

# Lenguajes de Programación

## Lenguajes de Programación

- Notaciones para describir cálculos a personas y máquinas.
- Todo el software se escribe en algún lenguaje de programación.

## Compiladores

- Definición: Programas que traducen un lenguaje fuente a un lenguaje destino.
- Función Principal: Traducción y reporte de errores en el programa fuente.

## Estudio de Compiladores

- Áreas Relacionadas:
- Lenguajes de programación
- Arquitectura de máquinas
- Teoría de lenguajes
- Algoritmos
- Ingeniería de software

## Estructura de un Compilador

- Componentes Principales: Traductores de lenguaje
- Tendencias: Evolución en lenguajes de programación y arquitectura de máquinas

## Objetivos del Estudio

- Comprender el diseño de compiladores.
- Explorar la relación con la teoría de ciencias computacionales.
- Aplicaciones más allá de la simple compilación.

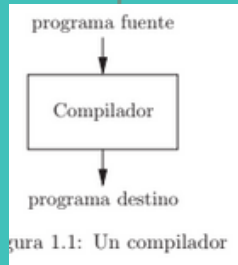


Figura 1.1: Un compilador

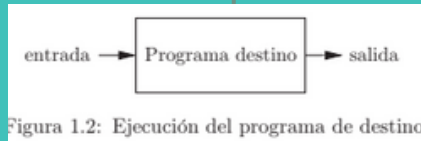


Figura 1.2: Ejecución del programa de destino

Si el programa destino es un programa ejecutable en lenguaje máquina, entonces el usuario puede ejecutarlo para procesar las entradas y producir salidas (resultados);

Un intérprete es otro tipo común de procesador de lenguaje. nos da la apariencia de ejecutar directamente las operaciones especificadas en el programa de origen (fuente) con las entradas proporcionadas por el usuario,

El programa destino en lenguaje máquina que produce un compilador es, por lo general, más rápido que un intérprete al momento de asignar las entradas a las salidas.



Figura 1.3: Un intérprete

## Aplicaciones

- Diseño e implementación de compiladores
- Reutilización de principios en diversas áreas

## Proceso de Compilación

- Análisis (Front-End)
- Divide el programa fuente en componentes.
- Aplica estructura gramatical.
- Genera Representación Intermedia.
- Crea una Tabla de Símbolos.

## Análisis Sintáctico

Utiliza Tokens para formar un Árbol Sintáctico.  
El analizador semántico utiliza el árbol sintáctico y la información en la tabla de símbolos para comprobar la consistencia semántica del programa fuente con la definición del lenguaje.

## Análisis Léxico

- Lee flujo de caracteres.
- Agrupa en Lexemas.
- Produce Tokens (Nombre-Token, Valor-Atributo).

## Síntesis (Back-End)

- Toma la Representación Intermedia y la Tabla de Símbolos.
- Genera el Programa Destino.
- Incluye Fase de Optimización de Código (opcional).

## Análisis Semántico

- Verifica coherencia y consistencia semántica.
- Generación de Código Intermedio:
- Crea una representación abstracta del código.