



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

Ing. Aeroespacial

Segundo semestre

Santiago de Querétaro, Qro. 27 de febrero del 2025

**Trabajo:
Práctica 4
“Pseudocódigos”**

**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:

Observaciones:

INTRODUCCIÓN

En esta práctica realizamos un pseudocódigo en el software “PSeInt” que representa la solución algorítmica de un problema en el cual requiere el uso de la estructura de control de flujo condicional. El pseudocódigo es una forma de escribir los pasos de un algoritmo de manera sencilla y entendible antes de convertirlo en código. El pseudocódigo es una descripción informal de un algoritmo o un programa, escrito en un lenguaje natural estructurado que imita la lógica del código pero sin seguir las reglas estrictas de la sintaxis de un lenguaje de programación. Su propósito principal es mostrar el flujo lógico de un programa o sistema de manera que cualquier persona, con o sin experiencia en programación, pueda entender los pasos que realiza el algoritmo. A diferencia del código real, el pseudocódigo no está destinado a ser ejecutado por un ordenador, sino que actúa como un puente entre la idea y el código final. Es una forma de planificar el proceso antes de lidiar con los detalles técnicos y sintácticos de un lenguaje de programación concreto.

. Primero analizamos el problema en el que definimos los datos de entrada y salida, y luego se diseña una solución eficiente. El pseudocódigo nos ayuda a organizar las ideas y facilita la programación, ya que usa una estructura clara que cualquier persona con conocimientos básicos puede entender. El pseudocódigo tiene reglas que ayudan a que sea claro y fácil de entender.

Estructura del programa: Todo pseudocódigo empieza con INICIO y termina con FIN. Dentro de esto van las instrucciones. Palabras en mayúsculas: Las palabras clave como LEER y ESCRIBIR siempre van en mayúsculas. Sangría o tabulación: Se usa para organizar mejor el código y hacerlo más fácil de leer. Entrada y salida de datos: LEER sirve para pedir datos al usuario. ESCRIBIR muestra mensajes o resultados en pantalla.

Los tipos de datos que podemos seleccionar en el pseudocódigo son: ENTERO: números enteros (+/-). REAL: números con decimales. BOOLEANO: solo puede ser VERDADERO o FALSO. CHARACTER: un solo símbolo o letra y CADENA: un conjunto de caracteres (texto). El pseudocódigo tiene varias reglas y estructuras que ayudan a organizar un algoritmo antes de programarlo. Para la declaración de variables se pueden declarar varias variables del mismo tipo usando arreglos, especificando la cantidad. También existen los registros, que agrupan diferentes tipos de datos en una misma variable. Las estructuras de control pueden ser secuenciales en las que las instrucciones se ejecutan una tras otra en el orden en que están escritas, condicionales que permiten ejecutar ciertas instrucciones dependiendo de una condición y repetitivas o iterativas estas últimas ejecutan un conjunto de instrucciones varias veces según una condición. Estas reglas hacen que el pseudocódigo sea claro, ordenado y permiten que un conjunto de instrucciones se ejecute varias veces mientras se cumpla una condición.

Uno de estos ciclos es MIENTRAS, que primero revisa si la condición es verdadera. Si lo es, ejecuta las instrucciones dentro del ciclo y vuelve a comprobar la condición. Esto se repite hasta que la condición se vuelve falsa, momento en el que el ciclo

termina y el programa sigue con el resto del código. Básicamente, MIENTRAS sigue ejecutando las instrucciones una y otra vez mientras la condición se cumpla. Pero si desde el inicio la condición es falsa, el ciclo nunca se ejecuta.

Las estructuras de repetición son también llamadas estructuras cíclicas, iterativas o de bucles. Estas estructuras permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica), mientras que una expresión lógica evalúa que se cumpla, es decir, que sea verdadera. Son estructuras fundamentales en la programación que permiten ejecutar repetidamente un bloque de código mientras se cumpla una condición específica. Esto evita la necesidad de escribir múltiples veces las mismas instrucciones, facilitando tareas repetitivas y mejorando la eficiencia y legibilidad del código.

Importancia de los ciclos en programación:

- Eficiencia y productividad: Al automatizar tareas repetitivas, los bucles reducen la cantidad de código necesario y minimizan errores humanos asociados a la repetición manual de instrucciones.
- Manejo de estructuras de datos: Son esenciales para recorrer y manipular estructuras como arreglos, listas o matrices, permitiendo operaciones como búsqueda, ordenamiento y modificación de datos.
- Control del flujo del programa: Los bucles proporcionan mecanismos para ejecutar ciertas acciones múltiples veces bajo condiciones específicas, lo cual es crucial para implementar algoritmos complejos y responder a eventos en tiempo real.

Áreas donde los bucles son ampliamente utilizados:

- Desarrollo de software: En aplicaciones de escritorio, móviles y web, los bucles gestionan tareas como la actualización de interfaces, procesamiento de entradas de usuario y manejo de eventos.
- Procesamiento de datos: Para analizar grandes volúmenes de información, los bucles permiten iterar sobre conjuntos de datos, aplicar transformaciones y extraer conocimientos relevantes.

El pseudocódigo es clave para describir la lógica de un programa de forma clara. Permite expresar ideas sin preocuparse por la sintaxis de un lenguaje específico, lo que facilita que personas con conocimientos limitados de programación entiendan el flujo del programa, también facilita la planificación del código. Los desarrolladores pueden definir los pasos del algoritmo antes de escribir código real, lo que ayuda a evitar errores de diseño.

DESARROLLO

En esta práctica se realizó el pseudocódigo basado en los diagramas de flujo que se trabajaron en la práctica anterior, y si se pudieron traducir literalmente de dfd a pseint, manteniendo su funcionamiento, hicimos algunos cambios pero estos solo fueron para cumplir los nuevos requisitos que tenía la práctica, cómo hacer que se repitan hasta que el usuario quiera salir.

Una de las cosas adicionales que tuvimos que hacer esta práctica fue Añadir un menú que permitiera que el usuario elija la operación que desea realizar, y que se repita hasta que el usuario desee salir.

Para conseguirlo se usaron dos elementos, un ciclo mientras que repite el algoritmo a menos que el usuario seleccione C, la opción correspondiente a la salida. Y también la función según, que nos permite ejecutar acciones distintas dependiendo de una variable, la cual en este caso es seleccionada por el usuario.

Proceso Mediamoda

```
Definir eleccion,decision Como caracter
Definir s, can, moda, a2, max, col, prom, j, num, A, p, q, i,W,conmod Como Real
Definir x1, x2, y1, y2 Como Real
Definir Distancia Como Real
Definir elec,existe como Logico
Escribir 'este programa te permitirá realizar diversas operaciones'
Mientras (eleccion≠('C')) Hacer
    Escribir 'A. Media y moda ', 'B. Distancia entre dos puntos ', 'C.Salir'
    Escribir 'Elige una opción:'
    Leer eleccion
    Según eleccion Hacer
```

Continuando con el menú se definió una secuencia de acciones a realizar dependiendo de la elección del usuario, siendo A: media y moda, B: la distancia entre 2 puntos y C para salir, además de esto en el caso que la variable leída no coincida con ninguna opción se le indica al usuario que debe ingresar un valor distinto y repite el ciclo.

También se agregaron algunas opciones adicionales que pueden coincidir con la opción seleccionada, esto usando 'o', lo que le permite al usuario poder ingresar minúsculas, o el texto que mostramos en las opciones, y así evitar algunos inconvenientes para el usuario.

```
Según eleccion Hacer
'A'o'Media y moda'o'a|:
Escribir 'Este programa te ayuda a calcular la media y la moda de cualquier cantidad de números'
'B'o'Distancia entre dos puntos'o'b|:
Escribir 'Este programa te ayuda a calcular la distancia entre 2 puntos dados por coordenadas en X y Y'
```

```

'C' o 'Salir' o 'c':
    Escribir 'el programa terminará...'
    eleccion="C"
De Otro Modo:
    Escribir 'El valor no es valido, ingrese otro valor'
FinSegún
FinMientras

```

En la opción C cambiamos la elección del usuario a 'C' después de mostrar el mensaje, para que sin importar lo que elija pueda ser comparado en el ciclo mientras en el que está el proceso.

Ahora hablando un poco más de la elección A (la media y moda de n números), para esta también se realizaron ciertos cambios, principalmente dos, el primero fue que se implementó una opción para que el usuario pueda decidir ejecutar la operación de nuevo, sin necesidad de regresar al menú principal.

```

'A' o 'Media y moda' o 'a':
    Escribir 'Este programa te ayuda a calcular la media y la moda de cualquier cantidad de números'
    Mientras (decision≠('No')) Hacer
        Escribir '¿Cuál es la cantidad de números que desea ingresar?'
        Leer col
        matriz(col)
        elec=falso
        Mientras (elec=falso) Hacer
            Escribir '¿Desea repetir la operación?'
            Escribir 'Sí/No'
            leer decision
            Segun decision Hacer
                'sí' o 'Si' o 'Si':
                    elec=verdadero
                'no' o 'No':
                    Escribir 'Regresará a la selección de operaciones'
                    elec=verdadero
                    decision='No'
            De Otro Modo:
                Escribir 'El valor no es valido, ingrese otro valor'
        Fin Segun
    finmientras
finmientras

```

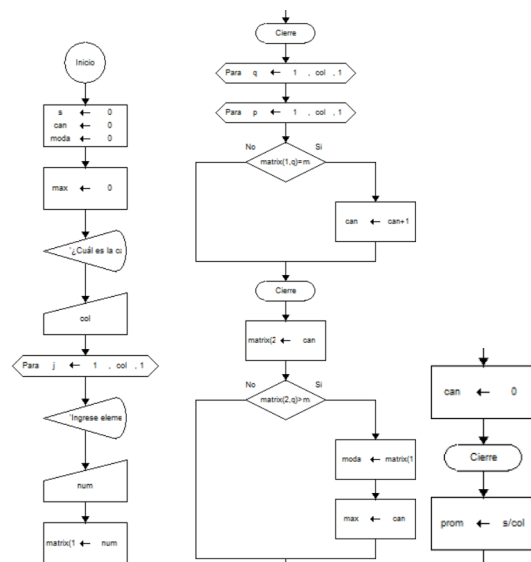
Para esto se utilizaron 2 ciclos mientras, el primero depende de decisión del usuario, la cual es ingresada al finalizar la operación, y mientras esta no sea 'No' la operación se realizará de nuevo, el segundo ciclo nos permite validar si la decisión del usuario está dentro de las opciones que contemplamos, este depende de una variable lógica llamada elec, y mientras la decisión del usuario no sea válida esta se mantendrá como falso, lo que repetirá la pregunta.

El segundo cambio que hicimos fue que ahora la operación se lleva a cabo en un subproceso, esto debido a que al estar dentro de un ciclo intentar dimensionar varias veces la misma matriz causaba errores, lo que se soluciona al ejecutarse dentro de un subproceso, al cual llamamos “matriz”

```

subproceso matriz(col)
    Dimensionar A(3,col)
    s ← 0
    can ← 0
    max ← 0
    Para j←0 Hasta col-1 Con Paso 1 Hacer
        Escribir 'Ingrese elemento: ', j+1
        Leer A[1,j]
        s ← s+A[1,j]
    FinPara
    Para q←0 Hasta col-1 Con Paso 1 Hacer
        Para p←0 Hasta col-1 Con Paso 1 Hacer
            Si A[1,q]=A[1,p] Entonces
                can ← can+1
            FinSi
        FinPara
        A[2,q]←can
        Si can>max Entonces
            moda ← A[1,q]
            max ← can
        FinSi
        can ← 0
    FinPara
    prom ← s/col

```



En este subproceso se repiten las cosas casi de la misma manera en que se realizaban en el dfd, pero aunque no fuera necesario se decidió implementar algo más, y es que en el dfd el proceso solo te daba una moda, aunque hubiera la misma cantidad de números.

```

Para i←0 Hasta col-1 Con Paso 1 Hacer
    Si A[2,i] = max Entonces
        Existe ← Falso
        Para W←0 Hasta conmod Hacer
            Si A[0,W] = A[1,i] Entonces
                Existe ← Verdadero
            FinSi
        FinPara
        Si Existe=Falso Entonces
            A[0,conmod]=A[1,i]
            conmod=conmod+1
        Fin Si
    FinSi
FinPara
Escribir 'El promedio es: ', prom

Escribir "La(s) moda(s) es/son: "
Para i←0 Hasta conmod-1 Con Paso 1 Hacer
    Escribir A[0,i], " "
FinPara

```



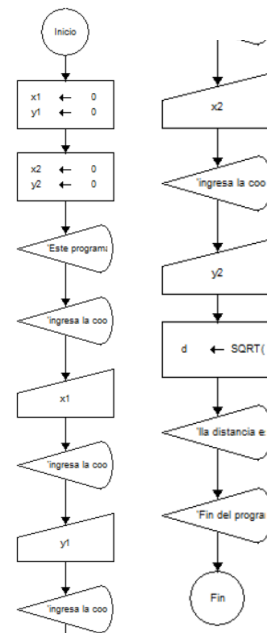
Para conseguir mostrar varias modas se necesitó, comparar las cantidades de cada número, obtenidas anteriormente, con el valor máximo y si estas son iguales registrar los datos del número al que pertenecen en una nueva columna de la matriz, y para evitar que el mismo número se registra múltiples veces creamos un ciclo para que compare todos los valores de la columna de modas con el valor del dato al que estamos analizando antes de que el número se registre. Si el número que se analiza ya está en la columna de modas, la variable lógica 'Existe' cambia a verdadera, lo que evita que se registre en la columna de moda, y si este no existe aún en las modas entonces lo registramos, además de aumentar en 1 el contador de cuantas modas tenemos, lo cual usamos en el ciclo, para saber cuántas modas comparar al valor y también a la hora de darle las modas al usuario.

En cuanto a la opción B (Distancia entre dos puntos), esta pudo mantenerse de forma prácticamente idéntica a como estaba en el dfd, solo añadiendo la misma opción que en el anterior para permitirle al usuario realizar la operación de nuevo si no desea regresar al menú

```

'B'o'Distancia entre dos puntos'o'b':
Escribir 'Este programa te ayuda a calcular la distancia entre 2 puntos dados por coordenadas en X y Y'
Mientras (decision#('No')) Hacer
    Escribir 'Ingrese la coordenada x1:'
    Leer x1
    Escribir 'Ingrese la coordenada x2:'
    Leer x2
    Escribir 'Ingrese la coordenada y1:'
    Leer y1
    Escribir 'Ingrese la coordenada y2:'
    Leer y2
    Distancia ← Raíz((x2-x1)²+(y2-y1)²)
    Escribir 'La distancia entre los dos puntos es: ', Distancia
    elec=falso
    Mientras (elec=falso) Hacer
        Escribir '¿Desea repetir la operación?'
        Escribir 'Sí/No'
        leer decision
        Segun decision Hacer
            'si' o 'sí' o 'Si' o 'Sí':
                elec=verdadero
            'no' o 'No':
                Escribir 'Regresará a la selección de operaciones'
                elec=verdadero
                decision='No'
        De Otro Modo:
            Escribir 'El valor no es valido, ingrese otro valor'
        Fin Segun
    finmientras
finmientras

```



y con la opción C solo le damos un mensaje al usuario para notificarle que el programa terminará.

```

'C'o'Salir'o'c':
    Escribir 'el programa terminará...'
    eleccion="C"
De Otro Modo:
    Escribir 'El valor no es valido, ingrese otro valor'
FinSegún
FinMientras
FinProceso

```

Los cambios implementados llevaron también a un cambio en las entradas y salidas, siendo que a las que ya se tenían en cuenta se añaden las entradas de la elección de operación, y la de decisión de si continuar o no el programa.

En cuanto a salidas se añaden varias salidas intermedias relacionadas a los menús:

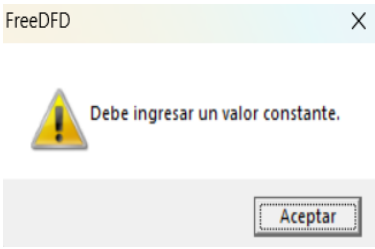
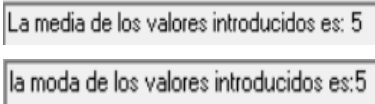
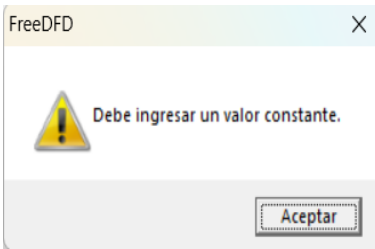
- 'este programa te permitirá realizar diversas operaciones'
- 'A. Media y moda ', 'B. Distancia entre dos puntos ', 'C.Salir'
- 'Elige una opción:'
- '¿Desea repetir la operación?'
- 'Sí/No'
- 'Regresará a la selección de operaciones'
- 'El valor no es válido, ingrese otro valor'
- 'El programa terminará...'
- Variable para validar (Válido)

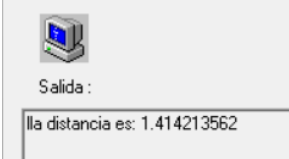

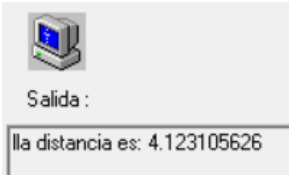
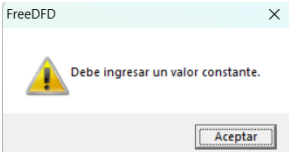
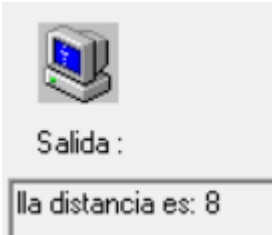
Además también tenemos algunas salidas relacionadas con los cambios en la media y la moda

- La moda ahora es la columna 0 de la matriz A, en vez de un único dato.
- Contador de moda (conmod)
- Variable para conocer si el valor ya existe (Existe)

En cuanto a pruebas de escritorio estas resultaron igual que en DFD

Iteración	Entrada	Salida	Salida DFD	Salida Pseint
1	1, 3, 3, 5, 3, 7	La moda de los valores introducidos es: 3 La media de los valores introducidos es: 3.67	La media de los valores introducidos es: 3.666666667 la moda de los valores introducidos es:3	El promedio es: 3.666666667 La(s) moda(s) es/son: 3 ¿Desea repetir la operación? Sí/No
2	1.5, 6, 8, 8, 8, 9, 13, 4	La moda de los valores introducidos es: 8 La media de los valores	La media de los valores introducidos es: 6.888888889 la moda de los valores introducidos es:8	El promedio es: 6.888888889 La(s) moda(s) es/son: 8 ¿Desea repetir la operación? Sí/No

		introducidos es: 6.88		
3	1, 2, a, #, 7	El valor introducido no pertenece a los números reales		Lin 63 (inst 1): ERROR 120: No coinciden los tipos (COL).
4	5	La moda de los valores introducidos es: 5 La media de los valores introducidos es: 5		El promedio es: 5 La(s) moda(s) es/son: 5 ¿Desea repetir la operación? Sí/No
5	4, 6, ?, f, %	El valor introducido no pertenece a los números reales		Lin 63 (inst 1): ERROR 120: No coinciden los tipos (COL).

Iteración	Entradas	Salidas	Salidas DFD	Salidas Pseint
1	x1, y1: 1,1 x2, y2: 2,2	1.4142	<p>Impresion por pantalla</p> 	<p>La distancia entre los dos puntos es: 1.4142135624 ¿Desea repetir la operación? Sí/No</p>
2	x1, y1: 1,1 x2, y2: 1,1	0		<p>La distancia entre los dos puntos es: 0 ¿Desea repetir la operación? Sí/No</p>
3	x1, y1: 5,7 x2, y2: 4,3	4.1231		<p>La distancia entre los dos puntos es: 4.1231056256 ¿Desea repetir la operación? Sí/No</p>
4	x1: at	"El valor es incorrecto, ingresa un valor diferente" Hace que el usuario reasigne un valor hasta detectar un valor adecuado.		<p>Lin 87 (inst 1): ERROR 120: No coinciden los tipos (X1).</p>
5	x1, y1: 1,0 x2, y2: 9,0	8		<p>La distancia entre los dos puntos es: 8 ¿Desea repetir la operación? Sí/No</p>

CONCLUSIONES

Matías Serrano- En mi opinión la práctica fue de utilidad debido a que me permitió explorar un programa que no conocía, y que permite visualizar el proceso de diferentes formas, lo cual me pareció conveniente. Además de esto me dio experiencia sobre cómo diferentes programas interpretan los mismos datos de formas distintas, y me dio la oportunidad de corregir y mejorar el código de la práctica anterior.

Edgar Reyes: Desde mi experiencia, fue una práctica interesante ya que es mi primera vez trabajando en un pseudocódigo y específicamente en PSeint, además que durante el desarrollo del pseudocódigo, fue interesante ver cómo cada parte del código tomaba forma y cómo las estructuras de control (condicionales y ciclos) ayudaban a que el programa funcionara de manera eficiente. Al principio, hubo algunos desafíos, especialmente en la implementación del cálculo de la moda pero al organizar bien la lógica, logramos que el programa fuera claro y funcional. En general, esta práctica fue muy desafiante y retadora para el equipo, pero también reforzó el pensamiento lógico y la estructura de un programa. Me sentí motivado al ver que el código funcionaba correctamente y permitía la interacción con el usuario de manera fluida. Fue un buen ejercicio para aplicar conceptos matemáticos en la programación y mejorar la forma en que estructuramos algoritmos en PSeInt.

REFERENCIAS

Robledano, A. (2024, 23 septiembre). Qué es pseudocódigo y por qué es esencial en programación. *OpenWebinars.net*.

<https://openwebinars.net/blog/que-es-pseudocodigo/>

Gálvez, J. A. S. (s. f.). *Estructuras de repetición*. Unidades de Apoyo Para el Aprendizaje - CUAIEED - UNAM.

<https://uapa.cuaed.unam.mx/sites/default/files/minisite/static/b62c06c7-6c36-48e9-9b56-9f164c1aaa76/UAPA-estructuras-repeticion/index.html>

Aurora. (2023, 18 enero). *¿Qué es un bucle en programación?* ID Digital School - Bootcamps.

<https://iddigitalschool.com/bootcamps/que-es-un-bucle-en-programacion/>