



Go



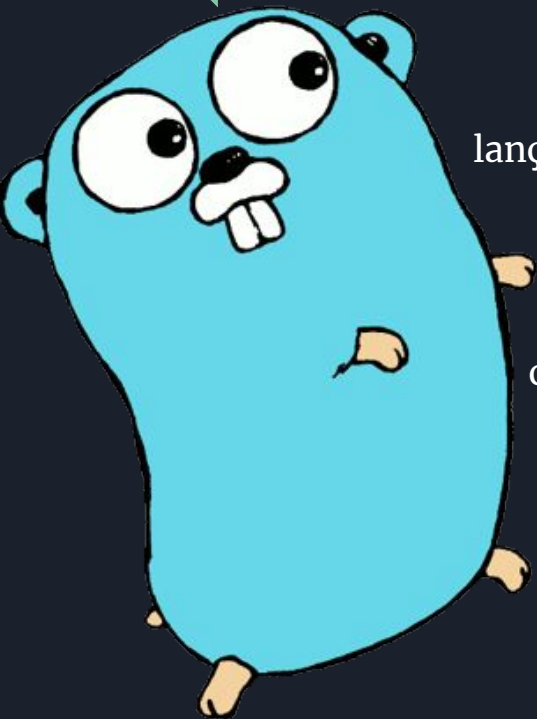
GO



LP - Marcelo Ladeira.
Samuel Carvalho dos Santos - 211010388.



História



Go é uma linguagem de programação que em 2007 foi iniciado o processo de desenvolvimento por Robert Griesemer, Rob Pike e Ken Thompson; em 2009 foi lançada oficialmente e recebe atualizações até os dias atuais com a versão 1.20 como atual.

Ela foi criada visando atender às novas necessidades de multiprocessamento que estavam aumentando, de forma mais simples, haja vista que, as linguagens como C/C++ possuíam considerável complexidade de desenvolver tais aplicações

O mascote (Gopher, esquilo) e o logotipo foram criados por Renée French



Projeto

Acerca das premissas, a sua simplicidade, sua eficiência, a sua Segurança, a sua biblioteca padrão rica e o suporte à concorrência dão os motivos suficientes para sua escolha para fazer implementações novas pois, ela foi projetada para atender esses requisitos principalmente para programas com paralelismo.

Sobre usuários do sistema (aquele que desenvolve, mantém ou administrador que trabalha diretamente com a linguagem) há uma boa quantidade de recursos, e usuários finais (cliente) é capaz de fornecer uma boa experiência; tendo assim grandes empresas inclusive no Brasil como utilizadora da linguagem como o iFood, o Mercado Livre e várias outras.

Apesar de permitir um estilo de programação semelhante à POO, ela é mais forte em implementação de programação concorrente.



Construtores

Os principais construtores são:

Arrays (estruturas que podem conter um número fixo de elementos para guardar e organizá-los.)

Slices (referências para arrays que somente se visualiza um trecho.)

Maps (tipos de dados que podem armazenar pares de chaves e valores relacionando-os.)

Goroutines (Threads ou linhas de execução independentes que podem ser executadas em paralelamente.)

Make (Se responsabiliza pela inicialização de um canal.)

Pointers (Os ponteiros são referências a outros objetos.)

Channels (canal digitado que pode enviar e receber valores através do operador "<-(Chan)).



Critérios de avaliação

.Legibilidade

A linguagem Go possui uma sintaxe mais clara e concisa que outras linguagens, o que facilita a compreensão e escrita dos códigos.

.Capacidade de Escrita

Essa forte característica em Go se deve à sua ortogonalidade e simplicidade que é dada através da facilidade de entendimento dos códigos curtos, que cumprem sua função sem necessitar de muitas linhas em relação ao Java e ao C.



Critérios de avaliação

.Confiabilidade

Se tratando de uma linguagem fortemente tipada, Go em tempo de compilação realiza as checagens, para que não ocorram bugs durante a execução do código.

.Custos e outros

Por sua semelhança com C, simplicidade e praticidade e boa documentação na comunidade, contribui para o aprendizado de desenvolvedores; por não necessitar de grandes códigos para o desenvolvimento das aplicações, menos horas serão gastas; a compilação é feita de forma rápida diminuindo seu custo.

Go é uma linguagem extremamente portátil entre sistemas operacionais; podendo ser utilizada para diversos tipos de aplicação.



Exemplos

-Exemplos de abertura de Package:

1.

`package main`

2.

`package greet`



Exemplos

-Exemplos de importação de bibliotecas:

1.

```
package main
import "fmt"
import "net"
import "github.com/yourbasic/graph"
```

2.

```
package main
import(
    "fmt"
    "net"
    "github.com/yourbasic/graph"
)
```




Exemplos

-Exemplo de implementação do if/else:

1.

```
package main
func main(){
    if 16 >= 40 {
        return true
    }
    else {
        return false
    }
}
```



Exemplos

-Exemplo para type struct:

```
package main
import "fmt"
type Pessoa struct{
    nome string
    idade int
}
func main(){
    p1 = Pessoa{
        nome: "João",
        idade: 24,
    }
}
```