



# MANUAL TÉCNICO

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

PROYECTO 1

# 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

## 4 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`

# 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:

The screenshot displays the T-Swift web application interface. At the top center, the title "T-Swift" is prominently displayed. Below the title, there are three green buttons: "UPLOAD", "CST", and "ERRORES", followed by a "TABLA DE SIMBOLOS" button. The interface is divided into two main sections: "Input Code" on the left and "Console Output" on the right. The "Input Code" section contains a text area with Swift code, including functions for printing arrays and a main function for processing student data. A "RUN >>>" button is located at the bottom left of the code area. The "Console Output" section shows the results of the code execution, displaying student names, course names, and their respective scores. The output is formatted with line numbers and color-coded text.

```
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)
```

```
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 [17, 40, 61, 20, 44]
9 Curso: Archivos
10 [56, 60, 51, 68, 70]
11 Curso: Comp2
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: Comp2
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: Comp2
28 [23, 79, 47, 62, 44]
29
30 Estudiante: Markel
```

# 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

