

Programación I

Introducción

Iván Cantador

Escuela Politécnica Superior

Universidad Autónoma de Madrid

- Algoritmos
- Ejemplos de algoritmos
- Estructura de un programa
- Lenguajes de programación

- **Algoritmos**
 - **Definición de algoritmo**
 - Pseudocódigo y elementos de un algoritmo
 - Diagrama de flujo de un algoritmo
- Ejemplos de algoritmos
- Estructura de un programa
- Lenguajes de programación

Definición de algoritmo (I)

- ¿Cómo hacer una paella?



- **Receta: Paella valenciana**

- **Ingredientes para 6 personas**

- 425 gramos de arroz
- 150 gramos de pollo
- 3 cucharadas de tomate
- 1,5 litros de agua
- 50 gramos de alcachofas
- 0,5 cucharitas de pimentón
- 5 cucharadas de aceite
- 50 gramos de judías verdes
- 1 pizca de azafrán

- **Instrucciones de elaboración**

1. Se calienta el aceite en la paella, y cuando empieza a humear, se echa el pollo troceado, dorándolo a fuego medio.
2. A continuación, se añade la verdura troceada y se sofríe durante un par de minutos.
3. Seguidamente, haciendo un hueco en medio de la paellera, se sofríe el tomate y cuando empieza a oscurecer, se echa el pimentón, con cuidado de que no se queme.
4. Se mueve bien y se añade agua hasta el borde de la paellera.
5. Se deja cocer la paella durante treinta minutos, rectificando de sal.
6. Se añade el arroz y las hebras de azafrán, teniendo cuidado de repartir bien todo el arroz por la paella, y se deja hervir a fuego vivo hasta que el arroz esté en su punto y el caldo se evapore por completo.
7. Se deja reposar la paella cinco minutos, tapada por un paño.

• Receta: Paella valenciana

• Ingredientes para 6 personas

- 425 gramos de arroz
- 150 gramos de pollo
- 3 cucharadas de tomate
- 1,5 litros de agua
- 50 gramos de alcachofas
- 0,5 cucharitas de pimentón
- 5 cucharadas de aceite
- 50 gramos de judías verdes
- 1 pizca de azafrán

• Instrucciones de elaboración

1. Se calienta el **aceite** en la paella, y cuando empieza a humear, se echa el **pollo** troceado, dorándolo a fuego medio.
2. A continuación, se añade la **verdura** troceada y se sofríe durante un par de minutos.
3. Seguidamente, haciendo un hueco en medio de la paellera, se sofríe el **tomate** y cuando empieza a oscurecer, se echa el **pimentón**, con cuidado de que no se quemé.
4. Se mueve bien y se añade **agua** hasta el borde de la paellera.
5. Se deja cocer la **paella** durante treinta minutos, rectificando de **sal**.
6. Se añade el **arroz** y las hebras de **azafrán**, teniendo cuidado de repartir bien todo el **arroz** por la paella, y se deja hervir a fuego vivo hasta que el **arroz** esté en su punto y el caldo se evapore por completo.
7. Se deja reposar la **paella** cinco minutos, tapada por un paño.

• Receta: Paella valenciana

• Ingredientes para 6 personas

- 425 gramos de arroz
- 150 gramos de pollo
- 3 cucharadas de tomate
- 1,5 litros de agua
- 50 gramos de alcachofas
- 0,5 cucharitas de pimentón
- 5 cucharadas de aceite
- 50 gramos de judías verdes
- 1 pizca de azafrán

• Instrucciones de elaboración

1. Se **calienta** el **aceite** en la paella, y cuando empieza a humear, se **echa** el **pollo** troceado, **dorándolo** a fuego medio.
2. A continuación, se **añade** la **verdura** troceada y se **sofríe** durante un par de minutos.
3. Seguidamente, haciendo un hueco en medio de la paellera, se **sofríe** el **tomate** y cuando empieza a oscurecer, se **echa** el **pimentón**, con cuidado de que no se quemé.
4. Se **mueve** bien y se **añade agua** hasta el borde de la paellera.
5. Se deja **cocer** la **paella** durante treinta minutos, **rectificando** de **sal**.
6. Se **añade** el **arroz** y las hebras de **azafrán**, teniendo cuidado de **repartir** bien todo el **arroz** por la paella, y se deja **hervir** a fuego vivo hasta que el **arroz** esté en su punto y el caldo se evapore por completo.
7. Se deja **reposar** la **paella** cinco minutos, tapada por un paño.

- **Receta: Paella valenciana**

- **Ingredientes para 6 personas**

- 425 gramos de arroz
- 150 gramos de pollo
- 3 cucharadas de tomate
- 1,5 litros de agua
- 50 gramos de alcachofas
- 0,5 cucharitas de pimentón
- 5 cucharadas de aceite
- 50 gramos de judías verdes
- 1 pizca de azafrán

- **Instrucciones de elaboración**

1. Se calienta el **aceite** en la paella, y **cuando empieza a humear**, se echa el pollo troceado, dorándolo a fuego medio.
2. A continuación, se añade la **verdura** troceada y se sofríe durante un **par de minutos**.
3. Seguidamente, haciendo un hueco en medio de la paellera, se sofríe el **tomate** y **cuando empieza a oscurecer**, se echa el pimentón, con cuidado de que no se quemé.
4. Se mueve bien y se añade agua hasta el borde de la paellera.
5. Se deja cocer la **paella** durante **treinta minutos**, rectificando de sal.
6. Se añade el arroz y las hebras de azafrán, teniendo cuidado de repartir bien todo el arroz por la paella, y se deja hervir a fuego vivo hasta que el **arroz** esté **en su punto** y el **caldo se evapore** por completo.
7. Se deja reposar la **paella cinco minutos**, tapada por un paño.

• Receta: Paella valenciana

• Ingredientes para 6 personas

- 425 gramos de arroz
- 1,5 litros de agua
- 5 cucharadas de aceite
- 150 gramos de pollo
- 50 gramos de alcachofas
- 50 gramos de judías verdes
- 3 cucharadas de tomate
- 0,5 cucharitas de pimentón
- 1 pizca de azafrán

• Instrucciones de elaboración

1. Se calienta el aceite en la paella, y cuando empieza a humear, se echa el pollo troceado, dorándolo a fuego medio.
2. A continuación, se añade la verdura troceada y se sofríe durante un par de minutos.
3. Seguidamente, haciendo un hueco en medio de la paellera, se sofríe el tomate y cuando empieza a oscurecer, se echa el pimentón, con cuidado de que no se quemé.
- ~~4. Se mueve bien y se añade agua hasta el borde de la paellera.~~
4. Se deja cocer la paella durante treinta minutos, rectificando de sal.
5. Se mueve bien y se añade agua hasta el borde de la paellera.
6. Se añade el arroz y las hebras de azafrán, teniendo cuidado de repartir bien todo el arroz por la paella, y se deja hervir a fuego vivo hasta que el arroz esté en su punto y el caldo se evapore por completo.
7. Se deja reposar la paella cinco minutos, tapada por un paño.

El orden importa

- Un **algoritmo** es una secuencia de operaciones bien definida, ordenada y finita que permiten hallar la solución a un problema
 - **Bien definida:** las operaciones se enuncian en un lenguaje formal, atendiendo a una sintaxis arbitraria (acordada)
 - **Ordenada:** las operaciones se ejecutan de manera secuencial, siguiendo un orden establecido
 - **Finita:** el número de operaciones a ejecutar para resolver el problema asociado al algoritmo es limitado

• Elementos de un algoritmo

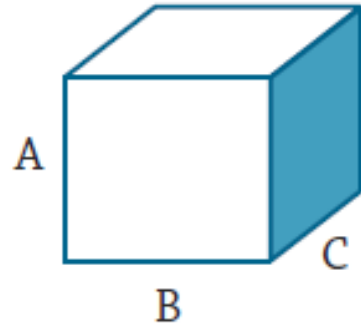
- Una definición de la **entrada** del algoritmo
 - En la receta de la paella: ingredientes, utensilios de cocina necesarios
- Un **lenguaje** formal en el que definir el algoritmo
 - En la receta de la paella: el castellano
- Una lista ordenada de **sentencias** (instrucciones) con las que resolver el problema asociado al algoritmo
 - En la receta de la paella: las instrucciones de preparación del plato
- Una definición de la **salida** del algoritmo
 - En la receta de la paella: el plato en sí si se hubiesen ejecutado satisfactoriamente todas las instrucciones de preparación, o “nada” en caso de que alguna instrucción no se pudiera haber ejecutado con éxito

- **Algoritmos**

- Definición de algoritmo
- **Pseudocódigo y elementos de un algoritmo**
- Diagrama de flujo de un algoritmo
- Ejemplos de algoritmos
- Estructura de un programa
- Lenguajes de programación

- **Pseudocódigo** = pseudo (supuesto) + código (instrucciones)
 - Describe un algoritmo utilizando una mezcla de frases en lenguaje natural, instrucciones de programación y palabras clave
 - Su objetivo es permitir a una persona (al programador) centrarse en los aspectos lógicos de la solución a un problema, evitando entrar en aspectos concretos de la sintaxis del lenguaje de programación a usar
 - Varía de un programador a otro pues no hay una estructura estándar
 - Supone un mecanismo ágil para el estudio y diseño de programas de ordenador

- Pseudocódigo para determinar el volumen de una caja de dimensiones A, B y C (Pinales & Velázquez, 2014)



1. Inicio.
2. Leer las medidas A, B y C.
3. Realizar el producto de $A * B * C$ y guardarlo en V ($V = A * B * C$).
4. Escribir el resultado V.
5. Fin.

Pinales, F.J., Velázquez, C.E. 2014. **Algoritmos resueltos con diagramas de flujo y pseudocódigo.** Universidad Autónoma de Aguascalientes, México. ISBN: 978-607-8285-96-9.

<http://www.uaa.mx/direcciones/dgdv/editorial/catalogo/docs/algoritmos.pdf>

- **Problema: *calcular la suma de los elementos de una lista de número enteros***

Ejemplo de pseudocódigo (II)

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Ejemplo de pseudocódigo (II)

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Salida: número entero SUMA

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Salida: número entero SUMA

Instrucciones:

Inicializar SUMA a 0

Ejemplo de pseudocódigo (II)

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Salida: número entero SUMA

Instrucciones:

Inicializar SUMA a 0

Recorrer los elementos de LISTA

Ejemplo de pseudocódigo (II)

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Salida: número entero SUMA

Instrucciones:

Inicializar SUMA a 0

Recorrer los elementos de LISTA

Para cada ELEMENTO recorrido: sumar ELEMENTO a SUMA

Ejemplo de pseudocódigo (II)

- **Calcular la suma de los elementos de una lista de número enteros**
 - Pseudocódigo - 1ª opción

Entrada: lista de números enteros LISTA

Salida: número entero SUMA

Instrucciones:

Inicializar SUMA a 0

Recorrer los elementos de LISTA

Para cada ELEMENTO recorrido: sumar ELEMENTO a SUMA

Devolver SUMA

Ejemplo de pseudocódigo (III)

- **Calcular la suma de los elementos de una lista de número enteros**
 - **Pseudocódigo - 2ª opción**

Entrada: lista de enteros LISTA

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

Para cada ELEMENTO de LISTA hacer:

$SUMA \leftarrow SUMA + ELEMENTO$

Devolver SUMA

Ejemplo de pseudocódigo (IV)

- **Calcular la suma de los elementos de una lista de número enteros**
 - **Pseudocódigo - 3ª opción**

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

--

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

SUMA \leftarrow 0

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

 SUMA \leftarrow SUMA + LISTA[i]

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

0

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

0

i

1

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

0

i

1

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

LISTA[i] = 8

SUMA

8

i

1

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

8

i

2

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

8

i

2

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

LISTA[i] = 5

SUMA

13

i

2

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

13

i

3

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

13

i

3

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

LISTA[i] = 7

SUMA

20

i

3

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

20

i

4

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

20

i

4

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

LISTA[i] = 8

SUMA

28

i

4

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

28

i

5

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

28

i

5

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

LISTA[i] = 4

SUMA

32

i

5

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

32

i

6

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq longitud(LISTA)$) hacer:

$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

$longitud(LISTA) = 5$

SUMA

32

i

6

Ejemplo de pseudocódigo (V)

- Calcular la suma de los elementos de una lista de número enteros
 - “Ejecución”

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA

LISTA

8	5	7	8	4
---	---	---	---	---

SUMA

32

i

6

Del pseudocódigo al código

- Ejemplo de codificación de pseudocódigo a una función en lenguaje de programación C

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:

$SUMA \leftarrow SUMA + \text{LISTA}[i]$

$i \leftarrow i + 1$

Devolver SUMA



```
int sumar(int lista[], int longitud)
{
    int suma = 0;
    int i = 1;
    while ( i <= longitud )
    {
        suma = suma + lista[i-1];
        i++;
    }
    return suma;
}
```

- Elementos vistos en el ejemplo de pseudocódigo dado

- **Argumentos de entrada**

LISTA

- **Variables**

SUMA, i

- **Asignaciones**

$i \leftarrow 1, i \leftarrow i + 1$

$SUMA \leftarrow SUMA + LISTA[i]$

- **Clausulas condicionales**

$i \leq \text{longitud}(LISTA)$

- **Elementos de repetición (bucles)**

Mientras ... hacer ...

- **Palabras reservadas**

entero, mientras, hacer, devolver

- **Instrucciones**

Asignaciones, operaciones, comparaciones

- **Funciones:** ejecutan su propio algoritmo

$\text{longitud}(x)$

Entrada: entero LISTA[]

Salida: entero SUMA

Instrucciones:

$SUMA \leftarrow 0$

$i \leftarrow 1$

Mientras ($i \leq \text{longitud}(LISTA)$) hacer:

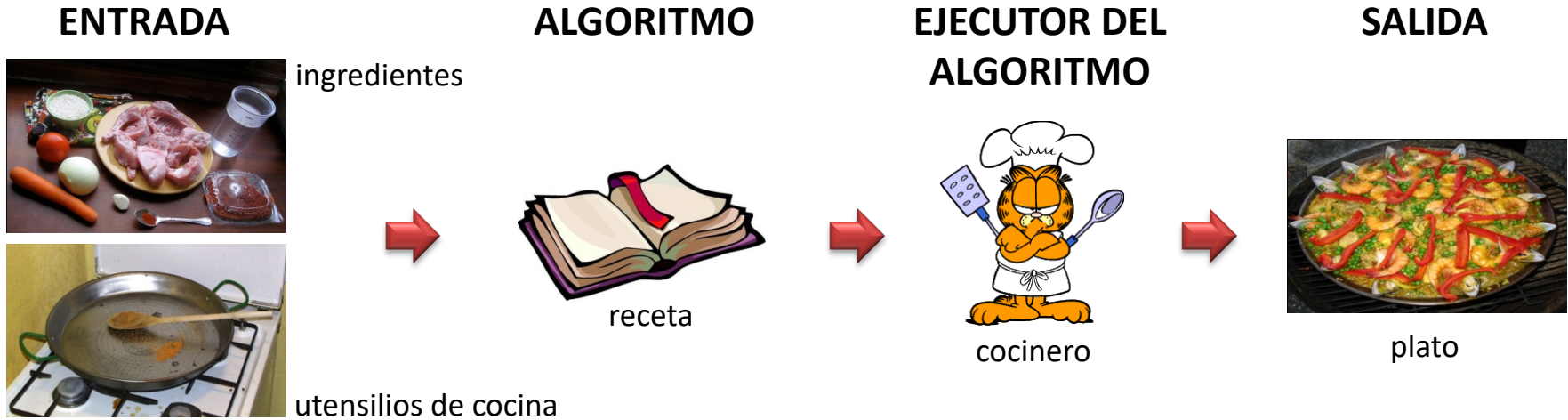
$SUMA \leftarrow SUMA + LISTA[i]$

$i \leftarrow i + 1$

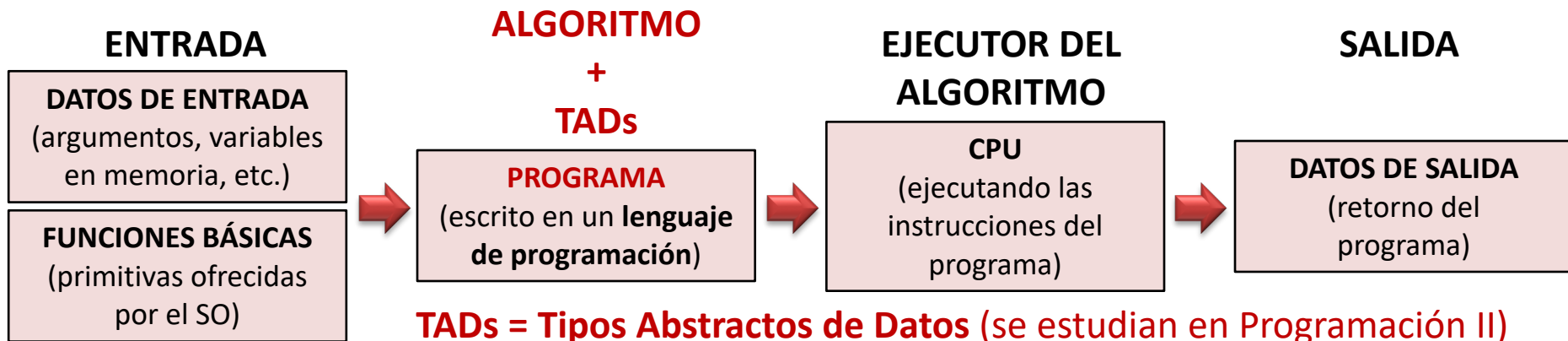
Devolver SUMA

- **Algoritmo: de la vida real al programa de ordenador**

- Resolución de un problema en la vida real



- Resolución de un problema en un ordenador



TADs = Tipos Abstractos de Datos (se estudian en Programación II)

- **Elaboración de un algoritmo**

- Definición de la **entrada** del algoritmo
 - ¿Qué datos de entrada se requieren?
 - ¿De qué tipos son los datos de entrada?
 - ¿Cómo se reciben los datos de entrada?
- Definición de la **salida** del algoritmo
 - ¿Qué tipo de salida se produce?
 - ¿Cómo se comunica la salida producida?
 - ¿Qué pasa si ocurre algún error en la ejecución de las sentencias?
- Especificación de una lista ordenada de **sentencias** con las que resolver el problema asociado al algoritmo
 - ¿Cuáles son las instrucciones precisas a ejecutar?
- Uso de un **lenguaje** formal en el que definir el algoritmo
 - ¿Qué lenguaje de programación usamos? C en “Programación I”
 - ¿Cómo es la sintaxis del lenguaje de programación a usar?

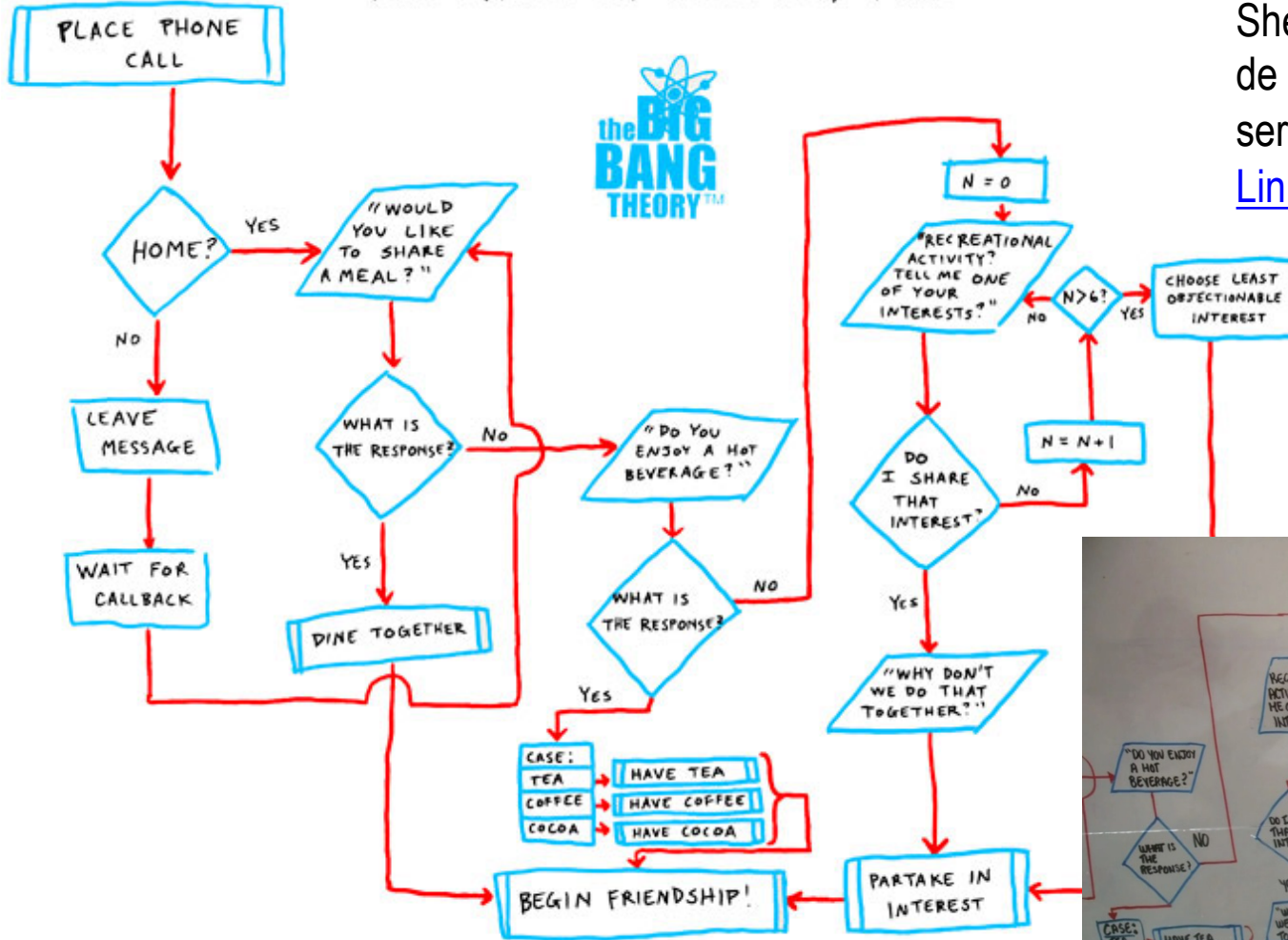
- **Algoritmos**
 - Definición de algoritmo
 - Pseudocódigo y elementos de un algoritmo
 - **Diagrama de flujo de un algoritmo**
- Ejemplos de algoritmos
- Estructura de un programa
- Lenguajes de programación

- **Diagrama de flujo** = representación gráfica de un algoritmo
 - Se utiliza no sólo en programación, sino también en disciplinas como la economía, industria y psicología cognitiva
 - El **Lenguaje Unificado de Modelado (UML)** es el lenguaje de modelado de sistemas software más utilizado en la actualidad; está respaldado por el Object Management Group (OMG), <http://www.omg.org/spec/UML>

Diagrama de flujo de un algoritmo (II)

THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



The Big Bang Theory
temporada 2, episodio 13

Sheldon decide hacerse amigo de Kripke para acceder a cierto servidor de la universidad

[Link a vídeo en YouTube](#)

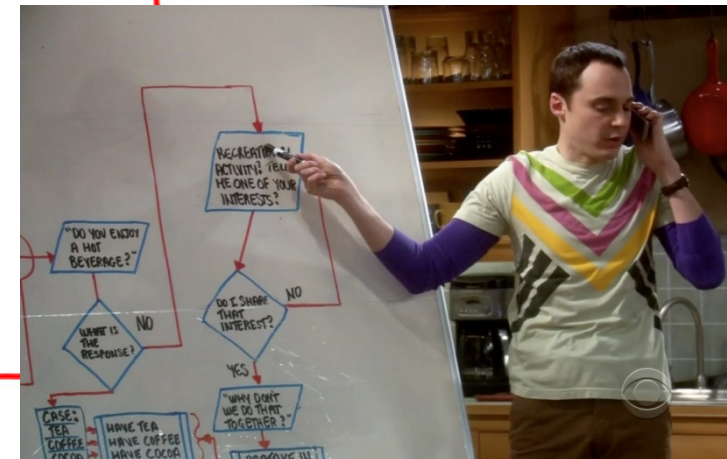




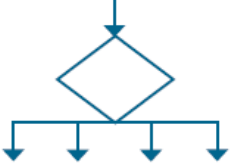





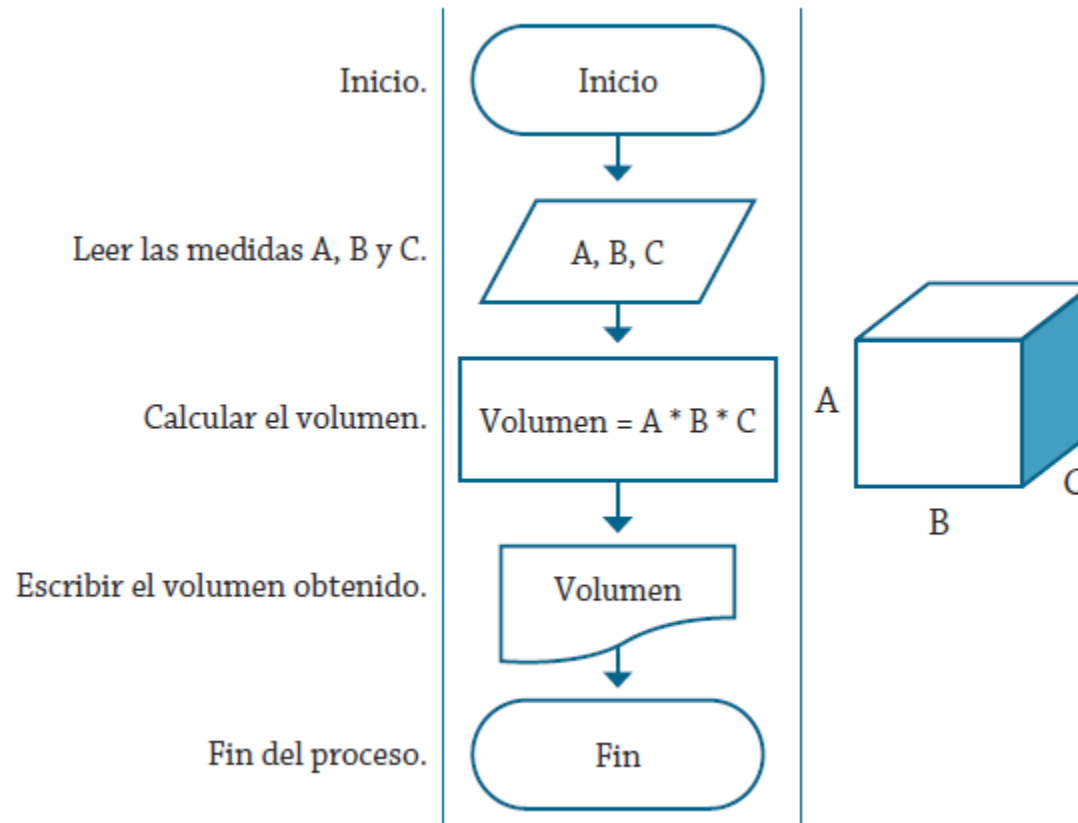
Diagrama de flujo de un algoritmo (III)

- Principales símbolos usados en diagramas de flujo

Símbolo	Significado
	Terminal /Inicio.
	Entrada de datos.
	Proceso.
	Decisión.
	Decisión múltiple.
	Imprimir resultados.
	Flujo de datos.
	Conectores.

Símbolo	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
^	Exponenciación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
< >	Diferente que
=	Igual que





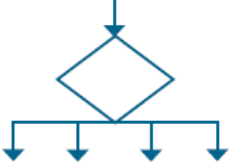



- Ejemplo de diagrama de flujo (Pinales & Velázquez, 2014)



- Algoritmos
- **Ejemplos de algoritmos (Pinales & Velázquez, 2014)**
 - **Algoritmos con estructuras secuenciales**
 - Algoritmos con estructuras selectivas
 - Algoritmos con estructuras repetitivas
- Estructura de un programa
- Lenguajes de programación

Diagrama de flujo de un algoritmo

- Principales símbolos usados en diagramas de flujo

Símbolo	Significado
	Terminal /Inicio.
	Entrada de datos.
	Proceso.
	Decisión.
	Decisión múltiple.
	Imprimir resultados.
	Flujo de datos.
	Conectores.

Símbolo	Operación
+	Suma
-	Resta
*	Multipliación
/	División
^	Exponenciación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
< >	Diferente que
=	Igual que

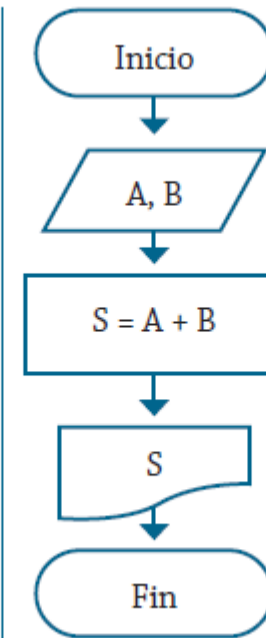
- Ej. 1: obtener la suma de dos números enteros

Pseudocódigo

1. Inicio
2. Leer A, B
3. Hacer $S = A + B$
4. Escribir S
5. Fin

Diagrama de flujo

Se hace la suma de los números y el resultado se asigna a S.



Se leen los números por sumar A y B.

Se escribe el resultado de la suma S.

Nombre de la variable	Descripción	Tipo
A	Primer número para sumar	Entero
B	Segundo número para sumar	Entero
S	Resultado de la suma	Entero

- Ej. 2: determinar la media de cuatro calificaciones

Pseudocódigo

Diagrama de flujo

Nombre de la variable	Descripción	Tipo
C1, C2, C3, C4	Calificaciones obtenidas	Real
S	Suma de calificaciones	Real
P	Promedio calculado	Real

- Ej. 2: determinar la media de cuatro calificaciones

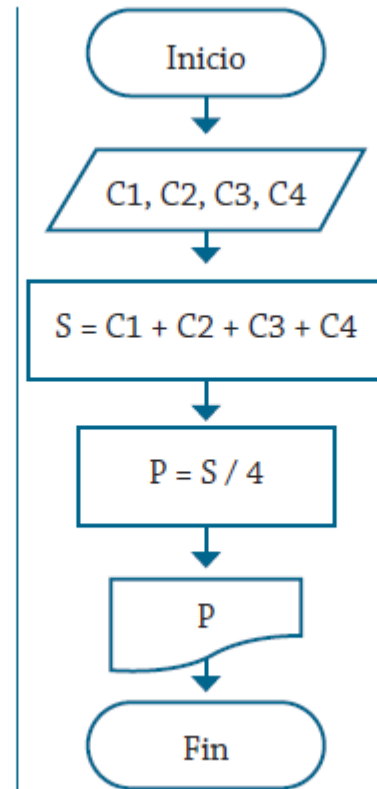
Pseudocódigo

- Inicio
- Leer C1, C2, C3, C4
- Hacer $S = C1 + C2 + C3 + C4$
- Hacer $P = S/4$
- Escribir P
- Fin

Diagrama de flujo

Se leen las calificaciones obtenidas C1, C2, C3, C4.

Se obtiene el promedio de las calificaciones y se asigna a P.



Se realiza la suma de las calificaciones respectivas y se asigna a S.

Se escribe el promedio obtenido P.

Nombre de la variable	Descripción	Tipo
C1, C2, C3, C4	Calificaciones obtenidas	Real
S	Suma de calificaciones	Real
P	Promedio calculado	Real

• Ej. 3: calcular el área de un círculo

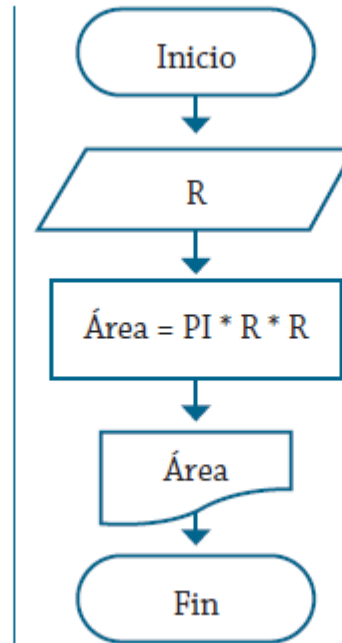
Pseudocódigo

1. Inicio
2. Leer R
3. Hacer $\text{Área} = \text{PI} * \text{R} * \text{R}$
4. Escribir Área
5. Fin

Diagrama de flujo

Se lee el radio de la circunferencia.

Se aplica la fórmula para obtener el área.



PI = 3.1416

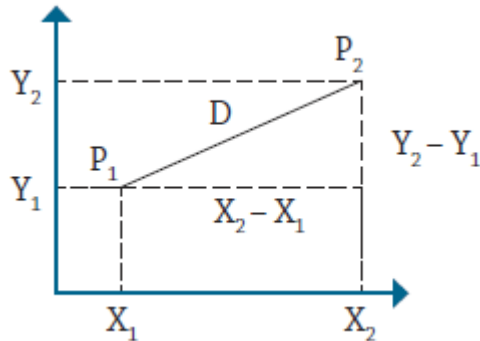
Se escribe el resultado del área obtenida.

Nombre de la variable	Descripción	Tipo
R	Radio de la circunferencia	Real
PI	El valor de 3.1416	Real
Área	Área de la circunferencia	Real

Ej. 4: calcular la distancia entre dos puntos

Pseudocódigo

1. Inicio
2. Leer X_1, Y_1
3. Leer X_2, Y_2
4. Hacer $X = X_2 - X_1$
5. Hacer $Y = Y_2 - Y_1$
6. Hacer $D = \text{SQRT}(X * X + Y * Y)$
7. Escribir D
8. Fin



Nombre de la variable	Descripción	Tipo
X_1, X_2	Abscisas	Real
Y_1, Y_2	Ordenadas	Real
X	Cateto de las abscisas	Real
Y	Cateto de las ordenadas	Real
D	Distancia entre puntos	Real

Diagrama de flujo

Se lee coordenadas del punto 1.

Se lee coordenadas del punto 2.

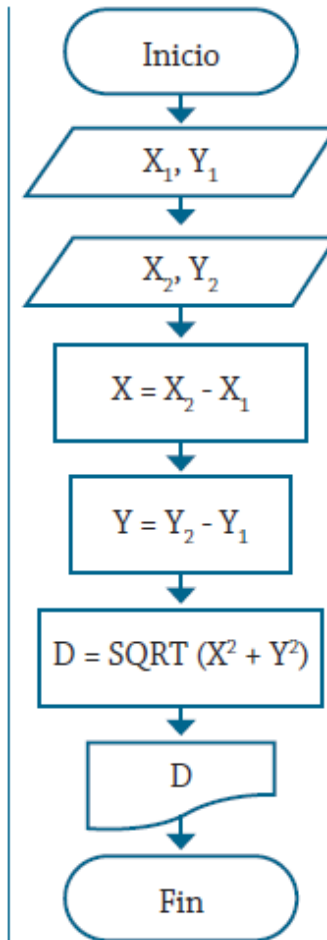
Se calcula el cateto de las abscisas.

Se lee coordenadas del punto 2.

Se calcula el cateto de las ordenadas.

FUNCIÓN
SQRT ⇒ raíz cuadrada.

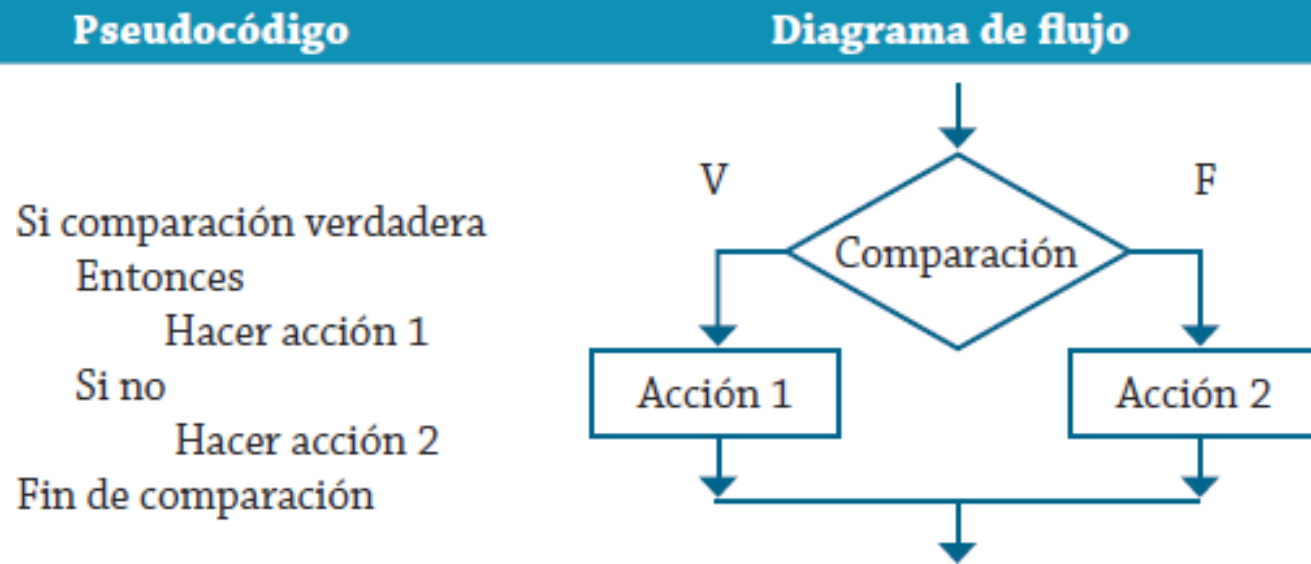
Se escribe la distancia entre los dos puntos.



- Algoritmos
- **Ejemplos de algoritmos (Pinales & Velázquez, 2014)**
 - Algoritmos con estructuras secuenciales
 - **Algoritmos con estructuras selectivas**
 - Algoritmos con estructuras repetitivas
- Estructura de un programa
- Lenguajes de programación

Estructuras selectivas

- En un algoritmo una **estructura selectiva** implica seguir o no un determinado flujo de secuencias, atendiendo al cumplimiento de cierta situación o condición (V = verdadero, *true*; F = falso, *false*)

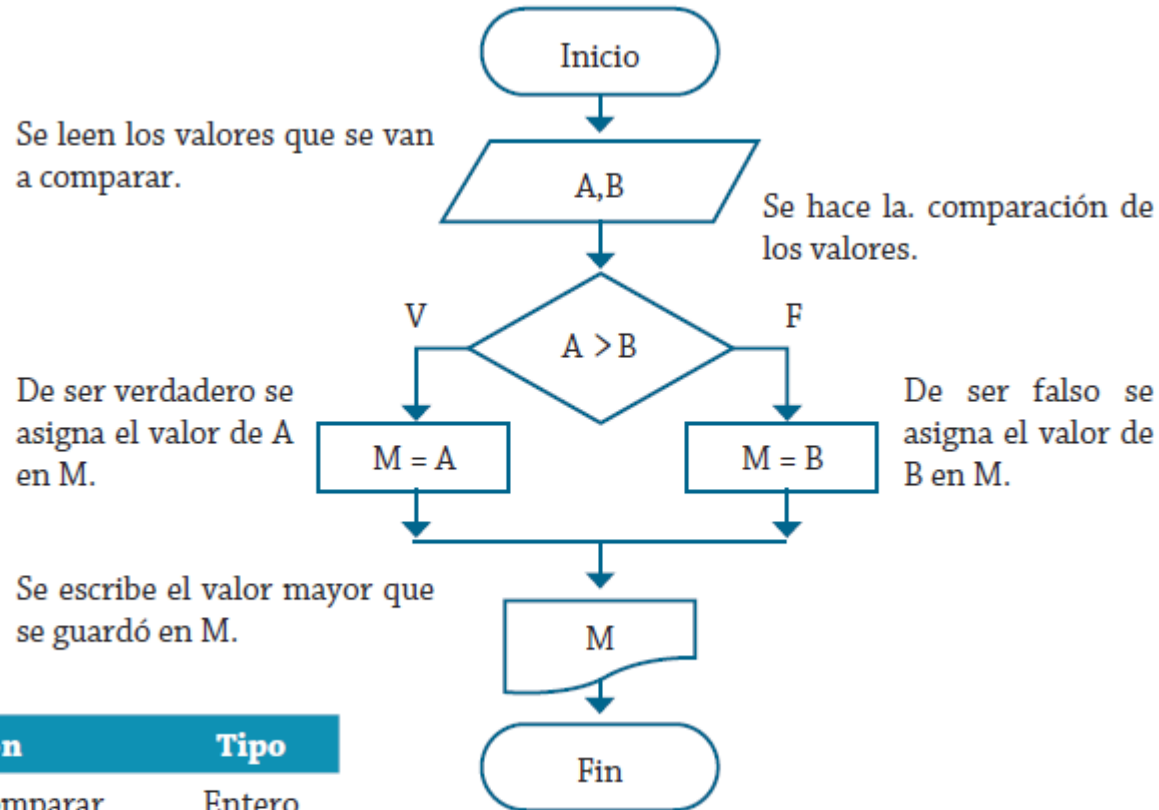


• Ej. 5: determinar el mayor de dos números enteros

Pseudocódigo

1. Inicio
2. Leer A, B
3. Si $A > B$
 - Entonces
Hacer $M = A$
 - Si no
Hacer $M = B$
 - Fin de comparación
4. Escribir "el mayor es", M
5. Fin

Diagrama de flujo



Nombre de la variable	Descripción	Tipo
A	Primer valor para comparar	Entero
B	Segundo valor para comparar	Entero
M	Resultado de la comparación	Entero

- Ej. 6: calcular el precio final y descuento de un traje, aplicando un 15% de descuento si su coste es superior a 2500\$ y 8% en otro caso

Pseudocódigo

1. Inicio
2. Leer CT
3. Si $CT > 2500$
Entonces
 Hacer $DE = CT * 0.15$
Si no
 Hacer $DE = CT * 0.08$
Fin de comparación
4. Hacer $PF = CT - DE$
5. Escribir "El precio final es", PF
6. Escribir "El descuento es" DE
7. Fin

Diagrama de flujo

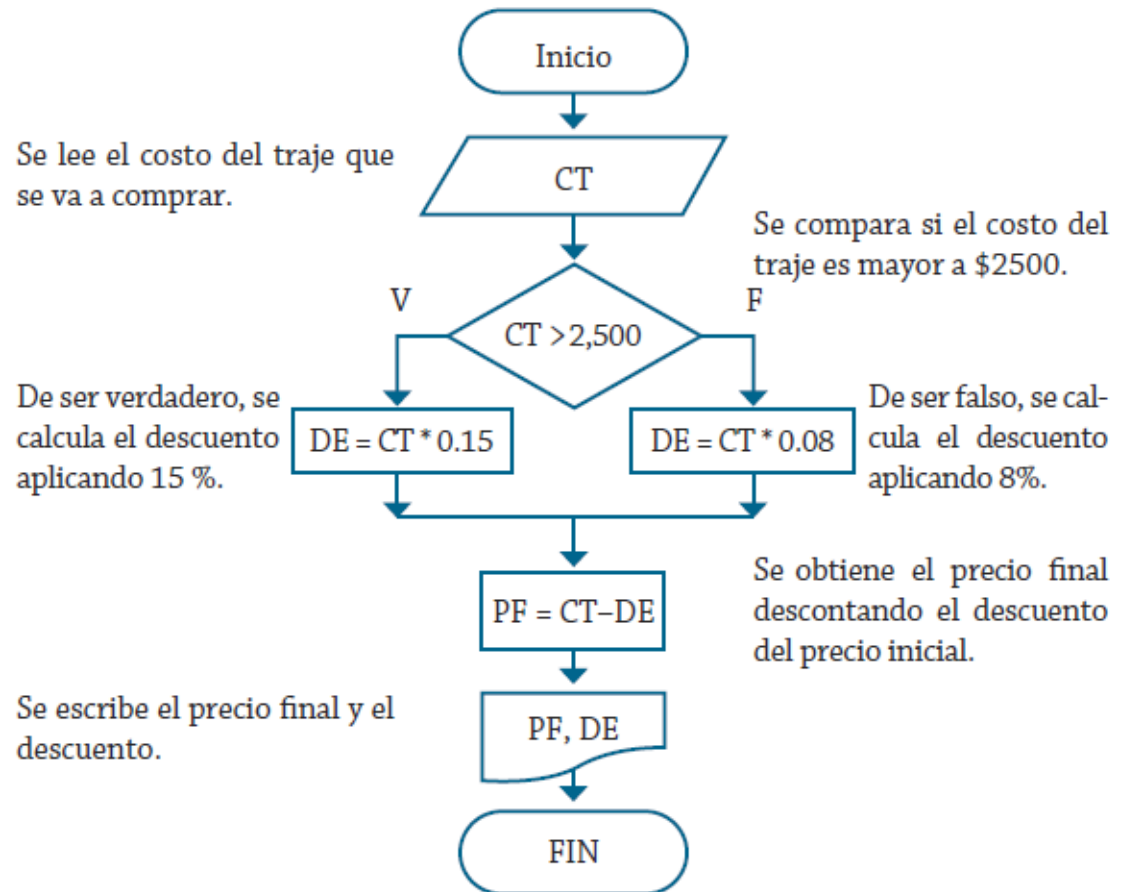
Nombre de la variable	Descripción	Tipo
CT	Costo del traje	Real
DE	Descuento que se obtendrá	Real
PF	Precio final del traje	Real

- Ej. 6: calcular el precio final y descuento de un traje, aplicando un 15% de descuento si su coste es superior a 2500\$ y 8% en otro caso

Pseudocódigo

- Inicio
- Leer CT
- Si $CT > 2500$
Entonces
Hacer $DE = CT * 0.15$
Si no
Hacer $DE = CT * 0.08$
Fin de comparación
- Hacer $PF = CT - DE$
- Escribir "El precio final es", PF
- Escribir "El descuento es" DE
- Fin

Diagrama de flujo



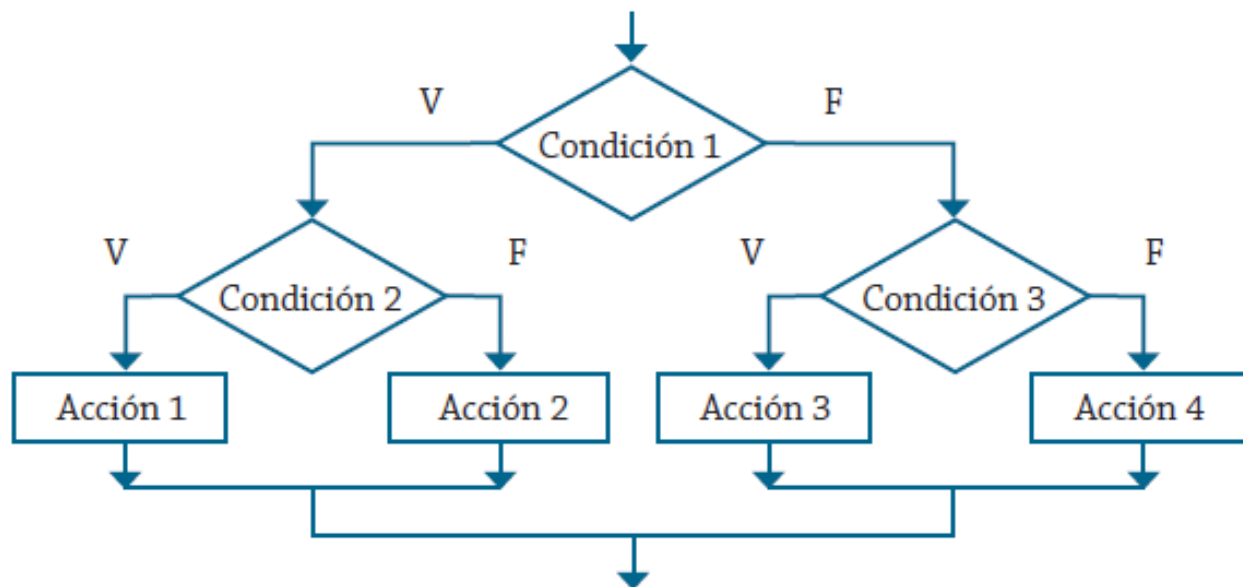
Nombre de la variable	Descripción	Tipo
CT	Costo del traje	Real
DE	Descuento que se obtendrá	Real
PF	Precio final del traje	Real

- Ejemplo de estructuras selectivas anidadas

Pseudocódigo

```
Si condición 1
  Entonces
    Si condición 2
      Entonces
        Acción 1
      Si no
        Acción 2
    Fin de comparación 2
  Si no
    Si condición 3
      Entonces
        Acción 3
      Si no
        Acción 4
    Fin de comparación 3
  Fin de comparación 1
```

Diagrama de flujo

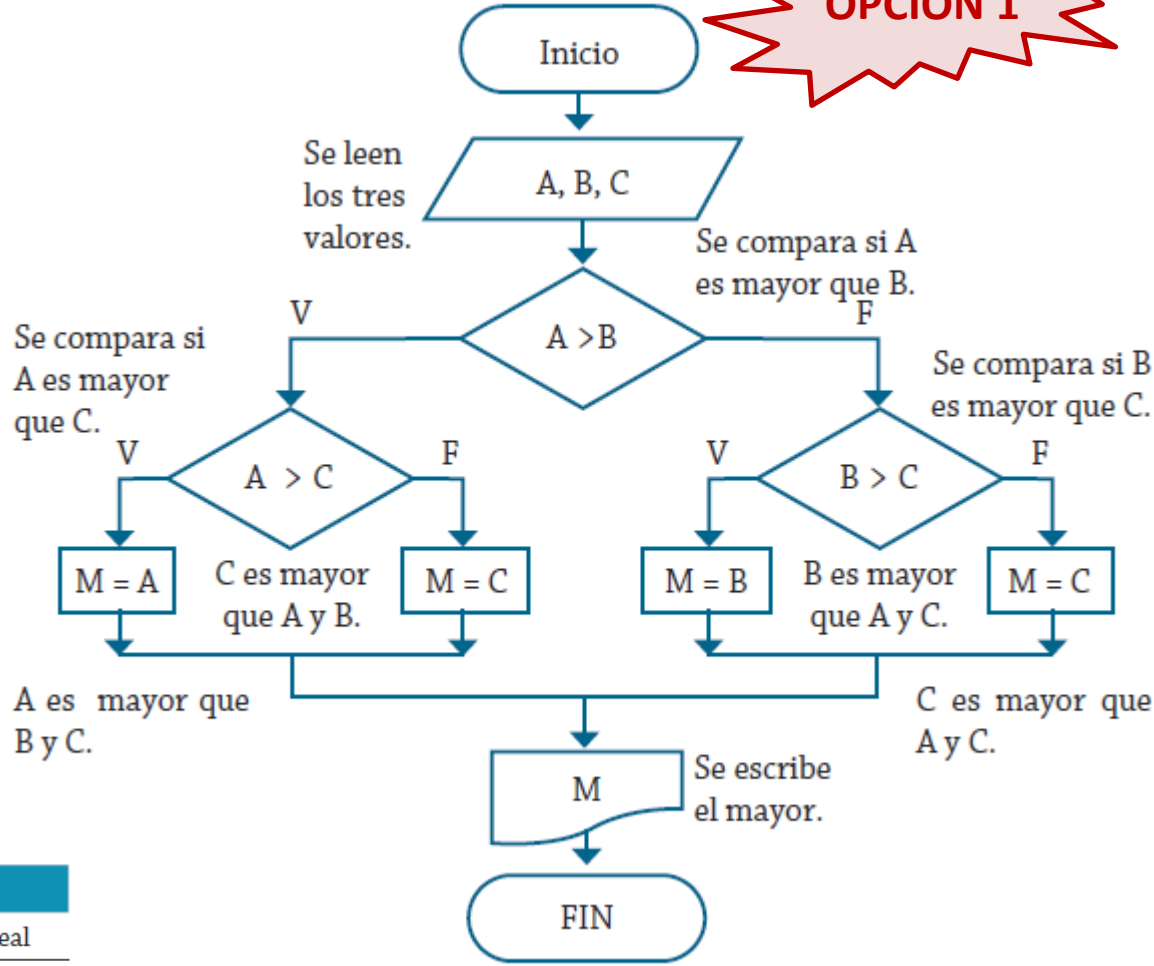


Ej. 7: determinar el mayor de tres números reales

Pseudocódigo

1. Inicio
2. Leer A, B, C
3. Si $A > B$
 - Entonces
 - Si $A > C$
 - Entonces $M = A$
 - Si no $M = C$
 - Fin compara
 - Si no
 - Si $B > C$
 - Entonces $M = B$
 - Si no $M = C$
 - Fin compara
4. Escribir "El mayor es", M
5. Fin

Diagrama de flujo



Nombre de la variable	Descripción	Tipo
A	Primer valor	Entero o real
B	Segundo valor	Entero o real
C	Tercer valor	Entero o real
M	Valor mayor	Entero o real

- Ej. 7: determinar el mayor de tres números reales

Pseudocódigo

1. Inicio
2. Leer A, B, C
3. Si $A > B$
 - Entonces
 - $M = A$
 - Si no
 - $M = B$Fin compara
4. Si $C > M$
 - Entonces
 - $M = C$Fin compara
5. Escribir "El mayor es", M
6. Fin

Diagrama de flujo



OPCIÓN 2

Nombre de la variable	Descripción	Tipo
-----------------------	-------------	------

A	Primer valor	Entero o real
---	--------------	---------------

B	Segundo valor	Entero o real
---	---------------	---------------

C	Tercer valor	Entero o real
---	--------------	---------------

M	Valor mayor	Entero o real
---	-------------	---------------

Programación I

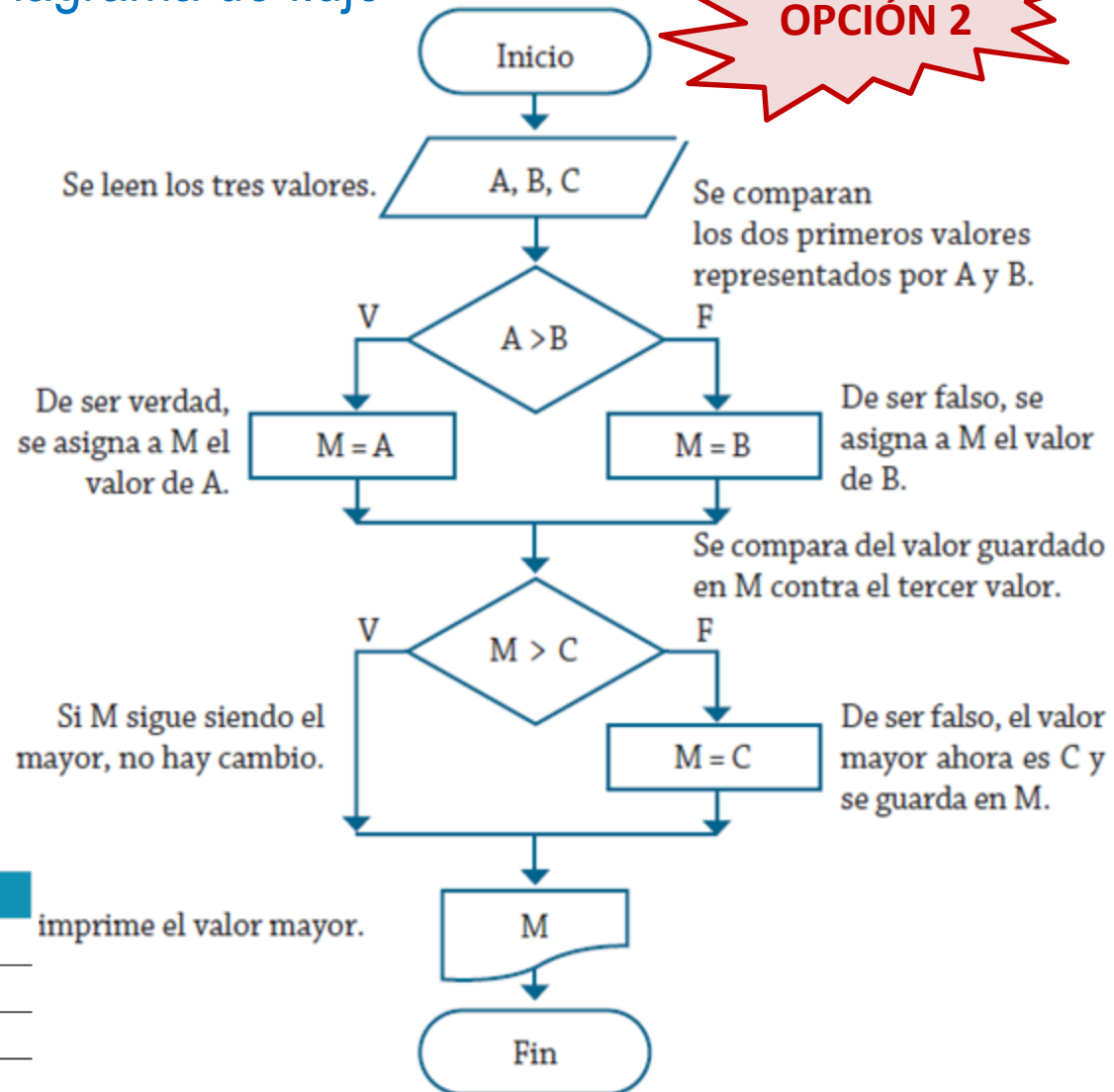
Escuela Politécnica Superior
Universidad Autónoma de Madrid

• Ej. 7: determinar el mayor de tres números reales

Pseudocódigo

- Inicio
- Leer A, B, C
- Si $A > B$
Entonces
 $M = A$
Si no
 $M = B$
Fin compara
- Si $C > M$
Entonces
 $M = C$
Fin compara
- Escribir "El mayor es", M
- Fin

Diagrama de flujo

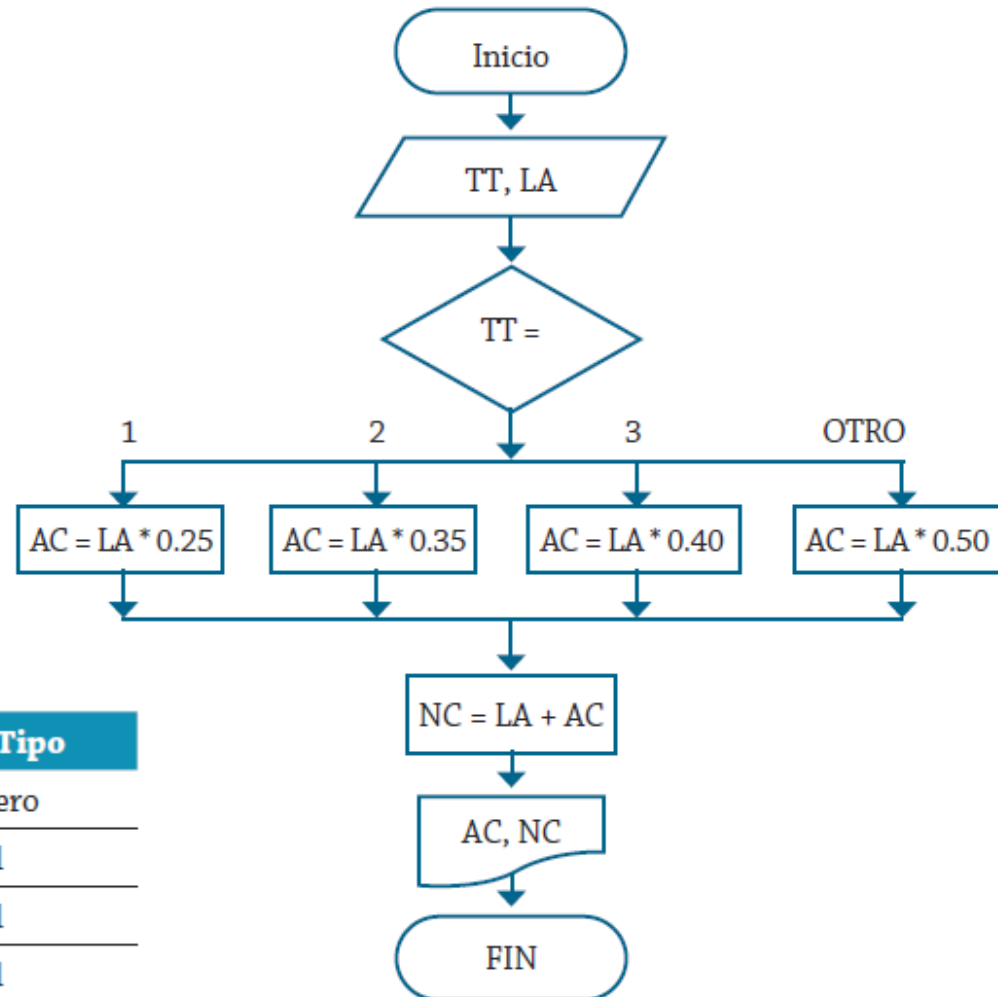


Nombre de la variable	Descripción	Tipo
A	Primer valor	Entero o real
B	Segundo valor	Entero o real
C	Tercer valor	Entero o real
M	Valor mayor	Entero o real

Estructuras selectivas con múltiples opciones

- Ejemplo de **estructuras selectiva con múltiples opciones** (en vez de sólo dos, V y F): *aumento del límite de crédito de una tarjeta bancaria en función de su tipo: 1, 2, 3 u otro*

- Inicio
- Leer TT, LA
 - Si NZ Igual a
 - 1: Hacer $AC = LA * 0.25$
 - 2: Hacer $AC = LA * 0.35$
 - 3: Hacer $AC = LA * 0.40$
 - Si no
 - Hacer $AC = LA * 0.50$
- Fin de comparación
- Hacer $NC = LA + AC$
- Escribir "El aumento de crédito", AC
- Escribir "Nuevo límite de crédito", NC
- Fin



Nombre de la variable	Descripción	Tipo
TT	Tipo de tarjeta	Entero
LA	Límite actual de crédito	Real
AC	Aumento de crédito	Real
NC	Nuevo límite de crédito	Real

- Algoritmos
- **Ejemplos de algoritmos (Pinales & Velázquez, 2014)**
 - Algoritmos con estructuras secuenciales
 - Algoritmos con estructuras selectivas
 - **Algoritmos con estructuras repetitivas**
- Estructura de un programa
- Lenguajes de programación

Estructuras repetitivas

- En un algoritmo una **estructura repetitiva** o **bucle** implica ejecutar una serie de secuencias de forma cíclica, atendiendo al cumplimiento de cierta situación o condición

A. Pseudocódigo

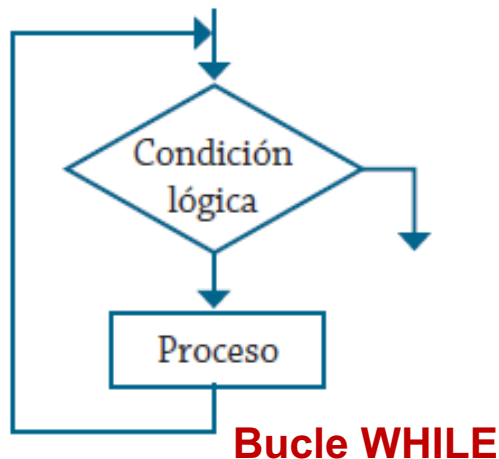
Mientras Condición lógica
Proceso
Fin Mientras

Repite
Proceso
Hasta Condición lógica

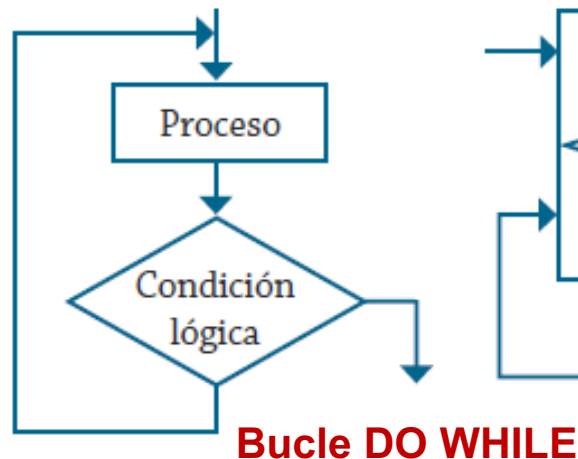
Desde valor inicial **Hasta** valor Final
Proceso
Fin Desde

B. Diagrama de flujo

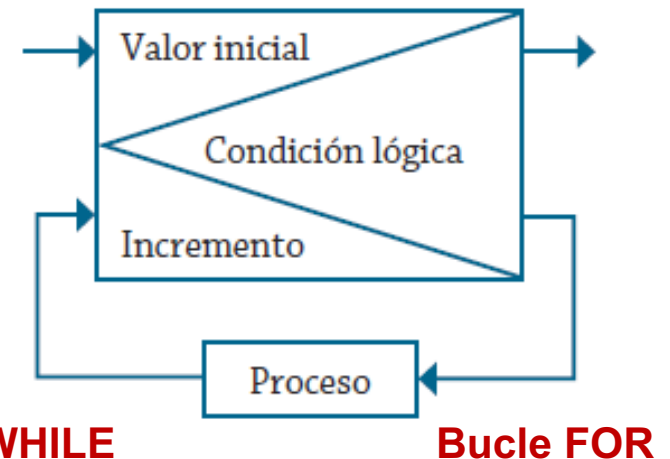
1. Mientras que



2. Hasta que



3. Desde, hasta que

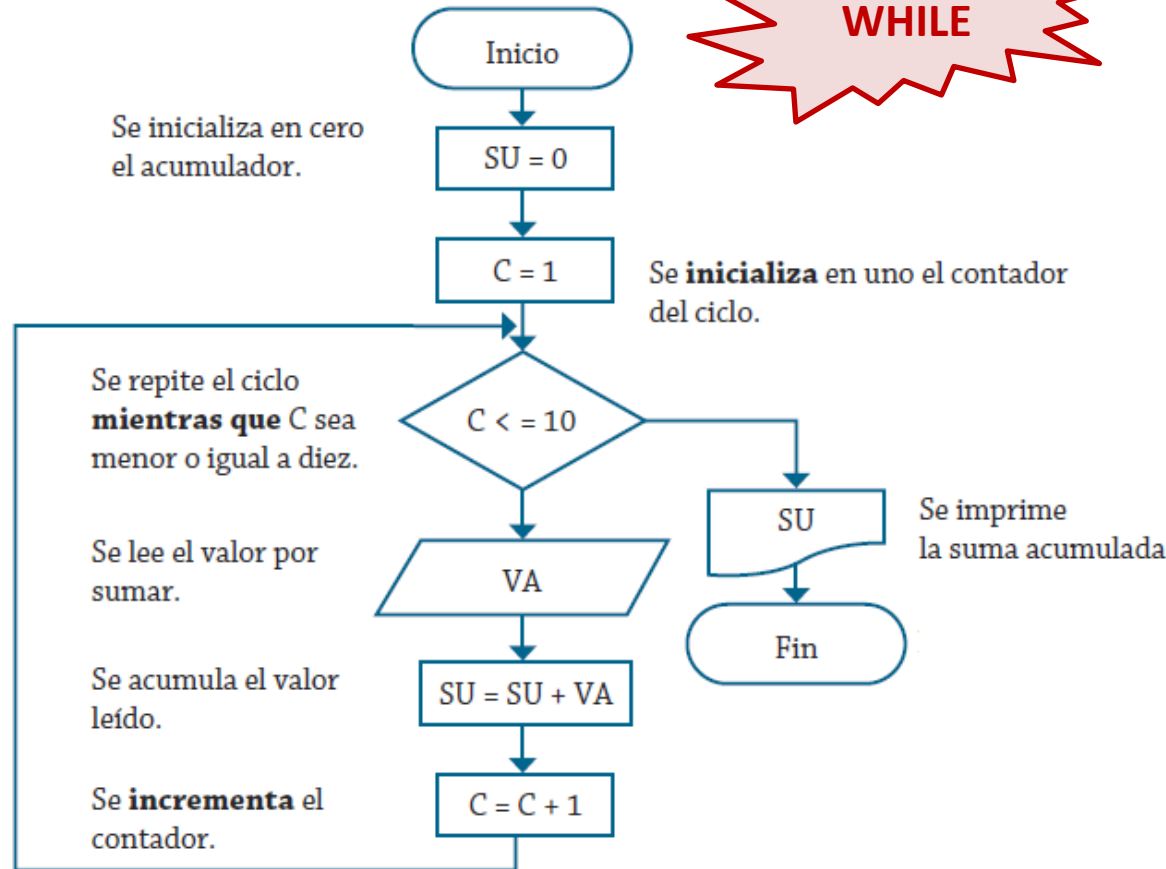


- Ej. 8: obtener el valor de 10 cantidades y calcular su suma

Pseudocódigo

- Inicio
- Hacer $SU = 0$
- Hacer $C = 1$
- Mientras $C \leq 10$
 - Leer VA
 - Hacer $SU = SU + VA$
 - Hacer $C = C + 1$Fin mientras
- Escribir SU
- Fin

Diagrama de flujo



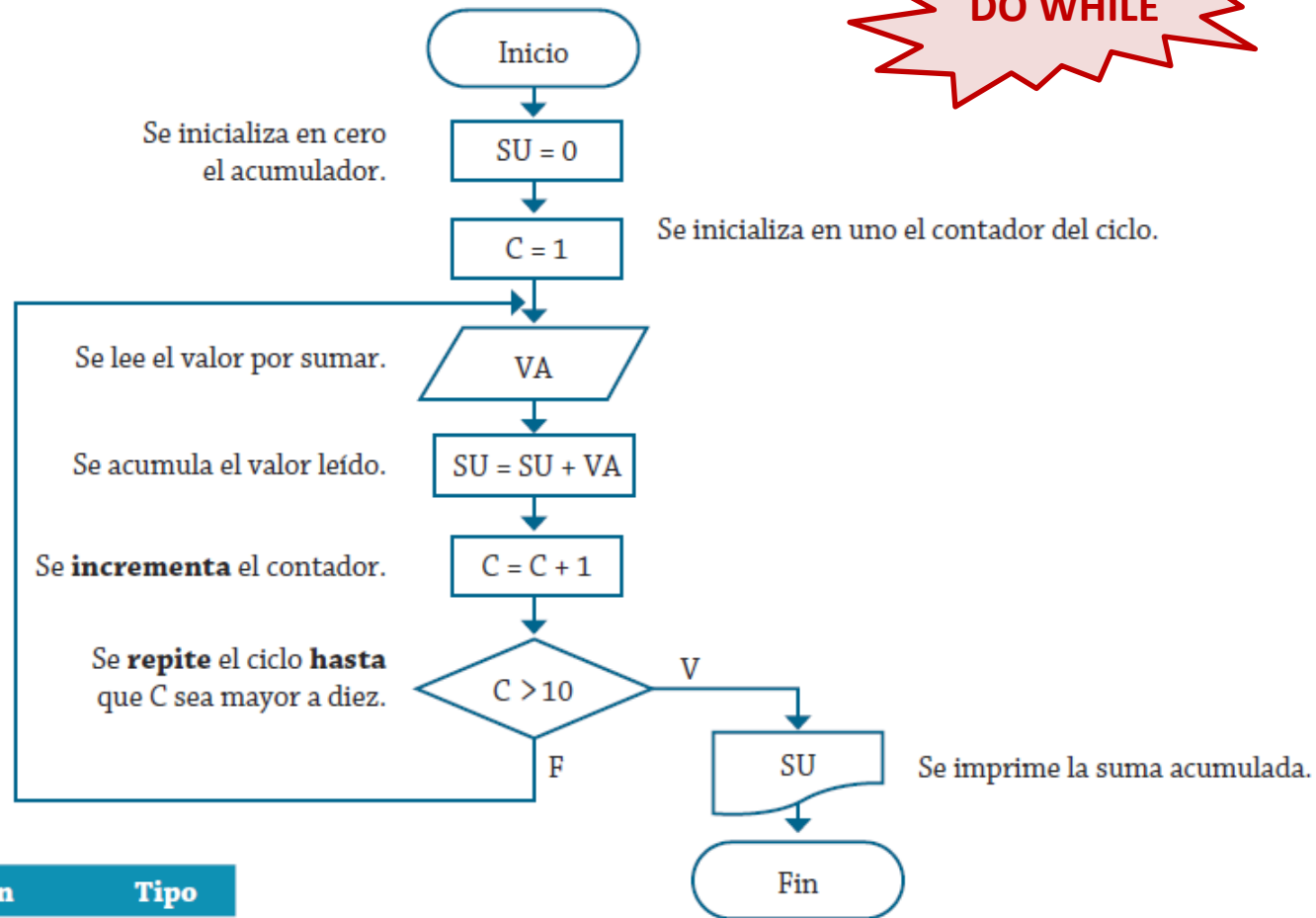
Nombre de la variable	Descripción	Tipo
C	Contador	Entero
VA	Valor por sumar	Real
SU	Suma de los valores	Real

- Ej. 8: obtener el valor de 10 cantidades y calcular su suma

Pseudocódigo

- Inicio
- Hacer $SU = 0$
- Hacer $C = 1$
- Repita
 - Leer VA
 - Hacer $SU = SU + VA$
 - Hacer $C = C + 1$Hasta que $C > 10$
- Escribir SU
- Fin

Diagrama de flujo



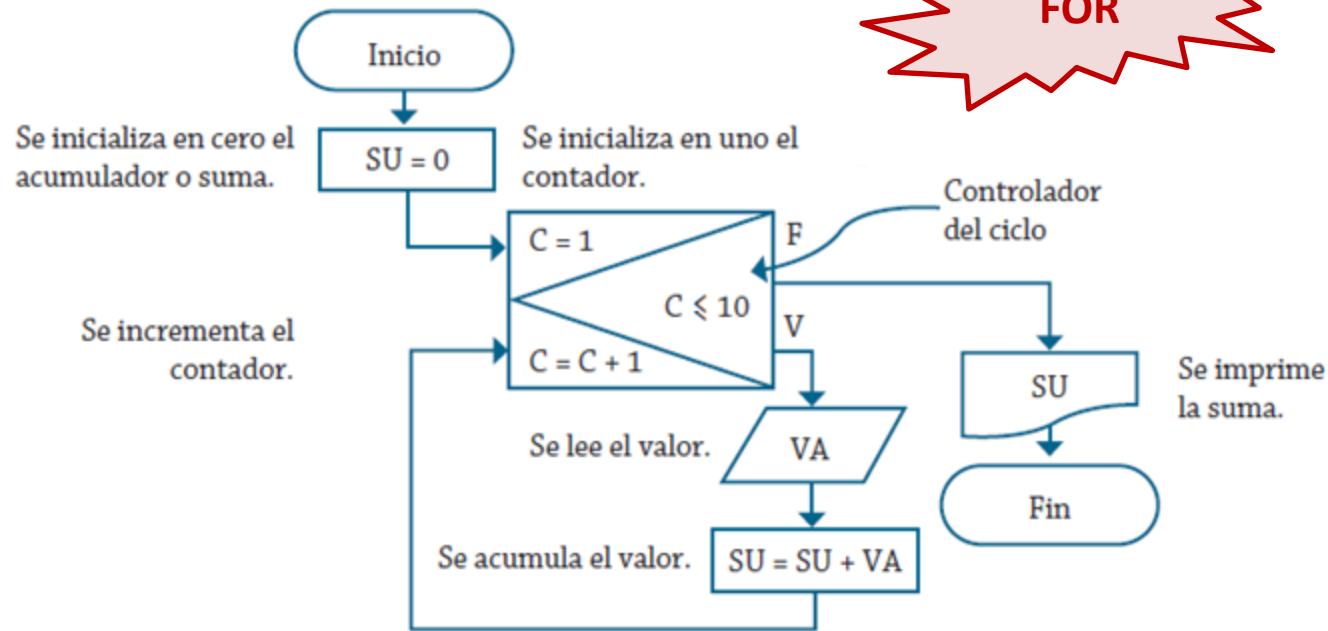
Nombre de la variable	Descripción	Tipo
C	Contador	Entero
VA	Valor por sumar	Real
SU	Suma de los valores	Real

- Ej. 8: obtener el valor de 10 cantidades y calcular su suma

Pseudocódigo

- Inicio
- Hacer $SU = 0$
- Desde $C = 1$ hasta $C = 10$
 - Leer VA
 - Hacer $SU = SU + VA$
 - Fin desde
- Escribir SU
- Fin

Diagrama de flujo



Nombre de la variable	Descripción	Tipo
C	Contador	Entero
VA	Valor por sumar	Real
SU	Suma de los valores	Real

- Algoritmos
- Ejemplos de algoritmos
- **Estructura de un programa**
 - **Programa = Algoritmo + Estructuras de datos**
 - **Ejecución de un programa en el ordenador**
- Lenguajes de programación

- **Variable**

- Se emplea para representar un **dato** requerido por un algoritmo
- En la arquitectura de un ordenador, (de forma lógica) se corresponde con una “**celda**” de memoria de un tamaño en Bytes fijo que se emplea para almacenar el dato
- **Asignación**: operación por la cual se guarda un dato en una variable
 - Una variable puede almacenar un único (conjunto de) dato en un momento dado
 - Si se asigna un dato sobre una variable que tuviera algún valor anterior, este último se pierdeEjemplo: $x \leftarrow 3$; $x \leftarrow 5$; $x \leftarrow 2$; El valor final de x es 2

- **Estructura de datos**

- Es la forma de organizar y almacenar los datos para facilitar su acceso y modificación
 - Ejemplo: la estructura de datos “Alumno” puede estar compuesta de los atributos “nombre”, “apellidos”, “edad”, “curso”, etc., que serán variables que almacenen los datos del alumno

- Programa (informático, de ordenador)

- Conjunto de instrucciones, escritas en un *lenguaje de programación* dado, que hacen al ordenador ejecutar un plan mental (**algoritmo**) para resolver un problema dado, haciendo uso de unas **estructuras de datos** determinadas
- Esta definición queda reflejada en la expresión dada por N. Wirth en 1980:

Programa = Algoritmo + Estructuras de datos

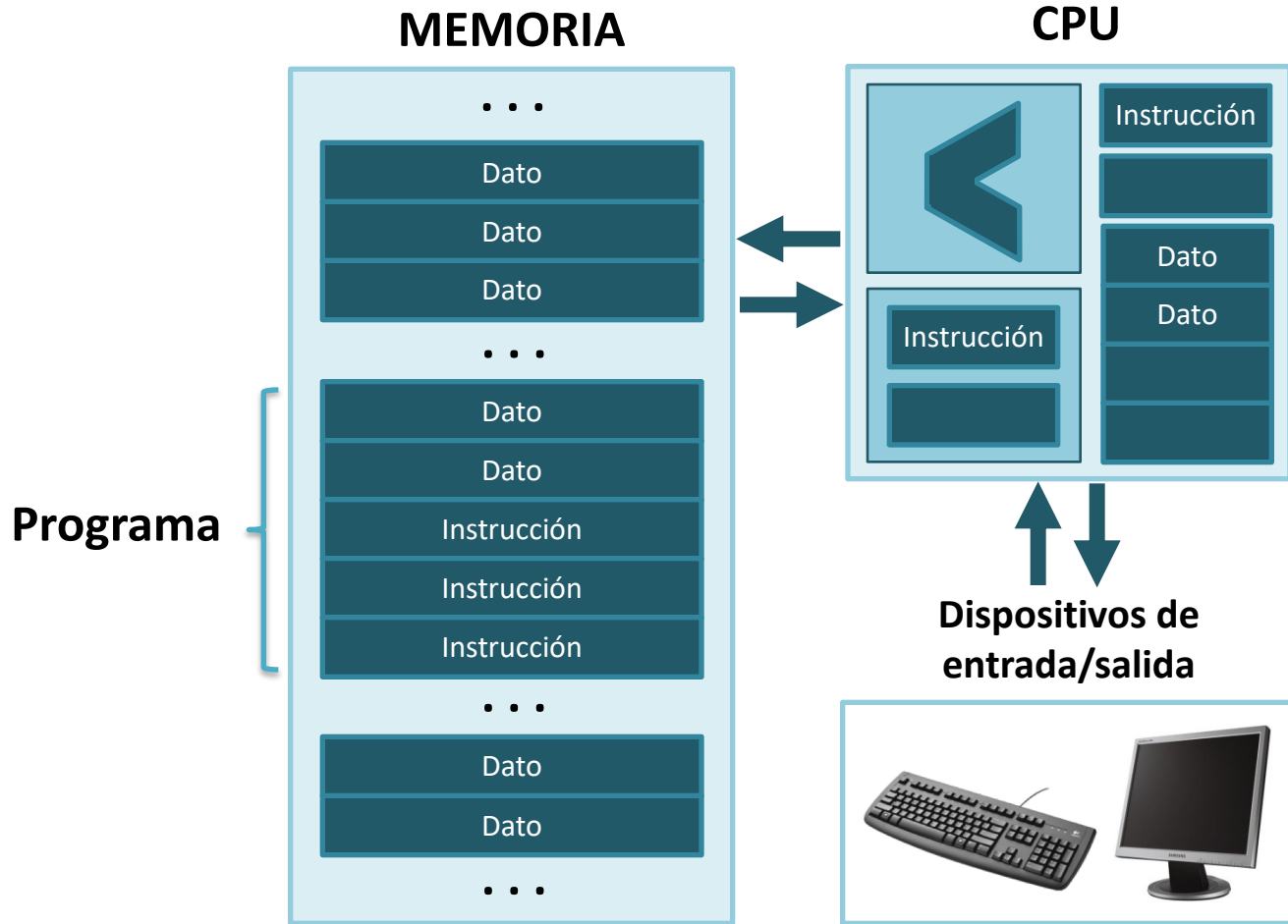
- Ejemplo de traducción de un pseudocódigo a lenguaje de programación C

```
int sumar(int lista[], int longitud) {
    int suma = 0, i = 1;
    while ( i <= longitud ) {
        suma = suma + lista[i-1];
        i++;
    }
    return suma;
}
```



Entrada: entero LISTA[]
Salida: entero SUMA
Instrucciones:
SUMA \leftarrow 0
 $i \leftarrow 1$
Mientras ($i \leq \text{longitud}(\text{LISTA})$) hacer:
SUMA \leftarrow SUMA + LISTA[i]
 $i \leftarrow i + 1$
Devolver SUMA

- Arquitectura Von Neumann



- Algoritmos
- Ejemplos de algoritmos
- Estructura de un programa
- **Lenguajes de programación**
 - Lenguajes máquina
 - Lenguajes ensamblador
 - Lenguajes de alto nivel
 - **Compiladores e intérpretes**

Lenguajes máquina

- En un ordenador, las **instrucciones** y los **datos** se almacenan, procesan y transfieren en formato binario (en base 2)
 - La unidad básica de información es el **bit**, que toma valores 0 ó 1
 - Un **Byte** está formado por 8 bits
- El **lenguaje máquina** está compuesto por un conjunto fijo de instrucciones en binario
 - Es el lenguaje de programación de más “bajo nivel”
 - Es interpretable por el procesador
 - No es inteligible para un ser humano
 - Depende del procesador

Programa que imprime por pantalla el mensaje “Hola mundo!”, escrito en lenguaje máquina (inventado)

```
00101101010111000100011000100
10101110101010101010101110010
10101010001101001111000111001
10101010110101010110100011100
10101000111100010101010111010
10101011010111000110011000010
10001010100101001010100011010
10101101101010001010101010011
01001000111000101110101010101
```

- Un **lenguaje ensamblador** es un lenguaje de programación de bajo nivel que constituye la representación más directa al “código máquina” que es legible por un programador
 - Tiene que ser transformado a lenguaje máquina para que sea procesable por el ordenador
 - Es inteligible para una persona (que conozca el lenguaje)
 - Depende del procesador

Programa que imprime por pantalla el mensaje “Hola mundo!”, escrito en lenguaje ensamblador para la arquitectura de procesador x86

```
.model small
.stack
.data
mensaje DB 'Hola Mundo!$'
.code
programa:
    mov ax, @data
    mov ds, ax
    mov dx, offset mensaje
    mov ah, 9
    int 21h
end programa
```

- Un **lenguaje de alto nivel** constituye el nivel de mayor abstracción para un programador, pues su representación es la más alejada al lenguaje máquina y la más cercana al “lenguaje natural”
 - En algún punto tiene que ser transformado a lenguaje ensamblador para que sea procesable por el ordenador
 - Es inteligible para una persona (que conozca el lenguaje)
 - No depende del procesador
- Ejemplos de lenguajes de alto nivel:
 - C, C++, C#, Java, Pascal, Basic, Visual Basic, Fortran, COBOL, LISP, Prolog, PHP, Python, Ruby, etc.

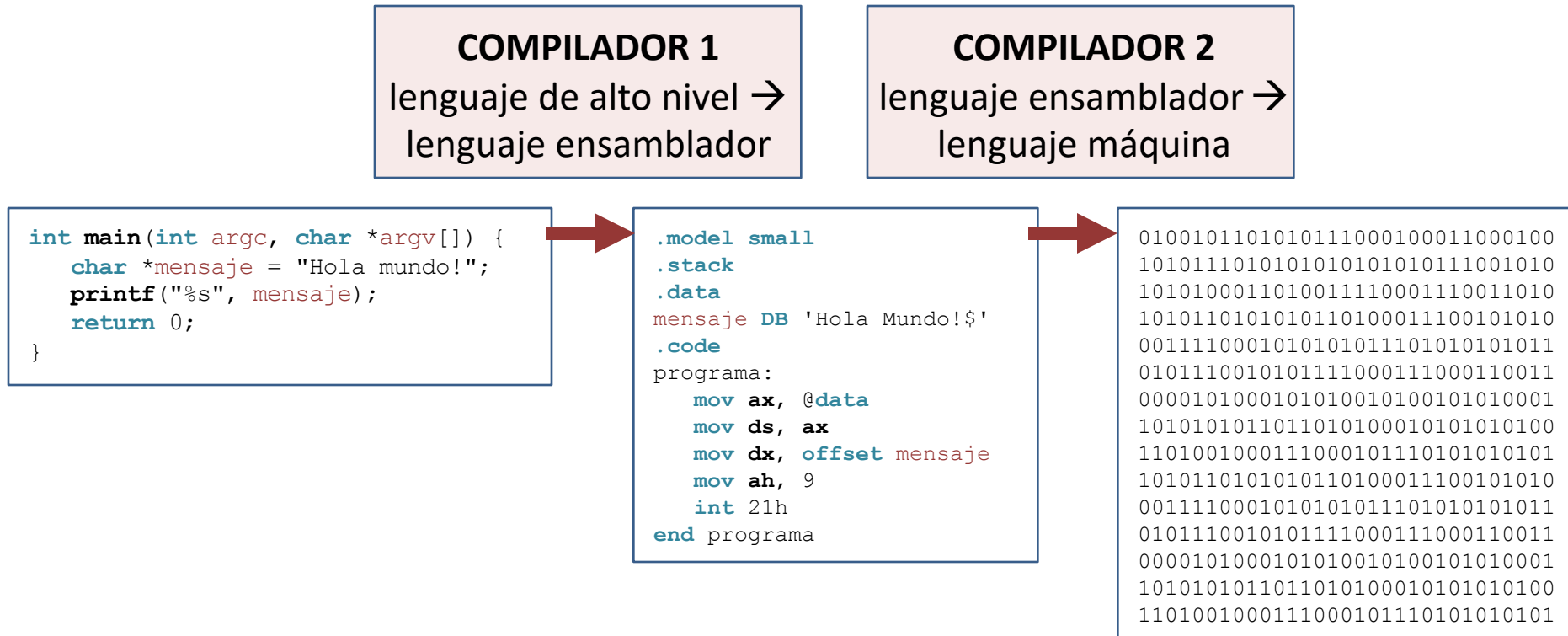
Programa que imprime por pantalla el mensaje “Hola mundo!”, escrito en lenguaje C

```
int main(int argc, char *argv[]) {  
    char *mensaje = "Hola mundo!";  
    printf("%s", mensaje);  
    return 0;  
}
```

- Un **compilador** es un PROGRAMA que traduce un PROGRAMA escrito en un lenguaje de programación a otro lenguaje de programación, generando un PROGRAMA equivalente que el ordenador es capaz de interpretar
 - Y entonces, un compilador:
 - ¿en qué lenguaje de programación está escrito?
 - ¿ha tenido que ser traducido por otro compilador?



- Un **compilador** es un PROGRAMA que traduce un PROGRAMA escrito en un lenguaje de programación a otro lenguaje de programación, generando un PROGRAMA equivalente que el ordenador es capaz de interpretar



- Un **intérprete** es un PROGRAMA capaz de analizar y ejecutar otro PROGRAMA, escrito en un lenguaje de alto nivel
 - A diferencia de un compilador, que transforma un programa de un lenguaje de programación a otro lenguaje de programación para que todas sus instrucciones se ejecuten en bloque por el ordenador, un intérprete hace la **transformación y ejecución instrucción a instrucción**

Programar en ordenador: ¿fácil o difícil?

Programar en ordenador: ¿fácil o difícil?

