



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): Ing. Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 22

No de Práctica(s): Práctica 4

Integrante(s): Edilberto Vicente Martínez

No. de lista o 49
brigada:

Semestre: 2026-1

Fecha de entrega: 23 de septiembre de 2025.

Observaciones:

CALIFICACIÓN: _____

Práctica 04: Diagramas de flujo

Objetivo:

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

Desarrollo:

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 manera 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).

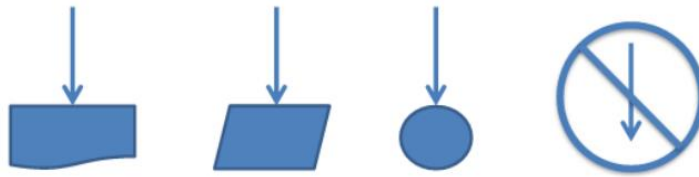


Figura 3. Conexión correcta de las líneas de dirección de flujo

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).

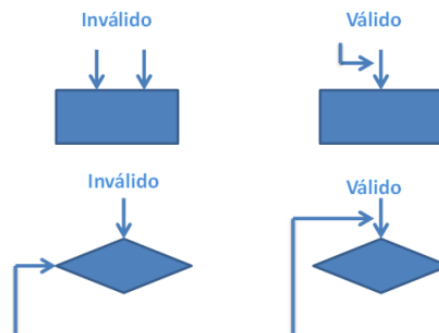
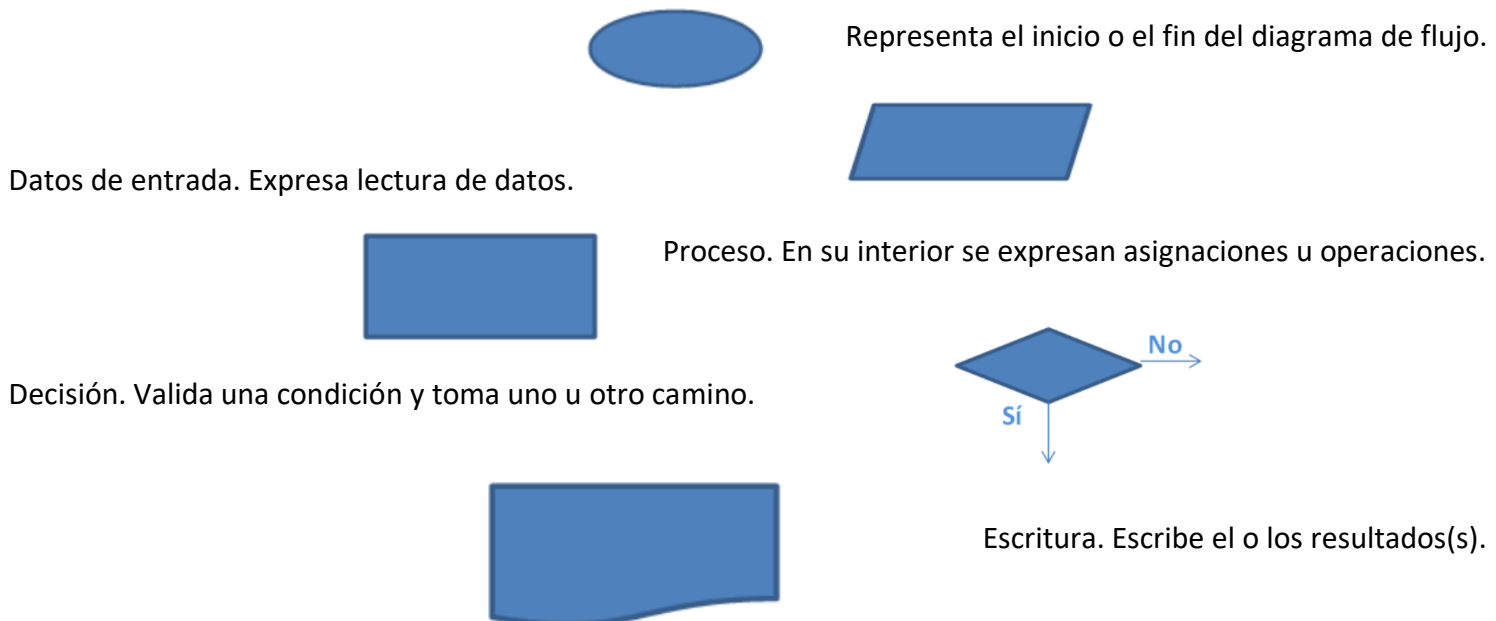
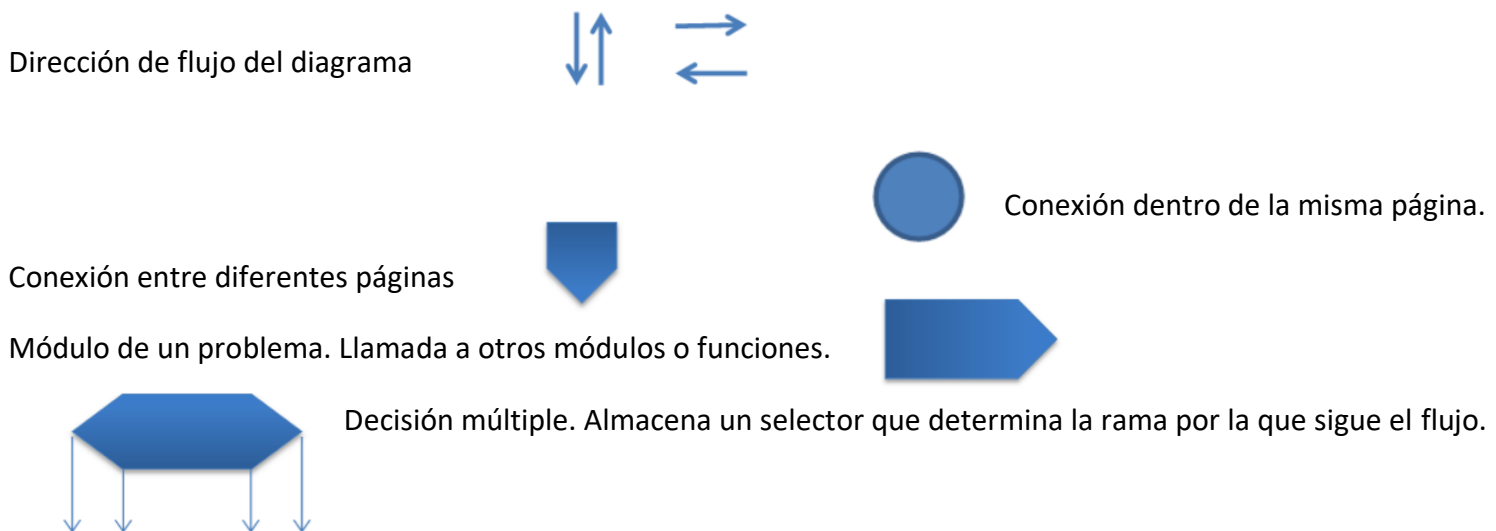


Figura 4. Uso de líneas de dirección

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.





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).

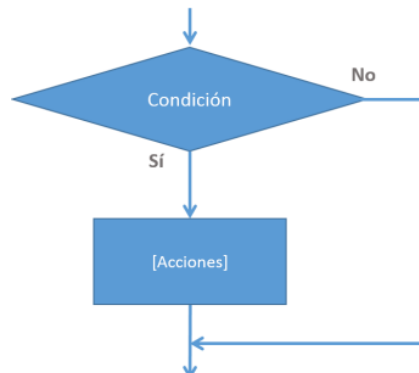


Figura 7. Estructura de control condicional simple

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].

Además de la estructura de control condicional simple SI, existe la estructura condicional completa nombrada SI-DE LO CONTRARIO (IF-ELSE, por sus términos en inglés). Véase la Figura 9,

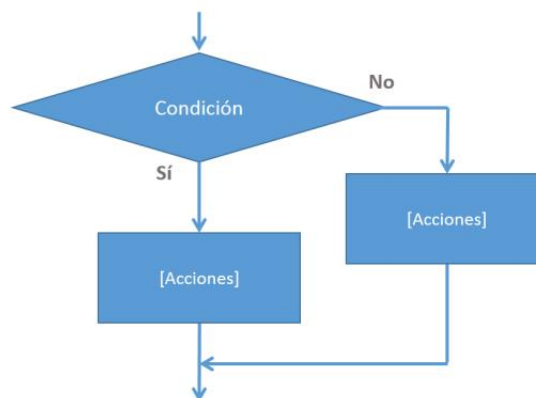


Figura 9. Estructura de control condicional SI-DE LO CONTRARIO

En esta estructura se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se realizan las instrucciones del bloque Sí. Si no se cumple la condición se realizan las instrucciones del bloque No. Al final de la estructura condicional, el flujo continúa con las estructuras que le sigan.

Otra estructura de control condicional es SELECCIONAR-CASO, la cual valida el valor de la variable que está en el hexágono y comprueba si es igual al valor que está definido en cada caso (ramas o líneas que emanan del hexágono). Si la variable no tiene el valor de algún caso entonces fluye por la última rama etiquetada con *, la cual es la única a la que no se le especifica un valor (Figura 11).

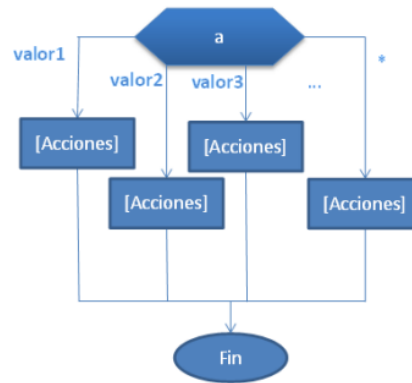


Figura 11. Estructura de control condicional SELECCIONAR-CAS

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. Véase la Figura 13.

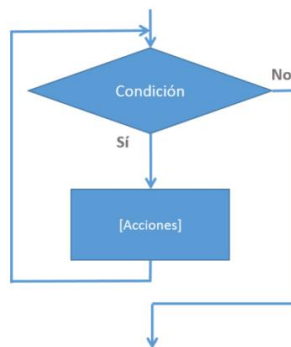


Figura 13. Estructura de control iterativa MIENTRAS.

La otra estructura de control iterativa es HACER-MIENTRAS, la cual primero realiza las instrucciones descritas en la estructura [Acciones] y después valida la expresión lógica; si la expresión lógica es verdadera, realiza nuevamente las [Acciones] y esto se repite mientras la expresión lógica se cumpla. Cuando ésta sea falsa, se rompe el ciclo y el flujo continúa con las estructuras que le sigan. Véase la Figura 15. Esta estructura asegura que, por lo menos, se realiza una vez el bloque de la estructura, ya que primero las realiza y después pregunta por la condición.

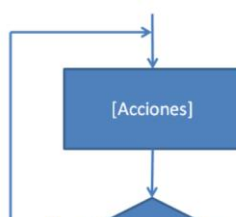


Figura 15. Estructura de control iterativa HACER-MIENTRAS

Ejemplo

La Figura 16 muestra el diagrama de flujo que usa la estructura de control iterativa HACERMIENTRAS el cual escribe el valor de la variable enteroValorInicial y luego realiza el incremento de esta variable para posteriormente evaluar la condicional, esto se repite mientras la condición sea verdadera.

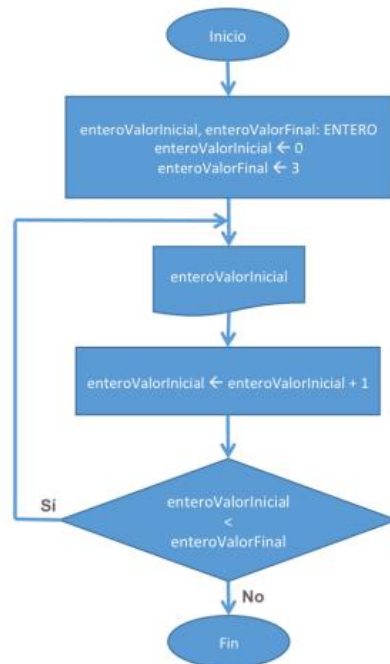
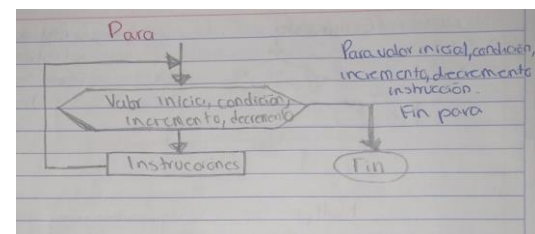
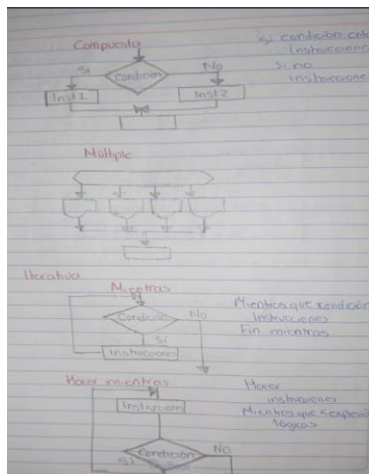
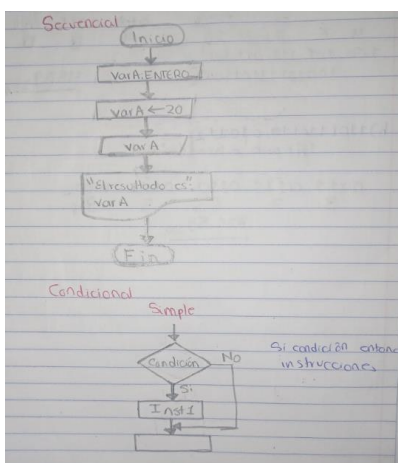


Figura 16. Diagrama de flujo que usa la estructura de control iterativa HACER-MIENTRAS.

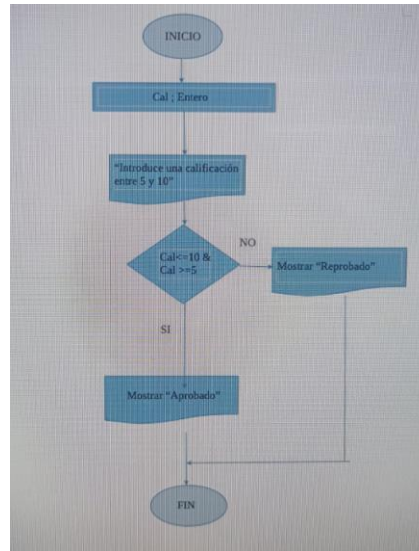
Dibuja las tres estructuras de control de flujo.

Dibujo de las 3 estructuras.

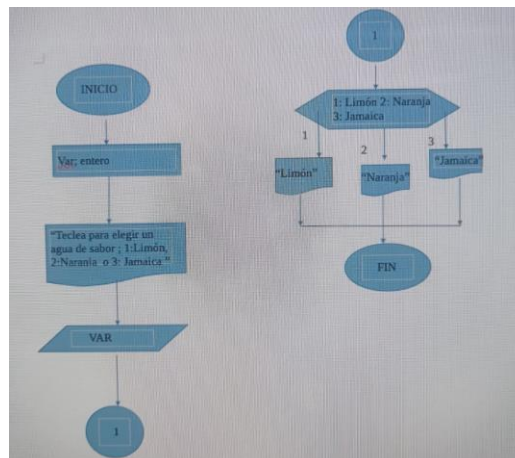


Ejercicios de clase:

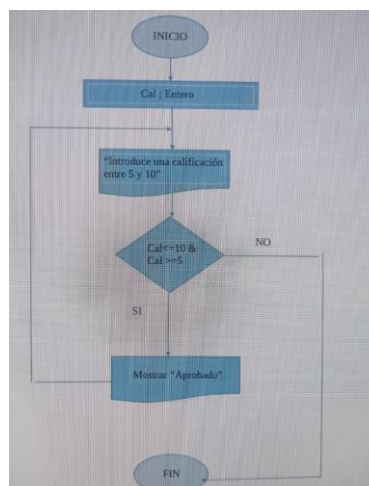
a) Genera un diagrama de flujo que solicite una calificación entre 5 y 10, mostrar en pantalla, en caso contrario, termina el programa (Estructura condicional simple)



b) Genera un menú de tema libre con mínimo 3 opciones (estructura condicional múltiple)

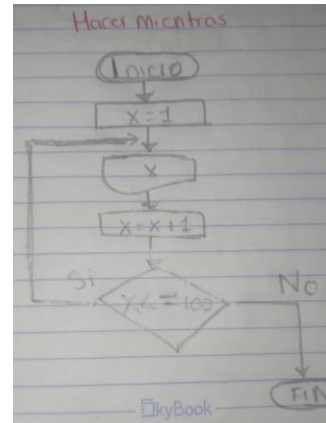
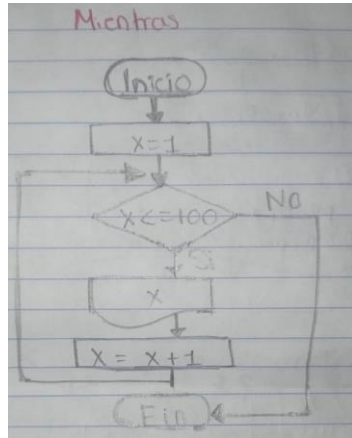


c) Genera un diagrama de flujo que solicite una calificación entre 5 y 10, mostrar en pantalla, en caso contrario, volver a pedir la calificación (Estructura hacer mientras)

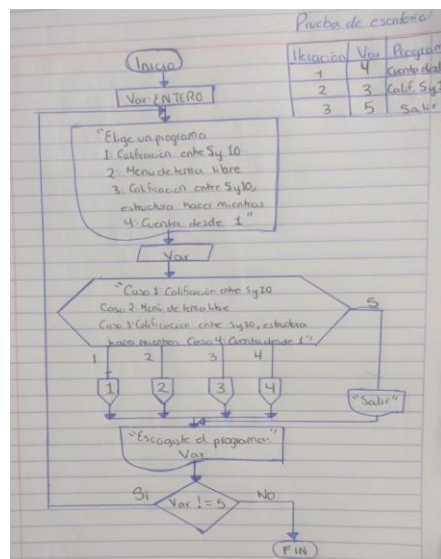


Tarea:

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



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.



Conclusión

En esta práctica aprendimos sobre las distintas maneras de realizar diagramas de flujo dependiendo del algoritmo que queramos representar, además se vio sobre lo importante que es el respetar las reglas para

cada de nuestros diagramas, ya que se debe de dar un buen seguimiento a esto porque queremos que nuestro algoritmo se desarrolle de una buena forma para que pueda ser útil y no tener errores a la hora de nuestras pruebas de escritorio.

En conclusión, el hacer esta práctica y los ejercicios nos sirvió para darnos una idea de cómo proponer soluciones a distintos problemas de nuestra vida cotidiana.

Bibliografía

*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 23 de septiembre de 2025 de <http://lcp02.fi-b.unam.mx/>