



**USAC**  
**TRICENTENARIA**  
Universidad de San Carlos de Guatemala

# Manual Técnico Proyecto 2

## Organización Lenguajes Compiladores 1

# Índice

Introducción .....	3
Objetivos.....	3
Conocimientos Previos .....	3
Requerimientos.....	3
Descripción de clases y funcionalidad .....	4

## Introducción

En este documento se describe los aspectos técnicos informáticos de la aplicación de escritorio, de manera que cualquier técnico informático pueda entender y comprender la lógica dentro del programa, para así poder darle mantenimiento y actualizarla si es necesario.

## Objetivos

Instruir al lector para el uso adecuado del código del programa, así mismo dar a conocer la lógica detrás del programa, de manera que se le facilite al lector la manipulación del código del programa.

## Conocimientos Previos

Los conocimientos que deberán tener las personas que manejen el programa son:

- Conocimiento de JavaScript, html, Jison y TypeScript.
- Manejo de gramaticas
- Manejo de servidor y react.
- Manejo de archivos Jison.
- Fases de un compilador.

## Requerimientos

- Netbeans 8.2 o superior
- 2gb de ram
- Intel Core 2 Duo 2
- windows xp, 7,8,10

## Descripción de clases y funcionalidad

Paquete	Clase	descripción
Código	Analizador.Jison	Contiene la gramática utilizada para poder ejecutar el programa con sus respectivas validaciones.
	Server.ts	Contiene la configuración del servidor que se esta utilizando para poder ejecutar el programa.
	Error.ts	Contiene los métodos para poder obtener los errores que se generen al momento de correr el programa, también contiene el código necesario para crear la tabla de errores.
	Aritmética.ts	Clase que contiene las validaciones para poder mostrar cualquier tipo de operación que contenga, enteros, booleanos, división, resta y multiplicación, también contiene el código para crear su AST.
	Lógica.ts	Contiene las validaciones para poder realizar operaciones con los operadores lógicos AND, OR y NOT.
	Relacional.ts	Contiene las validaciones para poder realizar operaciones con los símbolos relacionales que se solicitan, también contiene el código para generar su ast.
	Asignacion.ts	Contiene las validaciones para poder ejecutar diversas asignaciones que se realicen el momento de correr el programa.
	CicloFor.ts	Contiene las validaciones para poder ejecutar el programa al momento de utilizar un ciclo for, con las instrucciones que se le premitan.
Codigo	Declaración.ts	Clase encargada de las validaciones para poder ejecutar las declaraciones solicitadas al momento de correr el programa.
Codigo	DoUntil	Contiene las validaciones para poder generar el ciclo Do until según el usuario lo necesite.
	DoMientras.ts	Contiene las validaciones para poder generar el ciclo Do while según el usuario lo necesite.
	Funciones.ts	Contiene las validaciones necesarias para poder mostrar las funciones que se ejecuten en el programa.

Paquete	Clase	descripción
Código	lfinst.ts	Contiene las validaciones para poder ejecutar el ciclo if, else if, y varios casos de else dentro del ciclo if.
	imprimir.ts	Contiene la configuración del servidor que se esta utilizando para poder ejecutar el programa.
	incredmeto.ts	Contiene los métodos para poder obtener los errores que se generen al momento de correr el programa, también contiene el código necesario para crear la tabla de errores.
	LlamarFunciones.ts	Clase que contiene las validaciones para poder mostrar cualquier tipo de operación que contenga, enteros, booleanos, división, resta y multiplicación, también contiene el código para crear su AST.
	Metodo.ts	Contiene las validaciones para poder realizar operaciones con los operadores lógicos AND, OR y NOT.
	Mientras.ts	Contiene las validaciones para poder realizar operaciones con los símbolos relacionales que se solicitan, también contiene el código para generar su ast.
	NewInstructions.ts	Contiene las validaciones para poder ejecutar diversas asignaciones que se realicen el momento de correr el programa.
	Vectortipo1.ts	Contiene las validaciones para poder ejecutar el programa al momento de utilizar un ciclo for, con las instrucciones que se le premitan.
Código	Declaración.ts	Clase encargada de las validaciones para poder ejecutar las declaraciones solicitadas al momento de correr el programa.
FrontEnd	Árbol.js	Contiene método que permite mostrar el AST en la pagina principal del programa al momento de ejecutarlo.
	EditorCodigo.js	Contiene método que permite mostrar la interfaz de interacción del programa con el usuario.
FrontEnd	NavBar.js	Contiene la barra de navegación principal de la interfaz.
	App.css	Contiene el código para mostrar una interfaz amigable al usuario.
	Index.js	Contiene la interfaz principal de el programa.