

KEYRUS

# Desafios com Onboarding em Grandes Projetos Open Source

*falando de Kubernetes*

Ellen Körbes  
@ellenkorbes



KEYRUS

insight into value

# GERAÇÃO DE VALOR

## ANÁLISE DE DADOS E DIGITAL

A Keyrus auxilia as empresas a extrair todo o potencial de **Dados** e **Digital** objetivando aumentar seu desempenho, ajudando na transformação, e gerando novas alavancas de crescimento e competitividade



# PLAYER ESTRATÉGICO

## ALIANÇA CONSULTORIA E TECNOLOGIA



MANAGEMENT & TRANSFORMATION

DATA INTELLIGENCE

DIGITAL EXPERIENCE

# 3 UNIVERSOS DE EXPERTISE



Time de consultores especialistas dedicado ao desempenho das organizações

Uma oferta de consultoria inovativa orientada a gestão colaborativa

Knowhow e experiência para implementar governança de forma ágil



KEYRUS  
management



Extensa experiência em análise de dados

Experiencia e knowhow Consultivo -  
Tecnológico de Business Intelligence à Big Data

Pioneira com o curso de cientista de dados na  
Ecole polytechnique (França)



KEYRUS  
data



Expert em estratégia digital

User Experience: uma oferta global centrada na criação e gestão de interfaces digitais

Líder em consultoria e integração de Digital Commerce



K YRUS  
digita

KEYRUS  
data

## LEARNING EXPERIENCE



Formação customizada e capacitação de conteúdos inovadores, de forma, para times e/ou públicos específicos, com viés prático, baseado em experiência real de projetos (realidade do mercado), benchmarking e melhores práticas.



## QUEM?



# ELLEN KÖRBES

Atualmente:

Big Data Developer

12/2017 a 03/2018:

Developer full-time no projeto Kubernetes



## VAMOS FALAR SOBRE:

- O que faz o onboarding tão complicado.
- Soluções gerais para esses problemas.
- Como o projeto Kubernetes vem implementando essas soluções.
- A minha experiência trabalhando no projeto.



# KUBE-OQUÊ?

Wikipedia:

"Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications [...] It works with a range of container tools, including Docker."

cerca de  
**2 mil**  
contribuidores



+ de  
**2 milhões** de  
linhas de código

O problema: onboarding é difícil.

## O PROBLEMA DO ONBOARDING...

Como é que uma pessoa sozinha faz pra entender como todo o projeto funciona e começar a fazer contribuições?



## O PROBLEMA DO ONBOARDING...

O problema do Kubernetes é que... **tem muito Kubernetes.**

E aí o onboarding toma **tempo**.

Tempo é um problema, por questão de **comprometimento**:

"É um projeto aberto. E se eu investir nessa pessoa e amanhã  
ela sumir?"



## A SOLUÇÃO ATUAL: FUNCIONÁRIOS DE GRANDES EMPRESAS.



+



Empresas interessadas alocam funcionários ao projeto.

Exemplo: Minha equipe no Kubernetes era toda do Google.

Isso funciona pois o retorno sobre o tempo investido é (praticamente) garantido.



## A SOLUÇÃO ATUAL: FUNCIONÁRIOS DE GRANDES EMPRESAS.

Mas há problemas:

Não incentiva um processo de onboarding: o contribuidor corporativo não tem pressa.

Outros contribuidores não vão investir meses de tempo livre até conseguir contribuir.

## OUTRAS SOLUÇÕES: OUTREACHY AND GSOC.

### **OUTREACHY**

Estágios pagos para pessoas de grupos tradicionalmente sub-representados na tecnologia.

### **GOOGLE SUMMER OF CODE**

Estágios pagos para estudantes.

O problema do comprometimento? Se não trabalhar, não ganha. Resolvido.



Tá Ellen, mas eu não trabalho pro Google e esses estágios não servem pra mim. E agora?



desafios com onboarding em grandes projetos open source

# TAMO TUDO LASCADO ENTÃO?

Sim.



desafios com onboarding em grandes projetos open source

## TAMO TUDO LASCADO ENTÃO?

Sim.

A menos que se crie um processo onde contribuidores não precisem de ajuda para o onboarding inicial. Aí eles podem mostrar comprometimento primeiro, e receber mais recursos do projeto depois.



desafios com onboarding em grandes projetos open source

## TAMO TUDO LASCADO ENTÃO?

"Mas Ellen, não é assim que todo projeto open source funciona?"

É. Mas em projetos do tamanho do Kubernetes, não é tão simples assim.



desafios com onboarding em grandes projetos open source

## MAS SÉRIO, É TÃO COMPLICADO MESMO?

Primeiro passo. No Kubernetes, pra contribuir você precisa:

- Assinar o CLA.
- Achar um SIG pra trabalhar.
- Fazer o set up do seu ambiente de desenvolvimento.

Não sabe o que é um **CLA** ou um **SIG**?

Tá em um daqueles **70 docs** no repositório *community*. Procura lá.



desafios com onboarding em grandes projetos open source

MAS SÉRIO, É TÃO COMPLICADO MESMO?

Boa sorte.



## MAS SÉRIO, É TÃO COMPLICADO MESMO?

Sean Sullivan, contribuidor do SIG-CLI, funcionário do Google:

"Most projects have just one system. Say a team of 20 people, one system, complexity isn't too bad. But Kubernetes is a distributed system, and the amount of things you need to know..."

"The pool of knowledge is far deeper than at any of the other projects I have worked until now. You can spend years becoming an expert on this area, or that area... but then there are tons of other areas, and it's all changing rapidly. It feels overwhelming."



## MAS SÉRIO, É TÃO COMPLICADO MESMO?

Sean Sullivan, contribuidor do SIG-CLI, funcionário do Google:

"There are structures to organize this complexity, [...] It's only now that they're creating the kind of infrastructure to organize all of these separate groups across all of Kubernetes. There are, what, almost 40 SIGs now? And they all have different coding styles, different codebases."



# Minha experiência, parte 1.



## PRIMEIROS DIAS → TERROR.

- Kubernetes foi meu primeiro projeto de verdade.
- "Como fazer isso? Dá uma olhada aqui." Eram 200 mil linhas de código.
- Dias e dias e dias perdida. Natal e ano novo foram estresse total.
- "Será que isso é pra mim? Será que meu emprego anterior me aceita de volta?"



## TÁ. QUE QUE A GENTE FAZ ENTÃO?

- Trabalha.
- A comunidade do projeto tem que tornar esse onboarding mais fácil, mais claro, mais acessível.
- O objetivo é que novos contribuidores possam pegar o jeito e começar a contribuir sem precisar de muita ajuda.



Possíveis soluções: as fáceis.



## FORMAS **MOLEZINHA** DE TORNAR UM PROJETO ACCESSÍVEL.

- Arquivo CONTRIBUTING.md.
- Encourage a comunicação. Quem pode ajudar? IRC, Slack, e-mail?
- Tenha um código de conduta. Não é só pra enfeite.
- Fique de olho na atitude da sua comunidade
- Mantenha seu issues tracker organizado e use labels como “beginner friendly,” “help wanted,” e “good first PR.”



## Possíveis soluções: as difíceis.



desafios com onboarding em grandes projetos open source

## AS SOLUÇÕES FÁCEIS NÃO SÃO O SUFICIENTE...

...não em projetos dessa escala.

E as soluções difíceis não são difíceis por serem complicadas. Elas são simples.

O difícil é por em prática.



## TENHA BOA DOCUMENTAÇÃO.

- Sua documentação deve ter um ponto inicial. *Um ponto inicial.*
- E também um guia de quick start.
- E um roadmap.
- E... um monte de coisa.

Mas o mais importante: Mantenha um balanço entre dar informação demais, e dar informação de menos. Ambos são ruins.



## TENHA BOA DOCUMENTAÇÃO. MAS COMO?

- É necessário um **grupo ativamente trabalhando** nisso.
- Mantendo os **docs relevantes**, atualizados, e num padrão de qualidade.
- Use princípios de **story-telling**: Primeiro o contribuidor lê isso. Depois aquilo. Depois aquilo lá. Deve haver uma ordem lógica, uma progressão.
- Tacar um monte de arquivos numa pasta não é documentação!



## TENHA BONS COMENTÁRIOS NO CÓDIGO.

- Em Go, temos o godoc. Ele gera documentação a partir dos comentários no código.
- E sabe uma coisa legal que estão fazendo no Kubernetes?
- Numa PR, antes de ver o código, a gente abre o godoc. Se a documentação não fizer sentido por si só, a PR volta pra review. Sem nem olhar pro código.

Bom código não é auto-explicativo. Boas explicações são.



desafios com onboarding em grandes projetos open source

## ENSINE **AS MANHAS** PROS NOVATOS.

Meu sonho no Kubernetes:

Que algum developer experiente faça um screen capture do seu dia,  
narrando o que está fazendo.



## CRIE UM PROGRAMA DE MENTORING.

Isso é complicado pois:

- Tem o problema do comprometimento.
- Cada um tem uma idéia diferente sobre como mentoring deve ser.
- Ser um bom mentor é uma habilidade com a qual poucos nascem sabendo.

Idéias:

- Sessões de pair-programming.
- Code reviews.
- Conversas não-estruturadas, para tirar dúvidas.



desafios com onboarding em grandes projetos open source

## CRIE UM PROGRAMA DE MENTORING.

Mentoring é lindo <3

Mas é uma coisa individual. E aí temos um problema de escala.

Por isso que o ideal é...



desafios com onboarding em grandes projetos open source

## CRIE UM PROGRAMA DE MENTORING COLETIVO/EM GRUPO.

O que é? É mentoring individual, com audiência.

Idéias:

- Sessões de pair-programming... num hangout aberto.
- Code reviews... no YouTube.
- Conversas não-estruturadas... onde um monte de gente participa.



## CRIE UM PROGRAMA DE MENTORING COLETIVO/EM GRUPO.

### Vantagens:

- A questão do comprometimento individual desaparece.
- Há mais feedback para todas as partes envolvidas.
- O foco pode estar nas necessidades da comunidade, não de indivíduo isolado.



# Minha experiência, parte 2.



## O MOMENTO DA VIRADA.

- Aprendi que é difícil pra todo mundo. O problema não era eu.
- O momento "Me ajuda, caramba!"
- "Mas me ajuda assim, assim, e assado."
- "Precisamos de meetings. One-on-ones. Code reviews."



# Consertando: comunidade.



desafios com onboarding em grandes projetos open source

## DOCUMENTAÇÃO.

O contributor's guide do Kubernetes costumava ser... aqueles 70 docs jogados numa pasta.

No começo isso era bom, mas aí o projeto cresceu.

Nos últimos meses uma galera foi atrás de consertar essa situação e criar um guia claro e conciso.

Tá lindo.



desafios com onboarding em grandes projetos open source

## MENTORING.

Há um programa de mentoring coletivo dando seus primeiros passos.

São grupos de umas 10 pessoas estudando certos assuntos. A idéia é que estes grupos subam um degrau na hierarquia do Kubernetes.



## MEET OUR CONTRIBUTORS.

Existe um esquema chamado *Office Hours*. E agora também tem *Meet Our Contributors*. A mesma coisa, mas para contribuidores.

*"Meet Our Contributors gives you a monthly one-hour opportunity to ask questions about our upstream community, watch interviews with our contributors, and participate in peer code reviews."*

A primeira edição foi dia 7 de fevereiro. Eu participei!



# Consertando: código.



## A SITUAÇÃO ATUAL.

Sean:

*“For example, there are so many different clients that can talk to the API server. [...]”*

*“There’s no reason for how things are done [...] The interaction between components of each of them is a mystery.”*

*“Projects grow and grow and you just tack stuff onto it ‘til it’s a huge difficult mess.”*



## A SITUAÇÃO ATUAL.

Sean:

*"There are very few people who understands the whole scope of the client to the API server..."*

*"Even people who work on this as their day job only know parts of it; the complexity is substantial."*

*"Right now there's this massive monolith, and the path Phil is trying to take us towards is breaking this monolith into smaller tools."*



## A SITUAÇÃO ATUAL.

Phillip Wittrock, lead do SIG-CLI, funcionário do Google:

*"The biggest challenge for developers is incidental complexity.*

*"Ideally you'd develop self contained bits where you only need to understand how to use the library, not how it works.*

*"So if you have a system that's three libraries, [...] a developer can approach the system learning only about the program they're modifying, which is 1/4 of the system."*



## A SITUAÇÃO ATUAL.

Phillip Wittrock, lead do SIG-CLI, funcionário do Google:

*"If instead the program implements all the stuff in the libraries, the developer needs to understand the interactions of the full system [...] you get an explosion of complexity.*



## E O FUTURO?

Perguntei sobre que características tornam um projeto acessível para novos contribuidores:

*"I don't think they're approached quite like we need to make it easy for new developers to join this project, but more like, the code needs to be clean and well structured. Which is really 80% of making it easy for new developers to join."*



## O FUTURO IDEAL.

Phil:

*"Pieces consumed by groups other than the authoring group must be simple, well documented and developed independently from the components that consume them."*



## O FUTURO IDEAL.

Phil:

*"As Kubernetes contributor, I should be able to understand how to use the public libraries/tools published by groups I am not a member of without learning how they are implemented or reading their source code.*

*"As a Kubernetes contributor, I should be able to understand how to use the public libraries / tools published by groups I am not a member without soliciting the attention of a member of those groups."*



Você.



# QUER CONTRIBUIR, MAS TEM DIFICULDADES?

- Peça ajuda. Claramente.
- Tenha em mente que ajudar é difícil.
- E saiba que contribuir para um projeto desses é difícil.



desafios com onboarding em grandes projetos open source

# TEM EXPERIÊNCIA, MAS NÃO SABE COMO AJUDAR?

- Dá uma olhada nas disciplinas que de developer advocacy, developer evangelism, ou developer relations.
- Não precisa dar palestras. Foque na arte de conectar ferramentas com developers.



# Minha experiência, parte 3.



desafios com onboarding em grandes projetos open source

## NO FIM DESSA HISTÓRIA TODA...

- Recebi ajuda.
- Acabei pegando o jeito.
- Fiz contribuições, e agora tem um library lá com meu nome embaixo.



desafios com onboarding em grandes projetos open source

## QUAL A SENSAÇÃO AO FIM DO INTERNSHIP?

Desafio nenhum é grande demais.

Kubernetes for life.



Por fim...



## LINKS

A série de artigos que deu origem a esta talk:

<https://medium.com/deffectivego/onboarding-issues-in-large-scale-open-source-projects-lets-talk-kubernetes-part-1-a1a64c9c9cfe>

Talk 'Montanhas de Altitude Variável', sobre mentoring:

<https://www.youtube.com/watch?v=DgwFkclhnIM>

Meet Our Contributors #1:

<https://www.youtube.com/watch?v=iqtZK02j-LM>



## LINKS

Artigo 'What is Developer Advocacy':

<https://medium.com/@ashleymcnamara/what-is-developer-advocacy-3a92442b627c>

O novo Contributor's Guide do Kubernetes

<https://github.com/kubernetes/community/tree/master/contributors/guide>



## AGRADECIMENTOS:

♥ Paris Pittman

♥ Phillip Wittrock

♥ Sean Sullivan

♥ Nikhita Raghunath

♥ Antoine Pelisse



# ARE YOU AWESOME?

WE'RE HIRING!

Se inscreva em nosso site: <http://www.keyrus.com.br>

**BIG DATA**

**QLICK**

**DATA SCIENCE**

**POWER BI**

**UX DESIGNER**

**JAVA, ETL, POWER CENTER**



**BRASIL**



**recrutamento@Keyrus.com.br**



**Sabrinassp**



**+55 11 9.8287-1096**

**KEYRUS**  
insight into value

KEYRUS



insight into value

Ellen Körbes  
[@ellenkorbes](https://twitter.com/ellenkorbes)