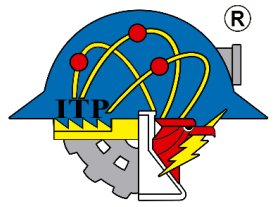




TECNOLÓGICO  
NACIONAL DE MÉXICO®



# Fases de un compilador

Lenguajes y autómatas

**Alumno:** García Cruz Andres

**N.C:** 21200598

**Profesor:** Baumé Lazcano Rodolfo

## FASES DE UN COMPILADOR

```
graph TD; Root[FASES DE UN COMPILADOR] --> A1[Análisis léxico]; Root --> A2[Análisis sintáctico]; Root --> A3[Análisis semántico]; Root --> A4[Generación de código intermedio]; Root --> A5[Optimización de código]; Root --> A6[Generación de código]; A1 --> T1_1[Reconocimiento de tokens]; A1 --> T1_2[Eliminación de comentarios y espacios en blanco]; A1 --> T1_3[Generación de una secuencia de tokens]; A2 --> T2_1[Análisis de la estructura gramatical]; A2 --> T2_2[Construcción del árbol de sintaxis abstracta (AST)]; A2 --> T2_3[Verificación de la sintaxis correcta del programa]; A3 --> T3_1[Verificación de la coherencia semántica]; A3 --> T3_2[Resolución de nombres y tipos]; A3 --> T3_3[Detección de errores semánticos]; A4 --> T4_1[Traducción del programa a una representación intermedia independiente de la plataforma]; A4 --> T4_2[Utilización de estructuras de datos intermedias.]; A5 --> T5_1[Mejora de la eficiencia del código generado]; A5 --> T5_2[Eliminación de código muerto]; A5 --> T5_3[Reducción del tamaño del código y mejora del rendimiento]; A6 --> T6_1[Traducción del Generación de código ensamblador o código de máquina]; A6 --> T6_2[Manejo de la gestión de la memoria y de los registros];
```

### Análisis léxico

Reconocimiento de tokens

Eliminación de comentarios y espacios en blanco

Generación de una secuencia de tokens

### Análisis sintáctico

Análisis de la estructura gramatical

Construcción del árbol de sintaxis abstracta (AST)

Verificación de la sintaxis correcta del programa

### Análisis semántico

Verificación de la coherencia semántica

Resolución de nombres y tipos

Detección de errores semánticos

### Generación de código intermedio

Traducción del programa a una representación intermedia independiente de la plataforma

Utilización de estructuras de datos intermedias.

### Optimización de código

Mejora de la eficiencia del código generado

Eliminación de código muerto

Reducción del tamaño del código y mejora del rendimiento

### Generación de código

Traducción del Generación de código ensamblador o código de máquina

Manejo de la gestión de la memoria y de los registros