

# Proyecto Final Compiladores

## Lenguaje Treebi

José Miguel Guzmán Chauca  
Compiladores  
Ciencia de la Computación  
Universidad Católica San Pablo  
Arequipa, Perú  
Email: jose.guzman@ucsp.edu.pe

### I. PROPÓSITO DEL LENGUAJE

Treebi es un lenguaje de dominio específico cuyo objetivo es la manipulación de árboles binarios de búsqueda (interacciones, visualizaciones, etc.). El alcance de Treebi incluye actualizaciones de datos (inserción, borrado), consultas y exportaciones de imágenes.

### II. TIPOS DE DATOS PERMITIDOS

- Caracteres: string (Los string comienzan y terminan con comillas simples y deben estar contenidos en una línea).
- Números: int, float
- Booleanos: bool
- Específico: tree

### III. IDENTIFICADORES

Los nombres que se deberían utilizar como identificadores son los mismos que se utilizan en 'C' o 'C++'.

### IV. PALABRAS RESERVADAS

Estas son las palabras reservadas que se utilizarán:

- string
- int
- float
- bool
- tree
- if
- else
- for
- avl
- simple

### V. OPERADORES

- Operador = : asignación básica de variables (caracteres, números y booleanos de la forma básica, para tree la asignación crea un árbol de un tipo predeterminado (simple) con el valor asignado)
- Operador + : adición básica
- Operador - : sustracción básica
- Operador \* : multiplicación básica
- Operador / : división básica
- Operador < : Menor
- Operador > : Mayor

- Operador <= : Menor o igual
- Operador >= : Mayor o igual
- Operador != : Diferencia
- Operador == : Igualdad
- Operador ++ : adición de un elemento al árbol.
- Operador -- : sustracción de un elemento del árbol.
- Operador === : Igualdad de árboles
- Operador !== : Diferencia de árboles
- Operador % : fusión de los elementos de dos árboles distintos en un árbol.
- Operador . : Selección del tipo de árbol (AVL o Simple)

### VI. ESTRUCTURAS DE CONTROL

- Secuencial: Cada una de las instrucciones serán separadas por el carácter “”.
- Condicional (Instrucción if) : instrucción que hace que se ejecuten unas sentencias u otras dependiendo del valor que toma una condición.
- Repetición (Instrucción for): Un bucle que hace que una instrucción o bloque de instrucciones se repitan un número determinado de veces mientras se cumpla la condición.

### VII. FUNCIONES

- eliminar\_arbol: Elimina el árbol correspondiente en memoria.
- visualizar: Presenta una imagen con el árbol correspondiente.
- exportar: Genera una visualización con el árbol correspondiente que se descarga al ordenador.

### VIII. COMENTARIOS

Los comentarios comienzan con “” y abarcan toda la línea desde el punto donde comienzan.

### IX. DELIMITADORES

El delimitador utilizado en cada línea del código será el salto de línea y marcará la orden para procesar la siguiente línea de código.

### X. REPOSITORIO

El código fuente al proyecto se encuentra en:  
[https://github.com/Xider1701/Treebi\\_Compiler.git](https://github.com/Xider1701/Treebi_Compiler.git)

## XI. GRAMÁTICA

```
stmt -> assign | calc | calc_a | func | ifstmt | forstmt
assign -> type id = lit | id.category | id % id
type -> string | int | float | bool | tree
lit -> number_literal | string_literal
category -> avl | simple
calc -> term expr
expr -> + term expr | - term expr |
term -> factor s_term
s_term -> * factor s_term | / factor s_term |
factor -> number_literal | id
cond -> factor_c < factor_c | factor_c > factor_c |
      factor_c <= factor_c | factor_c >= factor_c
factor_c -> id | number_literal
cond_i -> factor_i == factor_i | factor_i != factor_i
factor_i -> id | number_literal | string_literal
calc_a -> term_a expr_a
expr_a -> ++ term_a expr_a | -- term_a expr_a |
term_a -> number_literal | string_literal | id
cond_a -> id === id | id !== id
func -> eliminar_arbol ( id ) | visualizar ( id ) | exportar ( id )
list_c -> cond | cond_i | cond_a
ifstmt -> if ( list_c ) { stmt } else { stmt }
forstmt -> for ( int id = number_literal ; list_c ; calc ) { stmt }
```