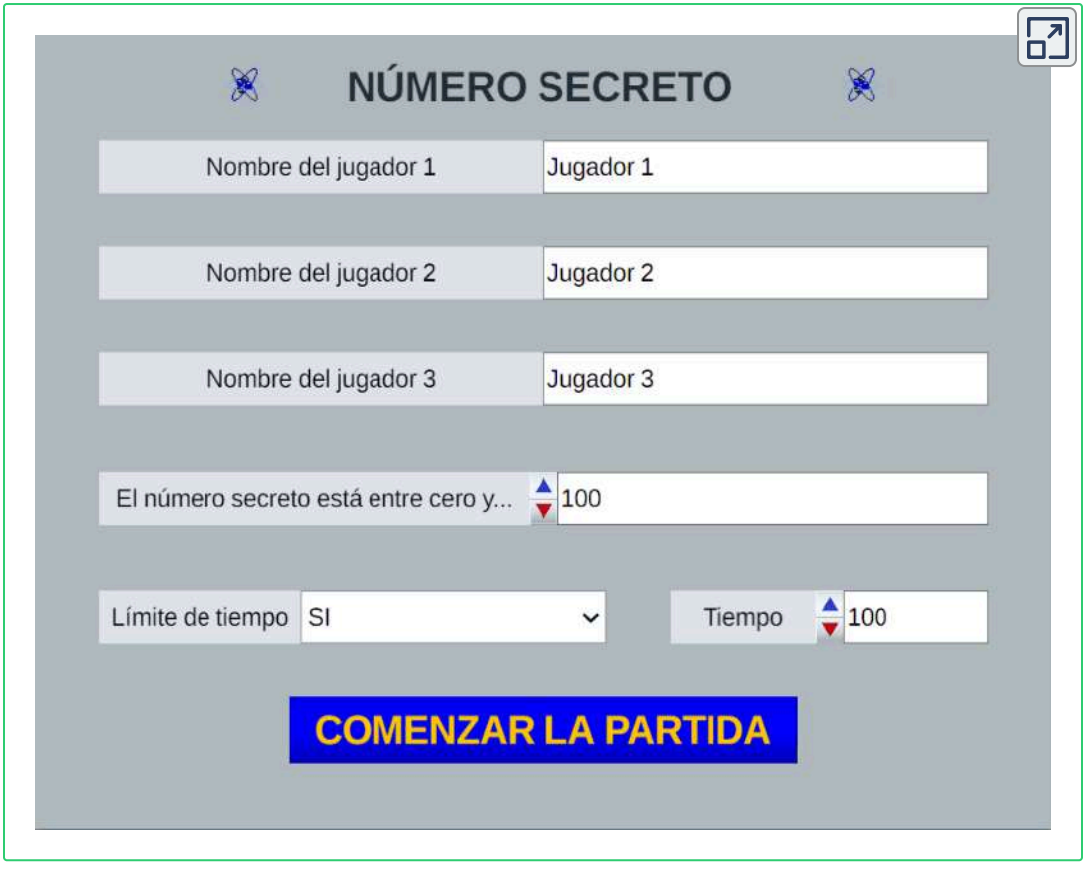

NÚMERO SECRETO
Nombre del jugador 1
Pablo
Nombre del jugador 2
Pedro
Nombre del jugador 3
Selene
El número secreto está entre cero y...
80
Límite de tiempo
SI
Tiempo
60
COMENZAR LA PARTIDA


NÚMERO SECRETO
El número secreto es:
El número secreto está entre 0 y 80
54
Turno de: Pablo
CIFRA PROPUESTA
5
COMPROBAR


NÚMERO SECRETO
El número secreto es: MAYOR
El número secreto está entre 0 y 80
39
Turno de: Pablo
CIFRA PROPUESTA
5
COMPROBAR
SIGUIENTE TURNO


NÚMERO SECRETO
El número secreto es: MAYOR
El número secreto está entre 0 y 80
0
Turno de: Pablo
CIFRA PROPUESTA
5
COMPROBAR
TIEMPO FINALIZADO
NUEVA PARTIDA


NÚMERO SECRETO
El número secreto es: 65
El número secreto está entre 0 y100
26
Turno de: Pablo
CIFRA PROPUESTA
65
COMPROBAR
GANA EL JUEGO: Pablo
NUEVA PARTIDA



- **Las escenas de los juegos deben insertarse en el lugar adecuado de la página web**, copiando el código correspondiente de la escena (que comienza con `<ajs` y finaliza con `</ajs>`) en el fragmento del código HTML donde se desee que aparezca.

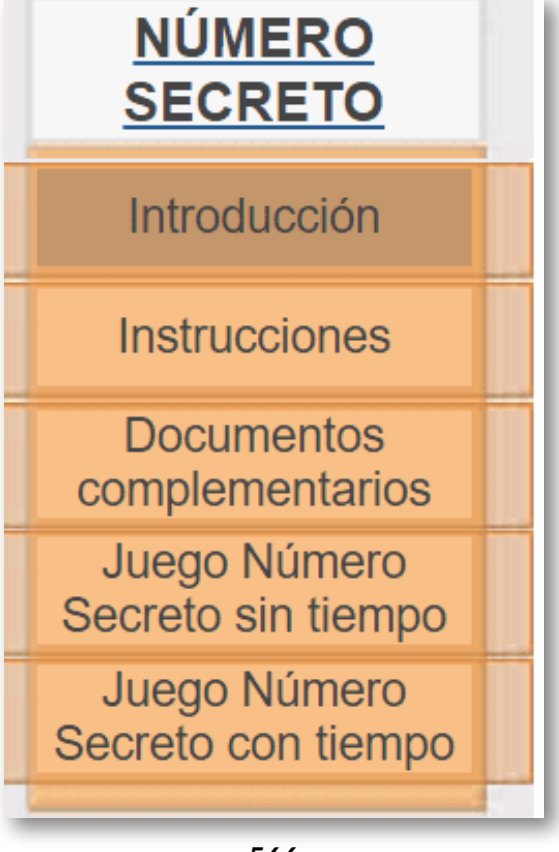
  Es importante recordar que la página web que contenga la escena debe incluir el fichero `descartes-min.js` o hacer referencia a su ubicación en línea. Además, en la propia escena se debe especificar la ruta relativa de este fichero respecto a la página donde se inserta. Asimismo, los ficheros de imagen, audio, vídeo, macro y biblioteca deben acompañar a la escena y encontrarse en las rutas relativas indicadas en la misma.

7.3.2 Índice

Hemos creado un índice que funciona como menú de navegación para toda la web, facilitando el acceso rápido y organizado a cada sección. Su estructura incluye: introducción, instrucciones o reglas del juego, documentos complementarios, juego del **"Número secreto"** sin límite de tiempo y juego del **"Número secreto"** con límite de tiempo.

Contar con un índice no solo mejora la organización de los contenidos, sino que también sirve como elemento de navegación esencial para la experiencia del usuario.

A continuación, se presenta una imagen ilustrativa del índice.



7.3.3 Introducción-presentación del juego

El texto de introducción del juego podría ser el siguiente:

"NÚMERO SECRETO" es un juego en el que participan tres jugadores y que gana el que más habilidad e intuición tenga para encontrar la cifra secreta. Se trata de un juego dinámico con efectos sonoros, sencillo y emocionante.

Además, se añadirá un logo que aparecerá en todas las páginas del juego, así como una carátula específica para la página de presentación. Ambos elementos se muestran a continuación.





También se puede contemplar la posibilidad de realizar un vídeo de presentación, o bien presentarlo mediante diapositivas, un PDF, o incorporando otros elementos interactivos.

7.3.4 Instrucciones

Las instrucciones o reglas del juego que se proponen son las siguientes:

Descripción

Juego en el que participan tres jugadores, quienes deberán encontrar un número secreto cuyo valor estará comprendido entre cero y un valor positivo que se seleccione al inicio de la partida.

Mecánica

Al comienzo, en la pantalla de configuración, los jugadores deberán introducir su nombre y el intervalo del número secreto, y pulsar el botón **"COMENZAR"**.

Los jugadores participan por turnos según el orden de introducción de los nombres. En su turno, cada jugador introduce una cifra en el control **"CIFRA PROPUESTA"** y pulsa **"COMPROBAR"**. Si acierta el número secreto, gana el juego y la partida finaliza. Si no acierta, se ofrece una pista indicando si la cifra propuesta es mayor o menor que el número secreto. A continuación, se pulsa el botón **"SIGUIENTE TURNO"** para pasar al siguiente jugador, quien seguirá el mismo procedimiento.

Existen dos versiones del juego: una sin límite de tiempo, en la que los participantes pueden probar indefinidamente hasta acertar, y otra con control de tiempo, en la que se establece un tiempo máximo. Si el tiempo se agota antes de que algún jugador encuentre el número secreto, la partida termina sin ganador. En tal caso se podrá jugar otra partida de desempate.

Objetivo

Ser el primer participante en encontrar el número secreto.

7.3.5 Documentación complementaria

Para registrar las respuestas que los participantes introducen en sus respectivos turnos, se proporciona un documento donde pueden anotarse por escrito las cifras propuestas, así como si coinciden con el número secreto, o si son mayores o menores que este.

Estos formularios pueden utilizarse tanto en formato papel como en formato digital.



FORMULARIO DE RESPUESTAS. NÚMERO SECRETO

NOMBRE:		FECHA:	
CURSO:		GRUPO:	

NÚMERO	CIFRA DADA POR EL JUGADOR	LA CIFRA DADA ES MAYOR, MENOR O EXACTA
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Estos materiales no son obligatorios, pero aportan un valor añadido al juego, mejorando su dinámica y su puesta en práctica.

El siguiente formulario permite al docente llevar un registro de los resultados de las distintas partidas realizadas por los participantes. De igual forma, el documento puede utilizarse en formato impreso o completarse digitalmente.

FORMULARIO DE CONTROL. NÚMERO SECRETO

Jugadores		**FECHA:**	
CURSO:		**GRUPO:**	

Nombre de los jugadores	Número de partidas ganadas	Número de partidas perdidas

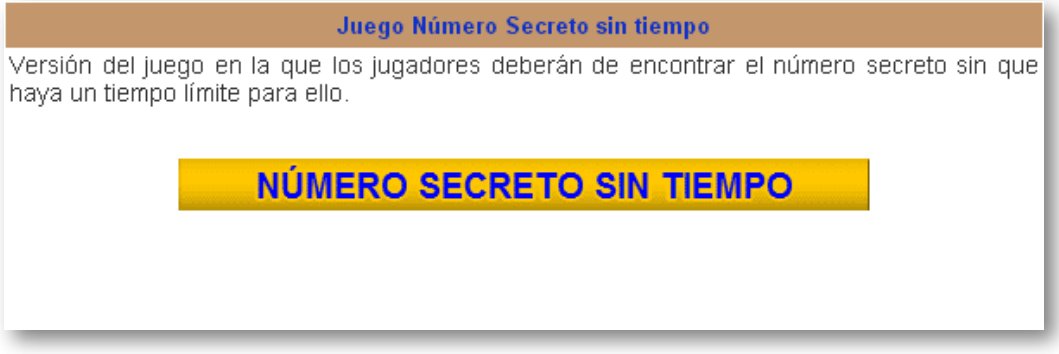
Además, se pueden diseñar otros elementos complementarios que hagan la experiencia del juego más interactiva y atractiva, como cartulinas con las respuestas que los jugadores levantan simultáneamente, diplomas para premiar la participación o los aciertos, insignias, cartas de juego, material audiovisual u otros recursos visuales y lúdicos que refuercen la dinámica del juego y su puesta en práctica.

7.3.6 Integración del juego didáctico en la web

Las escenas de los juegos se pueden insertar en la página web deseada copiando el código de la escena (que comienza con `<ajs` y finaliza con `</ajs>`) en la sección del código HTML donde se quiera mostrar. Es importante recordar que la página que contenga la escena debe incluir el fichero `descartes-min.js`, y que en la propia escena se debe indicar la ruta relativa de este fichero respecto a la ubicación de la página.

La escena puede incrustarse directamente en la web o mediante un botón (que en sí es otra escena), cuya función es abrir la página del juego en una nueva pestaña del navegador. También se puede utilizar un enlace web (URL) que abra el juego directamente.

En nuestro ejemplo, hemos optado por usar botones, creados como escenas de Descartes, para abrir las diferentes escenas del juego en nuevas pestañas del navegador.





7.3.7 Otras posibles aportaciones o variaciones

A continuación, presentamos algunas propuestas de variación o modificación que podrían implementarse sobre este juego, entre muchas otras posibles:

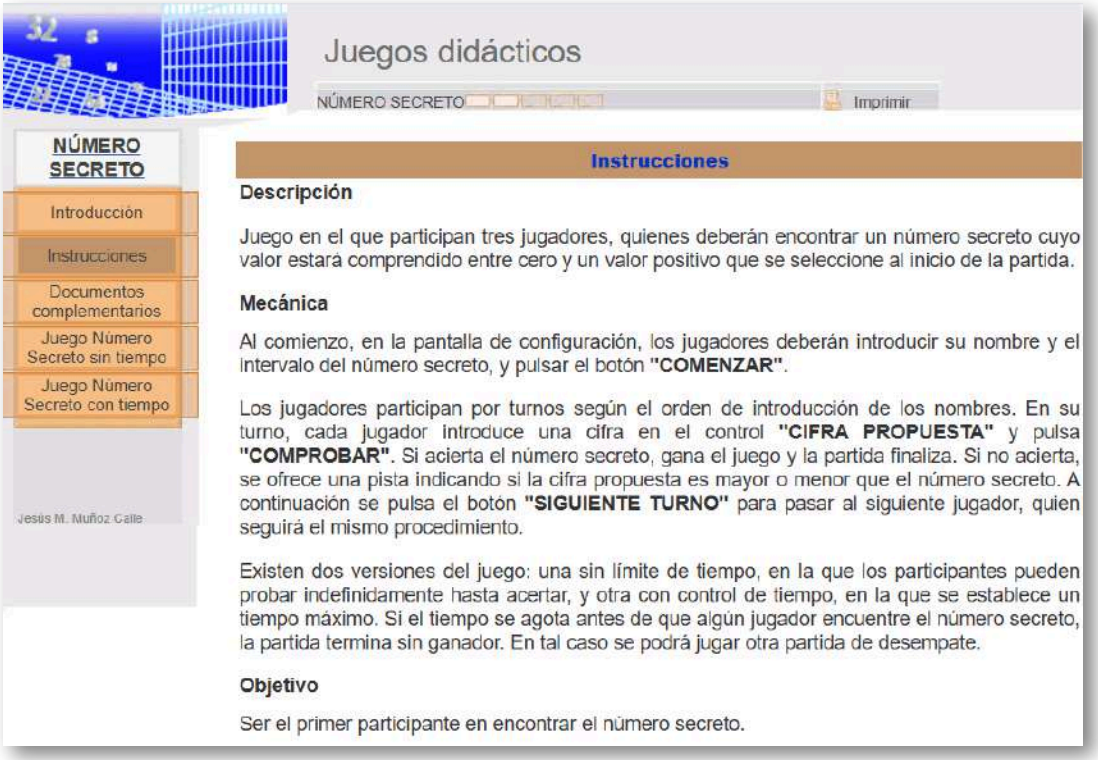
- Permitir variar el número de jugadores participantes, adaptando la dinámica a diferentes grupos.
- Establecer que el orden de participación de los jugadores sea aleatorio, aumentando la imprevisibilidad del juego.
- Ofrecer distintos tipos de pistas sobre el número secreto, por ejemplo: si es divisible por otro número, cuántos dígitos tiene, o conocer el valor de las unidades, decenas o centenas.
- Asignar puntuaciones a los jugadores, otorgando puntos extra al participante que se acerque más al número secreto en cada turno.
- Eliminar en cada ronda a los jugadores que se alejen más del número secreto, aumentando la competitividad.
- Dar a los jugadores un crédito que disminuya según la distancia de su cifra propuesta al número secreto, eliminando a aquellos que agoten su crédito.
- Introducir niveles de dificultad ajustables, modificando el intervalo del número secreto o el tiempo disponible para acertarlo.
- Incorporar elementos visuales o sonoros adicionales como efectos especiales, animaciones o sonidos al acertar o fallar.
- Permitir partidas cooperativas, en las que los jugadores trabajen en equipo para encontrar el número secreto antes de que se agote el tiempo.
- Registrar estadísticas de las partidas, como número de intentos, tiempo empleado o puntuaciones acumuladas, para analizar el rendimiento de los jugadores.

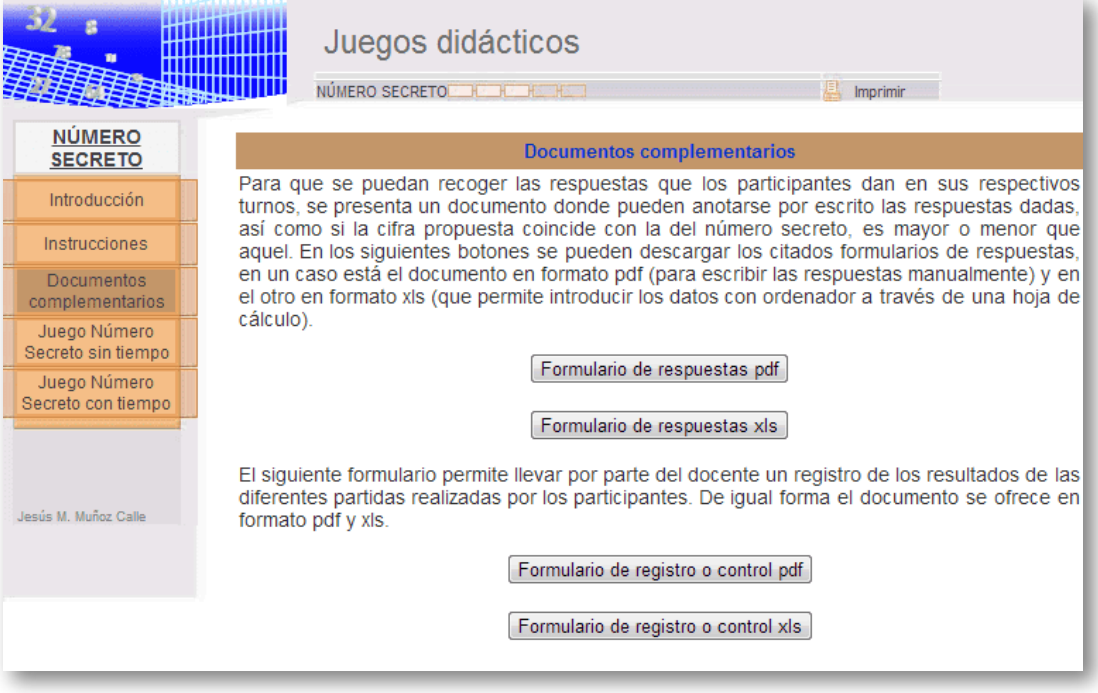
7.3.8 Web del juego completa

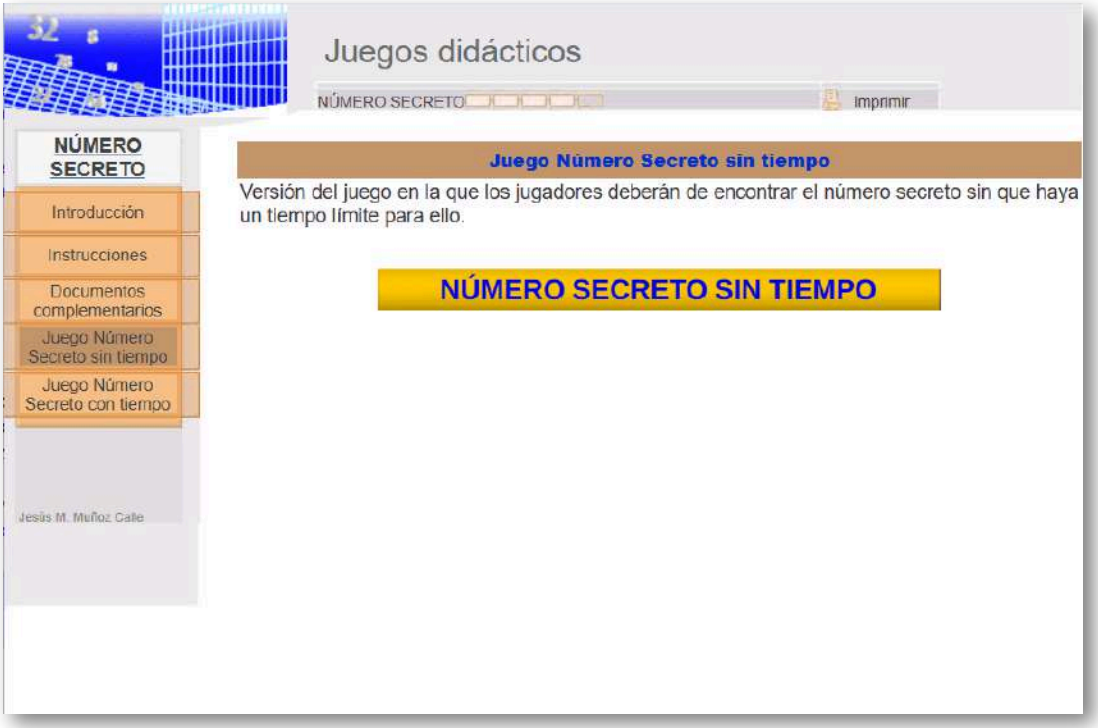
A continuación, se presenta un ejemplo de la web completa del juego, organizada en cinco páginas, con un índice vertical izquierdo común, un estilo consistente y coherente, y un orden lógico que facilita la navegación. En primer lugar, se sitúa la introducción, seguida de las instrucciones, luego los documentos de registro y control, y finalmente las escenas de los juegos en sus distintos formatos y modalidades, incluyendo tanto la versión sin límite de tiempo como la versión con tiempo limitado.

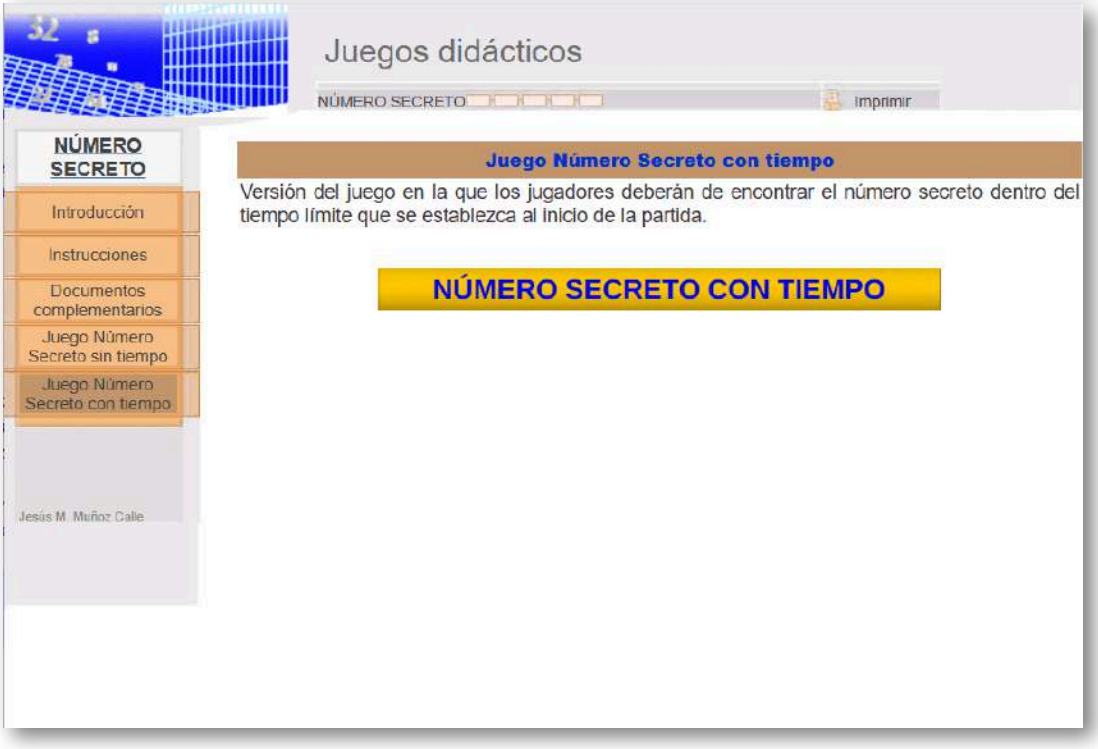
Por último, se mostrará la página web completa del proyecto de este juego didáctico, totalmente funcional y operativa, lista para ser utilizada y explorada por los usuarios.




Juegos didácticos
NÚMERO SECRETO
Imprimir
NÚMERO SECRETO
Introducción
Instrucciones
Documentos complementarios
Juego Número Secreto sin tiempo
Juego Número Secreto con tiempo
Jesús M. Muñoz Calle
Instrucciones
Descripción
Juego en el que participan tres jugadores, quienes deberán encontrar un número secreto cuyo valor estará comprendido entre cero y un valor positivo que se seleccione al inicio de la partida.
Mecánica
Al comienzo, en la pantalla de configuración, los jugadores deberán introducir su nombre y el intervalo del número secreto, y pulsar el botón "COMENZAR".
Los jugadores participan por turnos según el orden de introducción de los nombres. En su turno, cada jugador introduce una cifra en el control "CIFRA PROPUESTA" y pulsa "COMPROBAR". Si acierta el número secreto, gana el juego y la partida finaliza. Si no acierta, se ofrece una pista indicando si la cifra propuesta es mayor o menor que el número secreto. A continuación se pulsa el botón "SIGUIENTE TURNO" para pasar al siguiente jugador, quien seguirá el mismo procedimiento.
Existen dos versiones del juego: una sin límite de tiempo, en la que los participantes pueden probar indefinidamente hasta acertar, y otra con control de tiempo, en la que se establece un tiempo máximo. Si el tiempo se agota antes de que algún jugador encuentre el número secreto, la partida termina sin ganador. En tal caso se podrá jugar otra partida de desempate.
Objetivo
Ser el primer participante en encontrar el número secreto.


Juegos didácticos
NÚMERO SECRETO
Imprimir
NÚMERO SECRETO
Introducción
Instrucciones
Documentos complementarios
Juego Número Secreto sin tiempo
Juego Número Secreto con tiempo
Jesús M. Muñoz Calle
Documentos complementarios
Para que se puedan recoger las respuestas que los participantes dan en sus respectivos turnos, se presenta un documento donde pueden anotarse por escrito las respuestas dadas, así como si la cifra propuesta coincide con la del número secreto, es mayor o menor que aquel. En los siguientes botones se pueden descargar los citados formularios de respuestas, en un caso está el documento en formato pdf (para escribir las respuestas manualmente) y en el otro en formato xls (que permite introducir los datos con ordenador a través de una hoja de cálculo).
Formulario de respuestas pdf
Formulario de respuestas xls
El siguiente formulario permite llevar por parte del docente un registro de los resultados de las diferentes partidas realizadas por los participantes. De igual forma el documento se ofrece en formato pdf y xls.
Formulario de registro o control pdf
Formulario de registro o control xls

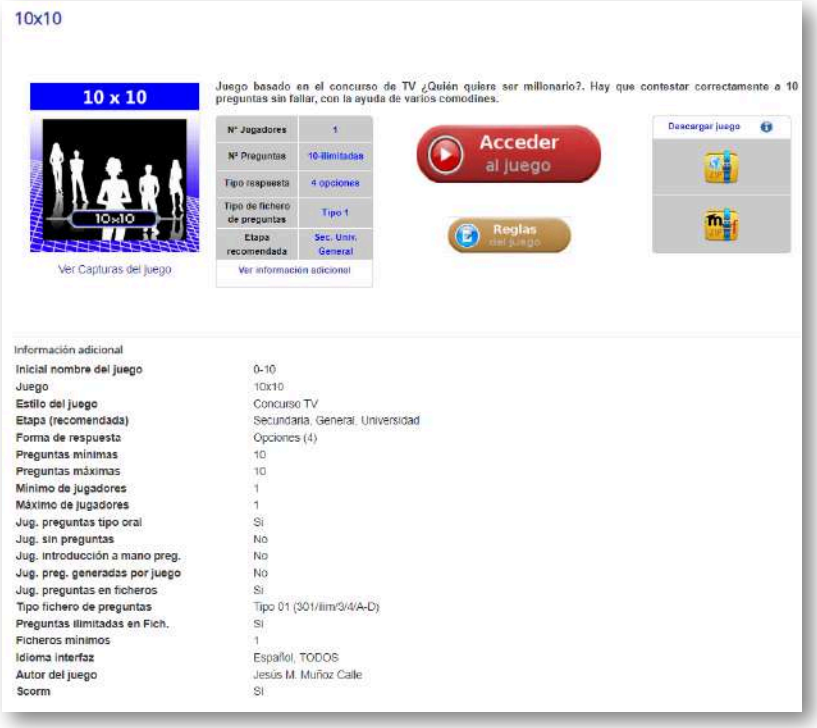

Juegos didácticos
NÚMERO SECRETO
Imprimir
NÚMERO SECRETO
Introducción
Instrucciones
Documentos complementarios
Juego Número Secreto sin tiempo
Juego Número Secreto con tiempo
Jesús M. Muñoz Calle
Juego Número Secreto sin tiempo
Versión del juego en la que los jugadores deberán de encontrar el número secreto sin que haya un tiempo límite para ello.
NÚMERO SECRETO SIN TIEMPO

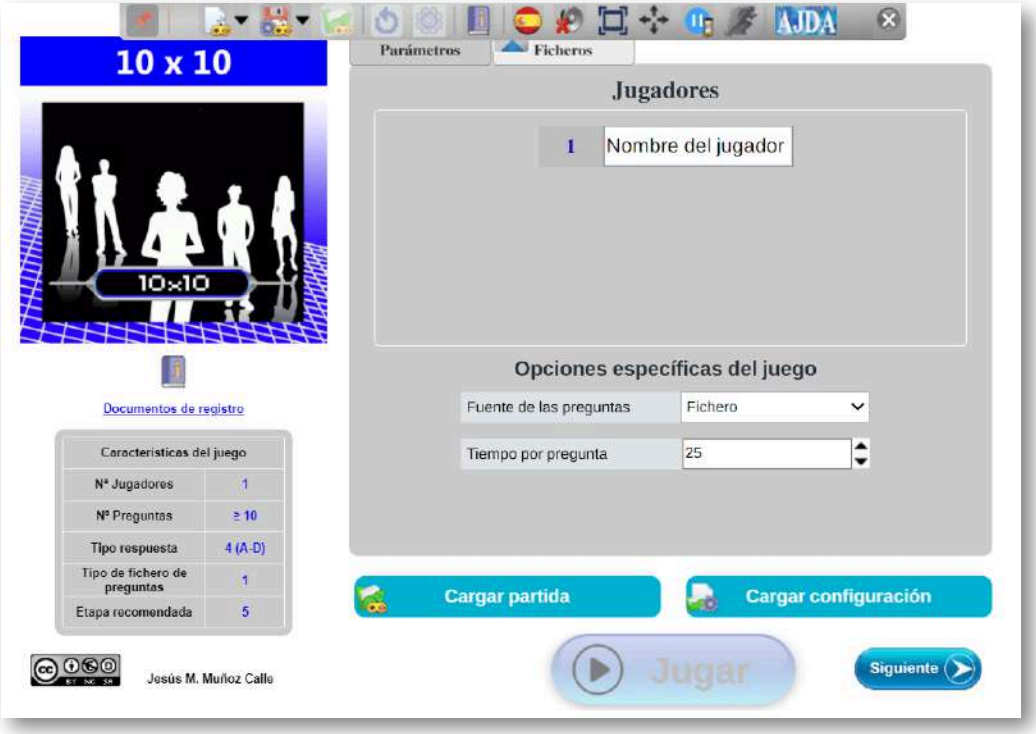

Juegos didácticos
NÚMERO SECRETO
Imprimir
NÚMERO SECRETO
Introducción
Instrucciones
Documentos complementarios
Juego Número Secreto sin tiempo
Juego Número Secreto con tiempo
Jesús M. Muñoz Calle
Juego Número Secreto con tiempo
Versión del juego en la que los jugadores deberán de encontrar el número secreto dentro del tiempo límite que se establezca al inicio de la partida.
NÚMERO SECRETO CON TIEMPO

A continuación, se presenta la página web completa del proyecto del juego didáctico **"Número secreto"**, incluyendo todas sus secciones: introducción, instrucciones, documentos de registro y control, y las escenas del juego en sus distintas modalidades. La web se ha diseñado con un estilo coherente, un índice de navegación claro y elementos interactivos que facilitan la experiencia del usuario.



Presentamos otro ejemplo de página web con juegos didácticos, en este caso correspondiente al juego **"10x10"** del Proyecto AJDA. Este modelo de diseño se ha empleado como referencia para la estructura y organización de todos los juegos del proyecto, proporcionando un estilo uniforme, navegación clara y una experiencia de usuario coherente y consistente.

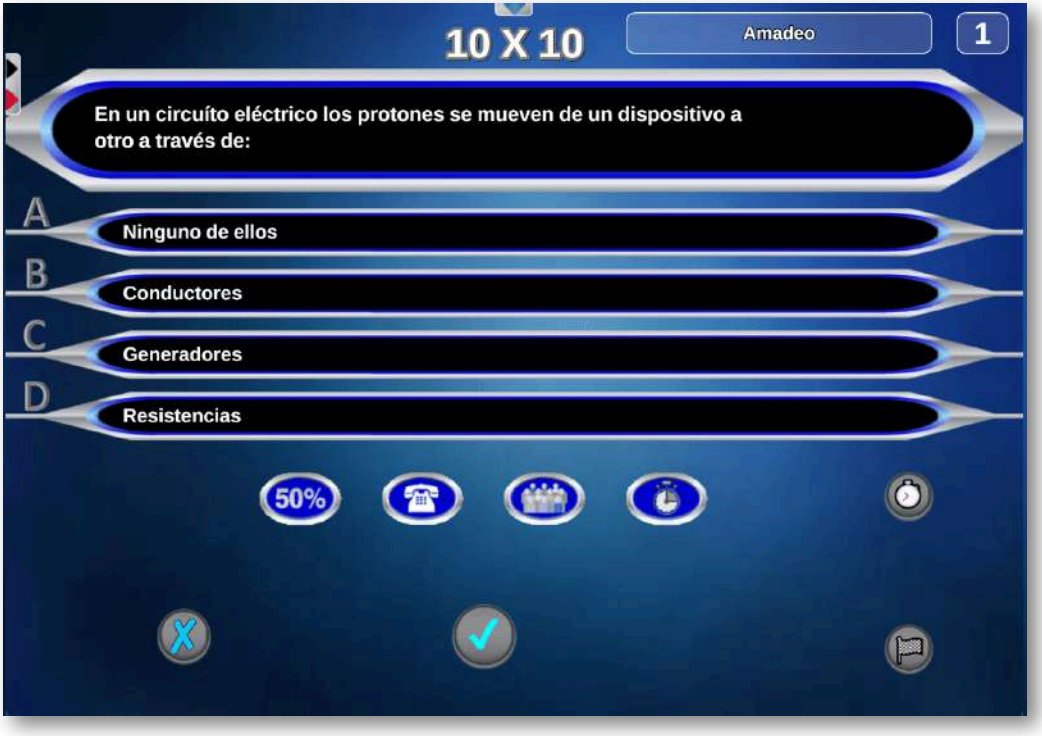

10x10
10 x 10
Juego basado en el concurso de TV ¿Quién quiere ser millonario?. Hay que contestar correctamente a 10 preguntas sin fallar, con la ayuda de varios comodines.
Nº Jugadores 1
Nº Preguntas 10-Ilimitadas
Tipo respuesta 4 opciones
Tipo de fichero de preguntas Tipo 1
Etapa recomendada Sec. Univ. General
Ver información adicional
Acceder al juego
Reglas del juego
Descargar juego
Ver Capturas del juego
Información adicional
Inicial nombre del juego 0-10
Juego 10x10
Estilo del juego Concurso TV
Etapa (recomendada) Secundaria, General, Universidad
Forma de respuesta Opciones (4)
Preguntas mínimas 10
Preguntas máximas 10
Mínimo de jugadores 1
Máximo de jugadores 1
Jug. preguntas tipo oral Si
Jug. sin preguntas No
Jug. introducción a mano preg. No
Jug. preg. generadas por juego No
Jug. preguntas en ficheros Si
Tipo fichero de preguntas Tipo 01 (301/lim/3/4/A-D)
Preguntas ilimitadas en Fich. Si
Ficheros mínimos 1
Idioma interfaz Español, TODOS
Autor del juego Jesús M. Muñoz Calle
Scorm Si


AJDA
10 x 10
Parámetros
Ficheros
Jugadores
1
Nombre del jugador
Opciones específicas del juego
Fuente de las preguntas
Fichero
Tiempo por pregunta
25
Documentos de registro
Características del juego
Nº Jugadores 1
Nº Preguntas ≥ 10
Tipo respuesta 4 (A-D)
Tipo de fichero de preguntas 1
Etapa recomendada 5
Cargar partida
Cargar configuración
Jugar
Siguiente
Jesús M. Muñoz Calle


10 x 10
10x10
Documentos de registro
Características del juego
Nº Jugadores 1
Nº Preguntas ≥ 10
Tipo respuesta 4 (A-D)
Tipo de fichero de preguntas 1
Etapa recomendada 5
Jesús M. Muñoz Calle
Parámetros
Ficheros
Selección del fichero de preguntas
Repositorio AJDA
-----Seleccionar fichero de la lista-----
Enlace web (URL)
OK
Fichero local
Seleccionar archivo
Ninguno archivo selec.
Estado carga:
Cargar fichero
Requisitos
Autoria:

Tema:

Nº Preguntas:
≥10
Tipo:
1
Orden aleatorio de preguntas
SI
Empezar por la pregunta
0
Orden aleatorio de opciones
SI
Barajar preguntas de nuevo
NO
Cargar partida
Cargar configuración
Anterior
Jugar


10 X 10
Amadeo
1
En un circuíto eléctrico los protones se mueven de un dispositivo a otro a través de:
A
Ninguno de ellos
B
Conductores
C
Generadores
D
Resistencias
50%

7.4 Proyecto enfocado hacia la realización de un material didáctico. Parte de diseño web.

La utilización de actividades educativas basadas en escenas interactivas contribuye a fomentar la participación activa de los estudiantes, facilita la comprensión de conceptos complejos mediante la visualización y manipulación de elementos, potencia el aprendizaje significativo y mejora la motivación al incorporar elementos participativos y dinámicos en el aula. Además, permite una evaluación más directa de la comprensión y desempeño de los alumnos a través de la interacción con los contenidos.

Diferenciaremos entre el diseño de las escenas interactivas y el diseño del resto de la página web, ya que las escenas requieren un enfoque centrado en la interacción, la programación de elementos y la visualización dinámica, mientras que el resto de la página se enfoca en la presentación de contenidos, navegación, estructura y elementos complementarios que acompañan a las escenas.

Los pasos o aspectos a considerar a la hora de diseñar escenas interactivas son los siguientes:

- **Definir la idea o diseño previo de la escena:** tener clara la escena que se desea crear, ya sea para explicar un concepto, realizar una animación sobre un tema, un ejercicio interactivo, una actividad de evaluación, etc.
- **Identificar los elementos necesarios:** controles, textos, gráficos, elementos de programación, espacios, entre otros, para evaluar la viabilidad de la escena y determinar si se puede desarrollar la idea prevista.
- **Planificar el desarrollo temporal o secuencial:** organizar la escena según su secuencia de interacción y los distintos momentos de aparición de elementos.

- **Introducir los elementos en el editor de escenas:** colocar los componentes de forma estructurada y coherente, respetando la relación entre ellos.
- **Probar y ajustar la escena:** realizar pruebas continuas para detectar errores, corregirlos y optimizar la funcionalidad y presentación.
- **Mejorar y enriquecer la escena:** añadir elementos o funcionalidades adicionales que incrementen la calidad, la operatividad y la experiencia de usuario.
- **Explorar variantes o nuevas escenas:** considerar posibles modalidades o adaptaciones de la escena para distintos contextos o actividades, o crear escenas relacionadas.
- **Integrar la escena en la web:** una vez realizada y optimizada, insertar la escena en el lugar adecuado dentro de la página web del material didáctico.

En cuanto a los elementos de la web del material didáctico, se recomienda incluir los siguientes:

- **Presentación e introducción:** título del material, nombre del autor o autora, descripción introductoria, licencia o copyright. También pueden incluirse imágenes de portada, logotipos, animaciones, vídeos, presentaciones u otros recursos relacionados con la temática del material.
- **Índice del material:** listado secuencial de los contenidos de la web, preferiblemente enlazando cada apartado con su página correspondiente. El índice puede presentarse en forma de menú de navegación.
- **Contenidos propios del material:** páginas con el contenido principal que se desea desarrollar, incluyendo las escenas interactivas originales creadas para el material.
- **Actividades:** apartado destinado a ejemplos, ejercicios o actividades de evaluación.

- **Otros contenidos o aspectos relacionados:** enlaces, artículos, materiales complementarios, anexos, u otros recursos que enriquezcan el material.
- **Bibliografía y referencias:** documentación y fuentes utilizadas en la elaboración del material.
- **Otros elementos opcionales:** cualquier recurso adicional que el autor o autora considere adecuado para mejorar la comprensión, presentación o funcionalidad del material.

Todos estos elementos de la web deben estar integrados de manera coherente. Para ello se recomienda:

- **Usar un estilo común y consistente:** aplicar uniformidad en tipografía, colores, imágenes, logotipos y símbolos.
- **Definir el número de páginas:** lo más habitual es distribuir el proyecto en varias páginas relacionadas mediante un índice común, hipervínculos y elementos de navegación. También es posible concentrar todos los elementos en una sola página; esta opción evita menús y enlaces, permitiendo tener a mano todos los elementos, pero puede generar un documento demasiado largo si no se organiza adecuadamente.
- **Ordenar los contenidos y páginas:** estructurar la información de manera lógica y coherente, facilitando la comprensión y navegación.
- **Colocar correctamente los elementos de navegación:** especialmente los enlaces, para asegurar que los usuarios puedan acceder fácilmente a todos los apartados del material.
- **Verificar la compatibilidad y accesibilidad:** asegurarse de que la web funcione correctamente en distintos navegadores y dispositivos, y que sea accesible.
- **Realizar pruebas y mantenimiento periódico:** comprobar su correcto funcionamiento y realizar las actualizaciones que se consideren oportunas.

7.5 Ejemplo de web basada en un material didáctico

Para ilustrar cómo puede enfocarse y ejecutarse una web basada en un material didáctico interactivo, presentamos un ejemplo completo en el que desarrollaremos tanto el diseño y creación de las escenas del material como la web y sus elementos asociados y complementarios.

Nuestro ejemplo concreto se centrará en los **"Factores de conversión"**, operaciones que permiten realizar de forma sencilla cambios de unidades, tales como conversiones de monedas, tiempo, longitud, masa, entre otras.

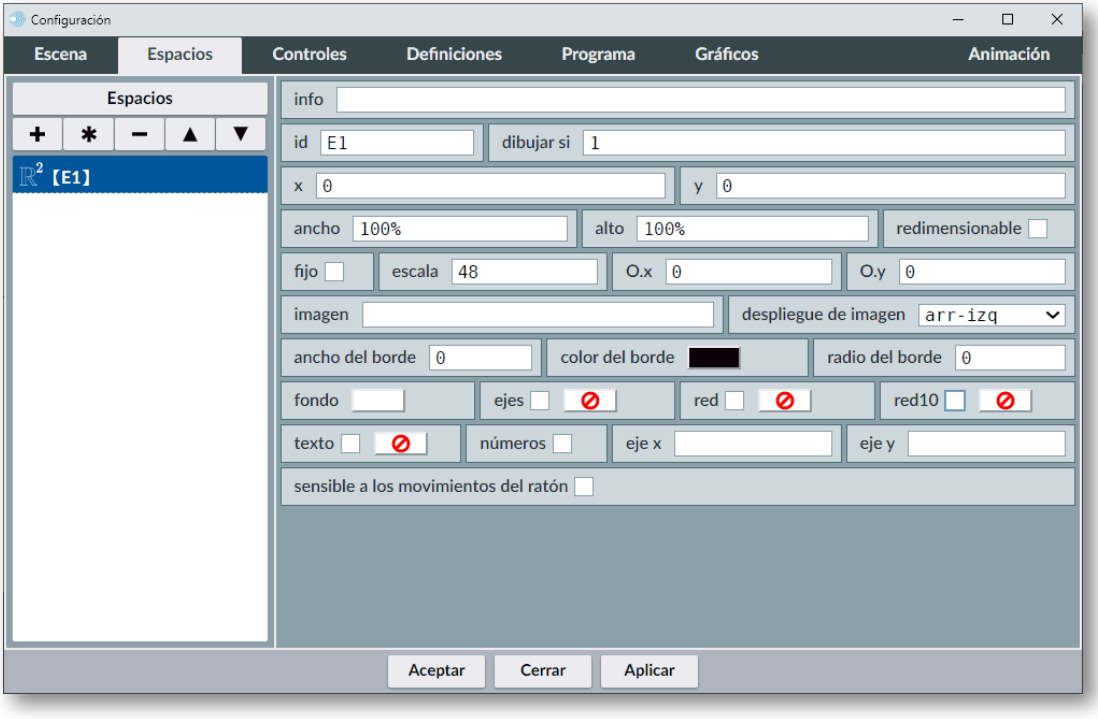
7.5.1 Diseño y realización de las escenas de un material didáctico

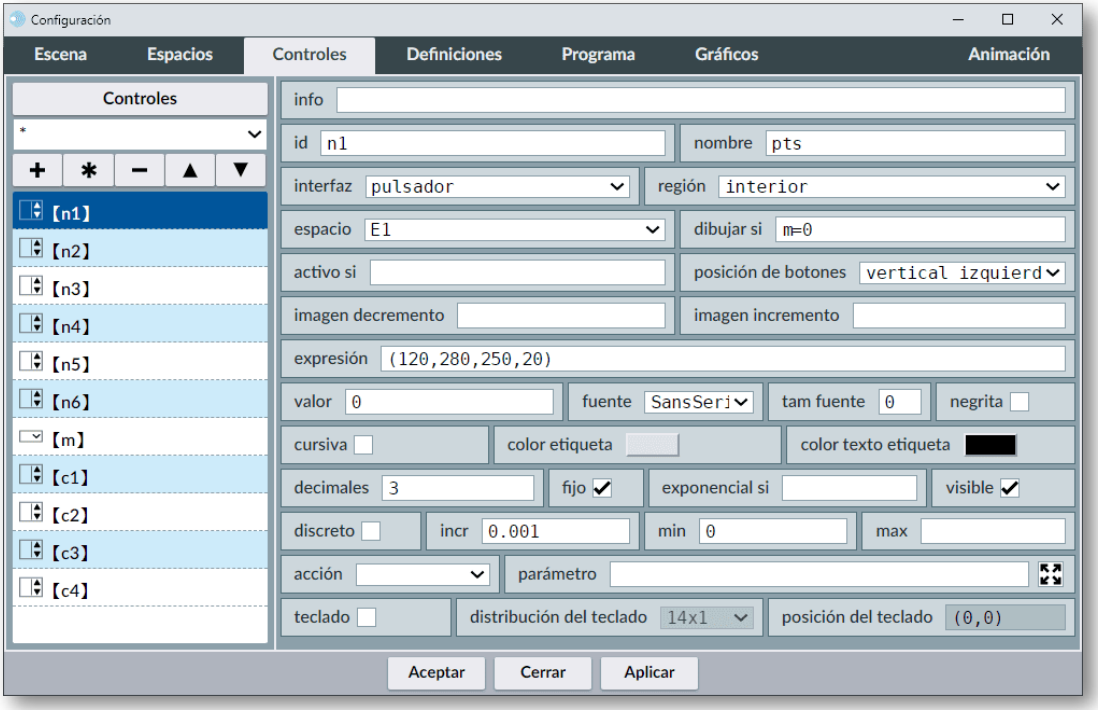
Las escenas de nuestro material didáctico se centrarán en operaciones con factores de conversión. A continuación, se describen los diferentes aspectos considerados para su realización:

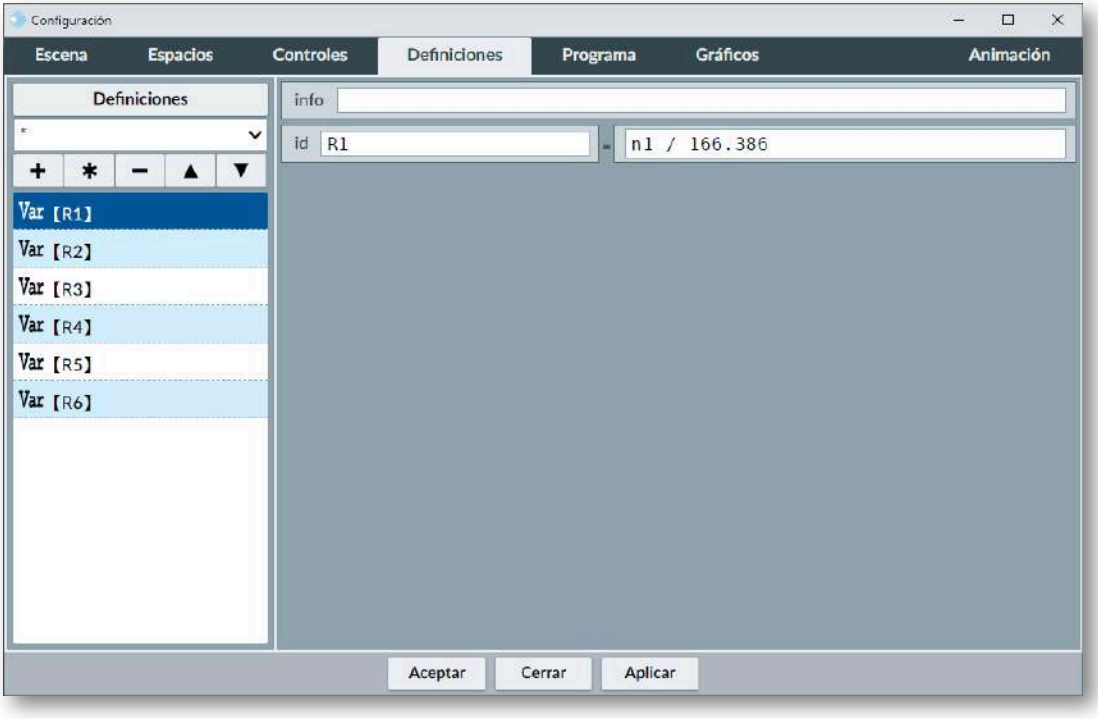
- **Idea o diseño previo:** las escenas se centran en operaciones con factores de conversión, abarcando múltiplos y submúltiplos. La escena permitirá al usuario introducir valores mediante controles en una unidad y mostrar el resultado en otra unidad, mostrando también los pasos intermedios y explicativos de las operaciones para facilitar la comprensión del proceso.
- **Elementos necesarios:** se requiere un espacio 2D donde se dispondrán los elementos gráficos, controles para la introducción de datos (pulsadores, menús y campos de texto), elementos de programación para realizar los cálculos de conversión y textos con fórmulas y explicaciones que guíen al usuario en la operación.

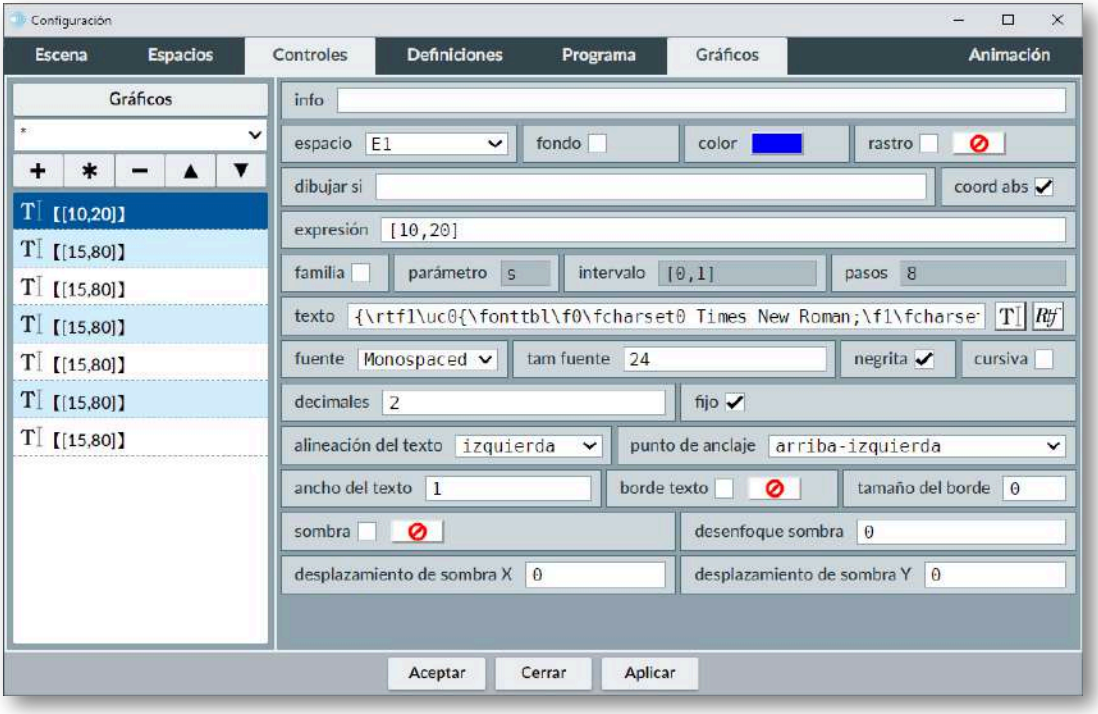
- **Desarrollo temporal o secuencial:** el usuario selecciona la unidad de medida de destino, introduce los datos a convertir y la escena muestra paso a paso las operaciones realizadas, destacando los valores intermedios y el resultado final, de manera clara y comprensible.
- **Elaboración en el editor de escenas:** se introducen el espacio, los controles, los gráficos y los elementos de programación necesarios, definiendo funciones de cálculo, condiciones y la disposición visual de los elementos para garantizar una interacción correcta y didáctica.

A continuación, se incluyen capturas de pantalla que muestran la configuración de los principales elementos empleados en la creación de una escena didáctica interactiva, permitiendo visualizar cómo se han dispuesto los controles, gráficos, textos y elementos de programación para su correcto funcionamiento.




Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Controles
[n1]
[n2]
[n3]
[n4]
[n5]
[n6]
[m]
[c1]
[c2]
[c3]
[c4]
info
id n1
nombre pts
interfaz pulsador
región interior
espacio E1
dibujar si m=0
activo si
posición de botones vertical izquierd
imagen decremento
imagen incremento
expresión (120,280,250,20)
valor 0
fuente SansSeri
tam fuente 0
negrita
cursiva
color etiqueta
color texto etiqueta
decimales 3
fijo
exponencial si
visible
discreto
incr 0.001
min 0
max
acción
parámetro
teclado
distribución del teclado 14x1
posición del teclado (0,0)
Aceptar
Cerrar
Aplicar


Configuración
Escena
Espacios
Controles
Definiciones
Programa
Gráficos
Animación
Definiciones
Var [R1]
Var [R2]
Var [R3]
Var [R4]
Var [R5]
Var [R6]
info
id R1
n1 / 166.386
Aceptar
Cerrar
Aplicar

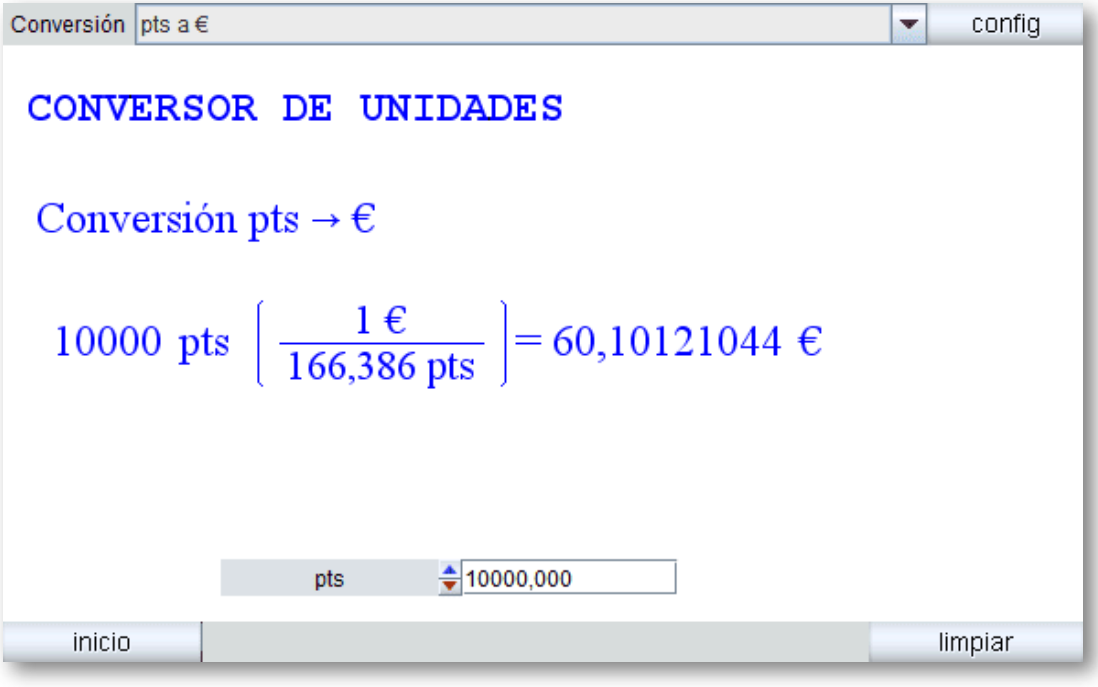


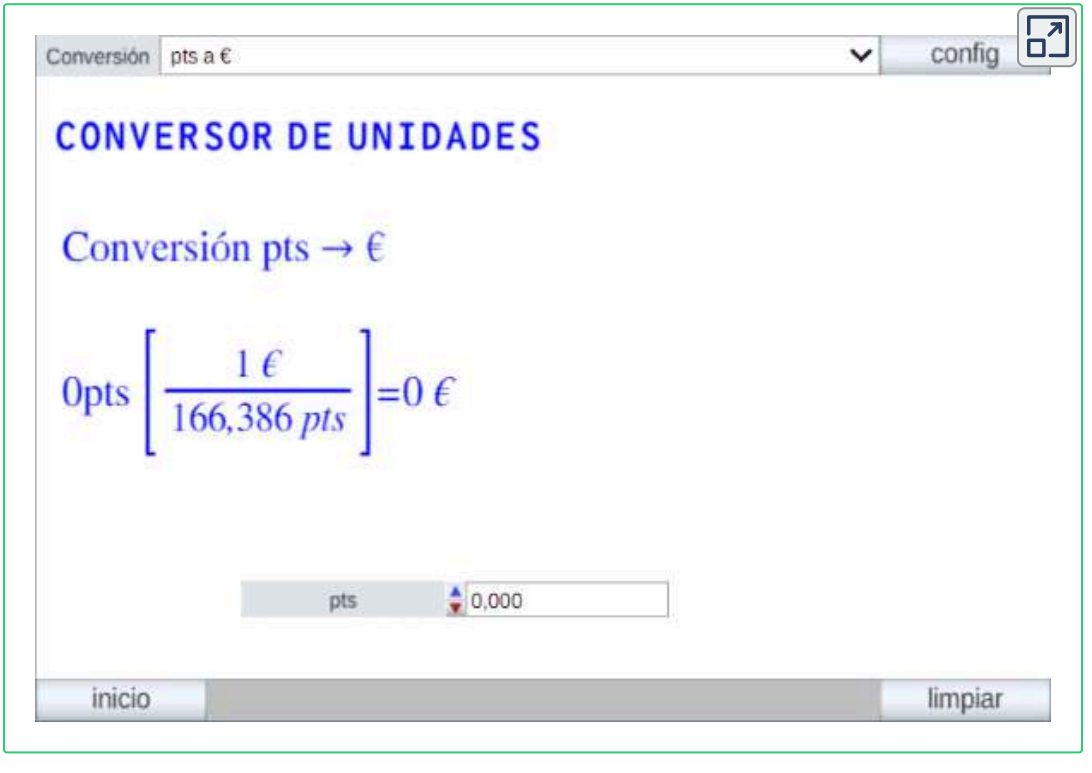
- **Probamos, corregimos y mejoramos** la escena mediante diversas pruebas de funcionamiento, aplicando los cambios y ajustes necesarios para optimizar su operatividad y presentación.

  Es frecuente que surjan errores o comportamientos no deseados durante la elaboración de la escena, por lo que es fundamental realizar varias pruebas hasta alcanzar el funcionamiento previsto.

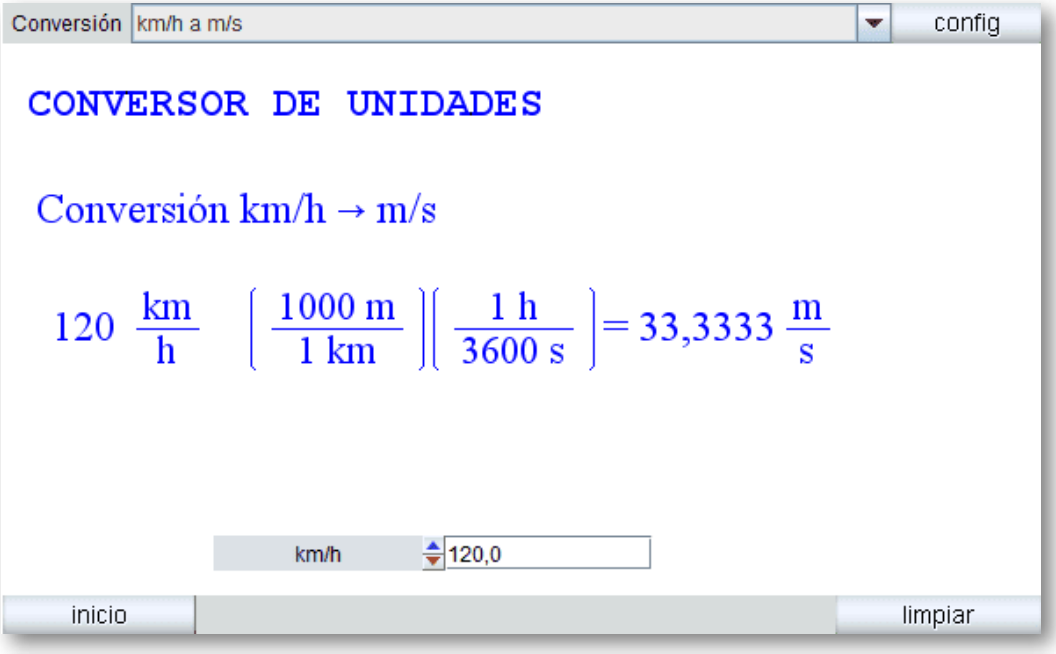
  A continuación, se presenta una captura de pantalla del desarrollo de la escena del conversor de unidades con el fin de ilustrar su funcionamiento.

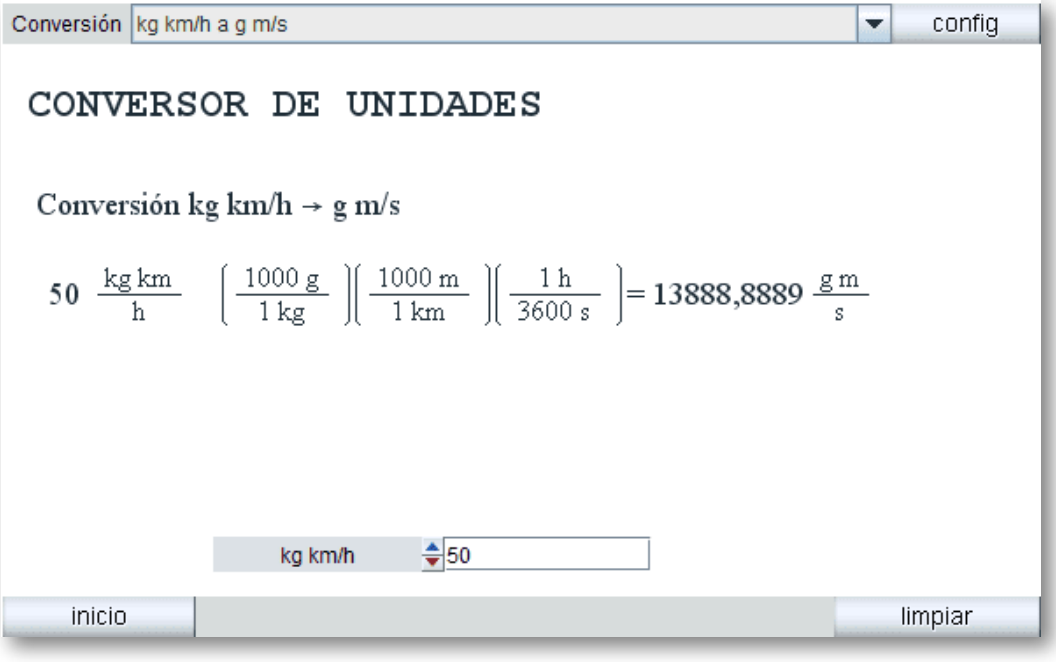
  También se incluye la escena interactiva completa sobre factores de conversión, creada para poder probarla y utilizarla directamente.

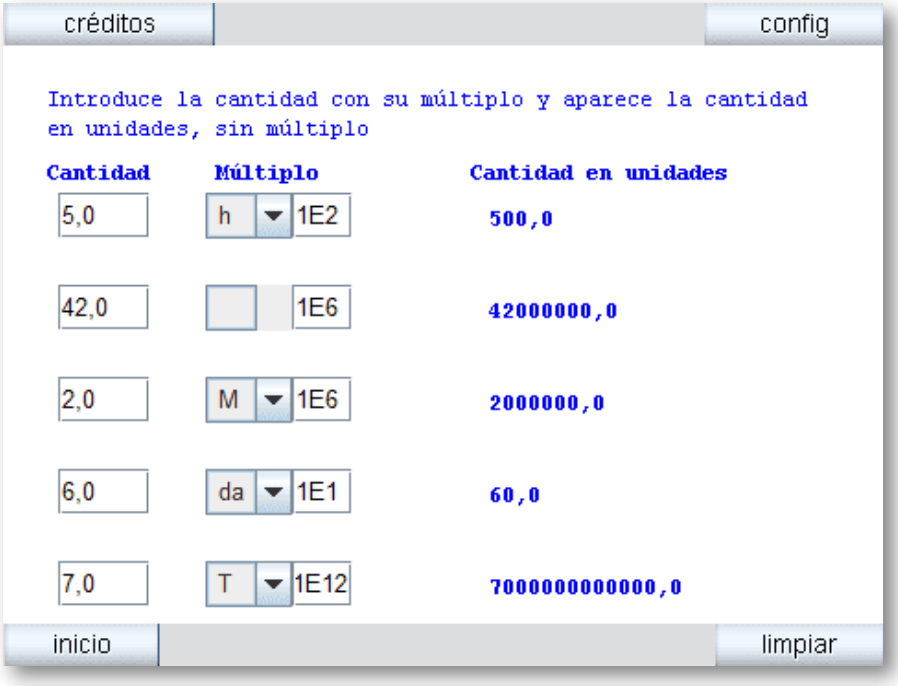

Conversión
pts a €
config
CONVERSOR DE UNIDADES
Conversión pts → €
10000 pts [1 € / 166,386 pts] = 60,10121044 €
pts
10000,000
inicio
limpiar

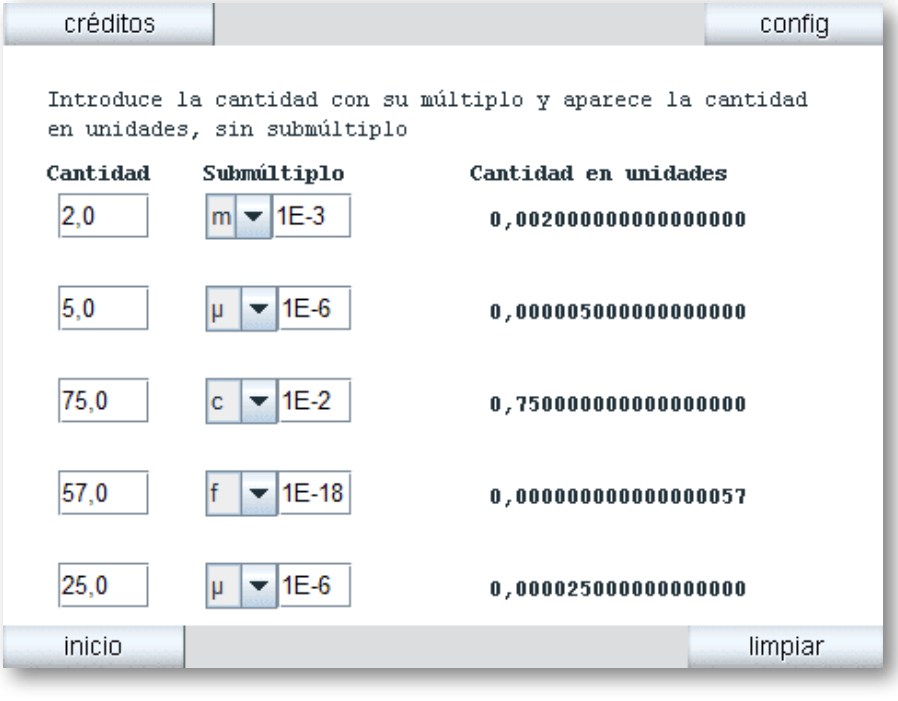

Conversión
pts a €
config
CONVERSOR DE UNIDADES
Conversión pts → €
0pts [1 € / 166,386 pts] = 0 €
pts
0,000
inicio
limpiar

- **Realizamos variantes o modalidades** de la escena para adaptarla a distintos contextos o actividades, las cuales se presentan a continuación.









- Tras la optimización las escenas deberán **insertarse** en el lugar adecuado de la **página web** del material didáctico.

7.5.2 Introducción-presentación

El texto de introducción del material didáctico elaborado podría ser el siguiente:

Un **factor de conversión** es una operación matemática que permite realizar cambios de unidades de la misma magnitud o calcular la equivalencia entre múltiplos y submúltiplos de una determinada unidad de medida.

Dicho de manera más sencilla, un factor de conversión es "una cuenta" que permite expresar una medida de diferentes formas sin alterar su valor. Algunos ejemplos habituales de aplicación son los siguientes:

- **Cambios monetarios:** euros, dólares, pesetas, libras, pesos, escudos...
- **Medidas de distancia:** kilómetros, metros, millas, leguas, yardas...
- **Medidas de tiempo:** horas, minutos, segundos, días, años, siglos...
- **Velocidades:** kilómetros/hora, metros/segundo, nudos, años-luz...

Además, se añadirá un **logo** que aparecerá en todas las páginas del material, tal como se muestra a continuación.

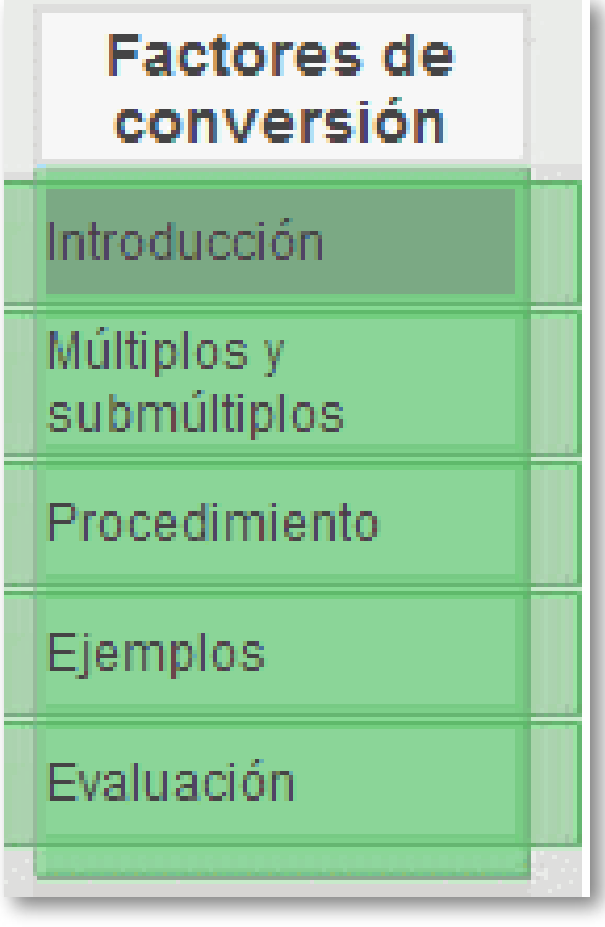


7.5.3 Índice

Hemos elaborado un **índice** que funciona como menú de navegación para la web, facilitando el acceso rápido y ordenado a todas las secciones del material. Su estructura incluye: introducción del tema, múltiplos y submúltiplos, procedimiento para realizar operaciones con factores de conversión, y ejemplos y actividades de evaluación.

Este índice es fundamental para guiar al usuario y asegurar una navegación clara y coherente por todo el contenido.

A continuación, se muestra una imagen del índice.



7.5.4 Contenidos

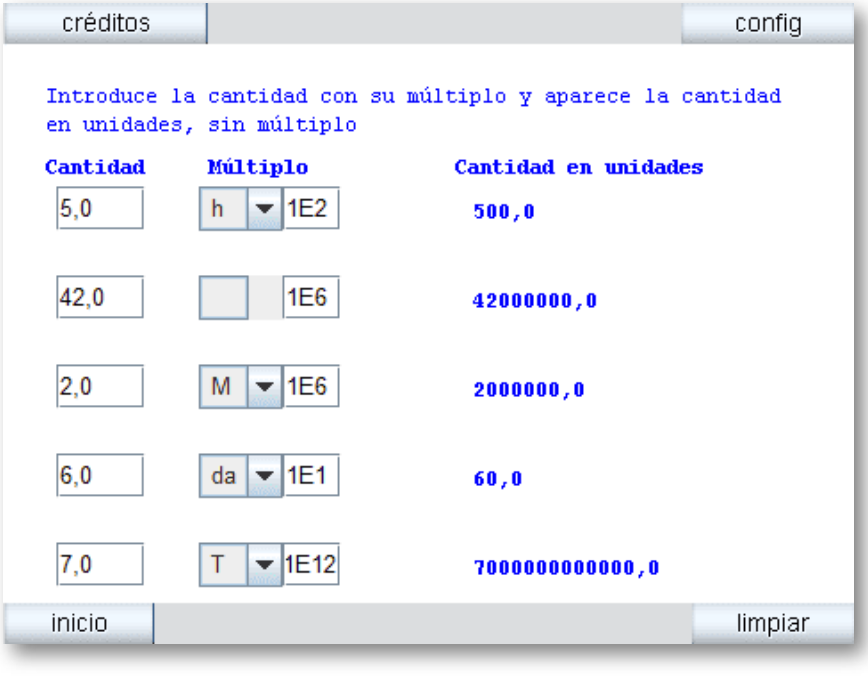
Los contenidos que se abordarán en este material didáctico son los siguientes:

- Definición de factores de conversión.
- Definición de múltiplos y submúltiplos.
- Listado de múltiplos y submúltiplos, junto con sus equivalencias.
- Procedimiento para utilizar los factores de conversión.

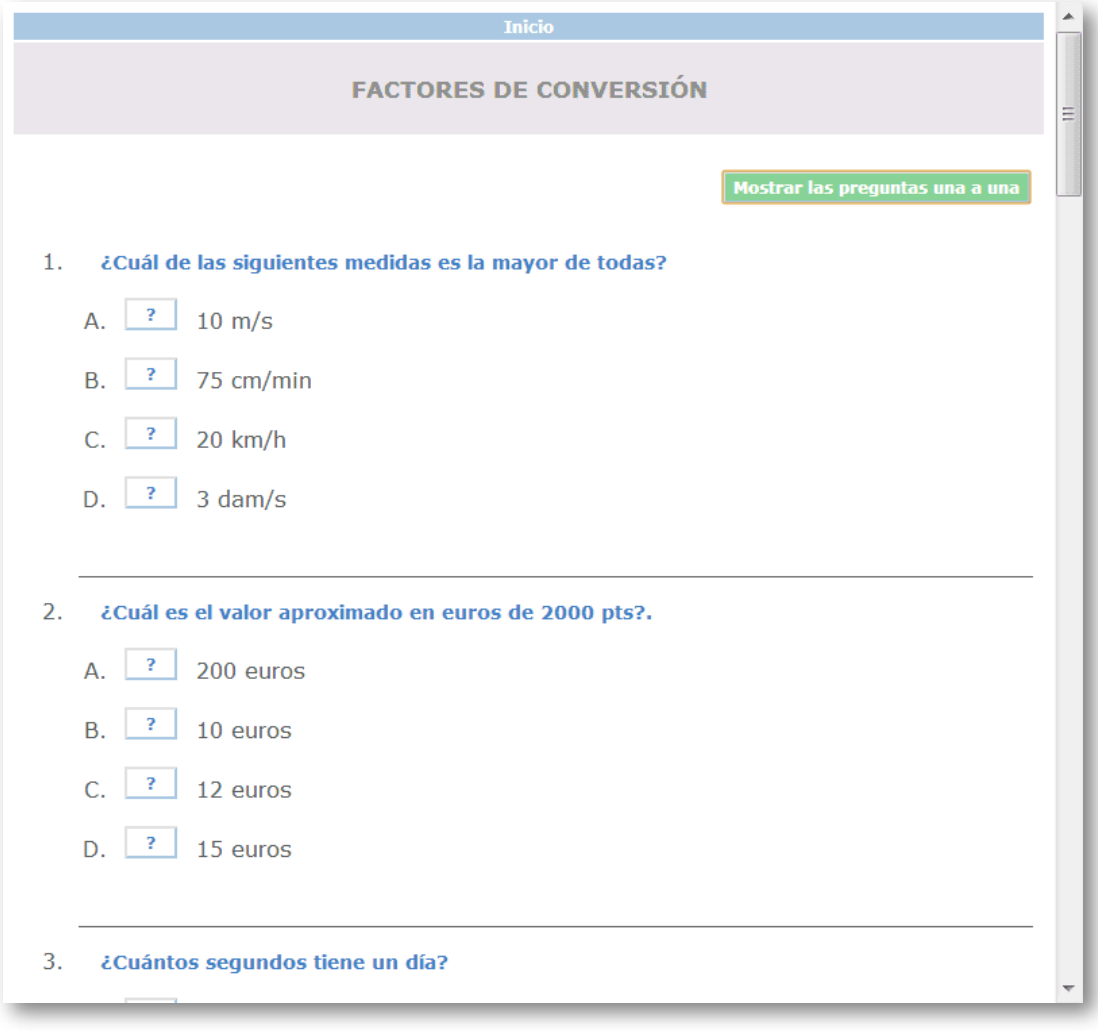
7.5.5 Actividades

Para este material se han incluido actividades de ejemplo y actividades de evaluación, estas últimas creadas con la aplicación **Hot Potatoes**. También podrían haberse elaborado utilizando alguno de los juegos AJDA, como el juego **"Test"** o **"Test con cifras"**.

Ejemplo de actividad realizada con factores de conversión:



Ejemplo de actividad de evaluación creada con **Hot Potatoes** e incluida en el proyecto de material didáctico interactivo:



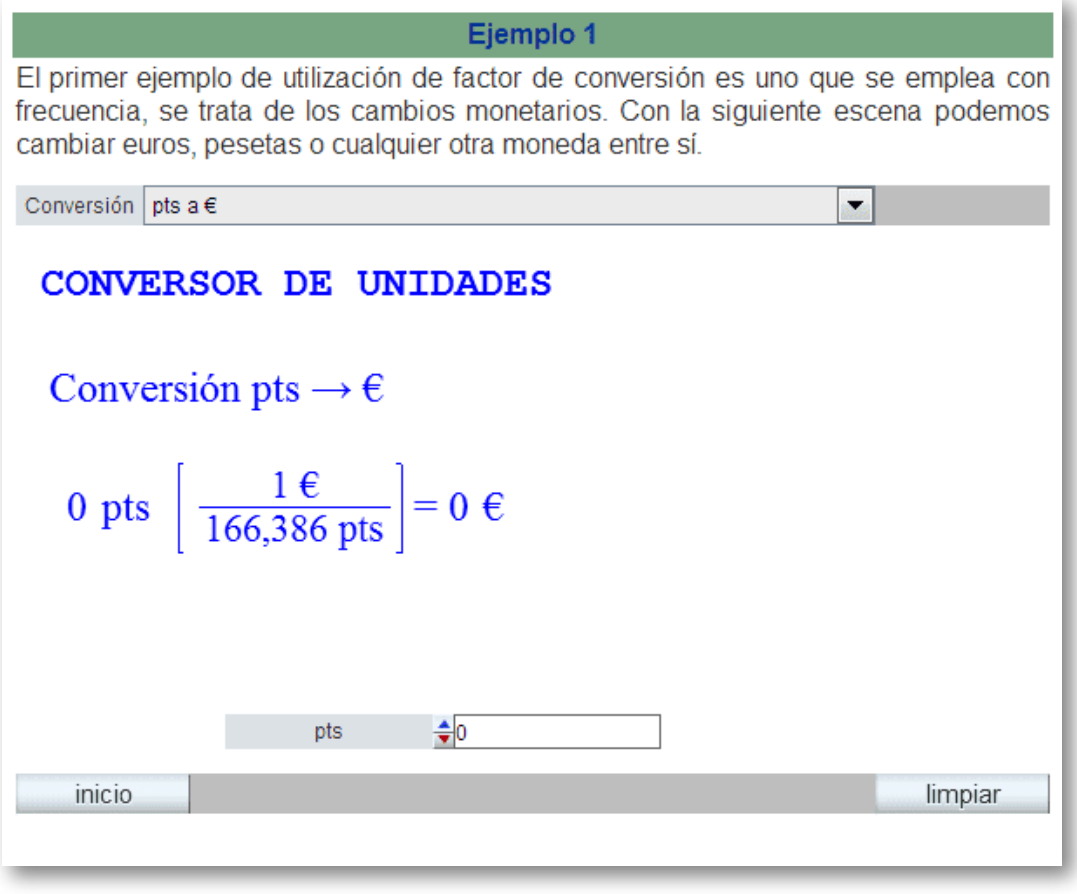
7.5.6 Integración de escenas en la web

Las escenas interactivas del material didáctico deben incorporarse en la página web en el lugar donde se desee que aparezcan. Para ello, se copia el código completo de la escena (que comienza con `<ajs` y finaliza con `</ajs>`) en la sección correspondiente del código HTML de la web.

Es fundamental que la página web que contenga la escena incluya también el archivo `descartes-min.js` o haga referencia a su ubicación en línea. Asimismo, en cada escena se debe especificar la ruta relativa del archivo con respecto a la ubicación de la página web para que funcione correctamente.

Las escenas pueden integrarse directamente en la web, mediante un botón que abra la escena en otra pestaña, o mediante un enlace (URL) que dirija al juego o actividad interactiva.

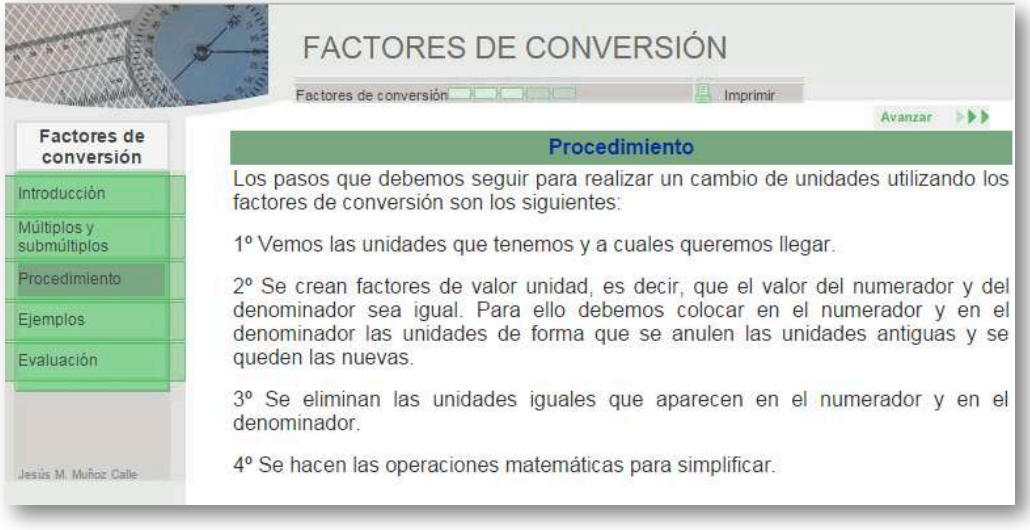
En el ejemplo del proyecto que presentamos, hemos optado por mostrar las escenas directamente dentro de las distintas páginas web, garantizando una experiencia fluida y accesible para el usuario.



7.5.7 Web de factores de conversión completa

A continuación, presentamos la web completa del material sobre factores de conversión, mostrando los distintos apartados y subapartados en un índice vertical izquierdo común. La web mantiene un estilo uniforme y coherente, y sigue un orden lógico que facilita la navegación y el acceso a los contenidos.







FACTORES DE CONVERSIÓN

Factores de conversión | Imprimir

Factores de conversión

- Introducción
- Múltiplos y submúltiplos
  - Múltiplos
  - Submúltiplos
- Procedimiento
- Ejemplos
- Evaluación

Jesús M. Muñoz Calle

Múltiplos

Múltiplos son aquellos prefijos que se colocan delante de la unidad y la multiplican por la unidad seguida de ceros. Los más empleados son:

Múltiplos	Símbolo	Valor que multiplica	Notación científica
Deca	da	x 10	x 10
Hecto	h	x 100	x 10^2
Kilo	k	x 1.000	x 10^3
Mega	M	x 1.000.000	x 10^6
Giga	G	x 1.000.000.000	x 10^9
Tera	T	x 1.000.000.000.000	x 10^{12}

créditos | config

Introduce la cantidad con su múltiplo y aparece la cantidad en unidades, sin múltiplo

Cantidad	Múltiplo	Cantidad en unidades
1	0	0,0
1	0	0,0
1	0	0,0
1	0	0,0
1	0	0,0

inicio | limpiar



FACTORES DE CONVERSIÓN

Factores de conversión | Imprimir

◀◀ Retroceder

Factores de conversión

- Introducción
- Múltiplos y submúltiplos
- Procedimiento
- Ejemplos
- Evaluación

Jesús M. Muñoz Calle

Procedimiento

Como ejemplo para ilustrar el procedimiento, vamos a pasar de 30 cm/s a m/h

1º Hay que cambiar los cm a m en el numerador y los segundos a horas en el denominador.

2º Vamos a multiplicar por factores de valor unidad de forma adecuada, es decir, para que se eliminen los centímetros y los segundos, y se queden los metros y las horas.

$$\frac{30\ \text{m}}{\text{s}}\left(\frac{1\ \text{km}}{1000\ \text{m}}\right)\left(\frac{3600\ \text{s}}{1\ \text{h}}\right)$$

3º Se eliminan las unidades iguales que aparecen en el numerador y en el denominador (mueve el puntero del ratón sobre las fracciones).

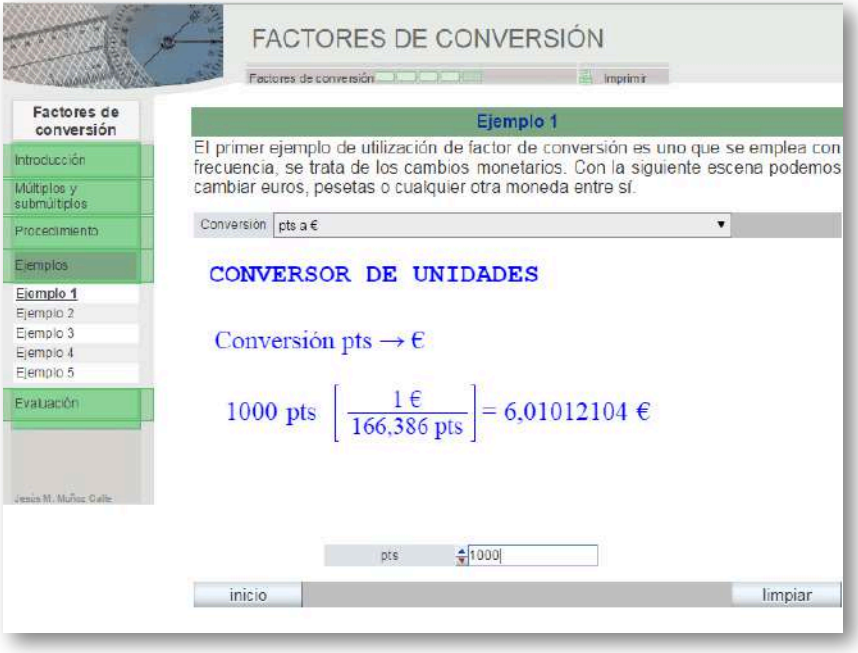
$$\frac{30\ \text{m}}{\text{s}}\left(\frac{1\ \text{km}}{1000\ \text{m}}\right)\left(\frac{3600\ \text{s}}{1\ \text{h}}\right)$$

4º Se hacen las operaciones matemáticas para simplificar (mueve el puntero del ratón sobre las fracciones).

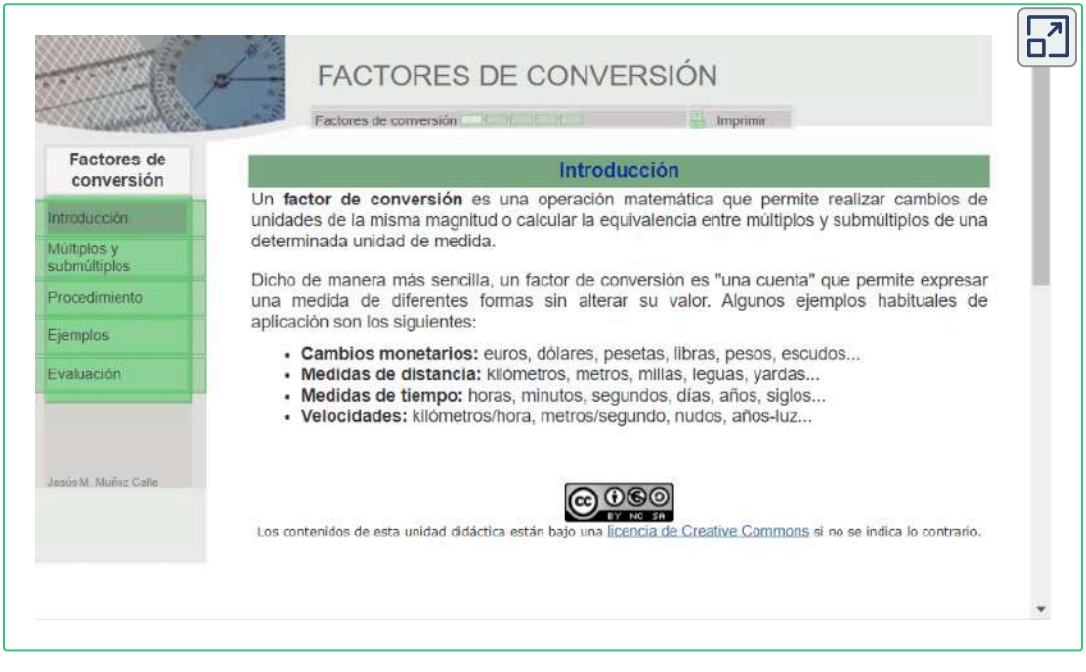
$$\frac{30\ \text{m}}{\text{s}}\left(\frac{1\ \text{km}}{1000\ \text{m}}\right)\left(\frac{3600\ \text{s}}{1\ \text{h}}\right)$$


FACTORES DE CONVERSIÓN
Factores de conversión
Imprimir
Factores de conversión
Introducción
Múltiplos y submúltiplos
Múltiplos
Submúltiplos
Procedimiento
Ejemplos
Evaluación
Jesús M. Muñoz Calle
Submúltiplos
Submúltiplos son aquellos prefijos que se colocan delante de la unidad y la dividen por la unidad seguida de ceros. Los más empleados son:
Submúltiplos
Múltiplos | Símbolo | Valor que multiplica | Notación científica
deci | d | x 0,1 | x 10-1
centi | c | x 0,01 | x 10-2
mili | m | x 0,001 | x 10-3
micro | µ | x 0,000001 | x 10-6
nano | n | x 0,000000001 | x 10-9
pico | p | x 0,000000000001 | x 10-12
créditos
config
Introduce la cantidad con su múltiplo y aparece la cantidad en unidades, sin submúltiplo
Cantidad
Submúltiplo
Cantidad en unidades
1,0
u
1E0
1,0000000000000000
inicio
limpiar


FACTORES DE CONVERSIÓN
Factores de conversión
Imprimir
Factores de conversión
Introducción
Múltiplos y submúltiplos
Procedimiento
Ejemplos
Evaluación
Jesús M. Muñoz Calle
Evaluación
Como actividades de evaluación o autoevaluación, se propone la realización de ejercicios que impliquen la realización de cambios de unidades. Mediante las correspondientes escenas se puede verificar si la resolución de las mismas ha sido la correcta.
Se recomienda realizar primero los ejercicios con papel y lápiz y a continuación utilizar las escenas interactivas.
Cuando se realicen las actividades se deben comprobar los errores cometidos, sus causas y una vez comprendido esto volver a realizar de nuevo las actividades.
Ejercicios



A continuación, se presenta la página web completa del proyecto del material didáctico **"Factores de conversión"**, incluyendo todas sus secciones y apartados.



7.6 Publicación de escenas

Las escenas de **Descartes** pueden publicarse en páginas web, blogs, wikis, plataformas educativas como Moodle o incluso en dispositivos móviles. Para insertar una escena en cualquier sitio web, se deben seguir estas indicaciones:

- Copiar el código de la escena (Ctrl+C) comprendido entre las etiquetas `<ajs name=...>` y `</ajs>`

  y pegarlo (Ctrl+V) en el lugar del código HTML donde se desee mostrar la escena.

- Incluir la referencia al archivo de la librería `descartes-min.js` mediante la etiqueta:

```
<script type='text/javascript'
src='http://arquimedes.matem.unam.mx/Descartes5/lib
/descartes-min.js'></script>
```

  o con la ruta relativa si se dispone localmente:

```
<script type='text/javascript'
src='ruta_relativa/descartes-min.js'></script>
```

- Si la escena utiliza imágenes, macros u otros ficheros, es necesario incluirlos junto a la escena y haber indicado su ruta relativa correctamente en la escena.

- Para embeber la escena en una ventana, se puede usar:

```
<iframe src="dirección_donde_esta_la_escena"
width="700" height="900"></iframe>
```

  Donde `src` es la dirección de la escena (debe comenzar por `http://` o `https://`) y `width` y `height` definen el tamaño del marco. Se pueden añadir otros parámetros dentro del `iframe`.

- Otra opción es enlazar la escena existente en Internet mediante un hiperenlace, indicando en el enlace la dirección de la escena.

7.7 Recomendaciones para la puesta en práctica

A continuación, se presentan algunas **recomendaciones** a tener en cuenta para la puesta en práctica de **actividades** basadas en **juegos didácticos** o materiales con **escenas interactivas** de Descartes. Aunque muchas resulten evidentes, no seguirlas puede impedir que la **actividad** se desarrolle correctamente o funcione de manera adecuada:

- Elegir **juegos**, **materiales** y **contenidos** adecuados al **grupo de alumnos** al que van dirigidos.
- Comprobar que los **documentos** y **recursos** funcionen correctamente en el **equipo** que se utilizará con el **alumnado**.
- Desarrollar **materiales** de distintos **niveles** para adaptarse a la **diversidad del aula**.
- Asegurarse de que las **actividades** contribuyan al **desarrollo de competencias**, la **motivación** del **alumnado** y la **convivencia** en el aula.
- Coordinar con el **alumnado** si va a participar en la **preparación** y **organización** de la actividad.
- Reservar el **aula** con antelación si se va a usar un **espacio de uso común**.
- Verificar que los **juegos** o **escenas** funcionan correctamente en el **ordenador** desde el que se van a usar.
- Contar con el **aula** y los **recursos necesarios**: **ordenador**, **proyector**, **pantalla**, **sonido**, **altavoces**; también pueden utilizarse **pizarras digitales interactivas** según la **dinámica** de la actividad.
- Tener el **aula o lugar donde se vaya a poner en práctica** listo antes de empezar: **ordenador encendido**, **proyector** o **pizarra digital** conectados, **altavoces** funcionando y **juego/material interactivo** preparado y cargado.

- Ensayar previamente para garantizar que todo funcione correctamente.
- Diseñar y comunicar claramente la **mecánica** de la **actividad** a los **participantes**.
- Si se desarrolla en varias **sesiones**, planificar un **calendario** con fechas, **contenidos** y desarrollo previsto.
- Preparar el **material complementario**: **fichas**, **tarjetas**, **documentos de registro y evaluación**, **carteles**, **premios**, etc.
- Tener listos otros **recursos informáticos** necesarios.
- Utilizar **bibliografía**, **publicaciones**, **tutoriales** y **vídeos** que apoyen la **planificación** de la **actividad**.
- Organizar la **disposición del aula**, **personal**, **decorados**, **iluminación**, **distintivos** de jugadores y equipos según sea necesario.
- Explicar desde el principio las **reglas** y **dinámica** de la **actividad** de manera clara, asegurando la implicación del **alumnado**.
- Garantizar una correcta **disposición** y **participación** de todos los **alumnos**.
- Repartir **materiales** y **documentos** necesarios, indicando su uso.
- Mantener un **ritmo** y **tempo** adecuado durante la **actividad**.
- Valorar y reconocer positivamente la **participación** de los **alumnos**.
- Registrar los aspectos **relevantes** que sirvan para la **evaluación**.
- Si la **actividad** se desarrolla en varias **sesiones**, procurar que comiencen y terminen de forma ordenada para mantener la **continuidad**.
- Si la **escena** o el **juego** se bloquea, recargar o actualizar la **página**; en ese caso, la **escena** se ejecutará desde el **inicio**.

7.8 Ideas y ejemplos sobre proyectos con escenas de Descartes

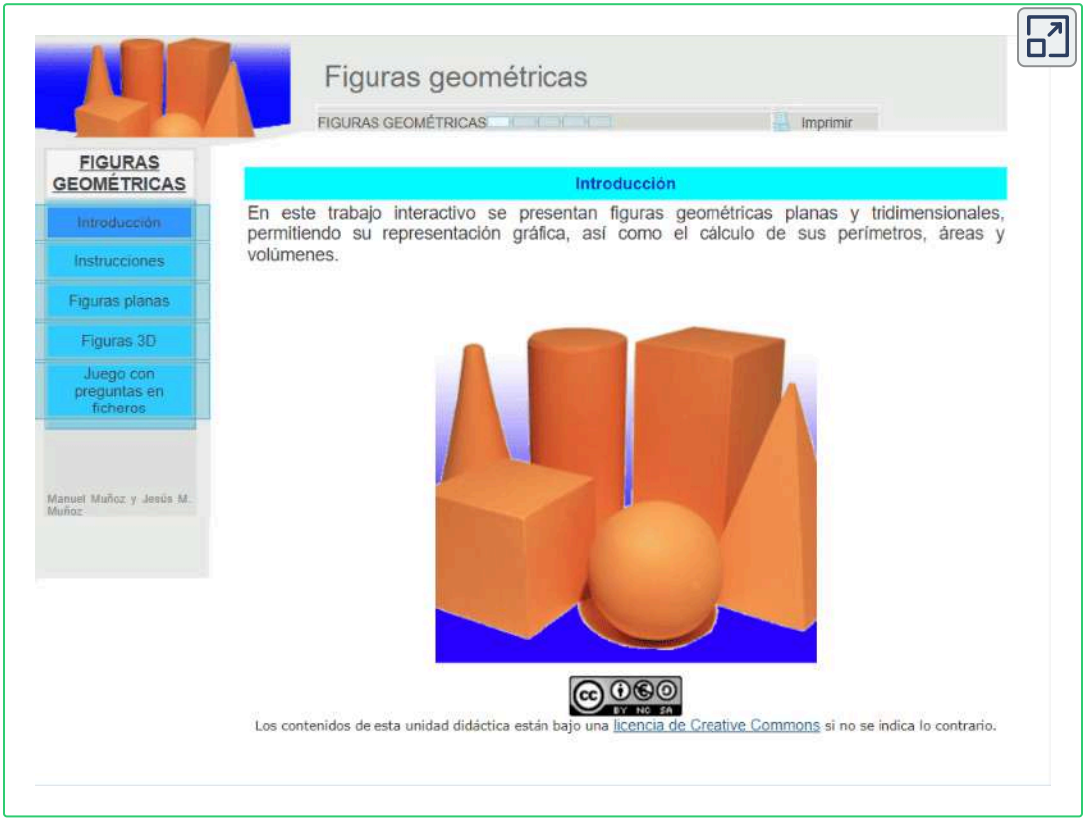
Como ya hemos comentado, una opción es realizar proyectos basados en **juegos didácticos**. En la web del **"Proyecto AJDA"** existe una amplia variedad de **juegos y recursos educativos** que pueden servir de referencia. La idea no es crear un juego complejo o difícil, sino **inspirarse** en alguno de los ya existentes para elaborar uno que resulte interesante y esté **adaptado a nuestras posibilidades**. Se pueden plantear, por ejemplo, **juegos de preguntas y respuestas**, de **adivinar palabras o frases**, **juegos con números**, de **asociación de imágenes**, con **paneles ocultos**, relacionados con algún **deporte**, **juegos clásicos**, entre otros.

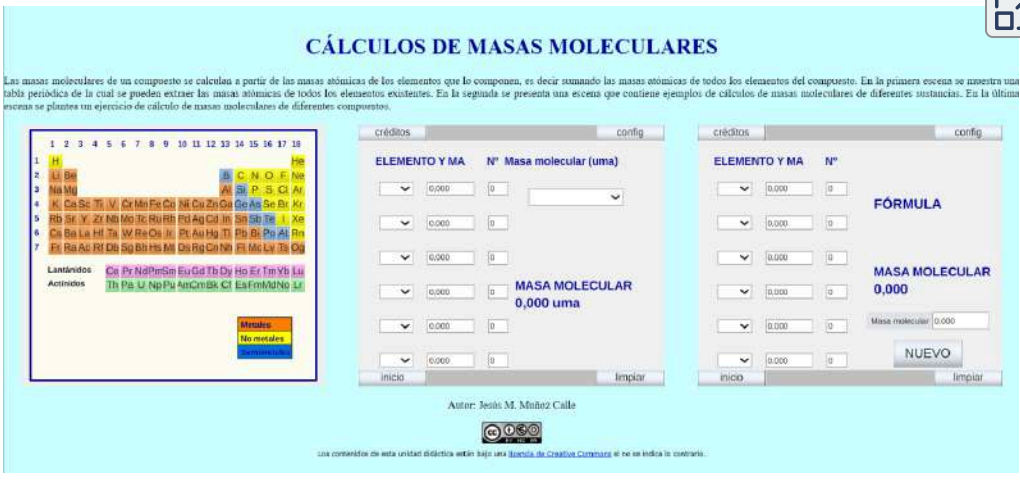
Otra posibilidad consiste en elaborar **una o varias escenas interactivas** sobre contenidos de una determinada materia. Algunos ejemplos podrían ser:

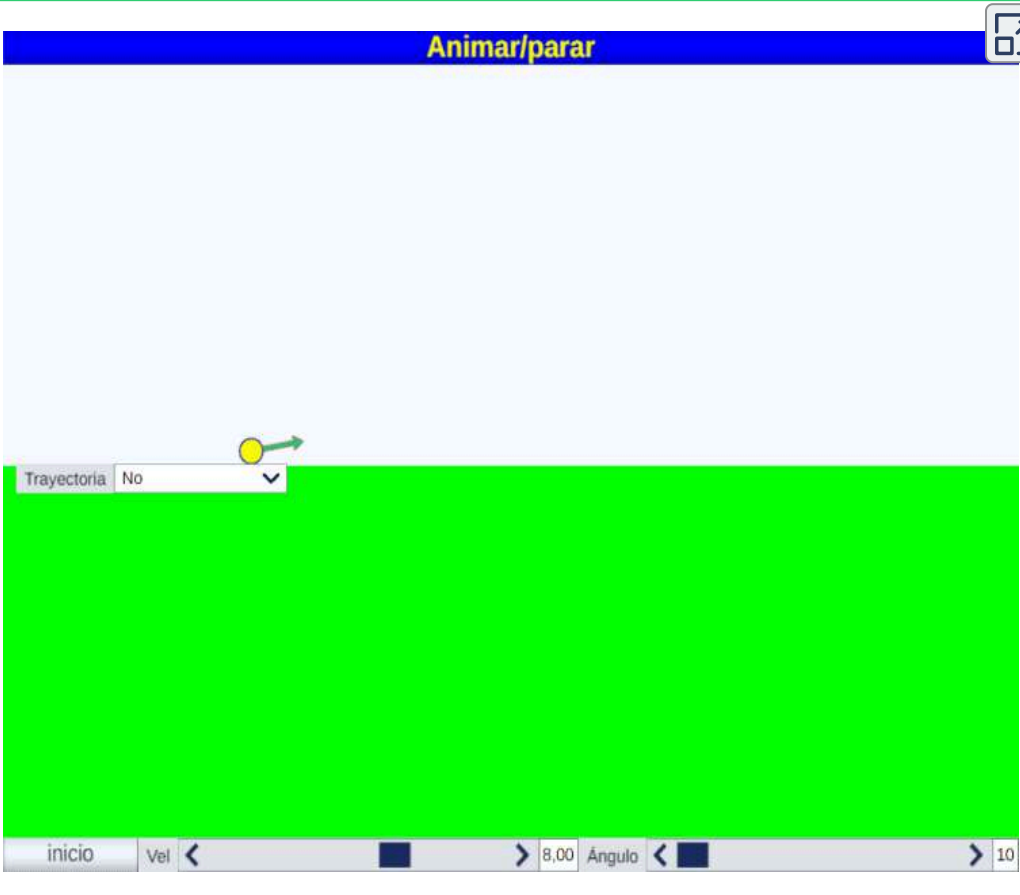
- **Matemáticas:** dibujo de **funciones**, **cuerpos geométricos**, **operaciones aritméticas**, cálculos **estadísticos** o de **probabilidad**, etc.
- **Lengua:** actividades de **completar poemas o frases**, **conjugación de tiempos verbales**, **división silábica**, etc.
- **Idiomas:** actividades de **traducción de palabras o frases**, **asociación de imágenes con vocabulario**, aprendizaje de **tiempos verbales**, etc.
- **Ciencias y tecnología: simulación de fenómenos** naturales o experimentales, **resolución de problemas**, elaboración de **modelos científicos** o **simulación de máquinas**.
- **Educación física:** simulación de **actividades físicas**, estudio de las **dimensiones de canchas y campos deportivos** o de las **reglas de distintos deportes**.

- **Humanidades:** actividades con **mapas**, relacionadas con **fechas y hechos históricos**, o análisis **demográfico**.
- **Música:** actividades con **notas y pentagramas**, **reconocimiento de instrumentos**, asociación de **canciones o melodías**, etc.
- **Infantil y primaria:** actividades con **colores**, **animales**, **operaciones sencillas** o **asociación de nombres y objetos**.
- **Formación profesional:** cualquier actividad relacionada con los **contenidos del módulo** correspondiente.

Todo lo anterior es solo una muestra de las muchas posibilidades existentes. En cualquier caso, será el **docente** quien mejor pueda seleccionar y diseñar un **proyecto** que se adapte a las **necesidades de su alumnado** y a su **práctica educativa**. A continuación, presentamos algunos ejemplos de **escenas** y **proyectos**.




CÁLCULOS DE MASAS MOLECULARES
Las masas moleculares de un compuesto se calculan a partir de las masas atómicas de los elementos que lo componen, es decir sumando las masas atómicas de todos los elementos del compuesto. En la primera escena se muestra una tabla periódica de la cual se pueden extraer las masas atómicas de todos los elementos existentes. En la segunda se presenta una escena que contiene ejemplos de cálculos de masas moleculares de diferentes sustancias. En la última escena se plantea un ejercicio de cálculo de masas moleculares de diferentes compuestos.
créditos
config
ELEMENTO Y MA
Nº
Masa molecular (uma)
MASA MOLECULAR
0,000 uma
FÓRMULA
MASA MOLECULAR
0,000
Masa molecular
NUEVO
inicio
limpiar
Autor: Jesús M. Muñoz Calle


Animar/parar
Trayectoria
No
inicio
Vel
8,00
Ángulo
10

Referencias

[1] Proyecto AJDA. Aplicación de juegos didácticos en el aula. *https://proyectodescartes.org/newton/juegosdidacticos/index.php*. (s. f.).

[2] Proyecto Descartes. RED EDUCATIVA DIGITAL DESCARTES. *https://proyectodescartes.org/descartescms/*. (s. f.).

[3] Jesús M. Muñoz Calle. "Juegos didácticos. Proyecto AJDA". *iCartesiLibri*. 2024.

[4] Jesús M. Muñoz Calle. "Colección de juegos didácticos AJDA ". *iCartesiLibri*. 2026.

[5] Jesús M. Muñoz Calle. "Gamificando con juegos AJDA". *iCartesiLibri*. 2024.

[6] Jesús M. Muñoz Calle. "El Gran Torneo de Magia y Hechicería". *iCartesiLibri*. 2026.

[7] Jesús M. Muñoz Calle. "Los Juegos de la Alquimia". *iCartesiLibri*. 2026.

[8] Juan Guillermo Rivera Berrío. "Inteligencias Artificiales Generativas 2024". *iCartesiLibri*. 2024.

[9] Juan Guillermo Rivera Berrío y Jesús M. Muñoz Calle. "Diseño de contenidos digitales con inteligencia artificial". *iCartesiLibri*. 2024.

[10] Juan Guillermo Rivera Berrío y Jesús M. Muñoz Calle. "Plantillas para libros con inteligencia artificial". *iCartesiLibri*. 2024.

[11] Herramienta DescartesJS. https://descartes.matem.unam.mx/. (s.f.).

[12] Alejandro Radillo Díaz, José Luis Abreu León y Joel Espinosa Longi. Documentación técnica de Descartes. (s.f.).

[13] Editorial iCartesiLibri (Red Educativa Digital Descartes). https://proyectodescartes.org/iCartesiLibri/index.htm. (s.f.).

[14] Juan Guillermo Rivera Berrío, Joel Espinosa Longi y Alejandro Radillo Díaz. "DescartesJS - Nivel I". *iCartesiLibri*. 2019.

[15] Juan Guillermo Rivera Berrío, Joel Espinosa Longi y Alejandro Radillo Díaz. "DescartesJS - Nivel II". *iCartesiLibri*. 2019.

[16] Juan Guillermo Rivera Berrío. "Plantillas DescartesJS". *iCartesiLibri*. 2022.

[17] Juan Guillermo Rivera Berrío. "Plantillas para libros". *iCartesiLibri*. 2023.

[18] Juan Guillermo Rivera Berrío y Joel Espinosa Longi. "Diseño de libros interactivos" (Segunda edición). *iCartesiLibri*. 2022.

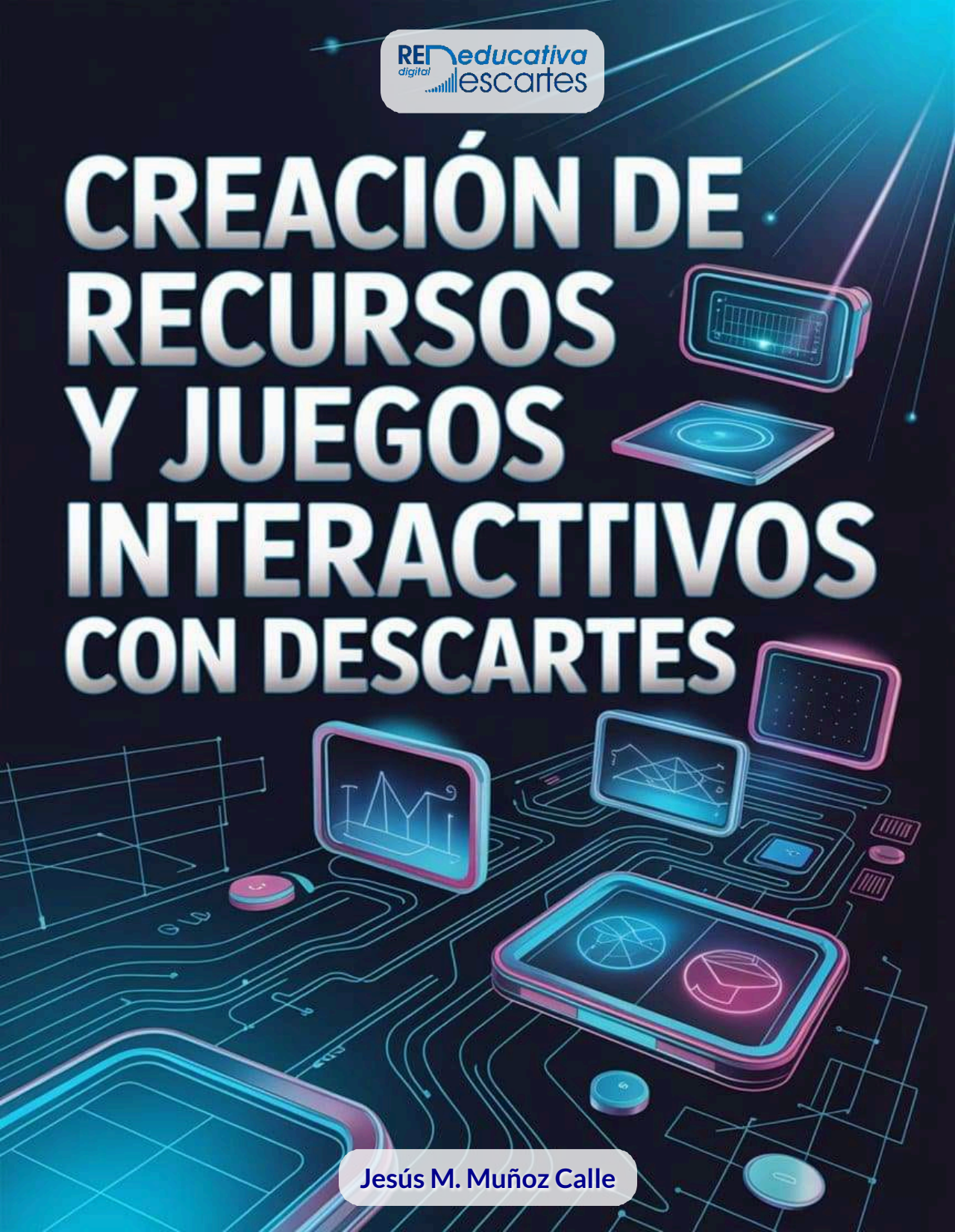
[19] Carlos Alberto Rojas Hincapié. "CURSO: Diseño de objetos interactivos con DescartesJS e IA". *iCartesiLibri*. 2025.

[20] Juan Guillermo Rivera Berrío y Joel Espinosa Longi. "Comunicación DescartesJS - GeoGebra" (Segunda edición). *iCartesiLibri*. 2022.

[21] Proyecto Prometeo (UNAM). https://www.proyectodescartes.org/Prometeo/index.htm. (s.f.).

[22] Subproyectos de la Red Educativa Digital Descartes. https://proyectodescartes.org/indexweb.php. (s.f.).

iCartesiLibri


RED educativa digital descartes
CREACIÓN DE RECURSOS Y JUEGOS INTERACTIVOS CON DESCARTES
Jesús M. Muñoz Calle