



# MANUAL TÉCNICO

Universidad San Carlos de Guatemala  
Organización de Lenguajes y Compiladores 2  
Pablo Josué Barahona Luncey  
202109715  
3560855890101

PROYECTO 2

# Acerca del programa

---

Este programa fue desarrollado con el lenguaje de programación Go como backend, utilizando también React para su interfaz gráfica en frontend. También se usó ANTLR, la cual es una herramienta para crear la gramática

Este programa cuenta con diferentes clases (explicación en las siguientes páginas) y el uso de librerías nativas de Go

El objetivo principal del programa es crear un intérprete del lenguaje de programación llamado Swift

# Backend

---

## Archivo main

Este es el archivo principal, este maneja las funciones de la interfaz y el uso de los botones. También redirige a las otras clases para iniciar el analizador.

## Archivo SwiftGrammar.g4 y SwiftLexer.g4

En este archivo se tienen establecidas las gramáticas a usar en el analizador. También acá mismo se hacen las expresiones regulares y todo lo relacionado al analizador

## 5 carpetas principales:

Se tienen 4 carpetas principales las cuales tienen los archivos necesarios para la ejecución

## environment

Esta carpeta se usa para manejar los entornos de ejecución, contiene el archivo AST, Environment, entre otros, los cuales sirven para el manejo de errores, tabla de símbolos, guardar los símbolos y los métodos.

- Existen símbolos para funciones y símbolos generales
- Se tiene acá la estructura del Struct, con su contenido y tipo
- El environment posee la ejecución de manejo de funciones y variables
- El AST maneja el print y los errores

# Backend

---

## expressions

En esta carpeta se manejan las expresiones. las cuales son necesarias para las instrucciones y estas permitan hacer la ejecución de forma correcta

- Acá se declaran las operaciones como suma y resta
- Se manejan los arreglos y vectores
- Se tienen métodos de expresión tales como count o empty
- Es importante porque todos retornan un valor ya sea entero, string o float

## instructions

En estos archivos usamos las expresiones y las ejecutamos, son importantes para convertir la información en una acción y esa acción ejecutarla.

- Se tiene el print que va a imprimir las expresiones
- Usamos acá las funciones e instrucciones como While, For, If, entre otras
- No retornamos nada ya que es una instrucción, solo enviamos los parámetros a environment con el print
- Podemos editar variables o declararlas

## interfaces

Este posee solo 2 archivos el cual maneja los parámetros que se deben de usar para las expresiones o instrucciones, en instrucciones usamos `interface{}` y en expresiones `environment.symbol`

# Backend

---

## generator

En esta carpeta manejamos un archivo generador de c3d, este archivo contiene los elementos importantes para el manejo del código. Tiene métodos los cuales se llaman y van aumentando y generando los arreglos temporales y las etiquetas necesarias para ello

# Frontend

Esta es la cara del programa, es el archivo frontend, el cual está creado con React JS, se utilizan diferentes librerías nativas de React, con el botón Run hacemos una petición al backend y este nos hace un return con el código ya analizado. Finalmente, la interfaz se muestra así:

## T-Swift

UPLOAD

CST

ERRORES

TABLA DE SÍMBOLOS

Input Code

```
1 func printArrayStr(_ arr: [String]) {
2   var out = ""
3   for i in 0...4 {
4     if i == (4) {
5       out += arr[i]
6     } else {
7       out += arr[i] + ", "
8     }
9   }
10  out += "]"
11  print(out)
12 }
13
14 func printArrayInt(_ arr: [Int]) {
15   var out = ""
16   for i in 0...4 {
17     if i == 4 {
18       out += String(arr[i])
19     } else {
20       out += String(arr[i]) + ", "
21     }
22   }
23   out += "]"
24   print(out)
25 }
26
27 func imprimirPunteo(_ ests: [String], _ cursos: [String], _ notas: [[[Int]]]) {
28   let ponderacion = ["1er parcial", "2do parcial", "3er parcial", "Lab",
29     "Examen final"]
30   printArrayStr(ponderacion)
```

Console Output

```
1
2 Imprimir notas por curso y estudiantes
3
4 [1er parcial, 2do parcial, 3er parcial, Lab, Examen final]
5
6 Estudiante: Lorena
7 Curso: Arquí
8 [37, 49, 61, 29, 44]
9 Curso: Archivos
10 [56, 60, 51, 68, 70]
11 Curso: Compl2
12 [47, 15, 39, 17, 74]
13
14 Estudiante: Rosendo
15 Curso: Arquí
16 [69, 74, 52, 34, 36]
17 Curso: Archivos
18 [24, 44, 50, 18, 76]
19 Curso: Compl2
20 [74, 60, 32, 63, 78]
21
22 Estudiante: Fermina
23 Curso: Arquí
24 [78, 14, 23, 52, 33]
25 Curso: Archivos
26 [28, 79, 77, 55, 24]
27 Curso: Compl2
28 [23, 79, 47, 62, 44]
29
30 Estudiante: Markel
```

RUN >>>

OPTIMIZAR

Optimización

```
1 /*-----HEADER-----*/
2 #include <stdio.h>
3 #include <math.h>
4 double heap[30101999];
5 double stack[30101999];
6 double P;
7 double H;
8 double t0, t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20;
9
10 /*-----NATIVES-----*/
11 void dbrust_printString() {
12   t11 = P + 1;
13   t12 = stack[(int)t11];
14   L1:
15   t13 = heap[(int)t12];
16   if(t13 == -1) goto L0;
17   printf("%c", (char)t13);
18   t12 = t12 + 1;
19   goto L1;
20   L0:
21   return;
```

# Frontend

---

Estos botones son importantes también. Su función es Abrir un nuevo archivo, mostrar los diferentes tipos de reportes que existen como el árbol CST, la tabla de Errores y la Tabla de símbolos

