Pseudo-**Parser**

Manual técnico

201906588 - Byron Estuardo Solís González

Segundo semestre - Septiembre 2022

1. Introducción

Se desarrolló un programa que permite trasladar un texto de pseudocódigo a código funcional en Lenguaje Python.

Para su creación se emplearon las herramientas de JFLEX y CUP, para la creación del analizador léxico y sintáctico respectivamente; La aplicación les emplea para validar la estructura del texto entrante y construir el código saliente.

El resultado fue un Pseudo-Parser de fácil uso, que además de la conversión a código le ofrece al usuario la visualización del comportamiento de la gramática del análisis sintáctico a través de un grafo y un reporte de errores en la entrada.

2. Requerimientos del sistema

a. Hardware

- Dispositivo de ejecución. Se necesita de un dispositivo capaz de ejecutar archivos .java, con visualizador de imágenes y archivos html.

b. Software

- i. Apache Netbeans IDE
- ii. JDK
- iii. Librerías
  - jflex-full-1.7.0
  - java-cup-11b
  - java-cup-11b-runtime

3. Procesos destacados

Antes de producir una salida de código la entrada debe pasar por 3 fases dentro de la aplicación, el análisis léxico, la primera fase en ejecutarse siendo la encargada de la creación de los tokens de entrada, y dos pasadas en el análisis sintáctico, las cuales se explican más a detalle a continuación

• Primera pasada – AST

En esta pasada se valida el correcto orden de los tokens producidos por en análisis léxico, Al tiempo que la información fluye entre la gramática se va creando un árbol de forma ascendente, siendo todos los símbolos tanto terminales como no terminales nodos de este. En estas producciones se crean nodos, retornan y enlazan nodos, siendo los símbolos de una producción los hijos del no terminal que la produce

A partir de un recorrido de este se crea el grafo.

• Segunda pasada – Traducción

Mientras que en la fase pasada se iban creando y enlazando nodos en un árbol, es esta se manejan solamente cadenas, que mediante la gramática va aceptando tokens también se va extrayendo y encadenando su contenido, dándole la forma del lenguaje destino.

Link Repositorio: https://github.com/Byron7h/OLC1-201906588.git