



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

Ingeniería aeroespacial

Semestre II

Fundamentos de Programación

Práctica IV

"Pseudocódigos"

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: 01 - Marzo - 2025

Fecha de revisión:

Calificación:

Observaciones:

Objetivo General

Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

Objetivo Particular

- Elaborar un pseudocódigo que represente la solución algorítmica de un problema en el cual requiera el uso de la estructura de control de flujo condicional.
- A través de un pseudocódigo, representar la solución algorítmica de un problema en el cual requiera el uso de la estructura de control iterativa.

Introducción

En esta práctica de laboratorio, nos sumergiremos en el mundo de la lógica algorítmica, utilizando pseudocódigo como nuestro lenguaje principal de diseño y PSeInt como nuestra herramienta de implementación. El pseudocódigo nos permite expresar la lógica de un algoritmo de manera clara y concisa, sin la complejidad de la sintaxis de un lenguaje de programación específico, facilitando la planificación y el diseño de nuestras soluciones.

El pseudocódigo es una herramienta fundamental en la programación, ya que permite expresar algoritmos de manera estructurada sin depender de la sintaxis de un lenguaje específico. Su uso facilita la organización del pensamiento lógico y la resolución de problemas antes de la implementación en un lenguaje de programación. Según diversas fuentes de ingeniería informática, el pseudocódigo es ampliamente utilizado en la enseñanza de algoritmos y estructuras de datos, ya que promueve el desarrollo del pensamiento computacional. Además, se emplea en la industria para diseñar y documentar soluciones antes de su codificación. Al combinar pseudocódigo con diagramas de flujo de datos (DFD), se obtiene una representación más clara de los procesos, lo que permite una mejor comprensión y optimización de los algoritmos.

Nos enfocaremos en dos tareas fundamentales que nos permitirán comprender y aplicar los conceptos de lógica algorítmica: el análisis de datos estadísticos básicos y el ordenamiento de números.

En primer lugar, exploraremos el análisis de datos estadísticos básicos, diseñando pseudocódigo para calcular la media, la mediana y la moda de un conjunto de números. Estas medidas nos proporcionan información valiosa sobre la tendencia central y la distribución de los datos. Para visualizar y validar nuestros algoritmos, utilizaremos diagramas de flujo de datos (DFD), herramientas visuales que nos permiten representar gráficamente el flujo de datos y las operaciones de nuestros programas.

En segundo lugar, nos adentraremos en el mundo del ordenamiento, diseñando pseudocódigo para ordenar tres números ingresados por el usuario de menor a mayor. El ordenamiento es una operación fundamental en la programación y se utiliza en una amplia variedad de aplicaciones.

Para implementar y probar nuestros pseudocódigos, utilizaremos PSeInt, un software educativo diseñado para facilitar el aprendizaje de la programación. PSeInt nos permite ejecutar nuestros algoritmos paso a paso, visualizar el flujo de ejecución y detectar posibles errores.

A lo largo de la práctica, prestaremos especial atención a la claridad y precisión de nuestro pseudocódigo, así como a la correcta aplicación de las estructuras de control. Utilizaremos DFDs para complementar el pseudocódigo, como herramienta para la visualización de los algoritmos de estadística básica y la agrupación de números.

Desarrollo

Es necesario mencionar que los dos problemas inicialmente planteados en prácticas que son:

- obtener el promedio de “n” números, su mediana y su moda.
- Ordenar 3 números de menor a mayor ingresados por el usuario.

Estos dos fueron ajustados a un menú con opciones para el usuario.

Establecer una función para validar que la entrada sea un número entero positivo.

```
1 Funcion esEntero ← ValidarEntero(entrada)
2   Definir i Como Entero
3   esEntero ← Verdadero
4
5   Si Longitud(entrada) = 0 Entonces
6     Escribir "Error: No se ingresó ningún valor. Intente nuevamente."
7     esEntero ← Falso
8   Sino
9     Para i ← 0 Hasta Longitud(entrada) - 1 Hacer
10      Si No (SubCadena(entrada, i, i) ≥ "0" Y SubCadena(entrada, i, i) ≤ "9") Entonces
11        Escribir "Error: Solo se permiten números enteros positivos. Intente nuevamente."
12        esEntero ← Falso
13      FinSi
14    FinPara
15  FinSi
16 FinFuncion
```

Esta función sirve para:

- Es necesario esta función ya que el programa solo es apto para números enteros positivos, si el usuario ingresa algún otro valor que no cumpla con esta condición, se imprimirá un mensaje indicando error
- Si el usuario ingresa alguna letra, o carácter especial se imprimirá en pantalla un mensaje indicando “error: solo se permiten números enteros positivos. Intente nuevamente”
- Si el usuario ingresa una entrada vacía se imprimirá en pantalla un mensaje indicando “Error: no se ingresó ningún valor. Intente nuevamente”.

Definir el menú de opciones para el usuario

```
19 Proceso MenuPrincipal
20   Definir opcion Como Caracter
21   Definir salir Como Logico
22   salir ← Falso
23   Repetir
24     Escribir "Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de
25     Escribir "Menú Principal"
26     Escribir "Seleccione una opción (a/b/c): "
27     Escribir "a) Ordenar tres números de menor a mayor"
28     Escribir "b) Calcular promedio, moda y mediana"
29     Escribir "c) Salir"
30     Leer opcion
```

El menú sirve para:

- Inicialmente se muestra un mensaje de bienvenida y presenta un menú con tres opciones: ordenar números / calcular promedio, mediana, moda / salir.
- El usuario debe ingresar una opción.
- El ciclo se repetirá (es decir, el menú se mostrará nuevamente) hasta que el usuario seleccione la opción de salir

Establecer los “casos”

```
32 Segun opcion Hacer
33     Caso "a":
34         OrdenarNumeros(1)
35     Caso "b":
36         CalcularEstadisticas(2)
37     Caso "c":
38         salir ← Verdadero
39     De Otro Modo:
40         Escribir "Error: Opción inválida. Intente nuevamente."
41 FinSegun
```

Establecer los casos sirve para:

- Esto sigue estando dentro del menú en el código
- Según sea el caso de que seleccione una opción se establecen subprocesos que ejecutara el programa si se selecciona esta opción.
- Si el usuario ingresa alguna otra opción que no sea “a”, “b”, “c”, se imprimirá un mensaje indicando error y solicitando que la ingrese nuevamente.

Preguntar al usuario si luego de haber hecho una operación, desea volver al menú o salir

```
44 Si salir = Falso Entonces
45     Repetir
46         Escribir "¿Desea volver al menú principal? (S/N): "
47         Leer opcion
48         Hasta Que opcion = "S" O opcion = "N" O opcion = "s" O opcion = "n"
49
50     Si opcion = "N" O opcion = "n" Entonces
51         salir ← Verdadero
52     FinSi
53 FinSi
54 Hasta Que salir
55
56 Escribir "Gracias por usar el programa. ¡Hasta luego!"
57 FinProceso
```

Esto sirve para:

- Al finalizar la ejecución de la opción indicada se le pregunta al usuario si desea volver al menú principal.
- Si a esta pregunta el usuario ingresa algún otro valor que no sea “S”, “N”, “s”, “n”. aparecerá repetidamente la pregunta hasta que ingrese alguna de las opciones mencionadas.
- El programa terminara de ejecutarse cuando el usuario presione la opción salir.
- Finalmente se muestra un mensaje agradeciendo por usar el programa, esperando que el usuario vuelva a usarlo en el futuro.

Establecer los subprocessos que se definieron en el menú : (ordenar 3 números de menor a mayor, ingresados por el usuario)

```
60 SubProceso OrdenarNumeros(1)
61   Definir a, b, c, aux Como Entero
62   Definir entrada Como Cadena
63
64   Escribir "Ingrese el primer número (solo enteros positivos):"
65   Repetir
66     Leer entrada
67   Hasta Que ValidarEntero(entrada)
68   a ← ConvertirANumero(entrada)
69
70   Escribir "Ingrese el segundo número (solo enteros positivos):"
71   Repetir
72     Leer entrada
73   Hasta Que ValidarEntero(entrada)
74   b ← ConvertirANumero(entrada)
75
76   Escribir "Ingrese el tercer número (solo enteros positivos):"
77   Repetir
78     Leer entrada
79   Hasta Que ValidarEntero(entrada)
80   c ← ConvertirANumero(entrada)
81
82   Escribir "Los números antes de ordenar son: ", a, ", ", b, ", ", c
83   Repetir
84     Si a > b Entonces
85       aux ← a
86       a ← b
87       b ← aux
88     FinSi
89     Si b > c Entonces
90       aux ← b
91       b ← c
92       c ← aux
93     FinSi
94   Hasta Que a < b Y b < c
95
96   Escribir "Los números ordenados son: ", a, ", ", b, ", ", c
97 FinSubProceso
98
```

Esto subprocesso para:

- a, b, c s son las tres variables donde se almacenarán los números ingresados por el usuario y se definen del tipo “entero”.
- aux: es una variable auxiliar que sirve para ordenar los números de menor a mayor, y también se define del tipo entero.
- La “entrada” se define de tipo Cadena para que se pueda almacenar la entrada del usuario antes de convertirla a un número entero.
- Se le solicita al usuario que ingrese tres números (uno por uno), para que el programa pueda hacer las verificaciones pertinentes, y dado sea el caso de error poder corregirlo
- Cuando el valor es válido, se convierte de “Cadena” a “Entero” y se almacena en las variables.
- Después de que el usuario ingresa los tres números, el programa los muestra en pantalla sin ordenarlos. Lo que permite al usuario ver los valores ingresados.
- Se uso el método burbuja para ordenar los números, donde:
 - Se compara a con b, si a es mayor que b, los valores de a y b se intercambian utilizando la variable auxiliar
 - Luego, se compara b con c, y si b es mayor que c, los valores de b y c se intercambian de la misma manera.
 - El proceso se repite haciendo intercambios hasta que los números estén ordenados de menor a mayor.
 - Una vez que el ciclo de ordenación termina, es decir, cuando $a < b$ y $b < c$, el programa muestra un mensaje en pantalla ya con los números ordenados

Establecer los subprocessos que se definieron en el menú (obtener promedio)

```
101 SubProceso CalcularEstadisticas(2)
102   Definir cantidad, i, j, aux Como Entero
103   Definir suma, mediana Como Real
104   Definir numeros Como Real
105   Definir maxFrecuencia Como Entero
106   Definir entrada Como Cadena
107   Repetir
108     Escribir "Ingrese la cantidad de números (solo enteros positivos y distinto de 0):"
109     Leer entrada
110     Si ValidarEntero(entrada) Entonces
111       cantidad ← ConvertirANumero(entrada)
112       Si cantidad = 0 Entonces
113         Escribir "Error: La cantidad de números no puede ser 0. Intente nuevamente."
114       FinSi
115     FinSi
116   Hasta Que ValidarEntero(entrada) Y cantidad > 0
117
118   Dimension numeros[cantidad]
119   suma ← 0
120
121   Para i ← 0 Hasta cantidad - 1 Hacer
122     Repetir
123       Escribir "Número ", i + 1, " (solo números enteros positivos):"
124       Leer entrada
125       Hasta Que ValidarEntero(entrada)
126
127       numeros[i] ← ConvertirANumero(entrada)
128       suma ← suma + numeros[i]
129     FinPara
130
131   suma ← suma / cantidad
132   Escribir "El promedio es: ", suma
133
```

Esto subprocesso para:

- Se indica el segundo subprocesso que es el encargado de calcular el promedio, mediana y moda.
- Se definen las variables
- Se le pide al usuario que ingrese la cantidad de números que desea ingresar. Si el valor es 0 o no es un número entero positivo, se muestra un mensaje de error y se vuelve a pedir la entrada.
- Se define la variable “números” como un arreglo que sirve para almacenar los números que el usuario ingrese, y “suma” como variable para calcular el promedio
- El programa pide al usuario que ingrese cada número (solo números enteros positivos). Si el valor ingresado no es un número válido, se le pide nuevamente hasta que ingrese un valor correcto.
- Cada número ingresado se agrega al arreglo “números” y se acumula en la variable “suma” para posteriormente calcular el promedio.
- El promedio se calcula dividiendo la suma total de los números entre la cantidad de números ingresados.
- Finalmente se muestra un mensaje en pantalla al usuario indicando cual es el promedio obtenido

Ordenar los números de menor a mayor para la mediana

```
135 Para i ← 0 Hasta cantidad - 2 Hacer
136   Para j ← 0 Hasta cantidad - 2 - i Hacer
137     Si numeros[j] > numeros[j + 1] Entonces
138       aux ← numeros[j]
139       numeros[j] ← numeros[j + 1]
140       numeros[j + 1] ← aux
141     FinSi
142   FinPara
143 FinPara
```

Esto sirve para:

- Se uso el método burbuja para ordenar los números de menor a mayor, este método recorre la lista varias veces, comparando pares de números adyacentes y variable auxiliar si están en el orden incorrecto. Con cada pasada, el número más grande se mueve hacia el final de la lista ordenada, por lo que en cada iteración se hacen menos comparaciones. Se repite el proceso hasta que toda la lista quede ordenada. Esto permite que la mediana se calcule correctamente, ya que requiere que los datos estén en orden ascendente

Calcular la mediana

```
146 Si cantidad MOD 2 = 1 Entonces
147     mediana ← numeros[Trunc(cantidad / 2)]
148 Sino
149     mediana ← (numeros[Trunc(cantidad / 2)] + numeros[Trunc(cantidad / 2) - 1]) / 2
150 FinSi
151 Escribir "La mediana es: ", mediana
```

Esto sirve para:

- Para este punto los números ya están ordenados
- Se usa la MOD 2 = 1 para verificar si la cantidad de números es impar, ya que esto indicara que hay un valor central en el listado ordenado.
- Hay dos clases de mediana:
 - Si la cantidad es impar , la mediana es el número en la posición Trunc(cantidad / 2) de la lista ordenada.
 - la cantidad es par , la mediana se calcula promediando los dos números centrales, ubicados en las posiciones Trunc(cantidad / 2) y Trunc(cantidad / 2) - 1 .
- Finalmente el resultado de la mediana con el mensaje "La mediana es: "

Calcular la moda

```
154 maxFrecuencia ← 0
155 Dimension frecuencia[cantidad]
156
157 Para i ← 0 Hasta cantidad - 1 Hacer
158     frecuencia[i] ← 0
159 FinPara
160
161 Para i ← 0 Hasta cantidad - 1 Hacer
162     Para j ← 0 Hasta cantidad - 1 Hacer
163         Si numeros[i] = numeros[j] Entonces
164             frecuencia[i] ← frecuencia[i] + 1
165         FinSi
166     FinPara
167
168     Si frecuencia[i] > maxFrecuencia Entonces
169         maxFrecuencia ← frecuencia[i]
170     FinSi
171 FinPara
172
173 Si maxFrecuencia = 1 Entonces
174     Escribir "No hay moda"
175 Sino
176     Escribir "Las modas son:"
177     Para i ← 0 Hasta cantidad - 1 Hacer
178         Si frecuencia[i] = maxFrecuencia Entonces
179             Escribir numeros[i]
180         FinSi
181     FinPara
182     Escribir "Con una frecuencia de: ", maxFrecuencia
183 FinSi
184
185 Escribir "Ha finalizado, gracias por usar el programa."
186 FinSubProceso
```

Esto sirve para:

- Inicializa la variable "maxFrecuencia" en 0 para llevar el control de la mayor frecuencia encontrada.
- Se establece un arreglo "frecuencia" del mismo tamaño que la cantidad de números ingresados .
- Se inicializan todos los valores de "frecuencia" en 0 para comenzar el conteo desde cero.
- Se cuenta cuántas veces aparece cada número en la lista comparándolo con todos los demás números.
- Se actualiza "maxFrecuencia" si encuentra una frecuencia mayor a la almacenada.
- Los casos de la moda son:
 - Si "maxFrecuencia" es 1 , significa que no hay moda porque todos los números son únicos.
 - Si hay moda(s), se imprimen los números que aparecen con mayor frecuencia y se indica la frecuencia con la que se repiten.
- Finaliza el proceso mostrando un mensaje de despedida .

Ejecución

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
>
```

- Al iniciar la ejecución aparece el mensaje de bienvenida indicando lo que hace el programa
- Se muestra el menú de opciones
- Se incluye la opción de salir en el menú.

Ejecución de ordenar números (si el usuario ingresa todo de manera correcta)

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
> 3
Ingrese el segundo número (solo enteros positivos):
> 2
Ingrese el tercer número (solo enteros positivos):
> 100
Los números antes de ordenar son: 3,2,100
Los números ordenados son: 2,3,100
¿Desea volver al menú principal? (S/N):
> n
Gracias por usar el programa. ¡Hasta luego!
*** Ejecución Finalizada. ***
```

- Se selecciono la opción a que corresponde a ordenar 3 números
- En todos los ingresos de entrada se le indica al usuario que tienen que ser números enteros positivos
- Se muestran los números antes de ordenar (el orden en el que el usuario ingreso los números)
- Se muestran los números ya ordenados de menor a mayor
- Se le pregunta al usuario si desea volver al menu
- En esta prueba se dijo que no quería volver al menu, por lo tanto el programa dejo de ejecutarse y muestra un mensaje indicando que ya termino.

Ejecución de ordenar números (si el usuario ingresa entradas invalidas)

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> t
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> |
```

- El usuario ingreso una opción que no se encuentra del menu por lo tanto se muestra un mensaje de error.
- Se le pregunta si desea volver al menu principal (en este caso se selecciono s)


```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> t
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> s
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
>
```

- El usuario decidió volver el menu por lo tanto se vuelven a mostrar el menu

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> t
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> s
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
Error: No se ingresó ningún valor. Intente nuevamente.
> |
```

- El usuario ingreso una entrada vacía, por lo tanto aparece un mensaje de error, indicando que no se ingreso ningún valor y diciendo que lo intente de nuevo

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> t
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> s
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
Error: No se ingresó ningún valor. Intente nuevamente.
> 3
Ingrese el segundo número (solo enteros positivos):
> A
Error: Solo se permiten números enteros positivos. Intente nuevamente.
>
```

- El usuario ingreso una entrada correcta
- Para la otra entrada el usuario ingreso un elemento no numérico, por lo tanto se muestra un mensaje diciendo error indicando que solo se permiten números enteros positivos e intentar de nuevo

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> t
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> s
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
Error: No se ingresó ningún valor. Intente nuevamente.
> 3
Ingrese el segundo número (solo enteros positivos):
> A
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> +
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> |
```

- Se ingreso un carácter especial, por lo tanto se muestra un mensaje indicando error y que lo intente nuevamente
- Esto seguirá apareciendo hasta que el usuario ingrese una entrada correcta

```
PSeInt - Ejecutando proceso MENUPRINCIPAL
*** Ejecución Iniciada. ***
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
Error: No se ingresó ningún valor. Intente nuevamente.
> 3
Ingrese el segundo número (solo enteros positivos):
> A
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> +
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> 4
Ingrese el tercer número (solo enteros positivos):
> -7
Error: Solo se permiten números enteros positivos. Intente nuevamente.
>
```

- Igual muestra error en números negativos ya que solo acepta enteros positivos

```
PSelint - Ejecutando proceso MENUPRINCIPAL
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
Error: No se ingresó ningún valor. Intente nuevamente.
> 3
Ingrese el segundo número (solo enteros positivos):
> A
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> +
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> 4
Ingrese el tercer número (solo enteros positivos):
> -7
Error: Solo se permiten números enteros positivos. Intente nuevamente.
> 5
Los números antes de ordenar son: 3,4,5
Los números ordenados son: 3,4,5
¿Desea volver al menú principal? (S/N):
> n
Gracias por usar el programa. ¡Hasta luego!
*** Ejecución Finalizada. ***
```

- Pese a todas las entradas incorrectas se muestran los números que son correctos y esta leyendo el programa
- Finalmente los ordena
- En este caso no se desea volver al menu
- La ejecución finaliza

Ejecución de calcular estadísticas (si el usuario ingresa todo de manera correcta)

```
PSelint - Ejecutando proceso MENUPRINCIPAL
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> b
Ingrese la cantidad de números (solo enteros positivos y distinto de 0):
> 4
Número 1 (solo números enteros positivos):
> 2
Número 2 (solo números enteros positivos):
> 1
Número 3 (solo números enteros positivos):
> 5
Número 4 (solo números enteros positivos):
> 4
El promedio es: 3
La mediana es: 3
No hay moda
Ha finalizado, gracias por usar el programa.
¿Desea volver al menú principal? (S/N):
> n
Gracias por usar el programa. ¡Hasta luego!
*** Ejecución Finalizada. ***
```

- Se selecciono la opción b que corresponde a calcular estadísticas
- En todos los ingresos de entrada se le indica al usuario que tienen que ser números enteros positivos
- Aparece el resultado del promedio, mediana, moda
- Se le pregunta al usuario si desea volver al menu
- En esta prueba se dijo que no quería volver al menu, por lo tanto el programa dejo de ejecutarse y muestra un mensaje indicando que ya termino.

Ejecución de ordenar números (si el usuario ingresa entradas invalidas)

```
PSeint - Ejecutando proceso MENUPRINCIPAL
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> b
Ingrese la cantidad de números (solo enteros positivos y distinto de 0):
> 0
Error: La cantidad de números no puede ser 0. Intente nuevamente.
Ingrese la cantidad de números (solo enteros positivos y distinto de 0):
> 3
Número 1 (solo números enteros positivos):
> 43
Número 2 (solo números enteros positivos):
> 1
Número 3 (solo números enteros positivos):
> 19
El promedio es: 21
La mediana es: 19
No hay moda
Ha finalizado, gracias por usar el programa.
¿Desea volver al menú principal? (S/N):
> n
Gracias por usar el programa. ¡Hasta luego!
*** Ejecución Finalizada. ***
```

- Las restricciones son las mismas que en el subprocesso anterior, también involucra entradas vacías, entradas incorrectas como letras, caracteres especiales, números negativos.
- Este subprocesso añade la restricción que la cantidad de números tiene que ser diferente de 0

```
PSeint - Ejecutando proceso MENUPRINCIPAL
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> B
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> S
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> A
Error: Opción inválida. Intente nuevamente.
¿Desea volver al menú principal? (S/N):
> S
Bienvenido, este programa te puede ayudar a ordenar 3 números de menor a mayor y obtener promedio, mediana y moda de n números
Menú Principal
Seleccione una opción (a/b/c):
a) Ordenar tres números de menor a mayor
b) Calcular promedio, moda y mediana
c) Salir
> a
Ingrese el primer número (solo enteros positivos):
>
```

- Otra restricción es que solamente las opciones del menú solo son válidas en minúsculas.

Conclusiones particulares

Integrante 1: Fryda Itzel Bastida Reyes

Al desarrollar esta práctica de pseudocódigo, pude reforzar la lógica de programación mediante la implementación de un menú interactivo con dos opciones clave: el ordenamiento de tres números de menor a mayor y el cálculo de estadísticas (promedio, mediana y moda). Este ejercicio no solo permitió aplicar estructuras condicionales y ciclos, sino que también facilitó la comprensión de cómo descomponer un problema en pasos más pequeños y manejables.

El proceso de escribir el pseudocódigo resultó útil para planificar la lógica del programa antes de su implementación. Además, incluir comentarios facilitó la comprensión del flujo del algoritmo y su propósito en cada paso.

En conclusión, el uso de pseudocódigo resultó ser una herramienta útil para organizar ideas antes de implementarlas en un lenguaje de programación real. A través de esta práctica, fortalecí mi capacidad de análisis y resolución de problemas, lo que sin duda será beneficioso en futuros proyectos, por lo tanto puedo decir que esta práctica fue de utilidad.

Integrante 2: Jesús Daniel Andrade Mendoza

En conclusión, esta práctica me ha sido de gran utilidad porque me ha proporcionado una base sólida en lógica algorítmica. Ahora me siento más seguro para abordar problemas de programación más complejos y diseñar soluciones eficientes. Estoy ansioso por seguir explorando el mundo de la programación y aplicar los conocimientos que he adquirido en esta práctica.

Conclusion General

En esta práctica de laboratorio, hemos explorado los fundamentos de la lógica de programación a través de la resolución de problemas prácticos relacionados con el análisis de datos y el ordenamiento. La capacidad de analizar datos y organizar información es esencial en cualquier campo, y la programación nos proporciona las herramientas para automatizar estos procesos.

Ocupando los conocimientos acerca de algoritmos y resolución de problemas hemos aplicado principios similares para diseñar algoritmos claros y eficientes utilizando pseudocódigo. El pseudocódigo nos ha permitido enfocarnos en la lógica de nuestros programas antes de preocuparnos por la sintaxis de un lenguaje de programación específico, facilitando la planificación y el diseño de nuestras soluciones.

Hemos utilizado estructuras de control condicionales e iterativas para implementar la lógica necesaria en nuestros programas, aprendiendo a tomar decisiones y repetir tareas de manera eficiente.

A lo largo de la práctica, hemos prestado especial atención a la claridad y precisión de nuestro pseudocódigo, consultando recursos como estructuras de control de flujo en Python y estructuras de control en C para reforzar nuestros conocimientos.

Referencias

- Flores, A. (2025, 21 de febrero). Fundamentos de programación (Clase). Escuela Nacional de Estudios Superiores Unidad Juriquilla.
- Bahit, E. (2022, 17 agosto). Estructuras de control de flujo. Python Aplicado. <https://www.pythonaplicado.com/estructuras-de-control>
- Curso, G. D. I.-. (s. f.). Estructuras de control iterativas. <https://webs.um.es/ldaniel/iscyp17-18/13-estructuraRepetitiva.html>
- Estructura de control condicional compuesta (if-else). (2023, 28 marzo). Portal Académico del CCH. <https://portalacademico.cch.unam.mx/cibernetica2/estructuras-condicionales/condicional-compuesta>
- Estructura de control condicional simple (if). (2023, 28 marzo). Portal Académico del CCH. <https://portalacademico.cch.unam.mx/cibernetica2/estructuras-condicionales/condicional-simple>
- ESTRUCTURA DE CONTROL DEL ALGORITMO. (s. f.). Issuu. https://issuu.com/rodrigo-alfonzo-oliveira-gonzalez/docs/tecnologia_unimar_-_rodrigo_oliveira/s/12868065
- Estructuras de control. (2021, 21 octubre). Página del Profe Luis Alvarado. <https://profeluisalvarado.wordpress.com/estructuras-de-control/>
- Hernández, M. C. (s. f.). Estructuras de control - Curso de C. <http://cuartas.es/c/estructuras-control/>
- Tit, M. L. M. P. (s. f.). 3 Estructuras de control | Introducción a la Programación. <https://mpru.github.io/introprog/estructuras-de-control.html>