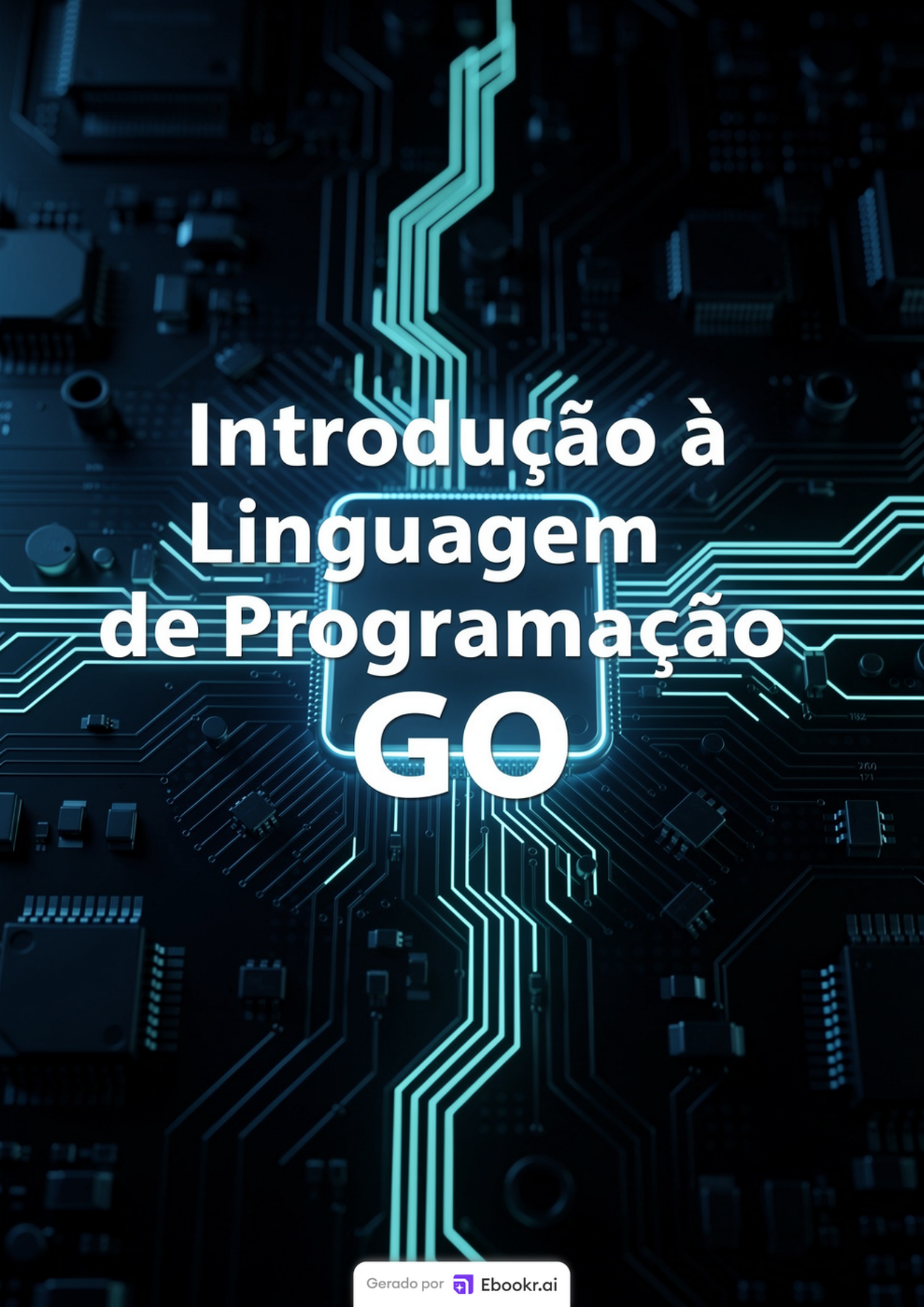


Sumário

Introdução

Seja bem-vindo(a) a esta jornada introdutória na fascinante Linguagem de Programação Go! Estamos animados por você ter escolhido explorar esse universo, onde simplicidade e eficiência se encontram para criar soluções poderosas. Go, desenvolvida pelo Google, é uma linguagem que visa facilitar a programação, permitindo que você escreva códigos claros e de fácil manutenção.

Desde seu lançamento em 2009, Go tem crescido em popularidade, especialmente entre desenvolvedores que buscam criar aplicações de alto desempenho. Sua sintaxe é simples e direta, o que a torna uma excelente opção tanto para iniciantes quanto para programadores mais experientes. Ao longo deste ebook, você terá a oportunidade de entender os fundamentos da linguagem Go e como ela pode ser aplicada em projetos do dia a dia.

Aprender Go pode abrir portas para diversas oportunidades no mercado de trabalho, uma vez que muitas empresas estão adotando esta linguagem para desenvolvimento de software, serviços em nuvem e muito mais. Não se preocupe se você é novo no mundo da programação; este material foi elaborado para guiá-lo de forma clara e acolhedora. Prepare-se para se aprofundar na Linguagem Go e descobrir como ela pode transformar sua maneira de programar!

CAPÍTULO 1

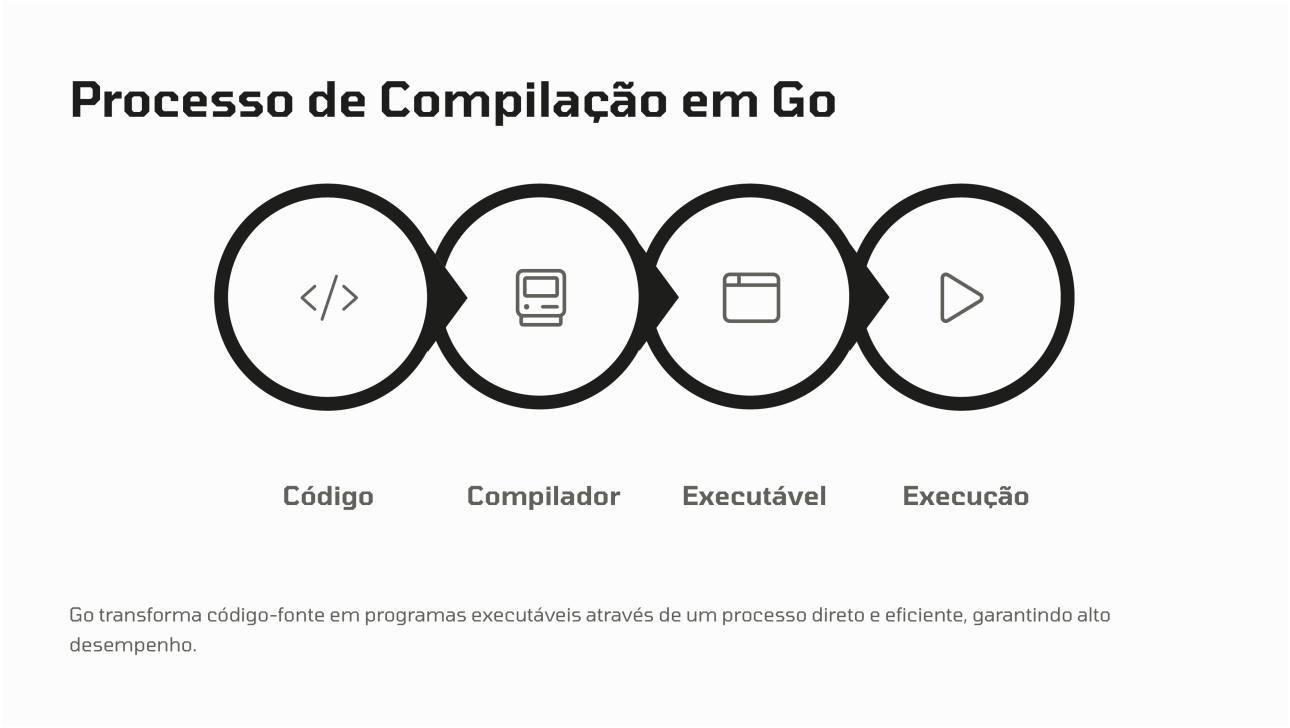
Introdução à Linguagem Go

Capítulo 1: Introdução à Linguagem Go

A linguagem Go, também conhecida como Golang, foi criada pela Google em 2007 e se tornou uma opção popular para desenvolvimento de software, especialmente em sistemas de alta performance. A ideia principal por trás do Go é simplicidade e eficiência, como um carro esportivo que combina velocidade com fácil manuseio.

Go é uma linguagem compilada, o que significa que o código é transformado em um arquivo executável antes de ser executado. Isso geralmente resulta em programas mais rápidos do que aqueles escritos em linguagens interpretadas, como Python. Imagine que compilar é como assar um bolo: você mistura os ingredientes e, em vez de comer a massa crua, você espera para saborear o produto final.

O diagrama a seguir ilustra o processo de compilação em Go, desde o código até a execução.



Um dos grandes atrativos do Go é sua sintaxe simples e clara. Por exemplo, para imprimir “Olá, mundo!”, o código é bem direto:

```go
package main

import "fmt"

func main() {
    fmt.Println("Olá, mundo!")
}
```

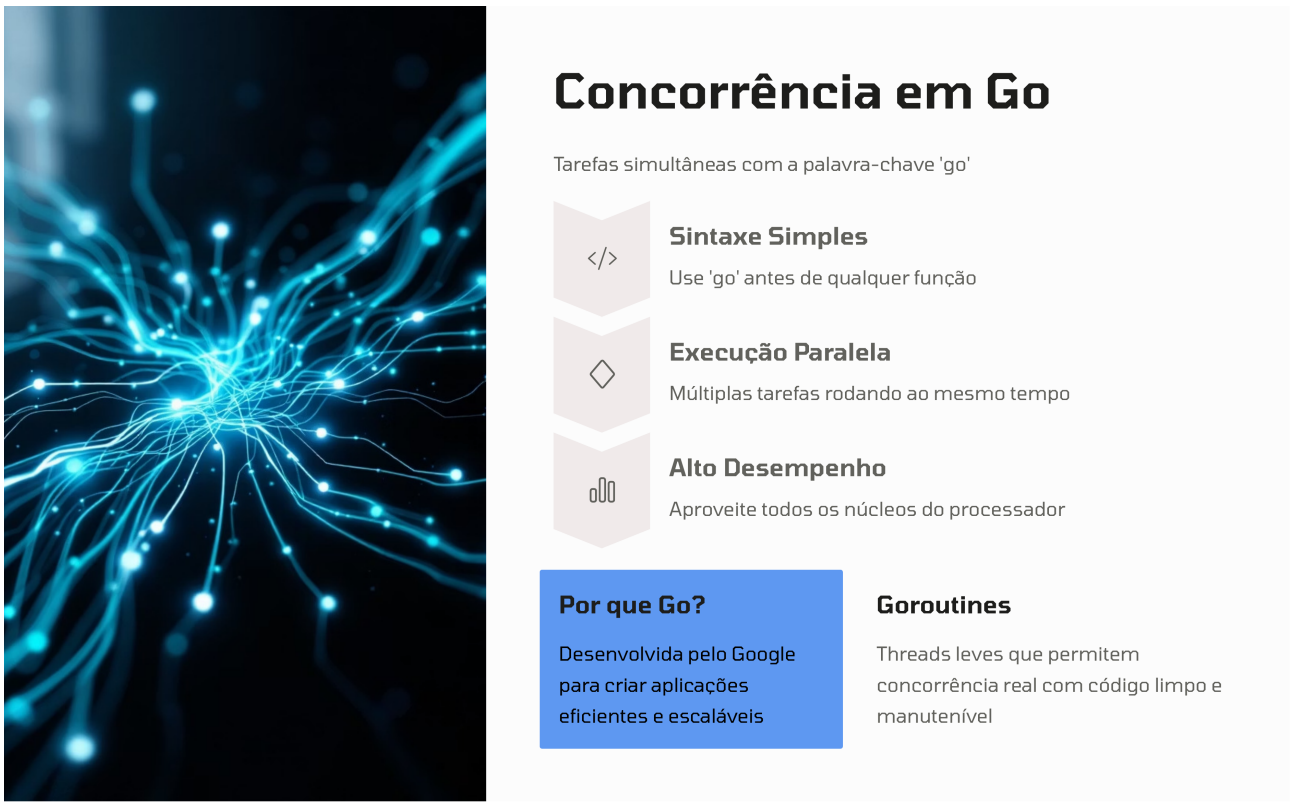
Aqui, o `package main` indica que este é o ponto de entrada do programa. A função `main()` é onde a execução começa. O `import "fmt"` é como abrir uma caixa de ferramentas, permitindo o uso de funções da biblioteca padrão para formatação, como `fmt.Println()`, que imprime texto no console.

Além da simplicidade, Go possui suporte embutido para concorrência, permitindo que múltiplas tarefas sejam executadas ao mesmo tempo de forma eficiente. Pense nisso como uma cozinha onde várias receitas podem ser feitas simultaneamente sem bagunçar tudo. A palavra-chave `go` é usada para executar funções de forma concorrente:

```go
func cozinhar() {
    fmt.Println("Cozinhando...")
}

func main() {
    go cozinhar() // Executa a função cozinhar de forma concorrente
    fmt.Println("Prato pronto!")
}
```

O infográfico a seguir ilustra como a palavra-chave ‘go’ permite a execução concorrente de tarefas em Go.



Quando você executa o código acima, “Prato pronto!” pode ser impresso antes ou depois de “Cozinhando…” dependendo de como o sistema executa as tarefas. Isso demonstra a natureza não determinística da concorrência em Go.

Uma dica prática é sempre começar com exemplos simples e ir aumentando a complexidade. Isso facilita a compreensão e evita frustrações. À medida que você se familiariza com a linguagem, explore os pacotes e bibliotecas que a comunidade de Go oferece. Com essa base, você estará bem encaminhado para construir aplicações robustas e eficientes em Go.

História e evolução do Go

A linguagem de programação Go, frequentemente chamada de Golang, foi criada por Google em 2007 e lançada oficialmente em 2009. Ela surgiu da necessidade de uma linguagem que fosse eficiente, simples e que suportasse a programação concorrente, ou seja, a capacidade de executar várias tarefas ao mesmo tempo. Imagine que você está gerenciando uma cozinha movimentada: você precisa que vários pratos sejam preparados ao mesmo tempo, mas de forma organizada e eficiente. Go foi projetada para ajudar os desenvolvedores a fazer isso em seus programas.

Os criadores da linguagem, Robert Griesemer, Rob Pike e Ken Thompson, queriam resolver problemas que eles enfrentavam com linguagens existentes, como C++ e Java, que eram complexas e lentas para compilar. Portanto, Go foi projetada para ser simples, fácil de aprender e com um tempo de compilação rápido, semelhante a uma refeição rápida que você pode preparar sem muito esforço.

Um dos principais recursos do Go é a sua abordagem para concorrência, que é implementada através de goroutines e canais. As goroutines permitem que funções sejam executadas de forma assíncrona, como se você tivesse vários cozinheiros trabalhando em diferentes pratos ao mesmo tempo. Aqui está um exemplo simples de como iniciar uma goroutine:

```
package main
import (
    "fmt"
    "time"
)

func cozinhar(prato string) {
    fmt.Printf("Cozinhando %s...\n", prato)
    time.Sleep(2 * time.Second) // Simula o tempo de cozimento
    fmt.Printf("%s pronto!\n", prato)
}

func main() {
    go cozinhar("massa")
    go cozinhar("sopa")
    time.Sleep(3 * time.Second) // Espera os pratos terminarem
}
```

Neste código, a função `cozinhar` é chamada para preparar dois pratos ao mesmo tempo. As goroutines permitem que cada prato seja cozinhado simultaneamente, e você pode ver como isso se assemelha a ter vários cozinheiros na cozinha. Após 3 segundos, ambos os pratos são finalizados, mostrando a eficiência da concorrência em Go.

Além disso, Go tem uma sintaxe limpa e concisa. A primeira versão da linguagem foi lançada com um conjunto básico de funcionalidades, mas logo ganhou popularidade e uma comunidade ativa. Essa comunidade contribuiu com bibliotecas e recursos, tornando Go uma escolha popular para desenvolvimento de sistemas, aplicativos web e muito mais.

Uma dica prática é explorar a documentação oficial do Go, onde você encontrará muitos exemplos e tutoriais. A linguagem é bastante acessível, e a prática constante ajudará você a se familiarizar rapidamente com seus conceitos e funcionalidades.

Em resumo, Go evoluiu de uma necessidade por eficiência e simplicidade em ambientes de desenvolvimento, e continua a ser uma ferramenta poderosa para desenvolvedores que desejam criar aplicações escaláveis e rápidas, aproveitando a sua abordagem única para concorrência.

Características principais da linguagem

A linguagem Go, também conhecida como Golang, tem várias características que a tornam única e atraente para desenvolvedores, especialmente iniciantes. Vamos explorar suas principais características de forma simples e prática.

Primeiramente, Go é uma linguagem **compilada**. Isso significa que o código que você escreve é transformado em um programa executável diretamente pelo computador. Essa característica traz desempenho e eficiência, além de facilitar a identificação de erros durante a compilação. Por exemplo, se você escrever um pequeno programa em Go:

```
package main
import "fmt"

func main() {
    fmt.Println("Olá, mundo!")
}
```

Ao compilar e rodar esse código, você verá a mensagem “Olá, mundo!” impressa no console. A vantagem é que, se houver um erro de sintaxe, o compilador o avisará antes mesmo de você executar o programa, economizando tempo na depuração.

Outra característica importante de Go é a sua **simplicidade**. A linguagem foi projetada para ter uma sintaxe clara e fácil de entender. Por exemplo, a declaração de variáveis é direta:

```
var idade int = 30
```

Aqui, estamos declarando uma variável chamada `idade` do tipo inteiro. Essa simplicidade permite que desenvolvedores iniciantes aprendam rapidamente os conceitos básicos, sem se perder em complexidades desnecessárias.

Além disso, Go possui um sistema de **concorrência** embutido, que facilita a execução de múltiplas tarefas ao mesmo tempo, algo essencial na programação moderna. Você pode criar goroutines, que são funções que podem ser executadas simultaneamente. Veja um exemplo:

```go
package main
import (
    "fmt"
    "time"
)

func contar() {
    for i := 1; i <= 5; i++ {
        fmt.Println(i)
        time.Sleep(time.Second)
    }
}

func main() {
    go contar() // Chama a função de forma concorrente
    fmt.Println("Contando...")
    time.Sleep(6 * time.Second) // Aguarda a goroutine completar
}
```

Neste código, a função `contar()` é executada em paralelo com a função `main()`. Enquanto ela conta até 5, a mensagem “Contando…” é exibida quase que simultaneamente. Isso demonstra como é fácil criar aplicações que fazem várias coisas ao mesmo tempo.

Por último, vale mencionar a **gestão de pacotes**. Go vem com um sistema integrado de gerenciamento de pacotes chamado `go mod`, que simplifica a importação e o uso de bibliotecas externas. Por exemplo, para usar uma biblioteca, você pode simplesmente rodar:

```
go get github.com/alguma/biblioteca
```

Isso facilita muito a vida do desenvolvedor, pois é possível adicionar funcionalidades rapidamente sem se preocupar com a configuração manual de dependências.

Em resumo, as principais características de Go - compilação, simplicidade, concorrência e gestão de pacotes - tornam a linguagem não apenas poderosa, mas também acessível para iniciantes. Essas qualidades fazem de Go uma escolha popular para o desenvolvimento de software moderno e eficiente.

Instalação e configuração do ambiente

Para começar a programar em Go, o primeiro passo é instalar e configurar o ambiente adequadamente. Imagine que você está montando uma nova cozinha: você precisa de ferramentas e espaço para cozinhar suas receitas. Da mesma forma, neste caso, o Go é a receita, e o ambiente é a sua cozinha.

1. Instalando o Go

Primeiro, você precisa baixar o instalador do Go. Vá para o site oficial golang.org e escolha a versão adequada para o seu sistema operacional (Windows, macOS ou Linux). Clique no link para baixar o arquivo.

Depois de baixar, siga as instruções específicas para o seu sistema:

- **Windows**: Execute o instalador e siga as instruções na tela. O Go será instalado na pasta `C:\Go`.
- **macOS**: Abra o terminal e execute o seguinte comando para instalar via Homebrew:

```
brew install go
```

- **Linux**: Extraia o arquivo baixado para o diretório `/usr/local`. Você pode usar o seguinte comando no terminal:*

```
sudo tar -C /usr/local -xzf go1.XX.linux-amd64.tar.gz
```

(Substitua `1.XX` pela versão que você baixou.)

2. Configurando as variáveis de ambiente

Após a instalação, você precisará configurar algumas variáveis de ambiente. Isso é como preparar sua cozinha, garantindo que tudo esteja no lugar certo. Para isso, adicione a seguinte linha ao seu arquivo de configuração do shell (como `.bashrc` ou `.zshrc`):

```
export PATH=$PATH:/usr/local/go/bin
```

Isso permite que você execute os comandos do Go de qualquer lugar no terminal. Para aplicar as mudanças, execute:

```
source ~/.bashrc  # ou source ~/.zshrc
```

3. Verificando a instalação

Agora, vamos garantir que tudo está funcionando corretamente. Abra o terminal e digite:

```
go version
```

Se tudo estiver certo, você verá a versão do Go instalada. Isso é como verificar se todos os utensílios estão na cozinha antes de começar a cozinhar.

4. Criando seu primeiro programa

Para testar se sua instalação está correta, vamos escrever um programa simples. Crie um novo arquivo chamado `hello.go` em qualquer diretório. Abra-o em seu editor de texto favorito e adicione o seguinte código:

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
}
```

Esse código é como sua primeira receita: ele imprime “Hello, World!” no terminal. Para executá-lo, navegue até o diretório onde está o arquivo e digite:

```
go run hello.go
```

Se você ver “Hello, World!” no terminal, parabéns! Você criou seu primeiro programa em Go. Isso é apenas o começo, mas é um passo importante para se aventurar no mundo da programação.

Dicas práticas:

- Sempre mantenha seu ambiente de desenvolvimento atualizado. Verifique regularmente novas versões do Go.
- Experimente usar um editor de texto com suporte a Go, como o Visual Studio Code, que pode ajudar com sugestões de código e formatação.

Agora que seu ambiente está configurado, você está pronto para começar a explorar a linguagem Go e suas funcionalidades!

Recapitulando...

Neste capítulo, você foi introduzido à linguagem Go, reconhecendo sua simplicidade e eficiência, fatores que a tornam uma escolha popular entre desenvolvedores. Exploramos sua história, destacando a criação pela Google e o foco em concorrência, ilustrado pelas goroutines, que permitem a execução simultânea de tarefas. Aprendemos também sobre as características principais da linguagem, como a compilação e a gestão de pacotes, que simplificam o desenvolvimento, especialmente para iniciantes. Por fim, você se familiarizou com o processo de instalação e configuração do ambiente, preparando-se para dar os primeiros passos na programação em Go. Agora, com essa base sólida, está pronto para mergulhar nos fundamentos da programação em Go, onde você começará a construir suas próprias aplicações e explorar o potencial dessa linguagem poderosa.