

## ANÁLISIS SINTÁCTICO

- Reconocimiento de la estructura del lenguaje.
- Construye un árbol de sintaxis abstracta (AST) para representar la jerarquía de las expresiones.

## ANÁLISIS LÉXICO

- Reconocimiento de los elementos del lenguaje.
- Divide el código fuente en tokens (palabras clave, identificadores, operadores, etc.).

## PROCESAMIENTO

- Transformaciones al archivo fuente antes de la compilación.
- Incluye la eliminación de comentarios, expansión de macros y resolución de directivas de preprocesador.

## ANÁLISIS SEMÁNTICO

- Reconocimiento de la coherencia de la entrada.
- Verifica la consistencia semántica, como tipos de datos, declaraciones y uso de variables.



# FASES DE UN COMPILADOR

## GENERACIÓN DE CÓDIGO INTERMEDIO

- Produce un código intermedio (como código de tres direcciones o código de máquina virtual).
- Facilita las optimizaciones y la portabilidad.

## OPTIMIZACIÓN DE CÓDIGO

- Mejora el código intermedio para reducir el tiempo de ejecución o el uso de recursos.
- Ejemplos: eliminación de código muerto, propagación de constantes, etc.

## GENERACIÓN DE CÓDIGO FINAL

- Traduce el código intermedio a lenguaje de máquina específico.
- Produce el archivo ejecutable final.