**UNIVERSIDADE FEDERAL DE ITAJUBÁ - CAMPUS DE ITABIRA**

ISRAEL BERNARDO DE ASSIS SILVA

CAUÃ NOLASCO DE CARVALHO ALVARENGA

GUILHERME BOMFIN FIGUEIREDO ARAUJO

LUIZ GUSTAVO MADEIRA FERREIRA

DIEGO NUNES CALDEIRA

**Explorando a linguagem de programação Golang**

**Itabira**

**2023**

**Explorando a linguagem de programação Golang**

**Feito por**

ISRAEL BERNARDO DE ASSIS SILVA

CAUÃ NOLASCO DE CARVALHO ALVARENGA

GUILHERME BOMFIN FIGUEIREDO ARAUJO

LUIZ GUSTAVO MADEIRA FERREIRA

DIEGO NUNES CALDEIRA

1. **Introdução**

Nesse seminário, apresentaremos sobre a introdução à linguagem Go, abordando sua história, origens e adoção ao longo do tempo. Em seguida, discute os paradigmas de programação suportados por Go e suas principais características. A sintaxe básica da linguagem e as estruturas de controle são exploradas, seguidas por uma discussão sobre recursos avançados como goroutines e gerenciamento eficiente de memória. O seminário conclui com exemplos práticos de programas em Go, uma discussão sobre casos de uso relevantes e uma comparação de Go com outras linguagens populares.

**II. Introdução à Linguagem Go**

A linguagem Go, desenvolvida pelo Google em 2007 e com sua primeira versão open-source lançada em 2009, surgiu em um contexto de busca por uma linguagem de programação que fosse simultaneamente acessível e capaz de enfrentar os desafios impostos pelos sistemas complexos da empresa. Os criadores de Go tinham como objetivo projetar uma linguagem que fosse fácil de usar, ao mesmo tempo em que fosse adequada para lidar com os complexos sistemas do Google.

Durante seu desenvolvimento, o Google promoveu diversas alterações na sintaxe e semântica da linguagem, buscando se afastar das semelhanças com a linguagem C. Essas mudanças tinham como propósito tornar Go mais acessível e autossuficiente para programadores iniciantes, além de torná-la uma escolha sólida para projetos de grande escala.

Embora inicialmente concebida para o desenvolvimento de sistemas, Go evoluiu ao longo do tempo e, atualmente, é mais amplamente utilizada no desenvolvimento web. Sua simplicidade e eficiência a tornam uma opção popular para criar aplicativos de servidor, APIs e outras soluções de software na era digital. A linguagem Go é apreciada por sua concorrência eficaz, excelente desempenho e facilidade de manutenção, o que a torna uma escolha atraente para muitos desenvolvedores e empresas.

**III. Características e Paradigmas de Programação**

Go é uma linguagem versátil que, embora predominantemente imperativa, incorpora elementos de programação concorrente e funcional. Suas principais características incluem simplicidade de sintaxe, eficiência de desempenho e suporte à concorrência nativa. A filosofia de design de Go enfatiza a importância da simplicidade, o que se reflete em sua sintaxe concisa e preferência por mecanismos mais básicos. Essas características fazem de Go uma escolha atraente para uma variedade de aplicativos e destacam sua flexibilidade na abordagem de problemas de programação.

**IV. Sintaxe e Estruturas de Controles Básicas**

Neste segmento do nosso seminário sobre a linguagem Go, exploraremos os elementos fundamentais da sintaxe e examinaremos as estruturas de controle básicas que influenciam o fluxo e a lógica do programa.

* **Tipagem Estática e Inferência de Tipos**:
* Go é notável por sua tipagem estática, onde os tipos de variáveis e constantes devem ser explicitamente declarados. Além disso, a linguagem oferece a capacidade de inferência de tipos, permitindo que o compilador determine automaticamente o tipo das variáveis com base em seus valores, melhorando a clareza do código.
* **Tipagem Forte**:
* Go é uma linguagem de tipagem forte, o que significa que os tipos de dados são estritamente aplicados. Conversões de tipos exigem intervenção explícita do desenvolvedor, promovendo a segurança do código.
* **Tipos de Dados Versáteis**:
* Go suporta uma variedade de tipos de dados, incluindo inteiros, números de ponto flutuante e booleanos. Além disso, a linguagem oferece recursos robustos para manipular strings e caracteres Unicode. Ela também inclui tipos de tamanho fixo, como int32 (rune) e uint8 (byte).
* **Variáveis e Constantes**:
* A declaração de variáveis em Go é realizada com a palavra-chave "var," permitindo inicialização no momento da declaração. As constantes, definidas com "const," possuem valores imutáveis durante a execução do programa.
* **Convenções de Nomenclatura**:
* Go segue estritamente as convenções de nomenclatura, com o camelCase sendo o padrão para nomes de variáveis e constantes, onde as palavras são escritas em letras minúsculas, com a primeira letra de cada palavra em maiúscula. O PascalCase é utilizado ao exportar variáveis ou constantes para fora do pacote.
* **Estruturas de Controle Básicas**:
* Além da sintaxe, Go possui estruturas de controle fundamentais para orientar o fluxo do programa. Estas incluem estruturas condicionais (if, else) para tomada de decisões com base em condições, bem como loops (for) para repetir ações. Go também oferece instruções de controle como break e continue para gerenciar o fluxo do programa em situações específicas.

A compreensão desses conceitos de sintaxe e estruturas de controle é essencial para se tornar um desenvolvedor habilidoso em Go. À medida que avançamos em nosso seminário, exploraremos tópicos mais avançados e práticos para aproveitar ao máximo essa linguagem de programação versátil.

**V. Recursos Avançados e Bibliotecas**

a. **Concorrência em Go e Goroutines**:

Go é conhecido por sua simplicidade na implementação de concorrência através do uso de goroutines. Goroutines são funções ou métodos que são executados simultaneamente com outras funções ou métodos. Goroutines são mais leves do que threads e a comunicação entre elas é feita usando canais.

b. **Gerenciamento de Memória Eficiente**:

Go tem um coletor de lixo embutido, o que significa que a alocação e desalocação de memória é feita automaticamente. Isso permite que os desenvolvedores se concentrem mais na lógica do programa em vez de gerenciar a memória manualmente. No entanto, entender como a alocação de memória funciona em Go pode ajudar a escrever códigos mais otimizados.

c. **Tratamento de Erros e Exceções**:

Em Go, o erro é o tipo básico usado para representar uma condição de erro, com o valor nulo representando nenhum erro. O tratamento de erros em Go é feito através de um retorno explícito de valores de erro, o que difere das exceções usadas em outras linguagens como Java e Python.

d. **Visão Geral das Bibliotecas Padrão**:

Go vem com uma série de bibliotecas padrão que fornecem funcionalidades úteis:

* **fmt**: Esta biblioteca fornece funcionalidades para formatação de I/O. Ela implementa E/S formatada com funções análogas às funções printf e scanf da linguagem C. Os 'verbos' de formato são derivados da linguagem C, mas são mais simples.
* **net**: Esta biblioteca é usada para programação de rede, incluindo TCP/IP, UDP, resolução de nomes de domínio e sockets de domínio Unix. Ela fornece uma interface portátil para I/O de rede.
* **net/http**: Esta biblioteca fornece funcionalidades HTTP cliente e servidor. Ela fornece implementações de cliente e servidor HTTP. As funções Get, Head, Post e PostForm fazem solicitações HTTP (ou HTTPS).
* **gin**: Gin é um framework web HTTP escrito em Go. Ele apresenta uma API semelhante à do Martini, mas com desempenho até 40 vezes mais rápido graças ao httprouter.

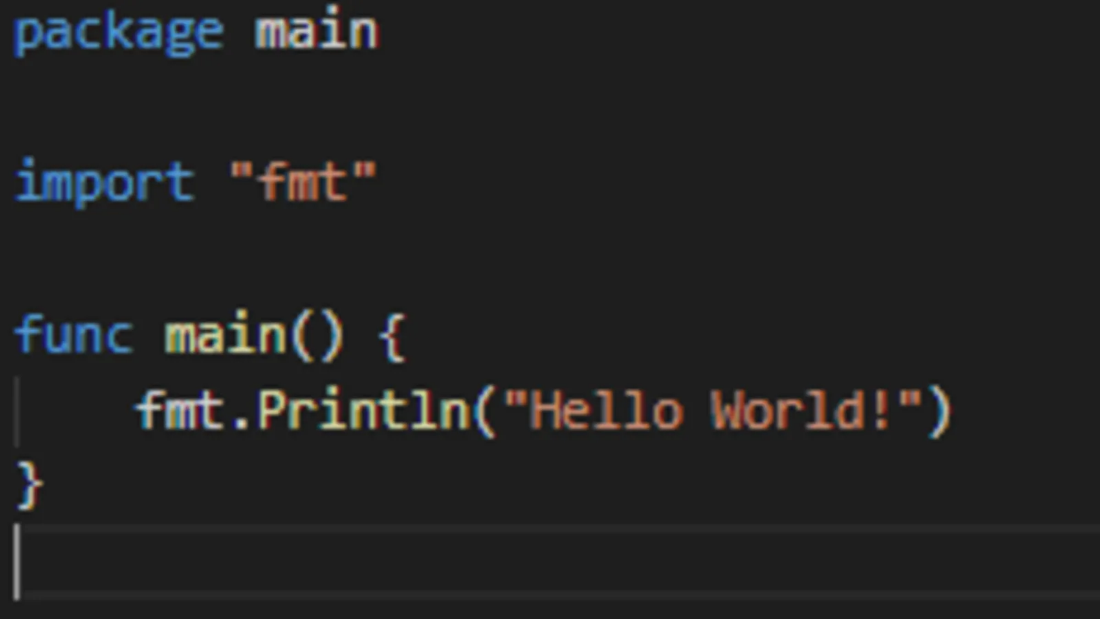
e. **Uso de Módulos e Pacotes Externos**:

Os módulos são a maneira como Go gerencia as dependências. Um módulo é uma coleção de pacotes Go que são liberados, versionados e distribuídos juntos. Os pacotes externos podem ser importados e usados em seu código, permitindo a reutilização do código e a organização eficiente do projeto.

**VI. Exemplos Práticos e Casos de Uso Relevantes**

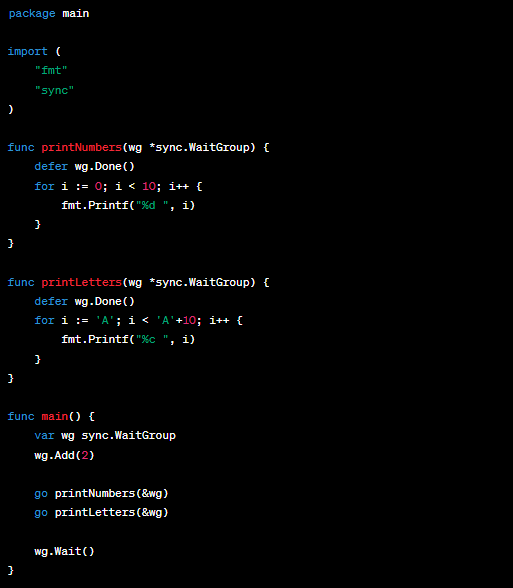
a. **Demonstração de Programas Simples em Go**:

Neste primeiro exemplo, temos um programa simples em Go que é um clássico entre os iniciantes. O código a seguir imprime "Olá, mundo!" na saída padrão.

  
}

b. **Exemplos de Uso de Goroutines para Concorrência**:

Aqui, temos um exemplo que demonstra o uso de goroutines em Go para imprimir números e letras simultaneamente. Essa funcionalidade permite que as tarefas sejam executadas concorrentemente, melhorando o desempenho e a eficiência.



**VII. Comparação com Outras Linguagens Populares**

**a. Comparação com linguagens como Python, Java, C++, etc.:**

Comparar Go com outras linguagens populares é fundamental para entender onde Go se destaca e onde pode ser mais apropriado.

* **Python**: Go oferece desempenho superior, especialmente em cenários de concorrência, devido à sua natureza compilada. Enquanto Python é conhecido por sua simplicidade e produtividade, Go brilha na eficiência e na escalabilidade.
* **Java**: Java é uma linguagem amplamente usada em aplicativos empresariais. Go é mais simples e eficiente, mas Java possui uma ampla gama de bibliotecas e é mais consolidado em grandes empresas.
* **C++**: Go é mais seguro e eficiente em termos de gerenciamento de memória em comparação com C++. No entanto, C++ oferece mais controle de baixo nível, o que pode ser necessário em alguns casos.

**b. Vantagens e desvantagens de usar Go em diferentes cenários:**

* **Vantagens de Go**: Go é uma excelente escolha para desenvolvimento de servidores, sistemas distribuídos e serviços de rede. Suporta concorrência de forma nativa, é eficiente em termos de recursos, e possui uma excelente biblioteca padrão.
* **Desvantagens de Go**: Pode não ser a melhor escolha para tarefas intensivas em computação ou desenvolvimento de GUIs complexas, pois possui menos bibliotecas e suporte do que linguagens como Python.

**c. Exemplos de empresas que adotaram Go e os resultados alcançados:**

Muitas empresas adotaram Go e obtiveram benefícios significativos. Aqui estão alguns exemplos notáveis:

* **Google**: Como a criadora de Go, o Google utiliza a linguagem em uma variedade de aplicativos internos e serviços, incluindo aplicações de alto desempenho e sistemas de gerenciamento de contêineres.
* **Uber**: A Uber adotou Go para melhorar o desempenho e a eficiência de seus sistemas. Eles relataram uma redução significativa no uso de CPU e memória em comparação com outras linguagens.
* **Dropbox**: O Dropbox utiliza Go para otimizar o desempenho de suas ferramentas de armazenamento e sincronização de arquivos. A linguagem foi escolhida devido à sua capacidade de suportar sistemas altamente concorrentes.

Esses exemplos demonstram o valor prático de Go em cenários do mundo real e como a linguagem tem sido benéfica para empresas em busca de desempenho e eficiência.

**VII. Conclusão**

Neste seminário, percorremos os principais aspectos da linguagem Go, desde sua sintaxe e estruturas de controle básicas até seus recursos avançados e bibliotecas padrão. Discutimos sua tipagem estática, tipagem forte e sua capacidade de suportar concorrência por meio de goroutines. Exploramos exemplos práticos e casos de uso em empresas do mundo real para ilustrar como Go pode ser aplicado em diversos cenários. Aprendizado é um processo contínuo, e Go oferece uma comunidade rica em recursos para aprimorar suas habilidades. Recomendamos a busca por tutoriais, cursos online, documentação oficial e fóruns de discussão para continuar a jornada de aprendizado em Go. A prática é fundamental, e a linguagem oferece muitos desafios interessantes para desenvolvedores.

Referências

Valueshot. Introduçao a linguagem.

https://www.valuehost.com.br/blog/golang/

StackOverflow. Caracteristicas e paradigmas.

https://pt.stackoverflow.com/questions/227429/quais-s%c3%a3o-as-principais-caracter%c3%adsticas-da-linguagem-go

TreinaWeb. Sintaxe e estruturas de controle.

https://www.treinaweb.com.br/blog/conhecendo-variaveis-e-constantes-na-golang

Dev, Coodesh, Medium, Go. Recursos avançados e bibliotecas.

https://dev.to/mfbmina/introducao-a-concorrencia-em-go-1c0g

https://coodesh.com/blog/candidates/frameworks-golang-veja-a-lista-dos-mais-populares-na-comunidade-tech/

https://medium.com/@phenrigomes/tratamento-de-erros-em-go-ba23cc01e6a1

https://go.dev/doc/faq#garbage\_collection

Appmaster, Dropbox, Uber, Go. Comparando a outras linguagens.

https://appmaster.io/pt/blog/go-vs-outras-linguagens-de-programacao

https://dropbox.tech/infrastructure/open-sourcing-our-go-libraries

https://www.uber.com/en-BR/blog/go-geofence-highest-query-per-second-service/

https://go.dev/solutions/google/