

# [M1] Laboratorio 1: Introducción a ANTLR



Alfredo Quezada

Eduardo Ramirez

## ¿Qué es ANTLR?

ANTLR (ANother Tool for Language Recognition) es una herramienta para generar analizadores léxicos y sintácticos a partir de una gramática formal escrita en un archivo `.g4`.

## Estructura de un archivo `.g4`

Un archivo `.g4` suele tener las siguientes secciones:

### 1. Encabezado (Opcional)

Se escriben directivas como el nombre del paquete o imports:

```
antlr
CopiarEditar
grammar NombreGramatica;
```

### 2. Parser Rules (Reglas sintácticas)

- Se escriben en minúscula.
- Definen la estructura del lenguaje (cómo se combinan los tokens).

```
antlr
CopiarEditar
```

```
expr: term ('+' term)*;
```

### 3. Lexer Rules (Reglas léxicas)

- Se escriben en mayúscula.
- Definen los tokens (palabras clave, identificadores, símbolos, etc.)

```
antlr  
CopiarEditar  
ID : [a-zA-Z]+ ;  
NUM : [0-9]+ ;
```

### 4. Acciones en código embebido (Opcional)

Código en el lenguaje objetivo (como Java o C#) para ejecutar durante el análisis:

```
antlr  
CopiarEditar  
{System.out.println("Token encontrado");}
```

### 5. Comentarios

En ANTLR se usa `//` o `/* */`, pero `#` **no se usa** como en Python, este se usa para etiquetar alternativas en reglas, permitiendo distinguir entre diferentes formas de una regla en el árbol de análisis.

## Archivo Driver ( **Driver.py** )

El archivo `Driver.py` es el punto de entrada del programa que utiliza la gramática generada por ANTLR. Sus partes principales son:

#### 1. Importación de módulos ANTLR:

Importa el lexer, parser y otros componentes generados.

## 2. **Lectura de entrada:**

Lee el archivo fuente o el texto a analizar.

## 3. **Creación de lexer y parser:**

Usa el texto de entrada para crear un lexer (tokenizador) y un parser (analizador sintáctico).

## 4. **Parseo:**

Ejecuta el parser sobre los tokens para obtener el árbol de análisis.

## 5. **Listeners o visitors:**

Puede usar listeners o visitors para recorrer el árbol y realizar acciones (por ejemplo, interpretación, generación de código, análisis semántico).

## **Enlace al video:**

Dentro del readme del repositorio.

## **Link del repositorio:**

<https://github.com/eduardordev/Compiladores>