

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

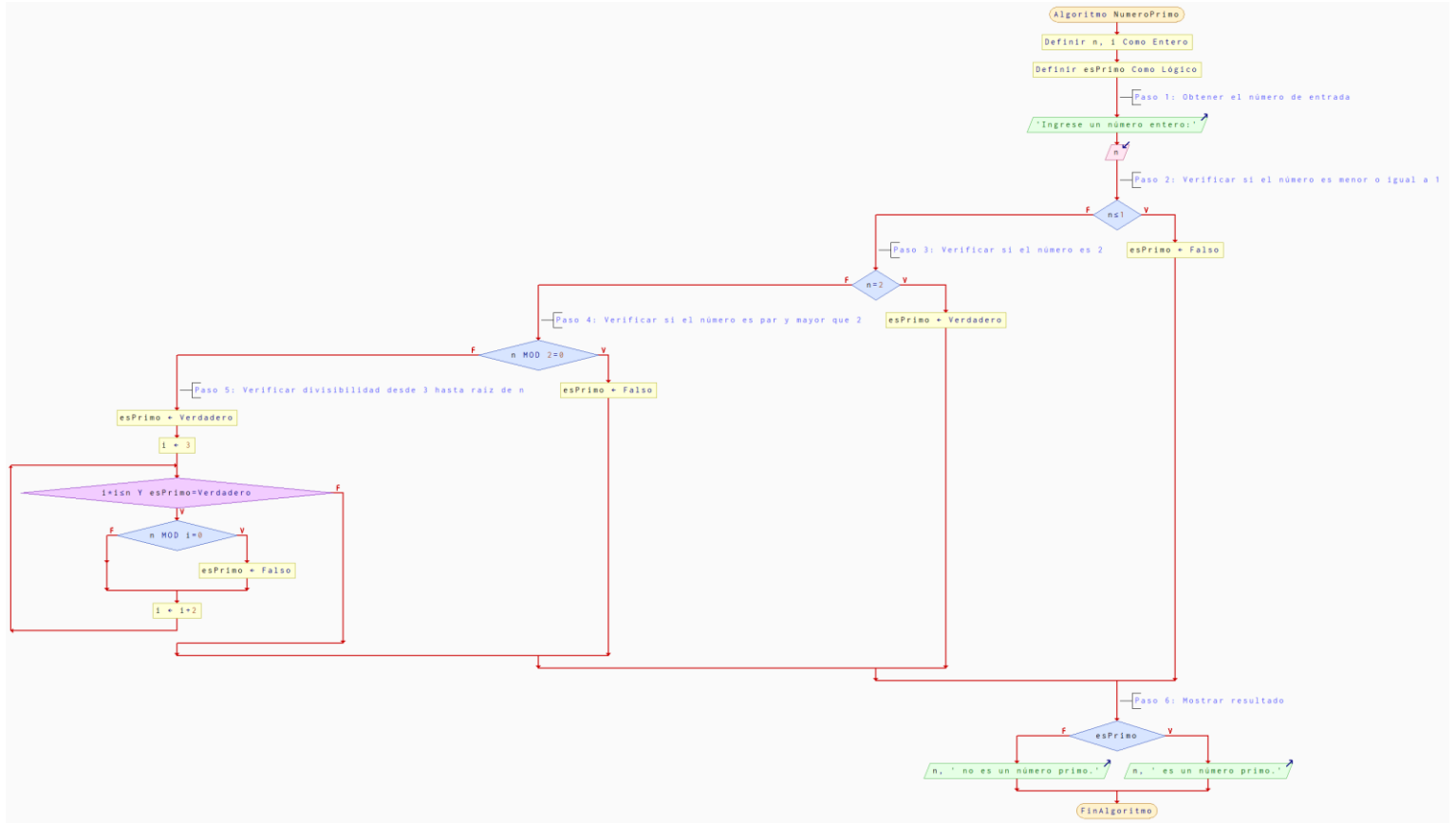
Si n es par y mayor que 2, no es primo (excepto el 2, ya considerado en el paso anterior). Devolvemos falso.

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.

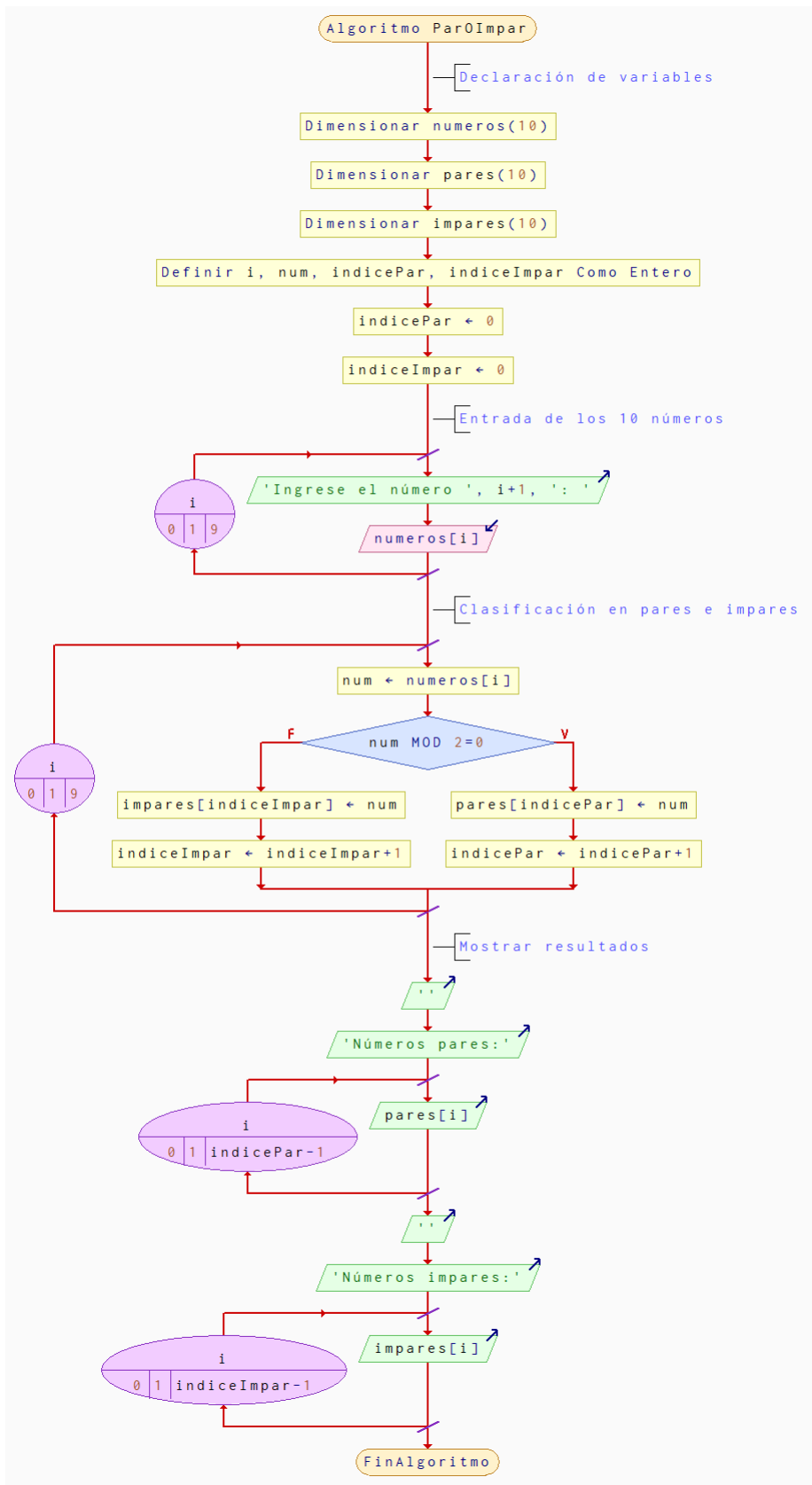
Pasos 1. Inicialización: Crea dos listas vacías: una para almacenar los números pares (pares) y otra para los números impares (impares).

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.

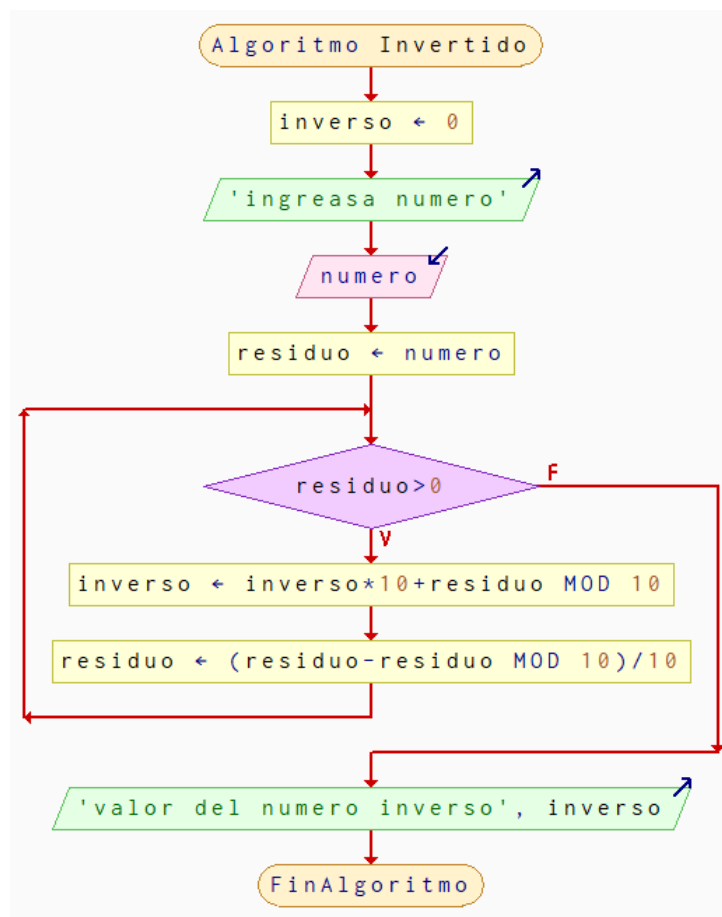
Salida: El número con sus dígitos invertidos.

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