

# *Fases del proceso de compilación*

- 1. Análisis léxico:** En este paso se leen los caracteres que se encuentran en código fuente agrupándolos en cadenas compuestas por elementos léxicos del programa. Cada elemento es una secuencia de caracteres. La consecución de letras que representa una unidad léxica se le conoce como lexeme o token (en inglés). Cuando se encuentra un nuevo identificador se anota en la tabla de símbolos, este lenguaje tendrá una posible lista de componentes que podrán ser codificados.
- 2. Análisis sintáctico:** Los componentes anteriormente nombrados se unen para formar frases, en las cuáles el significado de las partes no importa. Generalmente, las frases se muestran usando un árbol sintáctico. La separación entre el análisis léxico y el análisis sintáctico no es algo que se suele decidir a conciencia. Normalmente, el proceso de división consta de la simplificación de las tareas, dejando las construcciones guardadas en paréntesis u operadores. El análisis sintáctico utilizará técnicas más complejas que el léxico.
- 3. Análisis semántico:** En esta etapa será revisado el resultado del paso anterior, recopilando datos de la misma para construir una representación más abstracta. A la misma vez, en esta fase, se hace un seguimiento de los identificadores.
- 4. Generación de código intermedio:** Como su propio nombre indica, se crea un código intermedio para una máquina abstracta. Este código debe ser fácil de producir y de traducir. El código puede tener diferentes formas.
- 5. Optimización código intermedio:** El objetivo de esta fase es mejorar el código, reduciendo la cantidad de recursos necesarios y organizando la secuencia de declaraciones con el fin de acelerar la ejecución del programa.
- 6. Generación del código final:** Es la fase de generación del código definitivo. Las variables son asignadas a registros y cada instrucción del código intermedio se convierte en una serie de acciones que realizan la tarea.