

### 1. ¿Qué tarea realiza el analizador sintáctico en el proceso de compilación?

Es determinar la estructura sintáctica de un programa a partir de los tokens producidos por el analizador léxico, ya sea de manera explícita o implícita

### 2. ¿Qué relación existe entre el analizador léxico y el analizador sintáctico?

El analizador léxico se encarga de producir la secuencia de tokens a partir del código fuente mientras que el analizador sintáctico utiliza estos tokens como entrada para determinar la estructura gramatical del programa

### 3. ¿Qué es un árbol sintáctico y para qué sirve?

Representación estructurada de como los tokens se organizan según las reglas gramaticales establecidas y sirve para formar una estructura jerárquica del programa y facilitar la detección y recuperación de errores sintácticos

### 4. Menciona las dos categorías generales de algoritmos de análisis sintáctico.

Análisis sintáctico descendente y el análisis sintáctico ascendente

### 5. ¿Qué sucede si el analizador encuentra un error sintáctico?

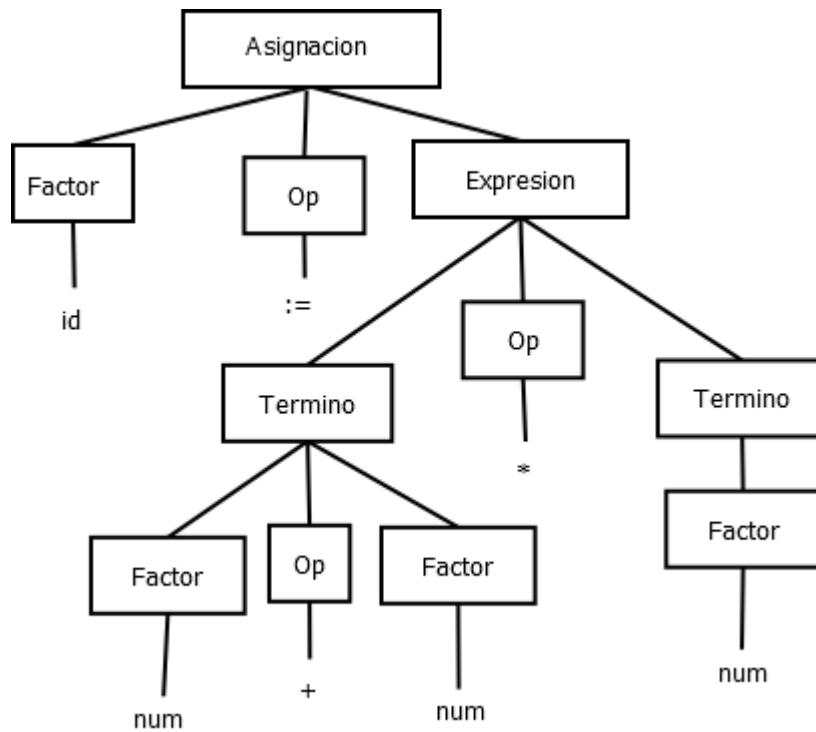
Muestra un mensaje de error, además de tener que recuperarse del error y continuar el análisis sintáctico para encontrar más errores. Durante la recuperación de errores puede inferir una posible versión de código corregida a partir de la versión incorrecta que se ejecuto

## Parte 2: Aplicación práctica

Analiza la siguiente secuencia de tokens generada por un analizador léxico: **id**

**:= num + num \* num ;**

1. Dibuja (a mano o con un programa) el **árbol de análisis gramatical** correspondiente.



### Parte 3: Reflexión

Responde en 3 o 4 líneas:

¿Por qué crees que el análisis sintáctico es una fase esencial antes de generar código en un compilador?

Porque la función principal del análisis sintáctico es verificar que el programa este escrito de acuerdo con las reglas del lenguaje, mediante la construcción de un árbol sintáctico en que establece las relaciones entre los elementos del programa. Además, que permite detectar y recuperar errores sintácticos antes de generar código, evitando que el compilador produzca código inejecutable