

Proyecto de semestre: Analizador léxico-sintáctico

Materia: Programación de Sistemas de Base 1

Duración: Semestral

1. Objetivo del Proyecto

El objetivo principal del proyecto es aplicar las técnicas de **análisis léxico y sintáctico** estudiadas en la materia, para resolver un problema práctico relacionado con el procesamiento de lenguajes o formatos estructurados. Los equipos deberán diseñar e implementar una herramienta que demuestre el uso de estas etapas fundamentales de un compilador, ya sea para interpretar, traducir o transformar información siguiendo reglas específicas.

2. Alcance del Proyecto

El proyecto se enfocará exclusivamente en las etapas de **análisis léxico** (reconocimiento de tokens) y **análisis sintáctico** (validación de la estructura según una gramática). No es necesario implementar fases posteriores como análisis semántico.

3. Detalles del Proyecto

El proyecto deberá incluir, como mínimo, los siguientes componentes:

3.1. Análisis Léxico

- Definición formal de los **tokens** del lenguaje o formato elegido.
- Implementación de un **lexer** (analizador léxico) utilizando herramientas como:
 - Herramientas específicas: **Lex**, **Flex**, **ANTLR**, **PLY** (Python Lex-Yacc).
 - Implementación manual (ejemplo: autómatas finitos).
- Manejo de errores léxicos (ejemplo: caracteres no reconocidos).

3.2. Análisis Sintáctico

- Implementación de un **parser** (analizador sintáctico) utilizando:
 - Herramientas como **Yacc**, **Bison**, **ANTLR**, **PLY**.
 - Algoritmos como descendente recursivo o LR.

- Generación de un árbol sintáctico (AST) o estructura intermedia.

3.3. Funcionalidad Adicional (Opcional)

- Interfaz de usuario sencilla (consola o gráfica).
 - Manejo de errores sintácticos (ejemplo: estructuras mal anidadas).
-

4. Entregables

1. Archivos Generados en el proyecto (p. ej. código fuente, librerías, archivos de entrada y salida, etc.):

- Implementación completa del lexer y parser.
- Ejemplos de entrada/salida para demostrar el funcionamiento.

2. Documentación Técnica:

- Descripción del lenguaje/formato trabajado.
- Especificación de tokens y reglas gramaticales.
- Explicación de decisiones de diseño y herramientas utilizadas.

3. Informe Final:

- Introducción, objetivos y justificación.
 - Diagramas relevantes (ejemplo: autómatas, árboles sintácticos).
 - Pruebas realizadas y casos de error manejados.
-
- Se deberá entregar todos el software relativo al proyecto en un enlace a un repositorio personal público y accesible.
 - El repositorio debe incluir además de los archivos del propio proyecto, un readme estructurado al menos con la siguiente información:
 - Nombre del Proyecto
 - Breve descripción del proyecto (1-2 líneas).
 - Información del Curso
 - Materia: Nombre de la materia.
 - Institución: Universidad o institución.
 - Semestre: Ej: “Primer semestre de 2023”.
 - Profesor(es): Nombre del profesor o responsables.
 - Integrantes del Equipo

- Se deberá entregar adicionalmente en la plataforma de Moodle un archivo PDF con el código fuente y capturas pantallas en distintas fases de la ejecución.
-

5. Consideraciones Finales

- Los equipos pueden ajustar el alcance según su complejidad, pero deben justificar cualquier desviación.
- Se recomienda utilizar herramientas y lenguajes conocidos (ejemplo: Python, Java, C++) para agilizar el desarrollo.
- La claridad en la definición de tokens y gramática es crítica para el éxito del proyecto.

Integridad Académica

El código debe ser original. Se permiten referencias externas, pero deben citarse. Cualquier plagio resultará en calificación cero.