

Descubriendo el Mundo de los Sistemas Embebidos con Arduino

Tecnología e Informática | Informática | Aprendizaje Basado en Problemas

Descripción

Este plan de clase está diseñado para que estudiantes de secundaria (12-15 años) conozcan el fascinante mundo de los sistemas embebidos a través de la plataforma Arduino. Aprenderán qué es Arduino, las características principales de sus placas y microcontroladores, el uso del entorno de desarrollo integrado (IDE) y la estructura básica de un programa (sketch) con las funciones `setup()` y `loop()`. Además, realizarán la instalación del IDE de Arduino en sus dispositivos y crearán una cuenta en la plataforma Tinkercad para simular proyectos. Esta experiencia es relevante porque los sistemas embebidos están presentes en muchos dispositivos cotidianos como electrodomésticos, automóviles y gadgets tecnológicos, por lo que comprender su funcionamiento les abre la puerta a la innovación tecnológica. El enfoque basado en problemas promueve que los estudiantes analicen y resuelvan situaciones reales, desarrollando pensamiento crítico y habilidades prácticas que pueden aplicar en su vida diaria y en futuros estudios tecnológicos.

Objetivos de Aprendizaje

- Analizar qué es Arduino y sus componentes principales, incluyendo placas y microcontroladores.
- Identificar y explicar la estructura básica de un sketch de Arduino (`setup()` y `loop()`).
- Instalar el IDE de Arduino en el equipo y crear una cuenta en Tinkercad para simulación de proyectos.
- Aplicar el entorno de desarrollo para crear y simular un programa sencillo en Arduino.

Recursos Necesarios

- Computadora con acceso a internet (1 por estudiante o pareja).
- Proyector o pantalla para presentación.
- Material audiovisual: video introductorio sobre Arduino (3-5 minutos).
- Manual impreso o digital con instrucciones para instalación del IDE de Arduino.
- Cuenta de correo electrónico para creación de cuenta en Tinkercad.
- Acceso a <https://www.arduino.cc> y <https://www.tinkercad.com>
- Guía impresa o digital con estructura básica de sketch de Arduino y comandos principales.

Requisitos Previos

- Conocimientos básicos de computación e internet.
- Experiencia previa con programas sencillos o entornos digitales.

- Habilidad para seguir instrucciones técnicas básicas.
- Comprensión básica de programación lógica (secuencias, ciclos).

Actividades

Fase de Inicio

Tiempo estimado:

20 minutos

Propósito de la sesión:

Docente: Explica que hoy explorarán qué es Arduino y cómo se usa para controlar dispositivos electrónicos, conectando tecnología con la vida cotidiana. Se enfatiza la importancia de los sistemas embebidos en objetos que usan diariamente.

Activación de conocimientos previos:

Docente: Presenta una pregunta detonadora: "¿Alguna vez se han preguntado cómo funciona el control remoto de la televisión o cómo se encienden y apagan las luces automáticamente?" Pide que comenten ideas breves en plenaria para activar su conocimiento sobre tecnología integrada en objetos comunes.

Motivación y enganche:

Docente: Muestra un video corto (3-5 minutos) que introduce Arduino, mostrando proyectos simples como un semáforo, luces que parpadean o sensores. Explica que ellos podrán crear y simular algo similar.

Contextualización:

Docente: Explica que Arduino es una herramienta que permite dar "vida" a la tecnología que usan todos los días y que hoy aprenderán las bases para comenzar a programar y simular proyectos que podrían mejorar su entorno o divertirse creando gadgets.

Estudiantes: Participan respondiendo la pregunta, observan el video con atención y escuchan la explicación, tomando notas si desean.

Fase de Desarrollo

Tiempo estimado:

80 minutos

Presentación del contenido:

Docente: Introduce el concepto de Arduino haciendo preguntas para guiar el aprendizaje: "¿Qué creen que es una placa Arduino?" y "¿Para qué sirve el microcontrolador?" Explica de forma sencilla con ejemplos visuales el hardware y

software, la función del IDE y la estructura básica de un sketch (setup() y loop()).

Actividad 1: Explorando la placa Arduino y su estructura

- **Objetivo específico:** Analizar qué es Arduino y sus componentes principales.
- **Instrucciones:**
 - **Docente:** Entrega imágenes impresas o muestra en pantalla la placa Arduino y señala sus partes: microcontrolador, pines, USB, etc.
 - Explica la función de cada parte con lenguaje sencillo.
 - Solicita que en parejas identifiquen y anoten las partes de la placa en una hoja o cuaderno.
 - Luego, explica la estructura de un sketch y escribe en el pizarrón las funciones setup() y loop() con ejemplos simples.
- **Organización:** Parejas
- **Producto:** Lista anotada con partes de la placa y breve explicación escrita de las funciones setup() y loop().
- **Tiempo:** 30 minutos
- **Rol docente:** Observa que identifiquen correctamente las partes, formula preguntas como "¿Para qué creen que sirve el microcontrolador?" y apoya con aclaraciones.

Transición:

Docente: Conecta la actividad con la siguiente diciendo: "Ahora que conocen la placa y el programa, vamos a configurar su entorno para que puedan programar y simular sus propios proyectos."

Actividad 2: Instalación del IDE de Arduino y creación de cuenta en Tinkercad

- **Objetivo específico:** Instalar el IDE y crear cuenta en Tinkercad para simulación.
- **Instrucciones:**
 - **Docente:** Guía paso a paso la instalación del IDE de Arduino en sus computadoras con ayuda de un manual impreso o digital.
 - Después, los estudiantes crean una cuenta en Tinkercad con su correo electrónico.
 - El docente muestra cómo ingresar a la sección de circuitos y crear un proyecto nuevo.
 - Los estudiantes exploran la interfaz simulando un circuito básico con un LED.
- **Organización:** Individual o parejas (según disponibilidad de equipos)
- **Producto:** Captura de pantalla o foto del proyecto creado en Tinkercad con un circuito básico.
- **Tiempo:** 35 minutos
- **Rol docente:** Apoya a quienes tengan dificultades técnicas, formula preguntas para reforzar comprensión, como "¿Qué es el IDE y para qué sirve?", y supervisa el avance.

Diferenciación:

- **Para estudiantes que terminan antes:** Se les invita a explorar funciones adicionales del IDE o crear un segundo circuito en Tinkercad con un botón y LED.
- **Para quienes requieren más apoyo:** El docente proporciona ayuda individualizada, material visual adicional y ejemplos más sencillos para reforzar comprensión.

Actividad 3: Primer sketch en Arduino (setup y loop)

- **Objetivo específico:** Aplicar la estructura básica de un sketch para crear un programa sencillo.
- **Instrucciones:**
 - **Docente:** Explica que van a escribir un programa que encienda y apague un LED en la simulación.
 - Proporciona el código base y guía a los estudiantes para que lo ingresen en el IDE o Tinkercad.
 - Los estudiantes ejecutan la simulación y observan el resultado.
 - Se promueve que modifiquen tiempos de encendido y apagado para experimentar.
- **Organización:** Individual o parejas
- **Producto:** Código del sketch funcionando y captura de la simulación.
- **Tiempo:** 15 minutos
- **Rol docente:** Supervisa que comprendan la función de setup() y loop(), pregunta "¿Qué hace cada parte del código?" y apoya en errores de sintaxis o ejecución.

Fase de Cierre

Tiempo estimado:

20 minutos

Síntesis:

Docente: Solicita a los estudiantes realizar un "ticket de salida" donde escriban en 3 frases qué es Arduino, para qué sirve el setup() y el loop(), y cómo instalaron el IDE o crearon su cuenta en Tinkercad.

Reflexión metacognitiva:

- ¿Qué aprendí hoy sobre los sistemas embebidos y Arduino?
- ¿Cómo me ayudó la simulación en Tinkercad a entender mejor la programación?
- ¿Qué parte del proceso me pareció más fácil o difícil y por qué?

Retroalimentación:

Docente: Revisa los tickets de salida y da comentarios verbales individuales o grupales resaltando aciertos y aclarando dudas comunes. Refuerza conceptos clave y felicita el esfuerzo y la participación.

Transferencia:

Docente: Explica que en futuras sesiones podrán crear proyectos más complejos y que lo aprendido les servirá para entender mejor la tecnología que los rodea y desarrollar habilidades para innovar.

Tarea o reto:

Docente: Propone investigar en casa qué otros dispositivos usan sistemas embebidos y traer un ejemplo para comentar en la próxima clase.

Estudiantes: Realizan el ticket de salida, responden preguntas de reflexión y reciben retroalimentación.

Evaluación

Tipo de evaluación: Diagnóstica al inicio con preguntas detonadoras; formativa durante las actividades prácticas; sumativa al cierre con el ticket de salida y productos digitales.

Criterios de evaluación:

- Analiza correctamente las partes y funciones de la placa Arduino (Objetivo 1).
- Identifica y explica la estructura básica de un sketch (Objetivo 2).
- Completa la instalación del IDE y creación de cuenta en Tinkercad correctamente (Objetivo 3).
- Aplica la programación básica para simular un circuito sencillo (Objetivo 4).

Instrumentos sugeridos:

- Lista de cotejo para observar identificación de componentes y explicación del sketch.
- Observación directa durante la instalación y simulación.
- Revisión del ticket de salida como evidencia escrita de comprensión.
- Capturas o fotos de proyectos en Tinkercad como evidencia tecnológica.

Evidencias de aprendizaje:

- Lista anotada de partes de la placa y explicación del sketch.
- Capturas de pantalla del proyecto en Tinkercad con circuito simulado.
- Código del sketch básico con funciones `setup()` y `loop()`.
- Ticket de salida con respuestas reflexivas.

Enriquecimientos

Desarrollo - Ejemplos

Ejemplos Prácticos y Casos de Estudio para "Descubriendo el Mundo de los Sistemas Embebidos con Arduino"

Estos ejemplos y casos de estudio están diseñados para que los estudiantes de secundaria (12-15 años) puedan conectar conceptos teóricos con aplicaciones reales usando Arduino, fomentando el aprendizaje activo mediante la metodología Aprendizaje Basado en Problemas (ABP).

Ejemplo Práctico 1: Semáforo Inteligente para Peatones

- **Problema planteado:** En la escuela o comunidad, los peatones a veces tienen dificultad para cruzar la calle de manera segura porque no hay un semáforo que los ayude.
- **Actividad:** Los estudiantes diseñan un semáforo básico con Arduino utilizando LEDs (rojo, amarillo, verde) para simular las luces y programan un sketch que controle el encendido y apagado según un ciclo definido.
- **Objetivos que conecta:** Conocer las placas y microcontroladores de Arduino, familiarizarse con la estructura básica del sketch (`setup()` y `loop()`), y usar el IDE para programar.
- **Recurso:** Uso de Tinkercad para simular el circuito y el código antes de hacer la práctica física.

Ejemplo Práctico 2: Sensor de Luz Automático para Lámpara de Escritorio

- **Problema planteado:** Cuando un estudiante estudia en la noche, quiere que la lámpara se encienda automáticamente cuando haya poca luz para no interrumpir su concentración.
- **Actividad:** Programar Arduino para leer un sensor de luz (LDR) y encender/apagar un LED que represente la lámpara según la cantidad de luz ambiental.
- **Objetivos que conecta:** Entender el microcontrolador y la estructura del sketch, instalación y uso del IDE, y creación de cuenta en Tinkercad para simulación.

Caso de Estudio: Control Básico de un Robot Seguidor de Línea

- **Problema planteado:** Un grupo de estudiantes quiere construir un robot que pueda seguir una línea negra en el suelo para participar en una competencia escolar.
- **Actividad:** Analizar cómo un Arduino puede controlar motores y sensores para seguir la línea. Se presenta el esquema básico del sketch con `setup()` para inicializar sensores y `loop()` para leer continuamente y tomar decisiones.
- **Discusión ABP:** Los estudiantes deben identificar qué componentes electrónicos se necesitan, cómo se programan las acciones básicas y cómo usar el IDE para escribir y cargar código.
- **Aplicación práctica:** En la sesión, se puede simular el robot en Tinkercad o proponer una versión simplificada con LEDs y sensores simulados.

Ejemplo Práctico 3: Alarma Simple con Sensor de Movimiento

- **Problema planteado:** En casa o en la escuela, se quiere un sistema sencillo que detecte movimiento y emita una alerta.
- **Actividad:** Programar Arduino para detectar señales de un sensor PIR y activar un buzzer o LED como alarma.
- **Objetivos que conecta:** Uso de la placa Arduino, comprensión del microcontrolador, programación básica con estructura `setup()` y `loop()`, y uso del IDE/Tinkercad.

Guía para el Docente: Implementación ABP

- Presentar el problema real al inicio para motivar la investigación y el análisis.

- Dividir a los estudiantes en equipos para que discutan posibles soluciones y planifiquen la programación y montaje en simulador.
- Guiar la instalación del IDE o creación de cuenta en Tinkercad para simular circuitos y código.
- Promover la reflexión final donde cada equipo explique cómo resolvieron el problema usando Arduino y qué aprendieron sobre la estructura del código y el entorno de programación.

Desarrollo - Gamificar

Elementos de Gamificación para la Fase de Desarrollo

Para motivar a los estudiantes y reforzar los objetivos de aprendizaje durante la sesión de 2 horas, se proponen las siguientes mecánicas de juego adaptadas a su edad y contexto educativo, alineadas con la metodología de Aprendizaje Basado en Problemas (ABP):

- **Desafío de Conocimiento "Reto Arduino":**

Durante la explicación y práctica del concepto de Arduino, sus placas y componentes, se plantean pequeñas preguntas tipo quiz en formato de "reto rápido" que los estudiantes deben responder en equipos o individualmente. Cada respuesta correcta suma puntos para su equipo.

- Ejemplo: ¿Qué es un microcontrolador? ¿Qué función tiene la estructura `setup()`?
- Duración: 10-15 minutos repartidos en pequeñas rondas.

- **Construcción del Sketch - Juego de Secuencias:**

Para aprender la estructura básica de un sketch (`setup()` y `loop()`), se propone un juego donde los estudiantes deben ordenar tarjetas con fragmentos de código para construir un programa simple. Al hacerlo correctamente, ganan "puntos de construcción".

- Las tarjetas pueden incluir comandos básicos, comentarios y estructuras.
- Duración aproximada: 15 minutos.

- **Misiones en Tinkercad:**

Una vez que instalen el IDE y creen su cuenta en Tinkercad, se les asignan "misiones" específicas para completar, como simular un LED que se enciende y apaga con un sketch básico.

- La finalización de cada misión otorga insignias digitales visibles para la clase (pueden ser imágenes en una presentación compartida o en un tablero).
- Duración: 30-40 minutos, con apoyo del docente.

- **Competencia Amistosa "Arduino Master":**

Al final de la sesión, se organiza una breve competencia donde los estudiantes responden preguntas y resuelven mini retos relacionados con Arduino para ganar puntos extra y un reconocimiento simbólico (certificado digital o sticker virtual).

- La competencia fomenta colaboración y refuerza el aprendizaje.
- Duración: 15 minutos.

Resumen de Gamificación y Tiempo Aproximado

Actividad Gamificada	Objetivo de Aprendizaje Reforzado	Duración Aproximada
Desafío de Conocimiento "Reto Arduino"	Comprender conceptos básicos de Arduino, placas y microcontrolador	10-15 minutos
Construcción del Sketch - Juego de Secuencias	Identificar la estructura setup()/loop() y fragmentos de código	15 minutos
Misiones en Tinkercad	Instalación del IDE, uso de Tinkercad, creación y simulación de sketches	30-40 minutos
Competencia Amistosa "Arduino Master"	Repaso y consolidación de conocimientos	15 minutos

Estos elementos de gamificación están diseñados para mantener el interés de los estudiantes, promover la colaboración y apoyar la comprensión de los conceptos clave del plan de clase sin perder el foco en los objetivos de aprendizaje.