



UNIVERSIDAD NACIONAL AUTONOMA DE
MÉXICO CAMPUS JURIQUILLA



INGENIERIA AEROESPACIAL.

Práctica 4.

PSEUDOCODIGO.

Óscar Abraham Murillo Rivas -

425039773 Sergio Reséndiz

Cabrera - 425061336

Saul Emiliano Revoreda Gómez - 425007921

Fundamentos de programación.

Andrés David Flores Ferro

Semestre 2

Grupo 2002

Fecha de

entrega:

02/03/2025

Fecha de

revisión:

02/03/2025

Actividad en clase 18 de 25 pts

Protocolo de práctica	58 de 75 pts
Portada	10 de 10 pts
Introducción	15 de 15 pts
Desarrollo	15 de 25 pts
Conclusiones	10 de 15 pts
Referencias	08 de 10 pts

CALIFICACIÓN TOTAL Entrega 76 pts

Observaciones y calificación:

- Todavía tiene poco del contenido de clase, pero al menos ya rescataron un párrafo.
- Agregar dato de tipo de archivo consultado en referencias [Web,PDF, Libro, etc] , no se baja por esta razón si no porque falta citar algo del contenido presentado en clase.
- En el desarrollo solo se menciona el contenido pero no hay desarrollo de teoría de lo que se hizo o porque (en su caso algunos casos si pero no todos , especificación indica descripción paso a paso de lo realizado y de su porque.

PSEUDOCÓDIGOS

Introducción.

En clase se nos brindó herramientas para poder realizar códigos, en este caso, usamos Pse-int, se aprendió como usar la sintaxis de pseudocódigo, para indicar que es una lectura de datos se utiliza la etiqueta “LEER”. Para indicar una escritura de datos se utiliza la etiqueta “ESCRIBIR”, así como la declaración de distintos tipos de variables (entero, real, lógico, carácter, cadena), el uso de operadores aritméticos y lógicos (!= , <=,etc). Las herramientas más útiles para elaborar la práctica son las estructuras de control secuencial , condicional e iterativas, necesarias para hacer seguir realizando una serie de instrucciones mientras se cumpla alguna expresión lógica.

PSeInt que significa (Pseudo Intérprete), es un Entorno de Desarrollo Integrado para pseudocódigo, un “lenguaje de programación” imperativo simple y que es totalmente en el idioma español. Es decir, PSeInt es un editor e intérprete de programas escritos en pseudocódigo. Su interfaz gráfica permite crear, almacenar, ejecutar y corregir fácilmente programas en pseudocódigo. La sencillez del lenguaje pseudocódigo lo hacen ideal para la enseñanza de la programación. Mediante el uso de un lenguaje de pseudocódigo simple y limitado y una interfaz de usuario intuitiva en español. Los estudiantes pueden comenzar a comprender conceptos básicos de estructuras de control, apoyados en un paradigma de programación modular y procedimental. Además, esta herramienta permite el uso de diagramas de flujo, funciones y procedimientos con parámetros, y también se admiten arreglos multidimensionales .

El software comenzó a desarrollarse en octubre de 2003 utilizando Borland C++ Builder debido a que era esta la herramienta que recomendaba la cátedra por la facilidad que brinda para el desarrollo de interfaces visuales (Manso et al., 2019). Se comenzó con una prueba de concepto, ya que era el primer proyecto que iba a desarrollar con C++. A esta inexperiencia con el lenguaje, los creadores de la herramienta le atribuyen errores de diseño y muchas ineficiencias en la implementación que con el tiempo fueron corrigiendo. Al ser este un software para uso exclusivamente didáctico, no se presentan grandes problemas de rendimiento, puesto que no se tienen que interpretar algoritmos de miles de líneas sumamente

complejos.

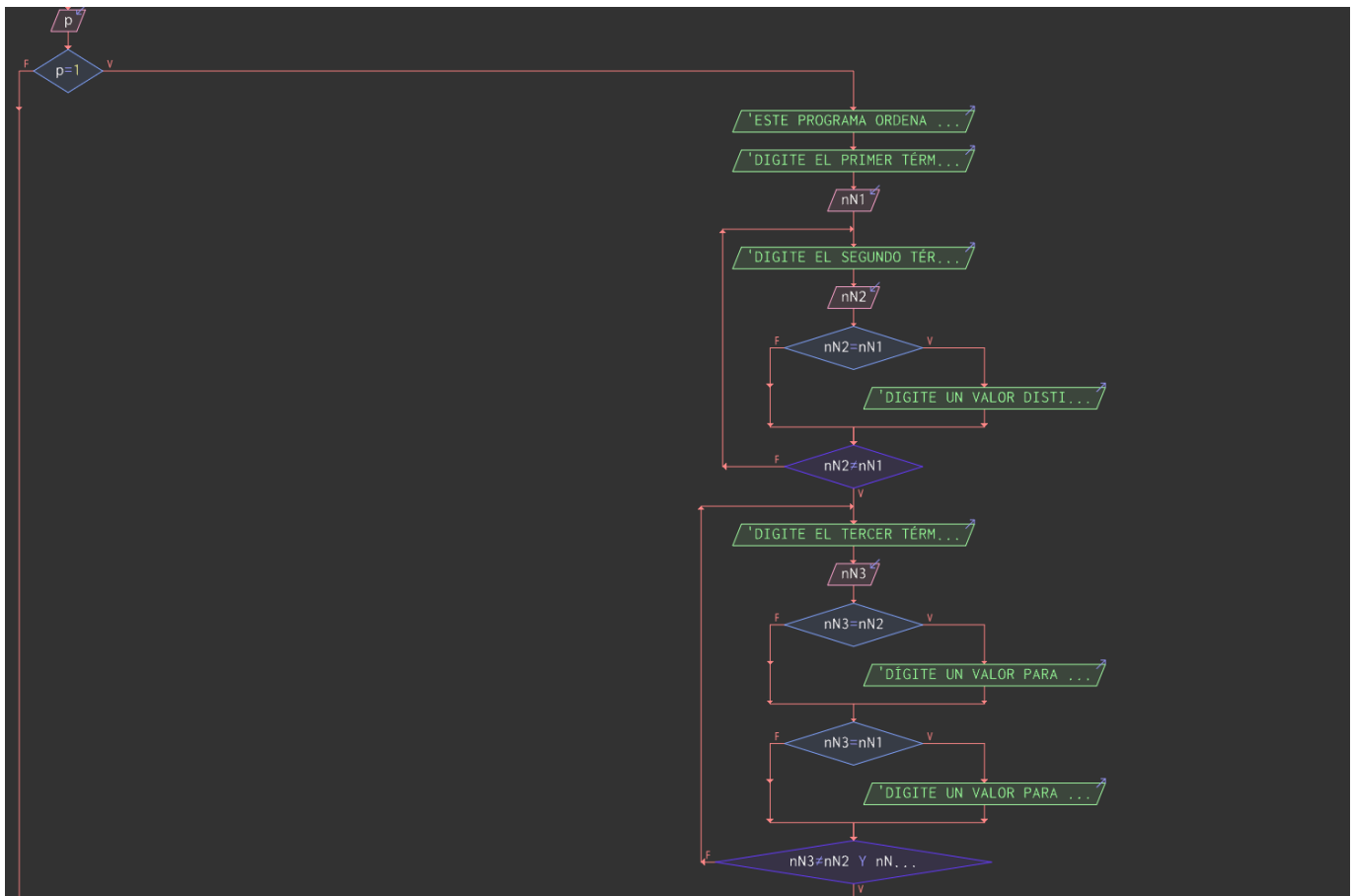
Desarrollo (Evidencias de práctica y desarrollo).

En la primera sección, definimos las variables que utilizaremos, definimos un total de para las dos operaciones, y una variable p para guardar la respuesta del usuario que tendrá que digitar en el menú de opciones, definimos las variables separadas para no confundirnos ya que el programa realiza dos operaciones y ambas con 3 números digitados por el usuario. En el inicio del código se aprecia (línea 16) el inicio de una estructura de control iterativa (repetir), que nos hará repetir ciertas instrucciones de código hasta que cierto condicional lógico se cumpla, el cual se encuentra hasta el final del programa.

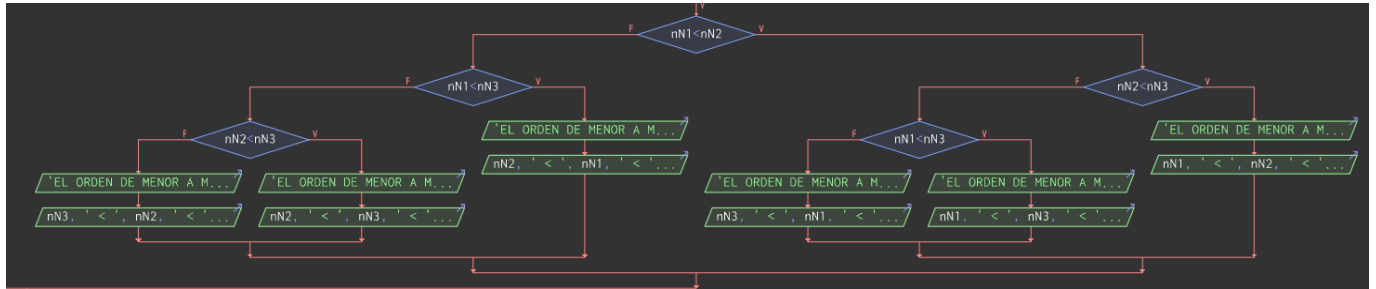
En esta primera sección el usuario tiene 3 opciones. Si elige la opción 1 el usuario digitara 3 números y éste los ordenará de menor a mayor. Si elige 2, el usuario tendrá que digitar 3 números para encontrar la media y su moda. Y la última opción 3 da la opción al usuario de cerrar el programa.

```
1  Algoritmo PRACTICA4
2
3  definir p como real
4
5
6  Definir nN1, nN2, nN3          como real
7
8  Definir N1 Como Real
9  Definir N2 Como Real
10 Definir N3 Como Real
11
12 Definir Media, moda Como Real
13
14
15
16 Repetir
17     Escribir ""
18     Escribir ""
19     Escribir " SELECCIONE LA OPERACIÓN QUE DESEA REALIZAR  "
20     Escribir ""
21     Escribir ""
22     Escribir "1) ORDENAR 3 NÚMEROS REALES DE MENOR A MAYOR  "
23     Escribir ""
24     Escribir ""
25     Escribir "2) CALCULAR LA MODA Y LA MEDIA DE 3 NUMEROS  "
26     Escribir ""
27     Escribir ""
28     Escribir "3) SALIR DEL PROGRAMA"
29     Escribir ""
30     Escribir ""
31     Escribir "DIGITE EL NÚMERO CORRESPONDIENTE PARA REALIZAR LA OPERACIÓN."
32     Escribir ""
33     Escribir ""
34 }
```

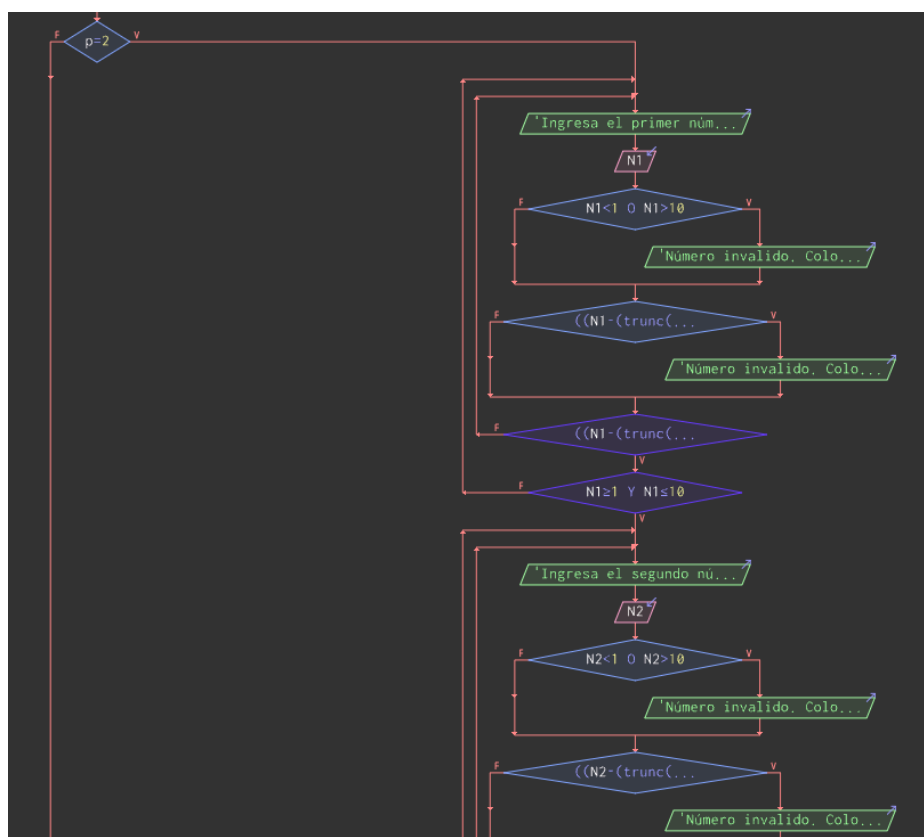
Usando, una herramienta bastante útil de Pseint, podemos ver en diagramas de flujo nuestro pseudocódigo. Aquí cuando el usuario ingresa 1 y se guarda en p, se despliega las opciones para pedir los datos al usuario, Donde se excluye la posibilidad de que el usuario repita los números, ya que no podría decirse que un mismo número es mayor o menor que sí mismo.



A continuación se aprecia en diagramas de flujo el uso de operadores lógicos para ordenar los números. Y también el mensaje de salida indicando el orden numérico. Como ya se realizó la operación para $p = 1$, el programa repetirá el proceso hasta que el usuario digite 3.

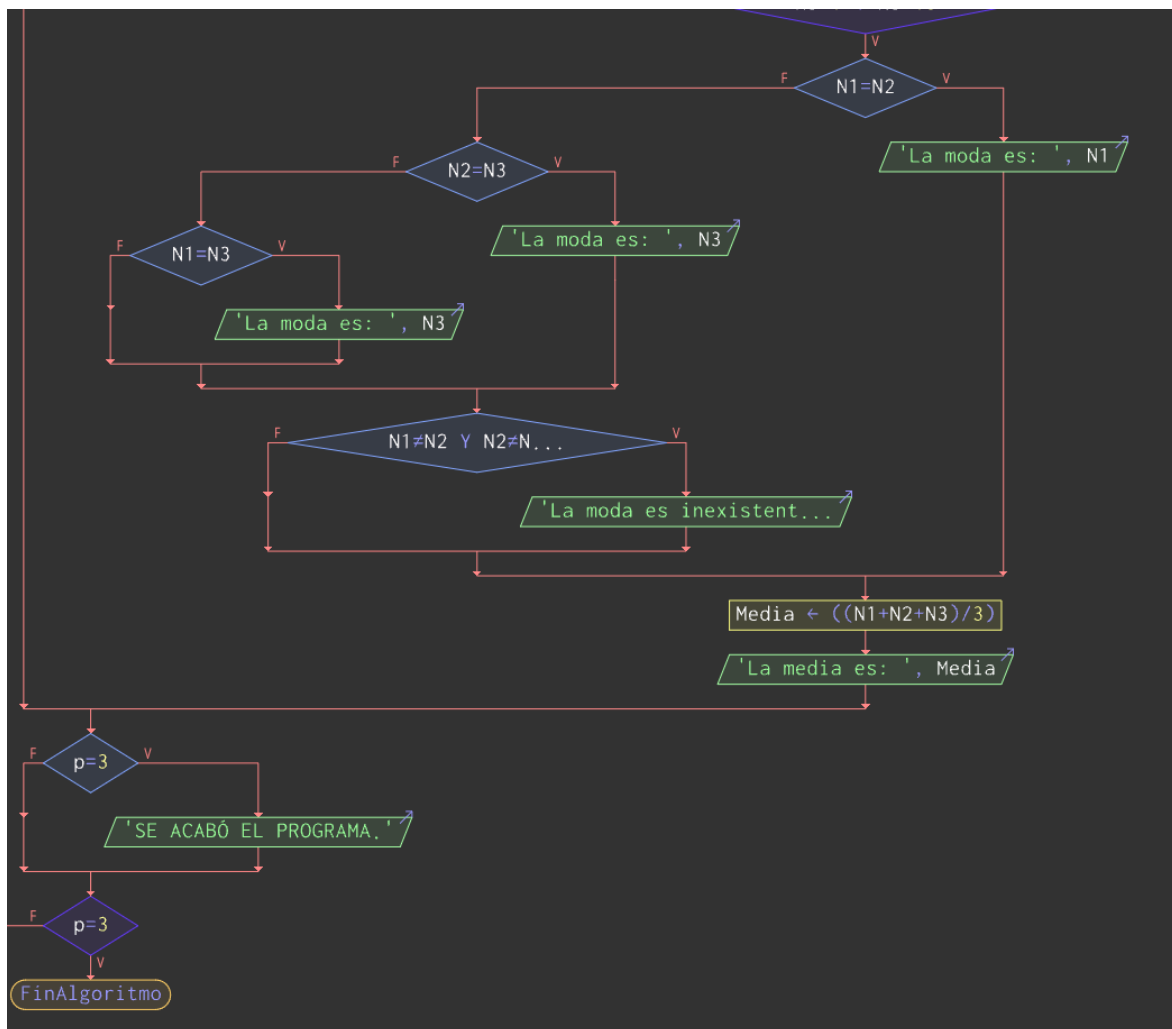


Ahora, cuando el usuario ingresa la opción 2, se evita que el usuario se salga del intervalo de 1 a 10, apoyándonos con la función trunc que nos ayuda a obtener la parte entera de un número, eliminando los decimales sin redondear el número. La cual no es bastante útil para que el usuario no ingrese valores que no sean enteros.



En la última parte del pseudocódigo, después de establecer las restricciones, primero vemos los condicionales lógicos para poder encontrar la moda y la media, y después se imprime con un mensaje de salida, que después nos llevaría al menú principal y continuará con la operación indefinidamente hasta que el usuario decida dejar de operar.

Al final vemos que cuando el usuario digita 3 el programa se cerrará ya que también se cumpliría el requerimiento lógico del ciclo repetir, y sólo arrojaría el mensaje de salida que cierra el programa.



Conclusiones.

Sergio: En la práctica del pseudocódigo pudimos ver diferentes maneras de aplicar lo hecho en el software dfd de una manera más práctica y entendible, siento yo que el programa pseint es mucho mejor en cuestión de hacer diagramas de flujo ya que conjunta esto con el pseudocódigo, logramos darle solución a nuestros problemas con las restricciones propuestas, Pseint es un programa muy completo.

Abraham: Esta práctica me ayudó a recordar muchas cosas de programación, ya que a pesar de ser pseudocódigo, es una herramienta bastante útil para entender la lógica de programación y entrar de lleno en ésta. Aún no sabíamos cómo hacer para que el usuario no digitara caracteres (cosa que vimos después) y por tanto, fue un poco frustrante la búsqueda de esta restricción teniendo la idea de hacerlo solo con diferentes estructuras de control y condicionales lógicos.

Saul: Esta práctica me estreso bastante, ya que había sido un día muy largo y aún así pudimos realizar lo poco mucho que entregamos ese día. El programa no lo podíamos dominar por más que intentábamos, el comando “!” (diferente de) no nos funcionaba, no sabemos si fue porque lo utilizamos mal o porque de plano no sirve el comando. Al final pudimos hacer algo para entregar ese día, ya con calma en nuestras casas logramos acomodar un poco los diferentes diagramas de flujo.

Referencias.

- Cairo, O. (2005). *Metodología de la programación*. México D.F. Alfaomega. 3a ed.
- Rodriguez, M.A. *Metodología de la programación a través de pseudocódigo*. McGraw Hill. 1a ed.
- Morán, G.L., Ponce, L. (2023). *Estudio comparativo entre pseint y raptor basado en la estructura while*. Ecuador. Editorial Alema. 1a ed.