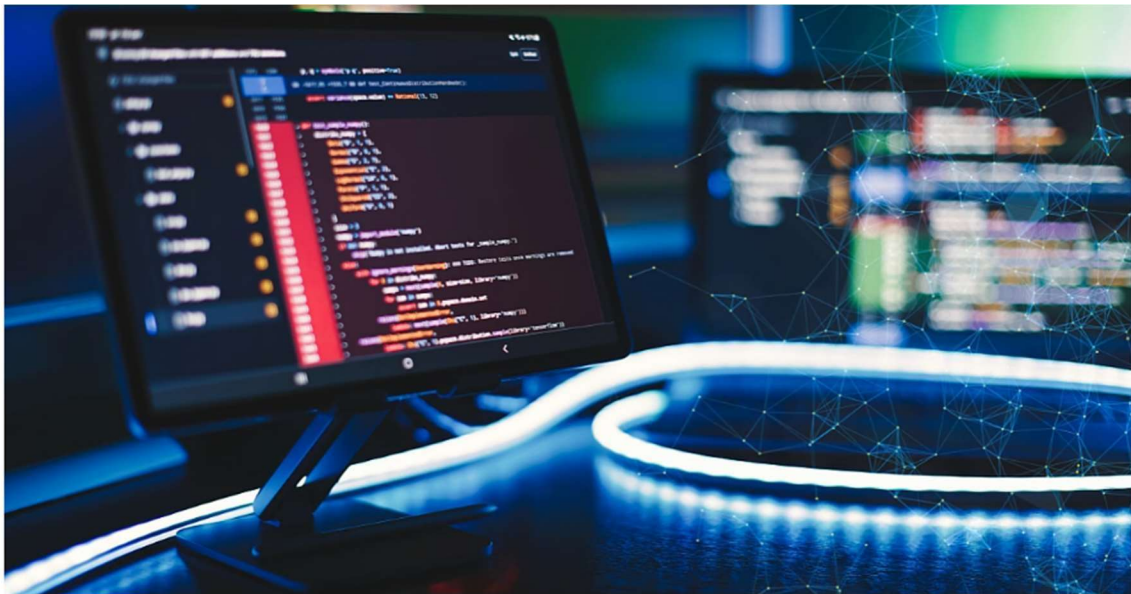


# Manual Técnico

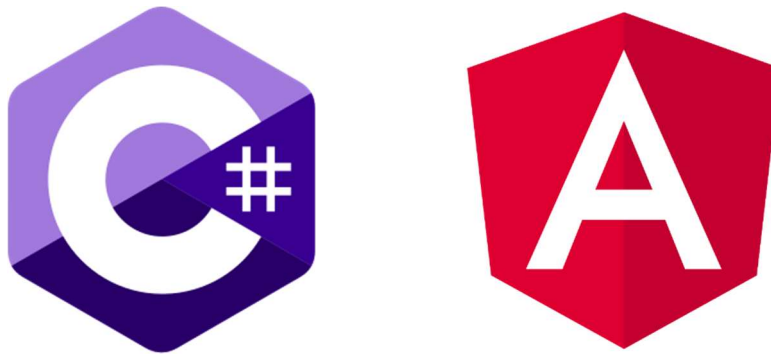


**Organización de Lenguajes y Compiladores 2**

**Francisco Daniel Peruch de León**

**Carné: 202100639**

Este proyecto consiste en una aplicación web la cual fue desarrollada usando el lenguaje typescript para el lado del backend y angular para el frontend. Esta aplicación tiene la finalidad de recibir una cadena de texto la cual puede ser ingresada por medio del cuadro de texto o cargando un archivo. Dicha cadena de texto es a la cual se le realiza el análisis para mostrar los respectivos resultados.



## Carpeta Analizador:

En esta carpeta se guardaran varias clases que fueron utilizadas a lo largo del programa para realizar el análisis correspondiente. A continuación se detalla el contenido de esta carpeta.

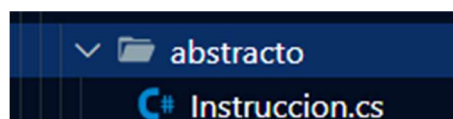
## Carpeta abstracto:

### Clase Instruccion:

Esta es una clase abstracta, de la cual heredaran las demás instrucciones y expresiones que se utilicen durante el proyecto. Los método y atributos que esta clase contine son los siguientes.

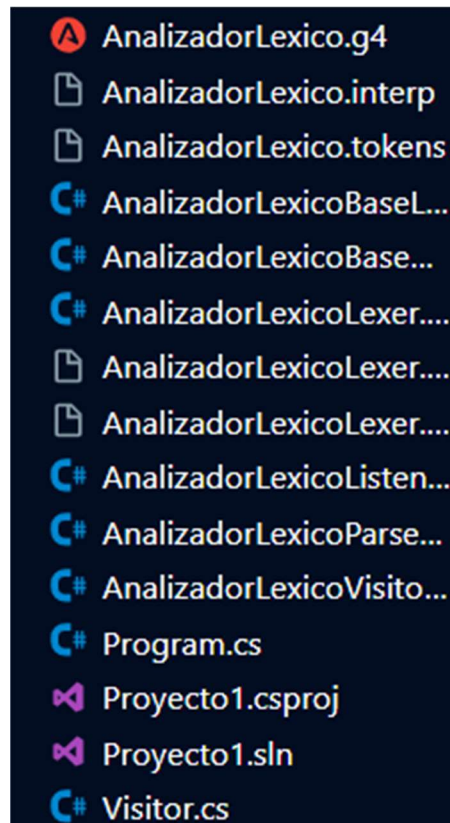
Atributos: Cada uno de los objetos de esta clase y los hereden de esta esta clase tendrán los atributos de tipo, linea y col.

Métodos: En esta clase se utiliza un método llamado Interpretar el cual será útil para los objetos de las clases que hereden de Instrucción debido a que por medio de esta se podrán realizar las instrucciones que sean requeridas.



## Clase gramatica-generada:

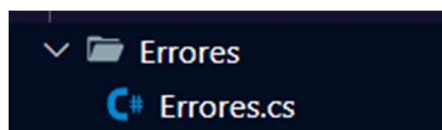
Esta clase fue la que se creo al utilizar jison, debido a que en jison se formulo la gramática correspondiente al proyecto, esta clase es la que generar un parser por medio del cual se puede llevar el análisis de una cadena que ingrese.



## Carpeta excepciones:

### Clase Errores:

Esta clase se utiliza para guardar información acerca de los errores encontrados durante el análisis. En esta se encuentra los siguientes atributos: tipo, desc, linea y columna. En los cuales se guardarán los datos referentes a los diferentes errores.



## Carpeta expresiones:

### **Clase AccesoVar:**

Esta clase hereda de la clase abstracta Instruccion, contiene los atributos y métodos utilizados para poder acceder al valor de una variable determinada. Esta tiene como atributos el valor de id, el cual será útil para encontrar el valor de la variable en cuestión.

Entre los métodos que esta contiene se encuentra el método Interpretar el cual recibe como parámetros los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos, por medio del atributo id se realiza una búsqueda dentro del parámetro tabla la cual devolverá el valor de la variable por medio de su método getVariable.

### **Clase Aritmeticas:**

Esta clase hereda de la clase abstracta Instruccion, esta contiene los atributos y métodos necesarios para realizar las operaciones aritméticas que sean requeridas. Esta tiene como atributos los valores de operando1, operando2 y operandoUnico que son de la clase Instrucción, además de operación de la clase OperadoresAritmeticos. Esta clase tiene dos constructores debido a que dependiendo de la operación que se necesita llevar a cabo, se necesita inicializar solo el atributo de valorUnico o los valores de operando1 y operando2. Entre los métodos que esta clase tiene se encuentran los siguientes:

interpretar: En este método se verifica cual de los atributos fue inicializado y si estos no son errores. Luego dependiendo del valor que contenga el atributo operación se envían los parámetros correspondientes al método correspondiente a la operación aritmética que se desea realizar.

- suma: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para realizar la suma. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la suma de cierta forma o no se realizará si los tipos no son los admitidos.
- resta: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para realizar la resta. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la resta de cierta forma o no se realizará si los tipos no son los admitidos.

- **multiplicacion:** Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para realizar la multiplicación. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la multiplicación de cierta forma o no se realizará si los tipos no son los admitidos.
- **division:** Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para realizar la división. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la división de cierta forma o no se realizará si los tipos no son los admitidos.
- **potencia:** Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para realizar la potencia. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la potencia de cierta forma o no se realizará si los tipos no son los admitidos.
- **modulo:** Este método recibe dos parámetros de tipo Object los cuales contendrán los valores del operando izquierdo y el operando derecho que se utilizarán para obtener el módulo de ambos números. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación de módulo de cierta forma o no se realizará si los tipos no son los admitidos.
- **negación:** Este método recibe un parámetro de tipo Object el cual contendrá el valor del operandoUnico que se utilizará para obtener la negación del número. Para esto se verificará el tipo y dependiendo de su tipo se realizará la operación de multiplicarlo por -1 o no se realizará si el tipo no es el admitido.





## **Clase Nativo:**

Esta clase hereda de la clase abstracta Instruccion, esta clase tiene como atributo la variable valor y el método interpretar el cual devolverá el atributo valor.

## **Relacionales:**

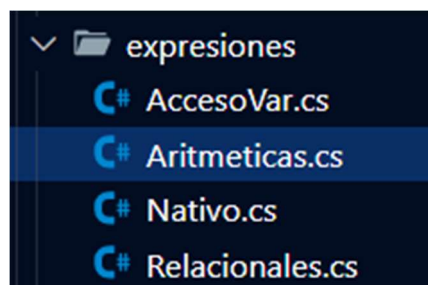
Esta clase hereda de la clase abstracta Instruccion, esta contiene los atributos y métodos necesarios para realizar las operaciones relacionales que sean requeridas. Esta tiene como atributos los valores de cond1, cond2 y condicionUnica que son de la clase Instrucción, además de relacional de la clase OperadoresRelacionales. Esta clase tiene dos constructores debido a que dependiendo de la operación que se necesita llevar a cabo, se necesita inicializar solo el atributo de condicionUnica o los valores de cond1 y cond2. Entre los métodos que esta clase tiene se encuentran los siguientes:

- interpretar: En este método se verifica cual de los atributos fue inicializado y si estos no son errores. Luego dependiendo del valor que contenga el atributo relacional se envían los parámetros correspondientes al método correspondiente a la operación relacional que se desea realizar.
- equals: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si son iguales y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o no se realizará si los tipos no son los admitidos.
- diferente: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si son diferentes y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o no se realizará si los tipos no son los admitidos.
- mayor: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si la condicion de la izquierda es mayor a la de la derecha y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o



no se realizará si los tipos no son los admitidos.

- menor: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si la condicion de la izquierda es menor a la de la derecha y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o no se realizará si los tipos no son los admitidos.
- mayorIgual: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si la condicion de la izquierda es mayor o igual a la de la derecha y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o no se realizará si los tipos no son los admitidos.
- menorIgual: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores de la condición de la izquierda y la condición de la derecha que se utilizarán para realizar la operación para verificar si la condicion de la izquierda es menor o igual a la de la derecha y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo de sus tipos se realizará la operación o no se realizará si los tipos no son los admitidos.
- andCondicion: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores que se utilizarán para realizar la operación AND entre ellas y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo si son de tipo booleano se realizará la operación o no se realizará.
- orCondicion: Este método recibe dos parámetros de tipo Object los cuales contendrán los valores que se utilizarán para realizar la operación OR entre ellas y retornar un valor de tipo booleano. Para esto se verificarán los tipos de cada uno y dependiendo si son de tipo booleano se realizará la operación o no se realizará.
- notCondicion: Este método recibe un parámetro de tipo Object el cual contendrá el valor que se utilizará para realizar la operación de negación. Para esto se verificará el tipo y dependiendo si es de tipo booleano se realizará la operación o no se realizará.





## Carpeta Instrucciones:

### **Clase AsignacionVar:**

Esta clase hereda de la clase abstracta Instruccion, tiene como atributos los valores de id que contendrá el nombre de la variable y exp de la clase Instruccion la cual contendrá el valor que se desea asignar a la variable.

En esta clase se encuentra el método interpretar, el cual recibe como parámetros los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos. Usando el atributo id se busca en el parámetro recibido tabla, en este se verifica si la variable con ese nombre existe sino se devuelve un error. Luego se utiliza el atributo exp para obtener el valor que se desea asignar a la variable, después de validar que tanto la variable como el valor a asignar son del mismo tipo y que la variable tiene mutabilidad se procede a asignarle el nuevo valor a la variable en cuestión.

### **Declaracion:**

Esta clase hereda de la clase abstracta Instruccion, esta tiene como atributos los valores de identificador y mutabilidad de tipo string además de valor de la clase Instruccion. En esta clase se encuentra el método interpretar por medio del cual se le asigna un valor a una nueva variable.

Para esto se utiliza el atributo valor para obtener el valor que se desea asignar a la variable, se verifica que este no sea un error. Luego se verifica que tanto la variable como el valor a asignar son del mismo tipo. Después se crea un objeto de la clase Símbolo el cual será contendrá los datos necesarios de la variable y se guarda ese objeto en el parámetro tabla de la clase tablaSimbolos, de último se hace una verificación de que la variable se haya guardado en la tabla de símbolos sino se devuelve un error.

### **Decremento:**

Esta clase hereda de la clase abstracta Instruccion, tiene como atributos los valores de id que contendrá el nombre de la variable que se desea incrementar su valor.

En esta clase se encuentra el método interpretar, el cual recibe como parámetros los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos. Usando el atributo id se busca en el parámetro recibido tabla, en este se verifica si la variable con ese nombre existe sino se devuelve un error. Luego se verifica si la variable es de tipo INT o DOUBLE para restarle a su valor actual la cantidad de 1, luego se verifica si la variable tiene mutabilidad. Sí tiene, entonces se le actualiza el valor a esa variable de lo contrario se devuelve un error.

### **For:**

Esta clase hereda de la clase abstracta Instrucción, tiene como atributos asignación, condición y actualización de la clase Instrucción e instrucciones que es una linkedlist de la clase Instruccion.

En esta clase se encuentra el método Interpretar, el cual recibe como parámetro los valores de arbol de la clase Arbo y tabla de la clase tablaSimbolos. En este método se crea una variable llamada newTable que será la nueva tabla de símbolos correspondiente a este ciclo. Después se crea un variable llamada res1 que contendrá el valor del atributo asignacion. Luego se realiza la verificación de que el atributo condicion es de tipo Booleano sino se devuelve un error. Luego dentro de un ciclo while se tiene como condición de salida a el atributo condicion, el cual se obtendrá su valor cada vez que se termine una iteración. Dentro del ciclo while se creará una nueva tabla de símbolos llamada newTable2. Luego se procederá a ejecutar a cada una de las instrucciones que contenga el ciclo for. Luego de terminar con las instrucciones, se obtiene el valor del atributo actualización.

### **If:**

Esta clase hereda de la clase abstracta Instrucción, tiene como atributos a condición de la clase Instrucción e instrucciones que es una linkedlist de la clase Instruccion.

En esta clase se encuentra el método interpretar el cual recibe como parámetro los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos. En este método se verifica que el atributo condicion no sea un error y que sea de tipo booleano. Luego se crea una nueva tabla de símbolos llamada newTabla de la clase tablaSimbolos. Luego, si la condicion es true, se procede a realizar todas las instrucciones que se guardan el atributo del mismo nombre.

### **Incremento:**

Esta clase hereda de la clase abstracta Instruccion, tiene como atributos los valores de id que contendrá el nombre de la variable que se desea incrementar su valor.

En esta clase se encuentra el método interpretar, el cual recibe como parámetros los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos. Usando el atributo id se busca en el parámetro recibido tabla, en este se verifica si la variable con ese nombre existe sino se devuelve

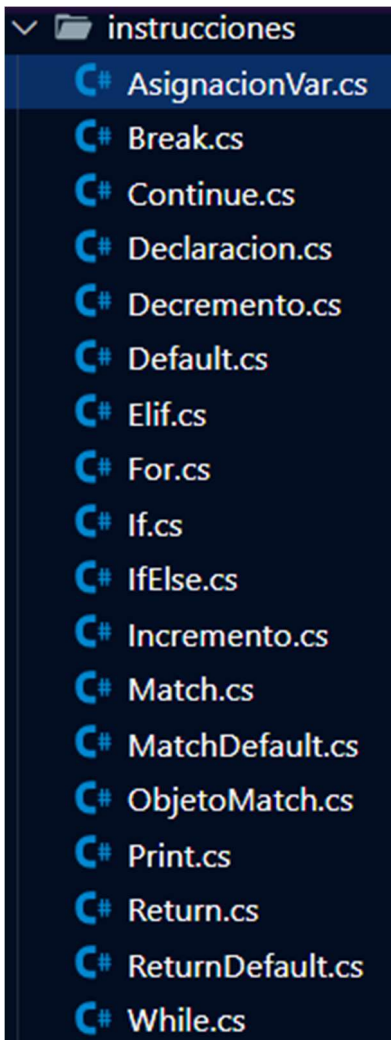
un error. Luego se verifica si la variable es de tipo INT o DOUBLE para sumarle a su valor actual la cantidad de 1, luego se verifica si la variable tiene mutabilidad. Sí tiene, entonces se le actualiza el valor a esa variable de lo contrario se devuelve un error.

### **Print:**

Esta clase hereda de la clase abstracta Instrucción, posee un atributo expresion de la clase Instruccion. En esta clase se encuentra el método interpretar el cual recibe como parámetro los valores de arbol de la clase Arbol y tabla de la clase tablaSimbolos. En este método se verifica que el atributo expresion no sea un error, luego se procede a utilizar el método Print del parámetro arbol, por medio del cual se envia la expresion que quiere mostrar en consola al realizar el análisis.

### **Llamada:**

En esta clase se maneja la llamada de funciones o método, debido a que el un objeto de tipo Método será diferente a uno de tipo Función, entonces al interpretar un objeto de la clase llamada. Este podrá buscar la función o método y ejecutar las instrucciones que esta guarde.



## Carpeta Símbolo:

### Clase Arbol:

En esta clase tiene como atributos a instrucciones que es una linkedlist que contiene valores de la clase Instrucción, tablaGlobal de la clase tablaSimbolos y errores que es una linkedlist que contiene valores de la clase Errores. En esta clase se encuentra diferentes métodos getter y setters que son los encargados de asignarle un valor a un atributo u obtener el valor de estos.

### Clase Símbolo:

En esta clase tiene como atributos a tipo de la clase Tipo, id y valor de tipo String y valor de tipo Object. En esta clase se encuentra diferentes métodos getter y setters que son los encargados de asignarle un valor a un atributo u

obtener el valor de estos.

### **Clase Tipo:**

Esta clase tiene como atributo a tipo de la clase enumerada tipoDato. En esta clase se encuentra diferentes métodos getter y setters que son los encargados de asignarle un valor a un atributo u obtener el valor de estos.

### **Clase tablaSimbolos:**

Esta clase tiene como atributos a tablaSimbolos de la clase tablaSimbolos, nombre de tipo String y tablaActual que es un HashMap. En esta clase se encuentra diferentes métodos getter y setters que son los encargados de asignarle un valor a un atributo u obtener el valor de estos. Además del método setVariable que sirve para guardar una variable dentro de la tabla de símbolos y getVariable que sirve para buscar el valor de la variable en la tabla de símbolos y devolverlo.

