

Reporte Fase 4: Front-End

Constanza Abarca 13-10000

Pedro Maldonado 13-10790

En esta fase del proyecto se recorrió nuevamente el AST extendido modificado en la Fase 3. Se creó la clase *evaluacion* que contiene una tabla de símbolos como atributo.

Se creó la función **evalArbol(AST)** para recorrer recursivamente los hijos del AST y realizar verificaciones del tipo del nodo y dependiendo de éste se interpreta la pieza de código correspondiente en BasicTran.

Para la interpretación de cada tipo de operación se utilizaron funciones, herramientas y operadores de Python3.

Para obtener los valores de las expresiones correspondientes, se utilizó la función **evalExp(exp)** que recibe una expresión, verifica su tipo y dependiendo de este, se extraen los operadores de forma recursiva hasta llegar al valor terminal y si es necesario se realizan las respectivas operaciones ya sean aritméticas, booleanas, relacionales, con arreglos y se reporta el error correspondiente a las operaciones aritméticas con la división por cero.

En algunos casos (Término y Variable de Arreglos) se utilizó la función **getValor(var, index)** que verifica y devuelve el valor correspondiente de la variable en la tabla de símbolos. Se reportan los errores correspondientes: Variables no inicializadas, indexación negativa, errores en arreglos de arreglos.

Para el caso de las asignaciones y variables se definió la función **setValor(var, val, index)** que ingresa un valor en la tabla de símbolos de la clase evaluación para la

respectiva variable y reporta los errores correspondientes a las asignaciones. Para el caso de los arreglos, estos son inicializados con valores None en cada posición. Para agregar elementos en los arreglos únicamente es posible mediante la asignación.

Una vez interpretadas todas las operaciones con las respectivas funciones es posible concluir que finalizó el proceso de desarrollo e interpretación de BasicTran.