



BELÉM/PA, 25/02/2023

PADRÕES DE CONCORRÊNCIA EM GO



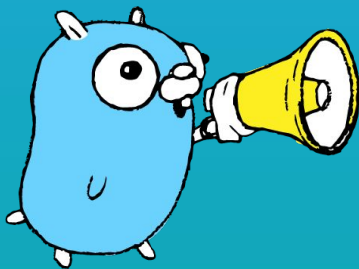
WALTER JUNIOR

Dev Back-end

Walter Junior



- Graduação em Análise de Des. de sistemas - UNAMA
- Pós-graduação em Desenvolvimento Web - Estácio
- Dev Back End (6 anos)
- Experiências
 - Cidade Aprendizagem
 - SESC/PA
 - Hostgator Brasil
 - PicPay
 - Pismo
 - Fretebras



BOM DIA!!!!

O que é Go? 01

Concorrência em Go 02

Concorrência vs. Paralelismo

Goroutines

Channels

Padrões de Concorrência 03

01

O que é Go?

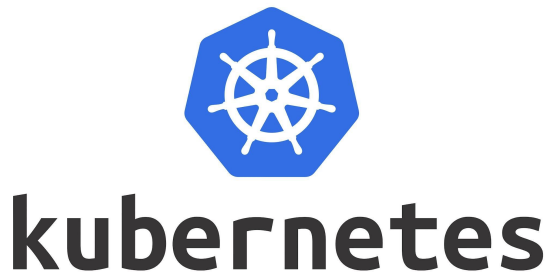
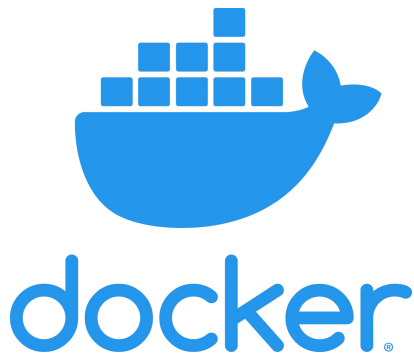
- Nasceu como um projeto interno do Google iniciado em 2007 por Ken Thompson, Rob Pike e Robert Griesemer
- Principais motivadores:
 - Longos períodos de compilação
 - Dificuldade em escalar grandes aplicações

- Os principais objetivos eram:
 - Tornar o desenvolvimento de servidores no google uma tarefa mais produtiva e eficiente
 - Buscavam criar uma linguagem expressiva, eficiente na compilação e na execução
 - Eficiente ao ponto de escrever programas confiáveis e robustos

O que é Go?



- Lançado como um projeto de código aberto em 2009
- Atualmente Go é uma linguagem de propósito geral
- Utilizada em diversos domínios de software



O que é Go?



Estrutura básica de um programa em Go

```
package main

import "fmt"

func main() {
    fmt.Println("Hello world")
}
```

- Simples
- Eficiente
- Confiável
- Produtiva
- Amigável

O que é Go?



Executando um arquivo em go

```
$ go run main.go
```

Fazendo o build arquivo em go

```
$ go build main.go
```

“

Go é uma linguagem de código aberto que facilita a criação de softwares simples, confiáveis e eficientes.

Do site go.dev

”



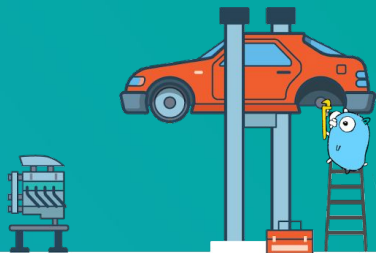
02

Concorrência em Go

- Nativa da Linguagem
- A capacidade de executar algumas tarefas de forma concorrente permite melhorar o desempenho de algumas aplicações de forma significativa

“

William Kennedy

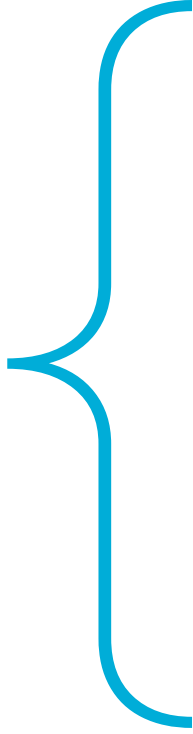


A concorrência em Go é a capacidade de as funções serem executadas de modo independente umas das outras.



Concorrência vs. Paralelismo

Paralelismo



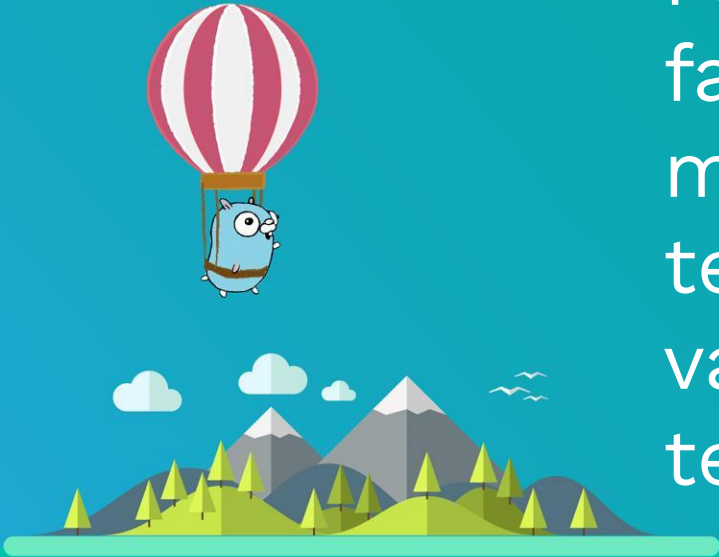
São tarefas executadas ao mesmo tempo. Só é alcançada quando porções de código são executadas simultaneamente em processadores físicos diferentes.

Concorrência

São tarefas que podem estar ou não sendo executadas ao mesmo tempo. Não é necessário mais de um processador físico para executar de forma concorrente.

“

Paralelismo diz respeito a fazer muitas tarefas ao mesmo tempo. Concorrência tem a ver com administrar várias tarefas ao mesmo tempo.



William Kennedy

”





Goroutines

Goroutines

Funções que são executadas de forma concorrente com outras funções

```
go print("Hello, World")  
go print("Olá, Mundo")
```

Para criar uma Goroutine utilizamos a palavra reservada **go**

```
go print("Hello, World")  
print("Olá, Mundo")
```

Canais (Channels)

Canais (Channels)



Canais são estruturas de dados que permitem uma comunicação segura entre goroutines

```
// Para criar um canal utilizamos a função  
// make() que recebe como parâmetro a palavra  
// reservada chan e o tipo de dado do canal
```

```
make(chan string)
```

Pode ser bidirecional (enviar e receber dados) ou unidirecional (enviar ou receber dados)

```
c := make(chan string)  
<-c // canal que envia dados  
c<- // canal que dados
```

Canais (Channels)

Canais **sem buffer** são canais sem capacidade para armazenar valores até que eles sejam recebidos

```
// Canal sem buffer  
make(chan int)
```

Canais **com buffer** são canais com capacidade para armazenar um ou mais valores até que eles sejam recebidos

```
// Canal com buffer  
// Capacidade de armazenar 10 strings  
make(chan string, 10)
```

Select



É uma instrução que permite aguardar as operações de comunicação de um ou mais canais com goroutines

```
select {  
    case msg1 := <-c1:  
        fmt.Println("recebido", msg1)  
    case msg2 := <-c2:  
        fmt.Println("recebido", msg2)  
}
```

03

Padrões de Concorrência

Generator

- É um padrão que busca encapsular a chamada de uma goroutine retornando um canal de comunicação

Multiplexing (fan-in)

- É um padrão que busca centralizar os dados de mais de um canal de comunicação dentro de apenas um canal.

Worker Pools

- É um padrão de concorrência que busca usar um número fixo de *workers* para executar várias tarefas em uma fila

- Os padrões de concorrência:
 - São soluções para resolver problemas que encontraremos ao lidar com concorrência em Go
 - Nos ajudam a escrever programas maiores e mais complexos

- Livro [“Programando em Go. Crie Aplicações com a Linguagem do Google”](#) (Caio Filipini)
- Livro [“Introdução à Linguagem Go: Crie Programas Escaláveis e Confiáveis”](#) (Caleb Doxsey)
- Livro [“A Linguagem de Programação Go”](#) (Alan Donovan e Brian Kernighan)
- Livro [“Go em Ação”](#) (William Kennedy, Brian Ketelsen e Erik St. Martin)
- Curso [“Aprenda Golang do Zero!”](#) (Otávio Gallego)
- Palestra de Rob Pike [“Google I/O 2012 - Go Concurrency Patterns”](#)
- Site oficial do Go [go.dev](#)

Obrigado!!!

