

Explora y Crea con Scratch: Programando Historias Interactivas

Tecnología e Informática | Informática | Aprendizaje Basado en Proyectos

Descripción

Este plan de clase está diseñado para que estudiantes de secundaria (12-15 años) aprendan a utilizar Scratch, una plataforma visual de programación, para crear proyectos interactivos que cuenten historias o resuelvan problemas reales. Los estudiantes desarrollarán habilidades de pensamiento lógico, creatividad y trabajo colaborativo al diseñar, programar y compartir sus propios proyectos digitales. Aprenderán conceptos básicos de programación como secuencias, eventos, condicionales y bucles, al mismo tiempo que desarrollan competencias en comunicación y solución de problemas.

El uso de Scratch permite a los estudiantes conectar la tecnología con su vida cotidiana, pues podrán crear animaciones, juegos o narrativas que reflejen sus intereses y experiencias. La metodología de Aprendizaje Basado en Proyectos fomenta la autonomía y colaboración, apropiándose del conocimiento mediante la práctica y la reflexión sobre su propio proceso creativo.

Este plan contribuye a desarrollar competencias digitales clave para el siglo XXI, promoviendo un aprendizaje activo, significativo y contextualizado, que empodera a los estudiantes para expresarse y resolver desafíos con herramientas tecnológicas.

Objetivos de Aprendizaje

- Analizar los conceptos básicos de programación visual en Scratch, incluyendo bloques de código, eventos y control.
- Diseñar y crear un proyecto interactivo utilizando Scratch que cuente una historia o resuelva un problema real.
- Colaborar efectivamente en equipos para planificar, programar y mejorar proyectos digitales.
- Evaluar y mejorar el propio proyecto y el de sus compañeros mediante retroalimentación constructiva.
- Reflexionar sobre el proceso de aprendizaje y las aplicaciones prácticas de la programación en su entorno cotidiano.

Recursos Necesarios

- Computadoras o laptops con acceso a internet (1 por estudiante o por pareja)
- Cuenta gratuita en la plataforma Scratch (<https://scratch.mit.edu/>)
- Proyector y computadora del docente para demostraciones
- Guía impresa o digital con instrucciones básicas de Scratch (bloques y funciones principales)
- Cuadernos o hojas para toma de notas y planificación de proyectos
- Material para lluvia de ideas: pizarras, plumones, notas adhesivas

- Videos tutoriales cortos sobre Scratch (preseleccionados por el docente)

Requisitos Previos

- Conocimientos básicos de uso de computadora e internet.
- Habilidades básicas para navegar en páginas web y manejar el mouse o touchpad.
- Experiencia previa mínima con entornos digitales o software educativo (de cursos anteriores).
- Capacidad para trabajar en equipo y comunicarse con sus compañeros.

Actividades

Sesión 1: Introducción a Scratch y primeros pasos en programación visual

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Conocer la plataforma Scratch, su interfaz y fundamentos básicos para iniciar un proyecto de programación visual.

Activación de conocimientos previos:

Docente: "¿Alguno ha usado o visto cómo se hacen juegos o animaciones en computadora? ¿Qué creen que se necesita para crear uno?"

Estudiantes: Responden en plenaria y comparten ideas.

Motivación y enganche:

Docente: Muestra un proyecto animado hecho en Scratch (ejemplo corto) y lanza el reto: "Hoy vamos a aprender a crear nuestras propias historias o juegos interactivos con Scratch, ¿quién quiere ser el próximo creador digital?"

Contextualización:

Docente: Explica cómo la programación está presente en muchas tecnologías que usan diariamente y cómo Scratch es una herramienta sencilla para iniciarse en la creación digital.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Mediante una demostración en computadora y proyector, el docente introduce la interfaz de Scratch, explicación breve de los bloques principales (movimiento, apariencia, eventos) y cómo crear un sprite y programarlo para que realice

acciones simples.

Actividades de aprendizaje activo:

• Actividad 1: Explorando la interfaz de Scratch

Objetivo: Analizar la estructura y herramientas básicas de Scratch.

Instrucciones:

- Docente guía a los estudiantes a abrir Scratch y explorar las secciones (escenario, sprites, bloques).
- Los estudiantes identifican qué hace cada área y prueban arrastrar bloques para mover un sprite.

Organización: Individual

Producto: Captura de pantalla o dibujo rápido del área de trabajo con etiquetas.

Tiempo: 15 minutos

Rol docente: Observa, responde dudas, hace preguntas como "¿Qué sucede si cambias este bloque?"

• Actividad 2: Crear una animación simple

Objetivo: Diseñar y programar un sprite para realizar movimientos y cambios visuales.

Instrucciones:

- Docente explica cómo usar bloques de movimiento y apariencia para animar un sprite.
- Estudiantes crean una secuencia donde su personaje se mueve y cambia de color o forma.

Organización: Parejas

Producto: Proyecto Scratch con animación básica.

Tiempo: 25 minutos

Rol docente: Asiste, plantea retos como "¿Puedes hacer que tu sprite diga algo?"

• Actividad 3: Compartir experiencias y observaciones

Objetivo: Reflexionar y comunicar aprendizajes iniciales.

Instrucciones:

- Cada pareja presenta brevemente su animación al grupo.
- Docente y compañeros comentan aspectos positivos y posibles mejoras.

Organización: Plenaria

Producto: Retroalimentación oral.

Tiempo: 5 minutos

Rol docente: Facilita y modera la discusión, destaca logros.

Diferenciación:

- Para estudiantes que avanzan rápido: Retos adicionales con bloques de sonido o control (esperar x segundos, repetir acciones).
- Para estudiantes que necesitan apoyo: Guías paso a paso impresas y acompañamiento individual para realizar la animación básica.

Transición:

Docente: "Ahora que conocen cómo mover y animar personajes, en la próxima sesión empezaremos a planear nuestro propio proyecto completo que cuente una historia o resuelva un problema usando estas herramientas."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Los estudiantes escriben en su cuaderno tres cosas nuevas que aprendieron hoy sobre Scratch.

Reflexión metacognitiva:

- ¿Qué parte de la interfaz de Scratch te pareció más fácil o difícil?
- ¿Cómo crees que puedes usar Scratch para contar una historia?
- ¿Qué te gustaría crear al finalizar este curso?

Retroalimentación:

Docente comenta y responde algunas respuestas, motivando la curiosidad.

Transferencia:

Se explica que en la próxima sesión comenzarán a diseñar sus proyectos y a programar más funciones.

Tarea o reto:

Explorar la plataforma Scratch en casa y probar crear un movimiento simple con cualquier sprite. Traer ideas para el proyecto.

Sesión 2: Planificación y diseño de proyectos en Scratch**Fase de Inicio**

Tiempo estimado: 10 minutos

Propósito de la sesión:

Conectar con los conocimientos previos y comenzar a planear el proyecto personal o grupal.

Activación de conocimientos previos:

Docente: "¿Qué aprendimos la sesión pasada sobre Scratch? ¿Qué ideas trajeron para sus proyectos?"

Estudiantes: Comparten breves respuestas y sus ideas.

Motivación y enganche:

Docente: Presenta ejemplos de proyectos terminados en Scratch relacionados con historias o soluciones a problemas reales para inspirar.

Contextualización:

Docente: Explica que para crear proyectos interesantes es fundamental planear qué quieren contar o lograr y cómo organizarán las acciones en Scratch.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Se introduce la importancia del diseño previo: guion gráfico (storyboard) y lista de acciones que debe tener el proyecto.

Actividades de aprendizaje activo:

- **Actividad 1: Lluvia de ideas y definición del proyecto**

Objetivo: Diseñar el concepto y propósito del proyecto Scratch.

Instrucciones:

- En grupos de 3-4, discuten y eligen un tema o problema para su proyecto (historia, juego, solución simple).
- Escriben el título y describen brevemente qué quieren crear.

Organización: Grupos de 3-4

Producto: Documento o papel con el título e idea general.

Tiempo: 15 minutos

Rol docente: Facilita la discusión, sugiere ideas y hace preguntas guía.

- **Actividad 2: Crear un storyboard**

Objetivo: Planificar visualmente las escenas o etapas del proyecto.

Instrucciones:

- Cada grupo dibuja en hojas o cuaderno las escenas o pasos del proyecto usando imágenes y texto.
- Incluyen qué acciones harán los sprites en cada escena.

Organización: Grupos de 3-4

Producto: Storyboard gráfico.

Tiempo: 20 minutos

Rol docente: Observa, sugiere mejoras y hace preguntas para clarificar ideas.

- **Actividad 3: Presentación rápida del plan**

Objetivo: Comunicar el diseño inicial y recibir retroalimentación.

Instrucciones:

- Cada grupo presenta su storyboard y explica su idea al grupo clase.

- Compañeros y docente aportan comentarios constructivos.

Organización: Plenaria

Producto: Retroalimentación oral.

Tiempo: 10 minutos

Rol docente: Modera y resalta aspectos positivos y áreas para mejorar.

Diferenciación:

- Para estudiantes avanzados: Pueden incluir detalles de programación prevista (bloques a usar).
- Para estudiantes con dificultades: Se proveen plantillas de storyboard con ejemplos para facilitar la organización.

Transición:

Docente: "Con su plan listo, en la próxima sesión comenzaremos a programar cada parte de su proyecto en Scratch."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Los estudiantes escriben en su cuaderno la idea principal de su proyecto y qué esperan lograr.

Reflexión metacognitiva:

- ¿Qué fue lo más fácil y lo más difícil al planear su proyecto?
- ¿Cómo trabajaron en equipo para definir la idea?
- ¿Qué cambiarías o agregarías a tu plan para hacerlo mejor?

Retroalimentación:

Docente comenta y reconoce la organización y creatividad mostrada.

Transferencia:

Se anticipa que el próximo paso es convertir ese plan en un proyecto real en Scratch.

Tarea o reto:

Pensar en qué personajes y escenarios usarán para su proyecto y buscar imágenes o ideas que puedan incluir.

Sesión 3: Programación básica de proyectos en Scratch

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Revisar el plan de proyecto y preparar la programación de las escenas iniciales.

Activación de conocimientos previos:

Docente: "¿Qué aprendimos sobre planificación? ¿Cómo podemos dividir nuestro proyecto en partes para programarlo?"

Estudiantes: Responden y recuerdan el storyboard.

Motivación y enganche:

Docente: Muestra cómo con bloques simples se pueden crear interacciones divertidas y que ellos harán eso mismo.

Contextualización:

Docente: Explica que la programación es como escribir instrucciones claras para que la computadora haga lo que queremos.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Se enseñan bloques de control (eventos al presionar bandera verde, esperar, repetir), condicionales simples (si-entonces), y manejo de sonido.

Actividades de aprendizaje activo:

• Actividad 1: Programar la primera escena

Objetivo: Crear el inicio del proyecto con eventos y movimientos básicos.

Instrucciones:

- En parejas o grupos según disponibilidad, los estudiantes programan la primera escena según su storyboard.
- Usan bloques para iniciar con la bandera verde, mover sprites y mostrar diálogos.

Organización: Parejas o grupos

Producto: Proyecto Scratch con escena inicial funcional.

Tiempo: 25 minutos

Rol docente: Apoya con dudas, sugiere bloques y verifica comprensión.

• Actividad 2: Agregar interacción básica

Objetivo: Incluir interacción simple con el usuario (clic, teclas).

Instrucciones:

- Docente explica cómo usar bloques de eventos para responder a clics o teclas.

- Estudiantes integran una acción que responda a alguna interacción (por ejemplo, que el sprite diga algo al hacer clic).

Organización: Parejas o grupos

Producto: Proyecto con interacción.

Tiempo: 15 minutos

Rol docente: Observa, guía y propone pequeños retos.

Diferenciación:

- Estudiantes avanzados pueden incluir condiciones más complejas o sonidos.
- Estudiantes con dificultades reciben apoyo personalizado o ejemplos de código para copiar y modificar.

Transición:

Docente: "Ya que programamos la primera parte, en la próxima sesión continuaremos con las escenas siguientes para completar la historia o solución."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Los estudiantes anotan en su cuaderno qué bloques usaron y qué aprendieron sobre la interacción.

Reflexión metacognitiva:

- ¿Qué fue lo más divertido de programar hoy?
- ¿Qué dificultades encontraste al crear interacciones?
- ¿Cómo te ayudó tu plan para hacer esta escena?

Retroalimentación:

El docente da comentarios positivos y sugiere continuar probando y mejorando.

Transferencia:

Invita a pensar en cómo seguirán desarrollando su proyecto en la siguiente sesión.

Tarea o reto:

Practicar en casa agregando una nueva interacción o movimiento a su proyecto Scratch.

Sesión 4: Programación avanzada y manejo de eventos en Scratch

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Revisar avances y presentar conceptos de programación avanzada como variables y bucles.

Activación de conocimientos previos:

Docente: "¿Qué escenas han avanzado? ¿Qué tipo de eventos usaron? ¿Conocen los bucles o variables?"

Estudiantes: Comparten experiencias y dudas.

Motivación y enganche:

Docente: Muestra un ejemplo donde se usan variables para contar puntos o controlar acciones, despertando interés.

Contextualización:

Docente: Explica que estas herramientas permiten hacer proyectos más dinámicos y personalizados.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Introducción a variables y bucles para repetir acciones o almacenar información.

Actividades de aprendizaje activo:

- **Actividad 1: Usar variables para contar eventos**

Objetivo: Programar un contador que registre acciones en el proyecto.

Instrucciones:

- Docente explica cómo crear y usar una variable en Scratch.
- Estudiantes agregan una variable que cuente por ejemplo clics o puntos en su proyecto.

Organización: Grupos o parejas

Producto: Proyecto con variable funcional.

Tiempo: 25 minutos

Rol docente: Apoya con dudas técnicas y motiva a experimentar.

- **Actividad 2: Programar bucles para animaciones continuas**

Objetivo: Implementar estructuras de repetición para animar sprites.

Instrucciones:

- Docente muestra ejemplos de bucles (repetir, por siempre).
- Estudiantes agregan animaciones que se repiten automáticamente.

Organización: Grupos o parejas

Producto: Proyecto con animaciones cíclicas.

Tiempo: 15 minutos

Rol docente: Observa avances y sugiere mejoras.

Diferenciación:

- Estudiantes avanzados: Integrar ambas herramientas para crear juegos sencillos con puntajes.
- Estudiantes con dificultades: Utilizar ejemplos guiados y repetir actividades con acompañamiento.

Transición:

Docente: "En la próxima sesión revisaremos cómo agregar sonidos y mejorar la interacción para hacer los proyectos más atractivos."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Los estudiantes resumen en una frase qué aprendieron sobre variables y bucles.

Reflexión metacognitiva:

- ¿Cómo te ayudaron las variables a mejorar tu proyecto?
- ¿Para qué sirven los bucles y cómo los usaste?
- ¿Qué te gustaría agregar la próxima vez?

Retroalimentación:

Docente reconoce la aplicación de conceptos y anima a seguir experimentando.

Transferencia:

Se anticipa la sesión siguiente, donde se integrarán sonidos y se pulirá el proyecto.

Tarea o reto:

Explorar sonidos en Scratch y pensar dónde podrían usarlos en su proyecto.

Sesión 5: Integrando sonidos y mejorando la interactividad

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Aprender a usar sonidos y mejorar la experiencia interactiva de los proyectos.

Activación de conocimientos previos:

Docente: "¿Qué sonidos conoces en Scratch? ¿Cómo podrías usarlos para hacer tu proyecto más divertido?"

Estudiantes: Responden y comparten ideas.

Motivación y enganche:

Docente: Reproduce un proyecto con sonidos integrados para captar interés.

Contextualización:

Docente: Explica cómo los sonidos hacen que los proyectos sean más atractivos y comunicativos.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Se explica cómo agregar sonidos desde la biblioteca o grabados, y cómo programar su reproducción según eventos.

Actividades de aprendizaje activo:

• Actividad 1: Añadir sonidos a eventos

Objetivo: Programar sonidos que se reproducen ante acciones del usuario.

Instrucciones:

- Docente muestra cómo insertar sonidos y usar bloques para reproducirlos.
- Estudiantes integran sonidos a sus proyectos en eventos clave (clics, movimientos, diálogos).

Organización: Grupos o parejas

Producto: Proyecto con sonidos activos.

Tiempo: 30 minutos

Rol docente: Asiste, corrige errores y anima a probar diversos sonidos.

• Actividad 2: Prueba y mejora colaborativa

Objetivo: Evaluar y mejorar proyectos con retroalimentación.

Instrucciones:

- Los grupos intercambian proyectos para probarlos y dar sugerencias.
- Reciben y anotan ideas para mejorar.

Organización: Grupos

Producto: Lista de mejoras o ajustes.

Tiempo: 15 minutos

Rol docente: Modera el intercambio y fomenta críticas constructivas.

Diferenciación:

- Estudiantes rápidos: Experimentan con grabar sus propios sonidos.
- Estudiantes que requieren apoyo: Reciben ejemplos concretos y acompañamiento para integrar sonidos.

Transición:

Docente: "En la última sesión, finalizaremos, presentaremos y reflexionaremos sobre nuestro aprendizaje."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Los estudiantes escriben dos beneficios que trae el uso de sonidos en sus proyectos.

Reflexión metacognitiva:

- ¿Cómo mejoraron tus proyectos con sonidos?
- ¿Qué aprendiste al probar otros proyectos?
- ¿Qué cambio te gustaría hacer antes de presentar?

Retroalimentación:

Docente destaca el esfuerzo y da sugerencias para el cierre.

Transferencia:

Se explica que en la siguiente sesión será la presentación final y reflexión.

Tarea o reto:

Revisar el proyecto para estar listos para mostrarlo.

Sesión 6: Presentación, reflexión y cierre del proyecto Scratch**Fase de Inicio**

Tiempo estimado: 10 minutos

Propósito de la sesión:

Preparar y organizar la presentación de los proyectos realizados.

Activación de conocimientos previos:

Docente: "¿Qué logros sienten que alcanzaron? ¿Qué mostrarán a sus compañeros?"

Estudiantes: Dialogan en parejas y se preparan.

Motivación y enganche:

Docente: Explica la importancia de compartir y aprender unos de otros.

Contextualización:

Docente: Recuerda que la programación es una forma de comunicar ideas y solucionar problemas.

Fase de Desarrollo

Tiempo estimado: 45 minutos

Presentación del contenido:

Se organiza la presentación grupal y se establecen criterios para dar retroalimentación respetuosa y útil.

Actividades de aprendizaje activo:

• **Actividad 1: Presentación de proyectos**

Objetivo: Comunicar el proceso y producto final del proyecto Scratch.

Instrucciones:

- Cada grupo presenta su proyecto explicando la idea, programación y desafíos.
- Compañeros y docente escuchan atentamente.

Organización: Plenaria

Producto: Presentación oral y demostración digital.

Tiempo: 30 minutos

Rol docente: Modera, anima y toma notas para retroalimentación.

• **Actividad 2: Reflexión y autoevaluación**

Objetivo: Evaluar el propio aprendizaje y proceso de trabajo.

Instrucciones:

- Estudiantes responden en su cuaderno preguntas de reflexión:
 - ¿Qué aprendí sobre programación con Scratch?
 - ¿Qué habilidades desarrollé trabajando en equipo?
 - ¿Qué haría diferente en otro proyecto?

Organización: Individual

Producto: Respuestas escritas.

Tiempo: 15 minutos

Rol docente: Recolecta respuestas y ofrece retroalimentación general.

Diferenciación:

- Estudiantes con mayor dominio pueden compartir ejemplos de código o dificultades superadas.
- Estudiantes que requieren apoyo pueden responder con dibujos o frases cortas.

Transición:

Docente: "Con esto concluimos nuestro proyecto Scratch, pero su aprendizaje puede seguir creciendo con más práctica."

Fase de Cierre

Tiempo estimado: 5 minutos

Síntesis:

Se realiza un mapa mental colectivo en pizarra con las cosas aprendidas y habilidades desarrolladas.

Reflexión metacognitiva:

- ¿Cómo me siento respecto a lo que logré en este proyecto?
- ¿Qué me gustaría aprender para seguir mejorando?
- ¿Cómo puedo usar estos conocimientos fuera de la escuela?

Retroalimentación:

Docente felicita el esfuerzo y destaca la importancia de la perseverancia.

Transferencia:

Invita a explorar más proyectos en Scratch y a compartir lo aprendido con familiares o amigos.

Tarea o reto:

Crear un nuevo proyecto personal en Scratch aplicando lo aprendido y compartirlo en la plataforma o con el docente.

Evaluación

Tipo de evaluación:

- **Diagnóstica:** Sesión 1 (activación de conocimientos previos sobre programación y uso de computadora)
- **Formativa:** Durante todas las sesiones en actividades prácticas, observación directa, retroalimentación continua y autoevaluación.
- **Sumativa:** Sesión 6, presentación final del proyecto y reflexión escrita.

Criterios de evaluación:

- Comprender y aplicar conceptos básicos de programación visual en Scratch (objetivo 1).
- Diseñar y desarrollar un proyecto interactivo coherente con planificación previa (objetivo 2).

- Participar activamente y colaborar en equipo para alcanzar metas comunes (objetivo 3).
- Incorporar mejoras basadas en retroalimentación recibida (objetivo 4).
- Reflexionar críticamente sobre el proceso de aprendizaje y su utilidad práctica (objetivo 5).

Instrumentos sugeridos:

- Lista de cotejo para seguimiento de actividades y participación.
- Rúbrica para evaluación del proyecto Scratch (planificación, programación, creatividad, presentación).
- Observación directa durante actividades prácticas.
- Autoevaluación escrita y coevaluación entre compañeros.
- Portafolio digital o físico con los productos generados: storyboard, proyectos Scratch, reflexiones.

Evidencias de aprendizaje:

- Capturas o proyectos guardados en Scratch que demuestran programación básica y avanzada.
- Storyboard y planificación escrita o gráfica.
- Presentaciones orales y demostraciones digitales en clase.
- Respuestas escritas de reflexión y autoevaluación.
- Retroalimentación documentada y ajustes realizados en proyectos.