



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Ariel Adara Mercado Martinez

Profesor(a):

Fundamentos de programación

Asignatura:

02

Grupo:

05

No. de práctica(s):

Pérez Jacobo Daniel

Integrante(s):

28

No. de lista o brigada:

2025-2

Semestre:

19(03/2025

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

ÍNDICE

Introducción.....	3
Problemas.....	6
Reflexión.....	10
Conclusión.....	12
Referencias.....	12

INTRODUCCIÓN

El pseudocódigo (Figura 1) es una descripción informal de un algoritmo o un programa, se escribe utilizando un lenguaje natural estructurado que imita la lógica del código que se desea ejecutar, sin embargo este pseudocódigo no sigue las reglas de manera estricta del lenguaje de programación, de manera que solo se asemeja a este. Su propósito principal es mostrar el flujo lógico de un programa y cómo es que este se va a llevar a cabo, para poder facilitar a las personas su entendimiento aun cuando estas no poseen algún tipo de experiencia en programación, de manera que comparten esta características de aclarar las cosas justo como los diagramas de flujo. La principal función del pseudocódigo es que funciona como una especie de “Boceto” para que de esa forma los programadores puedan darle forma y estructura a un código antes de que lo ejecuten, disminuyendo la posibilidad de presentarse errores, el pseudocódigo permite a los programadores concentrarse en el diseño del algoritmo sin preocuparse por los detalles técnicos.

```
procedimiento Visval (linein: array [1 .. n][2] de reales)  
  lineout[1]=linein[1];           El primer punto pasa a la línea de salida  
  npin=n;                        Inicialización n° puntos línea de entrada  
  npout=1;                       Inicialización n° puntos línea de salida  
  mientras (npin>2)  
    hallar punto con area efectiva mínima, amin;  
    eliminar este punto;  
    area[punto]=amin;           Se da un valor de área efectiva a cada punto  
  fin de mientras  
  desde (i=2; i<=npin-1; i++)  
    si (area[i]>=tolerancia)  
      npout++;  
      lineout[npout]=linein[i];  
    fin de si  
  fin de desde  
  npout++;  
  lineout[npout]=linein[npin]; El último punto pasa a la línea de salida  
fin del procedimiento
```

Figura 1: Ejemplo de pseudocódigo

Los bucles (Figura 2) son una herramienta muy importante en la programación, ya que nos permiten ejecutar un bloque de código varias veces seguidas, sin embargo y a pesar de la facilidad de su uso, es importante tener en cuenta que es fácil quedarse atrapado en un bucle infinito si no se escribe la instrucción de manera correcta o si se omite algún paso importante. Por lo tanto, es importante asegurarse de que los bucles tengan una condición de salida clara y que sean controlados adecuadamente.

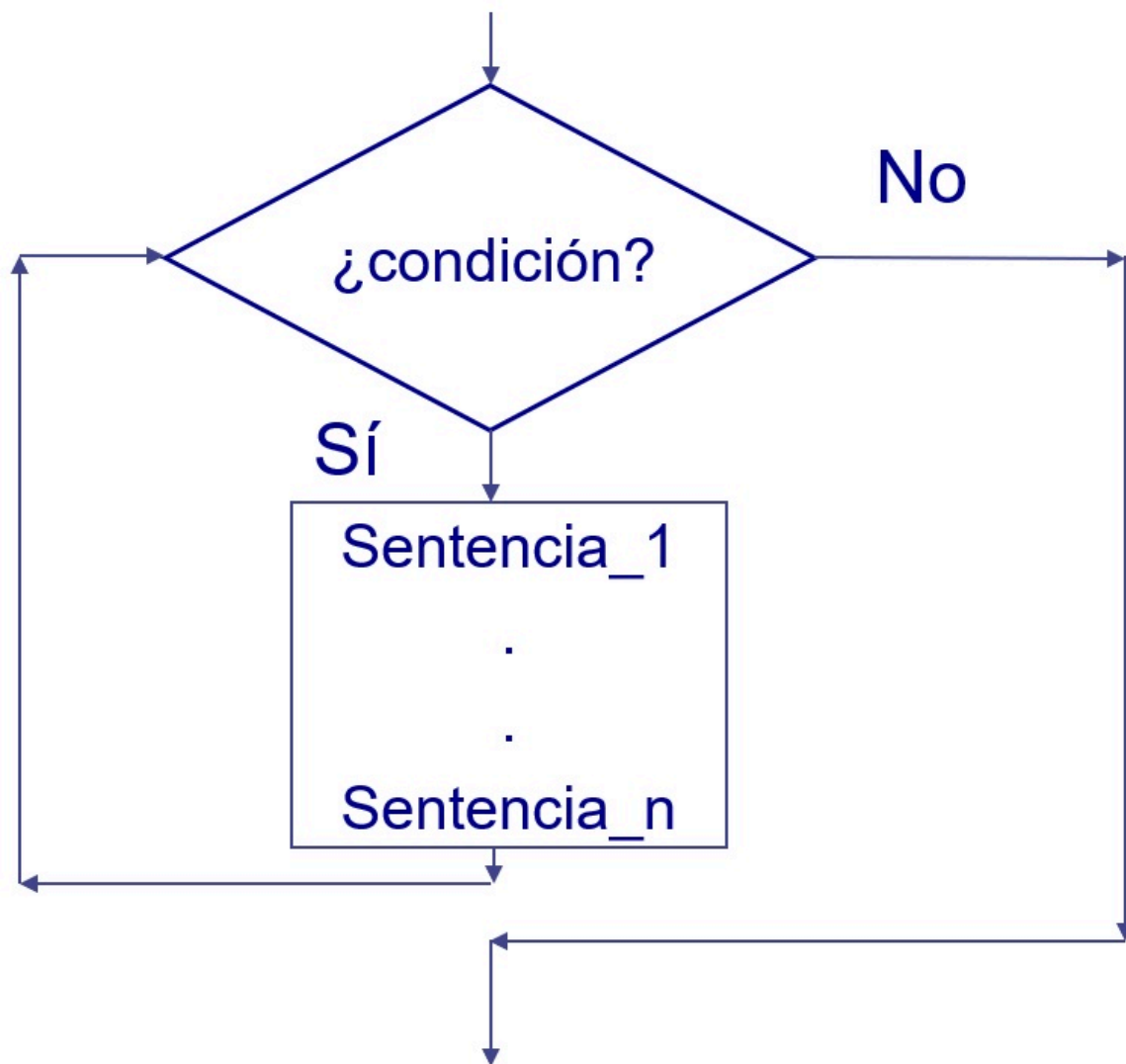


Figura 2: Bucles en pseudocódigo

PSelnt (Figura 3) es una herramienta que nos ayuda a tener una visión general de la programación y así poder comenzar en ella, ejecutando los primeros pseudocódigos, ideal para estudiantes de programación y es la herramienta que se utilizó para la elaboración de esta práctica. Mediante un pseudo lenguaje en español, podemos crear pseudocódigos y con ayuda

de herramientas extras tales como los diagramas de flujo, podemos simplificar todos los bucles y condiciones que se nos sea posible.

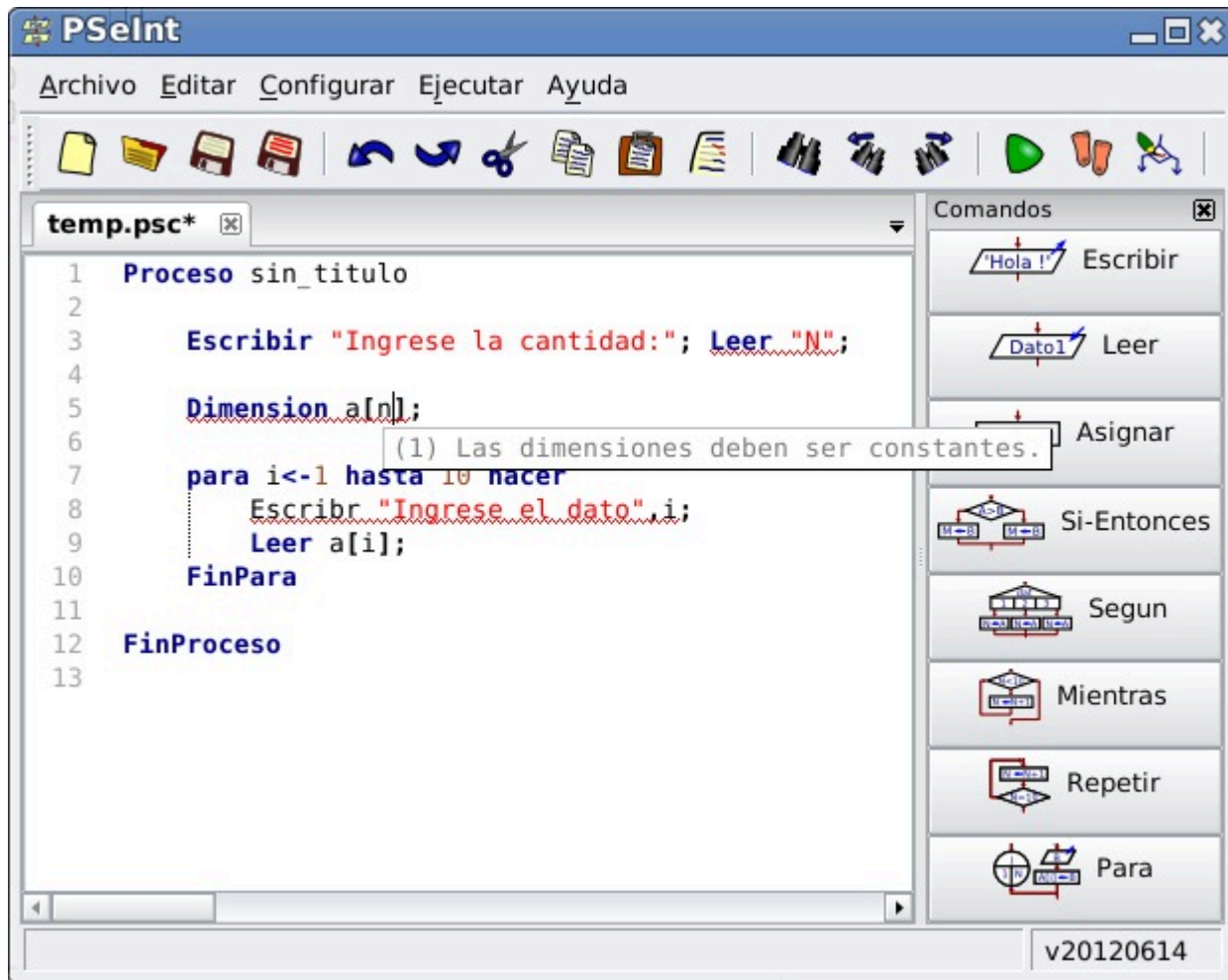


Figura 3: Ejemplo de pseudocódigo en PSeint

PROBLEMAS

1. Crear el pseudocódigo que determine si un número ingresado por el usuario es primo o no

```
1  Algoritmo N_primos
2      Escribir "Ingresar numero entero"
3      Leer n
4      d ← 2
5      primo ← Verdadero
6      Mientras d < n Hacer
7          Si n mod d = 0 Entonces
8              primo ← Falso
9          FinSi
10         d ← d + 1
11     FinMientras
12     Escribir "El numero es primo" , primo
13 FinAlgoritmo
14
```

```
*** Ejecución Iniciada. ***
Ingresa un numero entero
> 17
El numero es primoVERDADERO
*** Ejecución Finalizada. ***
```

2. Crear el pseudocódigo que genere la serie de Fibonacci hasta un número límite ingresado por el usuario

```
1  Algoritmo serie_fibonacci
2      Escribir " Ingresa el numero limite"
3      Leer n
4      a ← 0
5      b ← 1
6      Si a ≤ n entonces
7          Escribir a
8      FinSi
9      Si b ≤ n entonces
10         Escribir b
11     FinSi
12     c ← a + b
13     Mientras c ≤ n hacer
14         Escribir c
15         a ← b
16         b ← c
17         c ← a + b
18     FinMientras
19 FinAlgoritmo
20
```



```

PSelnt - Ejecutando proceso SERIE_FIBONACCI
*** Ejecución Iniciada. ***
Ingresa el limite
> 21
0
1
1
2
3
5
8
13
21
*** Ejecución Finalizada. ***

```

3. Crear el pseudocódigo que calcule el factorial de un número ingresado por el usuario.

```

1  Algoritmo Factorial
2      Definir n, cont como entero
3      cont ← 1
4      Escribir "Ingresa un numero entero"
5      Leer n
6      Si n < 0 entonces
7          Escribir "El numero debe ser positivo"
8      SiNo
9          Para a = 1 hasta n con paso 1 hacer
10             cont ← cont * a
11          FinPara
12          Escribir "El factorial de" , n , "es"
13      FinSi
14  FinAlgoritmo
15

```


*** Ejecución Iniciada. ***

Escriba el número para saber su factorial

> 7

El factorial del número 7 es: 5040

*** Ejecución Finalizada. ***

REFLEXIONA

Sobre la estructura y logica del diagrama de flujo: ´

- ¿Qué elementos básicos de un diagrama de flujo utilizaste en los ejercicios (por ejemplo, inicio/fin, procesos, decisiones, entradas/salidas)?

R. Utilice las decisiones (Si, sino, finsi) y condiciones, asi como los procesos y los bucles de “mientras”.

- ¿Cómo aseguraste que el flujo del algoritmo fuera claro y fácil de seguir? ´

R. Los algoritmos son claros y precisos en el sentido de que las instrucciones están bien definidas.

- ¿Identificas algún paso redundante o innecesario en tu diagrama? ¿Cómo lo optimizamos?

R. Ningún paso es redundante puesto que de ser el caso el algoritmo no podría ejecutarse.

- Sobre la toma de decisiones y condiciones:

- ¿Cómo manejaste las condiciones en los ejercicios (por ejemplo, en el cálculo del promedio o en la verificación de números primos)? ´

R. Añadir una condición, que en el caso de los números primos era si el residuo de la división de n por d era igual a 0, en caso de que se cumpliera se agrega una acción que se desea que el programa realice, en este caso era que asigna a primo como falso.

- ¿Qué pasará si el usuario ingresa un valor no válido (por ejemplo, un número negativo)?
¿Cómo podrías mejorar tu diagrama para manejar estos casos?

R. Se añade otra condición, por ejemplo

Si $n < 0$ entonces

Escribir “El número debe ser positivo”

- Description Sobre los bucles y repeticiones:

- ¿Que tipo de bucle utilizaste en los ejercicios (por ejemplo, while o for)? ¿Por qué elegiste ese tipo? ´
 - ¿Cómo garantizamos que el bucle terminará correctamente sin caer en un ciclo infinito? ´
 - ¿Podrás resolver el mismo problema sin usar un bucle? ¿Cómo lo harías?

R. Se utilizó el bucle “Mientras” de forma que se ejecutará el ciclo siempre y cuando se cumpliera con certa condición, de no ser el caso se omite, para evitar que cayera en un ciclo infinito hay que ser cuidadosos en donde colocamos la siguiente instrucción, de estar dentro del “mientras” lo más seguro es que caiga en un ciclo infinito.´

- Sobre la aplicación práctica: ´

- ¿En que situaciones de la vida real podrías aplicar los conceptos que practicaste con estos ejercicios?

R. Hoy en día la programación es imprescindible en el ámbito laboral debido a que la automatización se encarga de gran parte de las cosas, sin embargo es importante saber instruir a las máquinas que es lo que deben hacer, si no lo harán mal o simplemente no lo harán.´

CONCLUSIONES

Se elaboraron algoritmos de forma exitosa logrando su ejecución y entregando resultados positivos, en cuanto a los objetivos de la práctica, todo fluyó como debía, se utilizaron los bucles aprendidos en clase tales como “mientras” y “para”, así como las condiciones y decisiones que se analizaron en la práctica pasada “Diagramas de flujo”, además de que se nos introdujo a Pseint, la cual es una herramienta muy útil para empezar a trabajar en procesos básicos de programación.

Asimismo pude reforzar la forma en la que se interpretan los problemas, lo cual ha sido algo bastante complicado para mí, puesto que a pesar de que ya me están dando el problema aun se me hace un poco difícil poder expresar la solución en forma de algoritmo y por supuesto de pseudocódigo.

REFERENCIAS

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

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

Robledano, A. (2024b, septiembre 23). Qué es pseudocódigo y por qué es esencial en programación. *OpenWebinars.net*. <https://openwebinars.net/blog/que-es-pseudocodigo/>

¿Qué es PSeInt? – Fernando Monroy. (s. f.).

<https://fernandomonroytenorio.com/tema/que-es-pseint/>

Maldonado, D. (2023, 7 abril). *¿Qué es un Pseudocódigo?* Daniel Maldonado.

<https://danielmaldonado.com.ar/diccionario-de-hacking/que-es-un-pseudocodigo>

Bucle while — Fundamentos de Programación en C++. (s. f.).

https://www2.eii.uva.es/fund_inf/cpp/temas/6_control_flujo_iterativo/while.html

colaboradores de Wikipedia. (2025, 3 marzo). *PSeInt*. Wikipedia, la Enciclopedia Libre.

<https://es.wikipedia.org/wiki/PSeInt>