

# TÉCNICAS DE COMPILACIÓN

## Practico 1

Análisis Léxico y Sintáctico con ANTLR

**Fecha de Entrega:** 28/05/24

**Profesor:** Maximiliano Eschoyez

**Integrantes**

- Constanza Gigli
- Francisco Andreoli

**GITHGUB:**

<https://github.com/FranciscoAndreoli/TecnicasCompilacion>

# INTRODUCCIÓN

Este trabajo aborda lo aprendido en el periodo sobre Análisis Léxico y Sintáctico, realizando analizadores de código con ANTLR. El objetivo es construir un analizador para una versión reducida del lenguaje C. El programa es capaz de tomar un archivo escrito en C y generar el árbol sintáctico correspondiente. Este informe describe el proceso de desarrollo y los resultados obtenidos.

## DESARROLLO DE LA SOLUCION

### 1.PROBLEMÁTICA

El análisis léxico y sintáctico son pasos fundamentales en la compilación de un lenguaje de programación. El análisis léxico convierte una secuencia de caracteres en una secuencia de tokens, mientras que el análisis sintáctico organiza estos tokens en una estructura jerárquica conocida como árbol sintáctico. Nuestro trabajo se centra en reconocer bloques de código, declaraciones, asignaciones, operaciones aritmeticológicas, estructuras de control y declaración y llamada de funciones.

### 2.ANÁLISIS LÉXICO

Para el análisis léxico, definimos las reglas léxicas en ANTLR que permiten identificar tokens básicos del lenguaje C. Palabras clave (int, double, if, else, for, while, return), operadores (+, -, \*, /, =), delimitadores (;, (, ), {, }), identificadores y números.

### 3.ANALISIS SINTÁCTICO

El análisis sintáctico lo implementamos mediante la definición de una gramática en ANTLR que describe la estructura de programas en el lenguaje C. A continuación, se presentan las reglas principales utilizadas:

Instrucción: Incluye declaraciones, asignaciones, bucles (while, for), estructuras condicionales (if), retorno y llamadas a funciones.

Declaración: Define variables y su tipo (int, double).

Asignación: Asigna valores a las variables.

Expresión: Define operaciones aritméticas y lógicas.

Estructuras de Control: Define las estructuras if, for y while.

Llamada a Función.

## CONCLUSIÓN

El trabajo práctico nos permitió implementar un analizador léxico y sintáctico para una versión reducida de C utilizando ANTLR. La solución desarrollada cumple con los requisitos especificados, incluyendo el reconocimiento de bloques de código, declaraciones, asignaciones, operaciones aritméticas y lógicas, y estructuras de control. La validación mediante ejemplos de código demostró que el parser genera el árbol sintáctico correcto, proporcionando una base sólida para futuros desarrollos