



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL DE ESTUDIOS SUPERIORES
UNIDAD JURIQUILLA

Ingeniería Aeroespacial

Semestre II

Fundamentos de Programación

Práctica III

"Diagramas de Flujo en DFD"

EQUIPO 3

Integrante 1: Fryda Itzel Bastida Reyes

Cuenta: 322119071

Integrante 2: Jesús Daniel Andrade Mendoza

Cuenta: 425062704

Profesor: Ing. Andres David Flores Ferro

Fecha de entrega: 21 - Febrero - 2025

Fecha de revisión:

Calificación:

Observaciones:

Objetivo General

Elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

Objetivo Particular

- Elaborar un diagrama de flujo que represente la solución algorítmica de un problema, en el cual requiera el uso de la estructura de control condicional.
- Elaborar la representación gráfica de la solución de un problema, a través de un diagrama de flujo, en el cual requiera el uso de la estructura de control iterativa.

Introducción

En el presente reporte se analiza la práctica de programación realizada, abarcando desde la exploración de algoritmos y diagramas de flujo hasta la gestión de datos y la planificación de proyectos. Se ha buscado integrar los conceptos aprendidos en clase con una investigación exhaustiva, enriqueciendo así la comprensión de los procesos de desarrollo de programa.

Los diagramas de flujo son representaciones gráficas que permiten visualizar la secuencia de pasos de un algoritmo o proceso. Son herramientas fundamentales en la programación, ya que facilitan la comprensión y el diseño de soluciones a problemas complejos. Un diagrama de flujo es un esquema visual que muestra el camino que sigue un programa o proceso, utilizando símbolos gráficos para representar diferentes tipos de acciones y decisiones, y flechas para indicar el flujo de ejecución.

El análisis de los diagramas de flujo se basa en el "Programa DFD", lo que permitió explorar diferentes enfoques algorítmicos y comprender la importancia de la representación visual. Se profundizó en la necesidad de detallar procedimientos y consumir información de manera eficiente, ejemplificando esto con ejercicios de ordenamiento y la interpretación de diagramas de flujo con significados específicos. Asimismo, se exploró la representación de ciclos, donde el dato final se convierte en el punto de partida, destacando la importancia de la toma de decisiones y la validación de información.

A través del análisis de diversas situaciones, desde la gestión hasta la simulación de procesos de votación, se buscó aplicar los conceptos aprendidos a problemas prácticos y relevantes. Este enfoque permitió consolidar el conocimiento sobre diagramas de flujo y su aplicación en el desarrollo del programa, subrayando su papel esencial en la representación visual de procesos y en la optimización de la programación.

Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.

Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.

Desarrollo

Cabe mencionar que para este problema tuvimos que hacer algunas modificaciones, inicialmente habíamos planteado que el programa fuera capaz de calcular el promedio, mediana, y moda de un conjunto de números (n), es decir, inicialmente no restringimos un rango de números; sin embargo fue necesario hacerlo por las siguientes razones:

- Reduce la probabilidad de errores y facilita la depuración.
- Se optimiza el rendimiento en términos de tiempo y memoria, ya que los cálculos son constantes y no requieren un procesamiento complicado.
- Al trabajar con tres números, se tiene un control completo sobre los casos que pueden ocurrir, lo que hace que el programa sea más predecible y fácil de manejar.

Ahora que ya hemos aclarado este punto. Seguimos con la explicación:

CASOS

Siendo num1, num2, num3

PPROMEDIO

Este valor no necesita condición, solamente la operación correspondiente

MEDIANA

Valor medio en el conjunto de números (en este caso 3), como el número es impar, solo se toma el valor central

CASOS SIN MODA

- Num1, num2, num3
- Num1, num3, num2
- Num2, num1, num3
- Num2, num3, num2
- Num3, num1, num2
- Num3, num2, num1

CASOS CON MODA

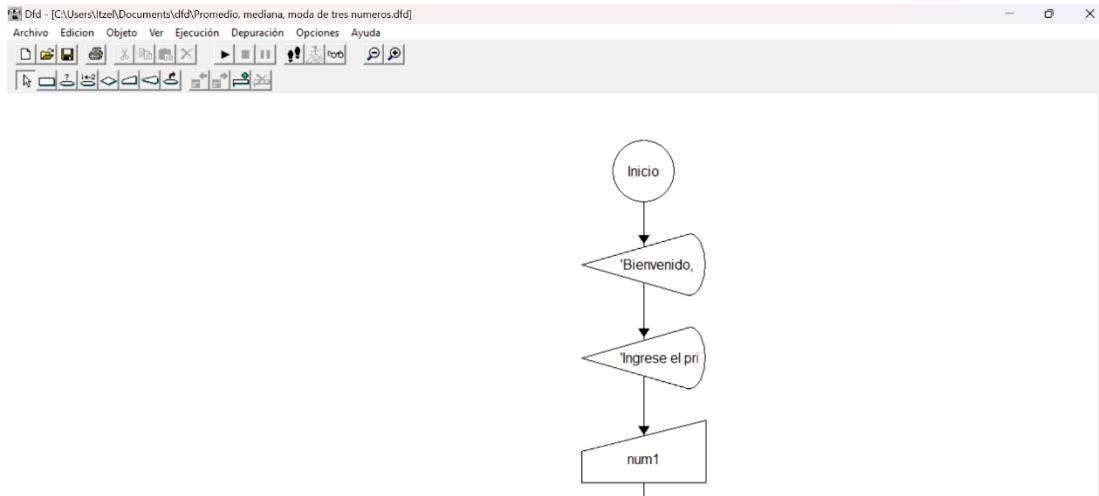
Aquí se representa con letras las posiciones que ocupan los números, es decir, de cada uno salen 3 casos diferentes, donde solo cambia el número, pero las posiciones son las mismas

- A,A,A
- AAB
- BBA

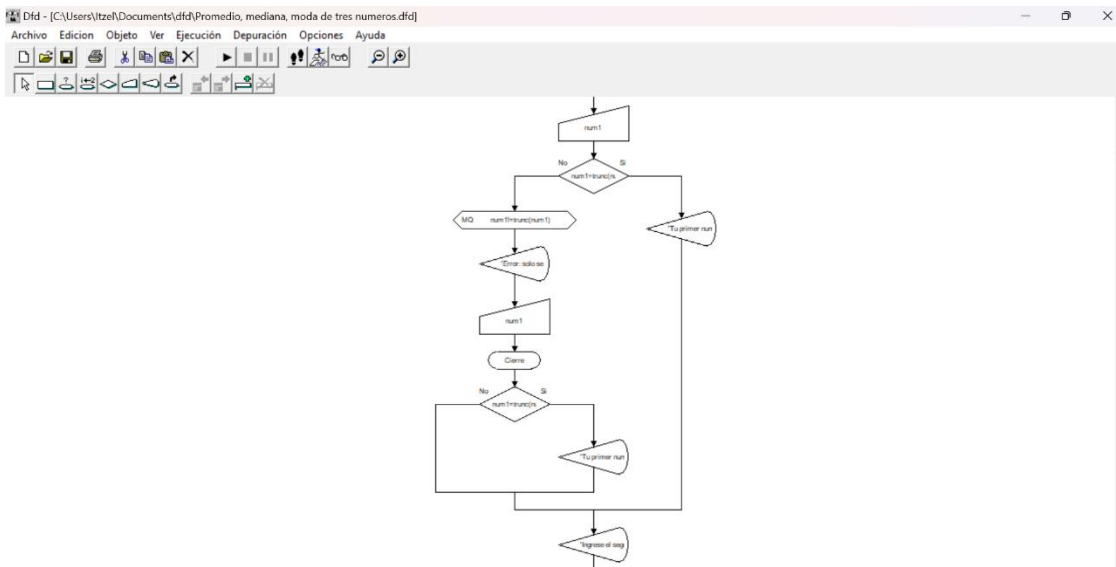
Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.

Desarrollo



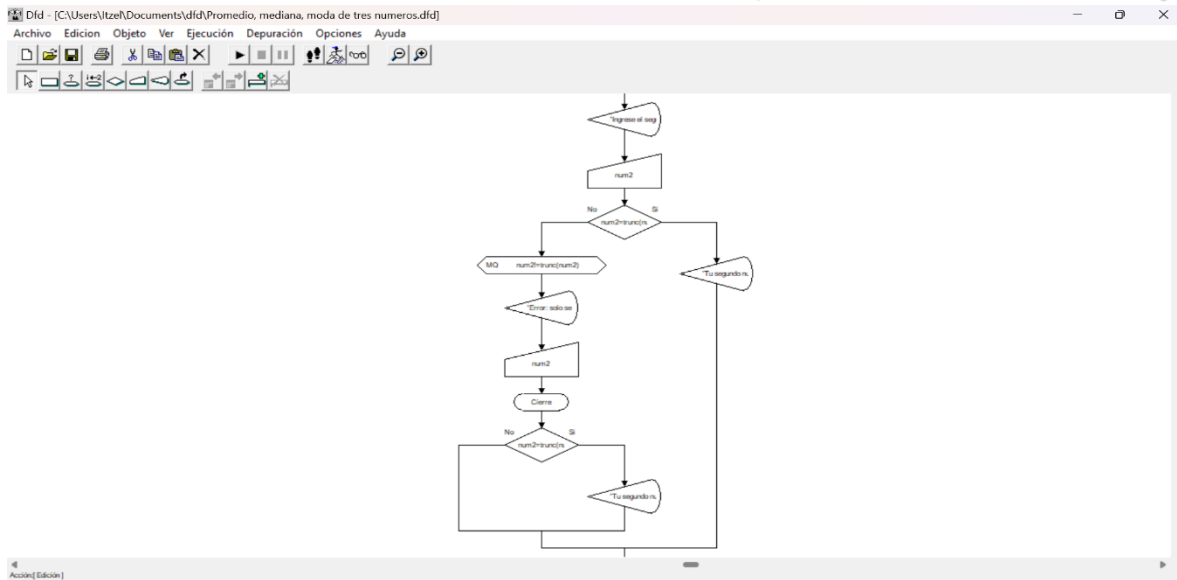
- El programa inicia con un mensaje de bienvenida, e indica lo que es capaz de hacer.
- Se le solicita al usuario que ingrese el primer número.
- El primer número, será leído en la variable (num1)



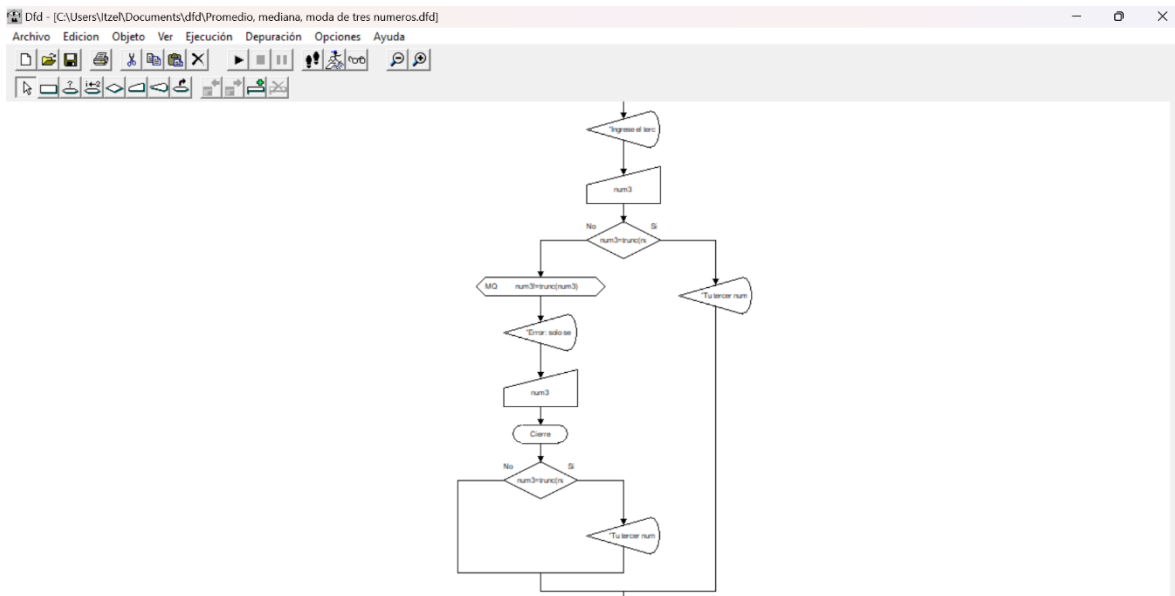
- El primer número ya lo leyó el programa
- El primer número es leído en la variable (num1)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.
- Si la corrección es válida, se le muestra en mensaje al usuario, sobre cual ha sido el número aceptado por el programa
- El programa continua.

Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.



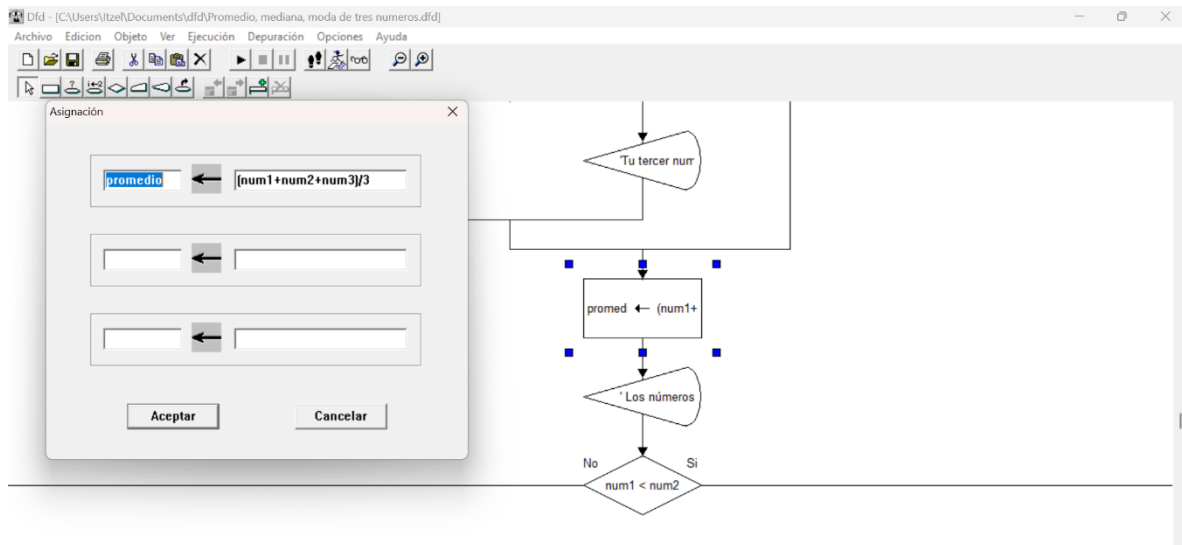
- Para este punto el programa ya guardo el primer número.
- El programa sigue indicando al usuario que ingrese el segundo número.
- El segundo número es leído en la variable (num2)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.
- Si la corrección es válida, se le muestra en mensaje al usuario, sobre cual ha sido el número aceptado por el programa
- El programa continua.



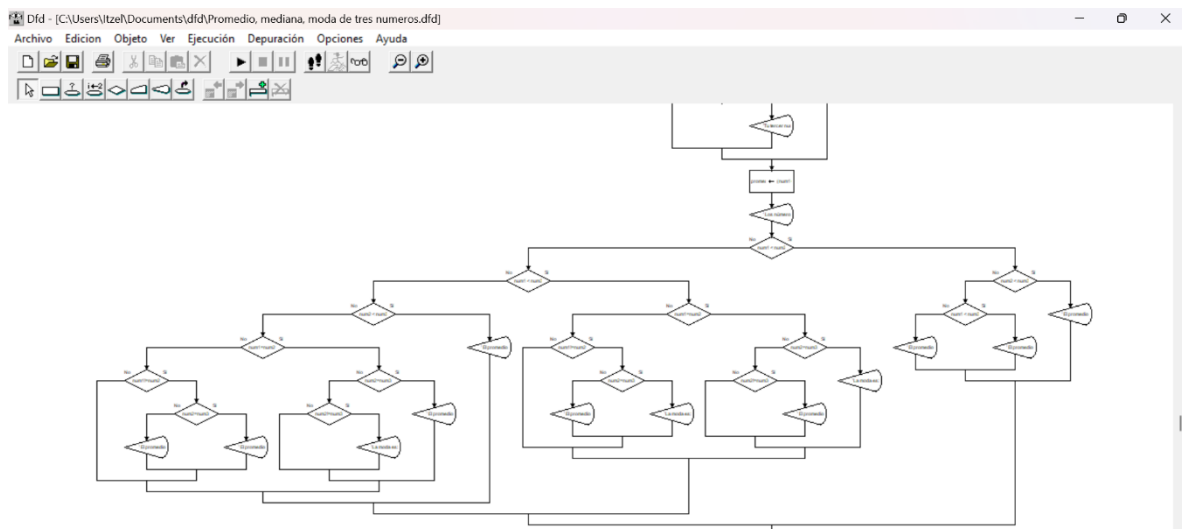
- Para este punto el programa ya guardo el segundo número.
- El programa sigue indicando al usuario que ingrese el tercer número.
- El tercer número es leído en la variable (num3)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.

Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.



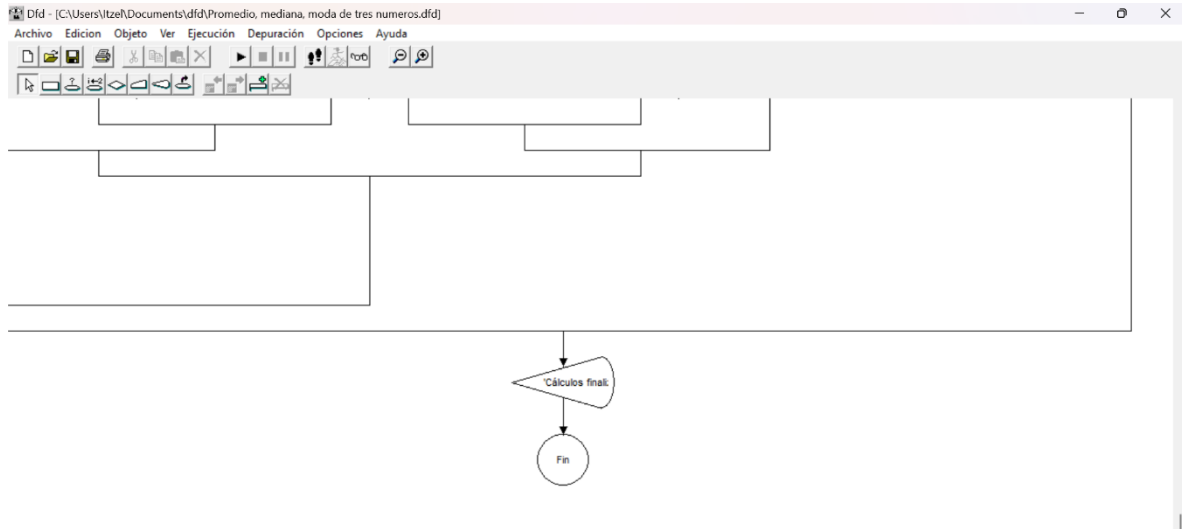
- Para este punto, el programa ya tiene los tres números
- Ahora sigue un cuadro de asignación, donde se ingresa la operación que va a ejecutar el programa y como va a ser leído, en este caso, es el “promedio”.
- Ahora se mostrara al usuario en un mensaje los tres números que ingreso.



- Ahora se establecen todas las condiciones posibles para mediana, y moda, ya que el usuario puede ingresar sus entradas sin ningún orden en específico.
- Junto a cada condición o caso, como salida se mostrará un mensaje con los resultados correspondientes.
- Para este punto el programa ya ha terminado de ejecutar todas las operaciones y condiciones correspondientes.

Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.

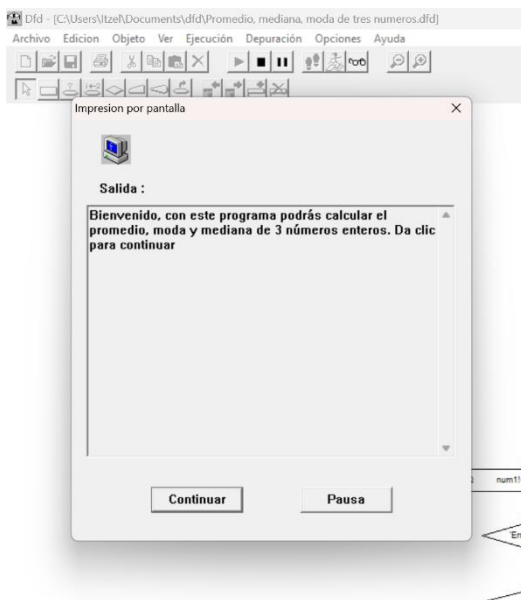


- Finalmente se le muestra al usuario un mensaje indicando que los cálculos han terminado, y agradeciendo por hacer uso del programa
- Fin programa

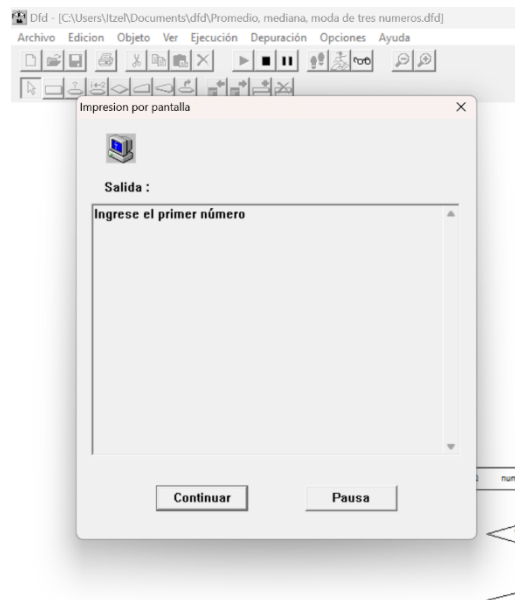
Ejercicio 1 *Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.*

Ejecución

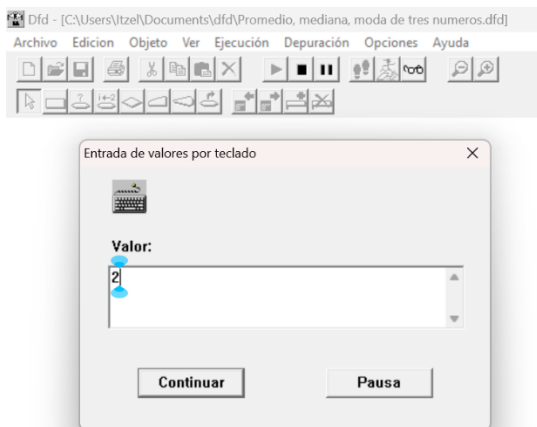
1



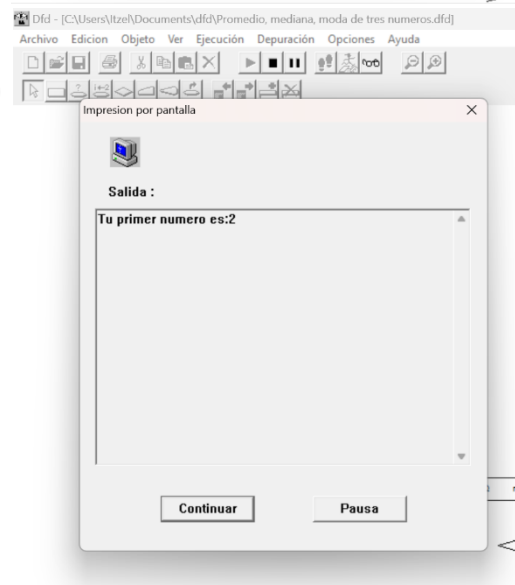
2



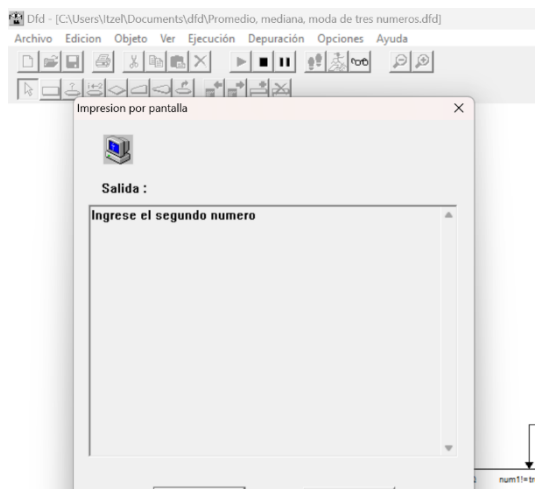
3



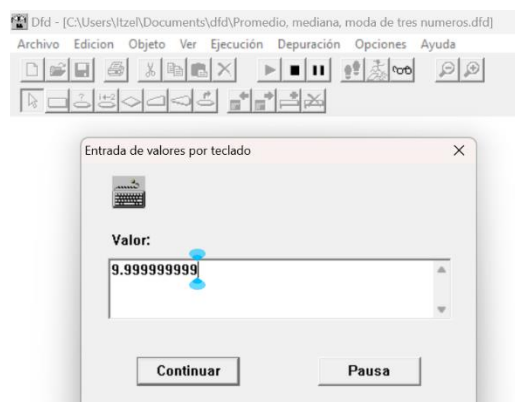
4



5

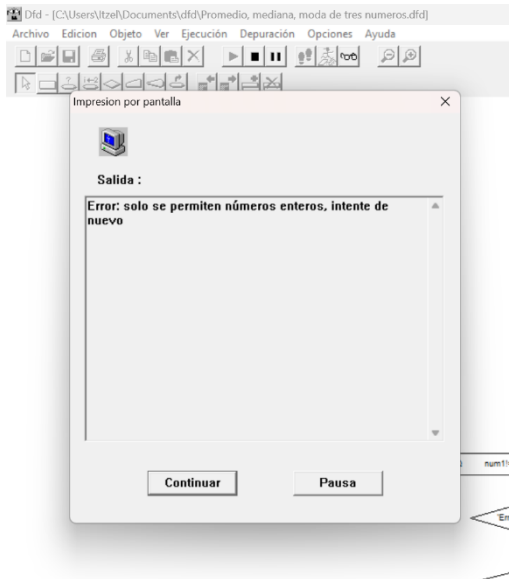


6

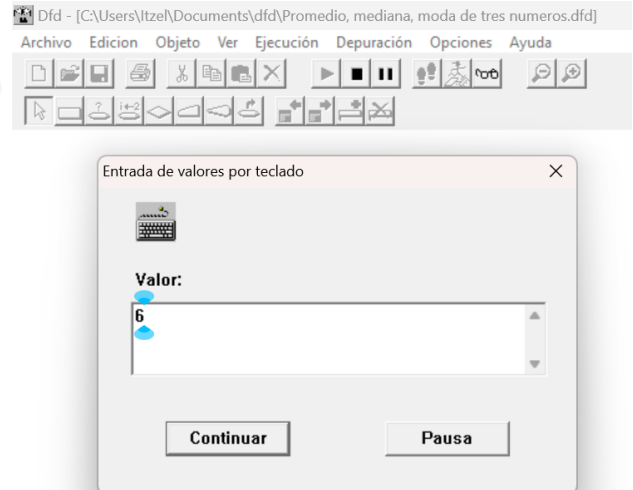


Ejercicio I *Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.*

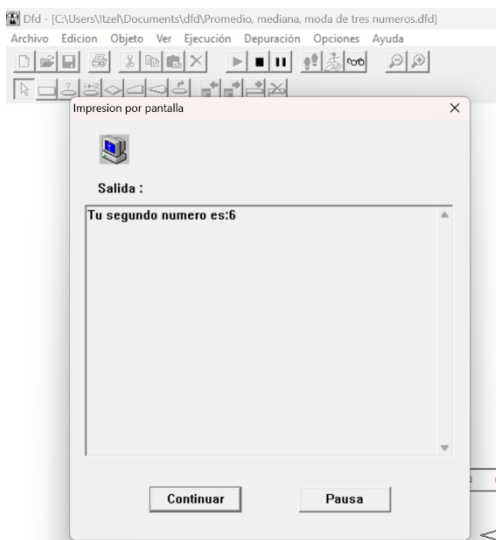
7



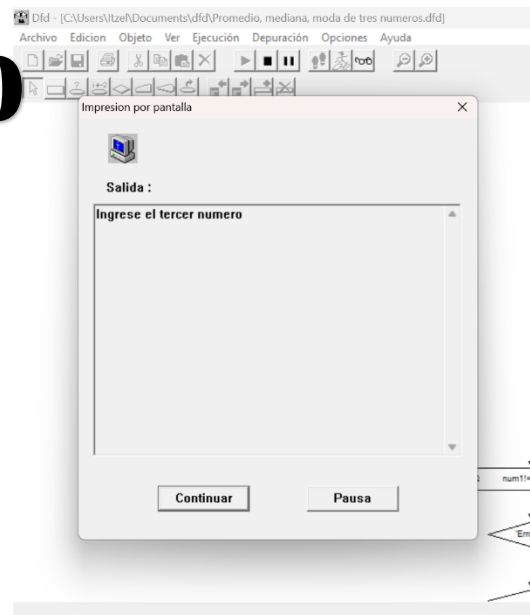
8



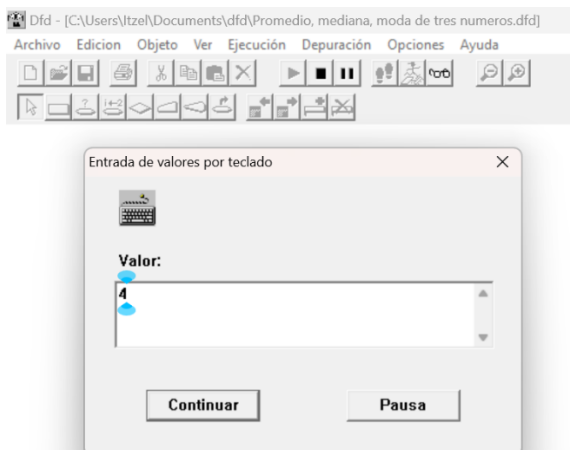
9



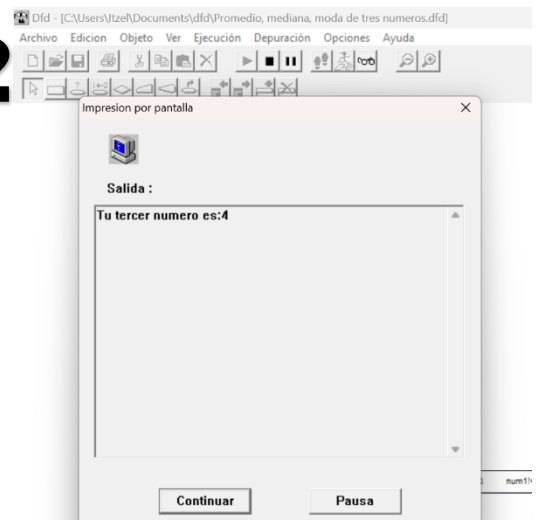
10



11



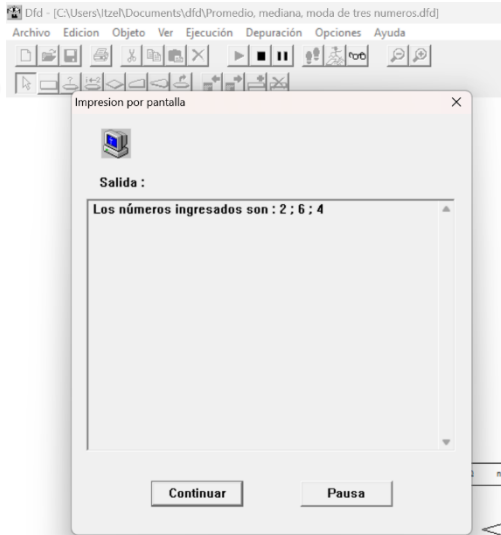
12



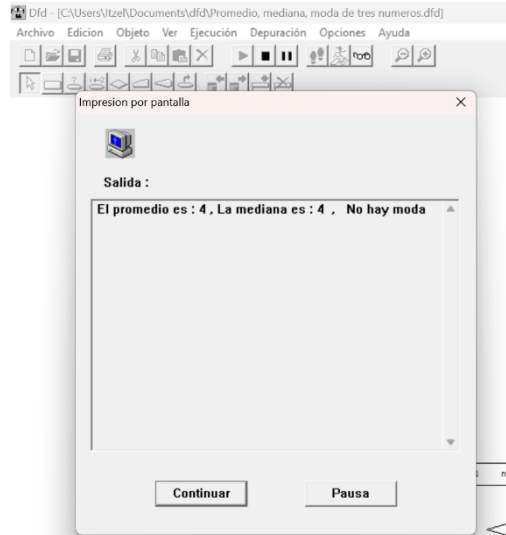
Ejercicio I

Promedio de tres numeros enteros indicados por el usuario, obtener su media y moda.

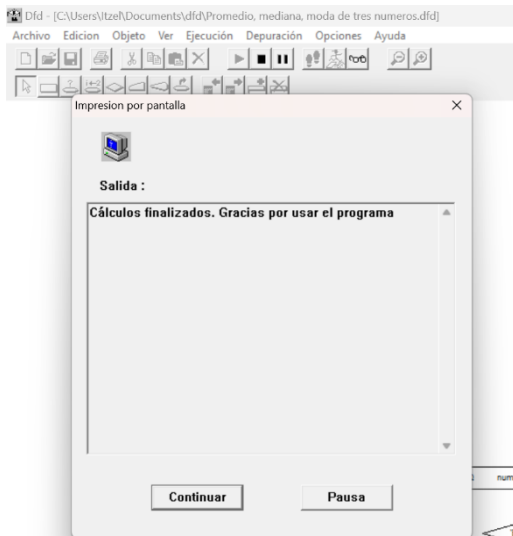
13



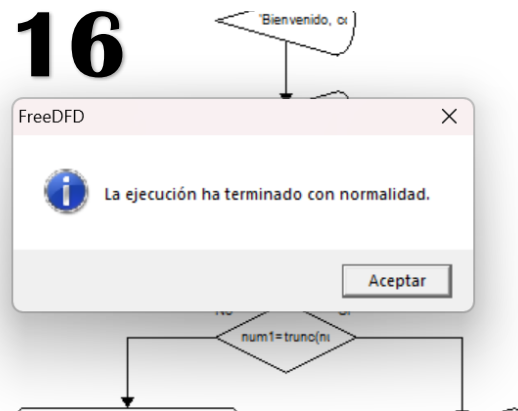
14



15



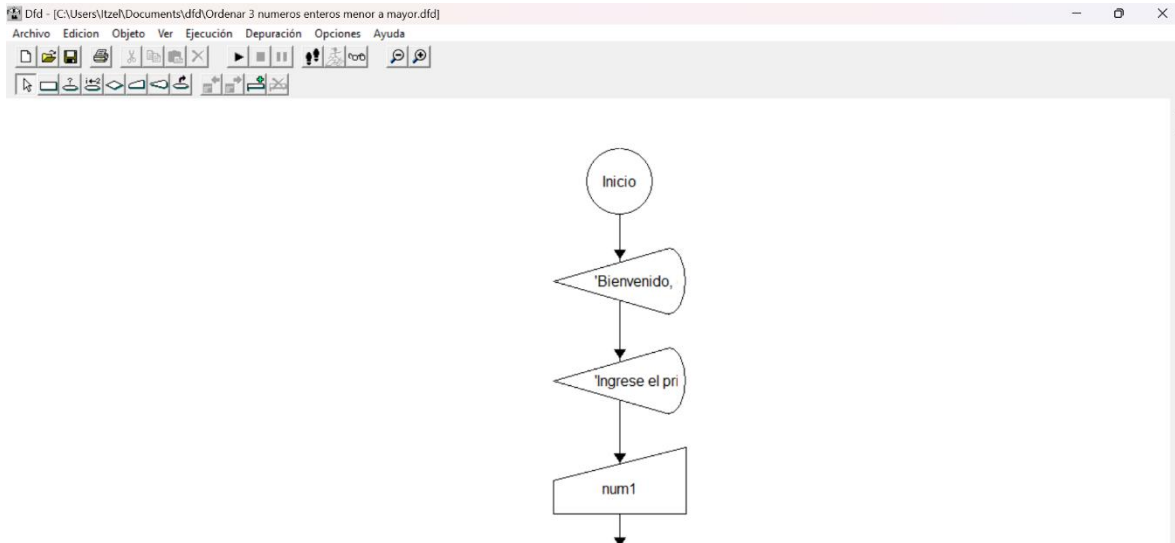
16



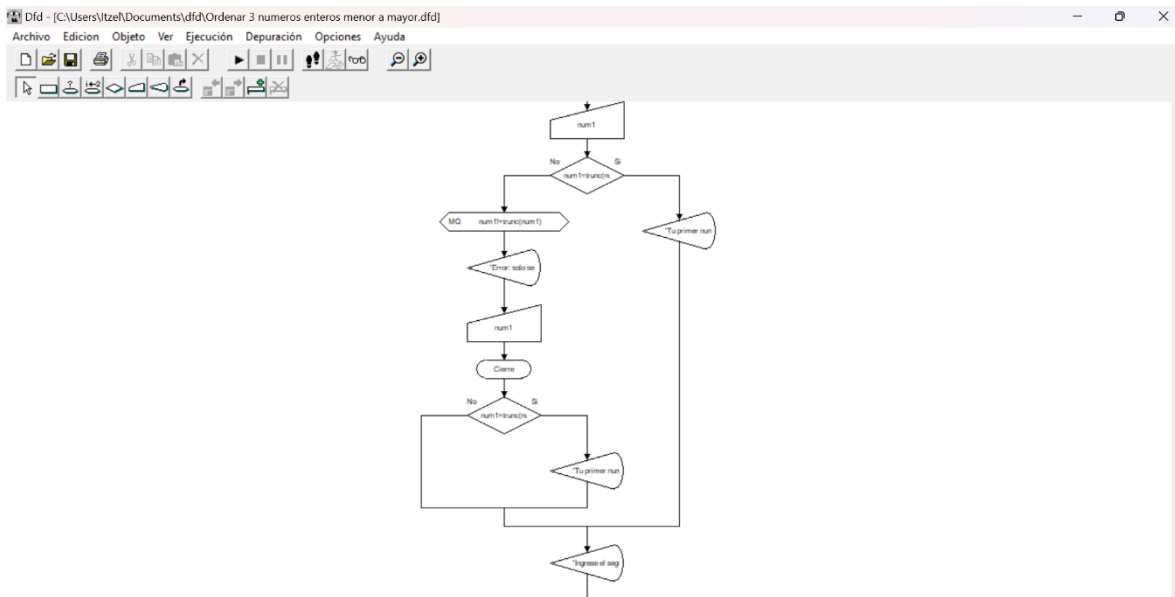
Ejercicio II

*Ordenar 3 números de menor a mayor,
ingresados por el usuario.*

Desarrollo



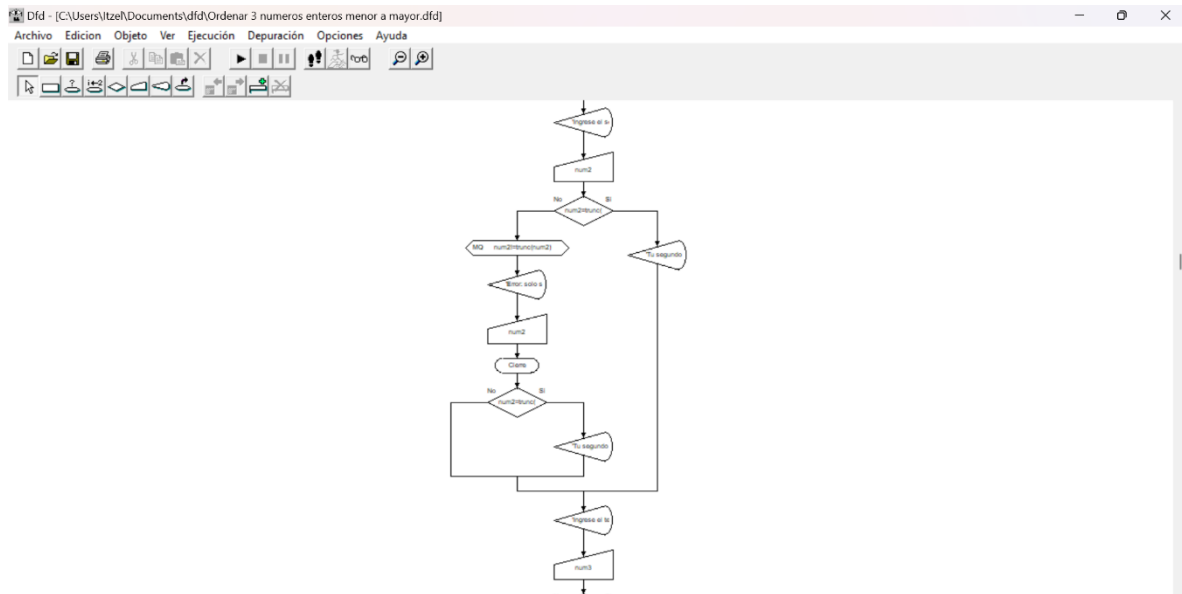
- El programa inicia con un mensaje de bienvenida, e indica lo que es capaz de hacer.
- Se le solicita al usuario que ingrese el primer número.
- El primer número, será leído en la variable (num1)



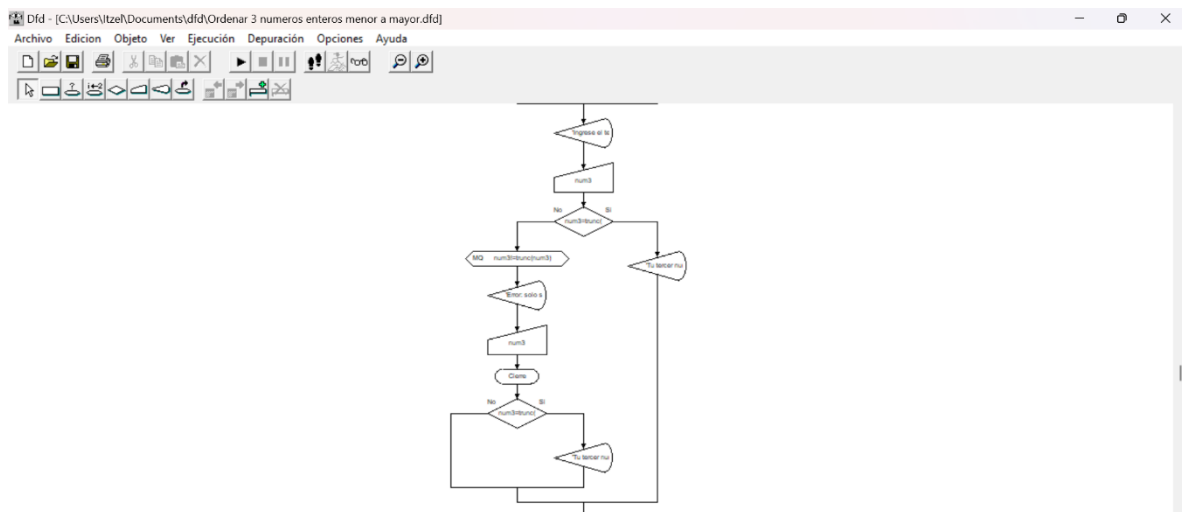
- El primer número ya lo leyó el programa
- El primer número es leído en la variable (num1)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.
- Si la corrección es válida, se le muestra en mensaje al usuario, sobre cual ha sido el número aceptado por el programa
- El programa continua.

Ejercicio II

Ordenar 3 números de menor a mayor, ingresados por el usuario



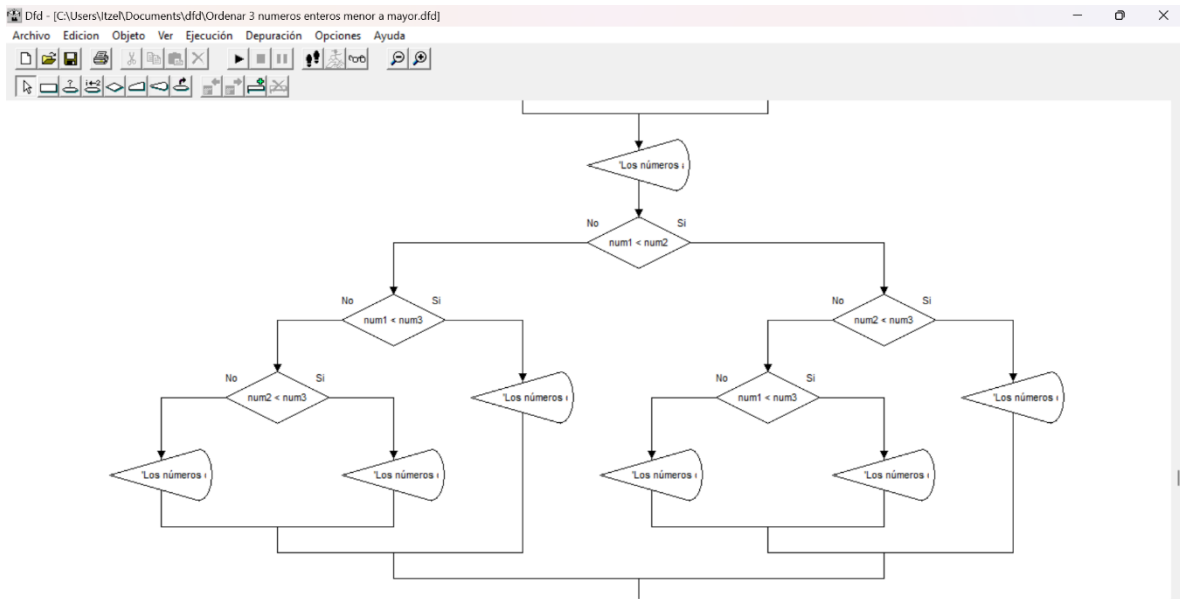
- Para este punto el programa ya guardo el primer número.
- El programa sigue indicando al usuario que ingrese el segundo número.
- El segundo número es leído en la variable (num2)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.
- Si la corrección es válida, se le muestra en mensaje al usuario, sobre cual ha sido el número aceptado por el programa
- El programa continua.



- Para este punto el programa ya guardo el segundo número.
- El programa sigue indicando al usuario que ingrese el tercer número.
- El tercer número es leído en la variable (num3)
- Si en el primer intento, el usuario ingresa una entrada valida, sigue con el programa.
- Si el usuario no ingresa una entrada valida, se genera un mensaje en pantalla diciendo que se ha cometido un error.
- El programa le pedirá al usuario una nueva entrada
- Si la corrección tampoco es válida, se repetirá la entrada de “corrección” hasta que ingrese un valor correcto.
- Si la corrección es válida, se le muestra en mensaje al usuario, sobre cual ha sido el número aceptado por el programa
- El programa continua.

Ejercicio II

Ordenar 3 números de menor a mayor, ingresados por el usuario



- Para este punto el programa ya guardo los tres números ingresados por el usuario.
- Se muestra un mensaje al usuario, indicando cuales fueron los tres números ingresados.
- Se muestran todos los casos pertinentes para poder ordenar los números de menor a mayor, y dado el caso me muestra un mensaje en pantalla con los números ordenados.

CASOS

Combinaciones cuando los tres números son distintos

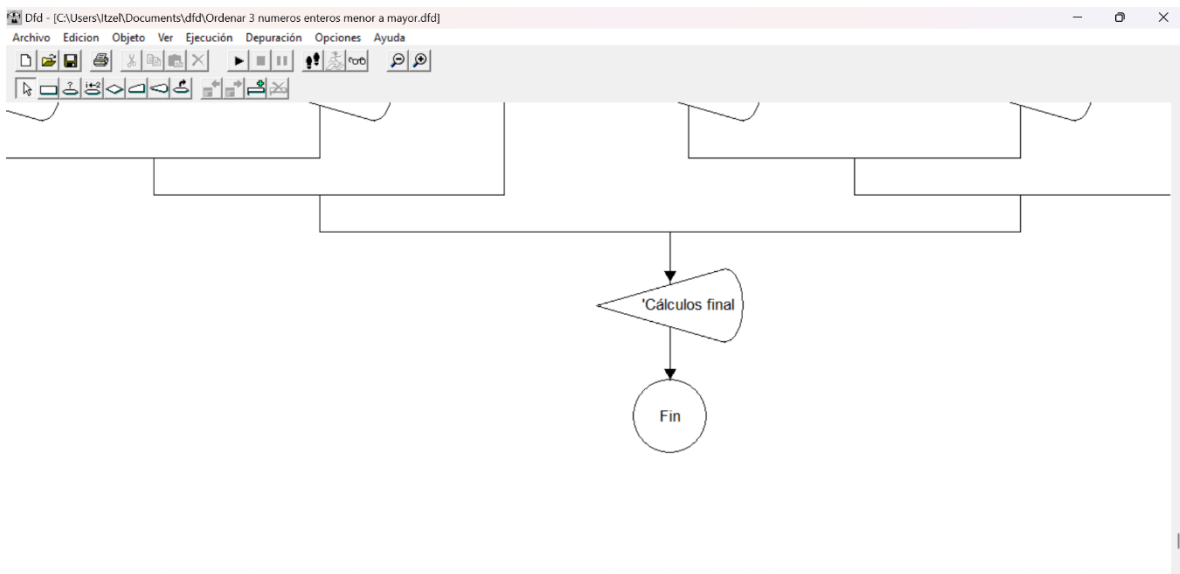
- Num1 < num2 < num3
- Num1 < num3 < num2
- Num2 < num1 < num3
- Num2 < num3 < num1
- Num3 < num1 < num2
- Num3 < num2 < num1

Combinaciones cuando los dos números son iguales

- Num1 = num2 < num3
- Num1 = num3 < num2
- Num2 = num3 < num1

Todos los números son iguales

- Num1 = num2 = num3



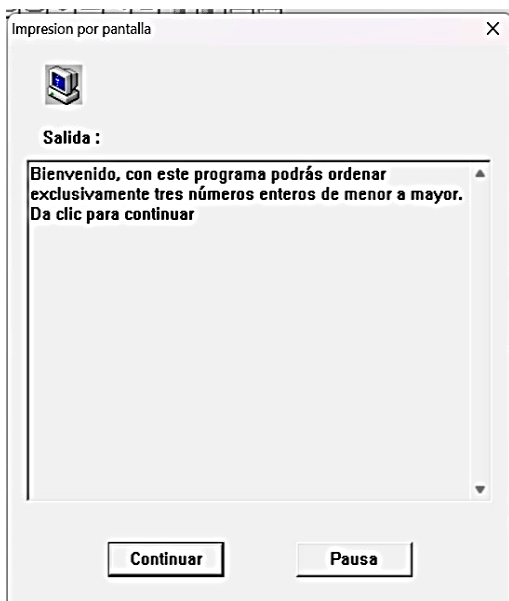
- Finalmente se le muestra al usuario un mensaje indicando que los cálculos han terminado, y agradeciendo por hacer uso del programa
- Fin programa

Ejercicio II

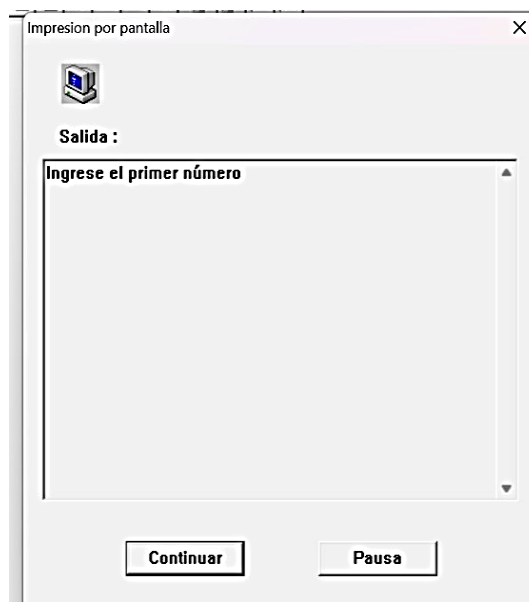
Ordenar 3 números de menor a mayor, ingresados por el usuario

Ejecución

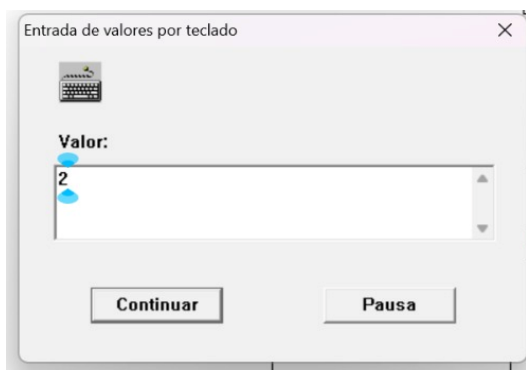
1



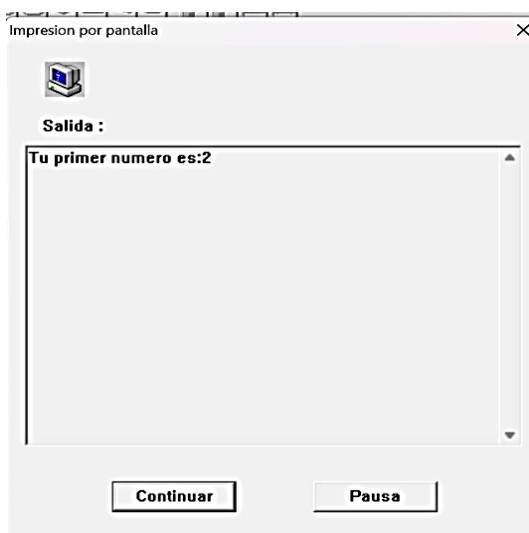
2



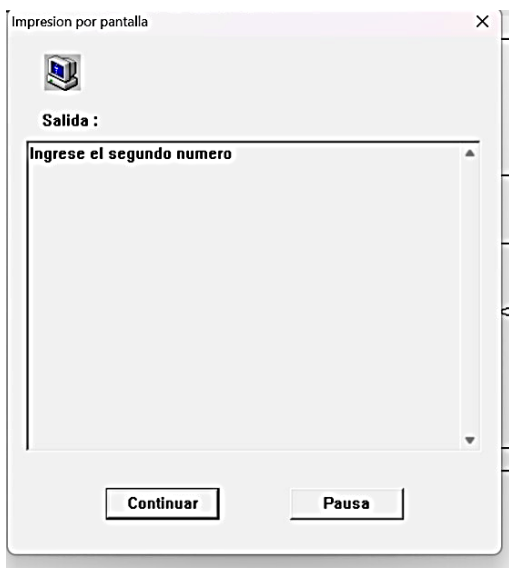
3



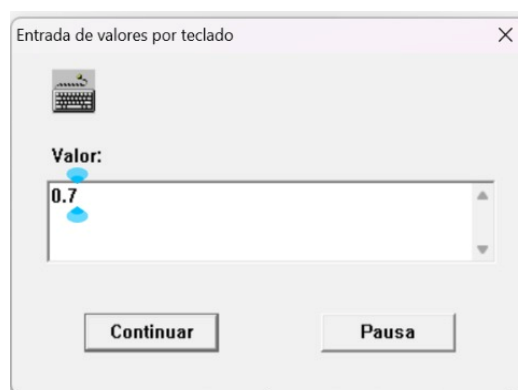
4



5



6



Ejercicio II

Ordenar 3 números de menor a mayor, ingresados por el usuario

7



8



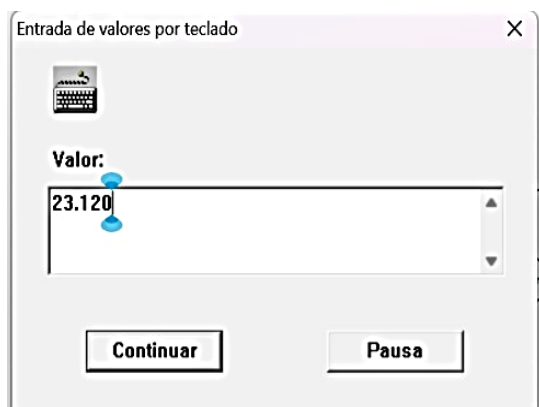
9



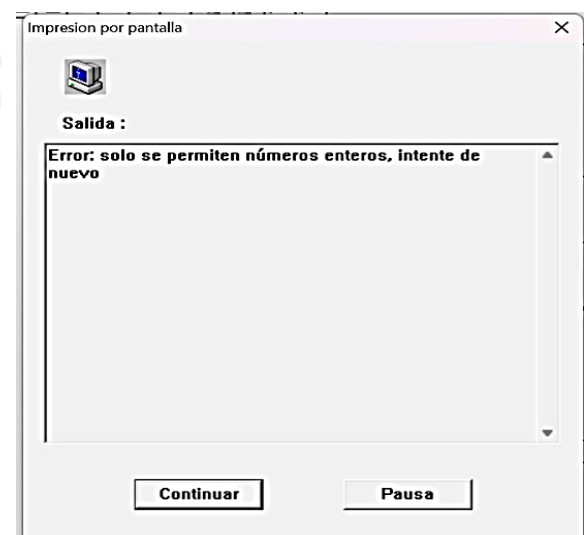
10



11



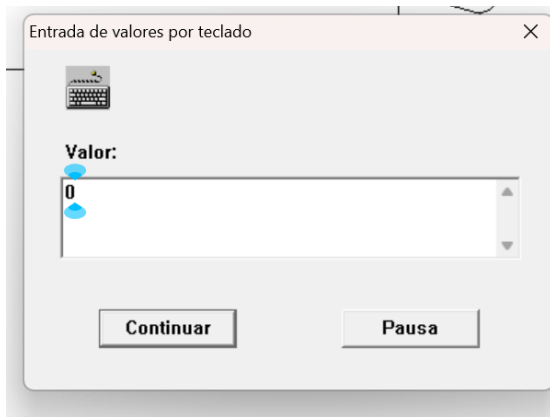
12



Ejercicio II

Ordenar 3 números de menor a mayor, ingresados por el usuario

13



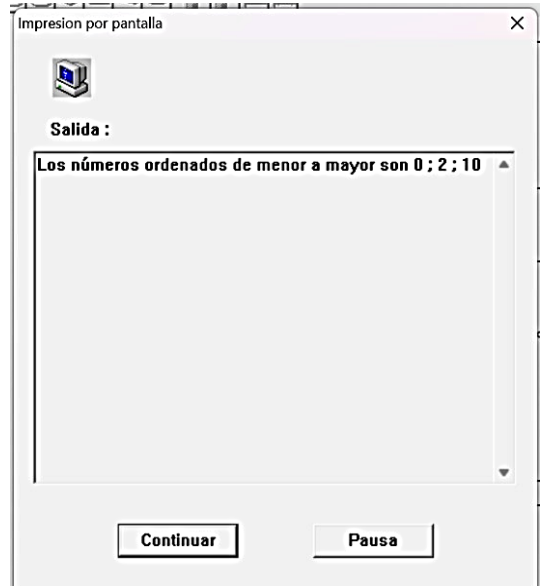
14



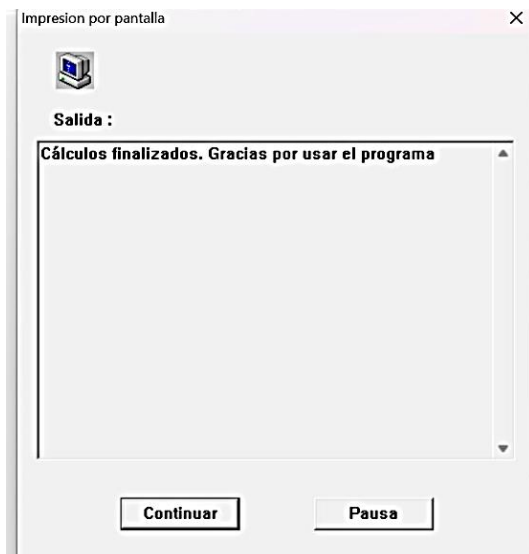
15



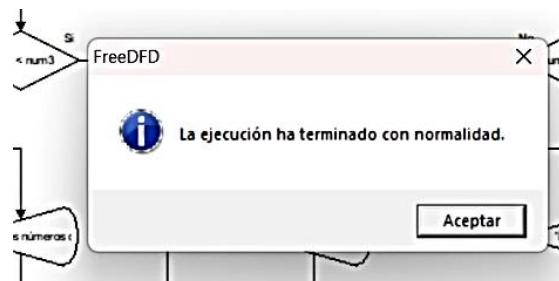
16



17



18



Conclusiones particulares

Integrante 1: Fryda Itzel Bastida Reyes

En base con mi experiencia a lo largo de la práctica puedo decir que aprender a hacer diagramas de flujo, en nuestro caso, diagramas de flujo de datos (DFD), es una habilidad fundamental para representar y comprender procesos de manera clara y estructurada. A lo largo de esta semana, explorar estos diagramas permitió no solamente mejorar la capacidad de modelar sistemas y flujos de información, sino también desarrollar un pensamiento más lógico y ordenado al analizar problemas.

La práctica resultó de gran utilidad, ya que nos hizo darnos cuenta de ciertas limitaciones necesarias que no habíamos considerado en la primera práctica. Fue precisamente el uso del diagrama de flujo lo que nos permitió identificar esos detalles que antes pasaron desapercibidos.

Puedo concluir que dominar el uso de los DFD no solo optimiza la organización de la información, sino que también constituye una herramienta clave en el análisis y diseño de sistemas eficientes. La experiencia de esta semana reforzó la idea de que un buen diagrama no solo representa un proceso, sino que también ayuda a detectar errores y oportunidades de mejora.

Integrante 2: Jesús Daniel Andrade Mendoza

Al término de esta práctica, puedo decir que ha sido un paso significativo en mi comprensión de la programación. Al sumergirme en el análisis de programas existentes, la creación de diagramas de flujo y la planificación de proyectos, he logrado conectar los conceptos teóricos con la aplicación práctica.

He aprendido que la programación va más allá de escribir código; implica un proceso de pensamiento estructurado, la capacidad de resolver problemas de manera lógica y la habilidad de comunicar ideas de forma clara. Los diagramas de flujo, en particular, me han permitido visualizar la lógica de los algoritmos de una manera que nunca antes había experimentado. Por lo tanto concluyo que la práctica fue de utilidad

Conclusion General

La práctica de programación realizada permitió un análisis profundo del Programa DFD, así como de diversos aspectos relacionados con el diseño de algoritmos, diagramas de flujo y procesos. A lo largo del proceso, se evidenció la importancia de estructurar adecuadamente los procedimientos y utilizar herramientas de diagramación para representar flujos de trabajo complejos. Además, se destacó la relevancia de la toma de decisiones en la resolución de problemas

y la validación de información, aspectos clave para garantizar la eficiencia y precisión en la programación.

Se exploraron conceptos como el manejo de entradas y salidas de datos, la asignación de variables y la creación de diagramas de flujo en equipo, lo que permitió una mejor comprensión de los procesos y su aplicación en situaciones reales. Asimismo, se reflexionó sobre la importancia de abordar problemas específicos y manejables, considerando la experiencia previa y los recursos disponibles.

Se ha comprobado que la correcta organización del flujo de información no solo facilita el desarrollo del código, sino que también mejora la escalabilidad y adaptabilidad del programa a diferentes contextos.

Finalmente, la exploración de herramientas de diagramación ha demostrado ser un recurso esencial para la planificación y documentación de procesos. Se concluye que la programación efectiva no solo requiere conocimientos técnicos, sino también una metodología estructurada que garantice la claridad y funcionalidad del código desarrollado.

Referencias

- Flores, A. (2025, 14 de febrero). Fundamentos de programación (Clase). Escuela Nacional de Estudios Superiores Unidad Juriquilla.
- *Diagrama de flujo de datos vs diagrama de flujo - ¿Cuál es la diferencia?* (s. f.). [https://miro.com/. https://miro.com/es/diagrama/diagrama-flujo-datos-vs-diagrama-flujo/](https://miro.com/.https://miro.com/es/diagrama/diagrama-flujo-datos-vs-diagrama-flujo/)
- Laprovittera, C. (2025, 3 enero). *Buenas prácticas de programación*. Álvaro Chirou. <https://achirou.com/buenas-practicas-de-programacion/>
- Maldonado, R. (2024, 10 abril). 8 Buenas prácticas en programación | KeepCoding Bootcamps. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/8-buenas-practicas-en-programacion/>
- Muñoz, J. D. (2024, 9 diciembre). *Curso de introducción a la programación con pseudocódigo*. PLEDIN 3.0. <https://plataforma.josedomingo.org/pledin/cursos/programacion/>
- *Qué es un diagrama de flujo de datos*. (s. f.). Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo-de-datos>