



universidad  
cenfotec\_  
La U de la informática

# Principios de Programación.

Programación estructurada.  
Modelado y solución algorítmica de problemas.

# Objetivos

1. Comprender el concepto de algoritmo.
2. Conocer las diversas representaciones de los algoritmos.
3. Aprender cuáles son los pasos para desarrollar programas estructurados.
4. Conocer las fases de desarrollo de un sistema de software.

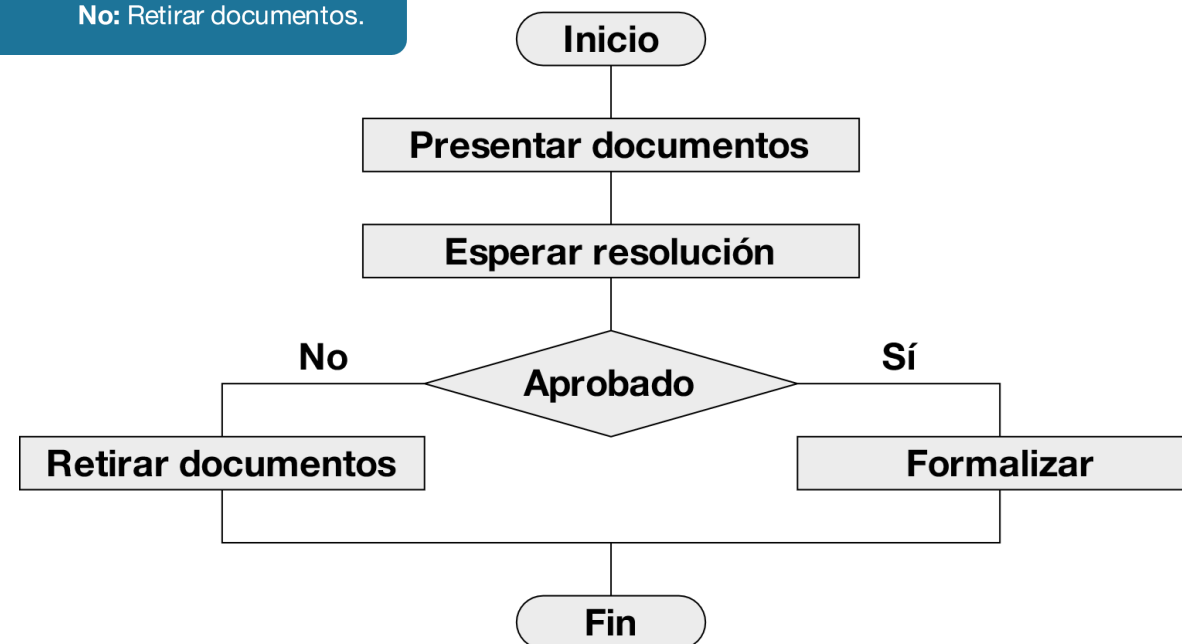
# Los Algoritmos

Un **algoritmo** es una secuencia de instrucciones que, realizados con fidelidad, darán como resultado una tarea realizada.

Los algoritmos son finitos, lo que significa que estas instrucciones tienen un inicio y un final.

## Solicitar un préstamo:

1. Presentar documentos.
2. Esperar resolución.
3. Aprobaron el préstamo:  
**Sí:** Formalizar.  
**No:** Retirar documentos.



# Representaciones de los Algoritmos

## 1. PSEUDOCÓDIGO.

Es la descripción del algoritmo en un lenguaje natural, entendible fácilmente por los humanos.

Las acciones que se describen serán ejecutadas por la computadora cuando el programa sea convertido a un lenguaje de programación.

### Solicitar un préstamo:

1. Presentar documentos.
2. Esperar resolución.
3. Aprobaron el préstamo:

**Sí:** Formalizar.

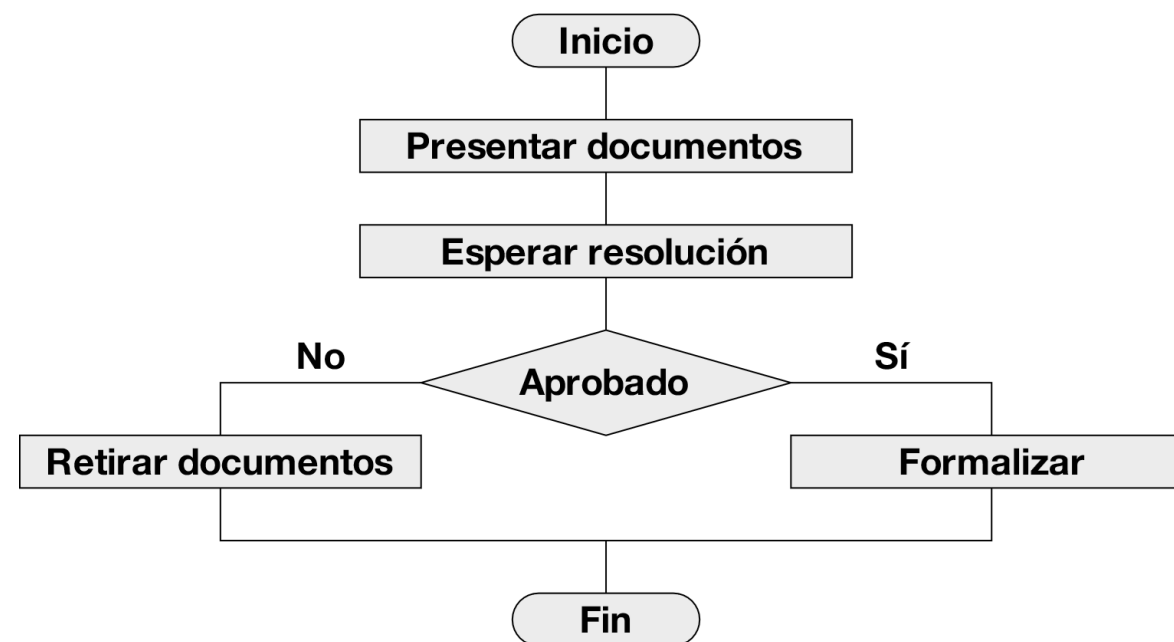
**No:** Retirar documentos.

# Representaciones de los Algoritmos

## 2. DIAGRAMA DE FLUJO.

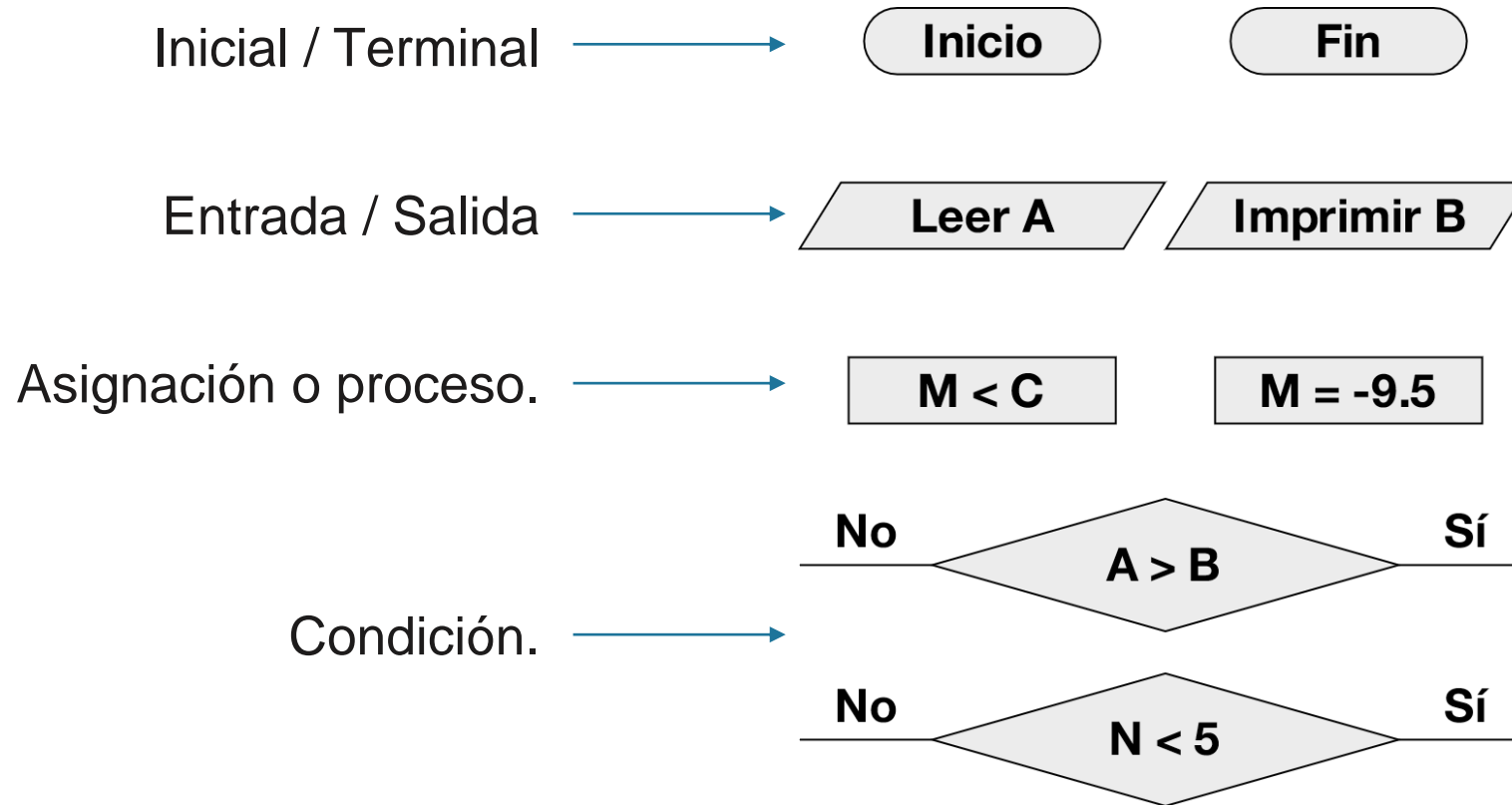
Es el diagrama con un conjunto de símbolos conectados por líneas que representan el algoritmo.

Las líneas de flujo determinan el orden secuencial del algoritmo.



# Representaciones de los Algoritmos

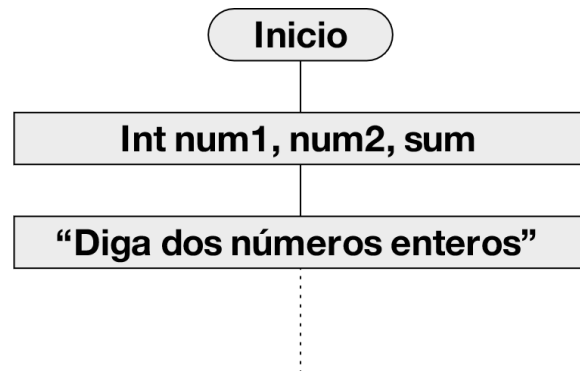
## Símbolos de los diagramas de flujo:



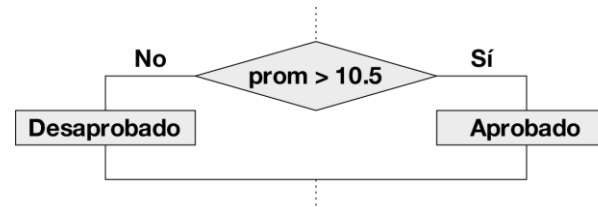
# Representaciones de los Algoritmos

Los algoritmos se fundamentan en tres estructuras de control:

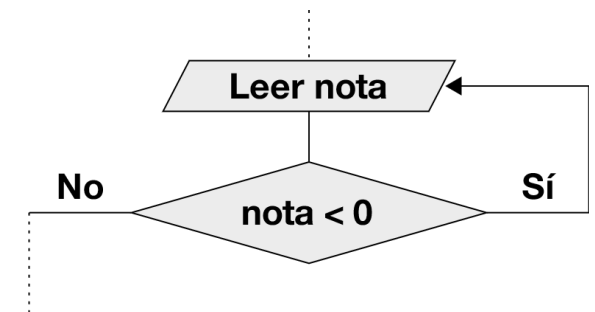
## Estructuras secuenciales



## Estructuras condicionales

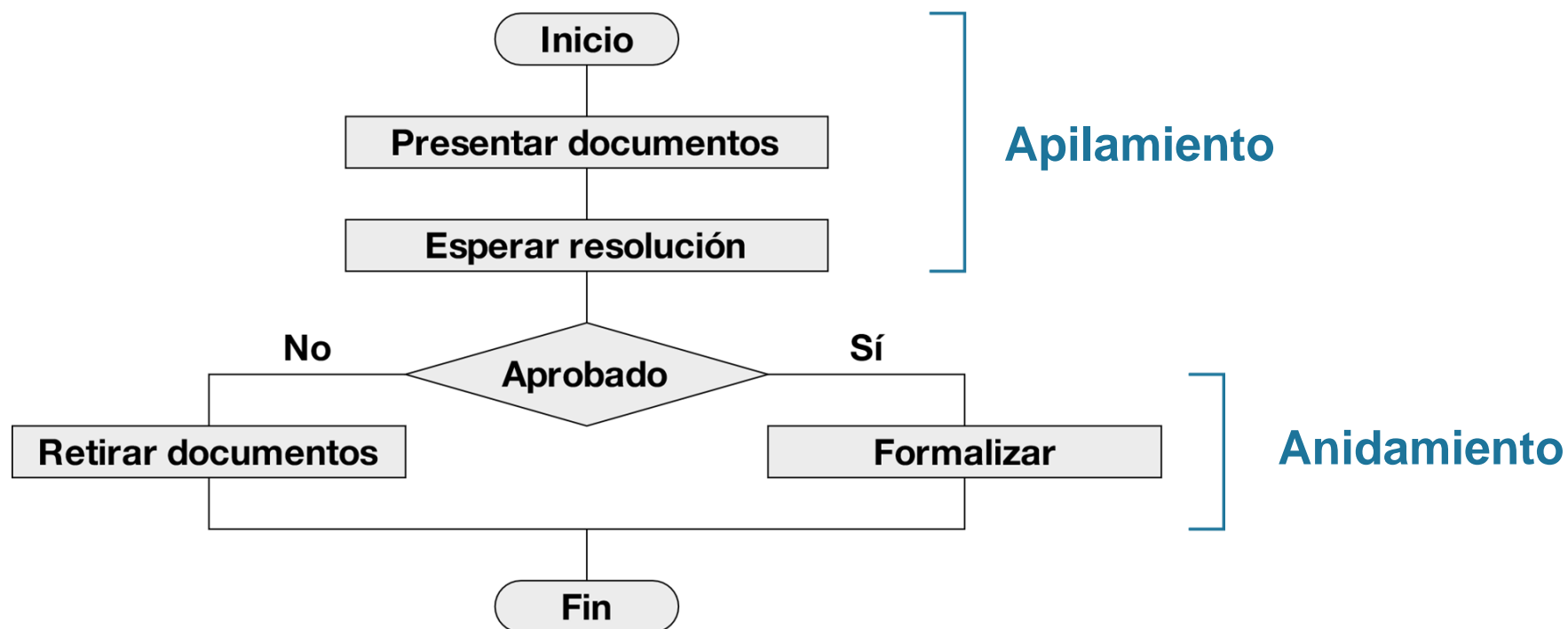


## Estructuras iterativas



# Representaciones de los Algoritmos

Estas estructuras se acomodan entre sí por **apilamiento** (una estructura después de la otra) y por **anidamiento** (una estructura dentro de la otra), para formar algoritmos complejos.

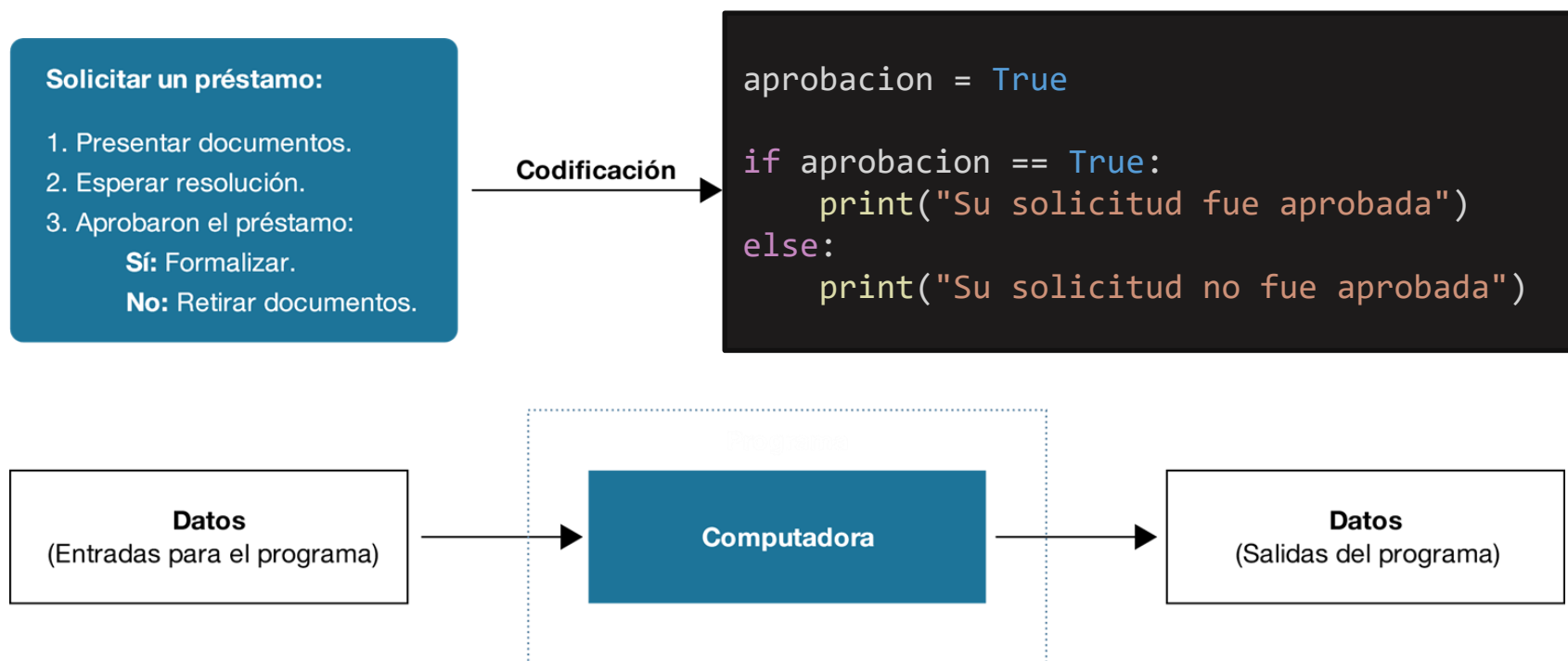




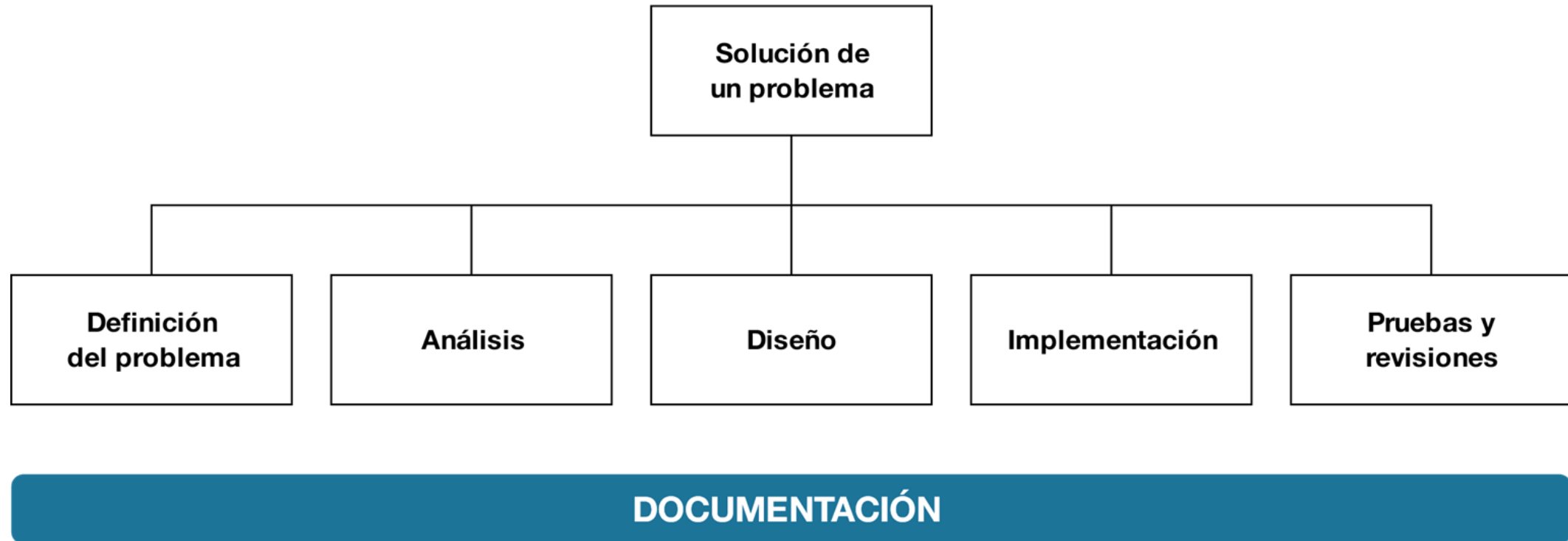
# Representaciones de los Algoritmos

## 3. PROGRAMA.

Es el conjunto de instrucciones a ser seguidas por la computadora, escritas en lenguaje de programación.



# Proceso para crear un programa estructurado



# Proceso para crear un programa estructurado



## 1. DEFINICIÓN DEL PROBLEMA.

Debe contener la descripción del problema bien definido, que pueda ser solucionado computacionalmente, con un inicio y un final claro.

Por ejemplo:

- Ir de la universidad a la casa.
- Cruzar un río y llegar al otro lado.

Un ejemplo no definido puede ser “*hallar la paz mundial*”, ya que no se tienen pasos estructurados ni una meta objetiva que nos indique el final de esta tarea.

# Proceso para crear un programa estructurado



## 2. ANÁLISIS DEL PROBLEMA.

Consiste en entender el problema y estudiarlo para encontrar los conceptos importantes y replantearlo en los siguientes conceptos:

- Comprender la situación.
- Definir insumos o datos de entrada.
- Definir los productos esperados o los datos de salida.
- Especificar el proceso a seguir para que nuestros datos de entrada se conviertan en datos de salida.

# Proceso para crear un programa estructurado



## 3. DISEÑO.

Se establece una solución basada en los conceptos descubiertos durante el análisis.

Se estructura la secuencia de pasos o algoritmo que la computadora deberá de seguir.

Tener en cuenta las varias representaciones del algoritmo:

- Pseudocódigo.
- Diagrama de flujo.

# Proceso para crear un programa estructurado



## 4. IMPLEMENTACIÓN.

Pasar el algoritmo de la solución a un lenguaje de programación, para que sea ejecutado por la computadora.

Se escoge el lenguaje de programación con el cual se desea implementar: C, C++, Basic, Cobol, Java, Visual Basic, Python, etc.

***¿Qué es lo más difícil de realizar, el algoritmo o la codificación?***

# Proceso para crear un programa estructurado



## 5. LISTA DE CHEQUEO Y PRUEBAS.

De varios casos para asegurar el correcto funcionamiento del sistema.

Es posible que se requiera hacer algunos cambios o ajustes a lo largo del tiempo, esto implica que necesitará mantenimiento.

Es importante enfatizar que, en cada una de las actividades del proceso de construcción del sistema, es necesario llevar a cabo una constante **documentación**.

# Proceso para crear un programa estructurado



## RESUMEN PARA CREAR UN PROGRAMA ESTRUCTURADO.

1. **Definición del problema:** generalmente es dado.
2. **Análisis del problema:** entender el problema y plantearlo.
3. **Diseño de la solución:** hacer el algoritmo que soluciones el problema.
4. **Implementación:** del programa correspondiente al algoritmo en lenguaje de programación.
5. **Principios de control de calidad:** hacer revisiones, pruebas, y mantenimiento si es necesario.



# Ejemplo de un programa estructurado

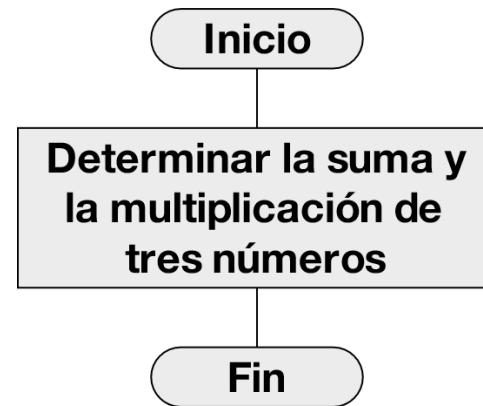
## 1. DEFINICIÓN DEL PROBLEMA.

*"Haga un programa que calcule la suma y la multiplicación de 3 números.  
El programa debe leer los tres números y luego imprimir la suma y la multiplicación de ellos."*

# Ejemplo de un programa estructurado

## 2. ANÁLISIS DEL PROBLEMA.

De la definición del problema se deriva el primer diagrama donde se indica el qué se desea obtener del algoritmo. No se especifican aspectos del cómo. Se describen las entradas (qué se necesita para resolver el problema) y las salidas (qué se obtiene del programa).



# Ejemplo de un programa estructurado

Las entradas y las salidas deben ir en lo posible con nombres significativos. Esto sirve para entender qué hace el programa, reutilizar el código más adelante, hacerle mantenimiento, cambios, etc. Además, es una buena práctica de programación.

Los nombres de las variables deberán cumplir con el **estándar de documentación** dado en el curso.

ENTRADAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Variable que contiene el primer número.	primer_numero	Real	15
Variable que contiene el segundo número.	segundo_numero	Real	10.6
Variable que contiene el tercer número.	tercer_numero	Real	-5

# Ejemplo de un programa estructurado

## SALIDAS

Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Variable que contiene la suma de los 3 números.	suma	Real	20.6
Variable que almacena la multiplicación de los 3 números	multiplicacion	Real	-795

# Ejemplo de un programa estructurado

## 3. DISEÑO DE LA SOLUCIÓN.

### Pseudocódigo.

**Algoritmo:** calculos

**Variables**

```
primer_numero, segundo_numero, tercer_numero : real  
suma : real  
multiplicacion : real
```

**Inicio**

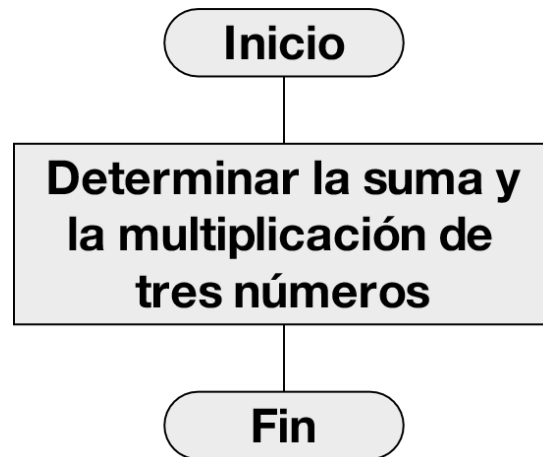
```
Leer primer_numero, segundo_numero, tercer_numero  
suma = primer_numero + segundo_numero + tercer_numero  
multiplicacion = primer_numero * segundo_numero * tercer_numero  
Imprimir suma, multiplicacion
```

**Fin**

# Ejemplo de un programa estructurado

## 3. DISEÑO DE LA SOLUCIÓN.

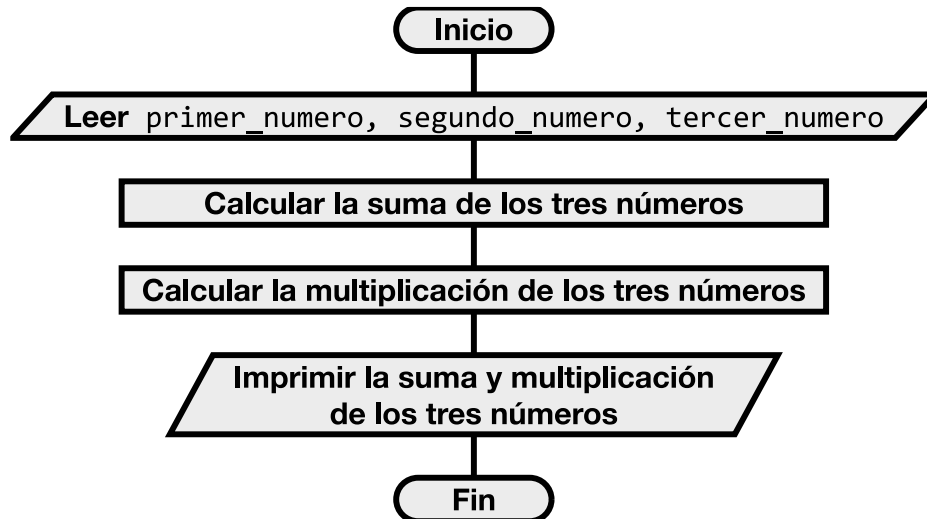
### Diagrama de flujo.



**Paso 1.** Descripción clara del problema en una frase.

La frase debe empezar con el verbo en infinitivo "***Determinar***".

# Ejemplo de un programa estructurado



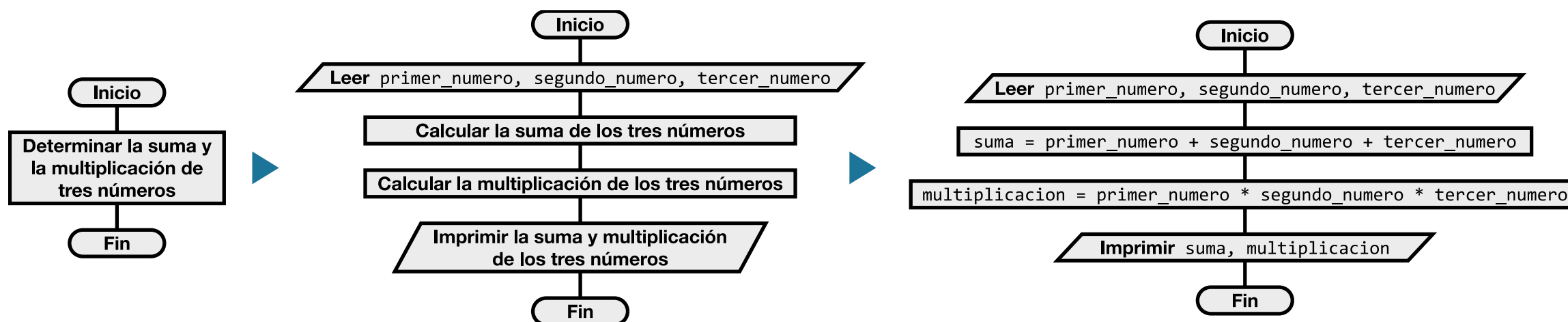
**Paso 2.** Separar las entradas del programa, el procesamiento de los datos y las salidas del programa. Si fuera necesario para clarificar las tareas intermedias, se describen con un verbo en infinitivo (ejemplo: calcular).



**Paso 3.** Se desglosa el procesamiento en pasos sencillos: operaciones aritméticas, lectura e impresión.

# Ejemplo de un programa estructurado

Resumen del proceso:





# Ejemplo de un programa estructurado

## 4. IMPLEMENTACIÓN.

La implementación de todos los algoritmos se realizarán en el lenguaje de programación **Python**, el cual se repasará más adelante.



# Ejemplo de un programa estructurado

## 5. CONTROL DE CALIDAD.

Actividades dirigidas a asegurar que la solución (sistema, programa, etc.) tenga las características deseables:

- Cumple con los requisitos.
- Está libre de defectos.
- Funciona correctamente.
- Es eficiente.
- Etcétera.

**Casos de prueba:** Para probar el programa con diferentes datos de entrada.

Caso #	Entradas	Resultados esperados	Resultados obtenidos	Encargado	
1	20	6.86	6.86	Probado por:	Alan Mathison Turing
	10.6			Fecha:	25 de agosto de 2020
	-5				
2	2.5	164.5	164.5	Probado por:	Alan Mathison Turing
	-9			Fecha:	25 de agosto de 2020
	500				

# Ejemplo de un programa estructurado

**Revisiones:** Lista de chequeo de revisiones aplicada a cada solución.

Aspecto	Descripción	Revisión #1		Revisión #2	
Compleitud	¿Están implementadas todas las funcionalidades especificadas para este programa?	Revisado por:	Alan Mathison Turing	Revisado por:	
		Fecha:	25 de agosto de 2020	Fecha:	
		# Defectos:	0	# Defectos:	
Condiciones no contempladas	¿Hay decisiones no exhaustivas?	Revisado por:	Alan Mathison Turing	Revisado por:	
		Fecha:	25 de agosto de 2020	Fecha:	
		# Defectos:	0	# Defectos:	
Comparaciones	¿Se hacen comparaciones entre variables de tipos incompatibles?	Revisado por:	Alan Mathison Turing	Revisado por:	
		Fecha:	25 de agosto de 2020	Fecha:	
		# Defectos:	0	# Defectos:	



universidad  
cenfotec\_  
La U de la informática