

Actividad | #1 | Algoritmos

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Devora

ALUMNO: Mónica Lázaró Mendez

FECHA: 29/07/2025

ÍNDICE

❖ Introducción

❖ Descripción

❖ Justificación

❖ Desarrollo

❖ Conclusión

❖ Referencias

INTRODUCCIÓN

Empecemos con ¿qué es un algoritmo? es un conjunto de instrucciones precisas, finitas, y ordenadas que describen una secuencia de operaciones para resolver un problema o realizar una tarea específica. Piensa en el como una receta de cocina: cada paso es crucial y si se altera el orden o se omite un paso, el resultado final podría ser diferente o incorrecto.

Se debe tener un numero definido de pasos. No puede continuar indefinidamente.

Las instrucciones deben ir ordenadas con secuencia. Son fundamentales en la informática y en muchas otras áreas. Son la base de la programación, permitiendo la creación de software, aplicaciones y sistemas que resuelven problemas complejos, el rendimiento de los sistemas informáticos es crucial.

DESCRIPCIÓN

La empresa MathTech requiere a un ingeniero en desarrollo de software, para crear distintas calculadoras. Aquí realizaremos tres calculadoras de algoritmos. Estos algoritmos están escritos en pseudocodigo, que es una forma de expresar la lógica sin la síntesis escrita de un lenguaje de programación.

La primera será: Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo, si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el siguiente mensaje “El número (número ingresado) no es primo”. Básicamente se encargará de identificar si un número es divisible entre 1 y el mismo.

La segunda será: Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo, si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10 números enteros, se debe determinar cuáles son pares y cuáles son impares.

La tercera será: Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los números al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

JUSTIFICACIÓN

Para el desarrollo de las tres calculadoras propuestas radica en la necesidad de herramientas educativas prácticas y accesibles. Cada calculadora aborda un concepto matemático fundamental de una empresa interactiva y sencilla, facilitando el aprendizaje y la comprensión.

La calculadora “los primos” es una calculadora siempre que permite verificar si un número es primo o no, refuerza este concepto de forma práctica y les permite examinar con diferentes números.

La calculadora “par/impar” aborda un concepto básico de aritmética la paridad de los números.

La calculadora “al revés” es un concepto importante en la comprensión del sistema de numeración decimal y una base para algoritmos más complejos. La intervención de un número requiere un entendimiento de la posición de los dígitos y su valor posicional.

Es una introducción práctica y progresiva a los conceptos matemáticos y a la programación. Nos permite comprender la lógica detrás de cada algoritmo, lo que facilita la transición hacia los conceptos más avanzados. Desarrollando habilidades en la resolución de problemas y el pensamiento computacional.

DESARROLLO

La primera calculadora “Primos”

Paso 1. Obtener el número de entrada

El primer paso es obtener el número entero del usuario. Esto puede hacerse utilizando una función de entrada o leyendo desde un archivo, dependiendo del contexto de tu programa. Llevaremos a este número n .

Paso 2. Verificar si el número es menor o igual a 1

Los números primos son mayores que 1. Si n es menor o igual a 1, no es primo. Devolvemos falso en este caso.

Paso 3. Verificar si el número es 2

El número 2 es el único número primo par. Si n es 2, es primo. Devolveremos verdadero.

Paso 4. Verificar si el número es par y mayor que 2

Si n es par y mayor que 2, no es primo (excepto el 2, ya considerado en el paso anterior). Devolvemos falso.

Paso 5. Verificar divisibilidad desde 3 hasta la raíz cuadrada de n

Este es el paso mas importante. Iteramos a través de los números impares desde 3 hasta la raíz cuadrada de n . Si n es divisible por alguno de estos números, no es primo. Debemos comprobar la divisibilidad solo hasta la raíz cuadrada de n porque si un numero tiene un divisor mayor que su raíz cuadrada, también tendrá un divisor menor que su raíz cuadrada. Para optimizar, podemos incrementar el contador de la iteración en 2 en cada paso ya que solo necesitamos verificar los números impares.

Paso 6. Si ninguna de las condiciones anteriores es verdadera, el número es primo. Si el bucle del paso 5 termina sin encontrar ningún divisor, significa que n es primo. Devolvemos verdadero.

La segunda calculadora ``Par/Impar``

Entrada: una lista de 10 números enteros.

Salida: Una lista indicando si cada número de la lista de entrada es par o impar.

Pasos 1. Inicialización: Crea dos listas vacías: una para almacenar los números pares (pares) y otra para los números impares (impares).

Paso 2. Iteración: Recorre la lista de 10 números de entrada. Para cada número:

- Verificación de paridad: Divide el número entre 2. Si el resto de la división es 0, el número es par; de lo contrario, es impar.
- Almacenamiento: Si el número es par, agrégalo a la lista `pares`. Si es impar, agrégalo a la lista `impares`.

Paso 3. Salida: Imprime las listas `pares` e `impares`, mostrando qué números son pares y cuáles impares.

Este código itera sobre la lista `números`. Para cada número, usa el operador módulo (%) para obtener el resto de la división entre 2. Si el resto es 0, el número se agrega a la lista `pares`; de lo contrario, se agrega a la lista `impares`. Finalmente, se imprimen ambas listas.

La tercera “Al Revés”

Entrada: Un número entero de cuatro dígitos.

Salida: El número con sus dígitos invertidos.

Pasos 1. Conversión a cadena: Convierte el número entero a una cadena de caracteres. Esto permite acceder a cada dígito individualmente.

Paso 2. Inversión de la cadena: Invierte el orden de los caracteres en la cadena. Existen varias maneras de hacer esto; una es usando slicing con un paso negativo.

Paso 3. Conversión a entero: Convierte la cadena invertida de nuevo a un número entero.

Este código primero convierte el número a una cadena usando `str()`. Luego, usa slicing `[::-1]` para invertir la cadena. Finalmente, convierte la cadena invertida de vuelta a un entero usando `int()`.

CONCLUSIÓN

En esta actividad se trabajaron con algoritmos, la empresa Math Tech necesitaba un ingeniero de software para desarrollar tres calculadoras: una para identificar números primos, otra para determinar un conjunto de diez números es pares o impares y la última para invertir los dígitos de un número de cuatro cifras. Cada calculadora requiere un algoritmo específico para cumplir su función.