

Facultad de Ingenieria

Escuela de Sistemas

OLC 1S 2022

Manual Técnico

Marco Antonio Solís González

202003220

La Facultad de Ingenieria desea un lenguaje llamado TypeWise que será un interprete, se realizó como aplicación WEB

Objetivos

- Reforzar los conocimientos de análisis léxico y sintáctico para la creación de un lenguaje de programación.
- Aplicar los conceptos de compiladores para implementar el proceso de interpretación de código de alto nivel.
- Aplicar los conceptos de compiladores para analizar un lenguaje de programación y producir las salidas esperadas.
- Aplicar la teoría de compiladores para la creación de soluciones de software.

Requerimientos

Visual code studio

Tener conocimiento basico de JavaScript y typescript

Tener conocimientos de Node js.

Windows 7 o superior

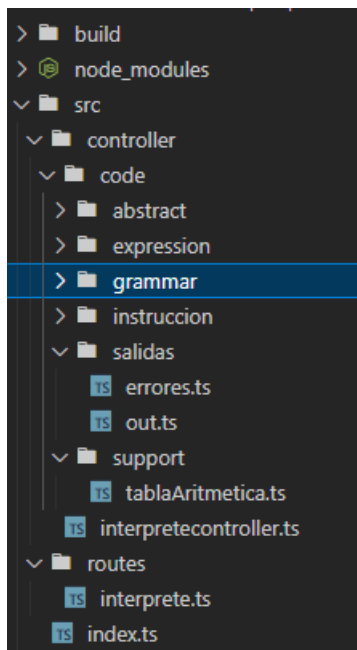
Tener acceso a internet

Alcance

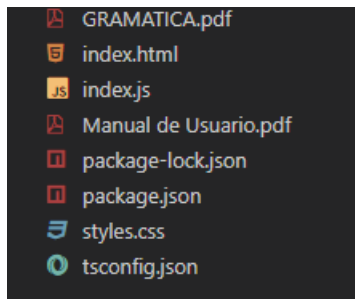
Crear un lenguaje funcional con un propio sello del cual pueda distinguirse, se puso en juego los conocimientos adquiridos, esto para poder realizar proyectos con mayor dedicación en el futuro.

Lógica del programa

Primeramente cabe recalcar que se trabajo por medio de un servidor de typescript creado con node js los archivos de este fueron



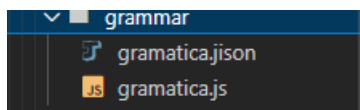
Esto es únicamente el servidor de typescript, luego para el frontend se utilizó un html puro con javascript y css, la interfaz está en la raíz de la carpeta



Luego en el servidor de typescript se crearon diferentes rutas para realizar las peticiones y al lograr tener acceso, se creó el código del intérprete

Carpeta

Grammar:



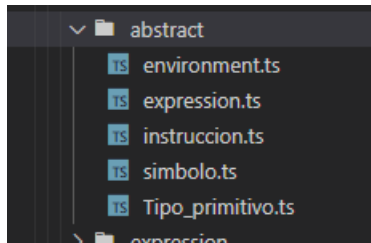
Para comenzar se creó un archivo json donde se realizó la gramática para nuestro proyecto, para mayor detalle puede leer el archivo de gramática que se subió junto con este proyecto.

Abstract:

Aquí se creó todas las clases abstractas de nuestro proyecto:

- **instrucción:** Este archivo fue creado con la finalidad de ser una clase padre para todas las ejecuciones de tipo Instrucción de nuestro código.
- **Environment:** Este manejará todos nuestros entornos creados para nuestro código

- Expression: Este fue creado con la finalidad de ser una clase padre para todas las ejecuciones de tipo Expression de nuestro código.
- Símbolo: Este archivo tenía la finalidad de representar a las variables de nuestro código.
- Tipo_primitivo: Aquí se definió una enumeración para guardar el tipo de dato que se este ingresando y retornar el valor y tipo.

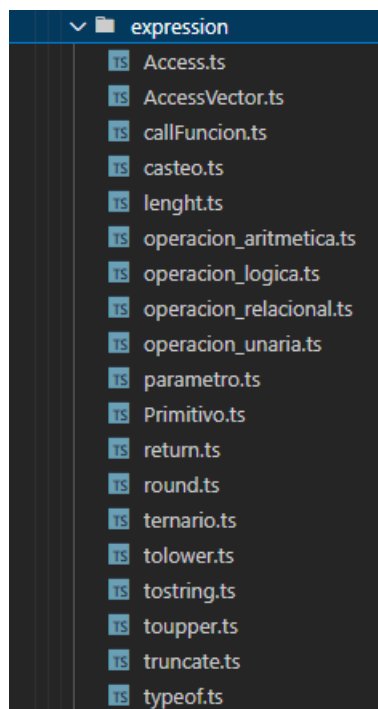


Expression:

Aquí se crearon todas las expresiones de nuestro código:

- Access: Se encarga de acceder a las variables
- AccessVector: Se encarga de acceder a los vectores
- CallFuncion: Se encarga de llamar a las funciones
- Lenght: Se encarga de realizar la función nativa length
- Operación Aritmética: Se encarga de realizar todas las operaciones aritméticas
- Operación lógica: Se encarga de realizar todas las operaciones lógicas.
- Operación relacional: Se encarga de realizar todas las operaciones relacionales
- Operación unaria: Se encarga de realizar todas las operaciones unarias
- Parámetro: se encarga de definir los parámetros de las funciones

- Primitivo: Se encarga de definir los valores primitivos como int, string, boolean, char, double.
- Return: Se encarga de retornar en las funciones:
- Round: Se encarga de realizar la función nativa rounda
- Ternario: SE encarga de realizar la función nativa ternario
- Tolower: Se encarga de realizar la función nativa tolower.
- ToString: Se encarga de realizar la función nativa ToString
- ToUpper: Se encarga de realizar la función nativa Toupper
- Truncate: Se encarga de realizar la función nativa Truncate
- Typeof: Se encarga de realizar la función nativa typeof



INSTRUCCIONES

En esta carpeta están todos los archivos que realizan unas ejecuciones de tipo instrucción

- Asignacion: Asigna variables
- Declaration: declara las variables del lenguaje
- Dowhile: realiza el dowhile del lenguaje
- For : realiza el for del lenguaje
- Función: realiza las funciones las declara y guarda
- If: realiza la ejecución if del lenguaje
- Print: realiza la ejecución print del lenguaje
- Statement: Aquí se realizan las instrucciones que están dentro de las funciones, if, else, while y dowhile
- Vector: Aquí se declaran y guardan los vectores
- While Aquí se realiza la ejecución while del lenguaje

