

PLAN DE CLASE

Programación de un juego sencillo en Scratch

En esta actividad, los estudiantes usarán Scratch para crear un juego simple, como un laberinto o un personaje que atrapa objetos.

**45-60
minutos**

Duración

**Edad recomendada
para este juego**

**10-12
años**

**Objetivos de
aprendizaje**

<CODE/>



- Comprender los conceptos básicos de la lógica de programación, incluidos bucles, secuencias y condicionales.
- Desarrollar habilidades de resolución de problemas y pensamiento crítico a través de la codificación.
- Crear y probar un prototipo de juego funcional.
- Fomentar la creatividad y la colaboración en un entorno digital.

Materiales y herramientas necesarios

- Ordenadores o tabletas con acceso a Internet.
- Cuentas de Scratch para estudiantes (<https://scratch.mit.edu>).
- Proyector para demostraciones del profesor.
- Auriculares (opcionales)



Orientación para profesores

Descripción de la actividad

Los estudiantes crearán un juego simple de Scratch, como un laberinto o un juego de atrapar, usando bloques de codificación de arrastrar y soltar.

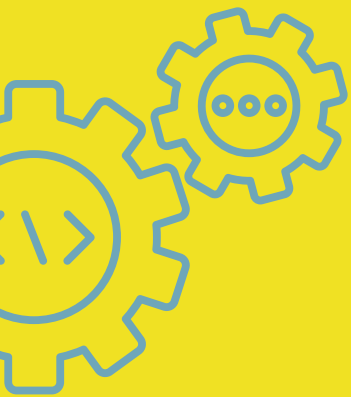
Diseñarán los personajes, establecerán las reglas del juego y probarán su funcionalidad.

Orientación para profesores



Preparación

- Familiarícese con la interfaz de Scratch y cree un ejemplo sencillo para demostrarlo en clase (ver [referencias](#)).
- Prepare una guía paso a paso o una presentación de diapositivas para presentar Scratch y los objetivos del proyecto.
- Divida a los estudiantes en grupos pequeños si van a trabajar en colaboración.
- Pruebe todos los dispositivos para garantizar la compatibilidad con Scratch.



Pasos de implementación

- **INTRODUCCIÓN:** Explique qué es Scratch y muestre una demostración rápida de un juego simple. Resalte también los bloques de codificación clave, como el movimiento, los bucles y las condiciones.
- **PLANIFICACIÓN:** Guíe a los estudiantes para que hagan una lluvia de ideas sobre el juego y esbocen sus diseños en papel. Anímelos a definir los objetivos del juego, como las condiciones para ganar o los obstáculos.
- **CODIFICACIÓN:** Ayude a los estudiantes a configurar sus proyectos de Scratch y a comenzar a codificar. Circule para ayudar a resolver problemas y alentar la experimentación con bloques.

Orientación para profesores

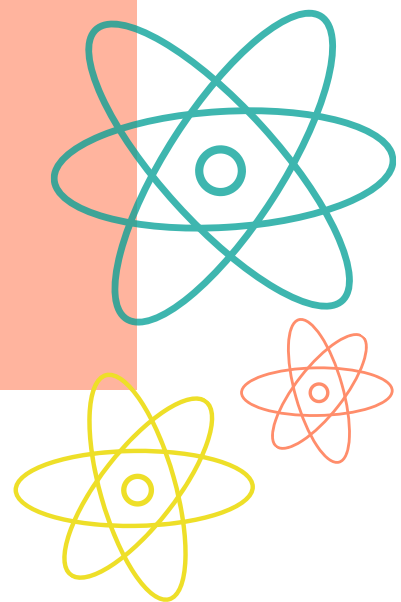
- PRUEBAS: Pida a los estudiantes que prueben sus juegos, identifiquen errores y los mejoren. Fomente la retroalimentación de sus compañeros para perfeccionar sus proyectos.
- EXHIBICIÓN: Permita que los estudiantes presenten sus juegos a la clase y expliquen su proceso de codificación.

Seguimiento y reflexión

Organice una "feria de juegos" donde los estudiantes jueguen los juegos de los demás y proporcionen retroalimentación.

Asígnales una breve tarea de reflexión:

- **¿Qué os gustó más?**
- **¿Qué fue lo más desafiante?**



Actividades estudiantiles

Descripción de la actividad	Resultado esperado	Integración de tecnología
Diseñar un plan de juego	Los estudiantes definirán el objetivo, las reglas y los personajes de su juego.	Utilice las opciones "fondo" y "sprite" de Scratch para visualizar diseños.
Movimientos del juego de código	Los estudiantes programarán sprites para que se muevan o interactúen entre sí.	Utilice bloques de "movimiento" y "evento" en Scratch.
Añadir lógica de juego	Los estudiantes usarán declaraciones condicionales para definir resultados.	Utilice bloques "si-entonces" y "transmisión" para la implementación de la lógica.
Prueba y depuración	Los estudiantes identificarán y corregirán errores de codificación.	Depure utilizando la vista previa del escenario y el resaltado de bloques de Scratch.
Compartir juego	Los estudiantes compartirán proyectos y discutirán sus opciones de codificación.	Utilice la función para compartir de Scratch o el proyector de clase

Preguntas reflexivas para estudiantes

- ¿A qué desafíos te enfrentaste al codificar tu juego y cómo los superaste?
- ¿Qué bloques de codificación fueron los más útiles para crear tu juego?
- Si pudieras añadir una nueva característica, ¿cuál sería y por qué?
- ¿Cómo utilizaste los comentarios de los demás para mejorar tu juego?
- ¿Qué aprendiste sobre trabajo en equipo o resolución de problemas a través de esta actividad?



Ideas de diferenciación

Estudiantes avanzados

Anímalos a agregar complejidad, como por ejemplo:

- Múltiples niveles o temporizadores.
- Lógica avanzada como sistemas de puntuación.
- Animaciones o efectos de sonido adicionales.

Pídeles también que ayuden a sus compañeros que necesiten ayuda, reforzando sus conocimientos.

Estudiantes con necesidades especiales

- Agrúpalos con un compañero para obtener apoyo adicional.
- Divide la tarea en pasos más pequeños y ofrezca guías visuales.
- Celebra su progreso en lugar de la perfección.

Consejos

- Mantenga las instrucciones simples y utilice elementos visuales para explicar los bloques de codificación.
- Fomente la creatividad permitiendo que los estudiantes personalicen sus juegos.
- Asigne tiempo adicional para la resolución de problemas y la depuración.
- Fomentar un entorno de apoyo donde los estudiantes se sientan cómodos pidiendo ayuda.
- Utilice sesiones de retroalimentación entre pares para fomentar la colaboración y el intercambio de ideas.



Materialles y referencias adicionales

[Sitio web de Scratch](#)

[Guía para empezar a usar Scratch](#)

[Tutoriales de codificación interactivos](#)

[Videotutorial sobre cómo utilizar Scratch para niños](#)



nekser



CC BY-SA 4.0 DEED



Co-funded by
the European Union

Cofinanciado por la Unión Europea. No obstante, las opiniones y puntos de vista expresados son exclusivamente los del autor o autores y no reflejan necesariamente los de la Unión Europea o la Agencia Nacional. Ni la Unión Europea ni la Agencia Nacional pueden ser consideradas responsables de las mismas.