***Tipos de gramáticas, la estructura de los compiladores y tabla de símbolos.***

1. **Tipos de gramática**
2. La gramática descriptiva:

intenta describir el uso actual de una lengua, evitando juzgar en forma prescriptiva.

1. La gramática funcional:

contempla tres normas básicas de adecuación: la tipológica, que implica la aplicación de reglas a cualquier lengua, la pragmática, que promueve la aplicación de los enunciados a la interacción en la comunicación, y la psicológica, por la que trata de ser compatible con los mecanismos psicológicos involucrados en el procesamiento de un lenguaje natural.

1. La gramática prescriptiva, o gramática normativa:

es la que presenta normas de uso para un lenguaje específico, tendiendo a desaconsejar las construcciones no estandarizadas. La gramática tradicional es típicamente prescriptiva.

1. La gramática tradicional:

La gramática prescriptiva es formulada usualmente en términos de los conceptos descriptivos heredados de la gramática tradicional.

1. La gramática generativa:

es un enfoque formal para el estudio sintáctico de las lenguas que puede en algunos sentidos generar las expresiones bien construidas de una lengua natural. Una rama de la teoría lingüística (psicolingüística) se basa en la gramática generativa, promovida por Noam Chomsky.

1. Las gramáticas formales aparecen en lingüística computacional. La sintaxis de cada lenguaje de programación se define de hecho por una gramática formal. En teoría de la informática y en matemática, la gramática formal define lenguajes formales. La Jerarquía de Chomsky describe varias clases importantes de gramática formal.
2. **La Estructura de los Compiladores**

La parte del análisis divide el programa fuente en componentes e impone una estructura gramatical sobre ellas. Después utiliza esta estructura para crear una representación intermedia del programa fuente. Si la parte del análisis detecta que el programa fuente está mal formado en cuanto a la sintaxis, o que no tiene una semántica consistente, entonces debe proporcionar mensajes informativos para que el usuario pueda corregirlo. La parte del análisis también recolecta información sobre el programa fuente y la almacena en una estructura de datos llamada tabla de símbolos, la cual se pasa junto con la representación intermedia a la parte de la síntesis.

La parte de la síntesis construye el programa destino deseado a partir de la representación intermedia y de la información en la tabla de símbolos. A la parte del análisis se le llama comúnmente el front-end del compilador; la parte de la síntesis es el back-end.

1. **Tabla de Símbolos**

Para almacenar los nombres de todas las entidades de forma estructurada en un solo lugar.

Para verificar si se ha declarado una variable.

Comprobación del tipo de implemento, comprobando las cesiones y las expresiones en el código fuente son semánticamente correcto.

Para determinar el alcance de un nombre (alcance de la resolución).

Una tabla de símbolos es simplemente una tabla que puede ser lineal o una tabla hash.