



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL DE ESTUDIOS SUPERIORES UNIDAD JURIQUILLA (ENES
JURIQUILLA)**

Ing. Aeroespacial Segundo semestre

Santiago de Querétaro, Qro. 20 de febrero del 2025

**Trabajo:
Práctica 3
“Diagramas de flujo”**

**Materia:
Fundamentos de Programación**

**Presenta:
Serrano Araujo Matías, No. de cuenta: 425030550
Reyes Vargas Edgar, No de cuenta: 322335080**

**Docente:
Ing. Andres David Flores Ferro**

Fecha de revisión:

01 / 03 / 2025

Observaciones:

- Introducción solo con contenido de clase, se agrega i algo breve de investigación adicional a lo mostrado en clase.
 - Agregar dato de tipo de archivo consultado en referencias [Web,PDF, Libro, etc]
-

Calificación:

Actividad en clase 25 de 25 pts

Protocolo de práctica	73 de 75 pts
Portada	10 de 10 pts
Introducción	13 de 15 pts
Desarrollo	25 de 25 pts
Conclusiones	15 de 15 pts
Referencias	10 de 10 pts

CALIFICACIÓN TOTAL Entrega 98 pts

INTRODUCCIÓN

En esta práctica aplicaremos los conocimientos previos de solución de problemas y algoritmos para poder hacer diagramas de flujo en los que resolveremos los problemas planteados en la anterior práctica, los cuales son calcular la distancia entre dos puntos y sacar la media y moda de una cantidad de números. Un diagrama de flujo es la representación gráfica de un proceso, es decir, muestra gráficamente el flujo de acciones a seguir para cumplir con una tarea específica, ósea es la representación gráfica de un algoritmo y una correcta y óptima construcción adecuada con los elementos correctos es necesario para poder hacer que nuestros diagramas de flujo funcionen.

A la hora de hacer diagramas de flujos nos encontramos con diversas formas y figuras en las que cada una representa una cosa y nos permiten estructurar nuestro diagrama de flujo. Un diagrama de flujo debe tener un inicio y final, las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales u horizontales, Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas a un símbolo. Los diagramas de flujo deben empezar de arriba a abajo.

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. Por tanto, es fundamental conocer los elementos que conforman este lenguaje gráfico.



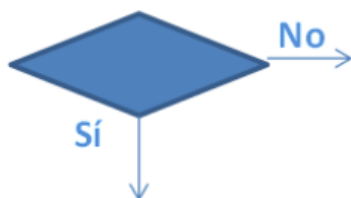
Representa en inicio o fin del diagrama de flujo



Aquí van los datos de entrada que expresa una lectura de datos



Proceso. En su interior se expresan asignaciones u operaciones.



Decisión. Valida una condición y toma uno u otro camino.



Escritura. Escribe el o los resultado(s).



Conexión dentro de la misma página.



Dirección del flujo del diagrama

También existen otros elementos que indican conexión entre páginas o de decisión múltiple. Todo programa, sin importar el lenguaje en el que esté escrito, tiene un flujo o secuencia. Este flujo indica el orden en el que deben ser ejecutadas las acciones. De hecho, esta secuencia es una característica importantísima de un algoritmo. Las estructuras de control de flujo sirven para modificar el flujo normal. Las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa, ya sea repitiéndolas un determinado número de veces, o decidiendo si se deben o no ejecutar. Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

Las estructuras de control secuenciales son declaraciones que se realizan una a continuación de otra en el orden en el que están escritas sin realizar ningún tipo de decisión o repetición como en el caso de que quieras sumar dos números, solo ingresas los dos números y te da el resultado.

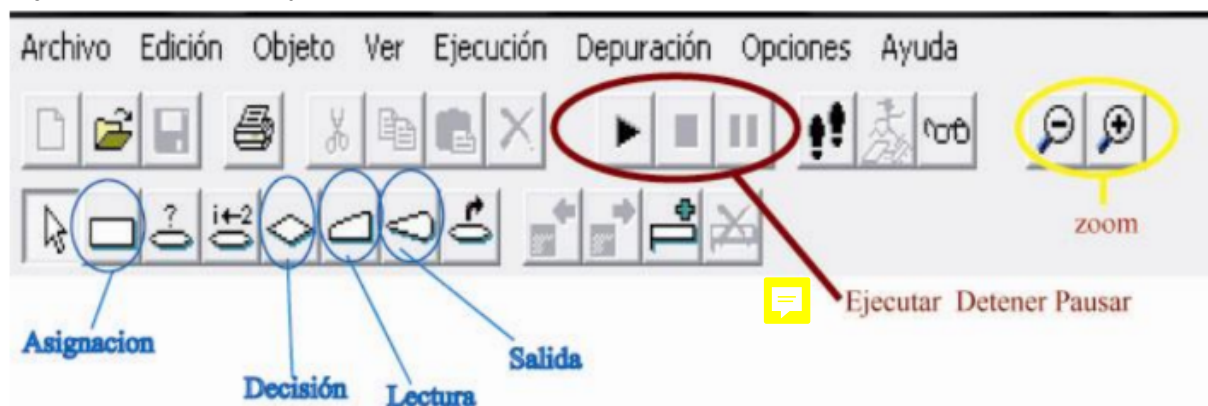
Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se realiza una acción o se realiza la otra). Se usa cuando hay que tomar una decisión, es decir, cuando se evalúa una condición y dependiendo de si es verdadera o falsa, se ejecuta un camino diferente.

Las estructuras de control de flujo iterativas o repetitivas (también llamadas cíclicas) permiten realizar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas: MIENTRAS y HACER- MIENTRAS. La estructura MIENTRAS primero valida la condición y si ésta es verdadera (Sí) procede a realizar el bloque de instrucciones de la estructura [Acciones] y regresa a validar la condición, esto lo realiza mientras la condición sea verdadera (Sí); cuando la condición sea Falsa (No se cumpla) se rompe el ciclo y el flujo continúa con las estructuras que le sigan, mientras que iterativa es HACER-MIENTRAS, la cual primero realiza las instrucciones descritas en la estructura [Acciones] y después válida la expresión lógica; si la expresión lógica es verdadera, realiza nuevamente las [Acciones] y esto se repite mientras la expresión lógica se cumpla. Cuando ésta sea falsa, se rompe el ciclo y el flujo continúa con las estructuras que le siguen. Mientras (While): Se ejecuta mientras la condición sea verdadera. Repetir (Do-While): Se ejecuta al menos una vez y luego se repite mientras la condición sea verdadera.

DESARROLLO

En esta práctica probamos los algoritmos que creamos en la práctica pasada, representandolos en diagramas de flujo de datos, una representación visual del flujo de datos a través de un sistema de información, para esto utilizaremos el software llamado DFD, el cual nos fue proporcionado por el profesor.

Dfd es un software diseñado para construir y analizar algoritmos. Este suministra herramientas de edición estructuradas con el objetivo de permitir crear diagramas de flujo de datos para la representación de algoritmos de programación. Después de haber ingresado el algoritmo representado por el diagrama, el programa permite ejecutarlo con el objetivo de analizarlo y depurarlo.

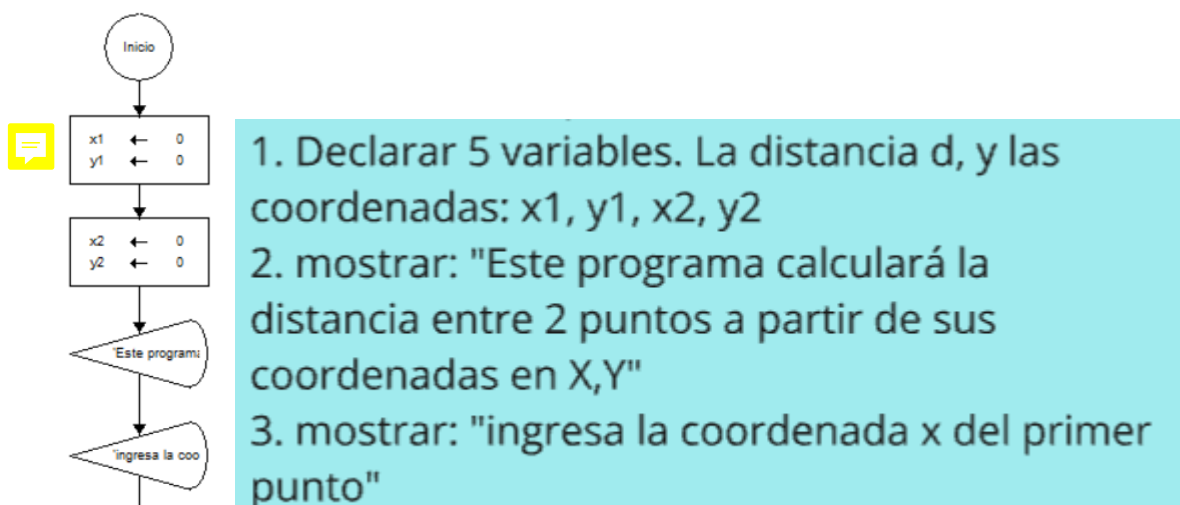


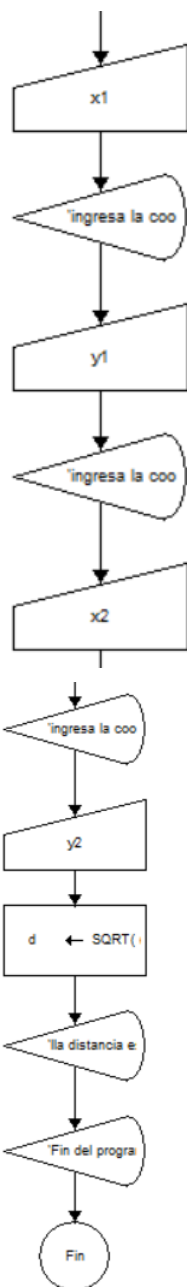
Comenzando con el primer problema no necesitamos las estructuras de control iterativo que se había considerado para comprobar que los distintos valores introducidos por el usuario se encuentren dentro del conjunto de los números reales con el que podemos trabajar, ya que el programa DFD los realiza automáticamente.

3. mostrar: "ingresa la coordenada x del primer punto"
4. leer y verificar el valor ingresado, en caso de no cumplir las restricciones señaladas ir al paso 17

17. mostrar: "El valor es incorrecto, ingresa un valor diferente", ir al paso 3

Pero además de esta circunstancia el código fue funcional al ser traducido de manera equivalente a lo que se había establecido en el texto.





4. leer y verificar el valor ingresado, en caso de no cumplir las restricciones señaladas ir al paso 17

5. Asignarle a la variable x1 el valor ingresado por el usuario

6. mostrar: "ingresa la coordenada y del primer punto"

7. leer y verificar el valor ingresado, en caso de no cumplir las restricciones señaladas ir al paso 18

8. Asignarle a la variable y1 el valor ingresado por el usuario

9. mostrar: "ingresa la coordenada x del segundo punto"

10. leer y verificar el valor ingresado, en caso de no cumplir las restricciones señaladas ir al paso 19

11. Asignarle a la variable x2 el valor ingresado por el usuario

12. mostrar: "ingresa la coordenada y del segundo punto"

13. leer y verificar el valor ingresado, en caso de no cumplir las restricciones señaladas ir al paso 20

14. Asignarle a la variable y2 el valor ingresado por el usuario

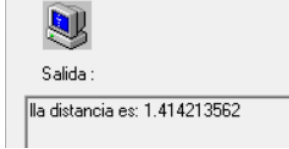

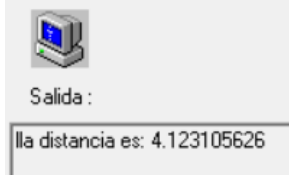
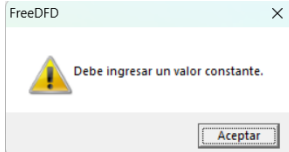

15. asignarle a la variable d el resultado de la operación: $\sqrt{[(x1-x2)^2+(y1-y2)^2]}$

16. mostrar: "La distancia es: ", d

Fin del programa

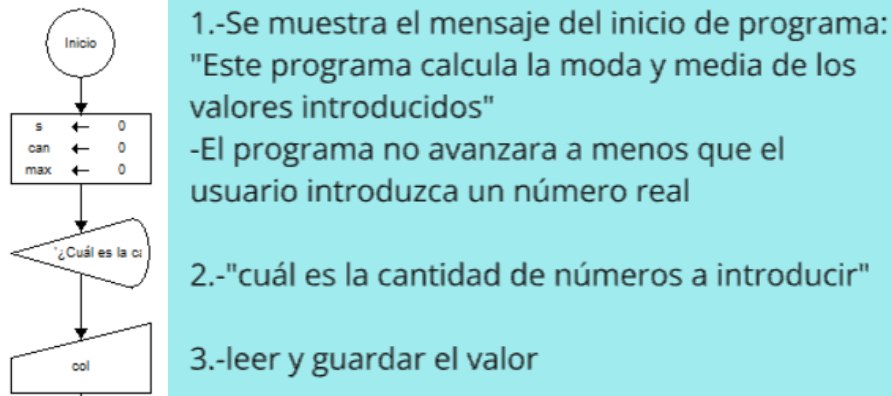
A lo largo de este programa le mostramos distintos mensajes al usuario para darle indicaciones de lo que debe hacer, posteriormente procesamos estas entradas para darle las salidas deseadas, proceso que se repite en el segundo programa.

Para terminar con el primer programa se decidió comprobar las pruebas de escritorio utilizando dfd para verificar su correcto funcionamiento.

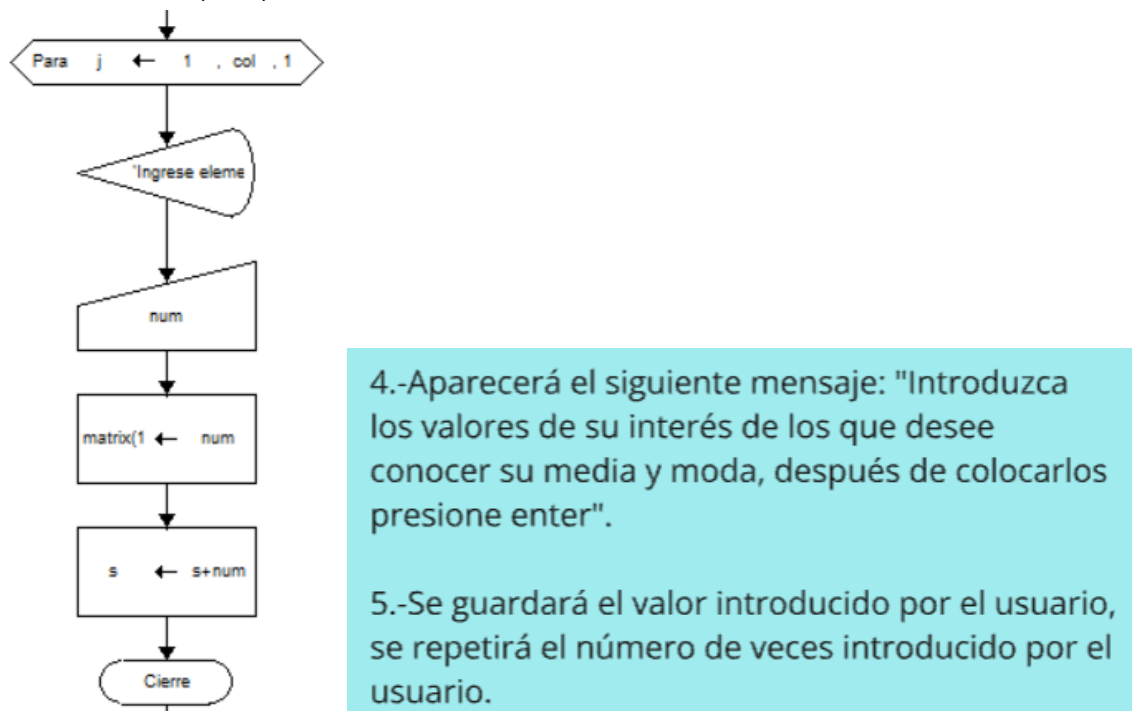
Iteración	Entradas	Salidas	Salidas DFD
1	x1, y1: 1,1 x2, y2: 2,2	1.4142	<div>Impresion por pantalla</div> 
2	x1, y1: 1,1 x2, y2: 1,1	0	
3	x1, y1: 5,7 x2, y2: 4,3	4.1231	
4	x1: at	"El valor es incorrecto, ingresa un valor diferente" Hace que el usuario reasigne un valor hasta detectar un valor adecuado.	<div>FreeDFD</div> 
5	x1, y1: 1,0 x2, y2: 9,0	8	



Para el segundo ejercicio fue necesario cambiar algunas cosas para permitir el funcionamiento del algoritmo.



El primer cambio fue que se necesitaron declarar algunos contadores que utilizaremos posteriormente siendo la sumatoria (s), la cantidad de datos iguales (can), y el valor máximo de la cantidad (máx).




El próximo cambio fue desarrollar un poco más el punto de guardar el valor introducido, ya que no se especificó muy bien la forma de hacerlo, en este caso hacemos que el usuario introduzca un valor, el cual es guardado en la primer fila de una matriz, pero sumándole uno a la columna, hasta llegar a la cantidad de números ingresada por el usuario.

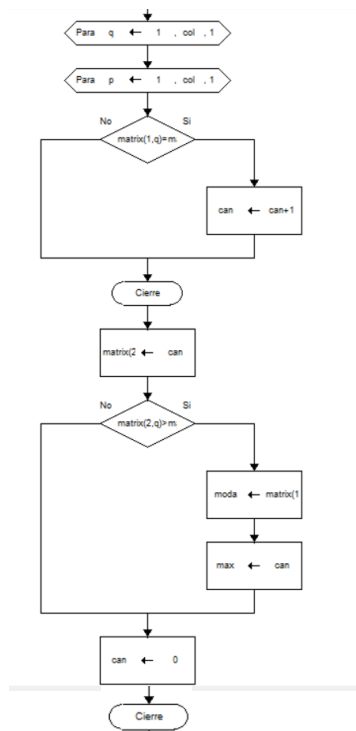
En este paso también aprovechamos el ciclo para calcular la sumatoria, sumando cada número nuevo al valor anterior de la sumatoria, empezando en 0.

Además de esto se cambió el mensaje a "Ingrese elemento: 'j'", el cual permite entender de mejor forma lo que el usuario debe hacer, siendo j la columna actual de la matriz.

6.-En caso de que los valores introducidos no cumplan con las restricciones aparecerá el siguiente mensaje: "El valor introducido no pertenece a los números reales"

-Se regresara al inicio  el programa para volver a introducir los valores

La validación de los valores al igual que en el caso anterior es automática y no necesita ningún paso para ejecutarla.

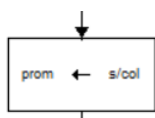


7.-La media se obtendrá como: La sumatoria de todos los términos / el número de términos.

8.-La moda será el valor que más aparece repetidamente

También hubo que desarrollar más los siguientes pasos, ya que aunque dimos la definición de estos, no dimos el método por el cual serían calculados.

En este caso creamos 2 ciclos que avancen de uno en uno hasta el número de datos ingresado, en el primer ciclo se elige una columna a comparar, mientras que el segundo cambia la columna con la que se comparará, manteniendo la fila en 1, repitiendolo hasta el número de datos establecidos, mientras se comparan si dos de los números resultan ser iguales se suma uno al contador de la columna, el cual al terminar la comparación se guardará en la segunda fila de la matriz. Para posteriormente comparar el valor con el valor máximo de la cantidad, y si la cantidad de la columna resulta ser mayor ocurrirán dos cosas, la primera es que el nuevo valor máximo sustituirá al anterior, y la segunda es que se establecerá la moda como el valor de el número al cual pertenece el contador.



También se cambió el orden ya que la media se calculará posteriormente para evitar que el cálculo se realice múltiples veces.



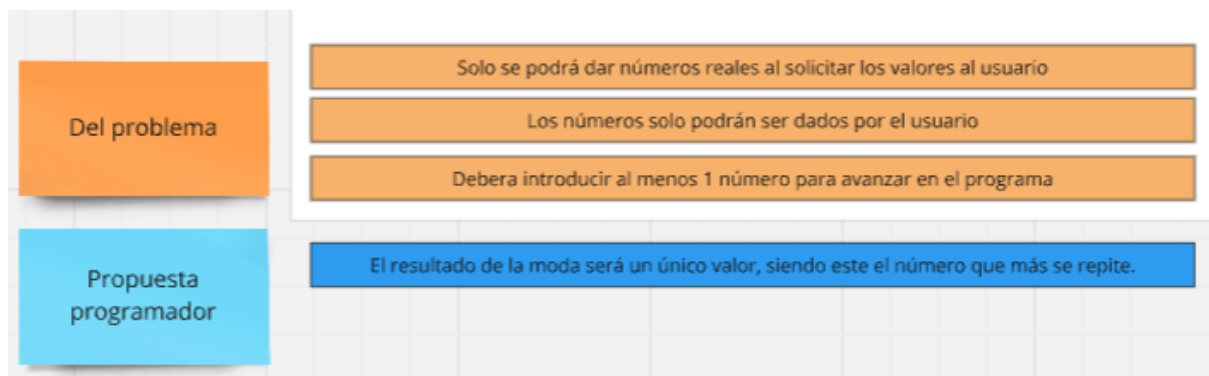
9.-Aparecerán los siguientes mensajes:

- "La moda de los valores introducidos es:" moda

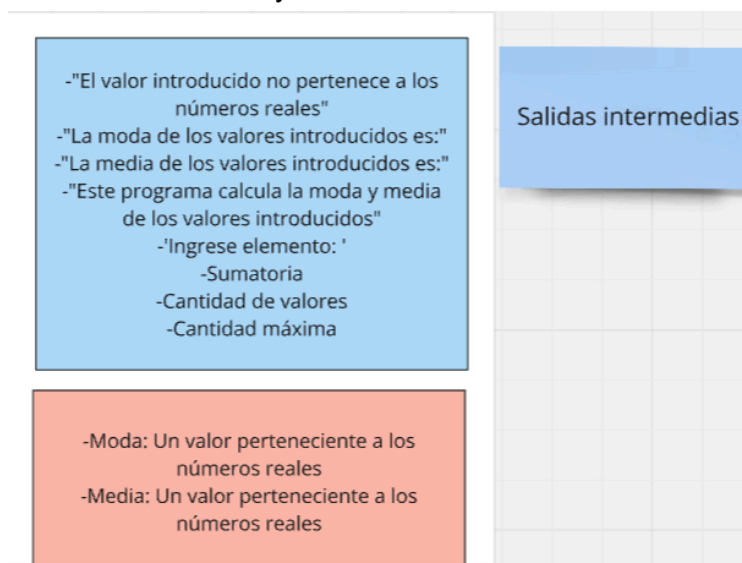
- "La media de los valores introducidos es:" media

FIN DEL PROGRAMA



En los mensajes finales no se cambió nada, además de separarlos por motivo estético, pero tuvimos la necesidad de añadir una restricción más a las ya existentes, siendo que este solo dará un valor único como resultado en la moda.



Cómo último cambio se añadieron algunas salidas intermedias extra siendo la sumatoria, la cantidad de valores, y la cantidad máxima



Para terminar se realizaron las pruebas de escritorio con el programa para verificar el funcionamiento.

Iteración	Entrada	Salida	Salida DFD
1	1, 3, 3, 5, 3, 7	La moda de los valores introducidos es: 3 La media de los valores introducidos es: 3.67	<div>La media de los valores introducidos es: 3.666666667</div> <div>la moda de los valores introducidos es:3</div>
2	1.5, 6, 8, 8, 8, 9, 13, 4	La moda de los valores introducidos es: 8 La media de los valores introducidos es: 6.88	<div>La media de los valores introducidos es: 6.888888889</div> <div>la moda de los valores introducidos es:8</div>
3	1, 2, a, #, 7	El valor introducido no pertenece a los números reales	<div>FreeDFD</div> <div>  Debe ingresar un valor constante. </div> <div>Aceptar</div>
4	5	La moda de los valores introducidos es: 5 La media de los valores introducidos es: 5	<div>La media de los valores introducidos es: 5</div> <div>la moda de los valores introducidos es:5</div>
5	4, 6, ?, f, %	El valor introducido no pertenece a los números reales	<div>FreeDFD</div> <div>  Debe ingresar un valor constante. </div> <div>Aceptar</div>

Conclusiones

Matías Serrano - La práctica me pareció muy interesante, al poder darme cuenta de lo mucho que fue necesario cambiar el texto para poder hacer que funcionara, además me pareció bastante útil ya que aunque sabía cómo funcionaban los diagramas de flujo no conocía ninguna herramienta como dfd, la cual me permitió observar el programa de una forma mucho más visual, y en ciertos casos entendible de lo que sería normalmente como un algoritmo de acciones a ejecutar en un lenguaje que en ocasiones puede parecer confuso.

Edgar Reyes Vargas: La práctica me pareció bastante interesante ya que me hizo darme cuenta que hay que ser demasiado específicos y dar demasiados detalles a la hora de hacer un algoritmo y querer pasarlo a diagrama de flujo, ya que me di cuenta que no había puesto todo lo que debía de considerar en el algoritmo, a la hora de querer hacer el diagrama de flujo me encontré con muchas trabas y barreras que tenía, así también que tuvimos que agregar para hacer que el diagrama funcione. Me permitió comprender y aplicar de manera efectiva los aprendizajes fundamentales en la resolución de problemas. También a través del desarrollo de la práctica creo que reforcé la importancia de representar procesos de forma clara y ordenada, facilitando la visualización del flujo lógico de ejecución. En conclusión, los diagramas de flujo son una herramienta esencial para la resolución de problemas y la estructuración de procesos, mejorando la capacidad de análisis y facilitando la comunicación de algoritmos de manera visual.

Referencias:

- E García Cano, E., & A Solano Gálvez, J. (2025, 22 enero). Guía práctica de estudio 04: Diagramas de flujo. Manual de Prácticas del Laboratorio de Fundamentos de Programación UNAM.
- Kosinski M., Lindemulder G. (2024, 22 de noviembre). Diagrama de flujo de datos. IBM. <https://www.ibm.com/mx-es/think/topics/data-flow-diagram>
- Grupo Smart. (2019). DFD 1.0: Diagramas de Flujo de Datos Manual de Referencia. https://drive.google.com/file/d/16UpWCocaps3moscekM0GgCQF5A_rkGG1/view?usp=sharing
- Nieva, G. (2020, 23 marzo). *Estructuras de control de flujo*. dCodinGames. <https://dcodingames.com/estructuras-de-control-de-flujo/>
- *Qué es un diagrama de flujo*. (s. f.-b). Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo>

