



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor(a):* Ing. Karina García Morales

*Asignatura:* Fundamentos de Programación (L)

*Grupo:* 22

*No. de práctica(s):* Práctica 4: Diagramas de flujo

*Integrante(s):* González Márquez José Luis

*No. de lista o brigada:* 20

*Semestre:* 2026-1

*Fecha de entrega:* Martes 23 de Septiembre de 2025

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo

El alumno elaborará diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

## Actividades:

- Elaborar un diagrama de flujo que represente la solución algorítmica de un problema, en el cual requiere 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 requiere el uso de la estructura de control iterativa.

## Introducción

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

## Formas de los diagramas de flujo

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de gráfica. A continuación, se muestran los elementos que conforman este lenguaje gráfico.

1. Todo diagrama de flujo debe tener un inicio y un fin (Figura 1).



Figura 1. Representación gráfica de Inicio y Fin

2. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales u horizontales, exclusivamente (Figura 2).



Figura 2. Representación gráfica de dirección de flujo

3. Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas a un símbolo (Figura 3).

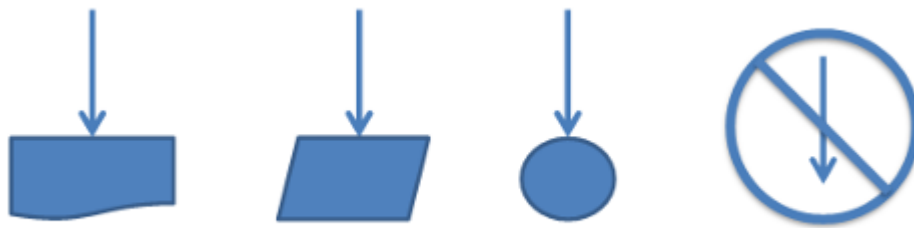


Figura 3. Conexión correcta de las líneas de dirección de flujo

4. El diagrama debe ser construido de arriba hacia abajo (top-down) y de izquierda a derecha (left to right).

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 página, es necesario utilizar y numerar los símbolos adecuados.

8. A cada símbolo solo le puede llegar una línea de dirección de flujo (Figura 4).

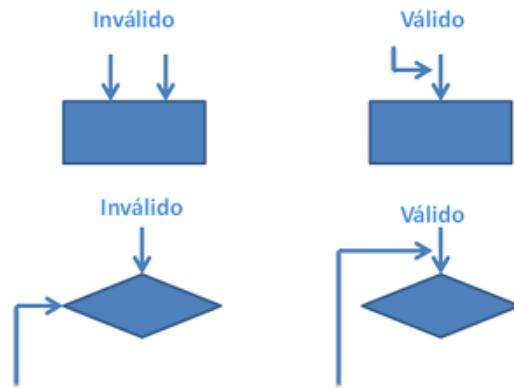


Figura 4. Uso de líneas de dirección

9. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. Por tanto, es fundamental conocer los elementos que conforman este lenguaje gráfico.



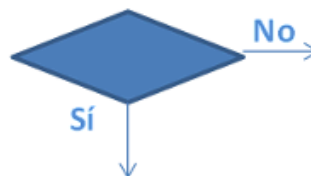
Representa el inicio o el fin del diagrama de flujo.

Datos de entrada. Expresa lectura de datos.



Proceso. En su interior se expresan asignaciones u operaciones.

Decisión. Valida una condición y toma uno u otro camino.

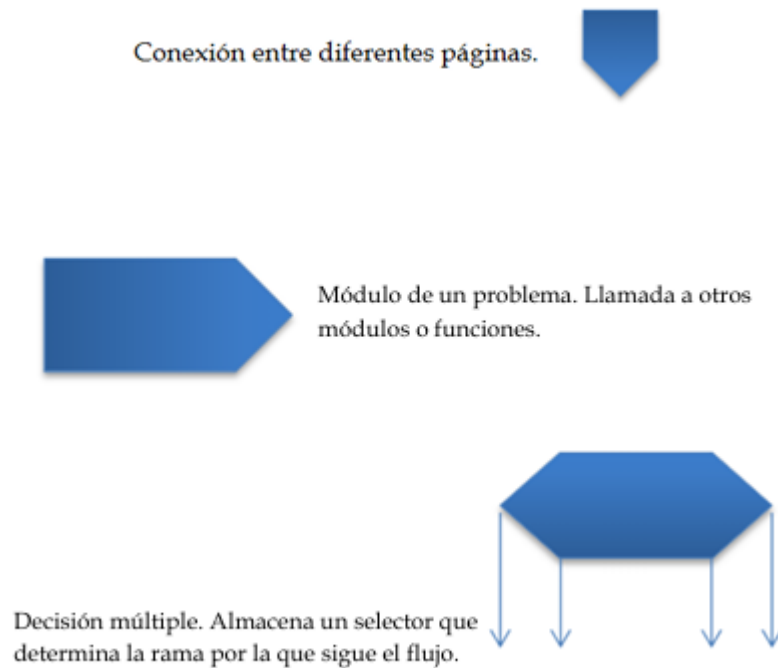


Escritura. Escribe el o los resultado(s).

Dirección de flujo del diagrama.



Conexión dentro de la misma página.



La Figura 5 muestra un ejemplo clásico de diagrama de flujo computacional. Se puede observar el uso de los diferentes símbolos para indicar las acciones y el flujo a seguir para la solución de problemas.



Figura 5. Ejemplo clásico de un diagrama de flujo computacional

### Estructuras de control de flujo

Las estructuras de control de flujo permiten tanto la ejecución condicional como la repetición de un conjunto de instrucciones.

Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

### Estructura de control secuencial

Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas (Figura 6).

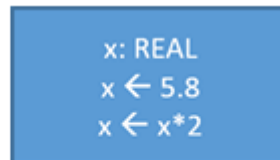


Figura 6. Ejemplos de estructuras de control secuencial.

### Estructuras de control condicionales (o selectivas)

Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se realiza una acción o se realiza la otra).

La Figura 7 muestra la estructura de control condicional más simple. Se denomina estructura condicional SI (IF por su término en inglés).

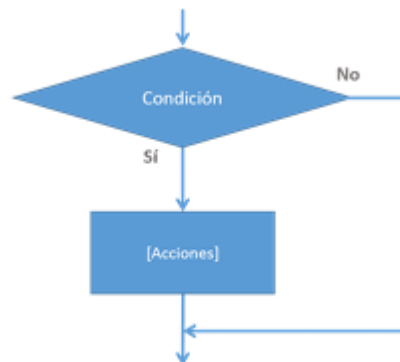


Figura 7. Estructura de control condicional simple.

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se realizan las instrucciones del bloque [Acciones]. Si no se cumple la condición, se continúa con el flujo del diagrama descartando las [Acciones].

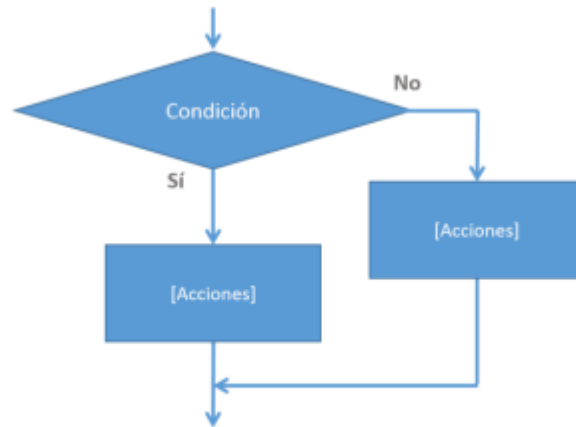


Figura 9. Estructura de control condicional SI-DE LO CONTRARIO

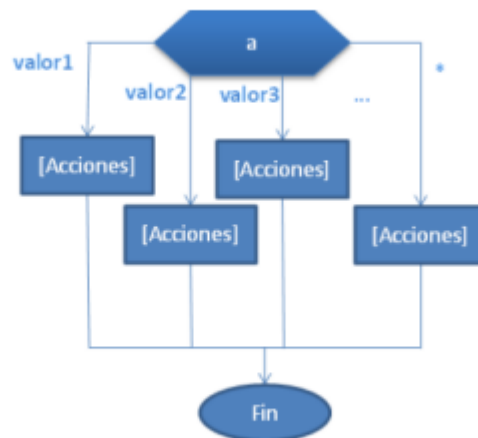


Figura 11. Estructura de control condicional SELECCIONAR-CASO

### Estructuras de control iterativas o repetitivas

Las estructuras de control de flujo iterativas o repetitivas (también llamadas cíclicas) permiten realizar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas: MIENTRAS y HACER- MIENTRAS. La estructura MIENTRAS primero valida la condición y si ésta es verdadera (Sí) procede a realizar el bloque de instrucciones de la estructura [Acciones] y regresa a validar la condición, esto lo realiza mientras la condición sea verdadera (Sí); cuando la condición sea Falsa (No se cumpla) se rompe el ciclo y el flujo continúa con las estructuras que le sigan.

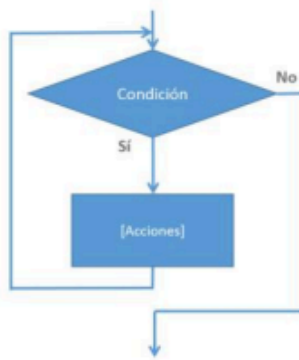


Figura 13. Estructura de control iterativa MIENTRAS

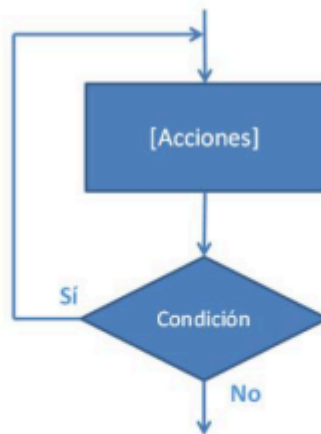
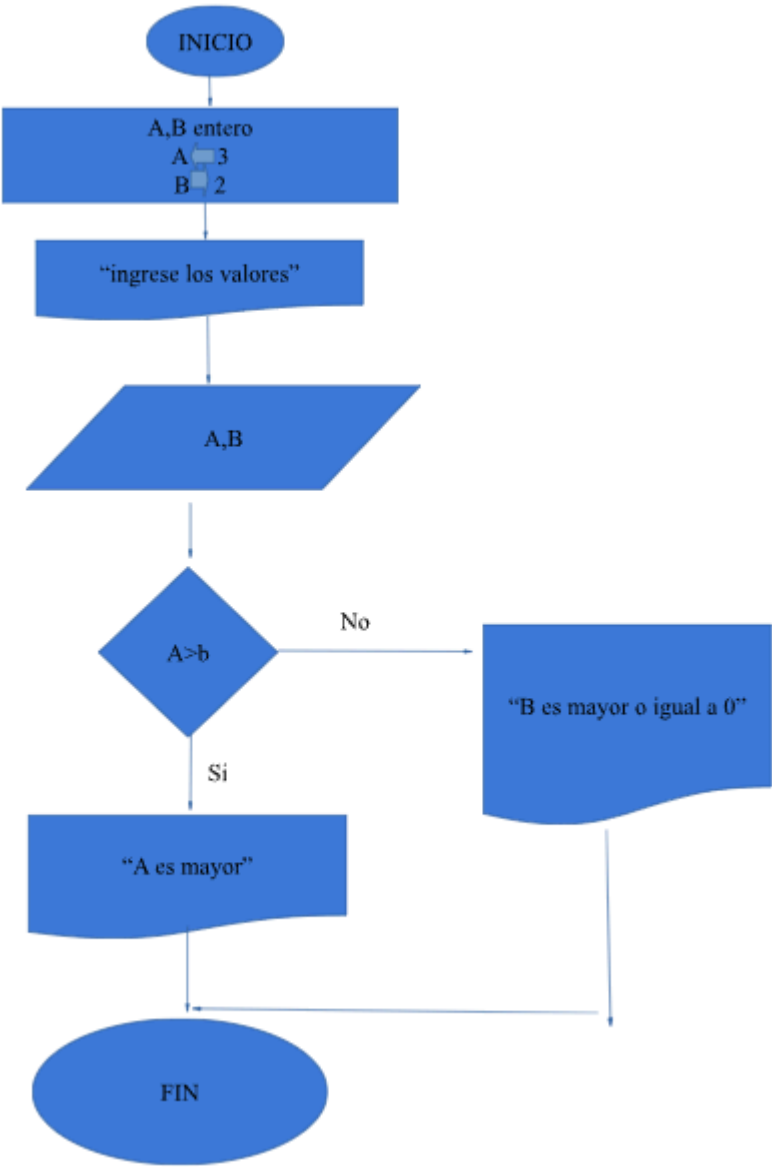


Figura 15. Estructura de control iterativa HACER-MIENTRAS



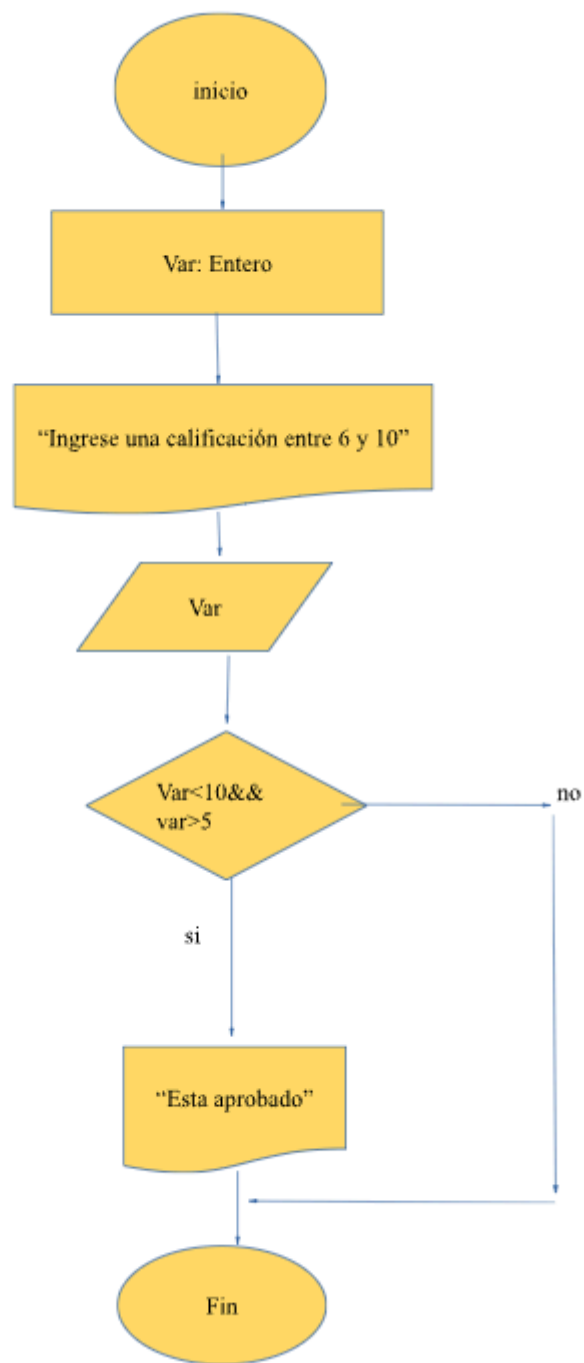
Ejercicios de clase:

1. Modificar la Figura 10 para que le solicite al usuario los valores de a y b



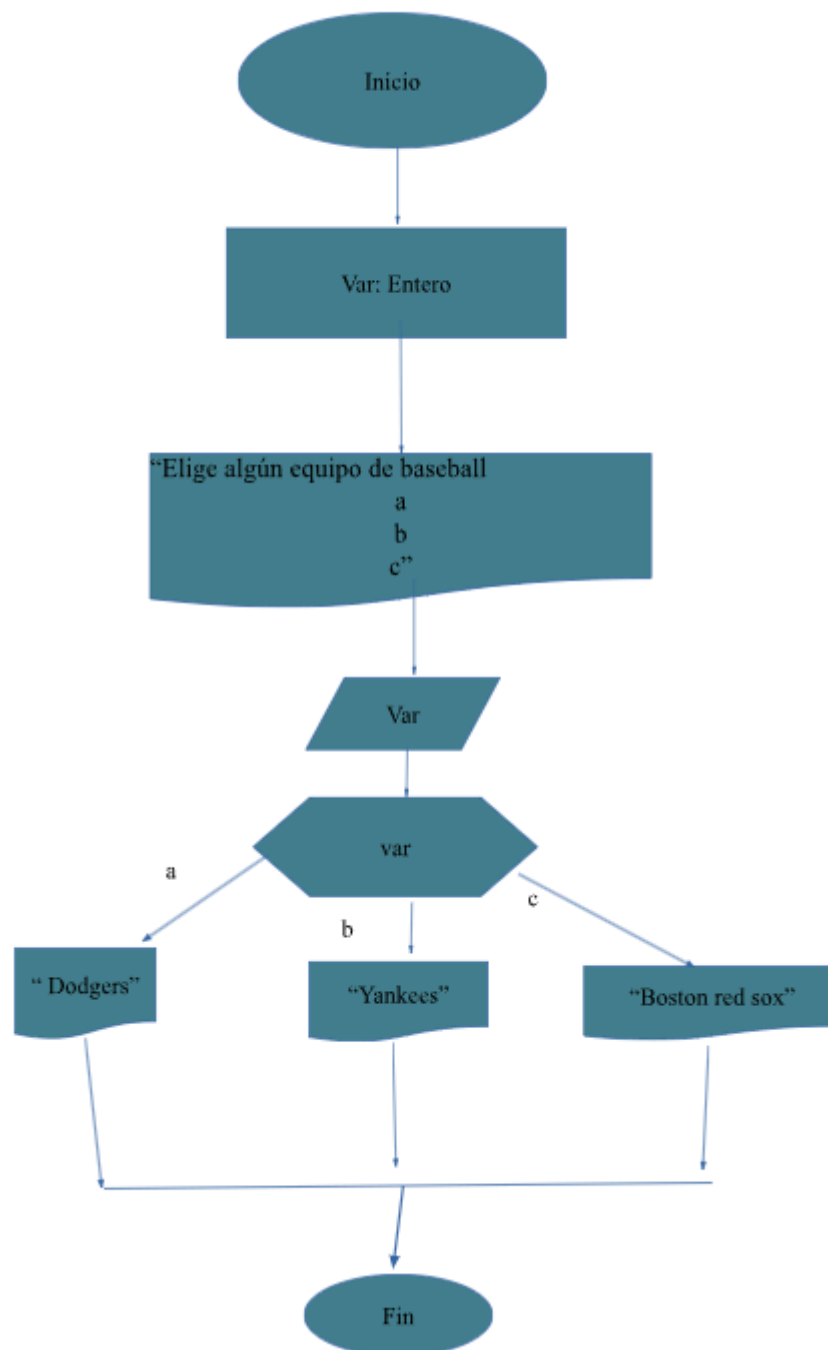
Numero de prueba	Var A	Var B	Resultado
1	3	7	B es mayor o igual a A
2	8	6	A es mayor
3	5	3	A es mayor

2. Solicita una calificación al alumno entre 5 y 10. En caso contrario ir a Fin, es decir, termina el programa.



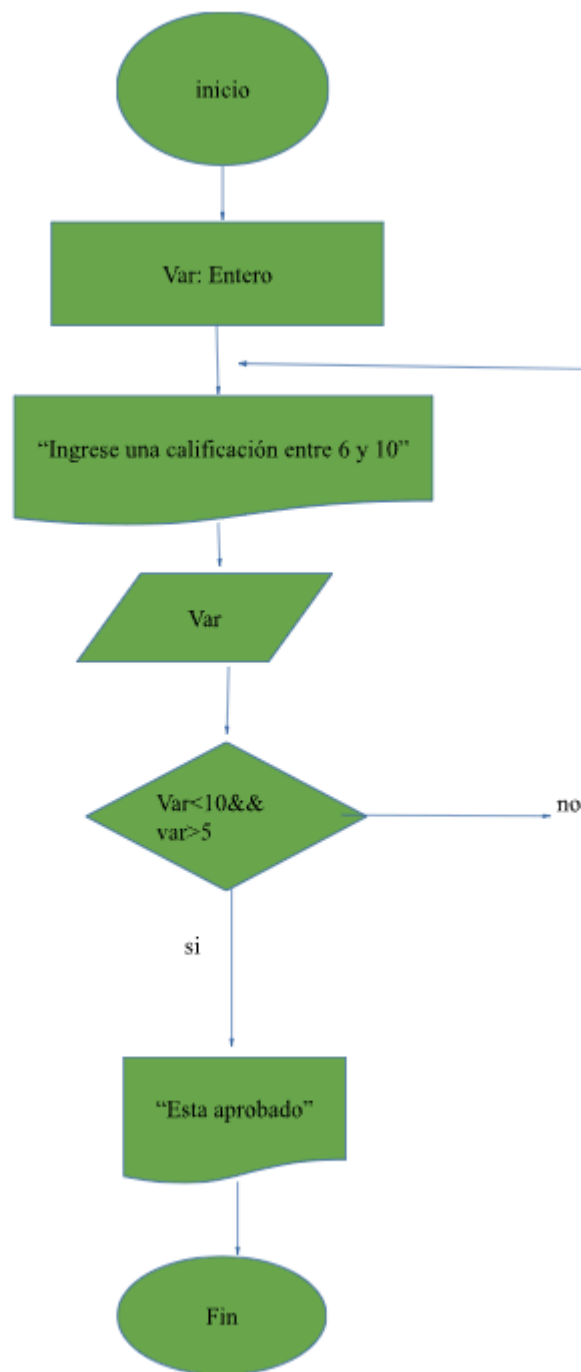
Numero de prueba	Calificaciòn	Es mayor a 5	Resultado
1	3	No	---
2	8	SI	Aprobado
3	9	SI	Aprobado

3. Desarrolla un menú libre de 3 opciones.



Numero de prueba	VAR	Resultado
1	A	Dodgers
2	B	Yankees
3	C	Boston Red sox

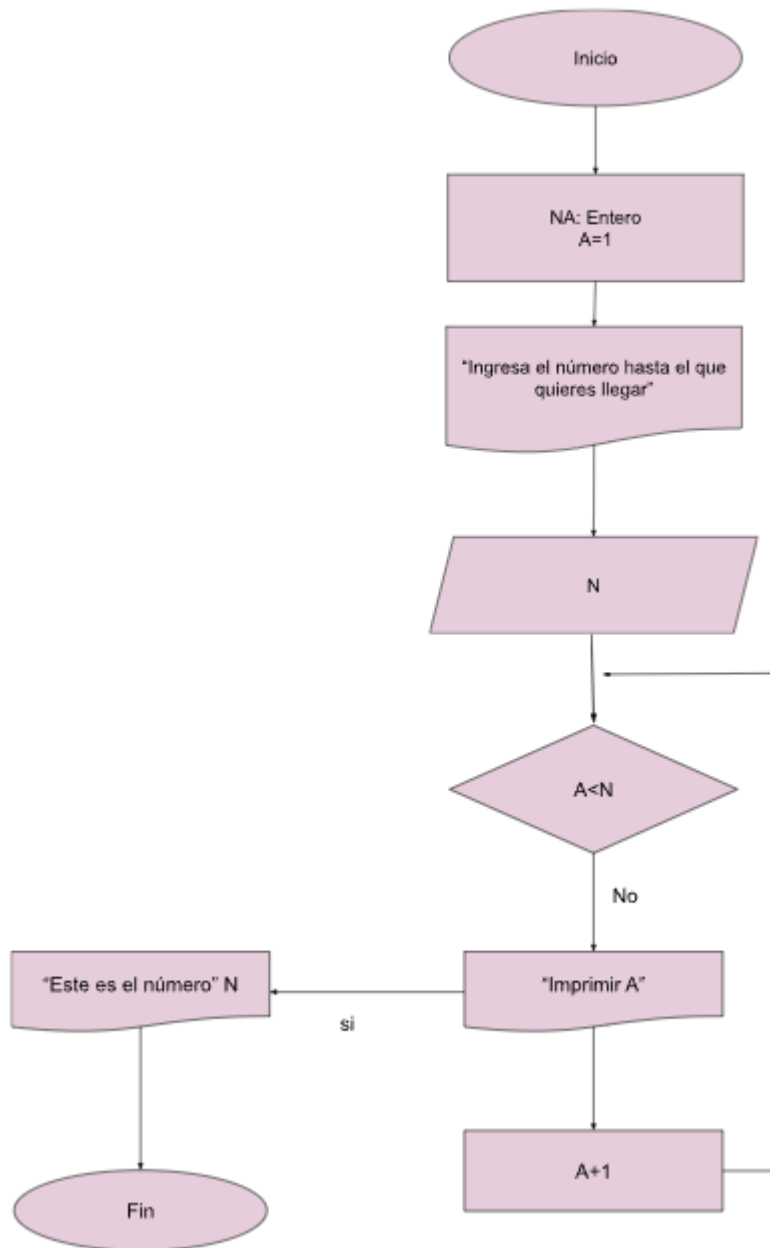
4. Solicitar una calificación entre 5 y 10. En caso de que no cumpla, volver a solicitar la calificación.



Numero de prueba	Calificación	Es mayor o igual a 6	Resultado
1	9	si	Aprobado
2	5	No	Reprobado
3	2	No	Reprobado

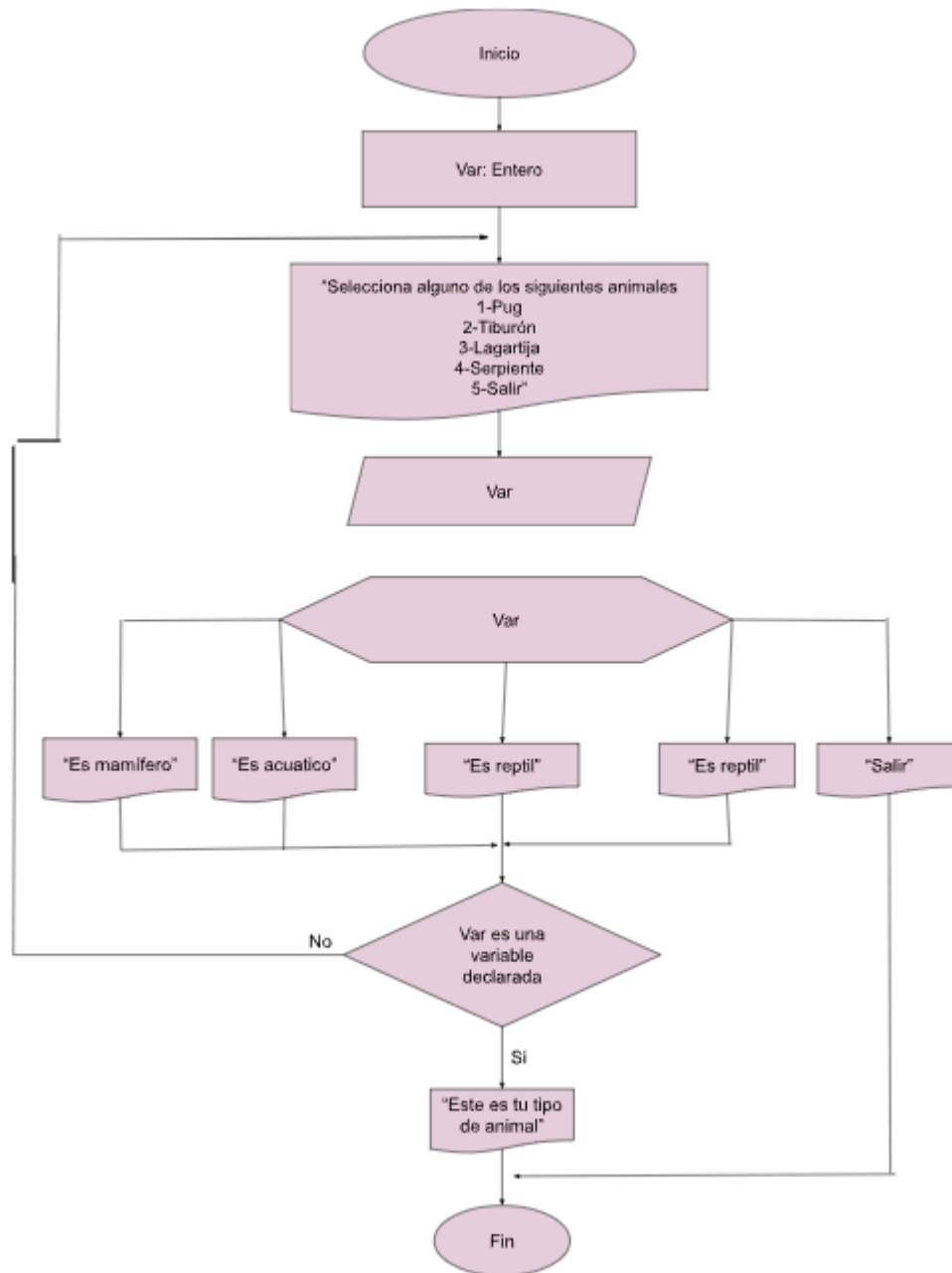
### Tarea 1.-

Solicita un número entero positivo y cuenta desde 1 hasta el número dado (Estructura hacer mientras y mientras).



Numero de prueba	Var N	Resultado
1	4	1,2,3,4
2	5	1,2,3,4,5
3	6	1,2,3,4,5,6

2.- Genera un menú de los 4 programas (3 de clase y el de tarea), agregar el ciclo hacer mientras, en donde valide, si el usuario no elige las opciones 1,2, 3 o 4 imprimir un letrero de "error"(es el default) y volver a mostrarle el menú, hasta que seleccione la opción 5 como salir



Numero de prueba	Opción	Resultado
1	4	Es reptil
2	5	Salir
3	2	Es acuatico

## **Conclusión:**

La verdad está práctica me gusto mucho, ya que desde un inicio aprendí mucho sobre bastantes cosas como lo fueron los algoritmos, ciclo de vida del software, solución de problemas, variables y pruebas de escritorio. Todo esto para poder llevar a cabo un proceso y que cumpla una función.

El primer ejercicio sinceramente me sorprendió porque era bastante sencillo, pero si te ponías a pensar era una forma muy fácil y completa para entender los algoritmos.

El segundo ya subía la dificultad, ya que te pedía hacer pasos más específicos para poder llegar al resultado final, que en este caso fue una estrella de seis puntos.

Para el último ejercicio ya no era algo tan práctico como lo había hecho anteriormente, aquí era algo más informático y con números.

Sin duda alguna me gusto mucho lo aprendido, también en cómo te introducen de la forma más sencilla posible para que puedas entender lo que viene más adelante, mi favorito fue hacer estrellas porque demuestra que de un simple dibujo se puede llegar al objetivo que en este caso fue entender y practicar lo aprendido.

## **Fuentes:**

Facultad de Ingeniería. (2025). Manual de prácticas del laboratorio de Fundamentos de Programación. Laboratorio de computación. Salas A y B. Universidad Nacional Autónoma de México. pp. 36-51. Recuperado el 13 de septiembre de 2025 de <http://lcp02.fi-b.unam.mx/>