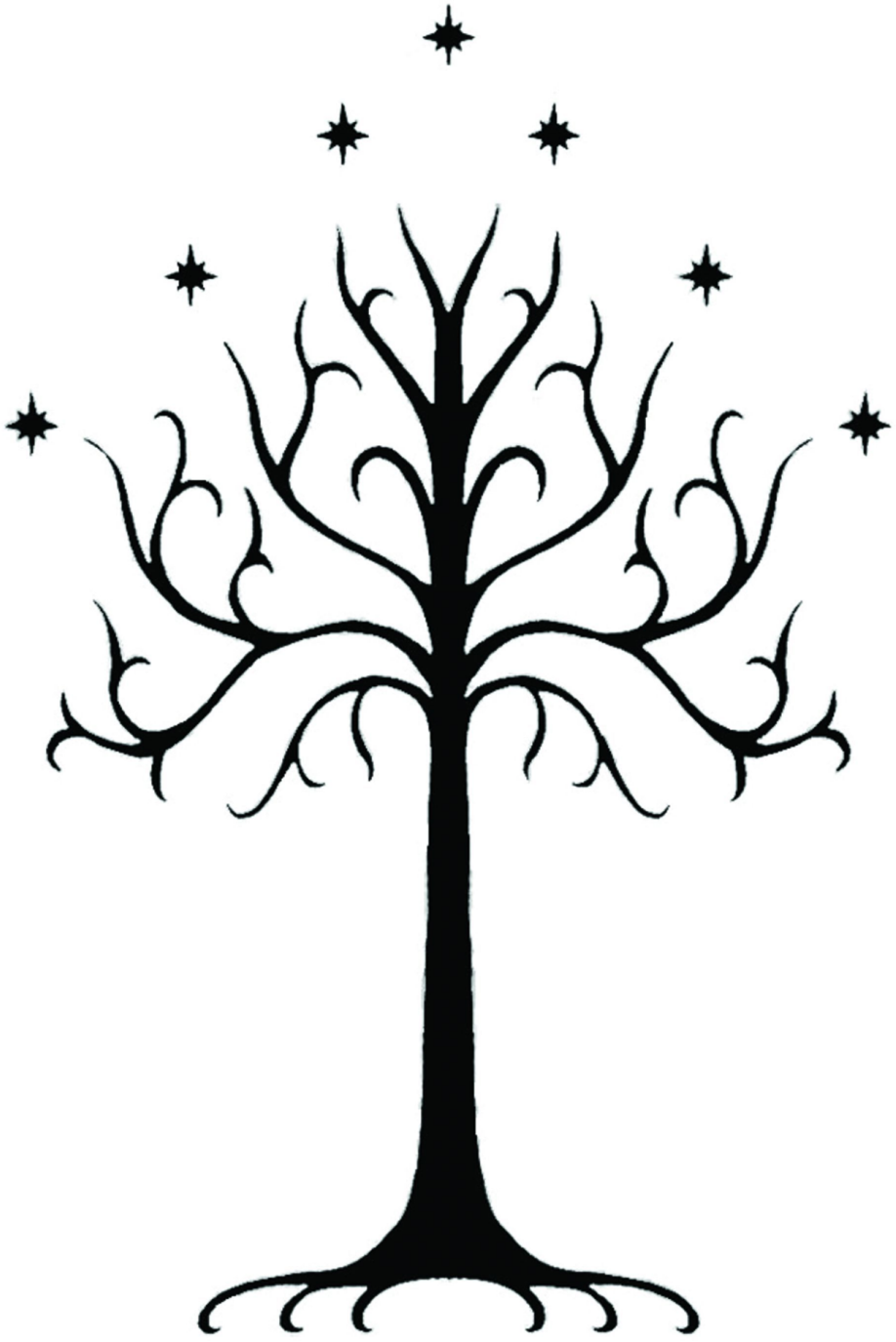
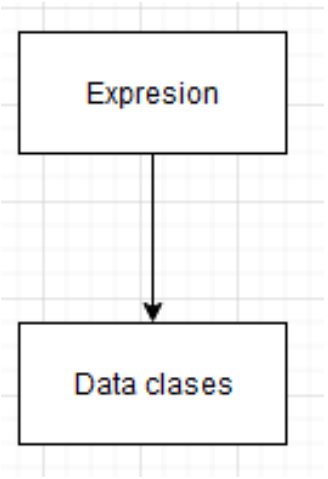
Anthony Samuel Ordoñez Son 201931185

# Manual técnico, Proyecto 2



Para el correcto funcionamiento del programa se usó un objeto de tipo *Expresión,* el cual sirvió de pivote para poder crear el resto de clases, todas las clases heredan de Expresión, y el resultado final del árbol sintáctico es una lista de expresiones.



Se crearon clases para cada tipo de sentencia posible en la gramática, de esta manera se pudo guardar la información de cada sentencia, conservando su valor y teniendo disponibilidad de manipular la manera en la que se muestra, permitiendo así la traducción del código.

## Clases creadas:

Asignación

BreakContinue

Case

Comentario

Condición

Console Log

Declaracion

DeclaracionFor

DoWhile

Else

Error

Expresion

For

Funcion

ListaCasos

ListaCondiciones

ListaExpresiones

ListaParametros

ListaString

ListaValores

Llamada

Parametro

Return

Simbolo

Switch

Tipo

Valor

ValorCondicion

While

Cada una de estas clases tiene el único propósito de almacenar información de algún tipo de dato o sentencia para poder ser mostrado o usado en el futuro;

## Gramática

Para la gramática se utilizó el analizador sintáctico CUP de tipo LALR, al ser un analizador ascendente, la gramática va formándose desde el fondo y se devuelve la raíz, el resultado de la raíz es una lista de expresiones. Al ser indefinido el tipo de expresión que vendrá, es importante que todas las clases hereden la Expresion.

La clase MainFrame.java, sirve a manera de frontend. Todas las operaciones pasan en el parser y en las clases, esta clase solo se encarga de recopilar información y mostrarla en pantalla.

## Tabla de símbolos

La tabla de símbolos fue creada a partir de un arreglo de expresiones, en un caso ideal, los valores se habrían encontrado en el mismo entorno en el que se requieren, por lo que cada rama del árbol tendría su propia sub tabla de símbolos, en esta caso se optó por una solución más simple, se utilizó una lista de valores, cada vez que en la gramática se encuentra un valor, se guarda en la tabla de símbolos para su futuro uso.

Anthony Samuel Ordoñez Son 201931185

# ÁRBOL AST

Para crear el árbol AST se usó la ventaja de haber guardado todo en objetos desde el principio, en este caso, cada objeto puede contener uno o más objetos a manera de sub nodos y cada subnodo funciona de igual manera, de modo que las ramas se crean al momento de la ejecución del programa.

El objeto *Raiz,* se encarga de recopilar esos objetos haciendo usó de recursividad, el resultado final es un árbol nacido del funcionamiento del parser.