

Scratch: fundamentos y navegación avanzada

Tecnología e Informática | Informática

Descripción del Curso

Este curso de Informática está diseñado para estudiantes mayores de 17 años y propone una trayectoria de aprendizaje que combina fundamentos de programación visual con prácticas de navegación avanzada. A lo largo de las unidades, los estudiantes desarrollarán habilidades técnicas, analíticas y de comunicación digital, aplicando conceptos de lógica de programación, estructuras, depuración y documentación para resolver problemas en contextos reales. El enfoque es práctico y orientado a proyectos, con evaluaciones que integran diseño, implementación, pruebas y presentación de resultados. La Unidad 8, titulada Proyecto final: integración de fundamentos y navegación avanzada, representa la culminación del curso. En esta unidad, los estudiantes crearán y presentarán un proyecto que integre los fundamentos de Scratch con técnicas de navegación avanzada. Se fomentará la justificación de decisiones de diseño, la depuración, la documentación y una presentación oral o escrita que explique las elecciones de implementación. Este proyecto final permitirá aplicar de forma cohesiva lo aprendido, demostrar capacidad de razonamiento crítico y comunicar de manera clara las soluciones propuestas. **Objetivo general.** Crear y presentar un proyecto final que integre fundamentos y navegación avanzada, justificando las decisiones de diseño, mostrando depuración y explicando las elecciones de implementación. **Específicos de la Unidad 8:** - Planificar un proyecto final con objetivos, alcance y criterios de éxito. - Aplicar depuración y pruebas para garantizar un funcionamiento robusto. - Presentar el proyecto, justificando decisiones de diseño y organizando la documentación de soporte.

Competencias

- Aplicar el pensamiento computacional para diseñar soluciones que integren Scratch y técnicas de navegación avanzada en situaciones reales.
- Desarrollar habilidades de depuración, pruebas y validación para garantizar la robustez de proyectos digitales.
- Justificar decisiones de diseño con fundamentos técnicos y comunicar de forma clara, precisa y adecuada la implementación.
- Expresar ideas y resultados de manera oral y escrita, favoreciendo la documentación de procesos y entregables.
- Trabajar de forma colaborativa, gestionar el tiempo y coordinar tareas para alcanzar objetivos de proyecto.
- Resolver problemas de manera autónoma y adaptativa, aplicando estrategias de aprendizaje continuo en contextos digitales.

Requerimientos

- Dispositivo con conexión a Internet y acceso a Scratch (en línea o versión instalada) para desarrollo de proyectos.
- Cuenta de usuario para almacenar y compartir entregables (p. ej., Scratch, Google Drive u otras plataformas designadas).

- Conocimientos básicos de Scratch: manejo de bloques, secuencias, eventos, bucles y manejo de variables.
- Compromiso para trabajar en las fases de planificación, implementación, depuración y presentación de resultados.
- Lecturas, actividades y entregas periódicas que exijan documentación de procesos y evidencia de pruebas.
- Habilidad para trabajar de forma individual o en equipo, según lo requiera el proyecto final, y para presentar resultados de manera oral o escrita.

Unidades del Curso

Unidad 1: Unidad 1: Identificación de los componentes de la interfaz de Scratch

Objetivos de Aprendizaje

- Reconocer visualmente el escenario, la paleta de bloques, el área de scripts y la lista de sprites dentro de Scratch.
- Describir la función de cada componente en el flujo de ejecución de un proyecto básico.
- Ejemplificar con una demostración simple cómo se conectan bloques para iniciar una acción en un sprite.

Contenidos Temáticos

1. Componentes de la interfaz: escenario, paleta de bloques, área de scripts, lista de sprites — descripción breve de cada elemento y su función en un proyecto.
2. Flujo de ejecución básico: cómo se ejecutan los bloques para obtener resultados simples al interactuar con los sprites.
3. Exploración guiada de Scratch: navegación inicial y prácticas de reconocimiento de elementos.

Actividades

- **Actividad 1: Exploración guiada de la interfaz** – El alumnado abre Scratch, identifica cada componente y describe su función en una nota breve. Puntos clave: ubicación del escenario, paleta de bloques, área de scripts y lista de sprites. Aprendizaje: reconocimiento visual y vocabulario básico.
- **Actividad 2: Construcción de un miniprograma** – Se eligen dos bloques simples para hacer que un sprite realice una acción básica tras activar un evento. Aprendizaje: conexión de bloques y flujo de acción.
- **Actividad 3: Demostración de flujo de proyecto** – El equipo presenta verbalmente el flujo de un proyecto sencillo, identificando dónde comienza, qué eventos disparan bloques y cómo evoluye la acción.

Evaluación

- Identificación correcta de los componentes de la interfaz en Scratch y explicación de su función (objetivo general).
- Capacidad para describir el flujo de ejecución de un proyecto mínimo y demostrar la conexión de bloques para realizar una acción simple (objetivos específicos 1 y 3).

- Participación en la exploración y claridad en la presentación de los componentes de la interfaz (competencia comunicativa y comprensión).

Unidad 2: Unidad 2: Lógica de bloques y secuencias básicas en Scratch

Objetivos de Aprendizaje

- Identificar bloques de diferentes categorías (Movimiento, Apariencia, Control, Eventos) y entender su función.
- Comprender cómo apilar bloques para crear secuencias lógicas de acciones.
- Ejemplificar con un par de casos prácticos cómo una secuencia puede cambiar la posición o la apariencia de un sprite.

Contenidos Temáticos

1. Categorías de bloques y su función (Movimiento, Apariencia, Control, Eventos) — descripción breve.
2. Conexión y flujo de bloques: anidamiento y secuenciación de acciones.
3. Ejemplos de secuencias simples: mover, cambiar disfraz, emitir sonidos.

Actividades

- **Actividad 1: Construcción de una secuencia básica** – Crear un script que mueva un sprite en una dirección, cambie su disfraz y reproduzca un sonido en secuencia. Aprendizaje: lógica de bloques, orden de operaciones.
- **Actividad 2: Experimentos con bucles simples** – Implementar un bucle que repita una acción (p. ej., desplazamiento repetido) y observar el resultado. Aprendizaje: repetición controlada.
- **Actividad 3: Desafío de flujo de acciones** – Diseñar una mini-tarea donde un evento dispara una serie de acciones encadenadas (salto corto, cambio de color, sonido) con al menos dos pasos consecutivos.

Evaluación

- Comprensión de la lógica de bloques y capacidad para crear una secuencia de acciones simples en Scratch (objetivo general).
- Habilidad para organizar bloques de forma correcta, asegurando el flujo deseado (objetivos específicos 1 y 2).
- Presentación de al menos dos ejemplos de secuencias y su efecto en el sprite (objetivo específico 3).

Unidad 3: Unidad 3: Responder a eventos y crear acciones encadenadas

Objetivos de Aprendizaje

- Configuración de un evento (clic o tecla) como disparador de una acción.
- Creación de una secuencia de al menos dos acciones tras el evento (p. ej., mover y cambiar apariencia).
- Intercambio de ideas sobre responsabilidad del usuario al interactuar con el proyecto (usabilidad básica).

Contenidos Temáticos

1. Eventos en Scratch: al hacer clic, al presionar tecla, entre otros disparadores — descripción breve.
2. Secuencias y acciones encadenadas tras un evento.
3. Buenas prácticas de interacción usuario-programa: retroalimentación visual y sonora.

Actividades

- **Actividad 1: Interacción clic-tecla** – Programa que responde al clic y, a continuación, ejecuta dos acciones (p. ej., movimiento y cambio de color). Aprendizaje: manejo de eventos y secuencias.
- **Actividad 2: Juego mínimo interactivo** – Crear una pequeña interacción entre dos sprites que responden a un evento y realizan dos acciones consecutivas cada vez que se activa.
- **Actividad 3: Prueba y evaluación de respuestas** – Probar diferentes eventos y registrar cuál respuesta genera cada secuencia de acciones, con una breve retroalimentación.

Evaluación

- Capacidad para definir y activar un evento y codificar una secuencia de al menos dos acciones (objetivo general).
- Claridad en la lógica de la respuesta basada en eventos y en la consistencia de las acciones encadenadas (objetivos específicos 1 y 2).
- Demostración de comprensión de interacción usuario-proyecto (objetivo específico 3).

Unidad 4: Unidad 4: Diseño de una animación corta con múltiples sprites y manejo de estado

Objetivos de Aprendizaje

- Definir variables para controlar el estado de la animación (p. ej., variable "estado" o "fase").
- Aplicar bucles y estructuras condicionales para coordinar comportamientos entre sprites.
- Diseñar una escena con interacción entre al menos dos sprites y una progresión narrativa simple.

Contenidos Temáticos

1. Variables para estado: definición, alcance y uso práctico.
2. Bucles y condicionales en animaciones inter-sprites.
3. Coordinación de dos sprites: ritmo, sincronización y respuesta al estado.

Actividades

- **Actividad 1: Preparar el estado con una variable** – Crear una variable de estado para una animación con dos sprites y usarla para alternar comportamientos. Aprendizaje: uso de variables para controlar flujo.
- **Actividad 2: Animación coordinada** – Desarrollar una secuencia donde un sprite persigue o acompaña al otro, con bucles y decisiones condicionales.

- **Actividad 3: Presentación de la animación** – Mostrar la animación a la clase y justificar las decisiones de estado y sincronización.

Evaluación

- Capacidad para crear y gestionar una variable de estado y para coordinar acciones entre dos sprites usando bucles y condicionales (objetivo general).
- Corrección de la lógica de animación y claridad en la presentación del diseño (objetivos específicos 1 y 2).
- Demostración de coherencia narrativa y técnica en la animación (objetivo específico 3).

Unidad 5: Variables y listas para almacenar información y controlar comportamiento

Objetivos de Aprendizaje

- Definir y usar variables para almacenar valores simples (p. ej., puntuación, estado).
- Crear y manipular listas para almacenar colecciones de datos (p. ej., secuencias de eventos, puntuaciones).
- Aplicar variables y listas para modificar comportamientos de sprites durante la ejecución.

Contenidos Temáticos

1. Concepto de variables y listas: definición, alcance y usos clave.
2. Operaciones básicas con variables y manejo de listas (añadir, eliminar, leer).
3. Ejemplos prácticos: sistema de puntuación, registro de historias simples, control de estados con listas.

Actividades

- **Actividad 1: Sistema de puntuación** – Implementar una variable para puntuación que aumente cuando ocurren acciones y se muestre en pantalla. Aprendizaje: lectura y escritura de variables.
- **Actividad 2: Lista de eventos** – Crear una lista para registrar acciones de usuario o eventos que afecten a un sprite y usarla para decidir el siguiente comportamiento.
- **Actividad 3: Demostración de efecto** – Integrar variable y lista en un mini-proyecto y analizar cómo influyen en el comportamiento del proyecto.

Evaluación

- Capacidad para crear y manipular variables y listas, y demostrar su efecto en el comportamiento de sprites (objetivo general).
- Claridad en el uso de estructuras de datos para sostener estado y decisiones de flujo (objetivos específicos 1 y 2).
- Presentación de un ejemplo práctico con análisis de impacto (objetivo específico 3).

Unidad 6: Resolución de problemas comunes: colisiones, detección y lógica

Objetivos de Aprendizaje

- Utilizar sensores y operadores para detectar colisiones y toques entre sprites.
- Aplicar operadores lógicos para tomar decisiones en función de condiciones múltiples.
- Diseñar respuestas de comportamiento adaptativas ante detecciones (colisiones, toques, límites).

Contenidos Temáticos

1. Detección de colisiones y toques: sensores y bloques relevantes.
2. Operadores lógicos y decisiones condicionales complejas.
3. Patrones de depuración y pruebas de comportamiento.

Actividades

- **Actividad 1: Colisiones y reacciones** – Diseñar un escenario donde dos sprites detectan colisiones y responden con una acción específica (detenerse, cambiar de dirección).
- **Actividad 2: Toma de decisiones basada en condiciones** – Crear un comportamiento que use condiciones compuestas (Y, O) para decidir qué acción tomar ante diferentes estados.
- **Actividad 3: Proceso de depuración** – Identificar errores comunes en interacciones y aplicar pasos de depuración para corregir comportamientos no deseados.

Evaluación

- Capacidad para implementar detección de colisiones y respuestas adecuadas (objetivo general).
- Uso correcto de operadores lógicos para tomar decisiones (objetivos específicos 1 y 2).
- Demostración de depuración y pruebas efectivas que aseguren el comportamiento esperado (objetivo específico 3).

Unidad 7: Unidad 7: Navegación avanzada por Scratch: gestión de proyectos y atajos

Objetivos de Aprendizaje

- Gestionar múltiples versiones de un proyecto y mantener un historial de cambios.
- Organizar recursos y materiales de apoyo para un proyecto de Scratch.
- Explorar extensiones y atajos para optimizar el flujo de trabajo.

Contenidos Temáticos

1. Gestión de proyectos y versiones: práctica de control de cambios.
2. Organización de recursos: bibliotecas, imágenes, sonidos y arte assets.
3. Extensiones y atajos: herramientas para ampliar Scratch y acelerar el desarrollo.

Actividades

- **Actividad 1: Control de versiones** – Crear y guardar diferentes versiones de un proyecto, compararlas y documentar cambios clave.
- **Actividad 2: Organización de recursos** – Clasificar y etiquetar recursos (sonidos, sprites, fondos) para un proyecto grande.
- **Actividad 3: Extensiones y atajos** – Explorar al menos una extensión disponible y aplicar atajos para acelerar la edición y depuración.

Evaluación

- Demostración de gestión de versiones y registro de cambios (objetivo general).
- Capacidad para organizar recursos de manera eficiente y justificar las elecciones (objetivos específicos 1 y 2).
- Adecuada adopción de extensiones y atajos para mejorar el flujo de trabajo (objetivo específico 3).

Unidad 8: Unidad 8: Proyecto final: integración de fundamentos y navegación avanzada

Objetivos de Aprendizaje

- Planificar un proyecto final con objetivos, alcance y criterios de éxito.
- Aplicar depuración y pruebas para garantizar un funcionamiento robusto.
- Presentar el proyecto, justificando decisiones de diseño y organizando la documentación de soporte.

Contenidos Temáticos

1. Planificación de un proyecto: objetivos, recursos y cronograma.
2. Depuración y pruebas: métodos y registro de hallazgos.
3. Presentación y documentación: estructura de informe y demostración en vivo.
4. Integración de extensiones y buenas prácticas de trabajo colaborativo.

Actividades

- **Actividad 1: Planificación del proyecto final** – Elaborar un plan con objetivos, fases, recursos y criterios de éxito. Aprendizaje: gestión de proyectos y claridad de propósito.
- **Actividad 2: Desarrollo y depuración** – Construir el proyecto final de Scratch, realizar pruebas sistemáticas y documentar hallazgos de depuración.
- **Actividad 3: Presentación final** – Presentar el proyecto ante la clase, justificando decisiones de diseño y mostrando evidencias de funcionamiento.

Evaluación

- Calidad del proyecto final en términos de funcionalidad, organización y uso de buenas prácticas (objetivo general).

- Manifestación de depuración y capacidad para justificar decisiones de diseño (objetivos específicos 1 y 3).
- Claridad de la presentación y documentación que acompañan al proyecto (objetivo específico 3).