



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I Ernesto Alcántara Concepción

Asignatura: Fundamentos de programación 1122

Grupo: 17

No de Práctica(s): 04

Integrante(s):

Asencio Morales Miguel Ángel
Guadarrama Herrera Ken Bryan
Sandoval Vásquez Manuel Elihú

Galván Romero Marco Polo
Mendoza Hernández Carlos Emiliano

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: Equipo 1

Semestre: 1er. Semestre

Fecha de entrega: 13 de octubre del 2021

Observaciones:

CALIFICACIÓN: _____

Práctica 04: Diagramas de flujo

Introducción

Un diagrama de flujo es la representación gráfica de un proceso, es decir, muestra gráficamente el flujo de acciones a seguir para cumplir una tarea específica.

Dentro de las ciencias de la computación, un diagrama de flujo es la representación gráfica de un algoritmo. La correcta construcción de estos diagramas es fundamental para la etapa de codificación, ya que, a partir del diagrama de flujo es posible codificar un programa en algún lenguaje de programación.

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. A continuación, se enlistan las características que conforman este lenguaje gráfico.

1. Todo diagrama de flujo debe tener un inicio y un fin.
2. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales u horizontales, exclusivamente.
3. Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas a un símbolo.
4. El diagrama debe ser construido de arriba hacia abajo y de izquierda a derecha.
5. La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación en el que se va a codificar la solución.
6. Se recomienda poner comentarios que expresen o ayuden a entender un bloque de símbolos.
7. Si la extensión de un diagrama de flujo ocupa más de una hoja, es necesario utilizar y enumerar los símbolos adecuados.
8. A cada símbolo solo le puede llegar una línea de dirección de flujo.
9. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

Objetivo

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

Actividades

- 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.

Para los siguientes siete problemas, analiza y describe el problema, las restricciones, los datos de entrada y de salida, es decir el resultado que se quiere obtener, su respectivo algoritmo y diagrama de flujo.

1. Construye el algoritmo y un diagrama de flujo, que al recibir como dato el radio de un círculo, calcule e imprima tanto su área como la longitud de su circunferencia. Dato: RAD (variable de tipo real que representa el radio del círculo).

PROBLEMA: Calcular e imprimir el área y la longitud de una circunferencia dependiendo del valor de su radio.

RESTRICCIONES: Se debe ingresar un valor numérico de tipo real para el radio.

DATOS DE ENTRADA: Rad (Radio)

DATOS DE SALIDA: Área y longitud de la circunferencia de radio x.

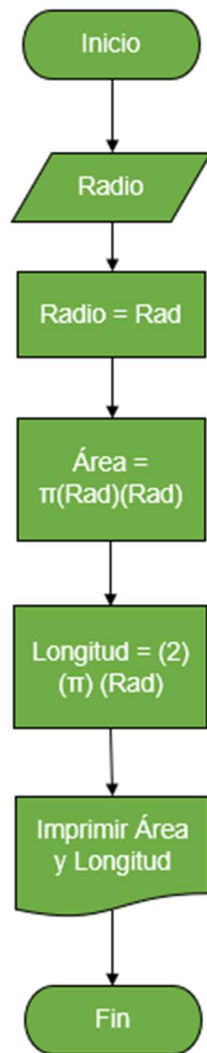
DOMINIO: Todos los números reales

SOLUCION:

Inicio

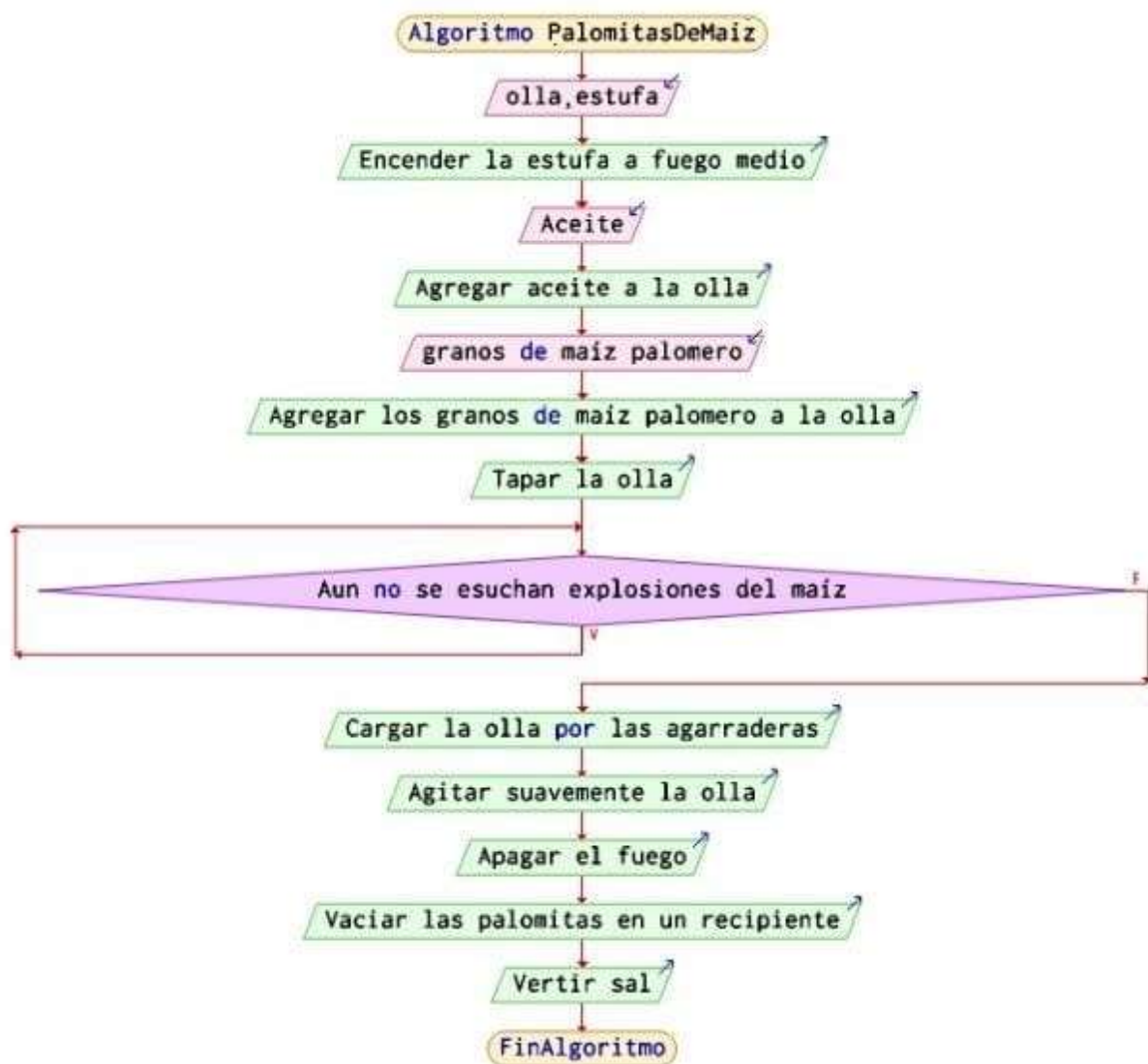
1. Solicitar el radio
2. Guardarlo en la variable Rad
3. Calcular Área = $\pi(\text{Rad})(\text{Rad})$
4. Calcular Longitud = $(2)(\pi)(\text{Rad})$
5. Imprimir el resultado de Área y Longitud

Fin

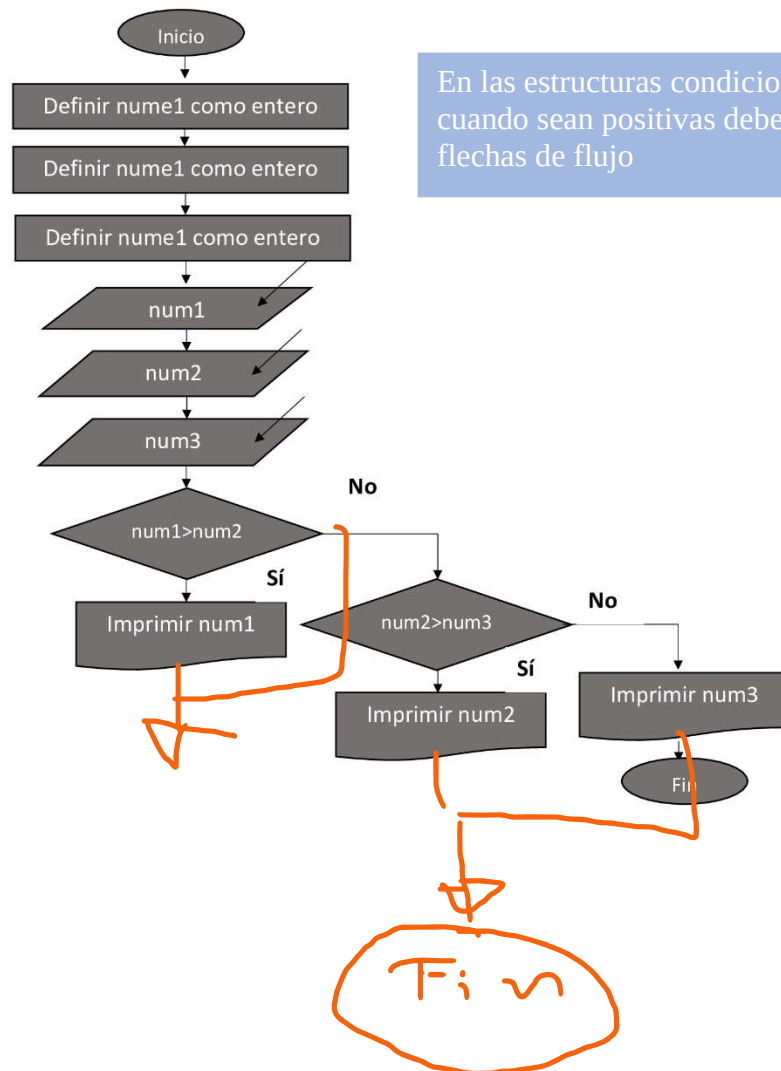


2. Del algoritmo realizado en la práctica número 3, construye su respectivo diagrama de flujo:

a) Hacer palomitas de maíz en una olla puesta al fuego con aceite, sal y maíz.

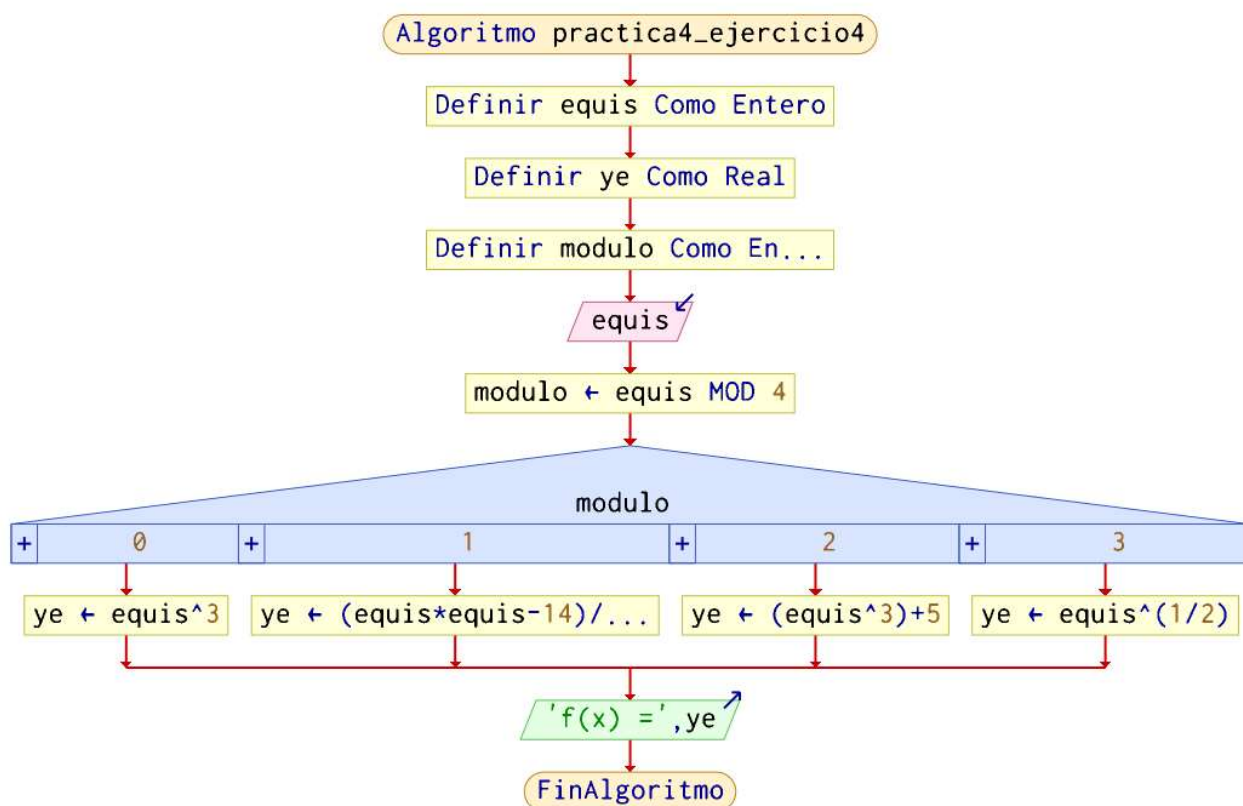


3. Construye un diagrama de flujo que, al recibir como datos tres números reales, identifique cuál es mayor. Considera que los números pueden ser iguales. Datos: N1, N2 y N3 (variables de tipo real que representan los números que se ingresan).



4. Construye un diagrama de flujo que, permita calcular el valor de $f(x)$ según la siguiente expresión

$$f(X) = \begin{cases} Y^3 & \text{Si } (Y \bmod 4) = 0 \\ (Y^2 - 14)/Y^3 & \text{Si } (Y \bmod 4) = 1 \\ Y^3 + 5 & \text{Si } (Y \bmod 4) = 2 \\ \sqrt{Y} & \text{Si } (Y \bmod 4) = 3 \end{cases}$$



5. Construye un diagrama de flujo y algoritmo para sumar los números enteros del 1 al 100 mediante una estructura “para” o estructura “mientras”.

PROBLEMA: Sumar números enteros del 1 al 100.

RESTRICCIONES: La suma debe ser del rango de números enteros comprendidos entre el 1 al 100.

DATOS DE ENTRADA: Números enteros del 1 al 100

DATOS DE SALIDA: Suma de números enteros del 1 al 100

DOMINIO: Números reales (1,100)

SOLUCION:

Inicio.

1. Creamos las variables i,Suma

2. Inicializar la variable $i = 0$

3. Mientras $0 \geq i \leq 100$

- 3.1. Guardar el valor de i en la variable entx, es decir, ent1, ent2,ent3, etc...

- 3.2. Calcular Suma agregando como sumando la variable entx, es decir,

Suma = ent1+ent2+ent3+etc...

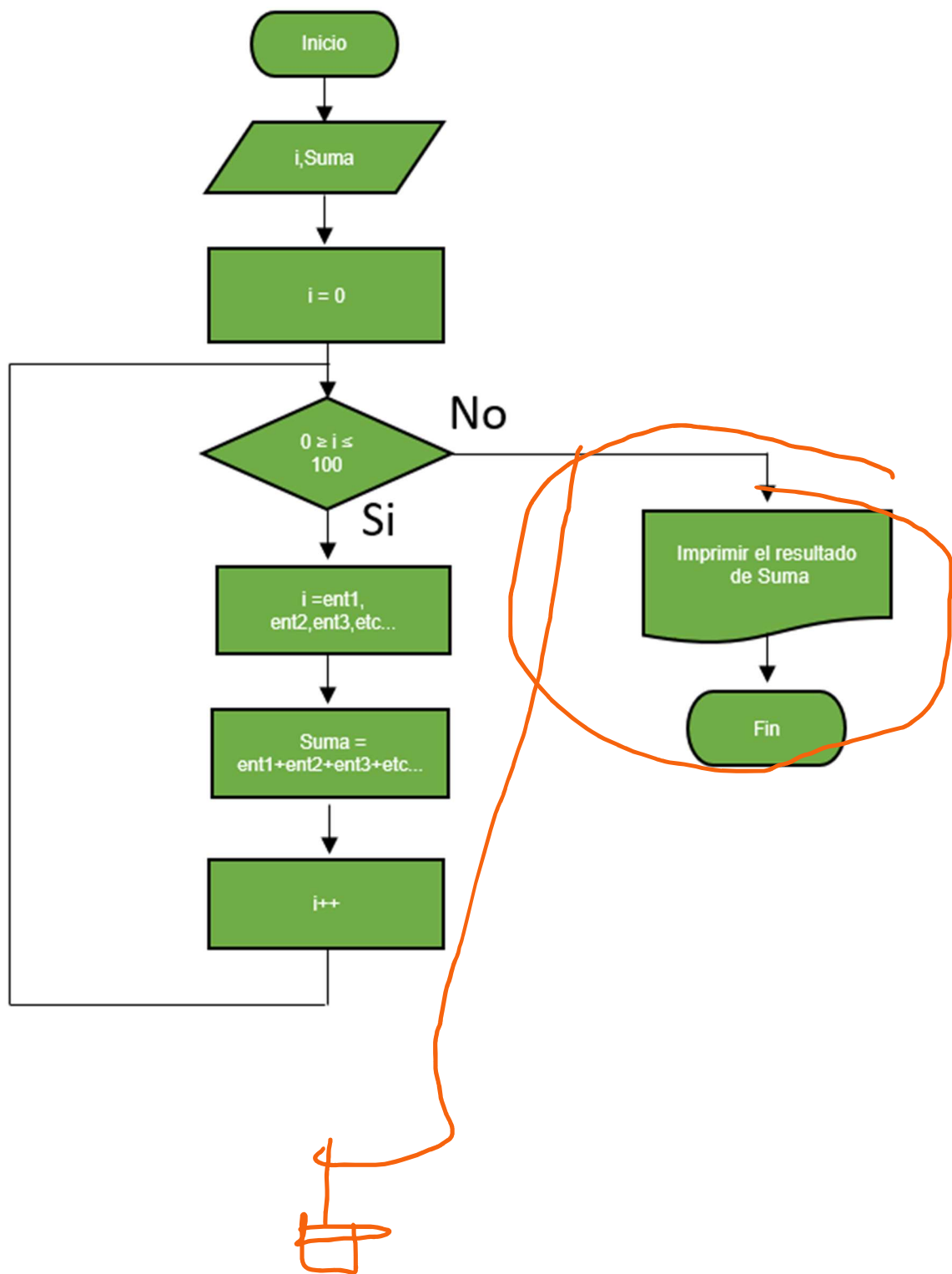
- 3.3. Incrementar i en 1

- 3.4. Regresar al punto 3

4. Una vez que se deje de cumplir la condición $0 \geq i \leq 100$

- 4.1. Imprimir el resultado de Suma

Fin



6. Construye un diagrama de flujo y su algoritmo, de un programa que sume diez números introducidos por teclado (Tip utiliza una estructura Mientras).

PROBLEMA: Sumar diez números introducidos por el teclado.

RESTRICCIONES: El programa debe solicitar la entrada de los números.

DATOS DE ENTRADA: Diez números reales.

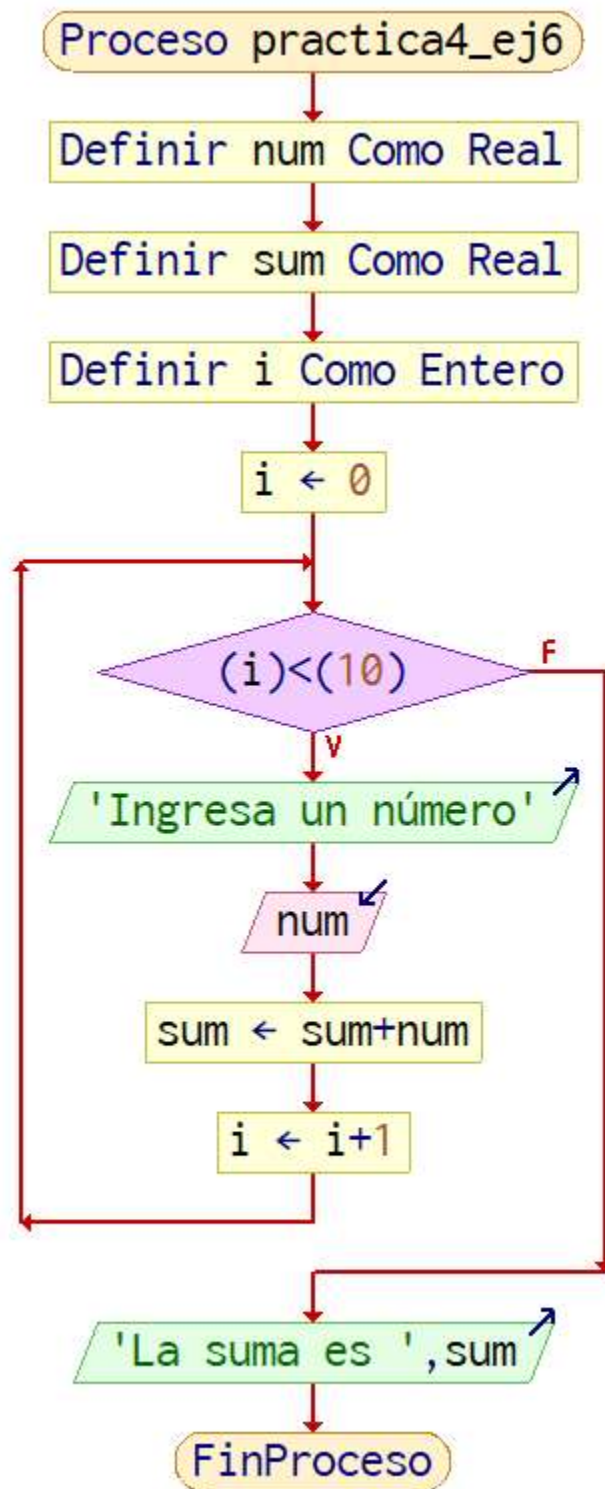
DATOS DE SALIDA: Un número real, resultado de la suma de los diez números enteros.

DOMINIO: Todos los números reales.

SOLUCIÓN:

1. Crear dos variables de tipo real llamadas *num* y *sum*.
2. Crear una variable de tipo entero llamada *i*, esta será el contador del ciclo.
3. Inicializar la variable *i* en 0.
4. Si *i* es menor que 10, deben ejecutarse las siguientes instrucciones.
 - 4.1 Imprimir en pantalla "Ingresa un número".
 - 4.2 Asignar el valor ingresado por el usuario a la variable *num*
 - 4.3 Incrementar a *sum* el valor de *num*.
 - 4.4 Incrementar *i* en 1.
 - 4.5 Regresar al punto 4.
5. Si *i* no es menor que 10, imprimir el valor de *sum*.

DIAGRAMA DE FLUJO:

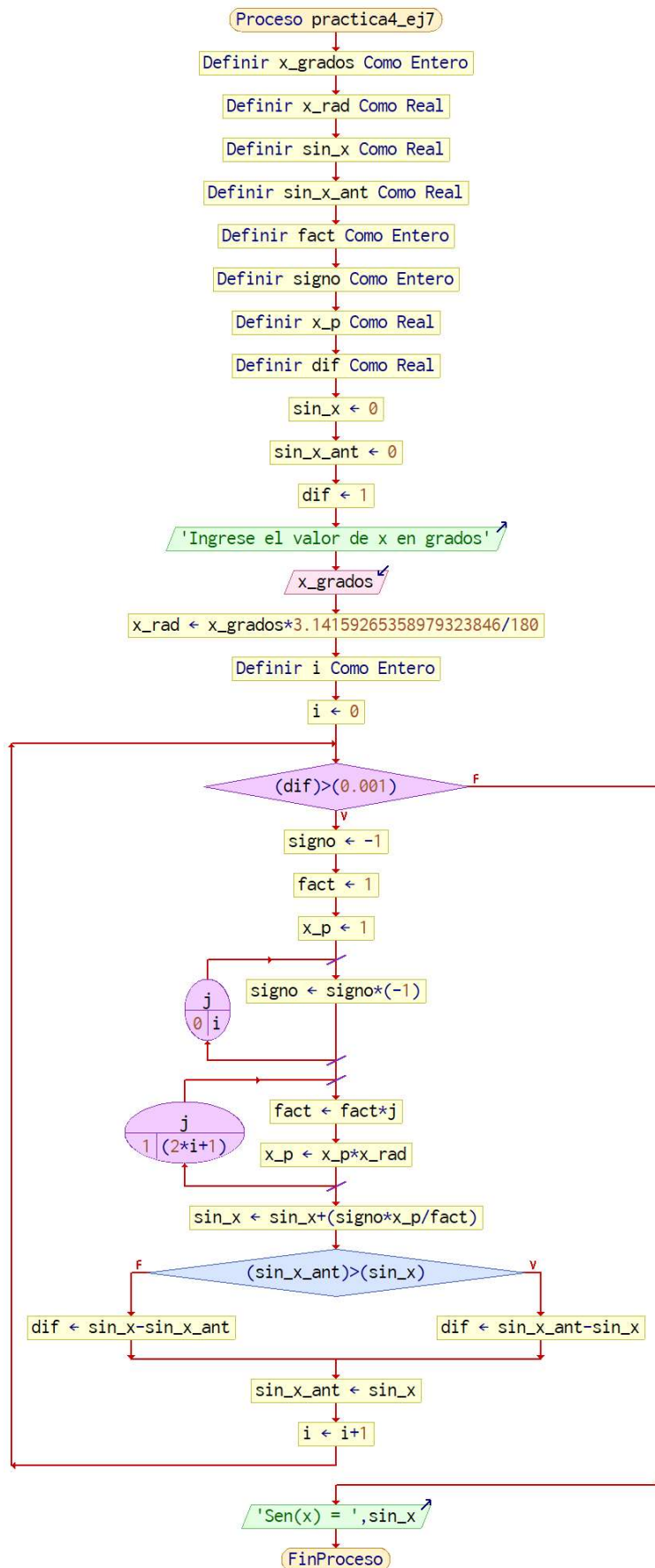


7. Construye un diagrama de flujo que, al recibir como dato una x cualquiera, calcule el $\sin(x)$ utilizando la siguiente serie.

$$\sin(x) = \frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

La diferencia entre la serie y un nuevo término debe ser menor o igual a 0.001. Imprima el número de términos requerido para obtener esa precisión. Dato: x (variable de tipo entero que representa el número que se ingresa).

DIAGRAMA DE FLUJO:



Conclusiones

Un diagrama de flujo permite visualizar y seguir el flujo de instrucciones a realizar en un programa computacional. A través de esta herramienta podemos observar de manera práctica los pasos que sigue nuestro algoritmo, una vez se tiene el diagrama de flujo, es más fácil para otras personas o desarrolladores (en caso de un proyecto colaborativo) el entender no solo para qué sirve un programa sino también que pasos sigue y en qué orden, como un tipo de mapa o croquis.

El uso y creación de algoritmos resulta eficiente y una opción válida a la hora de buscar soluciones a problemáticas de cualquier tipo, desde la resolución de operaciones aritméticas simples hasta el cálculo de funciones de cualquier tipo. Concluimos que es importante seguir las reglas de implementación de los diagramas de flujo, para que cualquier persona pueda verlo y entenderlo. Un buen diagrama de flujo soluciona el problema planteado, es claro y entendible para cualquier persona.

Aunque en la práctica están orientados a un contexto de programación, en nuestro día a día estamos rodeados de algoritmos que facilitan el cumplimiento de tareas y objetivos, como ejemplo están las recetas, instructivos, rutinas y tutoriales.

Una vez realizada la práctica 3, esperamos seguir aprendiendo acerca de algoritmos, diagramas de flujo y posteriormente el pseudocódigo y ejecución en el lenguaje de programación C.

El objetivo de la práctica era elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprenden un proceso. Con el diseño e implementación de los problemas planteados concluimos que en este punto se cumplieron los objetivos y se concluye la realización de esta práctica.