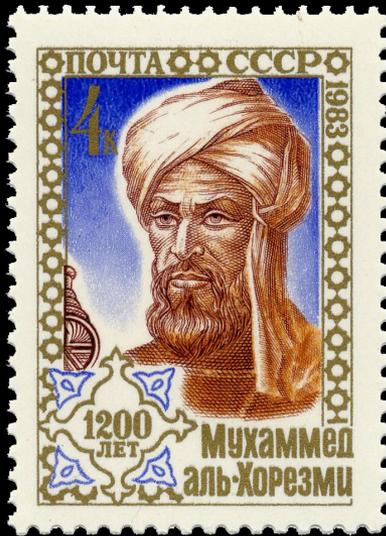


# Diagramación Lógica

Algoritmos y Representaciones



# ¿Qué es un Algoritmo?



En el S. XII (aprox.) el matemático persa *Abu Abdallah Muḥammad ibn Mūsā al-Jwārizmī*, más conocido como **Al-Juarismi**, desarrolla el concepto de un proceso escrito a seguir para alcanzar un objetivo.

La palabra *Algoritmo* deriva de este matemático persa que además es considerado el padre del álgebra.

# ¿Qué es un Algoritmo?

- Es un Método para resolver cierto tipo de problema mediante una secuencia de pasos precisos, definidos y finitos
- Es una serie de operaciones detalladas y consecutivas
- Es un conjunto de reglas para resolver una cierta clase de problemas
  - Se puede formular de muchas formas
  - Cuidando de que no exista ambigüedad

# ¿Qué es un Algoritmo?

- Es un procedimiento detallado para resolver un problema en una **secuencia de pasos consecutivos** y en un **tiempo finito**.
- Lleva desde un **estado inicial** a un **estado final**.
- Recibe una **Entrada** y entrega una **Salida**.
- Se especifica en base a operaciones elementales que modifican las variables internas y controlan el flujo de ejecución.

# Características de un Algoritmo

- **Preciso:** debe indicar el orden de realización en cada paso y estos no pueden tener ambigüedad.
- **Definido:** si se sigue dos veces con iguales datos, obtiene el mismo resultado cada vez.
- **Finito:** tiene fin; un número determinado de pasos.
- **Sencillo:** Debe ser legible.
- **Modular:** Particionable.

# Características de un Algoritmo

- **Eficiente:** Usa mínimos recursos.
- **Efectivo:** Alcanza las metas.
- **Correcto:** Sin error.
- **Tiempo finito:** Se ha de desarrollar en el menor tiempo posible.
- Todo Algoritmo debe tener cero ó mas entradas.
- Debe tener al menos una salida y ésta debe ser tangible.

# ¿Cómo se describe un algoritmo?

Lenguaje natural, coloquial.

Diagramas (de Flujo o Estructogramas).

Pseudocódigo.

Lenguaje de programación.

La precisión es importante. Un algoritmo no puede ser descrito de forma ambigua:

*Todos tienen que entender lo mismo (¡incluida la computadora!).*

# ¿Cómo se desarrolla un algoritmo?

- Con imaginación, hay que ser creativo.
- No reinventar la rueda.
- Dividir para reinar.
- Para ser efectivo se requiere practicar constantemente.
- Pueden existir varios algoritmos para resolver el mismo problema:
  - Depende de quién lo invente.
  - Depende de qué se quiere optimizar.
    - Ej: tiempo versus costo

# Algoritmo en Programación

*"Conjunto finito de instrucciones que ejecutadas en determinado orden permiten resolver una tarea dada."*

## Debe satisfacer:

- Que haya datos provistos externamente (Entradas).
- Debe producir al menos una información (Salida).
- Cada instrucción debe ser clara y sin ambigüedades.
- Para cualquier caso particular de los datos, el algoritmo debe tener un número finito de pasos y un tiempo finito de ejecución.
- Toda instrucción debe ser lo suficientemente elemental como para que una persona pueda llevarla a cabo con lápiz y papel.

# ¿Qué es una Instrucción?

Es un conjunto de símbolos que representa una orden para el ordenador: *la ejecución de una operación sobre los datos.*

Las instrucciones para ejecutar un algoritmo se escriben en un lenguaje de programación.

Se forman con símbolos tomados de un determinado repertorio o conjunto componentes léxicos

Se construyen siguiendo unas reglas precisas sintaxis.

# Datos y Variables

**Datos:** objetos simbólicos que representan objetos del mundo real.

Ej: una fecha determinada (10 de Marzo de 2003), Pi (3.14159).

**Variables:** Una variable es un nombre asociado a un elemento de datos que contiene un valor y que está ubicado en memoria.

Es un identificador y una referencia a una zona de memoria que contiene un valor. Este valor puede cambiar durante la ejecución del programa. No hace referencia a un objeto explícito.

Ej: velocidad del móvil, factor de crecimiento.

# Una metodología para crear algoritmos

## 1. Definición del problema

Conceptualización, Objetivo, Elementos involucrados

## 2. Conceptualización de la solución

Descomposición, Tareas, Variables

## 3. Especificación del algoritmo

Especificar la secuencia de actividades, Dependiendo de la complejidad del algoritmo subalgoritmos

## 4. Validación del algoritmo

Dominios, Ejecución

## 5. Limitaciones del algoritmo

Identificación de puntos débiles (condiciones críticas)

# Un ejemplo...

**Problema:** Hacer una taza de té a la inglesa.

Solución:

**Entrada:** taza, tetera, cuchara, saquito de té, agua y leche.

**Salida:** una taza de té a la inglesa.

Proceso:

1. Tomar la taza.
2. Colocar el saquito de té en la taza.
3. Colocar agua en la tetera.
4. Poner a calentar el agua hasta que hierva.
5. Verter el agua caliente en la taza.
6. Añadir leche a gusto.
7. Disfrutar.

# Otro ejemplo...

Problema: Sumar dos números.

Solución:

Entrada: dos números.

Salida: el resultado de la suma en memoria y la impresión del mismo por pantalla.

Proceso:

1. Leer el 1° número.
2. Leer el 2° número
3. Realizar la adición entre ambos números.
4. Guardar el resultado en la memoria.
5. Imprimir el resultado por pantalla.

# Uno más...

**Problema:** Aprobar o rechazar un pedido a fábrica en base a la solvencia del cliente.

Solución:

**Entrada:** el pedido.

**Salida:** mensaje de aceptación o rechazo.

Proceso:

1. Leer el pedido.
2. Examinar la ficha del cliente.
3. Si el cliente es solvente, aceptar el pedido. Sino rechazarlo.
4. Emitir el mensaje.

# más ejemplos de algoritmos.

- Los pasos para grabar un número telefónico en tu celular.
- El procedimiento para obtener tu pasaporte.
- Los pasos para invitar a alguien al cine.
- El procedimiento que sigues para inscribirte a una materia.
- Las instrucciones que te dan para resolver un examen.
- La receta que sigues para preparar una torta.
- Los pasos que sigues para prender el carbón para un asado.

# Diagramas de Flujo

- Es la representación gráfica de un algoritmo.
- Representa el flujo de ejecución mediante flechas y conectores que van desde el punto de inicio y el de final del proceso.
- Se utilizan símbolos precisos con significados definidos que representan cada uno de los pasos del algoritmo.
- Deben seguirse ciertas reglas para la creación de los diagramas.

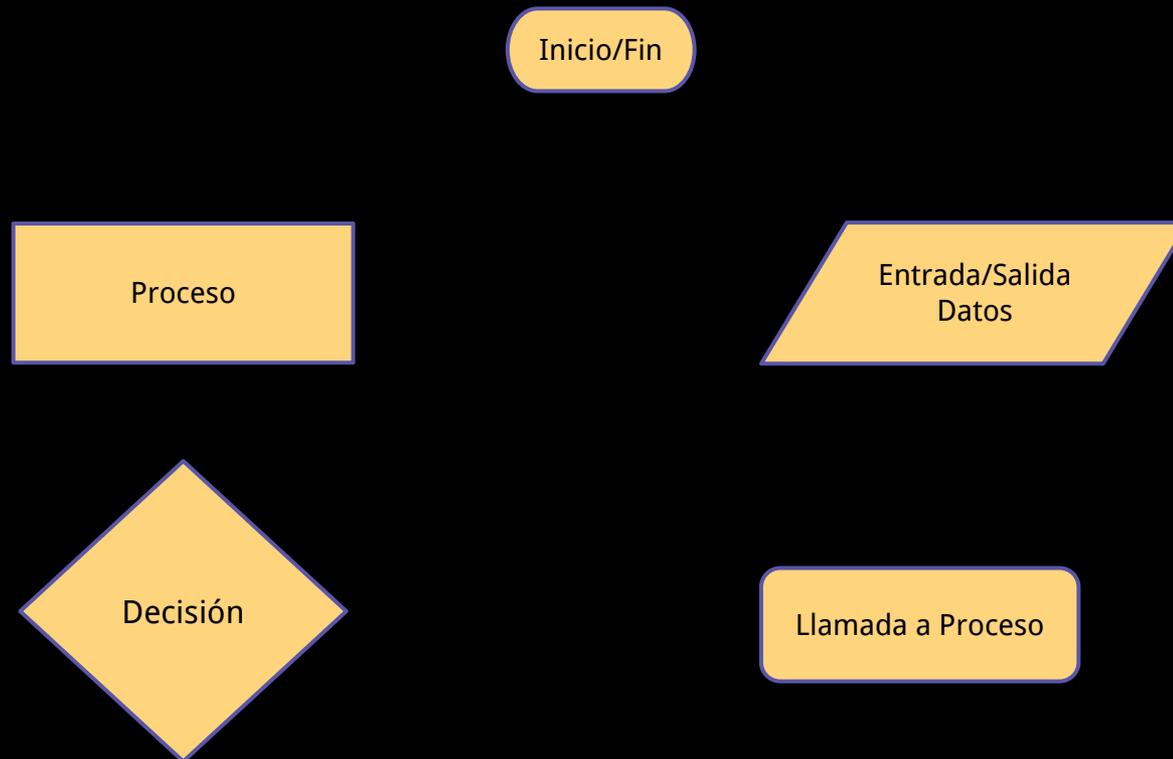
# Diagramas de Flujo

## Reglas:

- Los diagramas se realizan en el sentido de lectura (de arriba hacia abajo o de izquierda a derecha).
- Los símbolos se unen con flechas que indican el flujo de ejecución.
- Las flechas deben ser horizontales o verticales, nunca diagonales.
- Hay que evitar cruzar líneas, para eso puede separarse el flujo del diagrama usando conectores especiales. Esto sólo debe hacerse cuando es estrictamente necesario.
- No deben quedar líneas de flujo sin conectar, ni símbolos aislados.
- Todo texto escrito dentro de los símbolos debe ser legible y preciso, y en lo posible sintético.

# Diagramas de Flujo

Veamos los símbolos que deben utilizarse:



# Diagramas de Flujo

Ej: Leer un número e imprimir por pantalla el siguiente.

Entrada:

- número (variable numérica)
- siguiente (variable numérica)

Salida:

- siguiente (guardado y por pantalla)

