



Ingeniería en Computación

Lenguajes de Programación

Mini-Go Compiler

Segunda Entrega

Profesor:

Oscar Mario Víquez Acuña

Grupo 50, 51

Estudiantes:

Kevin Vinicio Varela Rojas

Anthony Andrés Jiménez Zamora

Fabián Rojas Ugalde

Sede San Carlos

03/04/2024

## **Soluciones e implementación:**

La realización de un proyecto de desarrollo de un compilador basado en Go implica una serie de desafíos técnicos que requieren soluciones ingeniosas y meticulosamente diseñadas. En este contexto, se han implementado diversas soluciones para abordar aspectos cruciales del proceso de compilación, cada una contribuyendo de manera significativa a la funcionalidad y eficacia global del sistema.

Una de las piedras angulares de esta implementación es la tabla de símbolos, una estructura de datos fundamental que desempeña un papel central en la identificación y gestión de los tokens que componen el lenguaje Mini Go. Esta tabla no solo actúa como un repositorio centralizado para la información sobre identificadores, variables y otros elementos del código fuente, sino que también facilita la navegación eficiente a través del mismo, lo que resulta esencial para tareas como el análisis léxico y semántico.

Por otro lado, el "checker" o verificador, representa un componente crítico que garantiza la conformidad del código fuente con las reglas y restricciones sintácticas establecidas por la gramática del lenguaje. Este conjunto de funciones está diseñado para evaluar cada aspecto del código en busca de posibles violaciones gramaticales, proporcionando así una capa adicional de seguridad y validación durante el proceso de compilación. La implementación cuidadosa de estas funciones garantiza que el código generado sea coherente y cumpla con los estándares definidos.

En última instancia, el éxito de este proyecto radica en la cuidadosa implementación y coordinación de estas soluciones técnicas, que trabajan en conjunto para garantizar un proceso de compilación fluido y confiable. A través

de un enfoque meticuloso y una atención meticulosa al detalle, se ha logrado crear un compilador basado en Go que cumple con los más altos estándares de calidad y precisión.

**Resultados obtenidos:**

Aspecto	Realizado	No Realizado
Tabla de símbolos	Se implementó una tabla de símbolos sólida que permite la identificación y gestión eficiente	

	de los tokens del lenguaje Mini Go.	
Checker	Se desarrolló un conjunto de funciones para verificar el cumplimiento de la gramática del lenguaje Mini Go, asegurando la coherencia y validez del código compilado.	
Validaciones	Se llevaron a cabo validaciones exhaustivas para garantizar la corrección sintáctica y semántica del código fuente.	
Integración	Integración exitosa de la interfaz web con el scanner y parser para compilar código Go	
Manejo de Errores	Se implementa código para administrar los errores del compilador	
Documentación	Creación de documentación del proyecto y del funcionamiento.	
Pruebas	Realización de pruebas para verificar la correcta funcionalidad del compilador	

## **Conclusiones:**

En primer lugar, es importante destacar el logro significativo que representa la culminación exitosa de este proyecto. La creación de un compilador funcional que pueda traducir código Mini Go a instrucciones ejecutables es un hito impresionante que refleja el arduo trabajo, la dedicación y la habilidad técnica del equipo de desarrollo involucrado. La integración exitosa de componentes clave, como el scanner, parser y checker, demuestra la capacidad para superar desafíos técnicos complejos y llevar a cabo soluciones efectivas.

Además, la realización de pruebas exhaustivas para verificar la funcionalidad del compilador es un paso crucial en el proceso de desarrollo de software. Estas pruebas no solo validan la corrección y robustez del código, sino que también brindan confianza en la calidad del producto final. La capacidad para identificar y gestionar errores durante las pruebas es especialmente destacable, ya que evidencia un enfoque riguroso hacia la calidad del software y la satisfacción del usuario.

La documentación completa y clara del proyecto y del funcionamiento del compilador es otro aspecto destacado. Una documentación detallada no solo sirve como una guía invaluable para los usuarios finales, sino que también facilita la colaboración entre miembros del equipo de desarrollo y futuras mejoras o actualizaciones del software. La transparencia y la accesibilidad en la documentación son elementos clave para garantizar la comprensión y la adopción exitosa del compilador por parte de la comunidad de usuarios.

**Bibliografía:**

ChatGPT. (n.d.). *ChatGPT*. Ayuda en minigo. Retrieved marzo, 2024, from <https://chat.openai.com/>