



TDC 2019 BELO HORIZONTE 13/06

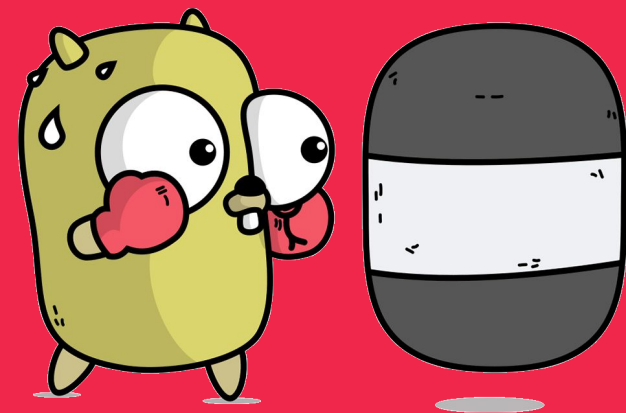
– Go for Devops



Jefferson Otoni

<http://s3wf.com>

@jeffotoni



Algumas aplicações feitas em Go 01

Características e Influências 03

Go 2018 Survey 04

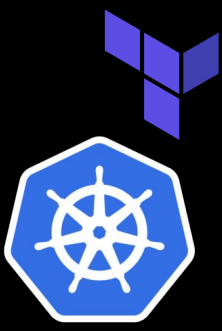
Serviços que mais crescem em Go 05

Domínios crescentes em Go 06

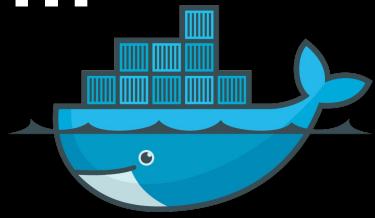
Bala de prata 07

Go além do Hello World 08

HashiCorp
Terraform



kubernetes



docker



```
***
***
****. .dPdh. _dPdhb. dPdhb.
dSP"SShdh"SShdhK. dS "dS
SS dSdhSS dS"YSSdh. dS dS
Ydb. dSYdD. .dP" XdPdh dS
"YdSS "YdP" dSdP"dS dS
dS
Ydb dSP
"YdP"
```



 Prometheus



Cockroach DB



Istio

 Grafana

 Dgraph



 **moby**
project

Características



- Compilada e estática
- Gc (Garbage Collector)
- Paradigma Concorrente
- Tipagem estática
- Semântica é clara
- Sintaxe é limpa
- É de uso Geral
- Plataformas: Windows, Linux, Mac e FreeBSD





INFLUÊNCIA

| | |
|--------------|--------|
| Occam | [1983] |
| Oberon | [1986] |
| Erlang | [1986] |
| Newsqueak | [1988] |
| Cocurrent ML | [1993] |
| Alef | [1995] |
| Limbo | [1996] |
| Go | [2009] |





Go 2018 Survey



API / RPC e CLI

Serviços que mais crescem em Go



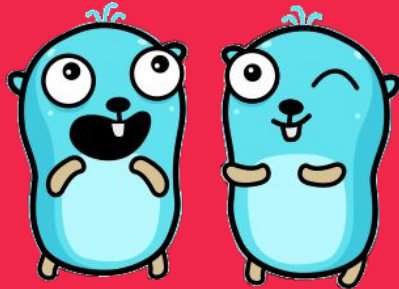
WEB & DevOps

65% 41%

Domínios que mais crescem em Go



Bala de **PRATA**
Não Existe





GO além do
“Hello World...”



—

Simplicidade...

confiabilidade



```
package main

import (
    "fmt"
)

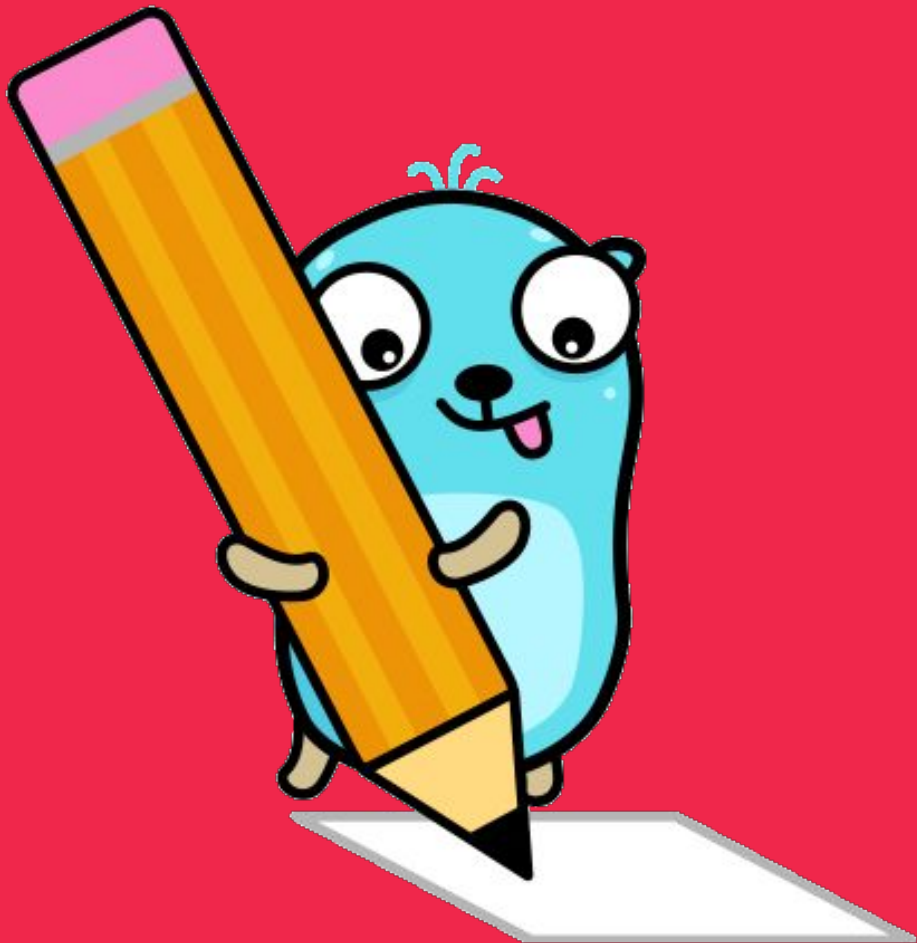
func main(){
    fmt.Println("Hello, World!")
}
```

GOPHER



—

Legibilidad...

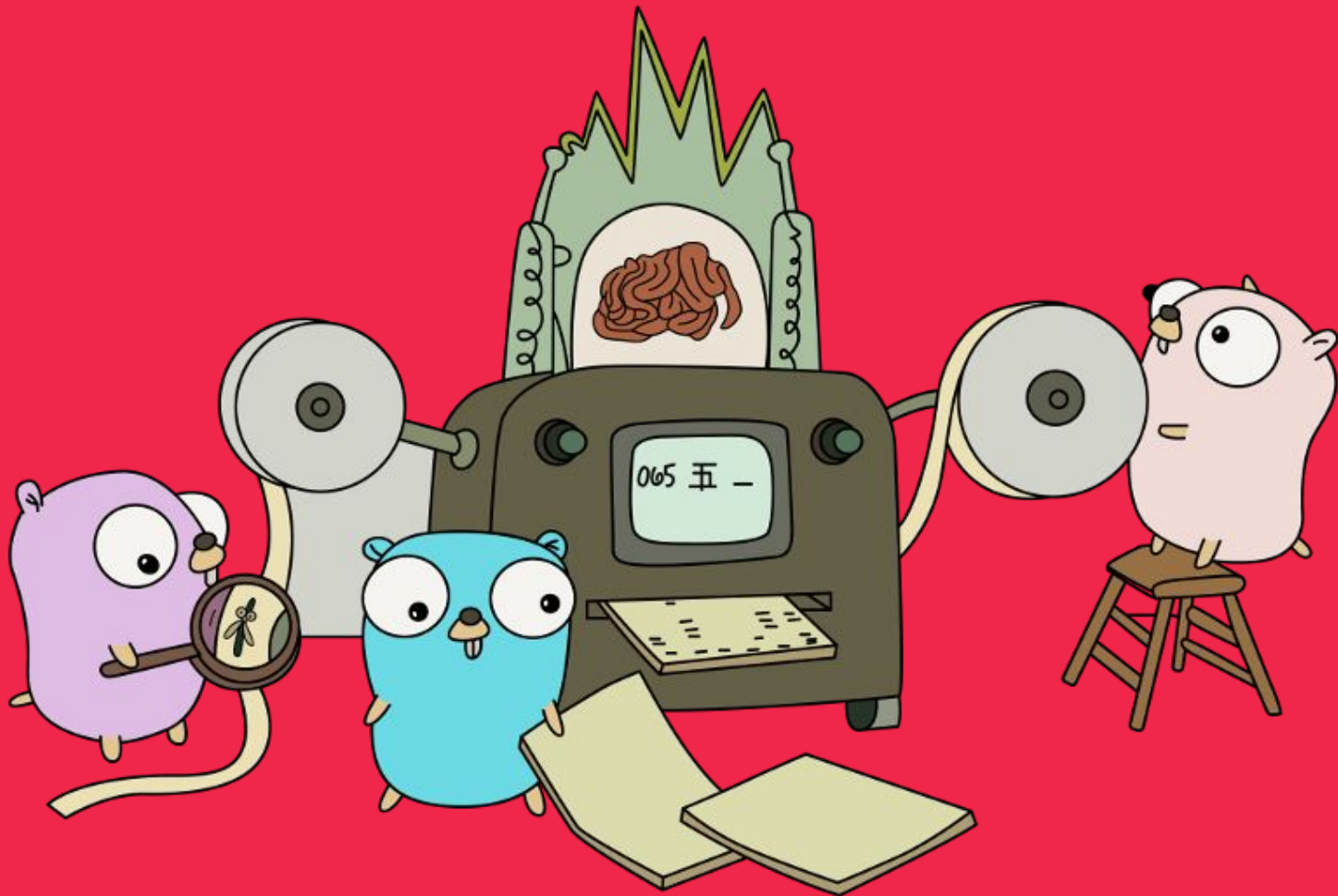


—



—

Produtividade...





—

Desempenho e
simultaneidade
concorrência



Menos é
exponencialmente **MAIS**



MAIS FÁCIL

- Aprender
- Entender
- Implementar
- Depurar
- Ajustar
- Evoluir





Polêmicas



GENERIC

?!





— Goroutine e não Threads



Ruptura/quebra
do paradigma



Goroutines

1

2

·

·

·

M

Dijkstra

Go scheduler

Threads

1

2

·

·

·

N





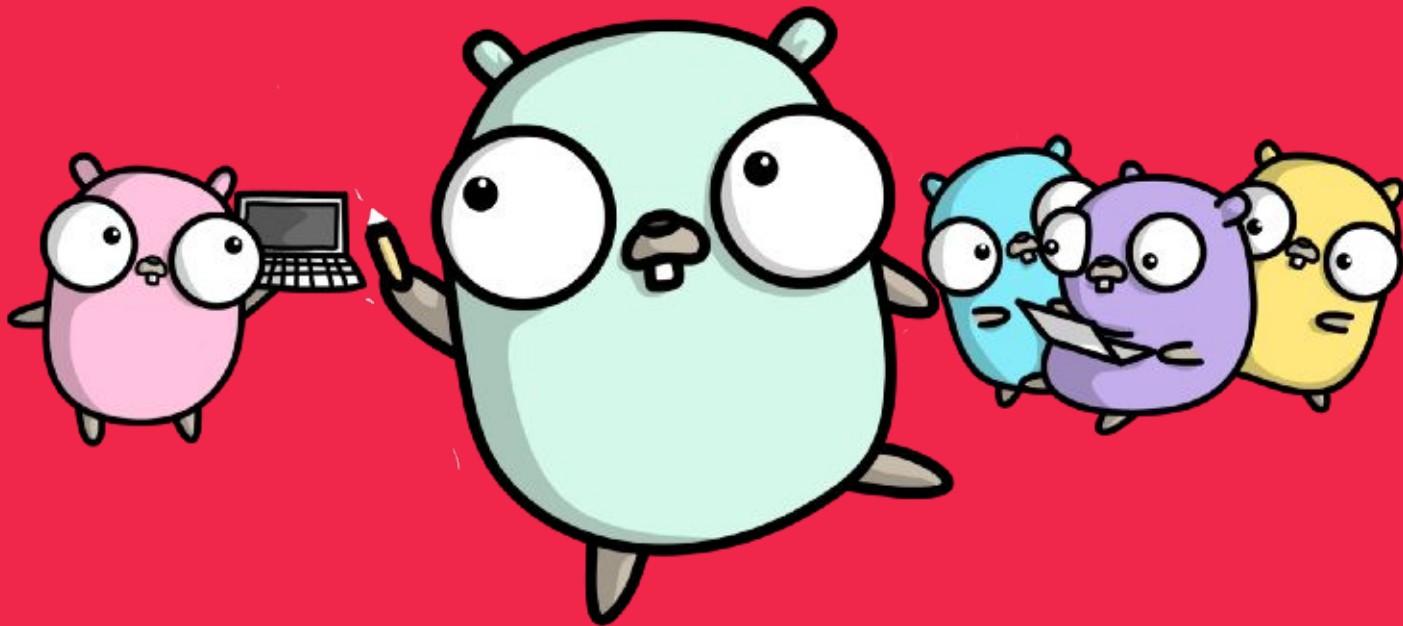
—
Concorrência

!=

Paralelismo



— Partner



Channel / Fan-out



```
func main() {  
  
    jobs := make(chan int, 10000)  
    results := make(chan int, 10000)  
  
    for w := 1; w <= 1000; w++ {  
        go worker(w, jobs, results)  
    }  
}
```

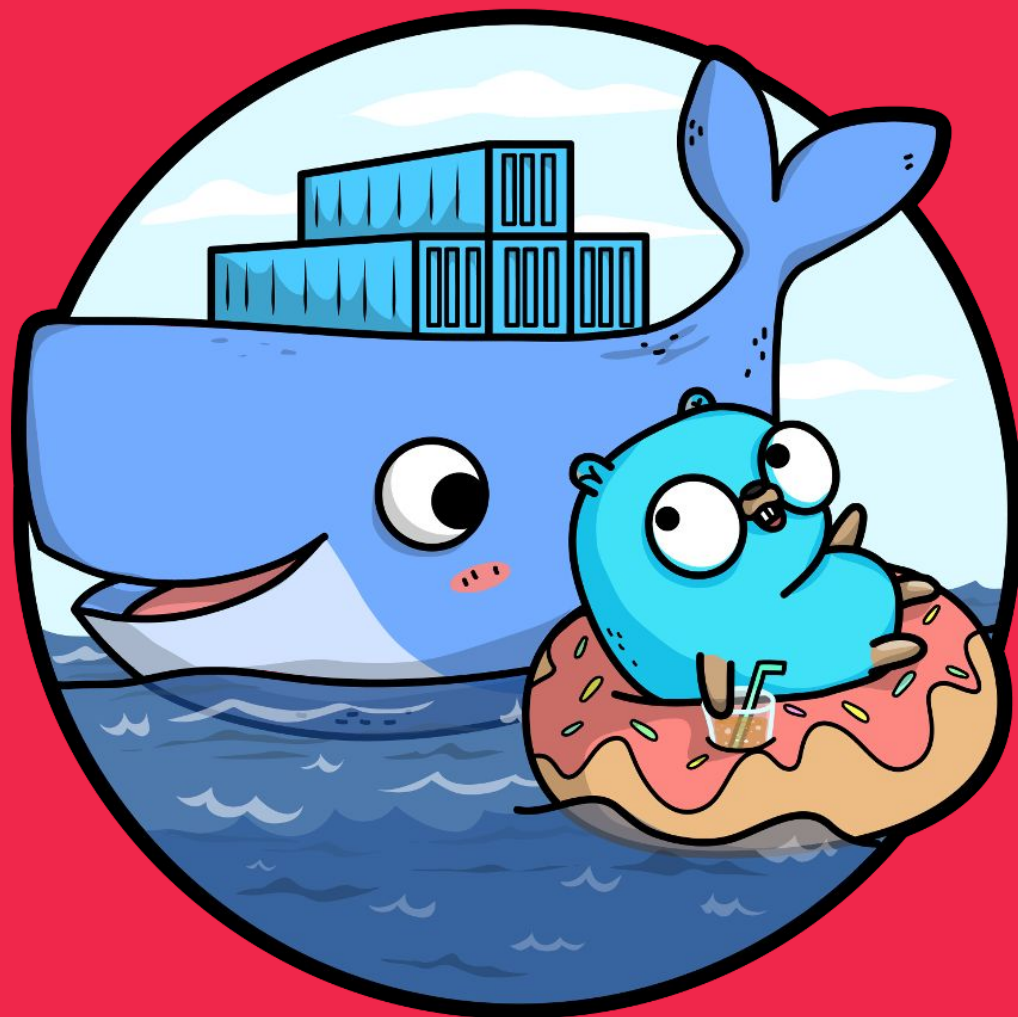


```
func fanIn(input1, input2 <-chan string) <-chan string {  
    c1 := make(chan string)  
  
    go func() {  
        for {  
            c1 <- <-input1  
        }  
    }()  
  
    go func() {  
        for {  
            c1 <- <-input2  
        }  
    }()  
    return c1  
}
```

Fan-in



Docker e Go





```
func Hello(w http.ResponseWriter, r *http.Request) {
    w.WriteHeader(http.StatusOK)
    w.Write([]byte("Hello, welcome to the world, Go!"))
}

func main() {

    mux := http.NewServeMux()

    mux.Handle("/api/hello", http.HandlerFunc(Hello))

    server := &http.Server{Addr: ":8080", Handler: mux}

    fmt.Printf("Server Run port: 8080\n")
    if err := server.ListenAndServe(); err != nil {
        log.Printf("Error while serving metrics: %s", err)
    }
}
```



```
#####  
# Dockerfile distroless  
#####  
FROM golang:1.12.0 as builder  
WORKDIR /go/src/main  
COPY . .  
RUN go install -v ./...  
  
#####  
# STEP 2 build a small image  
#####  
FROM gcr.io/distroless/base  
COPY --from=builder /go/bin/main /  
CMD ["/main"]
```




Referencias

<https://github.com/avelino/awesome-go>

<https://github.com/golang-devops>

<https://blog.golang.org/survey2018-results>

<https://github.com/jeffotoni/goexample/blob/master/api-simple/api3.go>

<https://medium.com/@jeffotoni/golang-simplificando-a-complexidade-o-inicio-145371d67711>

<https://github.com/MariaLetta/free-gophers-pack>

https://dave.cheney.net/practical-go/presentations/qcon-china.html#_simplicity

<https://github.com/avelino/awesome-go>



λ Thanks := &Origado{...}

Obrigado...



github.com/jeffotoni



[instagram.com/jeffotoni](https://www.instagram.com/jeffotoni)



[linkedin.com/in/jeffotoni/](https://www.linkedin.com/in/jeffotoni/)



https://t.me/go_br



<https://t.me/awsbrasil>



<https://t.me/devopsbh>

