
PROYECTO 1 - COMPSRIPT

202000119 – Sergie Daniel Arizandieta Yol

Resumen

Dicho proyecto esta enfocado en la aplicación de analizadores léxico, sintáctico y semántico en la aplicación de crear un lenguaje de programación de alto nivel.

Dicho proyecto fue desarrollado en Angular y Node.js y para el análisis de toda la aplicación se utilizó Typescript. Dicha aplicación se basa en un lenguaje que se proporcionara más adelante la información específica, pero si aplicaron modelos de programación para lograr dicho proyecto.

El proyecto hace utilización de estas tecnologías de una manera novedosa para la búsqueda, lecturas y escrituras optimas de los datos a gestionar para la Constructor de análisis léxico , sintáctico y semántico requerido en el 2022 (Se recalca que el proyecto no es una fase final).

Palabras clave

INTERPRETE, GRAMATICAS,
ANALIZADORES, ANGULAR, NODE JS

Abstract

This project is focused on the application of lexical, syntactic and semantic analyzers in the application of creating a high-level programming language.

This project was developed in Angular and Node.js and Typescript was obtained for the analysis of the entire application. Said application is based on a language that specific information will be provided later, but if they applied programming models to achieve said project.

The project makes use of these technologies in a novel way for the optimal search, reading and writing of the data to be managed for the Builder of lexical, syntactic and semantic analysis required in 2022 (It is emphasized that the project is not a final phase) .

Keywords

INTERPRETER, GRAMMAR, PARSER, ANGULAR,
NODE JS

Introducción

Los analizadores léxico, sintácticos y semánticos no son los únicos manejados en un compilador, pero para el enfoque del proyecto se utilizaron estos para la generación de una salida según funciones declaradas en un lenguaje establecido.

Así mismo la salida de las cadenas se verá reflejado en una consola simulada tratando de recuperarnos de diferentes errores que se presente en el proceso de ejecución.

También se aclara que se implemento un ATS para la funcionalidad del mismo.

Desarrollo del tema

Las tecnologías requeridas y fundamentales para el desarrollo del software son:

- ❖ Implementación de analizadores
 - léxicos
 - sintácticos
 - semánticos
- ❖ Generación de grafos mediante Graphviz
- ❖ Lenguaje definido

Conceptos generales

- **Compilador**
Compile es el proceso de transformar un programa informático escrito en un lenguaje en un programa equivalente en otro formato. Al programa que se encarga de compilar se le llama compilador.

- **Analizador léxico:**

Un analizador léxico o analizador lexicográfico es la primera fase de un compilador, consistente en un programa que recibe como entrada el código fuente de otro programa y produce una salida compuesta de tokens o símbolos.

- **Analizador sintáctico**

Un analizador sintáctico o simplemente analizador, es un programa informático que analiza una cadena de símbolos según las reglas de una gramática formal.

- **Analizador semántico**

Traducción del inglés-En lingüística, el análisis semántico es el proceso de relacionar estructuras sintácticas, desde los niveles de frases, cláusulas, oraciones y párrafos hasta el nivel de la escritura en su conjunto, hasta sus significados independientes del lenguaje.

- **Grafos en Graphviz:**

Graphviz es un software de visualizador de gráficos de código abierto, el cual implementa su extensión DOT para sus grafos.

➤ **DOT**

Es un lenguaje descriptivo de texto plano el cual es una forma simple de describir los grafos.

- **Lenguaje:**
El lenguaje a utilizar tiene instrucción específicas, las cuales son muy similares a un lenguaje de alto nivel, dicho lenguaje es miu extenso para poderlo explicar en este manual, pero se deja el link del mismo, proporcionado en organización de lenguajes y compiladores 1, 1er semestre del año 2022 Link:
<https://drive.google.com/file/d/11GZdGxl-0KQVoQyUij4yRJOE3K2wjQOR/view?usp=sharing>

Representación de los módulos del programa

Tabla I.

Módulos generales del proyecto.

<i>Clases</i>
Analizador
Interprete
AST
Expresiones
Instrucciones
Interfaces
Tabla de símbolos
Controlador

Descripción de módulos:

- **Analizador**
Este posee la gramática la cual acepta el lenguaje junto a las dependencias para ejecutarlas
- **Interprete**
Este contiene diferentes clases para analizar y ejecutar las funciones específicas del mismo

- **AST**
En lenguajes formales y lingüística computacional, un árbol de sintaxis abstracta (AST), o simplemente un árbol de sintaxis, es una representación de árbol de la estructura sintáctica simplificada del código fuente escrito en cierto lenguaje de programación.
- **Expresiones**
Son las declaraciones que se entiende como el conjunto de operandos y operadores, que después de ser ejecutados generan una salida.
- **Instruccions**
Son las declaraciones que generan algún tipo de cambio en la ejecución del programa.
- **Interfaces**
Es un conjunto de clases que funcionan como medio común para que los objetos no relacionados se comuniquen entre sí
- **Tabla de símbolos**
Es la tabla encargada de guardar las variables y declaraciones durante la ejecución del programa.
- **Controlador**
Es el principal modulo que controla la ejecución del programa y el recorrido del ASR para la generación de la salida.

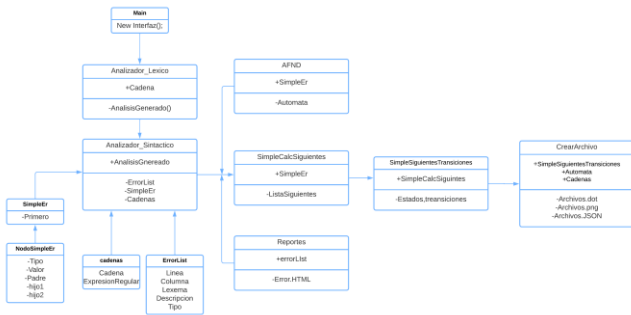


Figura 1. Diagrama de Clases.

Fuente: elaboración propia.

Conclusiones

El uso de nuevas tecnologías, implementaciones inclusive librerías que puedan hacer el trabajo de desarrollo es vital para una empresa viéndolo a gran escala, teniendo en cuenta dichas tecnologías se podrían generar lenguajes de programación de alto nivel al gusto de la persona y que sean novedosos para la sociedad así que ¿Cuál será tu nuevo lenguaje?

Referencias bibliográficas

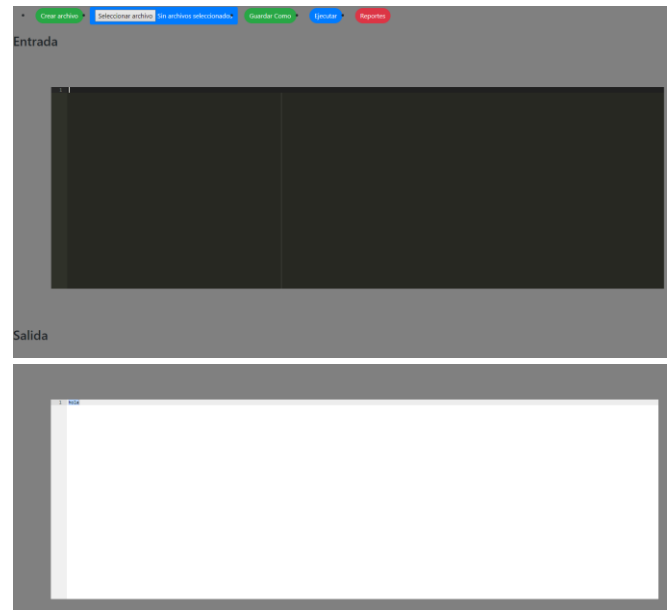
Arias Guerra, D. (2008) Estructura de datos Avanzadas (Revisado, ed., Vol. 9). Universidad de las Ciencias Informáticas. <https://cutt.ly/eWdkzyt>

Ellson, J. (2005) Graphviz. Graphviz org.
<https://graphviz.org/doc/info/lang.html>

Ferris Castell, R. (2004) Algoritmos y Estructura de Datos I (Revisado ed., Vol. 1) Universidad de Valencia
http://informatica.uv.es/jiguia/AED/oldwww/200102/Teoria/Tema_10.pdf

Anexos

Modelo “Intefaz”



Fuente: elaboración propia.