

Actividad | #2 | Diagramas de flujo

Introducción al desarrollo de software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: María Fernanda Madrazo Nasuno

FECHA: 31 de Marzo del 2025

Índice

<i>Introducción</i>	<i>3</i>
<i>Descripción</i>	<i>4</i>
<i>Justificación</i>	<i>5</i>
<i>Desarrollo (Primos)</i>	<i>6</i>
<i>Desarrollo (Par/Impar)</i>	<i>6</i>
<i>Desarrollo (Al Revés)</i>	<i>7</i>
<i>Conclusión</i>	<i>9</i>
<i>Referencias</i>	<i>10</i>

Introducción

Los diagramas de flujo son representaciones gráficas que simplifican la comprensión de procesos complejos mediante símbolos y flechas. En el contexto educativo, son especialmente útiles para enseñar algoritmos, ya que permiten visualizar cada paso de manera ordenada. Por ejemplo, para los programas Primos, Par/Impar y Al Revés, un diagrama de flujo muestra cómo se ingresan los datos, cómo se toman decisiones (como verificar si un número es divisible) y cómo se llega al resultado final. Estos diagramas no solo son herramientas para programadores, sino también recursos pedagógicos que ayudan a estudiantes a entender la lógica detrás de las operaciones matemáticas. Al ver un proceso dividido en etapas claras, los alumnos pueden identificar errores, seguir secuencias y reforzar su pensamiento crítico. Además, los diagramas de flujo son universales: no requieren conocimientos avanzados de programación para interpretarse, lo que los hace ideales para aulas con diversidad de habilidades. En resumen, son puentes entre la teoría abstracta y la práctica, facilitando el aprendizaje interactivo y la planificación eficiente de soluciones tecnológicas.

Descripción

El desarrollo de los diagramas de flujo para los tres algoritmos sigue una estructura lógica. Para Primos, el diagrama inicia con la entrada de un número, luego evalúa si es menor o igual a 1 (descarta primos) o igual a 2 (confirma primo). Si es mayor, un bucle revisa divisores desde 2 hasta la raíz cuadrada del número, usando rombos para decisiones y rectángulos para operaciones. Si se encuentra un divisor, el flujo se desvía a "No es primo"; de lo contrario, concluye en "Es primo". En Par/Impar, el diagrama repite un ciclo 10 veces: cada iteración pide un número, aplica la operación módulo 2 y, según el resto, muestra "Par" o "Impar". Finalmente, para Al Revés, el flujo incluye una validación inicial para asegurar que el número tenga 4 dígitos. Luego, extrae cada dígito mediante divisiones y residuos, reordena los valores y los combina en orden inverso. Cada paso se representa con símbolos específicos: óvalos para inicio/fin, rectángulos para cálculos y rombos para validaciones, conectados por flechas que guían el flujo de principio a fin.

Justificación

La utilidad de los diagramas de flujo radica en su capacidad para hacer visible lo abstracto. En algoritmos como Primos, donde la lógica depende de iteraciones y condicionales, el diagrama evita que los estudiantes se pierdan en explicaciones textuales. Por ejemplo, al ver un rombo con la pregunta " $\text{¿Número \% i} == 0?$ ", entienden que allí se decide si un número no es primo. Para Par/Impar, el bucle se representa con una flecha que retrocede, mostrando claramente cómo se repiten las 10 entradas. Esto es crucial en aulas donde los alumnos aprenden a base de ejemplos visuales. Además, los diagramas previenen errores comunes: en Al Revés, si un niño ingresa un número de 3 dígitos, el flujo muestra dónde debe añadirse una validación. También son inclusivos, ya que personas con dificultades para seguir código escrito pueden entender mejor las imágenes. En un mundo donde la tecnología avanza rápidamente, enseñar con diagramas de flujo prepara a los estudiantes para pensar de manera estructurada, una habilidad esencial en carreras técnicas y científicas.

Desarrollo (Primos)

La primera calculadora deberá de llevar por nombre 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.

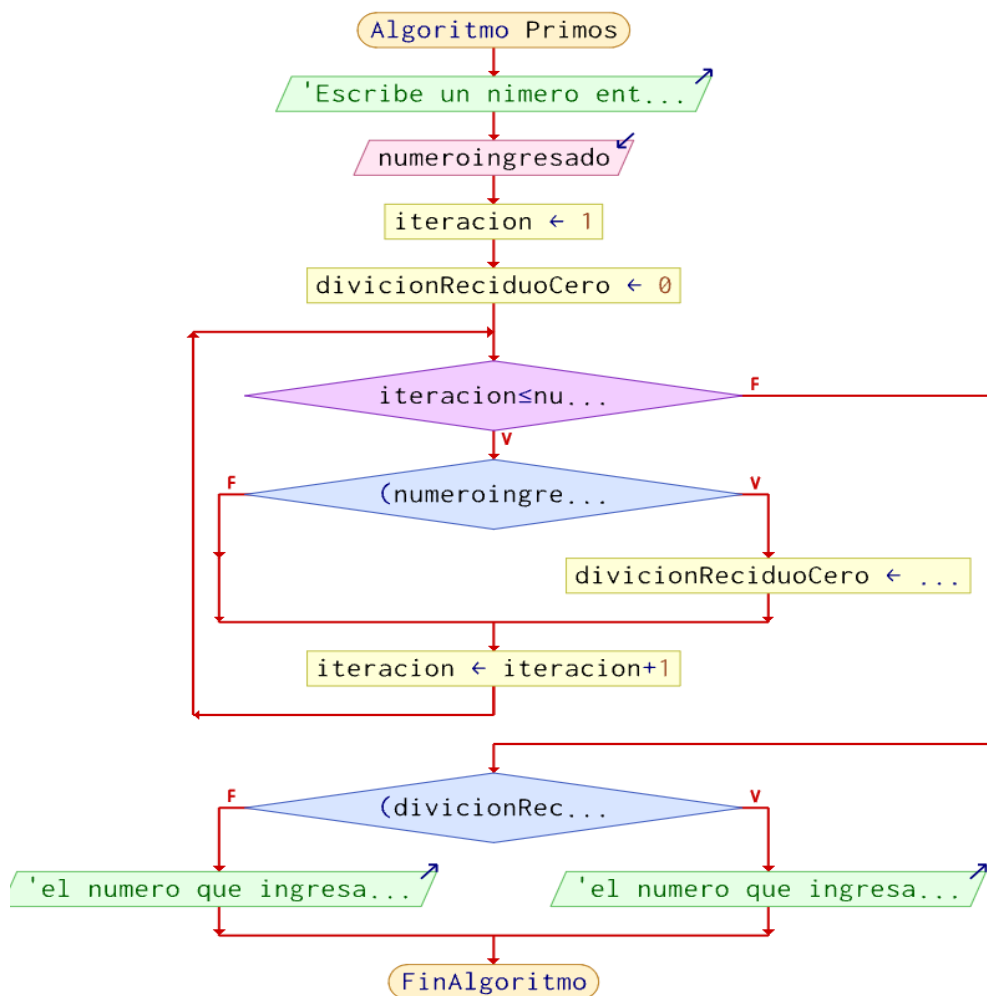
Objetivo: Determinar si un número es primo (solo divisible entre 1 y sí mismo).

Lógica detallada:

- Si el número es 0, 1 o negativo, no es primo.
- Si es 2, es primo (es el único número par primo).
- Para números mayores a 2:

Si se encuentra un divisor, no es primo.

Si no encuentra divisores, es primo.



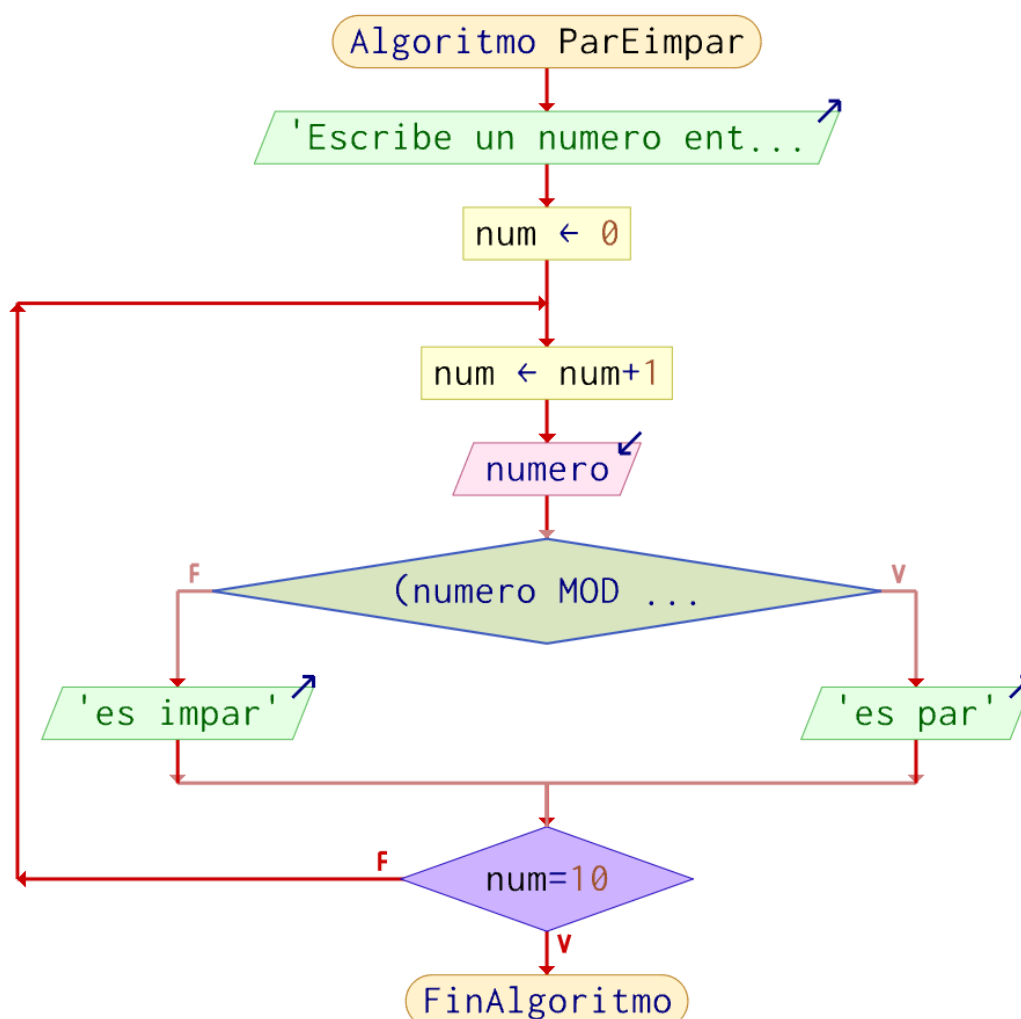
Desarrollo (Par/Impar)

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.

Objetivo: Clasificar 10 números como pares o impares.

Para cada número:

- Divídelo entre 2.
- Si no sobra nada (resto = 0), es par.
- Si sobra 1 (resto = 1), es impar.



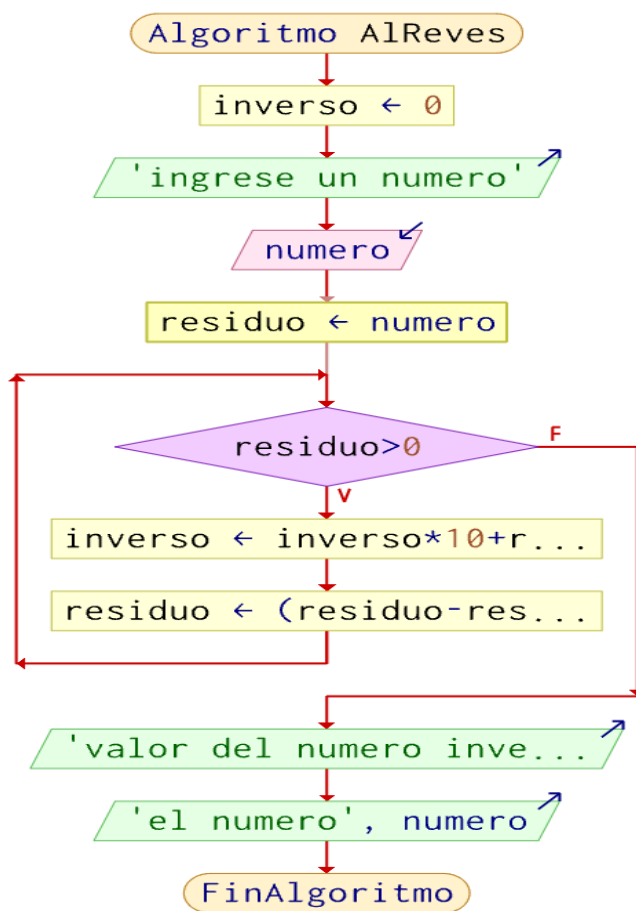
Desarrollo (Al Revés)

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.

Objetivo: Invertir un número de 4 dígitos.

Separa cada dígito:

- Primer dígito: Miles.
- Segundo dígito: Centenas.
- Tercer dígito: Decenas.
- Cuarto dígito: Unidades.
- Reordénalos al revés: Unidades \rightarrow Decenas \rightarrow Centenas \rightarrow Miles.
- Junta los dígitos invertidos.



Conclusión

Los diagramas de flujo son más que herramientas técnicas: son aliados en la educación. Para los algoritmos Primos, Par/Impar y Al Revés, estos diagramas no solo explican "cómo funciona" un programa, sino que también enseñan "por qué funciona". Al desglosar procesos en pasos visuales, los estudiantes entienden que las matemáticas y la programación se basan en reglas claras y secuenciales. Por ejemplo, un niño que ve cómo se invierte 7631 a 1367 mediante flechas y símbolos, comprende que la informática no es magia, sino lógica aplicada. Además, estos recursos fomentan la autonomía: los alumnos pueden crear sus propios diagramas para resolver problemas, practicando antes de escribir código. Finalmente, estas herramientas son un ejemplo de cómo soluciones simples pueden tener un gran impacto. No requieren equipos costosos ni conocimientos avanzados, lo que las hace ideales para cualquier aula. Más allá de ser útiles, inspiran curiosidad: un niño que juega a invertir números hoy podría mañana interesarse en programación o ingeniería. Así, estas herramientas no solo enseñan matemáticas, sino que también abren puertas al futuro.

Referencias

Novara, P. (2023). PSeInt (Versión 2023) [Software]. Universidad Nacional de Córdoba. <https://pseint.sourceforge.net/>

Relevancia: Se utilizó el entorno PSeInt para diseñar, probar y validar los algoritmos en pseudocódigo y en la realización del diagrama de flujo.