

Organização do nosso código

Primeira aproximação
sobre Go

Sumário

01 [Packages](#)

02 [Modules](#)



01

Packages

Packages

Lembre-se: cada arquivo de Go pertence a um package. Então, para declarar um desses arquivos usamos a seguinte sintaxe:

```
{ } package packagename
```

- O package deve ter o mesmo nome da pasta. Vários arquivos podem fazer parte do mesmo package.
- A declaração do package deve ser a primeira linha de código no arquivo fonte de Go.
- Todas as funções, tipos e variáveis definidas no arquivo fonte de Go passam a fazer parte do pacote declarado.
- O nome dos package dever ser uma única palavra, e toda minúscula.

Package e função main

Os programas em Go começam a ser executados no **package main**. Ele é um package que é usado com programas projetados para serem executáveis, e são conhecidos como comandos. Os outros programas são denominados simplesmente packages.

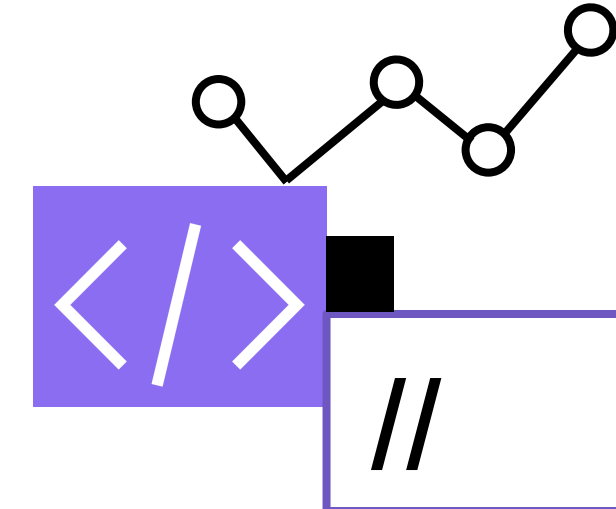
```
{}
```

```
package main
```

A função **main()** é especial, pois é o ponto de entrada de um programa executável. Vamos ver um exemplo de um programa executável em Go.

```
{}
```

```
func main() {  
    // Your code here  
}
```



Importações de packages

Go nos fornece duas formas de importar packages. Vamos ver quais são:

- Importação de packages individual:

{ }

```
import "fmt"
import "time"
```

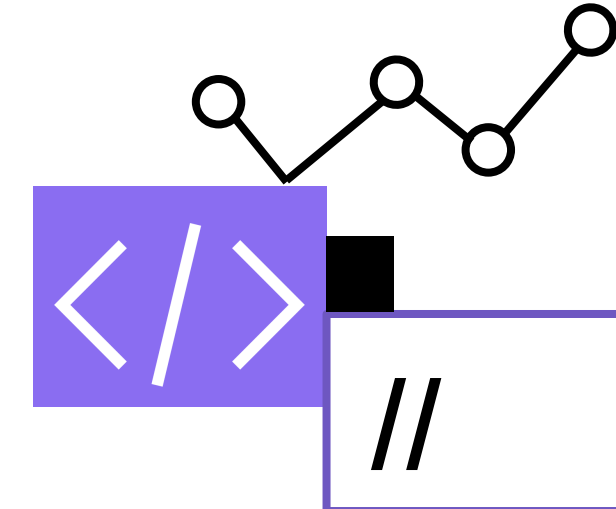
- Importação de packages em grupos:

{ }

```
import (
    "fmt"
    "time"
)
```

02

Modules



Como inicializar um module?

Antes de dar início a um módulo Go, devemos nos certificar de criar um novo repositório em GitHub - ou outros controles de versão (por exemplo, GitLab)-. Após isso, poderemos clonar o repositório. Para dar início a um módulo Go, usamos o seguinte comando:

Domínio

Nome do
módulo

```
{ } go mod init github.com/userGithub/go-simple-module
```

O domínio e o nome do módulo devem coincidir com o nome do repositório que já foi criado.

Após a inicialização do módulo, cria-se um arquivo **.go.mod**. Ele contém as dependências usadas na aplicação.

Como adicionar uma dependência?

É possível adicionar uma dependência à aplicação usando o comando **go get**. Go permite adicionar a ele o flag **-u** para baixá-la caso não exista, ou atualizar a dependência.

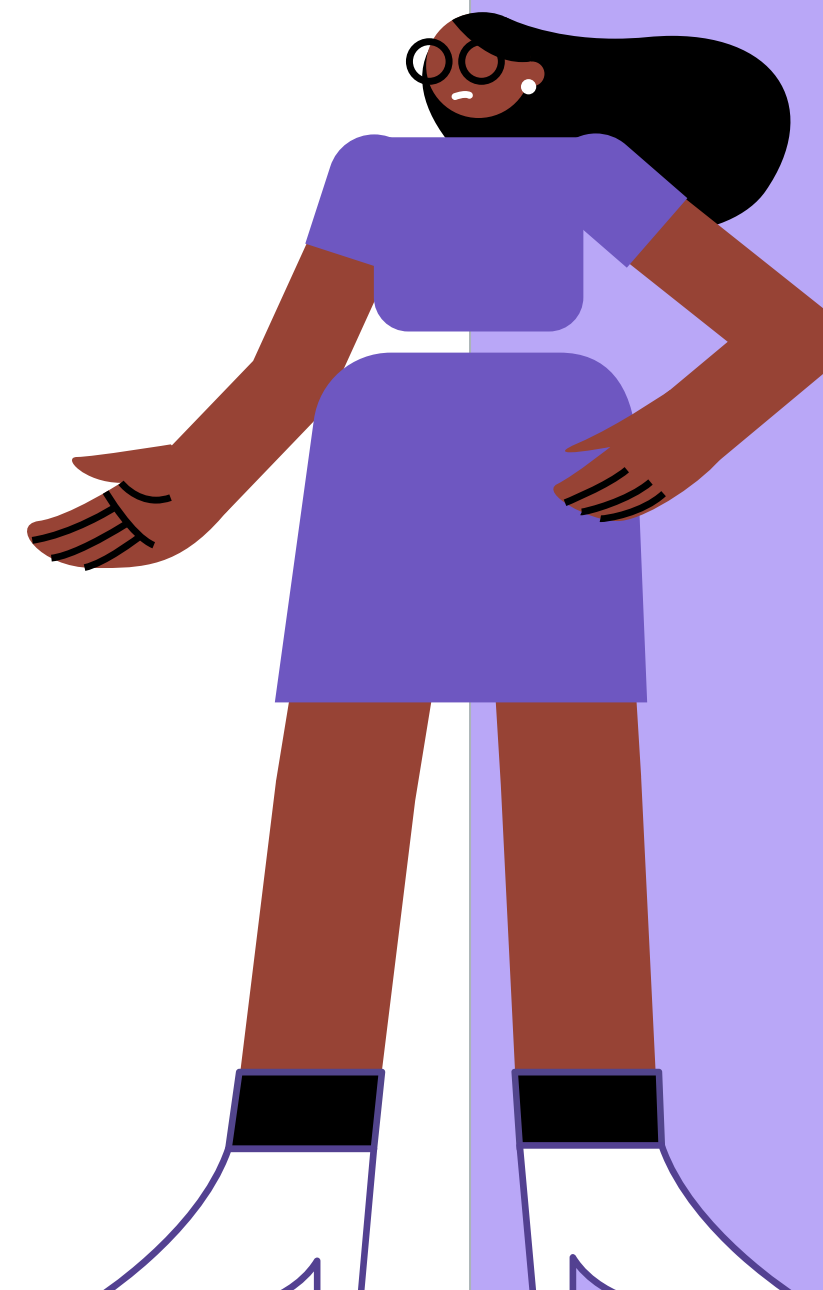
```
{ } go get -u github.com/gin-gonic/gin
```

Após adicionar a dependência, ela pode ser usada dentro dos nossos arquivos do módulo adicionado o **import**.

```
{ } import (  
    "fmt"  
    "github.com/gin-gonic/gin"  
)
```

Conclusões

Aprendemos como criar packages e importá-los, e vimos a importância da função main em um projeto. Além disso, focamos na inicialização de **modules** para o bom gerenciamento de dependências. Por fim, conseguimos usar o comando **go get** para adicionar dependências a um projeto. Tudo isso nos auxiliará a ter uma correta organização do código.



Muito obrigado!