

Actividad | 1 | Algoritmos

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Abel Arreola Flores

FECHA: 29/07/2025

Índice

Introducción	3
Descripción	3
Justificación	4
Desarrollo.....	5
Calculadora de números primos	5
Calculadora para determinar números pares o impares	6
Calculadora para invertir un número	7
Conclusión	8
Referencias.....	8

Introducción

El desarrollo de algoritmos es un medio fundamental para comprender los principios de la programación y como análisis previo para la construcción de aplicaciones y programas. Un algoritmo es una secuencia ordenada de pasos que permiten transformar una necesidad en una solución clara, precisa y repetible. Diseñar y representar algoritmos es una habilidad fundamental para cualquier persona que aspire a programar. Elaborar algoritmos correctamente no solo mejora la eficiencia del software, sino que ayuda a prevenir errores y a construir soluciones más robustas desde el inicio. Además, fortalece el pensamiento lógico, la capacidad de análisis y la resolución de problemas, habilidades que forman parte esencial del pensamiento computacional. Este pensamiento proporciona un marco estructurado para abordar problemas complejos, descomponiéndolos en partes manejables y aplicando soluciones lógicas, lo que es clave para el desarrollo de software y otras áreas tecnológicas. Dominar la creación y comprensión de algoritmos es un primer paso indispensable en la formación y aprendizaje del pensamiento lógico computacional, contribuyendo también a la toma de decisiones informadas y eficientes en diversos campos.

Descripción

Esta actividad tiene como objetivo desarrollar habilidades clave en el análisis, diseño y representación de algoritmos, para iniciarse en la programación y el desarrollo de software. A través de la resolución de problemas matemáticos, se aplican técnicas de descomposición lógica utilizando una descripción clara y ordenada de pasos y decisiones para guiar la solución. Este método permite centrarse en la lógica del proceso sin preocuparse inicialmente por detalles técnicos o sintaxis de un lenguaje específico, facilitando así la comprensión y validación de cada etapa. Cada algoritmo elaborado define con claridad sus entradas, procesos y salidas, promoviendo

un enfoque estructurado del pensamiento lógico computacional. Esto también permitirá detectar errores o redundancias con mayor facilidad. Representar correctamente un algoritmo es el paso previo esencial para su posterior codificación, asegurando soluciones claras, eficientes y escalables. En conjunto, la actividad sienta una base sólida para enfrentar futuros desafíos en programación y en la ingeniería del software.

Justificación

El valor de esta actividad radica en comprender que todo desarrollo de software comienza con un algoritmo bien estructurado. Saber construirlo paso a paso es esencial para resolver problemas de forma lógica y efectiva, incluso antes de escribir una sola línea de código. Trabajar mediante instrucciones claras y ordenadas permite centrarse en la secuencia lógica de acciones sin las barreras de la sintaxis de un lenguaje de programación, estimulando el pensamiento analítico y facilitando la comprensión del problema. Además, actúa como un lenguaje común entre compañeros, promoviendo la colaboración y permitiendo detectar errores desde etapas tempranas. Este enfoque no solo previene fallos futuros en la programación, sino que también fortalece habilidades como la organización, la síntesis y la claridad en la toma de decisiones. En definitiva, diseñar algoritmos de forma estructurada sienta las bases del pensamiento computacional, herramienta imprescindible para cualquier aspirante a programador o ingeniero de software.

Desarrollo

Calculadora de números primos

La calculadora de números primos es un programa que recibe un número entero positivo e indica si ese número es primo o no. Un número primo es aquel que solo se puede dividir exactamente entre 1 y sí mismo. Este programa es útil para introducir conceptos básicos de matemática y programación, como el uso de ciclos y condiciones para realizar comprobaciones, y ayuda a desarrollar el pensamiento lógico necesario para resolver problemas computacionales.

Algoritmo:

1. Iniciar el programa.
2. Solicitar al usuario que ingrese un número entero positivo.
3. Leer y guardar el número ingresado.
4. Verificar si el número es menor o igual a 1:
 - a. Si es menor o igual a 1, mostrar el mensaje: "El número no es primo" y terminar el programa.
 - b. Si es mayor a 1, continuar con el siguiente paso.
5. Inicializar un contador en cero.
6. Para cada número desde 1 hasta el número ingresado, hacer:
 - a. Dividir el número ingresado entre el valor actual del ciclo.
 - b. Si el residuo de la división es cero, aumentar el contador en uno.
7. Terminado el ciclo, verificar el valor del contador:
 - a. Si el contador es igual a 2, mostrar "El número sí es primo".
 - b. Si el contador es mayor a 2, mostrar "El número no es primo".
8. Terminar el programa.

Calculadora para determinar números pares o impares

Este programa solicita al usuario ingresar diez números enteros y para cada uno determina si es par o impar. Para ello utiliza la operación módulo 2, que permite identificar si un número es divisible por 2 sin residuo. La evaluación de números pares e impares es un ejercicio básico que ayuda a practicar el uso de ciclos, condiciones y operaciones aritméticas simples, fortaleciendo el razonamiento lógico y la interacción con el usuario.

Algoritmo:

1. Iniciar el programa.
2. Inicializar un contador de números procesados en 0.
3. Mientras el contador sea menor que 10, repetir:
 - a. Solicitar al usuario que ingrese un número entero.
 - b. Leer el número ingresado.
 - c. Calcular el residuo del número dividido entre 2.
 - d. Si el residuo es 0, imprimir: "El número (número ingresado) es par."
 - e. Si el residuo no es 0, imprimir: "El número (número ingresado) es impar."
 - f. Incrementar el contador en 1.
4. Imprimir el mensaje: "Proceso completado. Se han evaluado 10 números."
5. Terminar el programa.

Calculadora para invertir un número

Este programa solicita al usuario un número entero y devuelve el número invertido, es decir, con sus dígitos en orden reverso. Este tipo de programa es útil para practicar manipulación de números mediante operaciones aritméticas básicas como la división y el módulo, y para fortalecer la lógica de ciclos y acumulación de resultados. Además, representa un buen ejercicio introductorio para el desarrollo de algoritmos orientados a problemas de manipulación numérica.

Algoritmo:

1. Iniciar el programa.
2. Solicitar al usuario que ingrese un número entero.
3. Leer y almacenar el número ingresado en una variable numero.
4. Inicializar una variable numeroInvertido en 0.
5. Asignar a una variable auxiliar el valor de numero.
6. Mientras la variable auxiliar sea mayor que 0, repetir:
 - a. Obtener el último dígito usando la operación módulo 10 ($\text{digito} = \text{auxiliar} \% 10$).
 - b. Multiplicar numeroInvertido por 10 y sumar el dígito obtenido.
 - c. Dividir la variable auxiliar entre 10 usando división entera para eliminar el último dígito ($\text{auxiliar} = \text{auxiliar} // 10$).
7. Cuando el ciclo termine, mostrar el valor de numeroInvertido al usuario.
8. Terminar el programa.

Conclusión

Esta actividad permitió comprender de forma más profunda cómo el diseño de algoritmos es la base sobre la cual se construye todo programa funcional. Resolver problemas matemáticos, como identificar números primos, verificar paridad o invertir cifras fue clave para practicar estructuras lógicas como ciclos, condicionales y operaciones básicas.

Redactar los algoritmos en lenguaje natural, facilita el enfoque en la lógica más que en la forma, algo especialmente útil al comenzar en programación. La organización clara y explicada de cada algoritmo ayudó a unir teoría y práctica, fortaleciendo tanto el pensamiento computacional como la capacidad analítica.

Además de resolver problemas, se desarrollaron habilidades valiosas como la precisión en la redacción técnica y la claridad en la estructuración de ideas, indispensables en el trabajo colaborativo y profesional. En suma, esta experiencia reafirma que dominar la lógica algorítmica no solo mejora el aprendizaje, sino que prepara el camino hacia una programación más sólida y eficiente.

Referencias

Elipse AI. (2024). Algoritmo, lenguaje natural. Las conductas de artefactos y softwares. Blog Elipse AI. <https://blog.elipse.ai/algoritmo-lenguaje-natural.-y-las-conductas-de-artefectos-y-softwares>