
De código estável a software estável

— Implantando DevOps sem
saber o que era DevOps —



Jefferson Quesado

Sobre

- Jefferson Quesado
- Ciência da Computação
 - UECE 2012
- TotalCross/SoftSite (2013 - 2017)
- SoftSite (2017 - ...)

O que é DevOps?

- Não sabia na época
- Pula

2013, aplicações

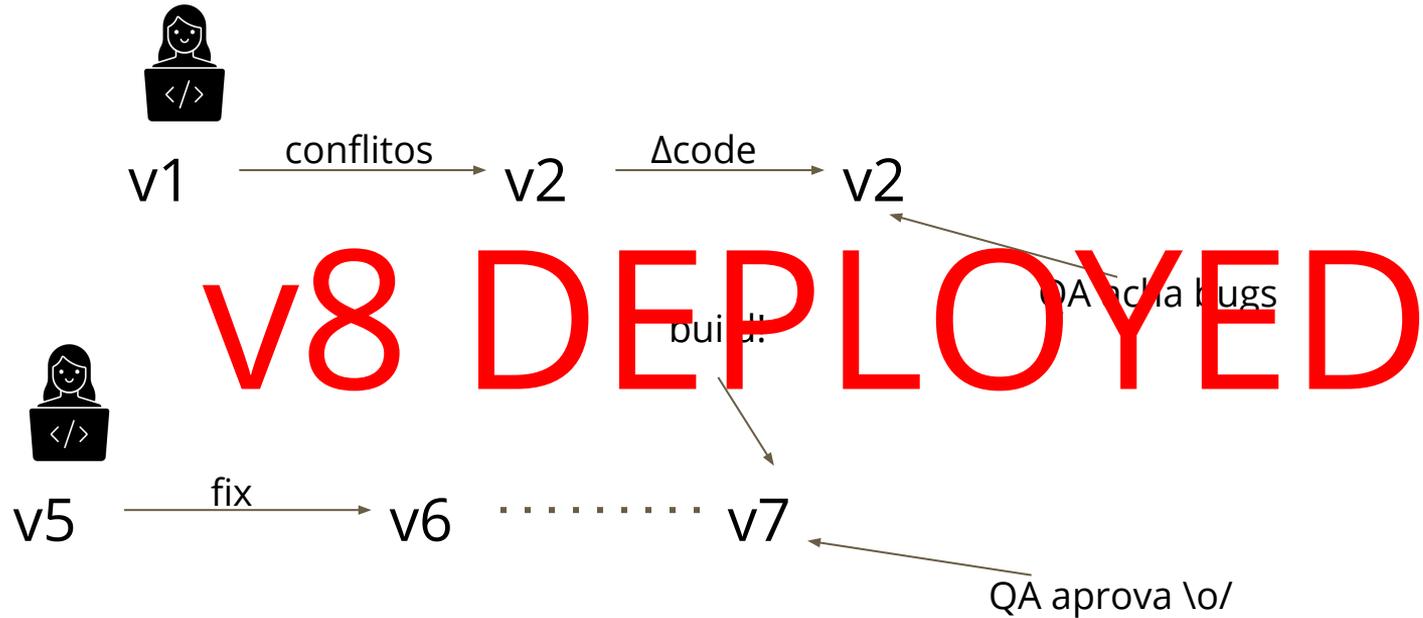
- Portal
 - GWT
- Mobile
 - TotalCross
- SyncServer*
- Import*,
WebService*,
WebService Client*

*: magia, só configuração, sem manutenção nem desenvolvimento

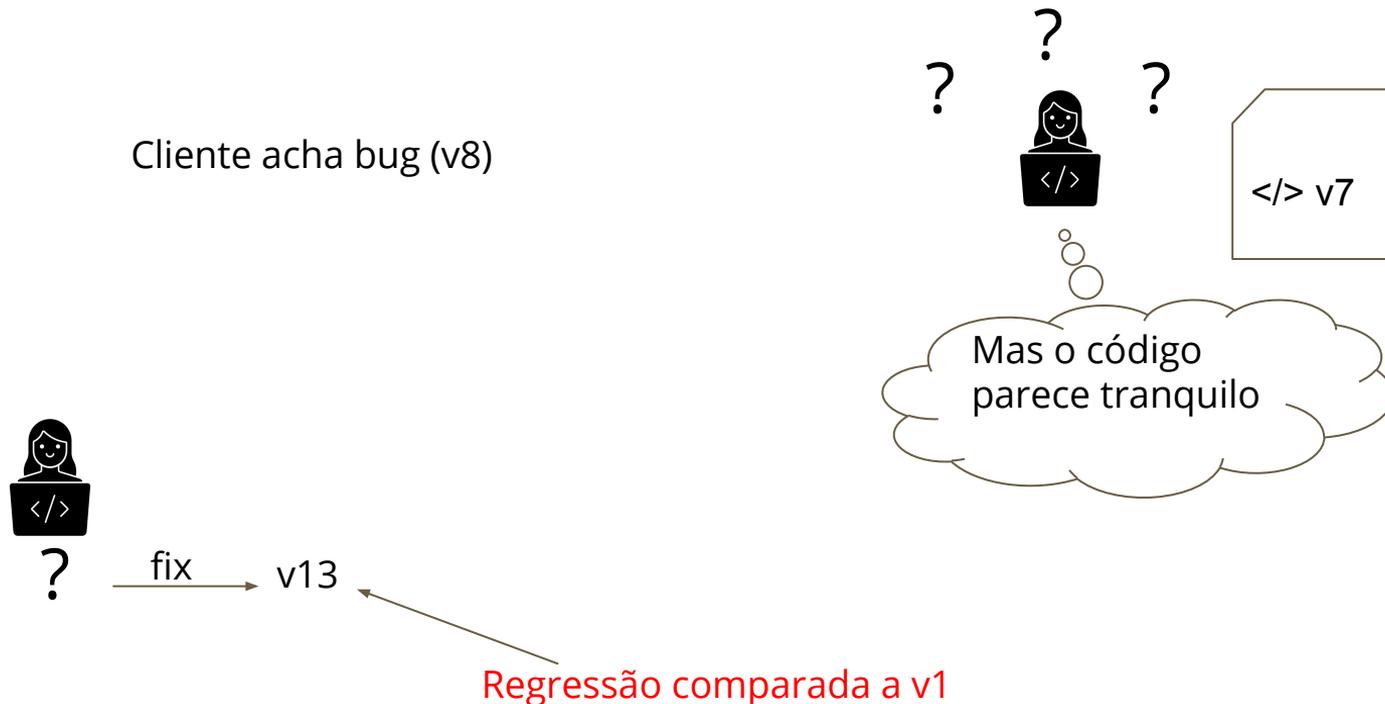
2013, ferramental (dev/build)

- SVN -> controle de código
- Ant -> empacotar o portal
- Jenkins -> sacred build
 - TotalCross instalado, certificados iOS, Java instalado

2013, pipeline de dev até ops (ou seria oops?)



2013, pipeline de dev até ops (ou seria oops?)



2013, pipeline de dev até ops (ou seria oops?)

QUAL VERSÃO DEPLOY?



fix fix fix

v18

QA testa v21...



fix fix fix

v30

v31

QA

v32 ok, cliente pode homologar

Cliente

homologar a v37 conforme requisitado, tá boa

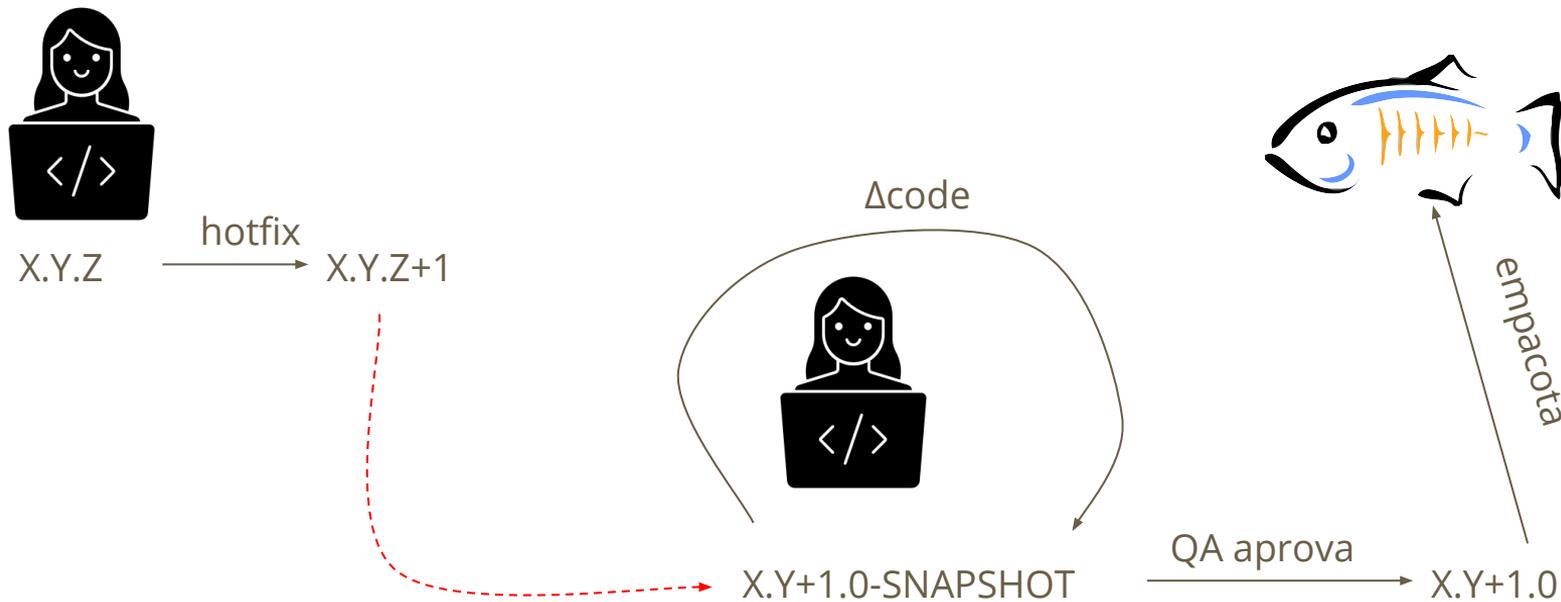
Ciclo desde v1 durou entre 6 a 8 meses

V42!!!!

Primeiras mudanças, portal

- Mavenização
 - Deps livres, build funfa
- Git
- GitFlow
- Code review
- GitLab (bitnami) interno
- SemVer--
- Plugin Jenkins + Maven
- Merge Request => webhook Jenkins
- Artifactory

Ciclo deploy, portal



Enquanto isso, no mobile...

- Configuração versão X cliente
 - SyncServer
 - O grande único XML
 - As pastas com as “persistências”
- Disponibilização APK via URL
- Necessidade de receber informações sem full sync

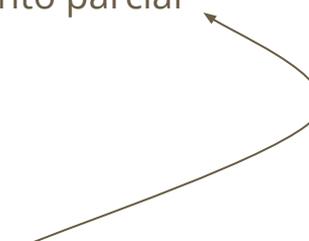
“Chega cá, Ops, deixa titio Dev ajudar”

Única manutenção no SyncServer desde 2013

- Relacionamento versão X cliente
 - Via tabelas SGBD
 - Adeus mexer no grande único XML
- Banco de ADM para conexões
 - Já existia para o portal ̄_(_ツ)_ /
- Persistência única da versão
 - Personalizável por cliente
- Framework/protocolo de extensão SyncServer
 - Recebimento parcial
 - Help *

Overengineering

Isso aqui tá ok



Situação fim 2014

- Mobile (SVN): ok
- Portal (git): melhor que 2014 *

*: atrasado funcionalidades comuns com o mobile

- Atraso estimado: 12 H.A.
- Equipe mobile: ~2 pessoas**
- Equipe portal: ~10 pessoas**
- Equipe ambos: ~5 pessoas**

** : número metafórico

Situação fim 2014

Merge request

- Qualidade de código
- Software estável
 - ... exceto develop => master 😄...
- Velocidade de lançamento?
 - 6-8 meses

Continua igualmente lento!!



New year resolutions

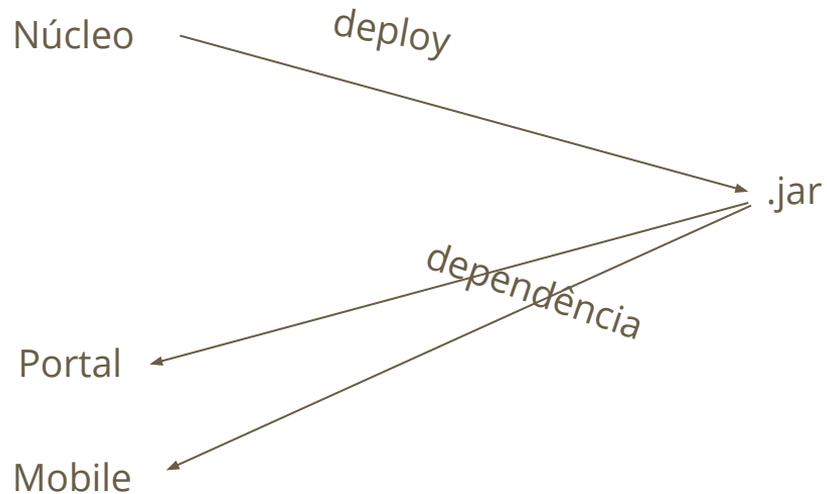
- Se é a mesma coisa, como juntar esses códigos?
- Dependency inversion!
 - Long live DAO
- mobile/portal => cascas
 - UI
 - interfaceamento com DB
- Mobile => git (março)
- Núcleo (maio)
 - RC junho
 - “Estável” outubro



Revisitando o build

Jenkins

Artifactory



Resultados 2015

- Mobile
 - 3.6.0 => 3.11.3
- Lançamentos 3 m...3m
- Long live Núcleo
- Estratégia de freeze:
 - Testado: mobile 3.7.0
 - AKA release-candidate
 - AKA RC
- Primeiros testes unitários tímidos

Explosão dos testes

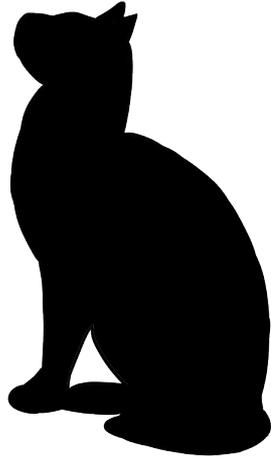
- Jul/2016
- Viva o núcleo!!!
 - Núcleo: =D
 - Mobile: =)
 - Portal: =(
- Meia dúzia => centenas
- Δcode Tributação?
 - TDD
- Taxonomia/cladística de testes

Minguou trimestre seguinte

Legado da época continua vivo
Validações ainda importantes

“Dev? O banco subiu no telhado...”

- Contorno para dúzias de milhares
 - Ainda penoso
- Falência total centenas de milhares



- Import, WS, WSC, Claudus no limite do modelo de integração

Consumo do banco basicamente coisas de integração e consultas do SyncServer

Sincronismo? TamoJunto

- Protocolo simples de transferência de dados
 - Gramática regular*
 - SAX
 - Morte à persistência/SyncServer**
-
- SQL Server => mobile
 - Possibilidade de suplantar integração

Morte ao merge hell!

- # Versão unificado
- Monorepo
- gitlab.com
 - Sim, meu bitnami morreu 2X

Master/develop por cliente?
Estratégia falida abandonada a anos...

Profusão de executáveis

- Tejotas...
 - Erp-importer
 - Erp-integrator
 - MidPlayer
- Jasperreports?
 - Derrubava o portal
 - Serviço externalizado
 - JRXMLS x Serviço

O build morreu!

Vida longa ao build!

“Coé, Fábio”

- TotalCross#184
- TotalCross#214
- TotalCross#220
- TotalCross#221
- TotalCross#274
- TotalCross#275
- TotalCross#403
- TotalCross#405
- TotalCross#407
- TotalCross#410
- TotalCross#411
- TotalCross#412
- TotalCross#413
- TotalCross#417
- TotalCross#439
- TotalCross#445
- TotalCross#464
- TotalCross#478
- TotalCross#481
- TotalCross#482
- TotalCross#483
- TotalCross#519
- TotalCross#526
- TotalCross#529
- TotalCross#536
- TotalCross#577
- TotalCross#583
- TotalCross#584
- TotalCross#590

O build morreu!

Vida longa ao build!

“Coé, Fábio”

- Headless => NPE
- StackOverflow em Linux headless
- FS case sensitive
- Formato da URL download SDK
- Sugestão para early fail
 - Meus minutos!!
- Por que paradas que caíram em desuso ainda no SDK?

O build morreu!

Vida longa ao build!

Jenkins

- Java
- Maven (plugin)
- TotalCross
- Certificados iOS
- Comandos de build

GitLab-CI

- Imagem Docker
- Imagem Docker
- Download
- Download seguro, shred
- Atributo "script" de job



Mark Hamill



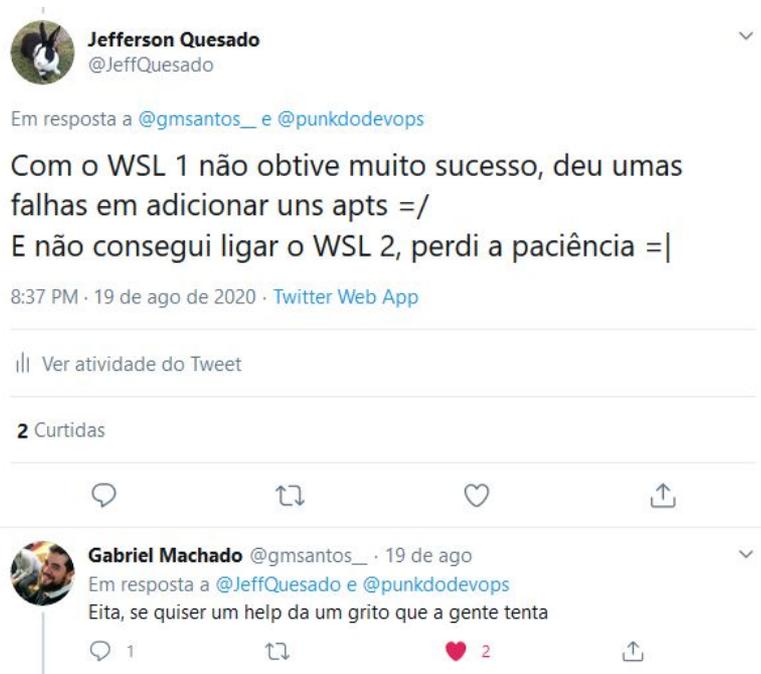
Mark Yaml

Máquina de build morreu!

Vida longa à máquina de build!

Por que mudar?

- Host parou
 - Ressuscitou depois
- Build vs Versionamento
- Give me metal and it'll run
 - Se pagou em julho/2020
- Escala
- Testar compilação por push*



TotalCross downloader

```
1 #!/bin/bash
2
3 get-major-minor() {
4     sed -E 's/^[^.*]\.[^.*]*\./\1/' <<<"$1"
5 }
6
7 amazon-tc-url() {
8     version="$1"
9     major_minor=`get-major-minor "$version"`
10
11     amazon_base=https://s3-us-west-2.amazonaws.com/totalcross-release
12     amazon_defacto=${amazon_base}/${major_minor}/TotalCross-${version}.zip
13
14     echo $amazon_defacto
15 }
16
17 echo "link para download: " `amazon-tc-url 4.2.0`
```

url-download-tc.sh hosted with ❤️ by GitHub

[view raw](#)

```
1 #!/bin/bash
2
3 curl -s https://maven.totalcross.com/artifactory/rep01/com/totalcross/totalcross-sdk/${infor
4     grep value |
5     head -n1 |
6     sed -E 's/^[^<]*>([<]*<).*$/\1/'
```

guess-tc-snapshot.sh hosted with ❤️ by GitHub

[view raw](#)

Maven-metadata.xml:
metadata.versioning.snapshotVersions.snapshotVersion[0].value



jeffque / totalcross-downloader.sh

Last active 17 months ago

[Code](#)

[Revisions](#) 2

tc-compiler-help

1. criar a biblioteca externa
 2. adicionar o plugin do TC no POM da dependência
 3. não colocar a tag `<platforms>` nas configurações do plugin
 4. adicionar `Lib` no final da tag `<name>` nas configurações do plugin
 5. rodar `TotalCross Package` pelo VSCode
 - alternativamente chamar o `tc.Deploy` por qualquer outra IDE
 6. criar o arquivo `all.pkg` dentro do projeto do seu aplicativo
 7. colocar `[L] /absolute/path/myLib.tcz` no `all.pkg`
- 3 bibliotecas externas ao projeto de código fechado
 - 1 biblioteca externa ao projeto de código aberto

tc-compiler-help

tc-compiler-help 

An easy way to help building TotalCross via Maven Forked from [GitHub](#)



O build/a publicação mobile



Antes

- Deploy núcleo => artifactory
- TC app
- Upload S3 manual
- Adição persistência manual
- Cliente x versão manual

v1

- `cd nucleo; mvn install; cd ..`
- `tc-compiler-help`
- Upload S3 manual
- Adição persistência manual
- Cliente x versão manual

vBleeding-Edge

- `./mvnw install -am -pl :Nucleo -P retrolambda`
- `tc-compiler-help`
- Upload S3 auto
- Adição persistência auto
- Cliente x versão manual

O build/a publicação portal

Antes

- Deploy núcleo => artifactory
- mvn package
- Download
- Para Glassfish
- Reza pra funfar

v1

- cd nucleo; mvn install; cd ..
- mvn package
- Download
- Para Glassfish
- Reza para funfar

vBleeding-Edge

- mvn package -am -pl :NuPortal
- GitLab-CI job manual
- Backup versão prod
- Para Docker img NuPortal antigo
- Sobe Docker img NuPortal novo

2013

- Restaur
- scriptão



Issue Tracker da Galera @devscansados · 31 de jul

O que significa BDSM ?

Só respostas erradas relacionado a TI

135

40

116



Jefferson Quesado

@JeffQuesado

Em resposta a [@devscansados](#)

Banco de dados sem migração

7:30 PM · 31 de jul de 2020 · [Twitter for iPhone](#)

17 Curtidas



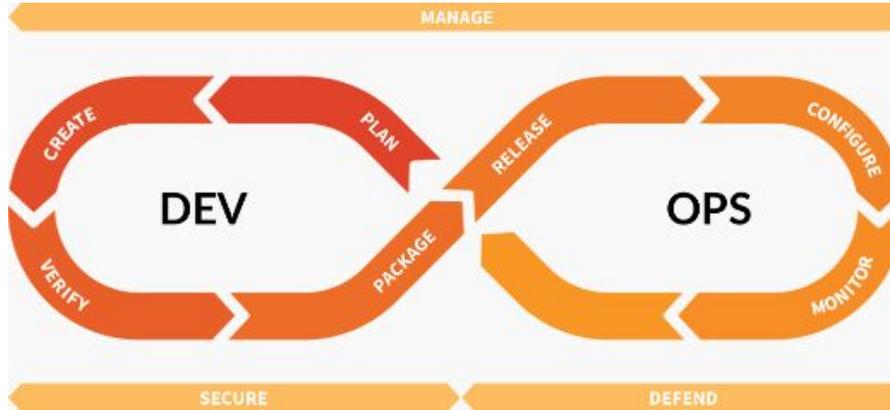
iseline (+)

roduzível
ão fácil do ponto

hema history
lha leitura

codings bizarros...

O que é DevOps?



- Cultura/filosofia
- Colaboração Dev + Ops
- Empatia

Preciso de DevOps?

O que referências pensam do assunto?

 **Gomex @ 127.0.0.1** @gomex · 31 de jul
Eu concordo! O DevOps tem que acabar, vamos falar sobre isso?

 **Issue Tracker da Galera** @devscansados · 31 de jul
Agora sem polemica, algo que todos concordam é que DevOps tem que acabar né?
[Mostrar esta sequência](#)

5 5 15

 **Gomex @ 127.0.0.1** @gomex

Então, como qualquer movimento de transformação seu objetivo é o fim de si próprio! Quando um time já atua em colaboração entre equipes e a comunicação é ampla, não precisaremos de DevOps, com isso ainda é sonho pra maioria dos lugares, ainda precisa muito disso ;)

6:41 PM · 31 de jul de 2020 · [Twitter for iPhone](#)

1 Retweetar 20 Curtidas

5 5

 **Principe do Oceano** @rponte

uma tarefa q tenho discutido bastante nas últimas semanas é sobre **abraçar o luxo** de ter não determinados problemas. Se não existe o problema então não vamos complexar a solução.

exemplo: se seu deploy permite janela então **abraçe esse luxo** e não se preocupe com zero-downtime.

12:59 PM · 22 de ago de 2020 em [Fortaleza, Brasil](#) · [Twitter for iPhone](#)

10 Retweets e comentários 71 Curtidas

 **Principe do Oceano** @rponte · 22 de ago
Em resposta a [@rponte](#)
se vc pode vc pode evitar problemas de concorrência a nível de infra então não precisa sofrer na implementação do código;

se vc não existe pré-requisitos de performance/escala então não te preocupa com isso agora.

3 1 20

Preciso de DevOps?

- Sem filosofia/cultura?
- Confiabilidade do software?
- Overengineering ou necessidade?
- Qual o risco/complexidade de Δ code?
 - dev => ops
- Qual o custo do deploy?

RESUMÃO:

- Simplificação?
- Custo?
- Confiança?
- Velocidade?

Experiência pessoal

Git/merge request

- Estabilização produto
- Confiança deploy
- Melhor qualidade
 - Menor custo
- Tempo conflito
 - Late merge strategy
 - Sandbox development

Mavenização

- Simplificação build
- Simplificação deps
- tc-compiler-help

Núcleo

- Custo dev
- Testabilidade
- Portal atraso técnico
 - “headstart”

Experiência pessoal

GitFlow

- **Confiança**
 - Estabilização RC
 - Sandbox develop
- **Confiança deploy**
- **Tempo conflito**
 - Long lived branch?
 - “Meta” de $|\Delta\text{code}| \leq 600$ linhas

MonoRepo

- **Simplificação revisão**
- **Simplificação build**
 - Infraestrutura
 - Processo de build

GitLab-CI

- **Ambiente de build**
 - Simplificado
 - Mais confiável*
- **Script build**
 - Acoplado à versão
 - Simples alterar
- **Deploy confiável***

Experiência pessoal

Parceria TotalCross

- Exequibilidade
- Build mais rápido
 - Menor custo build

Migração Banco

- Δ code confiável
- Aplicação
 - Confiável
 - Simplificada
 - Auditável

Dúvidas?



Obrigado!



Jefferson Quesado



@jefferson.quesado



@JeffQuesado



@jeffque