

Secuencias, bucles y condicionales

Tecnología e Informática | Informática

Descripción del Curso

Curso de Informática dirigido a estudiantes de 13 a 14 años, con énfasis en la Unidad 4: Proyecto práctico integrado. Esta unidad propone aplicar de forma integrada los conceptos aprendidos de secuencias, bucles y condicionales para resolver un problema práctico, como un contador de puntos. El proyecto final invita a los alumnos a diseñar, implementar y presentar su código, acompañando la versión funcional con una explicación de las decisiones tomadas durante el proceso de desarrollo. El enfoque es práctico y orientado a la comprensión de cómo las estructuras básicas de la programación se combinan para producir soluciones útiles en contextos reales. A lo largo del proyecto, los estudiantes planificarán entradas, salidas y reglas del problema, escribirán código estructurado y depurado, y comunicarán de forma clara las elecciones de diseño y el aprendizaje obtenido. Se fomenta el desarrollo del pensamiento lógico, la resolución de problemas, la capacidad de trabajar en equipo, la comunicación técnica y la responsabilidad digital. Al finalizar, se espera que el alumnado pueda convertir un problema cotidiano en una solución computacional funcional y justificar su enfoque ante un público, promoviendo la reflexión sobre el uso responsable de la tecnología.

Competencias

- Pensamiento lógico y resolución de problemas aplicado a escenarios prácticos.
- Uso correcto de secuencias, bucles y estructuras condicionales para crear soluciones funcionales.
- Planificación de proyectos: definir entradas, salidas y reglas del problema.
- Desarrollo y depuración básica de código con claridad y orden.
- Comunicación técnica: explicar decisiones de diseño y presentar el código de forma comprensible.
- Colaboración y trabajo en equipo, con distribución de tareas y gestión del tiempo.
- Presentación de resultados y defensa de las decisiones ante un público.
- Alfabetización digital y ética en el uso de tecnologías de la información.

Requerimientos

- Dispositivo con acceso a Internet (computadora, tablet o similar) para programar y entregar tareas.
- Entorno de desarrollo o plataforma educativa proporcionada por la escuela (p. ej., Python u otro lenguaje recomendado).
- Cuenta de usuario válida y acceso a las herramientas de programación y entrega de entregables.
- Cuaderno o espacio digital para notas, diagramas de flujo y planificación del proyecto.
- Tiempo asignado para planificar, escribir, probar y presentar el proyecto final.

- Guía de evaluación o rúbrica para el proyecto y criterios de retroalimentación.
- Compromiso con normas de seguridad y ética digital durante las actividades.

Unidades del Curso

Unidad 1: Unidad 1: Secuencias básicas

Objetivos de Aprendizaje

- Identificar qué es una secuencia de instrucciones y distinguirla de bucles y condicionales.
- Escribir un programa sencillo que sume dos números dados por el usuario.
- Ejecutar un programa en el que la salida depende de una secuencia de operaciones lineales y describir su resultado.

Contenidos Temáticos

1. **Tema 1:** Concepto de secuencia: una serie de instrucciones que se ejecutan en orden, sin saltos, repeticiones ni decisiones.
2. **Tema 2:** Entradas, salidas y operaciones aritméticas básicas necesarias para sumar dos números.
3. **Tema 3:** Escribir y probar un programa sencillo que sume dos números proporcionados por el usuario.

Actividades

- **Actividad 1: Exploración de instrucciones secuenciales**

Tema: Identificar la secuencia de acciones en una tarea cotidiana y trasladarla a código sencillo.

Puntos clave: entender el orden de ejecución; no usar decisiones ni repeticiones; verificar el resultado paso a paso.

Aprendizajes/Conclusiones: el programa debe seguir un orden fijo para producir una salida esperada; las instrucciones se ejecutan una vez.

- **Actividad 2: Escribir un programa de suma**

Tema: Crear un programa que tome dos números y los sume, mostrando el resultado.

Puntos clave: recibir entradas, realizar la operación aritmética y presentar la salida; evitar condiciones o bucles.

Aprendizajes/Conclusiones: comprensión de una secuencia lineal que produce un resultado directo a partir de entradas simples.

- **Actividad 3: Lectura y verificación de salidas**

Tema: Analizar el resultado de un programa secuencial y explicar por qué funciona.

Puntos clave: interpretar la salida, confirmar que corresponde a la secuencia de operaciones, identificar posibles errores de orden.

Aprendizajes/Conclusiones: habilidad para describir el comportamiento del programa y justificar el resultado obtenido.

Evaluación

- Competencia evaluada: capacidad para escribir y ejecutar una secuencia de instrucciones sin bucles ni condicionales.
- Criterios de logro:
 - El código realiza la tarea en un único flujo lineal sin decisiones.
 - La salida coincide con el resultado esperado de la operación aritmética.
 - La lectura del código demuestra comprensión de la secuencia y su orden.
- Instrumentos de evaluación: (a) ejecución de un programa de suma; (b) revisión de código; (c) preguntas orales simples sobre la secuencia.

Unidad 2: Bucles y repetición

Objetivos de Aprendizaje

- Comprender qué es un bucle y para qué sirve en la programación.
- Diseñar e implementar un bucle que repita una tarea (p. ej., contar) hasta alcanzar una condición de parada.
- Identificar posibles errores de bucle (bucle infinito, conteos incorrectos) y proponer depuración básica.

Contenidos Temáticos

1. **Tema 1:** ¿Qué es un bucle? Tipos básicos (repetición con y sin conteo) y su utilidad.
2. **Tema 2:** Diseño de un bucle que cuente números desde 1 hasta n o que repita una tarea hasta una condición.
3. **Tema 3:** Depuración sencilla de bucles para evitar bucles infinitos y conteos erróneos.

Actividades

- **Actividad 1: Construyendo un bucle de conteo**

Tema: Crear un bucle que cuente del 1 al 10 y muestre cada número.

Puntos clave: inicialización, condición de continuación, incremento.

Aprendizajes/Conclusiones: entender el ciclo de vida de un bucle y su parada controlada.

- **Actividad 2: Bucle que repite hasta una condición**

Tema: Pedir un número al usuario y repetir una tarea hasta que el usuario indique terminar (por ejemplo, ingresar -1).

Puntos clave: estructura de repetición con condición de parada, manejo de entrada.

Aprendizajes/Conclusiones: la repetición debe detenerse cuando se cumple la condición adecuada y evitar bucles infinitos.

- **Actividad 3: Depuración de bucles**

Tema: Analizar bucles con fallos comunes y proponer correcciones para obtener la salida esperada.

Puntos clave: detectar condiciones mal planteadas, cambios en el límite y el incremento.

Aprendizajes/Conclusiones: capacidad para diagnosticar y corregir errores de repetición.

Evaluación

- Competencia evaluada: diseño y ejecución de bucles que cumplen una condición de parada.
- Criterios de logro:
 - El bucle repite la tarea correctamente hasta la condición de parada.
 - El conteo o la salida son correctos y no hay bucles infinitos.
 - El razonamiento sobre la selección de la estructura de bucle es claro.
- Instrumentos de evaluación: código del bucle, explicación escrita y ejercicios de depuración.

Unidad 3: Unidad 3: Condicionales y combinaciones

Objetivos de Aprendizaje

- Comprender la función de los condicionales en la toma de decisiones dentro de un programa.
- Diseñar e implementar un programa que integre secuencias, bucles y condicionales para resolver un problema simple.
- Explicar por qué la lógica del programa produce el resultado observado, señalando el flujo de ejecución.

Contenidos Temáticos

1. **Tema 1:** Condicionales básicos (if/else) y condiciones simples.
2. **Tema 2:** Integración de estructuras: combinación de secuencias, bucles y condicionales.
3. **Tema 3:** Ejemplo práctico: clasificación de números u otro problema sencillo que requiera lógica condicional.

Actividades

• Actividad 1: Uso de condicionales

Tema: Implementar decisiones simples para clasificar números como positivos o negativos, con salida acorde.

Puntos clave: condición, rama verdadera y rama falsa, salida adecuada.

Aprendizajes/Conclusiones: comprender cuándo y cómo usar if/else para tomar decisiones en un programa.

• Actividad 2: Integración de estructuras

Tema: Diseñar un pequeño programa que combine secuencias, un bucle y una condicional para resolver un problema básico (p. ej., verificar si un número está en un intervalo).

Puntos clave: flujo de ejecución, interacción entre estructuras, claridad del código.

Aprendizajes/Conclusiones: capacidad de reunir diferentes estructuras para cumplir un objetivo.

- **Actividad 3: Análisis del comportamiento**

Tema: Analizar un programa que ya contiene secuencias, bucles y condicionales y describir por qué funciona y dónde podría fallar.

Puntos clave: explicación del flujo, efectos de las decisiones, límites de las condiciones.

Aprendizajes/Conclusiones: habilidad para justificar la lógica y las salidas esperadas del programa.

Evaluación

- Competencia evaluada: capacidad para ejecutar y justificar un programa que usa secuencias, bucles y condicionales.
- Criterios de logro:
 - El programa toma decisiones correctas según las condiciones dadas.
 - La combinación de estructuras funciona de manera coherente y predecible.
 - Se describe el comportamiento del programa y el razonamiento temporal de su ejecución.
- Instrumentos de evaluación: código funcional, explicación escrita del flujo y respuestas a preguntas sobre el comportamiento.

Unidad 4: Unidad 4: Proyecto práctico integrado

Objetivos de Aprendizaje

- Planificar un proyecto corto: definir entradas, salidas y reglas del problema.
- Escribir código que utilice secuencias, bucles y condicionales para una solución funcional.
- Presentar el código con una justificación clara de las decisiones de diseño y aprendizaje obtenido.

Contenidos Temáticos

1. **Tema 1:** Planificación del proyecto: alcance, entradas, salidas y criterios de éxito.
2. **Tema 2:** Implementación: diseño de la solución, escritura del código y prueba de funcionamiento.
3. **Tema 3:** Presentación y evaluación de la solución: explicación de decisiones y resultados.

Actividades

- **Actividad 1: Planificación del proyecto**

Tema: Definir un problema práctico (p. ej., contador de puntos) y planificar entradas, salidas y reglas de negocio.

Puntos clave: objetivos claros, especificación de entrada/salida, criterios de éxito.

Aprendizajes/Conclusiones: capacidad para diseñar una solución antes de codificar y establecer criterios de evaluación.

- **Actividad 2: Implementación del proyecto**

Tema: Escribir y probar el código que usa secuencias, bucles y condicionales para resolver el problema planteado.

Puntos clave: estructura ordenada, comentarios, pruebas de casos reales, depuración.

Aprendizajes/Conclusiones: integración de conceptos en una solución funcional y comprensible.

• **Actividad 3: Presentación y revisión**

Tema: Presentar el código y explicar las decisiones de diseño, además de reflexionar sobre posibles mejoras.

Puntos clave: claridad de la explicación, evidencias de funcionamiento, retroalimentación de pares.

Aprendizajes/Conclusiones: capacidad de comunicar ideas técnicas y justificar elecciones de implementación.

Evaluación

- Competencia evaluada: capacidad para diseñar, implementar y justificar una solución integrada que use secuencias, bucles y condicionales.
- Criterios de logro:
 - La solución funciona correctamente para casos de prueba relevantes.
 - El código está organizado, documentado y es legible.
 - La explicación demuestra comprensión de las decisiones tomadas y del aprendizaje obtenido.
- Instrumentos de evaluación: proyecto funcional, código comentado, informe de diseño y presentación oral o escrita.