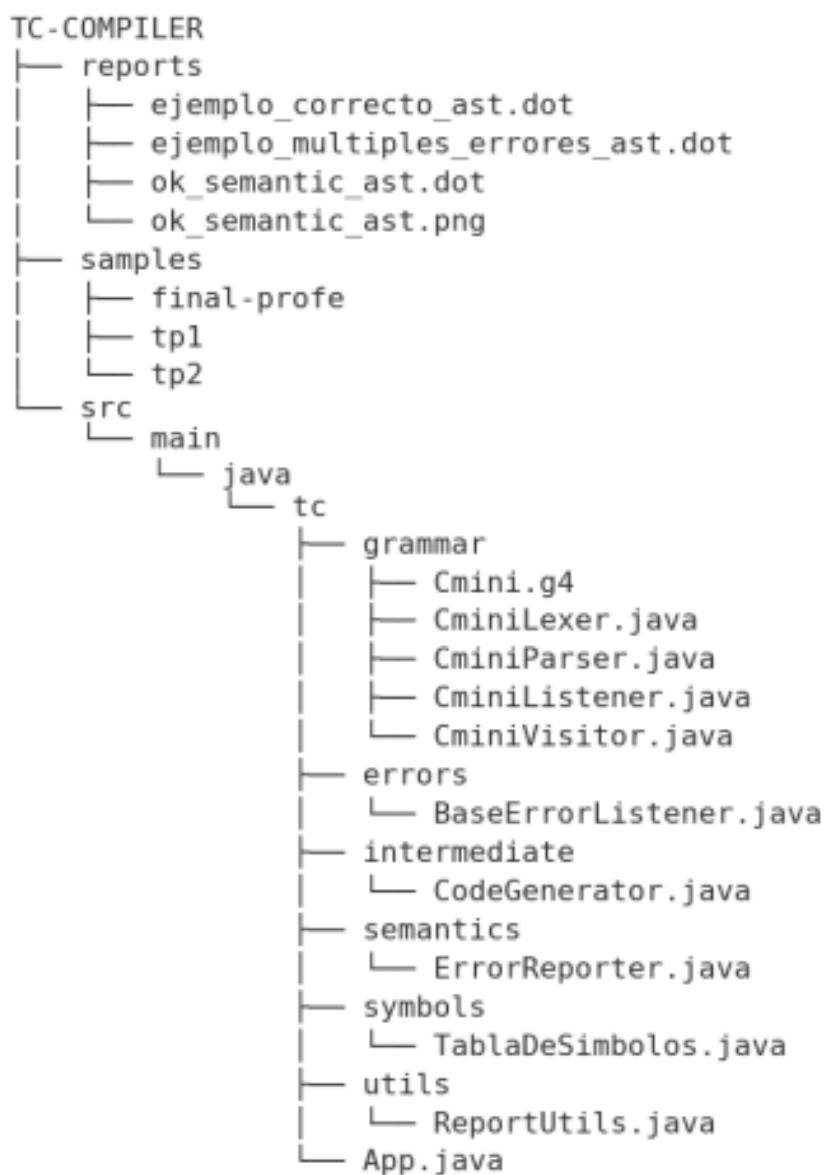


# Documentación del Proyecto TC-COMPILER

## Estructura del proyecto

El proyecto sigue una estructura típica con Maven. Incluye la gramática ANTLR en la carpeta grammar, módulos para errores, semántica, generación de código intermedio, tabla de símbolos, y utilidades. Los archivos de prueba se encuentran en samples y los reportes (árboles AST en formato DOT/PNG) en reports.

## Diagrama de la estructura de carpetas



## Herramientas utilizadas

- ANTLR4: genera Lexer, Parser, Listener y Visitor a partir de la gramática .g4. •
- Maven: gestión de dependencias y compilación.
- Graphviz (dot): genera imágenes PNG a partir de archivos DOT.
- Visual Studio Code: IDE utilizado.
- Prettier Java Plugin: extensión para formatear automáticamente el código.

## Instructivo paso a paso

**1. Instalación:** Instalar Java JDK 17+, Maven, ANTLR4 y Graphviz. En VS Code instalar la extensión Prettier Java Plugin.

**2. Generar clases ANTLR4:** Ejecutar: `mvn clean compile`

**3. Ejecutar un archivo de prueba:** `mvn exec:java`  
`-Dexec.args="samples/final-profe/ejemplo_correcto.c"`

**4. Generar PNG del árbol sintáctico:** `dot -Tpng reports/ok_semantic_ast.dot -o reports/ok_semantic_ast.png`

**5. Revisar resultados:** Archivos DOT/PNG en reports, logs intermedios en la raíz del proyecto.

Este instructivo resume el flujo completo: definición de gramática, compilación, ejecución de pruebas y generación de reportes gráficos. Facilita la comprensión de la arquitectura del compilador y su uso práctico.