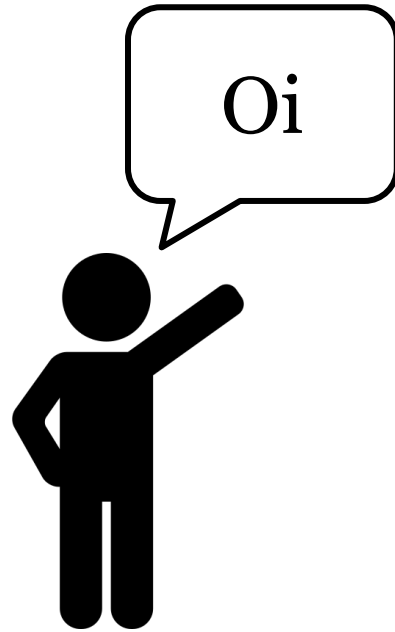


Infraestrutura Imutável

~~



Roney Gomes
@umbravirtual

Do que se trata essa apresentação:

- ❑ sobre a relação entre bugs e acúmulo de estado em aplicações
- ❑ sobre como infraestrutura e o acúmulo de estado se relacionam
- ❑ como as lições aprendidas no processo de resolver o primeiro problema influenciam na solução do segundo

Do que não se trata essa apresentação:

- ❑ sobre balas de prata
- ❑ sobre kubernetes ou soluções empacotadas de infraestrutura
- ❑ sobre tentar resolver os mesmos problemas que Google ou Spotify

Debugar:

- ❑ entender a sequência dos passos tomados pela aplicação
- ❑ entender como o estado se altera entre os passos
- ❑ se perder entre um passo e outro e começar tudo de novo:
 - ❑ muita informação pra se manter ao mesmo tempo na cabeça
 - ❑ o estado foi alterado por algum agente concorrente
 - ❑ um único passo alterou o estado significativamente

Debugar:

- ❑ entender a sequência dos **passos** tomados pela aplicação
- ❑ entender como o **estado** se altera entre os **passos**
- ❑ se perder entre um **passo** e outro e começar tudo de novo:
 - ❑ muita **informação** pra se manter ao mesmo tempo na cabeça
 - ❑ o **estado** foi alterado por algum agente concorrente
 - ❑ um único passo alterou o **estado** significativamente

Mutabilidade

=

Acúmulo de Estado

Programação Funcional & Imutabilidade:

[ ,  , ] + 

Programação Funcional & Imutabilidade:





❏ estado:

[, , ]

❏ operação:

+

Programação Funcional & Imutabilidade:

```
[ , ,   
].append()
```

Programação Funcional & Imutabilidade:

[, , , ]

Programação Funcional & Imutabilidade:

[ ,  , ] ++ 

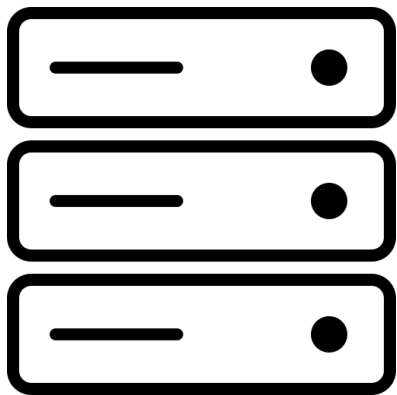
Programação Funcional & Imutabilidade:

[ ,  , ]

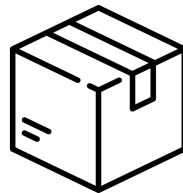
[ ,  ,  , ]

O que isso tem a ver com infraestrutura?

Infraestrutura Mutável:

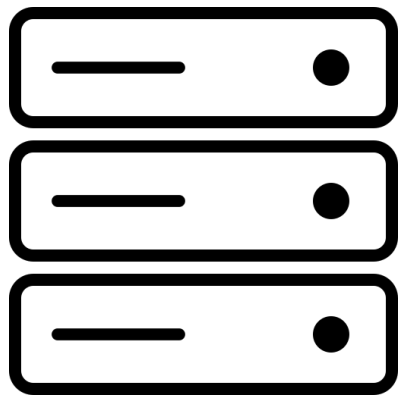


+

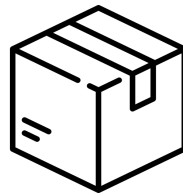


v1

Infraestrutura Mutável:



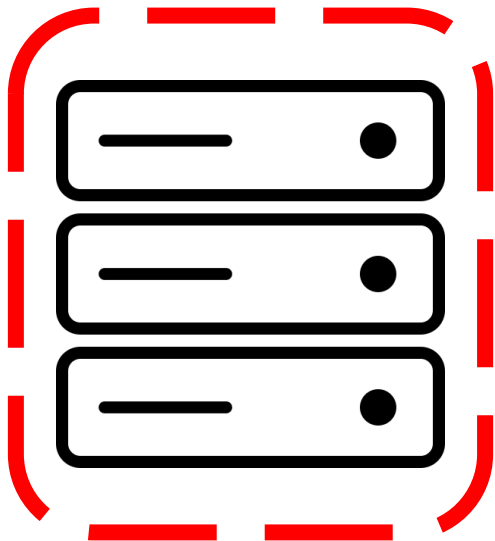
.append(



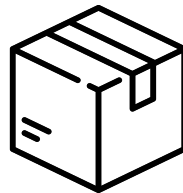
)

v2

Infraestrutura Mutável:



+

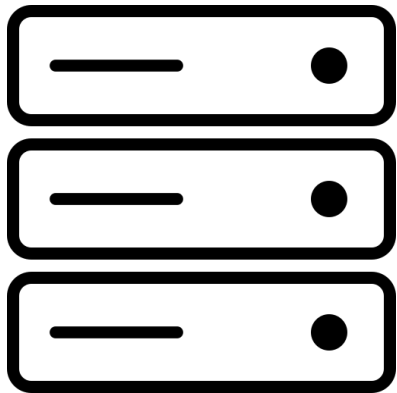


v2

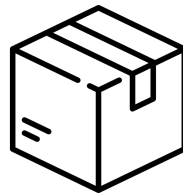
Infraestrutura Mutável:

- ❑ *drift* de configuração
 - ❑ desvio da configuração base
- ❑ servidores flocos de neve
 - ❑ difíceis de reproduzir devido inúmeras alterações
 - ❑ péssimos em cenários de falhas
 - ❑ difíceis de atualizar
- ❑ deployments complexos e vagarosos
- ❑ operações sempre apagando incêndios
- ❑ escalabilidade

Infraestrutura Imutável:

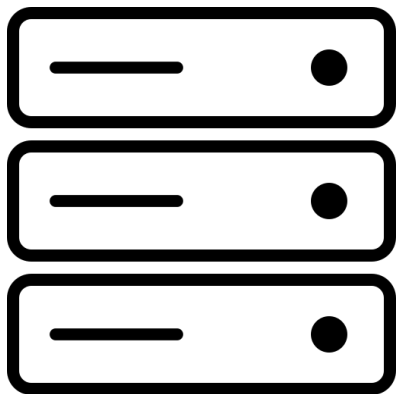


+

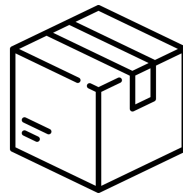


v1

Infraestrutura Imutável:

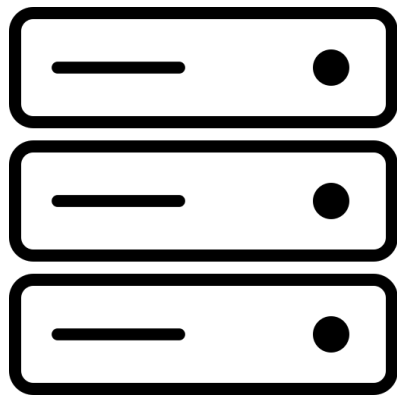


++

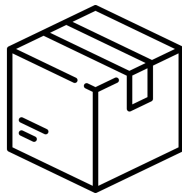


v2

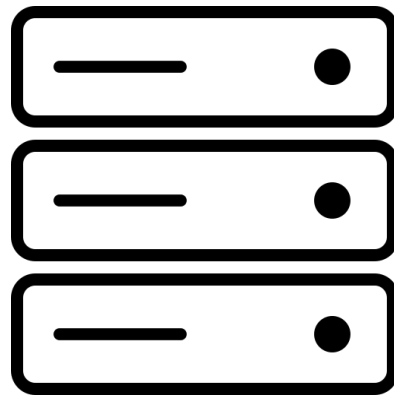
Infraestrutura Imutável:



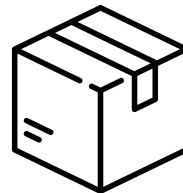
+



v1



+



v2

Infraestrutura Imutável:

- ❑ operação simplificada
- ❑ deployments contínuos com menos falhas
- ❑ mitigação de erros e ameaças
- ❑ redução de custos

Padrões de Infraestrutura Mutável:

- ❑ uma vez de pé, instâncias não são modificadas
- ❑ substitua instâncias para atualizá-las
- ❑ não deixe suas instâncias rodando por muito tempo
- ❑ escale suas instâncias sob demanda

Deployments Imutáveis:

- ❑ a partir de imagens base
- ❑ com docker

Deployments Partindo de Imagens Base:

- ❑ empacotamento da imagem base com a aplicação
- ❑ uma nova imagem base gerada a cada deployment
- ❑ processo demorado
 - ❑ muitas camadas de complexidade até chegar na aplicação
- ❑ uma imagem base pra cada tipo de aplicação
 - ❑ java, python, node etc.

Deployments Com Docker:

- ❑ a aplicação não faz parte da imagem base
 - ❑ imagem simplificada com runtime docker
- ❑ uma nova imagem docker gerada a cada deployment
- ❑ imagens docker são geradas muito mais rapidamente
- ❑ uma única imagem base pra todos os tipos de aplicação
 - ❑ ~~java, python, node etc~~ docker

Referências:

- ❑ STELLA, Josha. **Immutable Infrastructure**. 2016.
- ❑ BLANK-EDELMAN, David N. **Seeking SRE**. 2018.
- ❑ FOWLER, Martin. **PhoenixServer**.
 - ❑ <https://martinfowler.com/bliki/PhoenixServer.html>
- ❑ FOWLER, Martin. **SnowflakeServer**.
 - ❑ <https://martinfowler.com/bliki/SnowflakeServer.html>
- ❑ Imagens da apresentação extraídos de <https://flaticon.com>.

Obrigado!