

Lab1

Alfredo Quezada

Eduardo Ramirez

¿Qué es un archivo **.g4** en ANTLR?

Un archivo **.g4** contiene la gramática que ANTLR usa para generar automáticamente:

- Un **lexer** (analizador léxico)
- Un **parser** (analizador sintáctico)
- Listeners y visitors opcionales

Su estructura general es:

```
grammar NombreGramática;
```

```
parser rules...
```

```
lexer rules...
```

Partes principales de un archivo **.g4**

1. Declaración de la gramática

Ejemplo:

```
grammar MiniLang;
```

Define el nombre del lenguaje. ANTLR usa esto para nombrar las clases generadas.

2. Reglas del Parser (parser rules)

- Siempre empiezan con **minúscula**
- Definen la **sintaxis** del lenguaje (cómo se combinan los tokens)
- Son equivalentes a las reglas de una gramática libre de contexto

Ejemplo:

```
expr: expr '+' term
    | term
    ;
```

3. Reglas del Lexer (lexer rules)

- Siempre empiezan con **MAYÚSCULA**
- Definen cómo se reconocen los **tokens** del lenguaje
- Describen patrones de caracteres (similares a regex)

Ejemplo:

```
NUMBER: [0-9]+;
ID: [a-zA-Z_][a-zA-Z0-9_]*;
```

4. Acciones semánticas (opcionales)

Permiten ejecutar código dentro del parser, aunque en cursos introductorios no se usan.

5. Canales y modos (avanzado)

Para tokens que deben ignorarse (espacios, comentarios, etc.)

Ejemplo:

WS: [\t\r\n] → skip;

¿Qué hace el archivo **Driver.py** ?

Driver.py tiene como función:

1. Lee el archivo de entrada (por ejemplo `program_test.txt`)
2. Crea un **InputStream**
3. Pasa el texto al **lexer** → genera tokens
4. Pasa los tokens al **parser**
5. Ejecuta la regla inicial del parser (por ejemplo `program`)
6. ANTLR realiza análisis sintáctico
7. Si hay errores, ANTLR los muestra automáticamente
8. Si NO hay errores, el programa termina silenciosamente

Ejemplo:

```
input_stream = FileStream(argv[1])
lexer = MiniLangLexer(input_stream)
token_stream = CommonTokenStream(lexer)
parser = MiniLangParser(token_stream)
tree = parser.program()
```

¿Cómo ANTLR reporta errores?

ANTLR trae un **error handler por defecto** que:

- Muestra línea y columna
- Indica el token inesperado
- Dice qué esperaba ver

Ejemplo:

```
line 3:5 extraneous input '}' expecting ';'
```

Si no hay errores:

→ **No aparece nada en la consola.**

Enlace al video:

<https://youtu.be/tO4BW7jUiIg>