

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.

Actividad | #2 | Diagrama de flujo

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Mónica Lázaró Mendez

FECHA: 02/07/2025

ÍNDICE

➤ **INTRODUCCIÓN**

➤ **DESCRIPCIÓN**

➤ **JUSTIFICACIÓN**

➤ **DESARROLLO**

➤ **CONCLUSIÓN**

➤ **REFERENCIAS**

INTRODUCCIÓN

La empresa MathTech requiere a un ingeniero en desarrollo de software que sea capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas pública. Con lo cual desarrolla un diagrama de flujo para cada una de nuestras calculadoras.

La calculadora “los primos” tiene como objetivo identificar si su numero ingresado es primo. Con entrada un numero entero. Con su proceso de verificar si el número es divisible únicamente por uno y si mismo.

La salida de mantenerse indicando si el numero es primo o no.

En la segunda calculadora de nombre “par e impar” de igual manera cuenta con su objetivo el cual es determinar si 10 números ingresados son pares o no, su proceso para cada número, verificar si es divisible entre dos y su salida en listar los números ingresados clasificados.

La tercera calculadora de nombre “al revés” de igual manera cuenta con su proceso el cual es invertir un numero de cuatro dígitos, con su entrada de un numero entero de cuatro dígitos, con su proceso de invertir el orden de los dígitos del número, y su salida que es el numero invertido.

Al crear los diagramas de flujo se utilizan símbolos estándar para presentar cada etapa del proceso (inicio, fin, entrada/salida, proceso, decisión) para mayor claridad y constancia.

DESCRIPCIÓN

La empresa MathTech requiere a un ingeniero en desarrollo de software que sea capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas públicas, del cual incrementaremos un diagrama de flujo para completar nuestra activada uno que va de la mano esta sería la segunda etapa.

- La primera calculadora “Primos”, y su objetivo será identificar los números primos que se ingresen, “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 calculadora Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, 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.

- El último programa se llamará 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

Esta es la segunda etapa de nuestro trabajo se desarrollará un diagrama de flujo, ya en la primera fase se hizo el algoritmo de nuestras calculadoras.

El objetivo de esta investigación es desarrollar tres programas matemáticos con funcionalidades específicas:

- "Primos": Identificar si un número ingresado es primo, mostrando un mensaje claro que indique su condición (primo o no primo).
- "Par/Impar": Determinar, de 10 números enteros ingresados, cuáles son pares y cuáles son impares, proporcionando una clasificación precisa para cada uno.
- "Al Revés": Invertir un número de 4 dígitos ingresado por el usuario, mostrando el resultado de manera clara y correcta.

Estos programas buscan resolver problemas matemáticos básicos de manera eficiente, brindando respuestas rápidas y precisas al usuario.

DESARROLLO

La primera calculadora “Primos”

Paso 1. Obtener el número de entrada

El primer paso es obtener el numero entero del usuario. Esto puede hacer utilizado 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

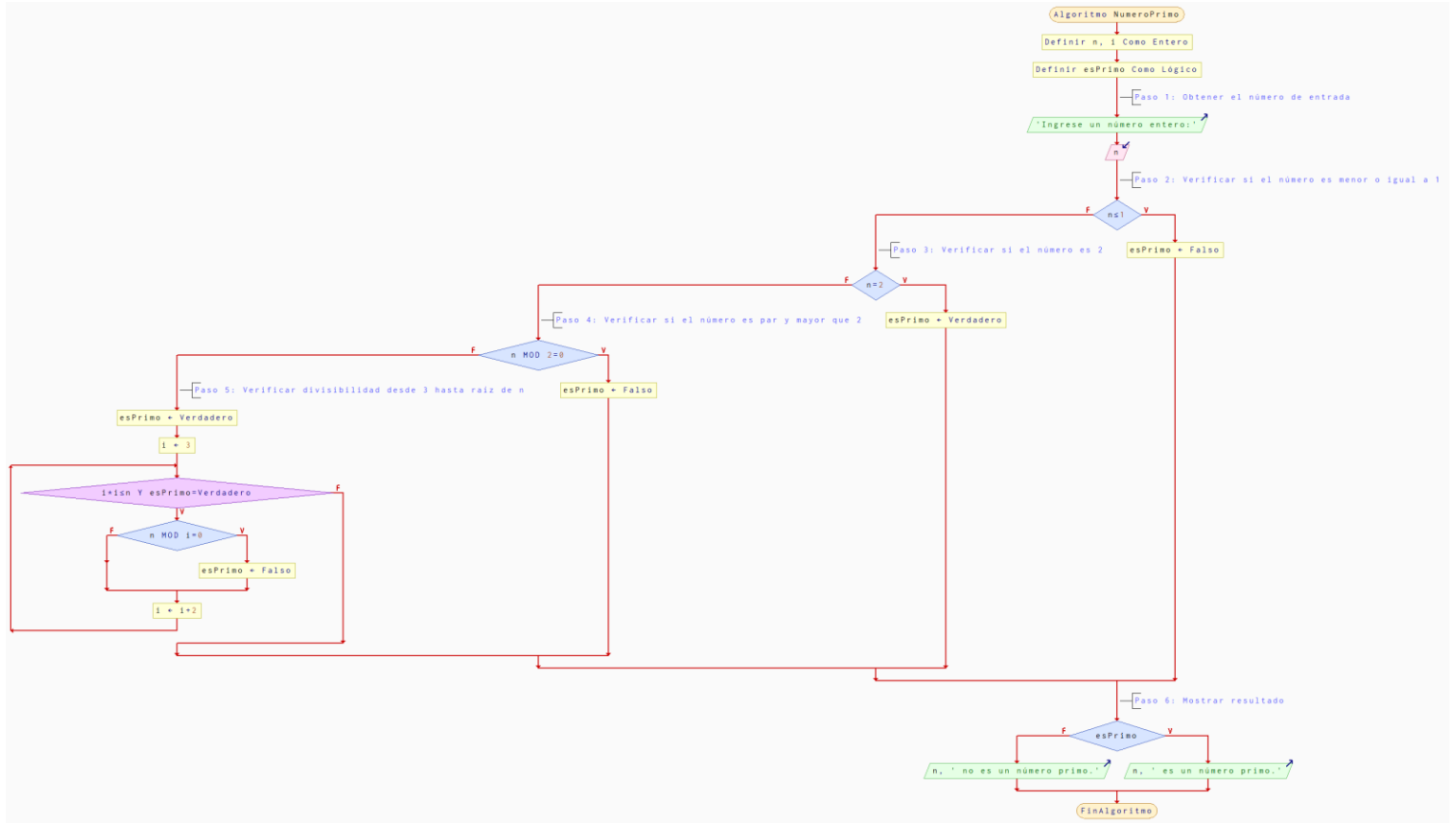
Si n es par y mayor que 2, no es primo (excepto el 2, ya considerado en el paso anterior). Devolvemos falso.

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

Este es el paso más 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.

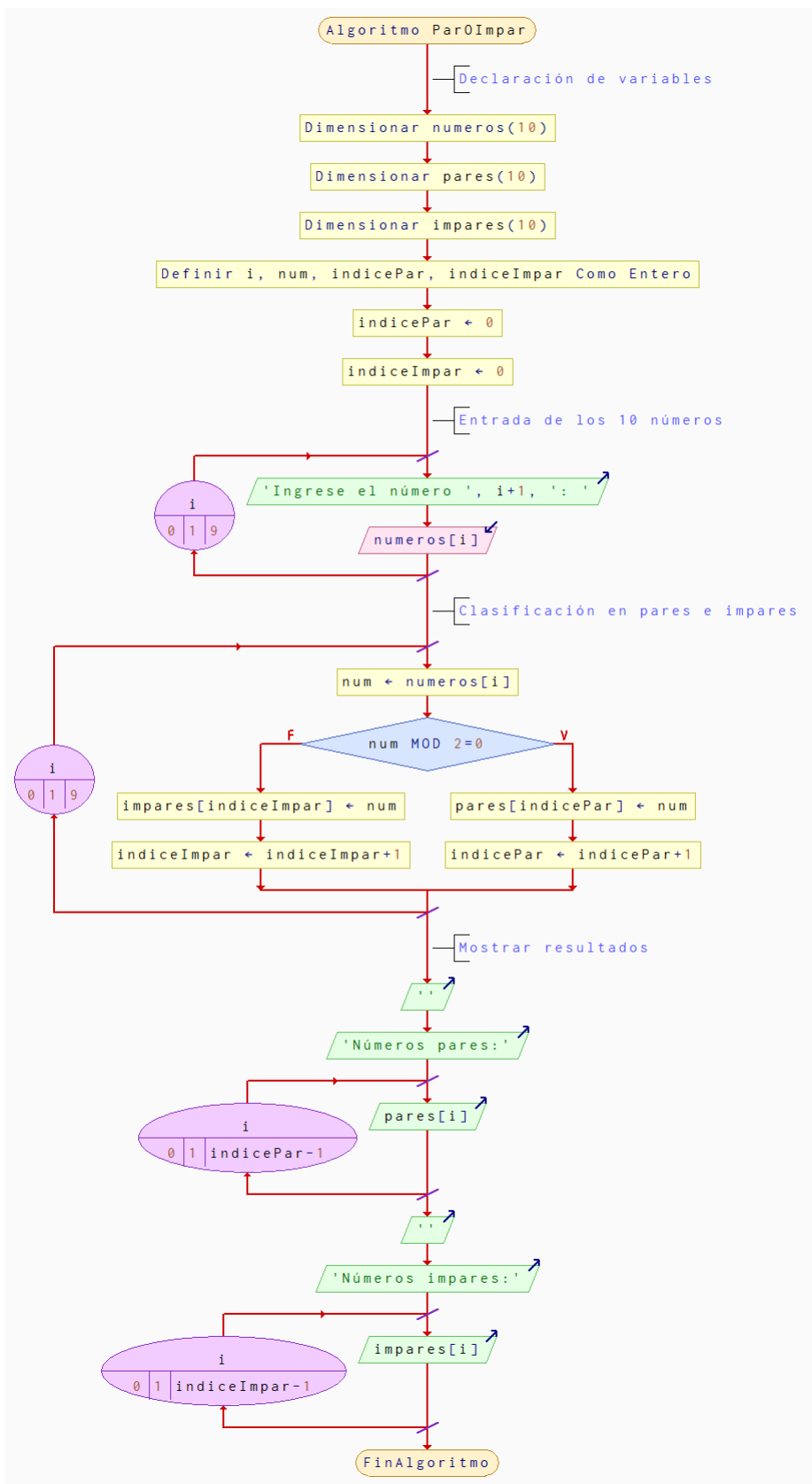
Pasos 1. Inicialización: Crea dos listas vacías: una para almacenar los números pares (pares) y otra para los números impares (impares).

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.

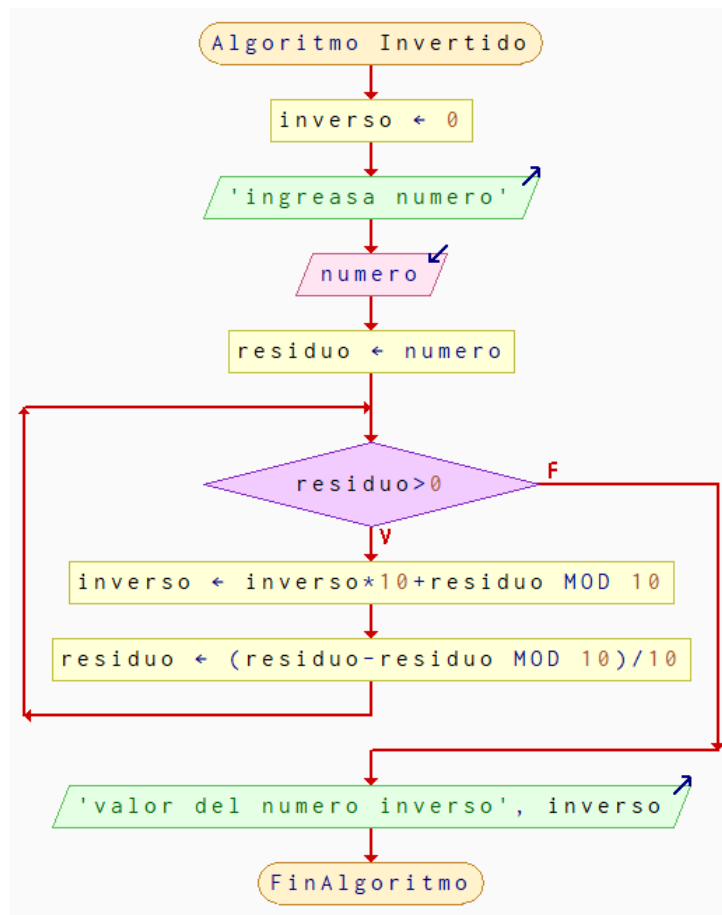
Salida: El número con sus dígitos invertidos.

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

Se realizaron diagramas de flujo, se complemento la actividad con la actividad 1 de algoritmos.

Este conjunto de tres programas básicos en PSeInt tiene como objetivo introducir al usuario en la lógica de programación estructurada mediante ejercicios sencillos pero representativos. Cada calculadora cumple una función específica y refuerza el uso de distintos conceptos fundamentales:

1. Calculadora Primos: Permite al usuario identificar si un número entero ingresado es primo. Refuerza el uso de condiciones, ciclos y operadores aritméticos para evaluar divisibilidad. Además, fomenta la comprensión matemática de lo que significa que un número sea primo.
2. Calculadora Par/Impar: Permite ingresar una lista de 10 números enteros y determina si cada uno es par o impar. Este programa refuerza el uso de arreglos, estructuras repetitivas (Para) y condicionales (Si). También introduce el uso de listas separadas para almacenar resultados diferentes.
3. Calculadora Al Revés: Solicita un número entero de cuatro dígitos y devuelve ese número con sus cifras invertidas. Este ejercicio fortalece el manejo de cadenas (SubCadena, Longitud), bucles decrecientes, y la conversión entre texto y números. También sirve como una introducción práctica al procesamiento de datos tipo texto.

En conjunto, estos tres programas no solo cumplen con su propósito funcional, sino que también ayudan a desarrollar habilidades clave para cualquier principiante en programación: entrada/salida de datos, control de flujo, operadores lógicos y aritméticos, y manipulación de cadenas y números.

REFERENCIAS

PSeInt aplicación

Actividad | 3 | Código en Lenguaje C

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Mónica Lázaró Méndez

FECHA: 04/07/2025

ÍNDICE

➤ **Introducción**

➤ **Descripción**

➤ **Justificación**

➤ **Desarrollo:**

❖ **Codificación**

❖ **Ejecución en el compilador**

➤ **Conclusión**

➤ **Referencias**

Introducción

En el contexto actual de la educación, es fundamental acercar a los estudiantes a herramientas tecnológicas que faciliten el aprendizaje de conceptos matemáticos básicos. Con este propósito, la empresa MathTech ha propuesto el desarrollo de tres programas simples, implementados en lenguaje C, que puedan ser utilizados en escuelas y colegios públicos como parte de su formación matemática y lógica computacional.

El proyecto consiste en la creación de tres calculadoras con funciones específicas:

1. Calculadora Primos, que permite al usuario ingresar un número entero y determina si es un número primo o no.
2. Calculadora Par/Impar, que solicita al usuario el ingreso de 10 números enteros y clasifica cada uno como par o impar.
3. Calculadora Al Revés, que recibe un número de cuatro dígitos y devuelve dicho número con sus cifras invertidas.

Estos programas no solo refuerzan habilidades aritméticas, sino que también sirven como ejercicios introductorios a la lógica de programación estructurada. Cada calculadora fue diseñada para ser sencilla, funcional, y accesible para usuarios con conocimientos básicos en informática, especialmente en entornos educativos.

Descripción

Se tiene como finalidad el desarrollo de tres programas matemáticos en lenguaje C, diseñados para ser implementados como herramientas educativas en instituciones públicas. Estas aplicaciones buscan apoyar el proceso de enseñanza-aprendizaje en el área de matemáticas y lógica computacional, facilitando la comprensión de conceptos básicos mediante una interfaz sencilla y clara.

Cada uno de los programas cumple con una función específica:

- **Primos:** Esta calculadora permite determinar si un número ingresado por el usuario es primo. Para ello, se evalúa si el número es divisible únicamente entre 1 y sí mismo, mostrando un mensaje claro según el resultado.
- **Par/Impar:** En esta calculadora, el usuario ingresa una serie de 10 números enteros. El programa analiza cada número e indica si es par o impar, permitiendo al estudiante identificar patrones de divisibilidad por 2.
- **Al Revés:** Esta herramienta solicita al usuario ingresar un número entero de cuatro dígitos. El programa invierte el orden de sus cifras y presenta el resultado. Este ejercicio estimula la lógica y el pensamiento secuencial.

Los tres programas están codificados utilizando estructuras fundamentales del lenguaje C, como condicionales, bucles, arreglos y manejo de entrada/salida, lo cual los convierte en una base ideal para estudiantes que inician en la programación estructurada.

Justificación

El uso de la tecnología en la educación es cada vez más importante, especialmente en la enseñanza de áreas como matemáticas y lógica, donde la práctica constante y el apoyo visual pueden marcar una gran diferencia en la comprensión de los estudiantes. En este contexto, surge la necesidad de desarrollar herramientas digitales simples, funcionales y educativas que puedan ser implementadas en entornos escolares.

La empresa MathTech, comprometida con la mejora de la calidad educativa, propone el desarrollo de tres calculadoras en lenguaje C, orientadas a fortalecer los conocimientos matemáticos básicos de los estudiantes de nivel básico y medio. Estas herramientas no solo permiten resolver ejercicios de forma interactiva, sino que también introducen a los usuarios en los fundamentos de la programación y el pensamiento lógico.

Implementar estas calculadoras en las escuelas públicas tiene múltiples beneficios:



- Favorece la comprensión de conceptos como números primos, paridad, y valor posicional.
- Estimula el razonamiento lógico y analítico.
- Promueve el aprendizaje activo y tecnológico.
- Brinda a los estudiantes la oportunidad de interactuar con programas hechos en un lenguaje de programación profesional (C).

Desarrollo

Codificación

Ejecución en el compilador

1. "Primos"

principal.c		Producción
<pre>54 // Compilador de C en línea para ejecutar programas C en línea 53 #include <stdio.h> // Librería estándar para entrada y salida 31 3 int principal () { 2 número entero ; // Declaramos una variable entera para guardar el número ingresado por el usuario 5 6 // Solicitar al usuario que introduzca un número 35 printf (" Ingrese un número entero: ") ; 50 scanf (" %d " , & numero) ; // Leemos el número ingresado el usuario 9 // Proceso para verificar si el número es par o impar 41 si (numero % 2 == 0) { 1 // Si el residuo de dividir el número entre 2 es 0, es par 45 printf (" El número %d es PAR. \n " , numero) ; 46 } demás { 43 // Si no, es impar 51 printf (" El número %d es IMPAR. \n " , numero) ; 44 } 42 52 devolver 0 ; // Fin del programa 8 }</pre>	<div></div> <div>Compartir</div> <div>Correr</div>	<p>Ingrese un número entero: 64 El número 64 es PAR.</p> <p>=== Ejecución de código exitosa ===</p>

2. Par/Impar

principal.c



Compartir

Correr

Producción

```

5 // Programa para determinar si un número es par o impar
22
1 #include <stdio.h>
4 #include <stdlib.h>
3
21 int principal ( ) {
2 // Paso 1. Inicialización: Crea dos listas (arrays) para
  almacenar números pares e impares.
7 int pares [ 10 ] , impares [ 10 ] ; // Asumiendo un máximo de
  y 10 impares (podría ser menos)
9 int núm_pares = 0 , núm_impares = 0 ; // Contadores para
  número actual de elementos en cada lista
40 int números [ 10 ] ; // Array para almacenar los 10 números de
  entrada
41 entero yo ;
34
35 printf ( " Ingrese 10 números enteros: \n " ) ;
36 para ( i = 0 ; i < 10 ; i ++ ) {
37     scanf ( " %d " , & numeros [ i ] ) ;
38 }
39
42 // Paso 2. Iteración: Recorre la lista de 10 números de entrada
  .
43 para ( i = 0 ; i < 10 ; i ++ ) {
46     //- Verificación de paridad: Divide el número entre 2. Si el

```

Ingrese 10 números enteros:

4
8
9
11
113
135
105
151
45
37

Números pares:

4 8

Números impares:

9 11 113 135 105 151 45 37

=== Ejecución de código exitosa ===

principal.c



Compartir

Correr

Producción

```

38     }
39
42     // Paso 2. Iteración: Recorre la lista de 10 números de entrada
    .
43     para ( i = 0 ; i < 10 ; i ++ ) {
44         //- Verificación de paridad: Divide el número entre 2. Si el
           resto de la división es 0, el número es par; de lo
           Por el contrario, es impar.
45         si ( numeros [ i ] % 2 == 0 ) {
22             //- Almacenamiento: Si el número es par, agrégalo a la
               lista de pares.
46             pares [ num_pares ] = numeros [ i ] ;
47             num_pares ++ ;
21     } demás {
48         // Si es impar, agrégalo a la lista de impares.
49         impares [ num_impares ] = numeros [ i ] ;
50         num_impares ++ ;
10     }
7     }
8
9     // Paso 3. Salida: Imprime las listas de pares e impares,
           mostrando qué números son pares y cuáles impares.
11     printf ( " \nNúmeros pares: \n " ) ;
34     para ( i = 0 ; i < num_pares ; i ++ ) {
35         printf ( " %d " , pares [ i ] ) .

```

▲ Ingrese 10 números enteros:

4

8

9

11

113

135

105

151

45

37

Números pares:

4 8

Números impares:

9 11 113 135 105 151 45 37

=== Ejecución de código exitosa ===

principal.c



Compartir

Correr

Producción

```

1     pares [ num_pares ] = numeros [ i ] ;
3     num_pares ++ ;
21  } demás {
4     // Si es impar, agrégalo a la lista de impares.
5     impares [ num_impares ] = numeros [ i ] ;
2   num_impares ++ ;
10  }
7  }
8
9  // Paso 3. Salida: Imprime las listas de pares e impares,
    mostrando qué números son pares y cuáles impares.
11  printf ( " \nNúmeros pares: \n " ) ;
34  para ( i = 0 ; i < num_pares ; i ++ ) {
35      printf ( " %d " , pares [ i ] ) ;
36  }
37  printf ( " \n " ) ;
38
39  printf ( " Números impares: \n " ) ;
41  para ( i = 0 ; i < num_impares ; i ++ ) {
40      printf ( " %d " , impares [ i ] ) ;
43  }
42  printf ( " \n " ) ;
44
45  devuelve 0 ;
6  }
```

Ingrese 10 números enteros:

4

8

9

11

113

135

105

151

45

37

Números pares:

4 8

Números impares:

9 11 113 135 105 151 45 37

=== Ejecución de código exitosa ===

4. Al Revés

Programiz Compilador de C en línea

principal.c



Compartir

Correr

Producción

```
35 #include <stdio.h>
40 #include <cadena.h>
36 #include <stdlib.h>
37
38 int principal ( ) {
41     // Paso 1: Declaración de variables
39     int numero ;
22     char cadena [ 5 ] ; // Suficiente para 4 dígitos + el carácter nulo '\0'
6     char cadena_invertida [ 5 ] ;
43     int numero_invertido ;
42     entero i , j ;
45
44     // Paso 2: Entrada del número
1     printf ( " Ingrese un número entero de cuatro dígitos: " ) ;
3     scanf ( " %d " , & numero ) ;
21
4     // Paso 3: Conversión a cadena
5     sprintf ( cadena , " %d " , numero ) ; // Convierte el entero a cadena
2
10     // Paso 4: Inversión de la cadena
7     int longitud = strlen ( cadena ) ;
8     para ( i = longitud - 1 , j = 0 ; i >= 0 ; i -- , j ++ )
9         cadena_invertida [ j ] = cadena [ i ] ;
```

Ingrese un número entero de cuatro dígitos: 7631
El número con sus dígitos invertidos es: 1367

=== Ejecución de código exitosa ===

Activar Windows
Ve a Configuración

Programiz

Compilador de C en línea

principal.c

Compartir

Correr

```
1 printf ( " Ingrese un número entero de cuatro dígitos: " );
3 scanf ( " %d " , & numero );
21
4 // Paso 3: Conversión a cadena
5 sprintf ( cadena , " %d " , numero ); // Convierte el entero
  cadena
2
10 // Paso 4: Inversión de la cadena
7 int longitud = strlen ( cadena );
8 para ( i = longitud - 1 , j = 0 ; i >= 0 ; i -- , j
9   cadena_invertida [ j ] = cadena [ i ] ;
11 }
34 cadena_invertida [ longitud ] = '\0' ; // Terminar la cadena
   invertida con el carácter nulo
35
36 // Paso 5: Conversión a entero
37 numero_invertido = atoi ( cadena_invertida ) ; // Convierte l
   cadena invertida a un entero
38
39 // Paso 6: Salida del resultado
41 printf ( " El número con sus dígitos invertidos es: %d\n " ,
   numero_invertido ) ;
40
43 devuelve 0 ;
42 }
```

Producción

Ingrese un número entero de cuatro dígitos: 7631
El número con sus dígitos invertidos es: 1367

=== Ejecución de código exitosa ===

Activar Win
Ve a Configurac

Conclusión

En resumen, estos tres programas cubren una variedad d conceptos fundamentales en la programación c, incluyendo entrada/salida, operaciones aritméticas, estructuras de control de flujo, manipulación de cadenas. Cada programa ilustra como estos conceptos pueden combinarse para resolver problemas específicos. Considerando los tres programas o calculadoras.

La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, “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 calculadora se llamará 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.

- El último programa se llamará 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.

Referencias

Se usaron las aplicaciones

<https://www.programiz.com/c-programming/online-compiler/>

https://www.onlinegdb.com/online_c_compiler