

Nombre: Abdiel Fernando José Otzoy Oztín
Carnet: 202300350

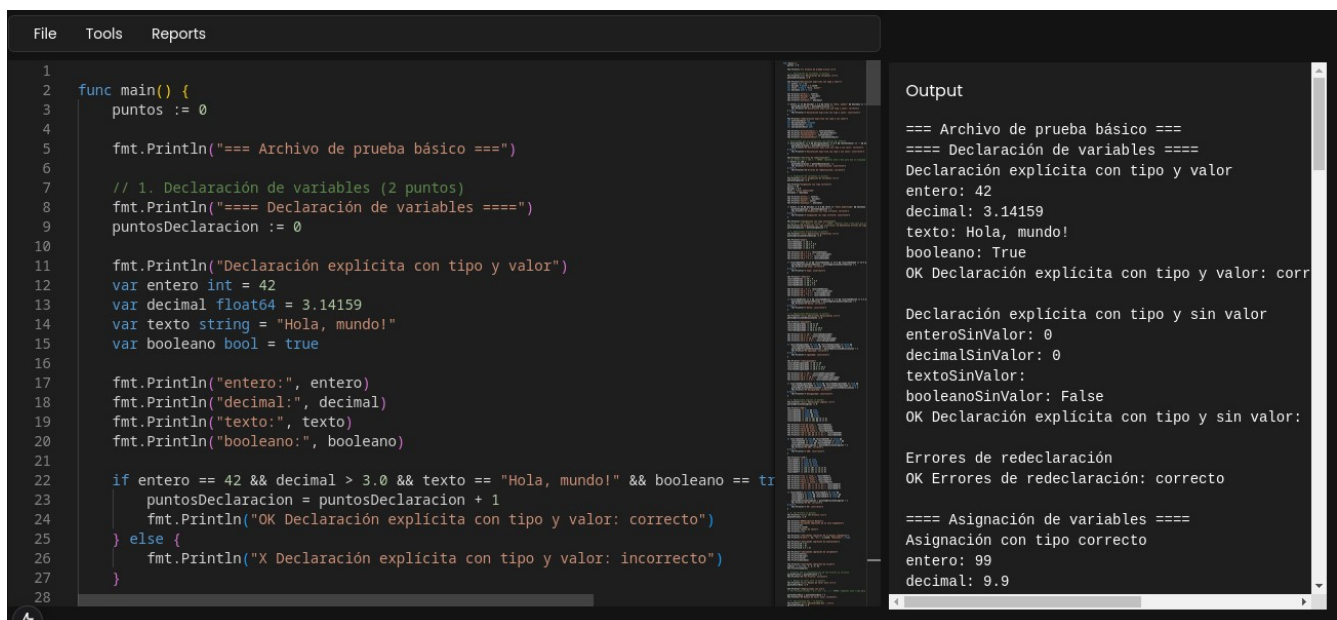
GOLIGTH

MANUAL DE USUARIO

Este proyecto tiene como objetivo desarrollar un intérprete para el lenguaje de programación Go utilizando ANTLR4 como herramienta de análisis sintáctico y C# para la implementación del intérprete. La API se construye sobre la plataforma .NET, lo que permite una integración fluida con un frontend desarrollado en Next.js 15. Este intérprete tiene la capacidad de analizar y ejecutar código Go, proporcionando retroalimentación en tiempo real sobre la ejecución del código y los errores que puedan surgir.

Descripción de funcionalidades

La aplicación cuenta con una interfaz amigable para el usuario, en donde lo primero que el usuario vería es esto



The screenshot displays the GOLIGTH application interface. On the left, a code editor shows Go code for a test file. The code includes variable declarations and a conditional check. On the right, the 'Output' panel shows the execution results, including variable values and success/failure messages for the declarations.

```
1 func main() {
2     puntos := 0
3
4     fmt.Println("=== Archivo de prueba básico ===")
5
6     // 1. Declaración de variables (2 puntos)
7     fmt.Println("==== Declaración de variables ====")
8     puntosDeclaracion := 0
9
10    fmt.Println("Declaración explícita con tipo y valor")
11    var entero int = 42
12    var decimal float64 = 3.14159
13    var texto string = "Hola, mundo!"
14    var booleano bool = true
15
16    fmt.Println("entero:", entero)
17    fmt.Println("decimal:", decimal)
18    fmt.Println("texto:", texto)
19    fmt.Println("booleano:", booleano)
20
21    if entero == 42 && decimal > 3.0 && texto == "Hola, mundo!" && booleano == true {
22        puntosDeclaracion = puntosDeclaracion + 1
23        fmt.Println("OK Declaración explícita con tipo y valor: correcto")
24    } else {
25        fmt.Println("X Declaración explícita con tipo y valor: incorrecto")
26    }
27 }
28
```

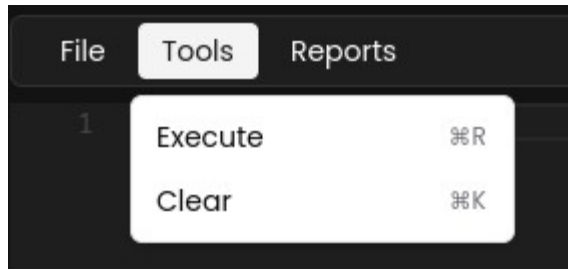
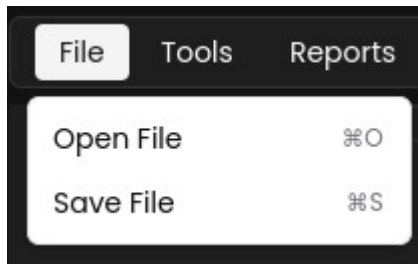
Output

```
=== Archivo de prueba básico ===
==== Declaración de variables ====
Declaración explícita con tipo y valor
entero: 42
decimal: 3.14159
texto: Hola, mundo!
booleano: True
OK Declaración explícita con tipo y valor: correcto

Declaración explícita con tipo y sin valor
enteroSinValor: 0
decimalSinValor: 0
textoSinValor:
booleanoSinValor: False
OK Declaración explícita con tipo y sin valor: correcto

Errores de redeclaración
OK Errores de redeclaración: correcto

==== Asignación de variables ====
Asignación con tipo correcto
entero: 99
decimal: 9.9
```



En la barra de navegación se tienen dos principales opciones, las cuales son las opciones para file y otra llamada tools, en file el usuario puede abrir archivos .glt y guardarlos, y en la parte de tools el usuario puede ejecutar el código cargado o limpiar la consola.

```

47
48     fmt.Println("\nErrores de redeclaración")
49     // var entero int = 100 // ! ERROR: Comentar esta línea para que el programa
50     if entero == 100 {
51         puntosDeclaracion = puntosDeclaracion - 1
52         fmt.Println("X Errores de redeclaración: incorrecto")
53     } else {
54         fmt.Println("OK Errores de redeclaración: correcto")
55     }
56
57     // 2. Asignación de variables (2 puntos)
58     fmt.Println("\n==== Asignación de variables ====")
59     puntosAsignacion := 0
60

```

```

decimalSinValor: 0
textoSinValor:
booleanoSinValor: False
OK Declaración explícita con tipo y sin valor:

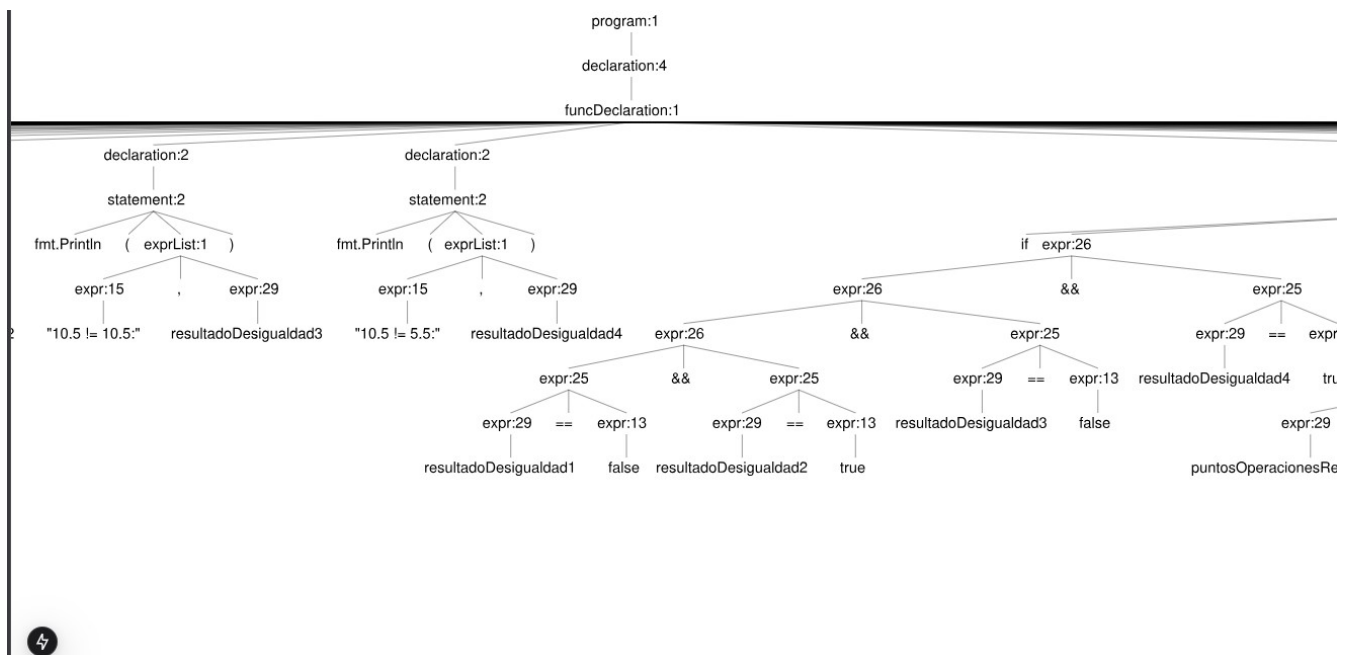
Errores de redeclaración
OK Errores de redeclaración: correcto

==== Asignación de variables ====
Asignación con tipo correcto
entero: 99
decimal: 9.9

```

Symbols table Errors ast Clear AST

La parte de reportes por temas de mejor visibilidad estará por debajo del editor de código, en donde se podrán generar los reportes para la tabla de simbolos, de errores y para el árbol de análisis sintáctico.



Para el reporte del árbol ast se podrá ver con ayuda del scroll el árbol completo


The screenshot shows a code editor with a C program. A 'Symbols Table' window is open, displaying the following table:

ID	Tipo simbolo	Tipo dato	Linea
puntos	int	variable	3
puntosDeclaracion	int	variable	9
entero	int	variable	12
decimal	float64	variable	13
texto	string	variable	14

The background code is a C program that tests variable declarations. The output window on the right shows the execution results, including variable values and messages about declaration correctness.

Así se verá el reporte de tabla de simbolos

En caso de que haya un error se verá de la siguiente manera:



Variable entero already declared at line 49
column 2

este es el área de output, en donde se podrá ver el resultado del código interpretado

```
Output

=== Archivo de prueba básico ===
==== Declaración de variables ====
Declaración explícita con tipo y valor
entero: 42
decimal: 3.14159
texto: Hola, mundo!
booleano: True
OK Declaración explícita con tipo y valor: correcto

Declaración explícita con tipo y sin valor
enteroSinValor: 0
decimalSinValor: 0
textoSinValor:
booleanoSinValor: False
OK Declaración explícita con tipo y sin valor: correcto

Errores de redeclaración
OK Errores de redeclaración: correcto

==== Asignación de variables ====
Asignación con tipo correcto
entero: 99
decimal: 9.9
```