
	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	1/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## Guía práctica de estudio 04: Diagramas de flujo

---



Elaborado por	Actualizado por:	Revisado por:
M.C. Edgar E. García Cano Ing. Jorge A. Solano Gálvez	M.C. Cintia Quezada Reyes M.C. Laura Sandoval Montaño	M.C. Laura Sandoval Montaño

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	2/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## Guía práctica de estudio 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.


### 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.

### 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.

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	3/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## 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

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	4/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

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

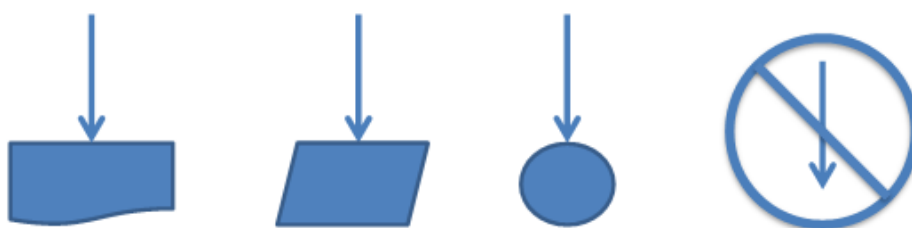


Figura 3. Conexión correcta de las líneas de dirección de flujo

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).

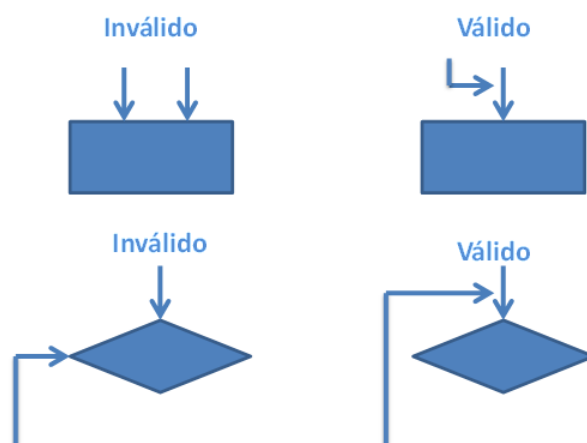



Figura 4. Uso de líneas de dirección

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	5/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

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.



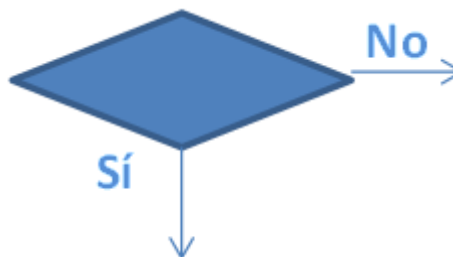
Representa el inicio o el fin del diagrama de flujo.


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.



	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	6/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			



Escritura. Escribe el o los resultado(s).

Dirección de flujo del diagrama.




Conexión dentro de la misma página.

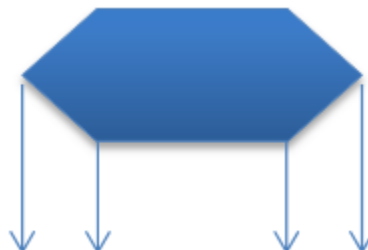
Conexión entre diferentes páginas.



Módulo de un problema. Llamada a otros módulos o funciones.

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	7/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Decisión múltiple. Almacena un selector que determina la rama por la que sigue el flujo.



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.

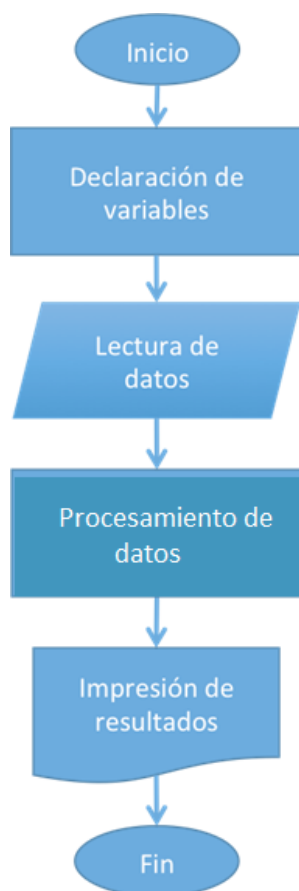



Figura 5. Ejemplo clásico de un diagrama de flujo computacional

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	8/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## 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).

```
x: REAL
x ← 5.8
x ← x*2
```


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).



	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	9/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

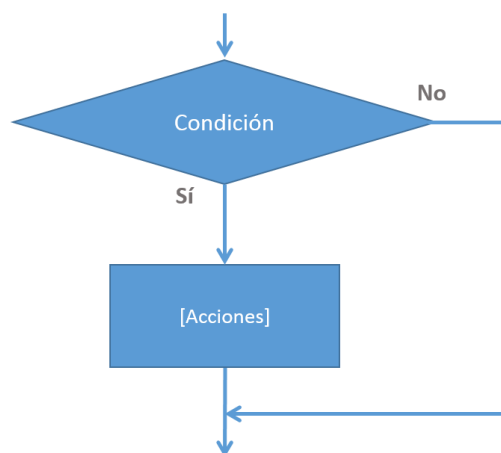


Figura 7. Estructura de control condicional simple.

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].

#### Ejemplo

La Figura 8 muestra el diagrama de flujo que usa la estructura de control condicional simple donde, dados dos números enteros asignados a dos variables, revisa si el valor asignado a la variable *a* es el mayor.

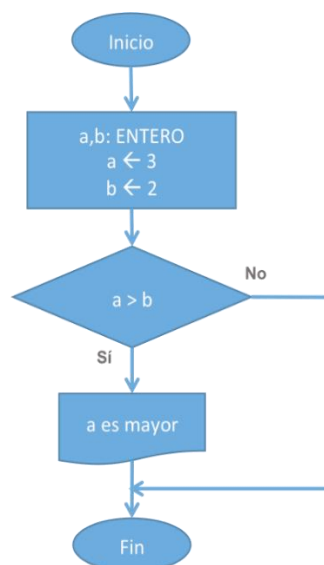



Figura 8. Diagrama de flujo que indica si el valor de la variable *a* es el mayor

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	10/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Prueba de escritorio:

Instrucción	a	b	salida
$a \leftarrow 3$	3		
$b \leftarrow 2$		2	
$a > b$			
			a es mayor

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,

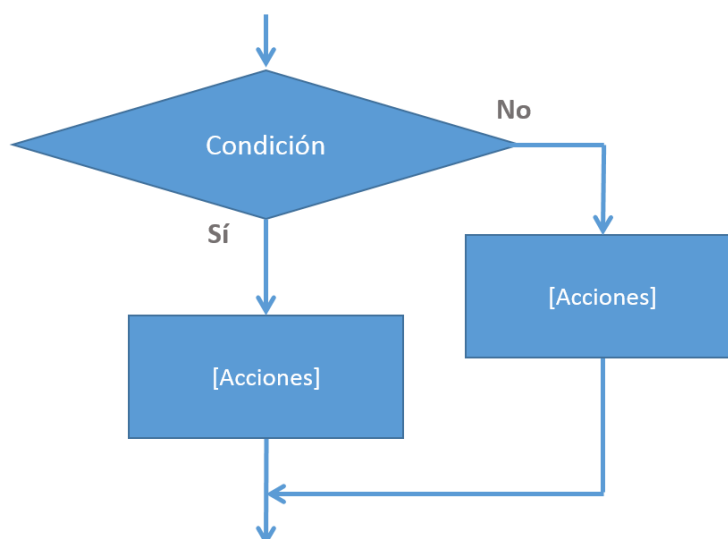



Figura 9. Estructura de control condicional SI-DE LO CONTRARIO

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.

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	11/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

### Ejemplo

La Figura 10 muestra el diagrama de flujo que usa la estructura de control condicional SI-DE LO CONTRARIO donde, dados dos números enteros asignados a dos variables, muestra la variable que almacena el valor mayor.

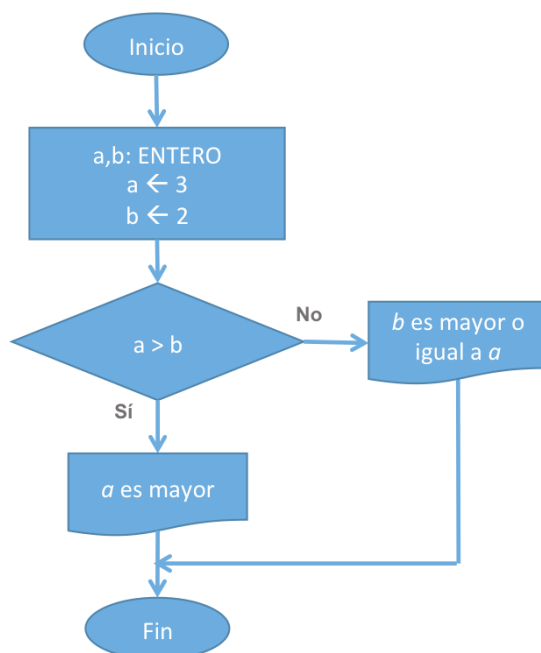



Figura 10. Diagrama de flujo que usa la estructura de control condicional SI-DE LO CONTRARIO

Prueba de escritorio:

Instrucción	a	b	salida
a ← 3	3		
b ← 2		2	
a > b			
			a es mayor

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).

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	12/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

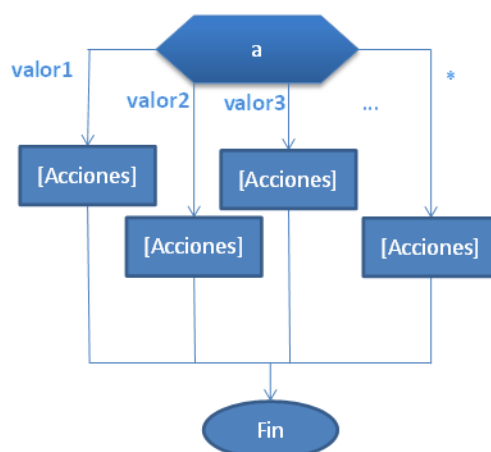


Figura 11. Estructura de control condicional SELECCIONAR-CASO

### Ejemplo

La Figura 12 muestra el diagrama de flujo que usa la estructura de control condicional SELECCIONAR-CASO donde, de acuerdo con el valor de la variable *a*, escribe un mensaje.

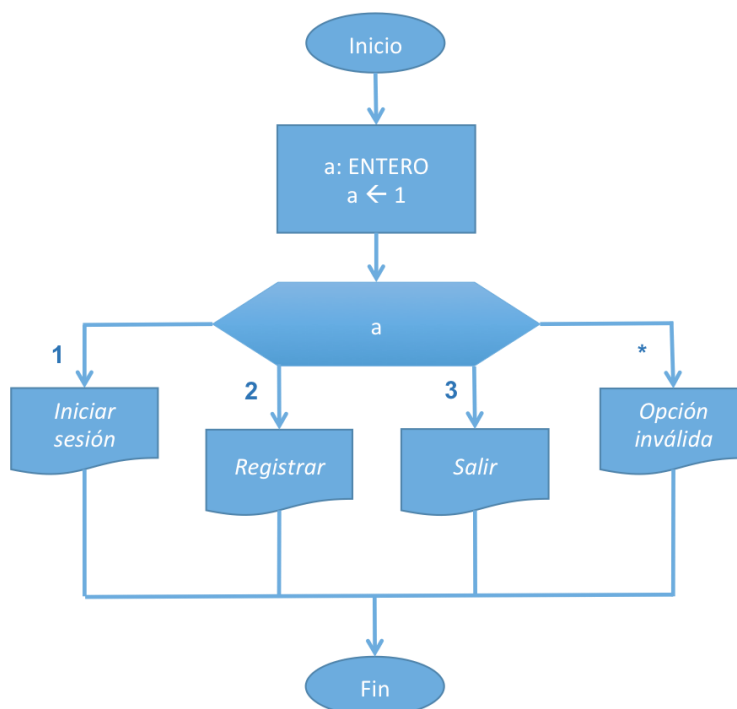



Figura 12. Diagrama de flujo que usa la estructura de control condicional SELECCIONAR-CASO

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	13/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Prueba de escritorio:

Instrucción	a	salida
$a \leftarrow 1$	1	
		Iniciar sesión

### 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.

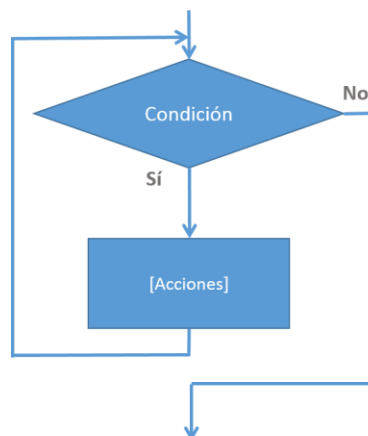



Figura 13. Estructura de control iterativa MIENTRAS

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	14/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

### Ejemplo

La Figura 14 muestra el diagrama de flujo que usa la estructura de control iterativa MIENTRAS donde se escribe el valor de la variable `valorInicial` mientras se cumpla que el valor de `valorInicial` sea menor al valor de `valorFinal`

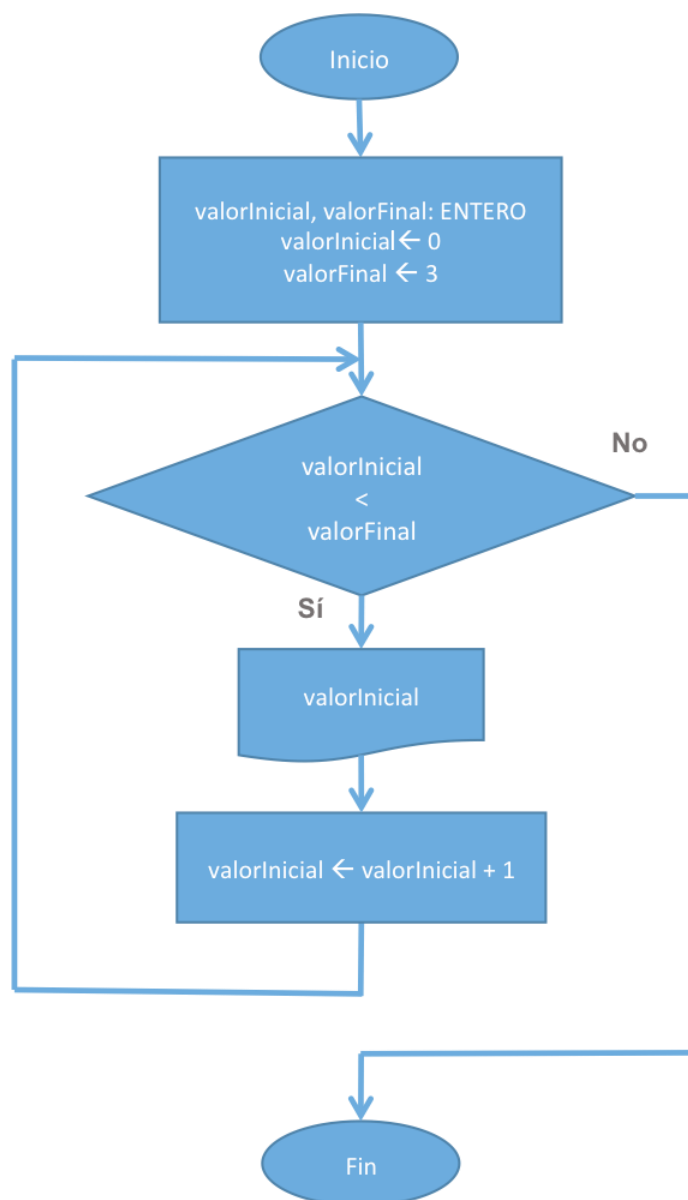



Figura 14. Diagrama de flujo que usa la estructura de control iterativa MIENTRAS

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	15/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Prueba de escritorio:

Instrucción	valorInicial	valorFinal	salida
valorInicial $\leftarrow$ 0	0		
valorFinal $\leftarrow$ 3		3	
valorInicial < valorFinal			
			0
valorInicial $\leftarrow$ valorInicial +1	1		
valorInicial < valorFinal			
			1
valorInicial $\leftarrow$ valorInicial +1	2		
valorInicial < valorFinal			
			2
valorInicial $\leftarrow$ valorInicial +1	3		
valorInicial < valorFinal			

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.

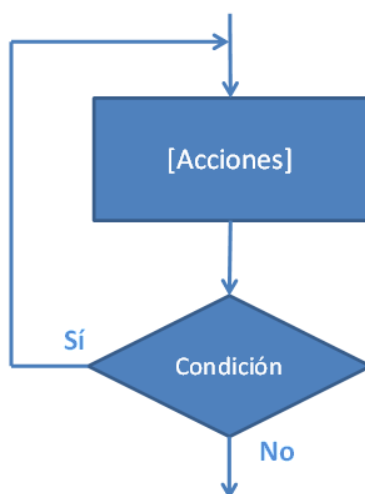



Figura 15. Estructura de control iterativa HACER-MIENTRAS

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	16/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

### Ejemplo

La Figura 16 muestra el diagrama de flujo que usa la estructura de control iterativa HACER-MIENTRAS 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.

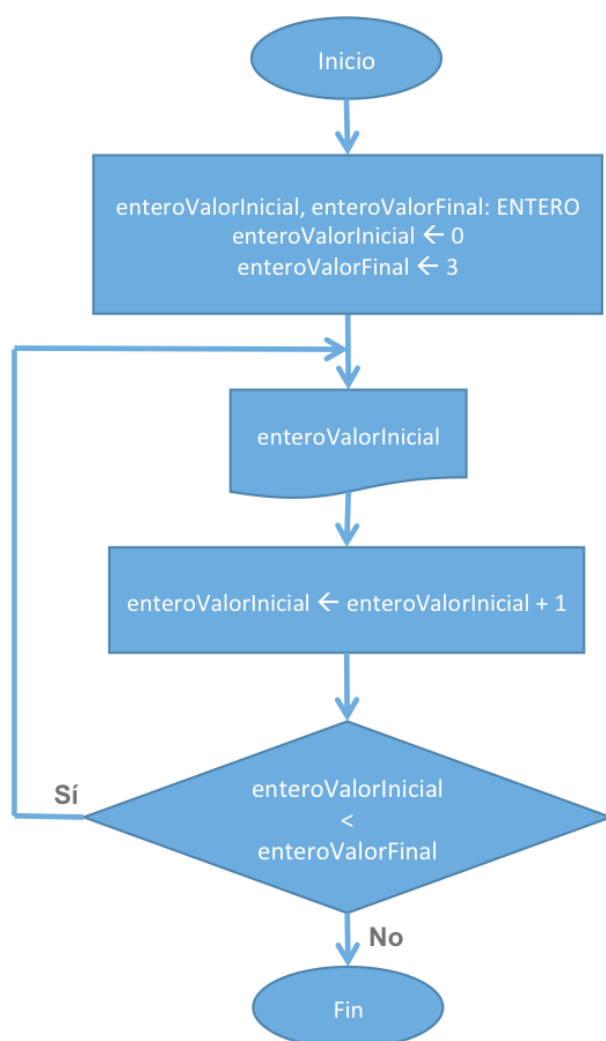




Figura 16. Diagrama de flujo que usa la estructura de control iterativa HACER-MIENTRAS



	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	17/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Prueba de escritorio:

Instrucción	enteroValorInicial	enteroValorFinal	salida
enteroValorInicial $\leftarrow$ 0	0		
enteroValorFinal $\leftarrow$ 3		3	
			0
enteroValorInicial $\leftarrow$ enteroValorInicial +1	1		
enteroValorInicial < enteroValorFinal			
			1
enteroValorInicial $\leftarrow$ enteroValorInicial +1	2		
enteroValorInicial < enteroValorFinal			
			2
enteroValorInicial $\leftarrow$ enteroValorInicial +1	3		
enteroValorInicial < enteroValorFinal			

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	03
		Página	18/18
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## Bibliografía

- Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.



- Metodología de la programación a través de pseudocódigo. Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill

