

# Go para programação paralela

## SECOMP

Alexandre Medeiros

`alexandre.n.medeiros@gmail.com`

Instituto de Computação  
Universidade Estadual de Campinas

14 de agosto de 2014

Uma versão mais completa desses slides e diversos exemplos  
podem ser encontrados na minha página!

`alemedeiros.sdf.org`

# O que é Go?

## Golang

Go, ou Golang, é uma linguagem de programação *Open Source* onde é fácil fazer programas simples, confiáveis e eficientes.



# O que é Go?

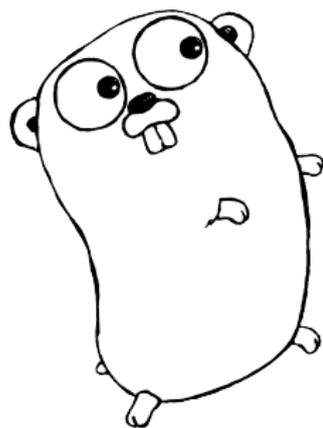
## Criadores

A linguagem foi desenvolvida por Robert Griesemer, Rob Pike e Ken Thompson no Google.



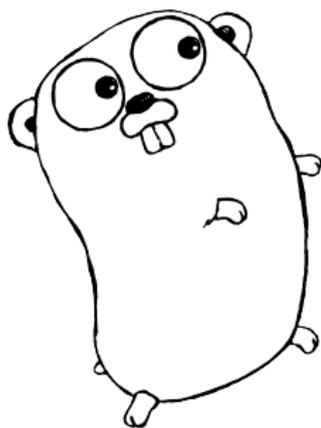
# Principais ideias

- Compilação rápida
- Execução rápida
- Simples de se programar



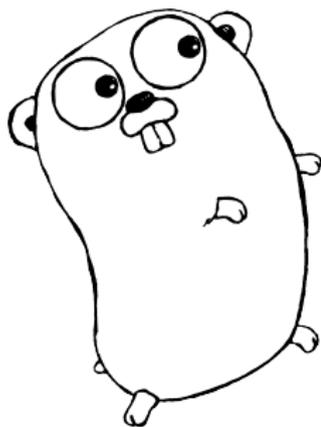
# Principais ideias

- Compilação rápida
- Execução rápida
- Simples de se programar



# Principais ideias

- Compilação rápida
- Execução rápida
- Simples de se programar



# Funcionalidades interessantes

- Sintaxe amplamente inspirada em C
- *Garbage-collected*
- Inferência de tipos de variáveis

## Funcionalidades interessantes

- Sintaxe amplamente inspirada em C
- *Garbage-collected*
- Inferência de tipos de variáveis

## Funcionalidades interessantes

- Sintaxe amplamente inspirada em C
- *Garbage-collected*
- Inferência de tipos de variáveis

# E mais importante!

Projetada para ser simples de se criar programas concorrentes.

# E mais importante!

Projetada para ser simples de se criar programas concorrentes.

# Hello, World!

```
package main

import "fmt"

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

Programa 1: hello.go

# goroutines

Uma goroutine é uma *lightweight thread*, gerenciada pelo *runtime*.

## Canais de comunicação

Os `channels` são a principal maneira de comunicação e sincronização entre `goroutines`.

# Canais de comunicação

```
func say(s string, done chan bool) {
    for i := 0; i < 3; i++ {
        time.Sleep(100 * time.Millisecond)
        fmt.Println(s)
    }
    done <- true
}

func main() {
    done := make(chan bool)
    go say("Hello ,_World!", done)

    if <-done {
        fmt.Println("Done_:_")
    }
}
```

Programa 2: channels.go

## Canais de comunicação

```
$ go run channels.go
```

```
Hello, World!
```

```
Hello, World!
```

```
Hello, World!
```

```
Done :)
```

## Conjunto de ferramentas completo

- go build
- go run
- gofmt
- godoc
- go get

## Conjunto de ferramentas completo

- go build
- go run
- gofmt
- godoc
- go get

## Conjunto de ferramentas completo

- go build
- go run
- gofmt
- godoc
- go get

## Conjunto de ferramentas completo

- go build
- go run
- gofmt
- godoc
- go get

## Conjunto de ferramentas completo

- go build
- go run
- gofmt
- godoc
- go get

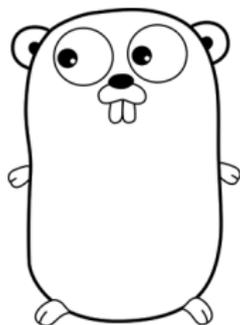
## Referências

- `tour.golang.org`
- `gobyexample.com`

# Divirta-se você também!

`play.golang.org`

Dúvidas?



Fim!

Contato: [alexandre.n.medeiros@gmail.com](mailto:alexandre.n.medeiros@gmail.com)