

Introducción a algoritmos y estructuras de control

Tecnología e Informática | Informática

Descripción del Curso

Esta unidad cierra el curso de Informática al proponer la implementación de un programa básico en un lenguaje de aula o pseudolenguaje que integra las tres estructuras de control principales: secuencia, selección y repetición. El objetivo es que el estudiante diseñe, codifique y verifique una solución simple a un caso práctico, validando su comportamiento mediante al menos dos casos de prueba. Aunque la unidad está orientada a estudiantes de 17 años en adelante, sus principios promueven el pensamiento computacional, la depuración y la comunicación clara de soluciones, habilidades transferibles a contextos educativos y profesionales. Durante el proceso, el alumno debe definir el flujo lógico del programa, traducirlo a pseudocódigo o lenguaje de aula, implementar la solución y analizar diferentes escenarios de prueba para identificar y corregir errores. Se enfatiza la claridad en la documentación del código, la justificación de las elecciones estructurales (cuándo emplear secuencia, selección o repetición) y la capacidad de explicar el razonamiento ante otros. Al finalizar la unidad, el estudiante habrá generado un programa funcional que combine las estructuras de control, acompañándolo de al menos dos pruebas que demuestren su correcto comportamiento y permitan evaluar posibles mejoras or debug.

Competencias

- Desarrollar pensamiento algorítmico y resolución de problemas mediante el diseño de soluciones simples y su validación. - Diseñar y documentar pseudocódigo o lenguaje de aula de forma clara, estructurada y legible. - Implementar un programa básico que integre secuencia, selección y repetición para resolver un caso práctico. - Ejecutar, analizar y depurar al menos dos escenarios de prueba, identificando errores y proponiendo correcciones. - Comunicar razonamientos, decisiones de diseño y resultados de pruebas de forma efectiva, favoreciendo la colaboración y la transparencia.

Requerimientos

- Computadora o dispositivo con acceso a un entorno de desarrollo para pseudolenguaje o lenguaje de aula. - Conocimientos previos básicos de estructuras de control: secuencia, selección y repetición. - Herramientas para escribir y ejecutar pseudocódigo o programas en lenguaje de aula. - Guía o plantilla para definir y registrar al menos dos casos de prueba. - Capacidad para interpretar resultados de ejecución y documentar hallazgos.

Unidades del Curso

Unidad 1: Unidad 1: ¿Qué es un algoritmo y sus componentes?

Objetivos de Aprendizaje

- Identificar la entrada, el procesamiento y la salida en ejemplos cotidianos.
- Describir el propósito de un algoritmo para resolver un problema concreto.
- Distinguir entre un problema y una solución algorítmica a través de ejemplos simples.

Contenidos Temáticos

1. **Tema 1: Definición de algoritmo y componentes** - Descripción corta: qué es un algoritmo y sus partes fundamentales (entrada, proceso y salida).
2. **Tema 2: Entrada, Proceso y Salida** - Descripción corta: cómo se alimentan los datos, qué se hace con ellos y qué se obtiene al final.
3. **Tema 3: Propósito de un algoritmo en la resolución de problemas** - Descripción corta: por qué conviene usar algoritmos para ordenar, calcular o decidir.

Actividades

1. **Actividad 1: Detectar algoritmos en la vida diaria** - Observa una tarea cotidiana (por ejemplo, preparar un sándwich) y describe la entrada, el proceso y la salida.
 - Puntos clave: identificar entrada, procesamiento y salida en situaciones reales.
 - Aprendizaje: comprender que todo procedimiento tiene una estructura clara y repetible.
2. **Actividad 2: Plantilla de algoritmo en un ejemplo sencillo** - Escribe en lenguaje natural los pasos de hacer una bebida simple, indicando qué información entra, qué se procesa y qué se obtiene.
3. **Actividad 3: Discusión guiada** - Analizar diferentes problemas y proponer cuál sería la entrada, el proceso y la salida de cada uno.

Evaluación

Se evaluarán los siguientes elementos para verificar el logro del objetivo general y sus específicos:

- Identificación correcta de entrada, proceso y salida en ejemplos dados (Evaluación de Objetivo Específico 1).
- Explicación clara del propósito de un algoritmo para resolver un problema (Evaluación de Objetivo Específico 2).
- Capacidad para distinguir entre problema y posible solución algorítmica a través de ejemplos simples (Evaluación de Objetivo Específico 3).

Unidad 2: Unidad 2: Secuencia — ejecución lineal de tareas

Objetivos de Aprendizaje

- Definir qué es la secuencia y qué implica un flujo lineal de instrucciones.
- Escribir pseudocódigo que siga una secuencia de pasos sin decisiones.
- Analizar el flujo de ejecución para identificar si hay saltos no deseados o interrupciones.

Contenidos Temáticos

1. **Tema 1: Flujo lineal y secuencia** - Descripción corta: cómo se encadenan pasos sin branching.
2. **Tema 2: Representación en pseudocódigo** - Descripción corta: traducción de una tarea lineal a pseudocódigo.
3. **Tema 3: Ejemplos cotidianos de secuencia** - Descripción corta: actividades diarias como secuencias de pasos simples.

Actividades

1. **Actividad 1: Secuencia para hacer una tostada** - Desarrolla una lista de pasos en orden para preparar una tostada.
 - Puntos clave: orden de operaciones, sin decisiones.
 - Aprendizaje: comprender la necesidad de un flujo lineal para completar la tarea.
2. **Actividad 2: Pseudocódigo para sumar números** - Escribe los pasos en pseudocódigo para sumar una lista de números dada.
3. **Actividad 3: Diagrama de flujo de una tarea diaria** - Dibuja el diagrama de flujo de un proceso simple sin ramas.

Evaluación

Se evaluarán los objetivos 1, 2 y 3 de la unidad a través de:

- Identificación del concepto de secuencia en ejemplos propuestos.
- Creación de pseudocódigo correcto y ordenado siguiendo una secuencia lineal.
- Análisis de un diagrama de flujo o representación de pasos para confirmar la ausencia de decisiones.

Unidad 3: Unidad 3: Selección — decisiones con condiciones

Objetivos de Aprendizaje

- Identificar cuándo usar una toma de decisión (si/entonces) en un problema sencillo.
- Escribir pseudocódigo que incluya estructuras de selección.
- Representar decisiones mediante diagramas de flujo para mostrar ramas de ejecución.

Contenidos Temáticos

1. **Tema 1: Introducción a la selección** - Descripción corta: cuándo se necesita decidir entre opciones.
2. **Tema 2: Condiciones verdaderas y falsas** - Descripción corta: evaluar en qué casos se cumplen o no las condiciones.
3. **Tema 3: Representación de decisiones** - Descripción corta: uso de diagrama de flujo para ramas condicionales.

Actividades

1. **Actividad 1: Decisiones simples** - Crear un pseudocódigo con una condición para decidir si una persona es mayor de edad.
 - Puntos clave: evaluación de condiciones, rutas de ejecución distintas.
 - Aprendizaje: entender cómo una condición cambia el camino a seguir.
2. **Actividad 2: Pseudocódigo con selección** - Escribir un algoritmo que indique si un número es positivo, negativo o cero.
3. **Actividad 3: Diagrama de flujo de una decisión** - Representar gráficamente una decisión simple (p. ej., si llueve, llevar paraguas; si no, no).

Evaluación

Se evaluarán los objetivos 1-3 a través de:

- Capacidad para identificar cuándo aplicar una selección.
- Precisión en la escritura de pseudocódigo con condiciones.
- Representación correcta de ramas en diagramas de flujo y su interpretación.

Unidad 4: Unidad 4: Repetición — estructuras de bucle

Objetivos de Aprendizaje

- Reconocer cuándo es adecuado usar un bucle (repetir hasta una condición).
- Escribir pseudocódigo con repetición (while/for) para resolver problemas simples.
- Ejecutar de forma manual un algoritmo con repeticiones para observar salidas en distintos escenarios.

Contenidos Temáticos

1. **Tema 1: Concepto de repetición** - Descripción corta: explicación de bucles y su utilidad.
2. **Tema 2: Tipos de bucles** - Descripción corta: for y while, diferencias y casos de uso.
3. **Tema 3: Ejemplos de repetición** - Descripción corta: tareas repetitivas simples y su automatización.

Actividades

1. **Actividad 1: Contar números con un bucle** - Escribe un pseudocódigo que imprima los números del 1 al 10 usando un bucle.
2. **Actividad 2: Media de una lista** - Utiliza un bucle para sumar una lista de números y calcular la media.
3. **Actividad 3: Diagrama de flujo de una tarea repetitiva** - Dibuja un diagrama que muestre un ciclo de verificación hasta cumplir la condición.

Evaluación

Se evaluarán los objetivos 1-3 mediante:

- Capacidad para identificar cuándo aplicar una repetición y cuál tipo de bucle elegir.
- Corrección del pseudocódigo que implementa una repetición.
- Precisión en la simulación manual del bucle y en la interpretación de su salida.

Unidad 5: Unidad 5: Pseudocódigo y diagramas de flujo

Objetivos de Aprendizaje

- Convertir una descripción algorítmica en pseudocódigo y viceversa.
- Diseñar diagrama de flujo que refleje correctamente la lógica de un problema sencillo.
- Comparar las dos representaciones por claridad y facilidad de comprensión.

Contenidos Temáticos

1. **Tema 1: Conceptos de pseudocódigo** - Descripción corta: reglas básicas y notación común.
2. **Tema 2: Diagramas de flujo** - Descripción corta: símbolos y flujo de control básico.
3. **Tema 3: Puentes entre representaciones** - Descripción corta: convertir entre pseudocódigo y diagrama de flujo.

Actividades

1. **Actividad 1: De un enunciado a pseudocódigo** - Tomar un enunciado simple y escribir su pseudocódigo correspondiente, luego dibujar su diagrama de flujo.
2. **Actividad 2: De pseudocódigo a diagrama de flujo** - Tomar un pseudocódigo sencillo y representar su flujo como diagrama de flujo.
3. **Actividad 3: Análisis de claridad** - Comparar dos representaciones para el mismo problema y justificar cuál es más clara para un lector no experto.

Evaluación

Se evaluarán los objetivos 1-3 mediante:

- Exactitud en la conversión entre representaciones.
- Capacidad de lectura y comprensión de diagramas y pseudocódigo.
- Justificación de la representación más clara para comunicar la solución.

Unidad 6: Unidad 6: Simulación y ejecución manual de algoritmos

Objetivos de Aprendizaje

- Realizar seguimiento paso a paso de un algoritmo para obtener su salida.
- Identificar posibles errores de lógica durante la simulación.
- Comparar escenarios distintos para entender el comportamiento del algoritmo.

Contenidos Temáticos

1. **Tema 1: Métodos de simulación** - Descripción corta: técnica manual y herramientas simples para simular ejecución.
2. **Tema 2: Seguimiento de pasos** - Descripción corta: cómo registrar cada paso y la salida intermedia.
3. **Tema 3: Análisis de escenarios** - Descripción corta: cambiar entradas y observar salidas para validar el algoritmo.

Actividades

1. **Actividad 1: Seguimiento manual de un bucle** - Simula la ejecución de un bucle que cuenta desde 1 hasta 5 y registra cada iteración y salida.
2. **Actividad 2: Simulación de una condición y un bucle** - Dibuja una tabla de estados para un algoritmo que verifica si un número es par y, si no, incrementa hasta que sea par.
3. **Actividad 3: Análisis de resultados** - Compara salidas en diferentes escenarios y explica por qué cambian.

Evaluación

Se evaluarán los objetivos 1-3 mediante:

- Capacidad de seguir el flujo paso a paso con precisión.
- Identificación de errores de lógica durante la simulación.
- Justificación de resultados para distintos escenarios de entrada.

Unidad 7: Unidad 7: Comparación de soluciones y claridad de las estructuras

Objetivos de Aprendizaje

- Identificar distintas estrategias para resolver un problema sencillo.
- Evaluar la claridad y la complejidad de las estructuras utilizadas en cada solución.
- Justificar la elección de la solución más adecuada para un público objetivo y un contexto dado.

Contenidos Temáticos

1. **Tema 1: Estrategias alternativas** - Descripción corta: comparar enfoques lineales vs. condicionales y bucles.
2. **Tema 2: Criterios de claridad** - Descripción corta: legibilidad, simplicidad, mantenimiento y extensibilidad.
3. **Tema 3: Ejemplos prácticos** - Descripción corta: resolver un problema simple con dos enfoques y compararlos.

Actividades

1. **Actividad 1: Análisis de dos soluciones** - Resolver un problema (p. ej., encontrar el mayor de tres números) con dos enfoques diferentes y comparar su claridad.

2. **Actividad 2: Justificación escrita** - Redactar una justificación de cuál enfoque es más claro para principiantes y por qué.
3. **Actividad 3: Debate en clase** - Discusión guiada sobre ventajas y desventajas de cada solución.

Evaluación

Se evaluarán los objetivos 1-3 mediante:

- Presentación de al menos dos soluciones para un problema y análisis de su claridad.
- Justificación objetiva de la solución más adecuada considerando la simplicidad de las estructuras de control.
- Participación en el análisis y debate en clase.

Unidad 8: Unidad 8: Implementación de un programa básico y casos de prueba

Objetivos de Aprendizaje

- Diseñar un programa pequeño que integre las tres estructuras de control principales (secuencia, selección y repetición).
- Escribir el programa en pseudolenguaje y/o lenguaje de aula.
- Ejecutar y analizar al menos dos escenarios de prueba, identificando errores y corrigiéndolos.

Contenidos Temáticos

1. **Tema 1: Diseño de un programa básico** - Descripción corta: planificar un proyecto corto que combine las estructuras aprendidas.
2. **Tema 2: Pseudolenguaje o lenguaje de aula** - Descripción corta: sintaxis básica y construcción de estructuras de control.
3. **Tema 3: Casos de prueba** - Descripción corta: crear y ejecutar al menos dos casos de prueba para validar el programa.

Actividades

1. **Actividad 1: Desarrollo de un pequeño programa** - Escribe un programa en pseudolenguaje que, por ejemplo, tome una lista de números, calcule su suma y la media, usando secuencia, selección y repetición cuando proceda.
2. **Actividad 2: Casos de prueba** - Definir al menos dos escenarios de entrada y registrar las salidas esperadas frente a las salidas obtenidas.
3. **Actividad 3: Depuración guiada** - Ejecutar el programa con los casos de prueba y corregir errores lógicos o de sintaxis simples.

Evaluación

Se evaluarán los objetivos 1-3 mediante:

- Correcta integración de secuencia, selección y repetición en un solo programa.
- Precisión y claridad del pseudolenguaje o lenguaje de aula utilizado.
- Éxito al ejecutar dos o más casos de prueba y justificar las salidas obtenidas o corregidas.