

Laboratorio de Organización de Lenguajes y  
Compiladores 1, Sección C



## PROYECTO 2 MFMScript

**MANUAL USUARIO**

FECHA: 04/11/2022

Juan Josue Zuleta Beb  
Carné: 202006353

## **Objetivos del Sistema**

El sistema está dirigido a un público general que desee interactuar con un pseudo parser capaz de transformar una entrada de código en salidas en consola, esperando la respuesta intuitiva sobre una aplicación sencilla pero potente que permite escribir código y esperar una salida funcional.

También se espera que pueda ser de utilidad para comprender mejor el funcionamiento del procesamiento de letras símbolos o cualquier otro carácter mediante la aplicación de análisis Lexico. Así como la comprensión del funcionamiento del análisis sintáctico por medio de una gramática de recursión por la izquierda, o mejor conocida como gramática recursiva por la izquierda.

Se desea dejar reflejado los conocimientos adquiridos en el curso de Organización de lenguajes y compiladores 1 entre los puntos recientemente visto sobre las primeras 2 fases de un compilador.

## Información del Sistema

Es una aplicación de interfaz gráfica que implementa el framework de Vue.js para el despliegue visual por medio de quasar y también el framework de Node.js para poder interconectar todas las partes de proyecto, en su núcleo esta programado en Typescript, así como complementado con javascript, cuenta con botones funcionales y con funciones desarrolladas especialmente para las especificaciones del proyecto.

Esta aplicación permite la lectura de comandos de entrada en formato de código similar a javascript y de la misma forma permite la salida de consola del código ejecutado.

Parte de las funciones es crear diferentes tipos de métodos, declaraciones y sentencias dando la sensación de estar escribiendo en un IDE normal, con la diferencia que este maneja un lenguaje único y particular, del cual se deben conocer las reglas léxicas y sintácticas para poder interactuar con el mismo.

## **Requisitos Mínimos del Sistema**

Sistema operativo 64 bits

- Microsoft Windows 10/8/7/Vista/2003/XP (incl.64-bit)
- macOS 10.5 o superior
- Linux GNOME o KDE desktop
- Procesador a 1.6 GHz o superior
- 1 GB (32 bits) o 2 GB (64 bits) de RAM (agregue 512 MB al host si se ejecuta en una máquina virtual)
- 3 GB de espacio disponible en el disco duro
- Disco duro de 5400 RPM
- Tarjeta de vídeo compatible con DirectX 9 con resolución de pantalla de 1024 x 768 o más.
- Navegador web (Recomendado: Google Chrome)

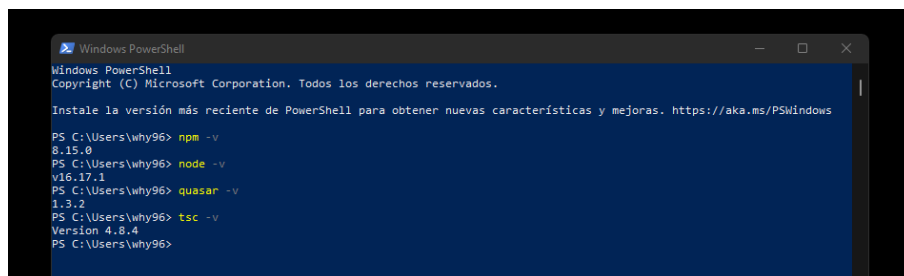
# BIENVENIDO A MFMScript – FIUSAC

## Inicialización del programa

Para poder acceder a la inicialización del programa se debe estar seguro de tener instaladas todas las librerías necesarias para el despliegue, las cuales son:

- Node.js
- npm
- Typescript
- Javascript
- Quasar

Una vez instaladas las librerías en el sistema deberá corroborar las versiones:



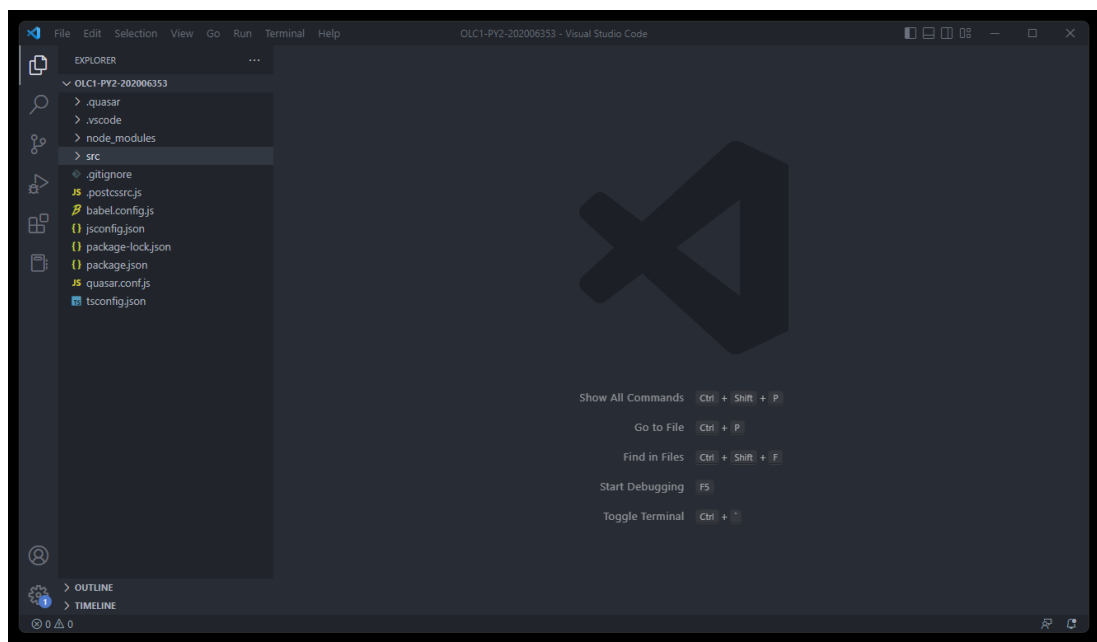
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

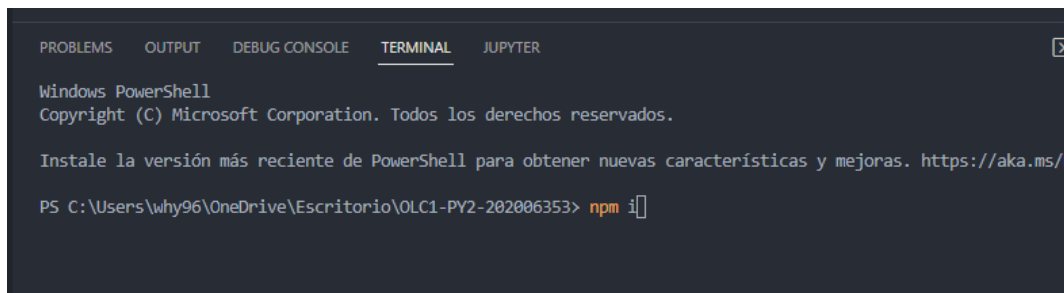
PS C:\Users\why96> npm -v
8.15.0
PS C:\Users\why96> node -v
v16.17.1
PS C:\Users\why96> quasar -v
1.3.2
PS C:\Users\why96> tsc -v
Version 4.8.4
PS C:\Users\why96>
```

*Nota: las versiones instaladas no influyen o entran en conflicto con el proyecto.*

Una vez instaladas las librerías y componentes, ingrese a visual studio code y abra el directorio del proyecto, una vez lo tenga abierto vera algo similar a esto:



Una vez aquí, abra una nueva terminal y proceda a ejecutar el siguiente comando:



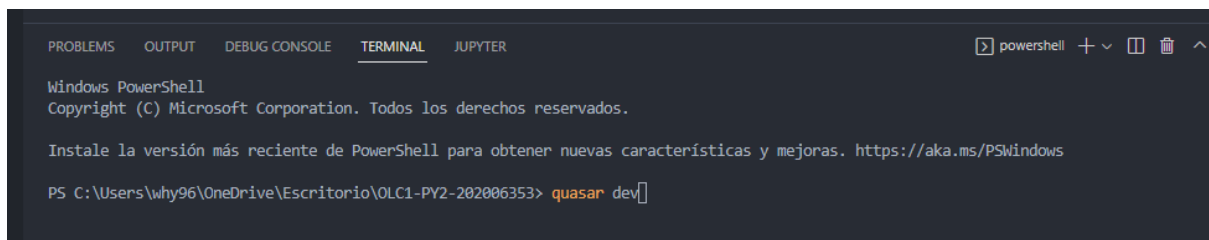
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PowerShellWindows

PS C:\Users\why96\OneDrive\Escritorio\OLC1-PV2-202006353> npm i
```

*Nota Importante: Evite realizar cambios en el código fuente únicamente ejecute el comando:*

Una vez instalada el entorno de Node.js tendrá instaladas las dependencias del mismo. Y podrá ejecutar el proyecto con el siguiente comando:



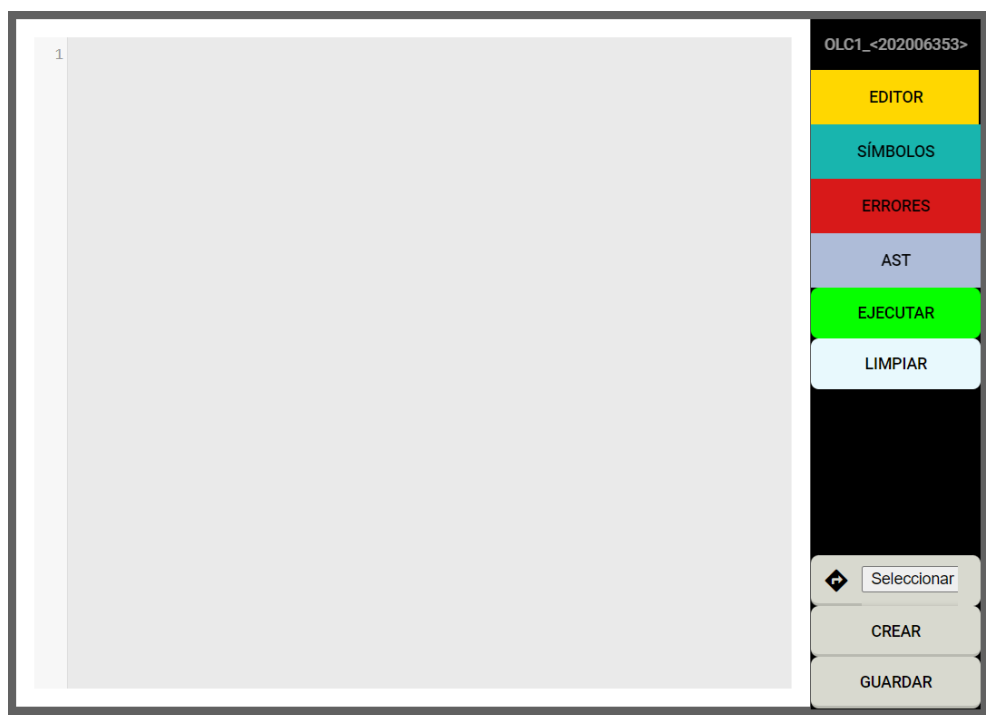
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell + v [icon] [icon] ^
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PowerShellWindows

PS C:\Users\why96\OneDrive\Escritorio\OLC1-PV2-202006353> quasar dev
```

Una vez ejecutado tendrá un despliegue en su navegador por defecto en el puerto 5000, mas específicamente la ruta seria: <http://localhost:5000/#/>

## Ventana del menú principal de la aplicación:



Función de las opciones se describe de la siguiente forma:

**VENTANA DE SIMBOLOS:**

En esta ventana podremos visualizar el reporte de símbolos generados en la ejecución del código, dichos símbolos pertenecen al lenguaje y son reconocidos exitosamente por el parser:

TABLA DE SIMBOLOS

VARIABLE <====> Id: variable\_1 <====> Tipo: null <====> Fila: 18 <====> Columna: undefined  
VARIABLE <====> Id: variable\_2 <====> Tipo: null <====> Fila: 19 <====> Columna: undefined  
VARIABLE <====> Id: variable\_3 <====> Tipo: null <====> Fila: 20 <====> Columna: undefined  
VARIABLE <====> Id: variable\_4 <====> Tipo: null <====> Fila: 21 <====> Columna: undefined  
VARIABLE <====> Id: variable\_5 <====> Tipo: null <====> Fila: 22 <====> Columna: undefined  
VARIABLE <====> Id: variable\_11 <====> Tipo: null <====> Fila: 18 <====> Columna: undefined  
VARIABLE <====> Id: variable\_22 <====> Tipo: null <====> Fila: 19 <====> Columna: undefined  
VARIABLE <====> Id: variable\_33 <====> Tipo: null <====> Fila: 20 <====> Columna: undefined  
VARIABLE <====> Id: variable\_44 <====> Tipo: null <====> Fila: 21 <====> Columna: undefined  
VARIABLE <====> Id: variable\_55 <====> Tipo: null <====> Fila: 22 <====> Columna: undefined  
VARIABLE <====> Id: var\_1 <====> Tipo: 1 <====> Fila: 23 <====> Columna: undefined  
VARIABLE <====> Id: tm1\_ <====> Tipo: null <====> Fila: 24 <====> Columna: undefined  
VARIABLE <====> Id: tmp2\_ <====> Tipo: null <====> Fila: 24 <====> Columna: undefined  
VARIABLE <====> Id: tmp3\_ <====> Tipo: null <====> Fila: 24 <====> Columna: undefined  
VARIABLE <====> Id: tmp4\_ <====> Tipo: 1 <====> Fila: 24 <====> Columna: undefined  
VARIABLE <====> Id: var\_2 <====> Tipo: 1 <====> Fila: 25 <====> Columna: undefined  
VARIABLE <====> Id: var\_2\_1 <====> Tipo: 1 <====> Fila: 26 <====> Columna: undefined  
VARIABLE <====> Id: var\_2\_2 <====> Tipo: 1 <====> Fila: 27 <====> Columna: undefined  
VARIABLE <====> Id: var\_2\_3 <====> Tipo: 1 <====> Fila: 28 <====> Columna: undefined  
VARIABLE <====> Id: var\_2\_4 <====> Tipo: 1 <====> Fila: 29 <====> Columna: undefined  
VARIABLE <====> Id: var\_3 <====> Tipo: 2 <====> Fila: 30 <====> Columna: undefined  
VARIABLE <====> Id: var\_3\_1 <====> Tipo: 2 <====> Fila: 31 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_1 <====> Tipo: 0 <====> Fila: 32 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_2 <====> Tipo: 0 <====> Fila: 33 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_3 <====> Tipo: 0 <====> Fila: 34 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_4 <====> Tipo: 0 <====> Fila: 35 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_5 <====> Tipo: 0 <====> Fila: 36 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_6 <====> Tipo: 0 <====> Fila: 37 <====> Columna: undefined  
VARIABLE <====> Id: var\_4\_7 <====> Tipo: 0 <====> Fila: 38 <====> Columna: undefined

OLC1\_<202006353>  
EDITOR  
SIMBOLOS  
ERRORES  
AST  
EJECUTAR  
LIMPIAR  
Seleccionar  
CREAR  
GUARDAR

Imagen con fines ilustrativos.

**VENTANA DE SIMBOLOS:**

En esta ventana podremos visualizar el reporte de errores generados en la ejecución del código, dichos errores no pertenecen al lenguaje o no son reconocidos exitosamente por el parser o pueden producirse errores a nivel sintáctico lo cuales también aparecerán en esta tabla:

Tabla de Errores

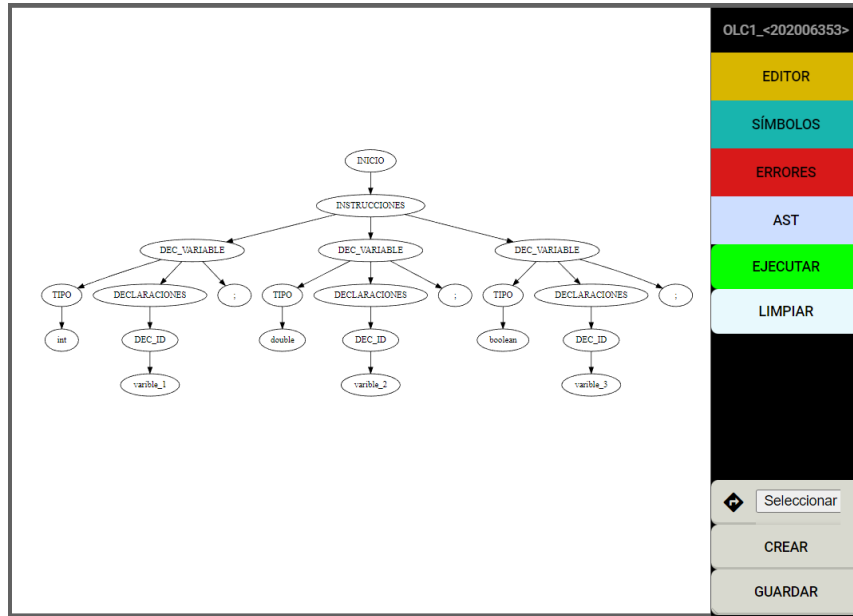
| #                  | Tipo       | Descripcion                       | Linea    | Columna |
|--------------------|------------|-----------------------------------|----------|---------|
| 1                  | lexico     | El valor "@" no es valido         | 2        | 1       |
| 2                  | sintactico | No se esperaba el token: "number" | 1        | 0       |
| Errores por página |            |                                   | 1-2 of 2 |         |

OLC1\_<202006353>  
EDITOR  
SIMBOLOS  
ERRORES  
AST  
EJECUTAR  
LIMPIAR  
Seleccionar  
CREAR  
GUARDAR

Imagen con fines ilustrativos.

## VENTANA DE AST:

En esta ventana podremos visualizar el reporte de AST generado en la ejecución del código, dichos AST es la representación sintáctica de la ejecución de nuestro código y es la representación lógica de la implementación de nuestro lenguaje:



*Imagen con fines ilustrativos.*

## BOTON EJECUTAR:

Envía al parser la entrada en condigo para que pueda ser reconocida por el analizador léxico y sintáctico y de cumplirse el formato del código devuelve una salida en consola con las especificaciones de los métodos y funciones.

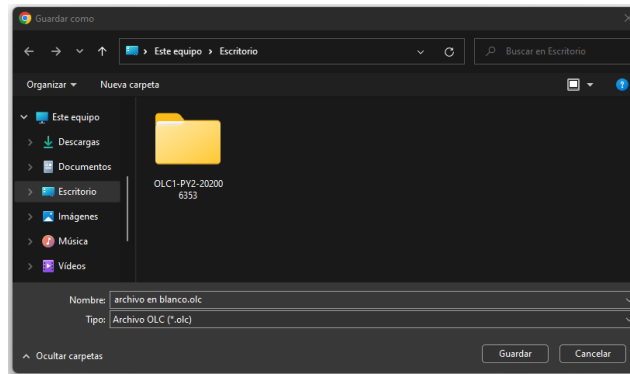
## BOTON LIMPIAR:

Limpia el área de trabajo para poder escribir en el área en limpio.

## SELECCIONAR ARCHIVO:

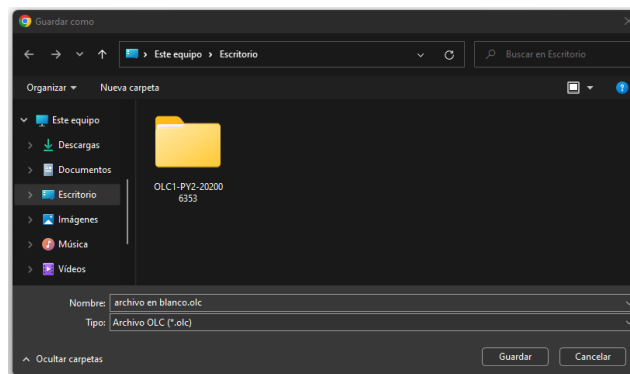
En estas opciones encontramos una forma de cargar un archivo al área de trabajo con la instrucción seleccionar. Una vez seleccionada esta opcion veremos una ventana semejante a esta:





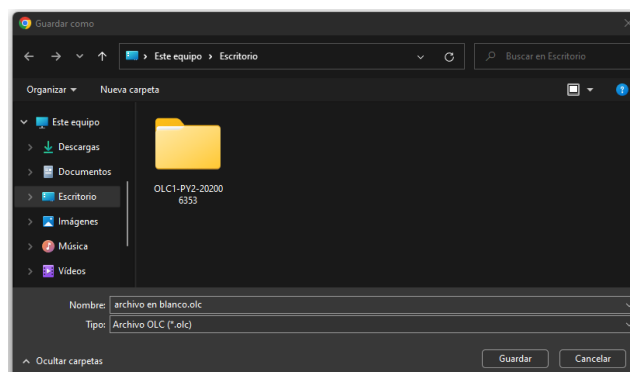
## CREAR ARCHIVO:

En estas opciones encontramos una forma de crear un archivo en blanco con la instrucción crear. Una vez seleccionada esta opción veremos una ventana semejante a esta:



## GUARDAR ARCHIVO:

En estas opciones encontramos una forma de Guardar un archivo con el estado actual del área de trabajo con la instrucción Guardar. Una vez seleccionada esta opción veremos una ventana semejante a esta:



Repositorio del proyecto: <https://github.com/BeKingGO/OLC1-202006353>