



**Instituto Politécnico Nacional**  
La Técnica al Servicio de la Patria

**Unidad Profesional Interdisciplinaria de  
Ingeniería Campus Tlaxcala UPIIT**

# Fundamentos de Programación

Esaú Eliezer Escobar Juárez

Ingeniería en Inteligencia Artificial (IIA)

---

# ¿Qué es programar?

- Decirle a un tonto muy rápido **exactamente** lo que tiene que hacer.
- Especificar la estructura y el comportamiento de un programa, probar que el programa realiza su tarea con rendimiento aceptable.
- Programa: Transforma entrada en salida





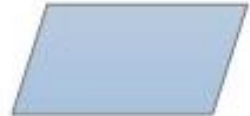


- Algoritmo: Secuencia de pasos y operaciones que se deben realizar para resolver un problema.

# Algoritmo

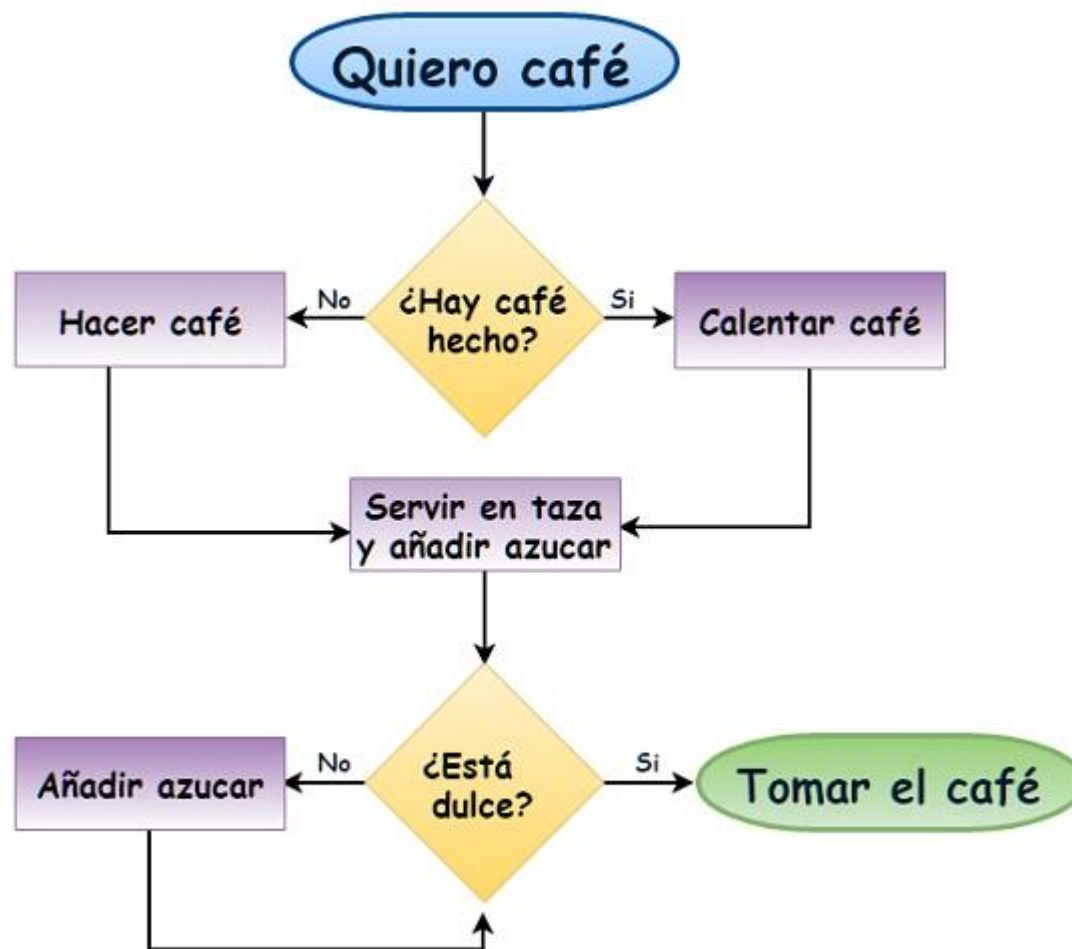
- Término utilizado en matemáticas, lógica, ciencias de la computación y disciplinas relacionadas
- Conjunto de instrucciones que permiten solucionar un problema.
- Características
  - Finito: tienen inicio y fin.
  - No-ambiguos: son concretos y claros.
  - Ordenado: van en secuencia.
  - Abstracto: son un modelo.
- [https://www.ted.com/talks/kevin\\_slavin\\_how\\_algorithms\\_shape\\_our\\_world?language=es](https://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world?language=es)

# Expresar un algoritmo: Diagramas de flujo

- Es una representación gráfica de un algoritmo o proceso

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Linea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

# Ejemplo



- [Diagrama de Flujo de la amistad Sheldon Cooper](#)

# Expresar un algoritmo: Pseudocódigo

- Pseudocódigo: *falso lenguaje*, el prefijo *pseudo* significa *falso*
- Es una descripción de alto nivel de un algoritmo que emplea una mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación.
- Este método es mas compacto, mas fácil de escribir y mas fácil de transcribir a un lenguaje de programación que el diagrama de flujo.

# Ejemplo

**ALGORITMO** Sumar

**INICIO**

**ESCRIBIR**("Dime dos números para sumar: ");

**LEER**(Numero1, Numero2);

Resultado <- Numero1 + Numero2;

**ESCRIBIR**("La suma es: ", Resultado);

**FIN**

- [PSeInt](#)
- <http://pseint.sourceforge.net/index.php?page=descargas.php>

# *El lenguaje C*

- Lenguaje creado por Dennis M. Ritchie en 1972
- Lenguaje de nivel medio:
  - — Estructuras típicas de los lenguajes de alto nivel
  - — Construcciones para control a nivel de máquina
- Lenguaje sencillo (pocas palabras reservadas)
- Componente estructural básico: la función (subprograma)
- Programación modular
- Distingue entre mayúsculas y minúsculas
- Palabras reservadas (o clave): en minúsculas



# Primer programa

Programa que muestra un saludo en la pantalla

```
#include<stdio.h>

int main (void)
{
    //Imprimir en pantalla "Hola mundo!"
    printf("Hola mundo!");

    return 0;
}
```

# Elementos del lenguaje

- Instrucciones
- Datos: literales, variables, tipos
- Subprogramas (funciones)
- Comentarios
- Directivas

# Primer programa

```
#include<stdio.h>
```

Directiva

Subprograma  
o Función

```
int main (void)
```

```
{
```

Comentario

```
//Imprimir en pantalla "Hola mundo!"
```

```
printf("Hola mundo!");
```

Instrucciones

```
return 0;
```

Datos

```
}
```

# Análisis del programa

Biblioteca

```
#include<stdio.h>
```

```
int main (void)
```

Cabecera de la función

```
{  
    //Imprimir en pantalla "Hola mundo!"  
    printf("Hola mundo!");  
  
    return 0;  
}
```

Cuerpo de la función

Bloque de código

**Las instrucciones terminan con ;**

# Análisis del programa

```
#include<stdio.h>
```

Tipo de dato

```
int main (void)
```

Cabecera de la función

```
{
```

Instrucción

```
//Imprimir en pantalla "Hola mundo!"  
printf("Hola mundo!");
```

Cadena de caracteres

Literales

Instrucción

```
return 0;
```

Número

```
}
```

Cuerpo de la función

# Análisis del programa

```
#include<stdio.h>
```

```
int main (void)
```

Cabecera de la función

```
{
```

Palabras reservadas

```
//Imprimir en pantalla "Hola mundo!"
```

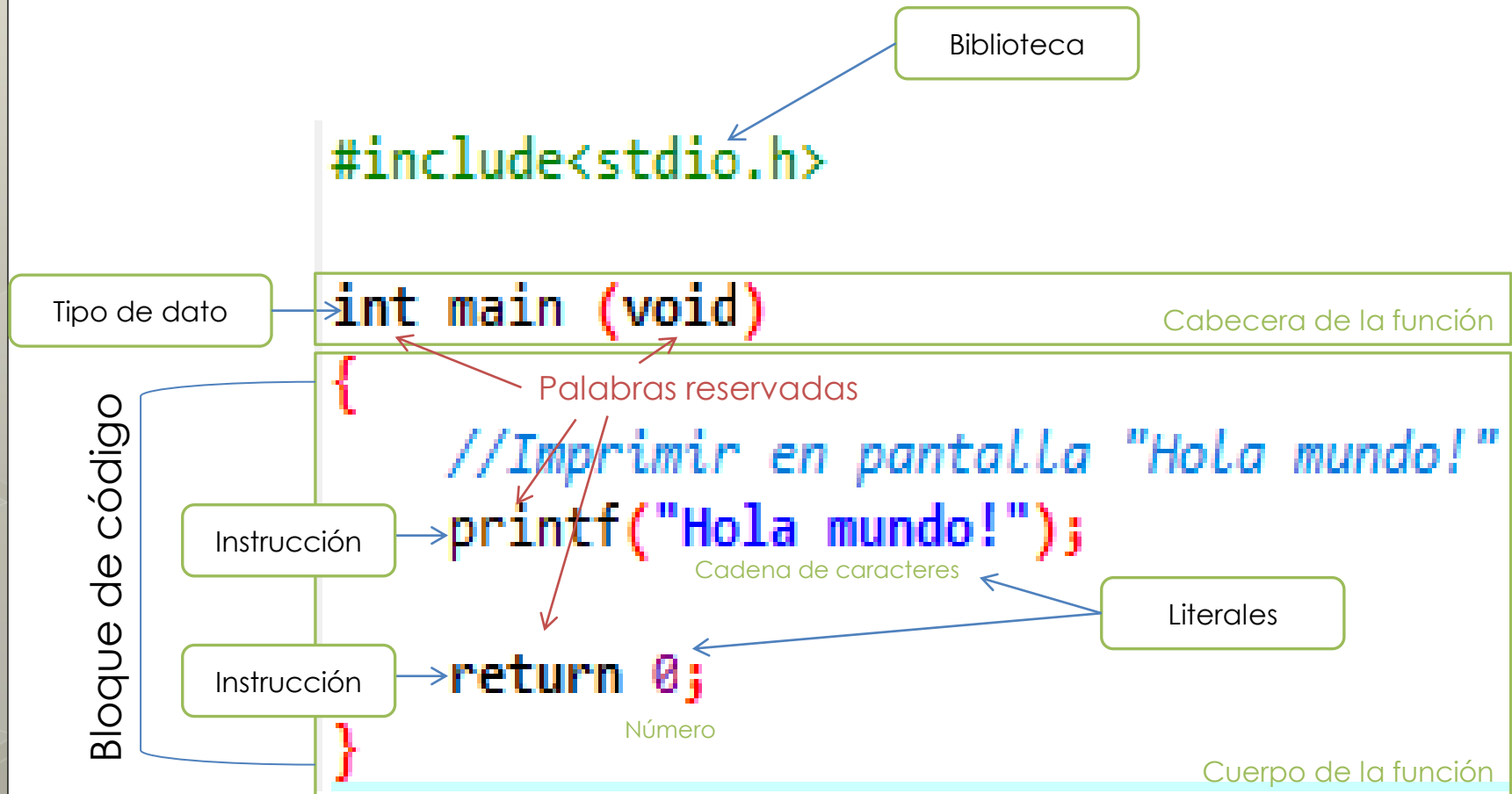
```
printf("Hola mundo!");
```

```
return 0;
```

```
}
```

Cuerpo de la función

# Análisis del programa



**Las instrucciones terminan con ;**



- *Hola Mundo!*

- Casi todo es *infraestructura*

- Sólo

```
printf( "Hola Mundo!" );
```

hace algo palpable

La infraestructura (notación, bibliotecas y otro soporte) hace nuestro código simple, completo, confiable y eficiente

*¡El estilo importa!*



