

Nombre: Abdiel Fernando José Otzoy Oztín

Carnet: 202300350

Proyecto 1 – GoLigth

Introducción

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.

Funcionamiento

El intérprete se basa en un diseño de arquitectura de software que incluye los siguientes componentes:

1. **ANTLR4:** Se utiliza para generar un analizador sintáctico a partir de una gramática definida para el lenguaje Go. Esto permite descomponer el código fuente en una estructura de árbol que puede ser procesada por el intérprete.
2. **C#:** El código generado por ANTLR4 se integra en una aplicación .NET, donde se implementan las reglas de ejecución del lenguaje Go. Esto incluye la gestión de variables, funciones, estructuras de control y la ejecución de expresiones.
3. **API .NET:** La API expone endpoints que permiten al frontend enviar código Go para su ejecución. La API procesa el código, lo analiza, lo ejecuta y devuelve los resultados al frontend.
4. **Next.js 15:** El frontend se encarga de la interfaz de usuario, permitiendo a los usuarios escribir y enviar código Go, así como visualizar los resultados y errores de ejecución.

Requerimientos

Para ejecutar este proyecto, se requieren los siguientes componentes:

- **.NET SDK:** Versión 5.0 o superior.

- **ANTLR4:** Herramienta para la generación de analizadores sintácticos.
- **Node.js:** Para la ejecución del frontend en Next.js.
- **Un editor de código:** Como Visual Studio o Visual Studio Code.

Funcionalidades de CompilerVisitor.cs

El archivo CompilerVisitor.cs es una parte fundamental del intérprete, ya que se encarga de visitar los nodos del árbol de sintaxis abstracta (AST) generado por ANTLR4 y ejecutar las acciones correspondientes. A continuación, se describen algunas de las funcionalidades clave de esta clase:

1. **Visita de Programas:** El método VisitProgram itera sobre las declaraciones en el contexto del programa y las procesa.
2. **Declaración de Variables:** El método VisitVarDeclaration maneja la declaración de variables, asignando valores por defecto si no se proporciona uno.
3. **Declaración de Funciones:** El método VisitFuncDeclaration permite la declaración de funciones, incluyendo funciones asociadas a estructuras.
4. **Ejecución de la Función main:** El método ExecuteMain busca y ejecuta la función main, que es el punto de entrada del programa Go.
5. **Gestión de Entornos:** La clase utiliza un entorno (Environment) para almacenar variables y funciones, permitiendo la gestión de ámbitos y el acceso a variables locales y globales.
6. **Manejo de Estructuras de Control:** Métodos como VisitIfStmt, VisitForStmt, y VisitSwitchStmt permiten la ejecución de estructuras de control, evaluando condiciones y ejecutando bloques de código según sea necesario.
7. **Impresión de Resultados:** El método VisitPrintStmt se encarga de imprimir resultados en la consola, formateando adecuadamente los valores.
8. **Gestión de Errores:** La clase incluye mecanismos para lanzar excepciones en caso de errores semánticos, como la declaración de variables duplicadas o el acceso a índices fuera de rango.

