



Actividad | 2 | Diagramas de Flujo

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora.

ALUMNO: Rodrigo Flores Vázquez.

FECHA: 23 de noviembre 2024

Índice

1. Introducción.....	3
2. Descripción.....	3
3. Justificación.....	4
4. Desarrollo.....	5
4.1. Diagrama números primos.....	5
4.2. Diagrama número impar e impar.....	6
4.3. Diagrama números invertidos.....	7
5. Conclusión.....	8
6. Referencias.....	9

1. Introducción.

En esta segunda actividad desarrollaremos el algoritmo de los tres programas que se piden en nuestra unidad, como primer paso, seremos capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas públicas,

La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen. La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o 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.

Para ello para realizar la actividad asignada, es muy importantes seguir los pasos para la elaboración de dichos programas: Para cada problema matemático, se deberá:

1. Realizar el algoritmo de cada programa.
2. Realizar su diagrama de flujo por cada uno.
3. Codificarlo finalmente en lenguaje C.

Anteriormente ya se ha realizado el algoritmo de cada programa, ahora se deberá plasmarlo en un diagrama de flujo para comprenderlo de mejor manera, es por ello que en esta actividad también aprenderemos que es un diagrama de flujo, para que sirve y cuales son sus características. Asimismo, en esta segunda actividad aprenderemos a distinguir entre las variables y las constantes. De la misma manera, conoceremos los distintos tipos de datos que existen, como el lenguaje C.

2. Descripción.

En el contexto de nuestro curso de Introducción al Desarrollo de Software, la manipulación de datos como lo vimos en nuestras unidades anteriores, es un tema fundamental porque constituye una de las tareas más comunes en el desarrollo de software, sin importar el tipo de aplicación que se esté construyendo. En términos generales, la manipulación de datos implica recoger, transformar, almacenar y utilizar información en un programa de manera efectiva y eficiente.

Asimismo, los diagramas de flujo y el lenguaje de programación C como lo hemos visto en las unidades 1, 2 3, son herramientas fundamentales en la enseñanza de la programación, ya que ayudan a representar y estructurar el pensamiento lógico detrás de un algoritmo o proceso antes de implementar dicho algoritmo en un lenguaje de programación y es lo que haremos en nuestra actividad.

Puedo destacar que lo que aprenderemos son temas importantes porque involucra las habilidades básicas y avanzadas que todo programador necesita para trabajar con información de manera efectiva. No solo implica entender cómo se gestionan los datos dentro del código, sino también cómo interactúan con sistemas externos y cómo se pueden procesar de manera eficiente para satisfacer los requerimientos de una aplicación, todo esto nos ayudara para desarrollar software funcional y robusto, que es la base de cualquier carrera en desarrollo de software como lo estamos haciendo.

3. Justificación.

En esta actividad los diagramas de flujo son una representación gráfica de un algoritmo o proceso, usan símbolos específicos para representar diferentes tipos de operaciones y decisiones. Son útiles para visualizar cómo fluye el control dentro de un programa y cómo interactúan sus diferentes partes.

Elementos básicos de un diagrama de flujo:

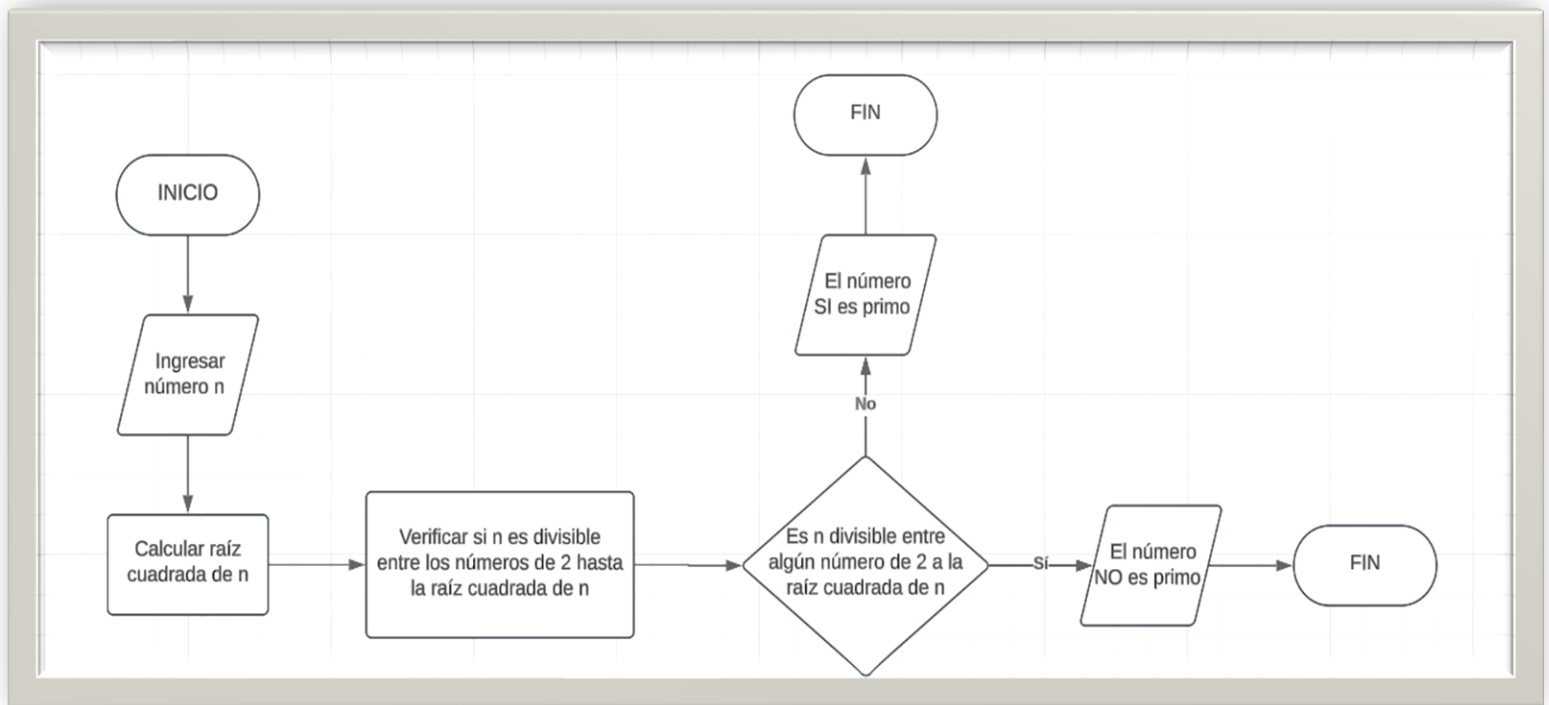
- Óvalo (inicio y fin): Representa el inicio y el fin del proceso.
- Rectángulo (proceso): Representa una operación o instrucción, como asignar un valor a una variable.
- Rombo (decisión): Representa una condición lógica (si/entonces). El flujo se bifurca según si la condición es verdadera o falsa.
- Paralelogramo (entrada/salida): Representa la entrada o salida de datos, como leer un valor del usuario o mostrar un resultado en pantalla.
- Flechas: Indican la dirección del flujo del proceso.

Una vez que tenemos el diagrama de flujo, podemos traducirlo a un programa en el lenguaje de programación C, como lo haremos en el desarrollo de nuestro trabajo. El diagrama de flujo sin duda nos ayuda a comprender la lógica del programa antes de escribir código. Al tener el diagrama, uno puede ver claramente el flujo del proceso y cómo deben organizarse las operaciones. Una vez comprendido el tema, se puede pasar a traducir ese flujo a código al lenguaje C y en nuestro trabajo lo desarrollaremos.

4. Desarrollo.

- Realizar los diagramas de flujo por cada ejercicio solicitado (3 en total). Guardar como imagen los diagramas y agregarlos al presente documento. Cada diagrama debe ir en su respectiva sección conforme al índice y añadir una explicación para cada diagrama correspondiente, donde se especifique la lógica que se le ha dado a cada elemento utilizado, con el fin de comprender el proceso.

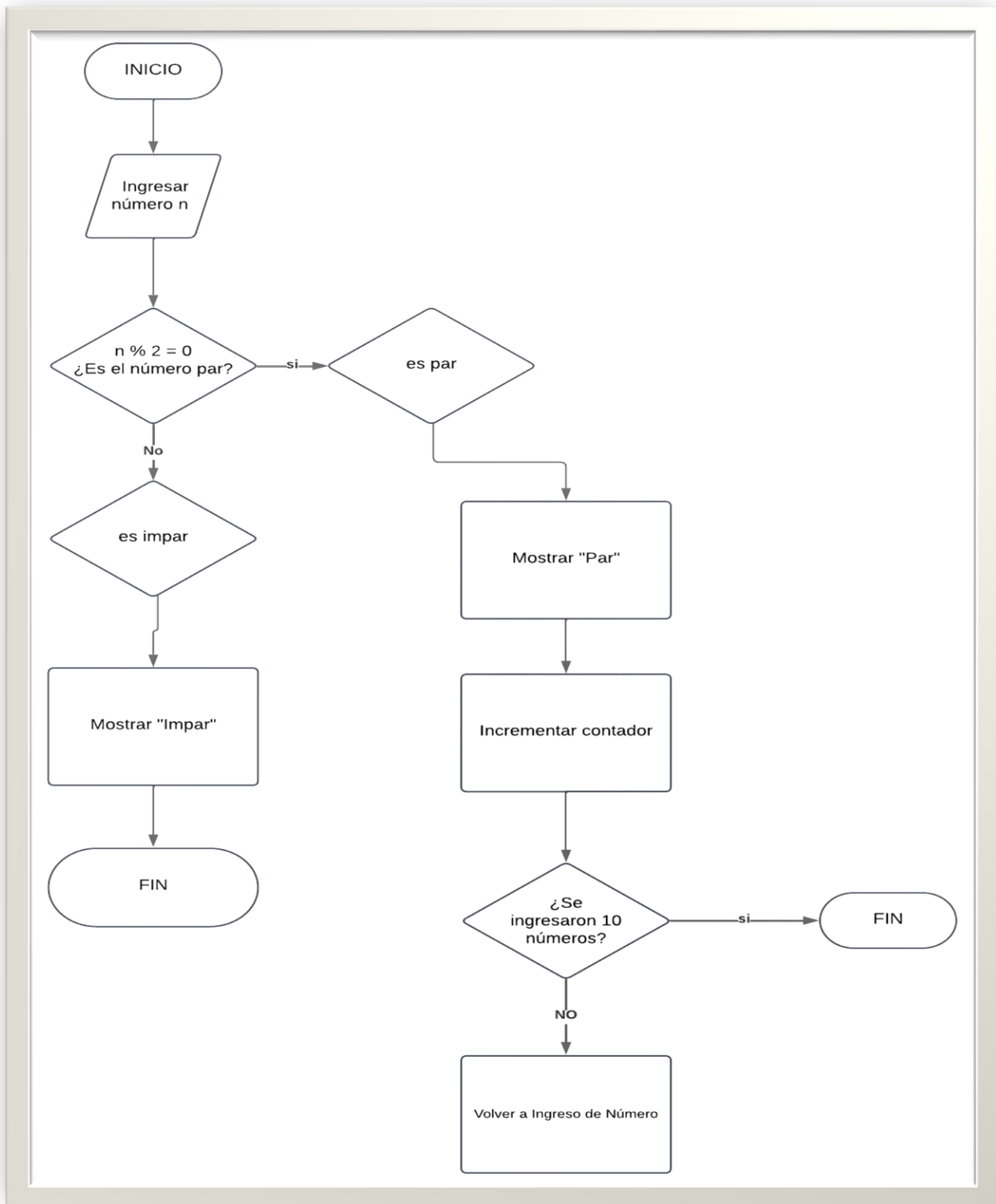
4.1. Diagrama números primos.



Explicación del Diagrama:

- **Inicio:** El programa comienza.
- **Ingreso del número (n):** El usuario debe ingresar un número.
- **Comprobación de si el número es mayor que 1:** Si el número es menor o igual a 1, el flujo va a la salida que indica que el número no es primo.
- **Comprobación de divisibilidad:**
 - Se verifica si el número es divisible por cualquier número entre 2 y la raíz cuadrada del número. Si es divisible, el número no es primo.
 - Si el número no es divisible por ningún número en este rango, se considera primo.
- **Resultado:** El sistema imprime si el número es primo o no.
- **Fin:** El proceso finaliza.

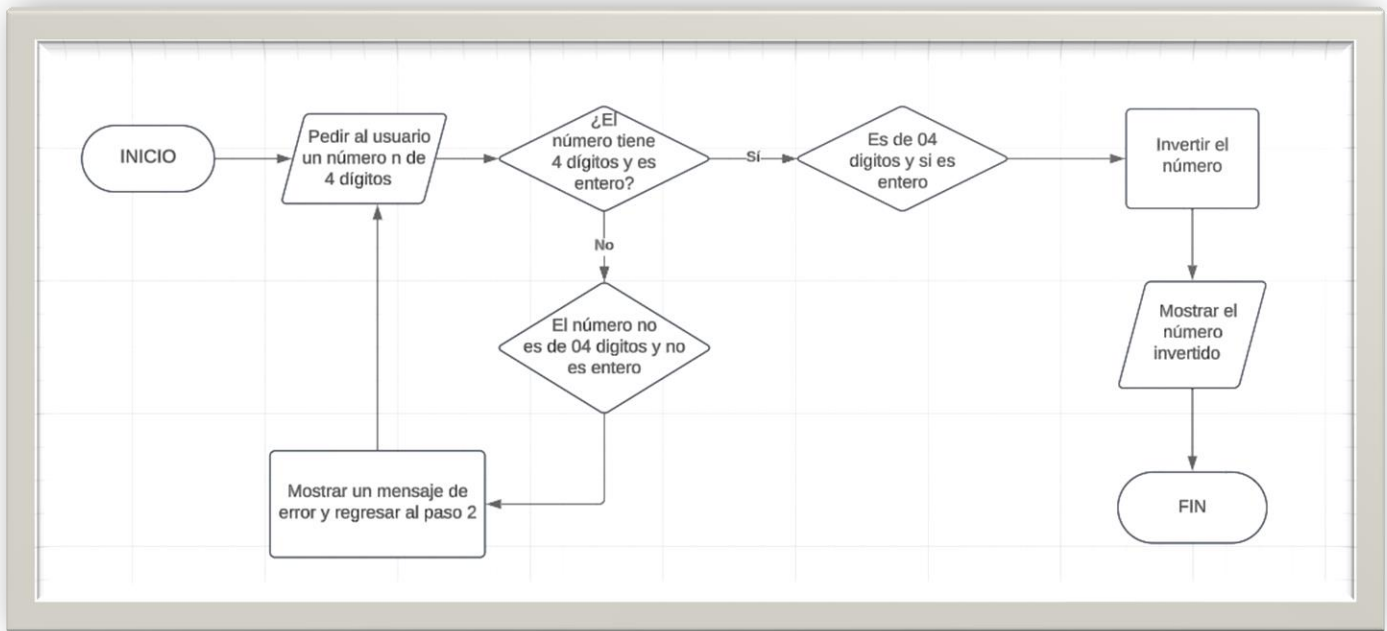
4.2. Diagrama número impar e impar.



Explicación.

- **INICIO:** El programa comienza.
- **Ingreso de Número:** El programa pide al usuario que ingrese un número.
- **Verificación de Par/Impar:** El número ingresado se divide entre 2. Si el residuo de la división ($n \% 2$) es 0, es un número par; de lo contrario, es impar.
- **Mostrar Resultado:** Se indica si el número es par o impar.
- **Contador:** Se incrementa el contador para llevar el control de cuántos números se han ingresado.
- **Repetición o Fin:** Si el contador es menor que 10, el flujo regresa al paso de ingreso de número. Si ya se ingresaron los 10 números, el proceso termina.

4.3. Diagrama números invertidos.



Explicación.

- **Inicio:** El programa empieza.
- **Entrada del número:** Se solicita al usuario que ingrese un número de 4 dígitos.
- **Validación:** El programa verifica que el número ingresado:
 - Tiene exactamente 4 dígitos.
 - Es un número entero (no tiene letras ni símbolos).
- **Proceso de inversión:** Si la validación es correcta, el programa invierte el número.
- **Mostrar el resultado:** Se muestra el número invertido.
- **Fin:** El programa termina.

5. Conclusión.

Durante esta unidad y trabajo pudimos conocer y comprender los diagramas de flujo. Indicamos que los diagramas de flujo son una herramienta útil para diseñar algoritmos y visualizar cómo deben fluir los datos y las decisiones en un programa. También entendimos que el lenguaje C es un lenguaje estructurado que permite implementar esa lógica de manera eficiente, y su uso de estructuras condicionales, ciclos, y operaciones matemáticas lo hace ideal para traducir los diagramas de flujo a código.

Aprendemos y poder por obra los diagramas de flujo junto con la programación C ayuda a entender la relación entre la planificación del programa a través de diagramas de flujo y su implementación en código, lo que fortalece las habilidades de resolución de problemas en la programación. De la misma manera, vimos la forma de declararlos y emplearlos en programas.

A su vez, conocimos los tipos de datos que hay principalmente en lenguaje C, así como la función de cada uno de ellos y la forma en que se utilizan para desarrollar nuestra actividad. En definitiva, es muy importante que conozcamos esta información, ya que nos ayudará a crear programas de manera eficaz y rápida. Por consiguiente, seremos capaz de resolver problemas computacionales de todo tipo en cualquier organización.

6. Referencias.

Academia Global (s.f.). Introducción al desarrollo de Software. Unidad 1 “Resolución de problemas computacionales”

Academia Global (s.f.). Introducción al desarrollo de Software. Unidad 2 “Manipulación de datos”

Diego Moisset de Espanes. (2020, 31 mayo). *Codificación del diagrama de flujo en el lenguaje C* [Video]. YouTube. <https://www.youtube.com/watch?v=rqcwNXtjgJA>

Moisset, D. (s. f.). *Codificación del diagrama de flujo en C++*. <https://www.tutorialesprogramacionya.com/cmasmasya/detalleconcepto.php?codigo=130&punto=3&inicio=0>

Qué es un diagrama de flujo. (s. f.). Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo>

Diagrama de flujo - Programación en C. (s. f.). Solución Ingenieril. https://solucioningenieril.com/programacion_en_c/diagrama_de_flujo#google_vignette

Lucid visual collaboration suite: Log in. (s. f.). https://lucid.app/documents#/home?folder_id=recent

Link a Github.

<https://github.com/Rodrigofloresgs/Introducci-n-al-Desarrollo-de-Sotfware>