

FSID

*Fundamentos de los Sistemas de Información
Digitales*

Lenguajes de programación

Año 2025



Lenguajes de programación

Un lenguaje de programación le permite a un programador desarrollar los grupos de instrucciones que constituyen un programa.

Se han desarrollado muchos lenguajes de programación diferentes, cada uno con su vocabulario, gramática y uso únicos.

Lenguajes de programación

- Como todo lenguaje, el mismo posee reglas de escritura llamada **Sintaxis**.
- La totalidad de las instrucciones escritas en un lenguaje de programación se denomina "**código fuente**".
- El código fuente no puede ser interpretado por un CPU solamente pueden interpretar Lenguaje Máquina
- Para traducir el código fuente en código de máquina se utilizan compiladores o intérpretes.

Lenguajes de programación

- Lenguaje de primera Generación
- Lenguajes de Segunda Generación
- Lenguajes de Tercera Generación
- Lenguajes de Cuarta Generación

Lenguajes de programación

- El CPU sólo comprende **instrucciones**
- A través de UN **lenguaje de programación** se plasman las instrucciones que conforman **algoritmos**

“Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema”
(RAE)

“es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permiten llevar a cabo una actividad mediante pasos sucesivos...”
(Wikipedia)

Lenguajes de programación

- Lenguaje Máquina

- Instrucciones escritas en lenguajes que puede entender el microprocesador: “ceros” y “unos”.
- Baja productividad
- Alta eficiencia / performance
- Dependencia directa con la arquitectura de hardware del microprocesador

0011 0000 0000 0000

Lenguajes de programación

● Lenguaje Ensamblador

- Introduce el concepto de instrucciones nemotécnicas
- Baja productividad
- Alta eficiencia / performance
- Aparece el concepto de “traducción”

```
mov ax,cs  
mov ds,ax  
mov ah,9  
mov dx, offset Hello  
int 21h  
xor ax,ax  
int 21h
```

```
Hello:  
db "Hello World!",13,10,"$"
```

Lenguajes de programación

- Lenguaje de Alto nivel
- Instrucciones fáciles de recordar
- Mejora la productividad
- Mayor abstracción con respecto al hardware.

Ejemplos: Java, C, .Net.

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

```
int main()
```

```
{
```

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

```
return 0;
```

```
}
```


Lenguajes de programación

- Traductor

- ¿Qué es?
- ¿A qué se denomina “Programa Objeto”?

Tipos de traductor

- Intérpretes
- Compiladores

Lenguajes de programación

- Intérprete (proceso)

PROGRAMA
FUENTE

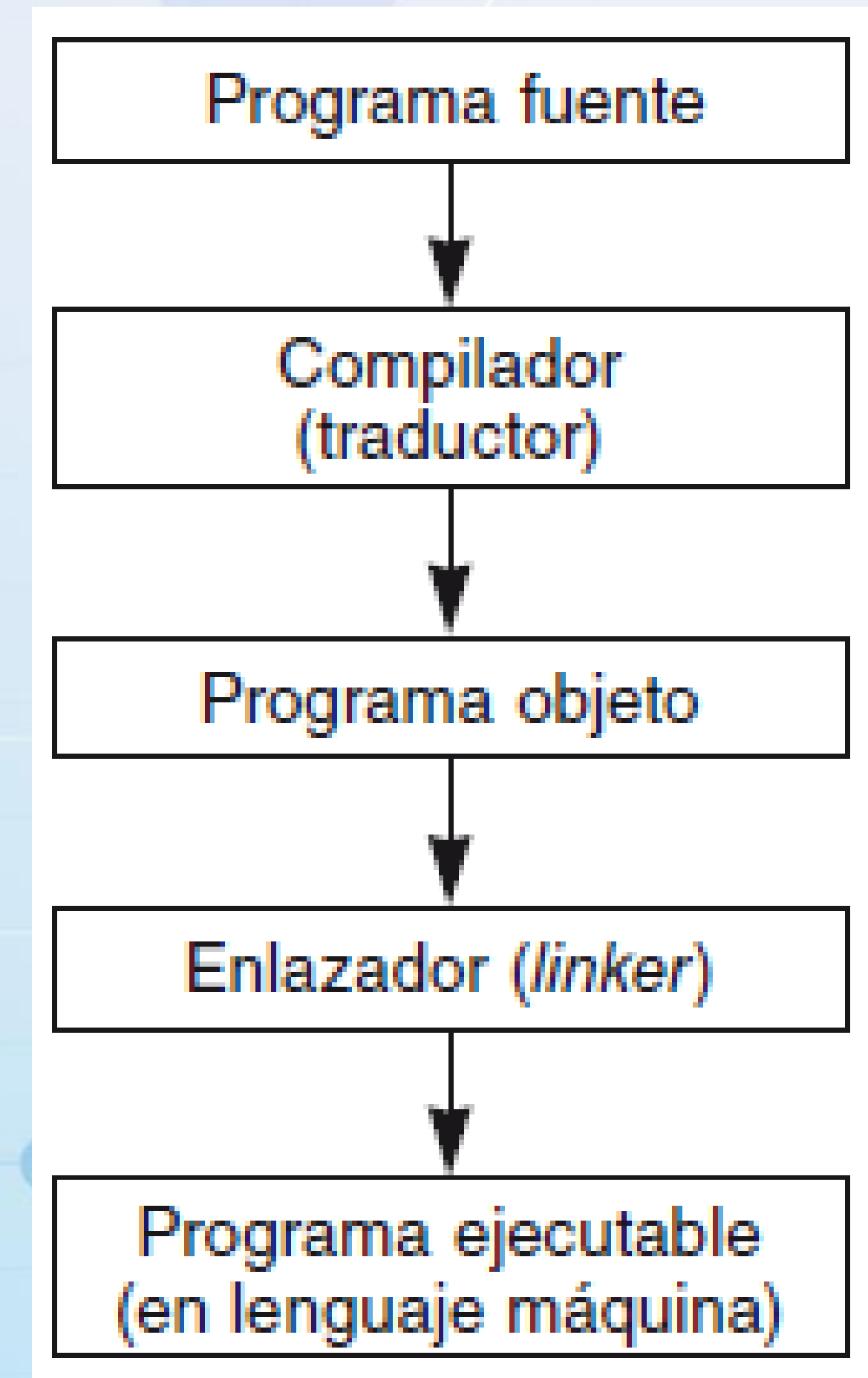
INTÉRPRETE

TRADUCCIÓN Y
EJECUCIÓN

Lenguajes de programación

- Compilador (proceso completo)

1. Escritura del programa fuente con un editor
2. Introducir el programa fuente en memoria
3. Compilar el programa con el compilador seleccionado.
4. Verificar y corregir errores de compilación (listado de errores).
5. Obtención del programa objeto.
6. El enlazador (linker) obtiene el programa ejecutable.
7. Se ejecuta el programa y, si no existen errores, se tendrá la salida del programa



Lenguajes de programación

- Paradigmas de programación ¿Qué son? ¿Cuáles existen?

Procedimental

- Instrucciones
- Estructuras de control (loops, if)
- Módulos

Programación orientada a objetos

- Clases – Entidades – Herencia
- Métodos
- Datos

Paradigmas de programación

De acuerdo a los lenguajes de programación que se usen, se deben adoptar determinados paradigmas para darle continuidad al proceso de forma fluida y sin contratiempos. Sin embargo, existen muchos paradigmas de programación, analizaremos los 5 más importantes.

1. Paradigma imperativo

El paradigma imperativo o de procedimientos es, probablemente, uno de los paradigmas más conocidos en el mundo de la programación. Como su nombre lo indica, este es un método que permite desarrollar programas a través de procedimientos. Mediante una serie de instrucciones, se explica paso por paso cómo funciona el código para que el proceso sea lo más claro posible.

2. Paradigma funcional

Una de las características del paradigma funcional es que este, como su nombre lo indica, trabaja a través de determinadas funciones matemáticas. Este es un tipo de paradigma que se usa, principalmente, en el ámbito académico más que en el comercial. A diferencia del paradigma imperativo, aquí importa más el “qué” y no tanto el “cómo” se desarrolla un proyecto.

Paradigmas de programación

3. Paradigma declarativo

El paradigma declarativo es aquel que se preocupa por el resultado final desde el inicio. Determinar de forma automática la ruta a seguir para conseguir una solución puede resultar muy eficaz a la hora de programar, solo se necesita tener claridad en torno al proceso que se va a llevar adelante.

4. Paradigma reactivo

El paradigma reactivo está enfocado en analizar el flujo de datos, ya sean finitos o infinitos, con el fin de responder a las necesidades que se presenten durante el desarrollo de los proyectos en términos de escalado, y para procurar una reacción inmediata al cambio de valores que se producen por los flujos de datos.

5. Paradigmas de programación orientada a objetos

Este tipo de paradigma de programación ofrece una guía que permite identificar cómo trabajar con él a través de objetos y planos de código. Este tipo de paradigma se constituye por piezas simples u objetos que al relacionarse entre sí forman diferentes componentes del sistema que estemos trabajando.

Lenguajes de programación

- Otros conceptos!

¿API?

¿Frameworks?

FSID

GRACIAS

Año 2025

