

Idea de Robot con Microbit, Arduino, Scratch y Minecraft!

Tecnología e Informática | Pensamiento Computacional | para estudiantes de secundaria (12-15 años) | 4 semanas

Descripción del Curso

Este curso está diseñado para estudiantes de secundaria interesados en explorar el fascinante mundo de la robótica y la programación mediante la integración de plataformas tecnológicas accesibles como Microbit, Arduino, Scratch y Minecraft. A lo largo de cuatro semanas, los estudiantes aprenderán a diseñar y construir un proyecto tecnológico que combine hardware y software para resolver un problema real, aplicando conceptos de pensamiento computacional y lógica de programación.

El curso está dirigido a jóvenes de 12 a 15 años con interés en la tecnología, sin necesidad de experiencia previa, proporcionando un enfoque práctico y colaborativo que promueve la creatividad, el trabajo en equipo y el aprendizaje basado en proyectos. Se enfatiza el desarrollo de habilidades para programar dispositivos físicos y simular soluciones en entornos virtuales, fomentando un aprendizaje activo y significativo.

Al finalizar, los estudiantes serán capaces de diseñar, programar y controlar un robot o simulación digital que integre diversas herramientas tecnológicas, demostrando comprensión en electrónica básica, programación visual y textual, así como capacidad para resolver problemas mediante soluciones innovadoras y funcionales.

Objetivos Generales

- Identificar y describir los componentes básicos de Microbit, Arduino, Scratch y Minecraft para su uso en proyectos tecnológicos.
- Planificar y diseñar un proyecto integrador que combine robótica y programación para resolver un problema real o simulado.
- Programar dispositivos físicos y entornos virtuales utilizando lenguajes visuales y de texto adaptados a su nivel.
- Implementar y probar soluciones tecnológicas funcionales que respondan a necesidades planteadas.
- Evaluar y comunicar los resultados del proyecto, reflexionando sobre el proceso y posibles mejoras.

Competencias

- Diseñar y desarrollar proyectos tecnológicos que integren hardware y software para resolver problemas específicos.
- Aplicar conceptos básicos de programación utilizando entornos visuales y textuales como Scratch, Microbit y Arduino.
- Utilizar plataformas digitales como Minecraft para simular y controlar dispositivos y procesos automatizados.
- Analizar y solucionar problemas mediante el pensamiento computacional y la lógica programática.
- Colaborar en equipos multidisciplinarios para planificar y ejecutar proyectos tecnológicos.
- Comunicar de manera clara y efectiva el proceso y resultados de un proyecto tecnológico.

Requerimientos

- Conocimientos básicos de informática y manejo de computadora.
- Acceso a una computadora con conexión a internet para usar Scratch, Minecraft y software de Microbit y Arduino.
- Kit básico de Microbit y/o Arduino con sensores y actuadores.
- Cuenta en plataformas educativas como Scratch y Minecraft Education Edition o acceso a Minecraft Java con mods educativos.
- Software gratuito para programación y simulación (Microsoft MakeCode, Arduino IDE, Minecraft).
- Motivación para aprender y trabajar en equipo.

Unidades del Curso

Unidad 1: Introducción a la robótica y programación con Microbit y Scratch

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar los componentes básicos del Microbit y del entorno Scratch mediante la observación y exploración guiada.
- Al finalizar la unidad, el estudiante será capaz de describir el funcionamiento de sensores y actuadores en proyectos de robótica utilizando ejemplos prácticos en Microbit y Scratch.
- Al finalizar la unidad, el estudiante será capaz de programar acciones sencillas en Microbit y Scratch que involucren la interacción entre sensores y actuadores, siguiendo instrucciones paso a paso.
- Al finalizar la unidad, el estudiante será capaz de analizar y corregir errores básicos en programas simples realizados en Microbit y Scratch para asegurar su correcto funcionamiento.
- Al finalizar la unidad, el estudiante será capaz de explicar cómo la programación visual facilita la creación de proyectos robóticos mediante la presentación de un breve informe o exposición.

Contenidos Temáticos

1. Introducción a la robótica y programación visual

- Concepto de robótica: definición y aplicaciones prácticas en la vida cotidiana.
- Importancia de la programación visual: ventajas para principiantes y relación con la robótica.
- Presentación general del Microbit y Scratch como plataformas educativas.

2. Componentes básicos del Microbit y el entorno Scratch

- Microbit: descripción y función de los componentes principales (LEDs, botones, sensores integrados, pines de entrada/salida).
- Entorno de programación Scratch: interfaz, bloques básicos, escenario, sprites y cómo se organizan los scripts.

- Exploración guiada para identificar visualmente estos componentes y herramientas.

3. Sensores y actuadores en proyectos de robótica

- Definición de sensores y actuadores: qué son y su función en un robot.
- Ejemplos concretos de sensores en Microbit (botones, acelerómetro, sensor de luz).
- Actuadores en Microbit y Scratch: LEDs, sonidos, movimiento de sprites.
- Cómo se conectan y comunican sensores y actuadores en proyectos simples.

4. Programación de acciones sencillas con Microbit y Scratch

- Creación de programas básicos en Microbit que respondan a la interacción con sensores (por ejemplo, presionar un botón para encender LEDs).
- Programación en Scratch para controlar sprites mediante eventos y sensores simulados.
- Uso de bloques de control, eventos y sensores en ambos entornos.
- Guía paso a paso para elaborar proyectos simples que integren sensores y actuadores.

5. Análisis y corrección de errores básicos en programas

- Identificación de errores comunes en programas sencillos (bloques mal ubicados, lógica incorrecta, falta de eventos).
- Estrategias para depurar y corregir estos errores en Microbit y Scratch.
- Prácticas de prueba y ajuste para asegurar el funcionamiento correcto de los programas.

6. Comunicación y reflexión sobre la programación visual y robótica

- Elaboración de un informe o presentación breve que explique cómo la programación visual ayuda en la creación de proyectos robóticos.
- Discusión sobre experiencias personales y beneficios de usar Microbit y Scratch.
- Reflexión sobre el aprendizaje obtenido y posibles aplicaciones futuras.

Actividades

Actividad 1: Explorando el Microbit y Scratch

Objetivo: Identificar los componentes básicos del Microbit y del entorno Scratch mediante la observación y exploración guiada.

Descripción paso a paso:

- El docente presenta físicamente un Microbit y proyecta la interfaz de Scratch en pantalla.
- Los estudiantes observan y describen en grupos pequeños los componentes visibles del Microbit (LEDs, botones, pines) y los elementos del entorno Scratch.
- Se realiza una dinámica de preguntas y respuestas para reforzar la identificación de cada componente.
- Los estudiantes completan una ficha de observación donde anotan las funciones principales de cada componente.

Organización: Grupos de 3-4 estudiantes.

Producto esperado: Ficha de observación completa con componentes identificados y funciones descritas.

Duración estimada: 45 minutos.

Actividad 2: Programando con sensores y actuadores en Microbit

Objetivo: Programar acciones sencillas en Microbit que involucren la interacción entre sensores y actuadores, siguiendo instrucciones paso a paso.

Descripción paso a paso:

- El docente muestra un ejemplo básico donde al presionar un botón del Microbit se encienden los LEDs.
- Los estudiantes acceden al editor de Microbit en línea y replican el programa siguiendo una guía paso a paso.
- Luego modifican el programa para cambiar la secuencia de LEDs o agregar sonidos.
- Finalmente, prueban sus programas en el simulador o en el dispositivo físico si está disponible.

Organización: Individual o parejas.

Producto esperado: Programa funcional que responda a la interacción con sensores y active actuadores.

Duración estimada: 60 minutos.

Actividad 3: Creación de un proyecto en Scratch con sensores simulados

Objetivo: Programar acciones sencillas en Scratch que involucren la interacción entre sensores simulados y actuadores.

Descripción paso a paso:

- El docente explica cómo usar eventos y sensores simulados (por ejemplo, detectar la posición del ratón o teclas presionadas).
- Los estudiantes diseñan un pequeño proyecto donde un sprite responde a la interacción del usuario (por ejemplo, cambiar de color o moverse).
- Se enfatiza el uso de bloques de control, eventos y apariencia para simular sensores y actuadores.
- Finalmente, los estudiantes presentan su proyecto a sus compañeros explicando cómo funciona la interacción programada.

Organización: Parejas o grupos pequeños.

Producto esperado: Proyecto Scratch que muestre interacción entre eventos/sensores simulados y acciones del sprite.

Duración estimada: 60 minutos.

Actividad 4: Diagnóstico y corrección de errores en programas simples

Objetivo: Analizar y corregir errores básicos en programas simples realizados en Microbit y Scratch.

Descripción paso a paso:

- El docente proporciona ejemplos de programas con errores comunes (bloques mal colocados, lógica errónea).

- Los estudiantes trabajan en parejas para identificar los errores y proponer correcciones.
- Se discuten en plenaria las soluciones y se realizan pruebas para verificar el funcionamiento correcto.
- Finalmente, cada pareja presenta un breve informe describiendo el error y cómo lo corrigieron.

Organización: Parejas.

Producto esperado: Informe breve y programa corregido que funcione correctamente.

Duración estimada: 45 minutos.

Actividad 5: Presentación y reflexión sobre la programación visual en robótica

Objetivo: Explicar cómo la programación visual facilita la creación de proyectos robóticos mediante la presentación de un breve informe o exposición.

Descripción paso a paso:

- Los estudiantes preparan un breve informe o presentación (oral o digital) que resuma su experiencia aprendiendo a programar con Microbit y Scratch.
- Se enfatiza la explicación de los beneficios de la programación visual y la importancia de sensores y actuadores en robótica.
- Se realiza una ronda de exposiciones frente al grupo, con retroalimentación del docente y compañeros.

Organización: Individual o grupos pequeños.

Producto esperado: Informe escrito o presentación oral que explique la programación visual y sus aplicaciones robóticas.

Duración estimada: 45 minutos.

Evaluación

Evaluación diagnóstica

Qué se evalúa: Conocimientos previos sobre robótica, programación visual, y familiaridad con Microbit y Scratch.

Cómo se evalúa: Mediante una encuesta de preguntas abiertas y cerradas al inicio de la unidad que incluye identificación de componentes y conceptos básicos.

Instrumento sugerido: Cuestionario digital o impreso con preguntas de selección múltiple y respuesta corta.

Evaluación formativa

Qué se evalúa: Progreso en la identificación de componentes, comprensión de sensores y actuadores, habilidad para programar acciones sencillas y corregir errores.

Cómo se evalúa: Observación directa durante las actividades, revisión de productos parciales (fichas, programas, informes), y retroalimentación continua.

Instrumento sugerido: Rúbricas de desempeño para actividades prácticas, listas de cotejo para identificación de componentes, y registros anecdóticos del docente.

Evaluación sumativa

Qué se evalúa: Capacidad para integrar conocimientos y habilidades en un proyecto final que incluya programación con sensores y actuadores, análisis de errores y presentación explicativa.

Cómo se evalúa: A través de la entrega y presentación de un proyecto final en Microbit y Scratch, acompañado de un informe o exposición que explique la programación visual y su aplicación.

Instrumento sugerido: Rúbrica que contemple aspectos técnicos (funcionamiento del programa), comprensión teórica (explicación en informe/presentación) y calidad del trabajo final.

Unidad 2: Fundamentos de Arduino y programación textual básica

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir los componentes básicos del hardware Arduino y su entorno de programación Arduino IDE, utilizando terminología técnica adecuada.
- Al finalizar la unidad, el estudiante será capaz de conectar sensores y actuadores básicos a una placa Arduino siguiendo esquemas y normas de seguridad establecidas.
- Al finalizar la unidad, el estudiante será capaz de escribir y depurar programas simples en lenguaje de programación textual para Arduino que controlen dispositivos físicos según instrucciones específicas.
- Al finalizar la unidad, el estudiante será capaz de ejecutar pruebas funcionales para verificar el correcto funcionamiento de los sensores, actuadores y programas implementados en Arduino.

Unidad 3: Simulación y control en Minecraft

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de identificar y describir los comandos básicos y mods educativos en Minecraft para simular procesos automatizados dentro del juego.
- Al finalizar la unidad, el estudiante será capaz de programar secuencias de comandos en Minecraft que controlen elementos del entorno virtual para representar soluciones tecnológicas simples.
- Al finalizar la unidad, el estudiante será capaz de diseñar y ejecutar una simulación en Minecraft que integre conceptos de automatización y control, aplicando lógica de programación aprendida.
- Al finalizar la unidad, el estudiante será capaz de evaluar el funcionamiento de su simulación en Minecraft, identificar fallas o mejoras y ajustar la programación para optimizar el control de los elementos.

Unidad 4: Diseño y desarrollo del proyecto integrador

Objetivos de Aprendizaje

- Al finalizar la unidad, el estudiante será capaz de planificar y diseñar un proyecto integrador que combine Microbit, Arduino, Scratch y Minecraft para resolver un problema real o simulado, siguiendo un esquema de trabajo en

equipo.

- Al finalizar la unidad, el estudiante será capaz de programar y conectar dispositivos físicos y entornos virtuales utilizando lenguajes visuales y de texto, aplicando pruebas y ajustes para asegurar el funcionamiento correcto del proyecto.
- Al finalizar la unidad, el estudiante será capaz de implementar y evaluar soluciones tecnológicas funcionales, identificando errores y realizando mejoras basadas en pruebas y retroalimentación.
- Al finalizar la unidad, el estudiante será capaz de comunicar y presentar el proyecto integrador, explicando el proceso de diseño, desarrollo y las posibles mejoras, utilizando un lenguaje claro y adecuado para su audiencia.