



Python^{Live de}

#106 - CI - Integração contínua

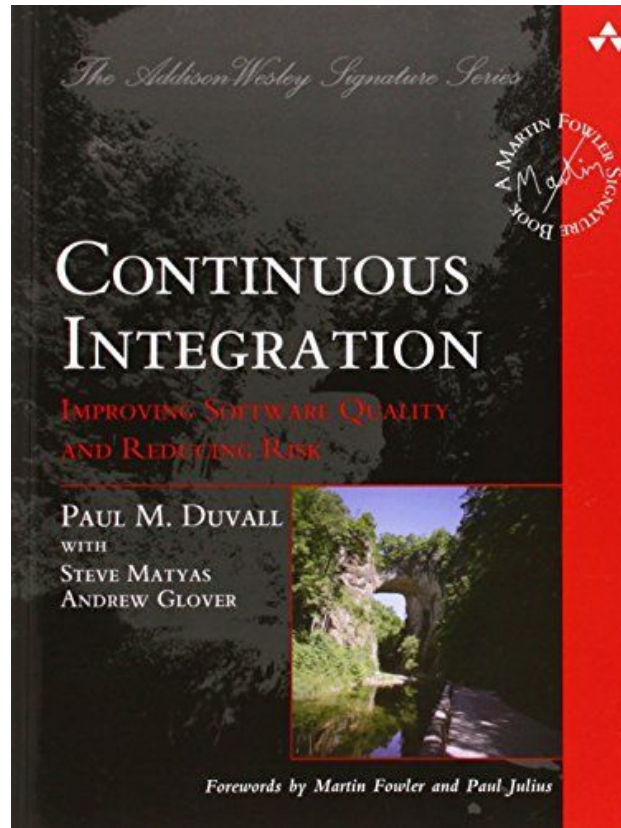
Obrigado!

apoia.se/livedepython

```
dunossauro at bouman in ~/git/apoiase on master*
```

```
$ python apoiadores.py
```

Alexandre Possebom	Alysson Oliveira	Amaziles Carvalho	Andre Machado
Andre Rodrigues	Bernardo Fontes	Bruno Guizi	Carlos Augusto
Cleber Santos	Cleiton Mittmann	David Reis	Dayham Soares
Diego Ubirajara	Edimar Fardim	Eliabe Silva	Eliakim Moraes
Elias Soares	Emerson Lara	Eugenio Mazzini	Fabiano Silos
Fabiano Teichmann	Fabiano Gomes	Fernando Furtado	Franklin Silva
Fábio Serrão	Gleison Oliveira	Guilherme Ramos	Hemilio Lauro
Humberto Rocha	Hélio Neto	JOAO COELHO	JONATHAN DOMINGOS
Jean Vetorello	Johnny Tardin	Jonatas Oliveira	Jonatas Simões
João Lugão	Jucélio Silva	Júlia Kastrup	Leon Teixeira
Lucas Nascimento	Magno Malkut	Marcus Salgues	Maria Boladona
Matheus Francisco	Nilo Pereira	Pablo Henrique	Paulo Tadei
Pedro Alves	Rafael Galleani	Regis Santos	Renan Moura
Renato Santos	Rennan Almeida	Rodrigo Ferreira	Rodrigo Vaccari
Sérgio Passos	Thiago Araujo	Tiago Cordeiro	Tyrone Damasceno
Vergil Valverde	Vicente Marcal	Wander Silva	Wellington Carlos
Wellington Camargo	Welton Souza	William Oliveira	Willian Gl
Yros Aguiar	Falta você	Falta você	Falta você



Livro base de hoje

Kathy Sierra

"Há uma grande diferença entre dizer 'Coma uma maçã por dia' e realmente comer a maçã"

Integração contínua é uma técnica “das quais é mais fácil de pregar do que praticar. Porque a frequência de adoção da técnica é miseravelmente baixa”.

Paul M. Duvall, Steve Matyas Andrew Glover - 2007



Antes de explicar, o que queremos fazer?

- Builds de integração
- Reconstrução de bancos de dados
- Executar testes automatizados
- Deploys / implantação
- Feedbacks constantes

BUILDS??

Builds

Build costuma ser uma palavra genérica, não tem relação explícita com *buildar a aplicação*, como usamos em linguagens compiladas por exemplo.

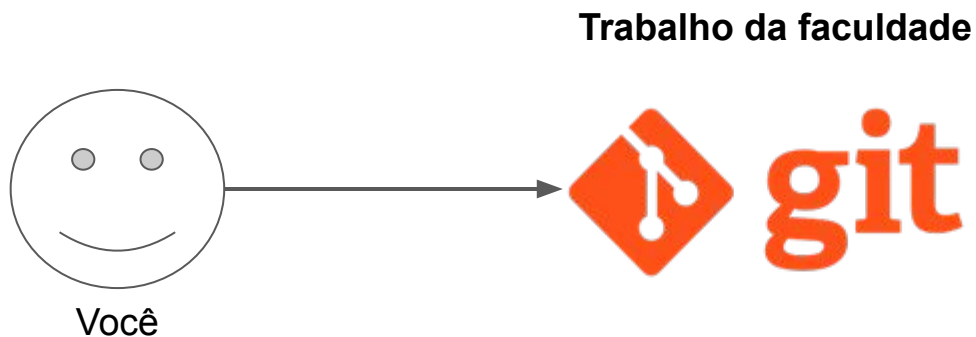
Quer dizer que vai acontecer um “procedimento” que vai ver se as coisas estão saudáveis no código que está no seu gerenciador de versão.

Builds

Builds podem ser compostos por muitos passos. Como:

- Compilar o código
- Rodar os testes
- Fazer inspeção no código
- Fazer um deploy para produção

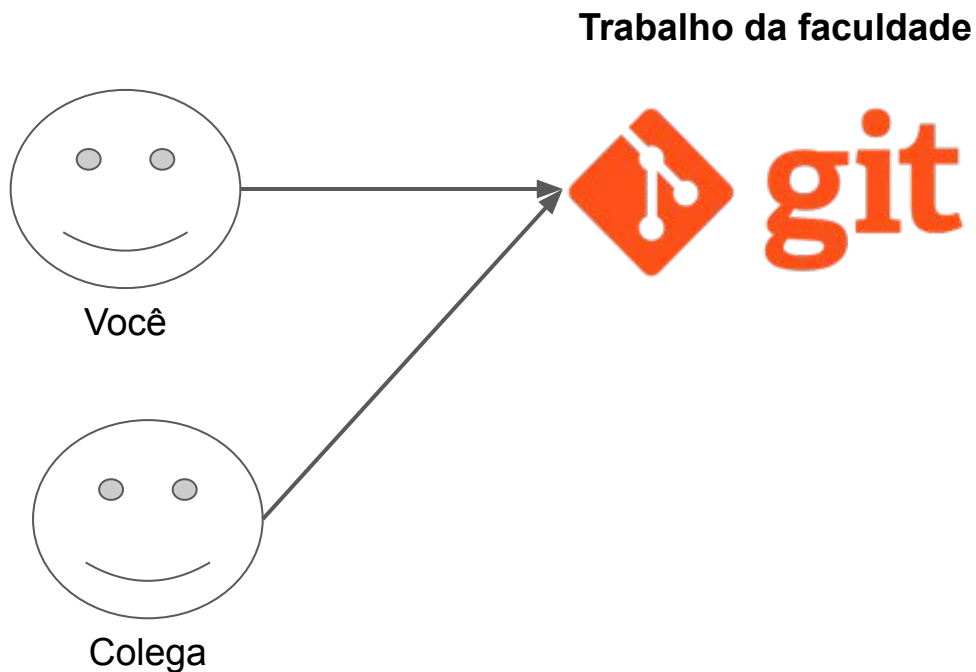
Vamos imaginar um caso



```
● def soma(x, y):  
    return x + y
```

```
● def subtração(x, y):  
    return x - y
```

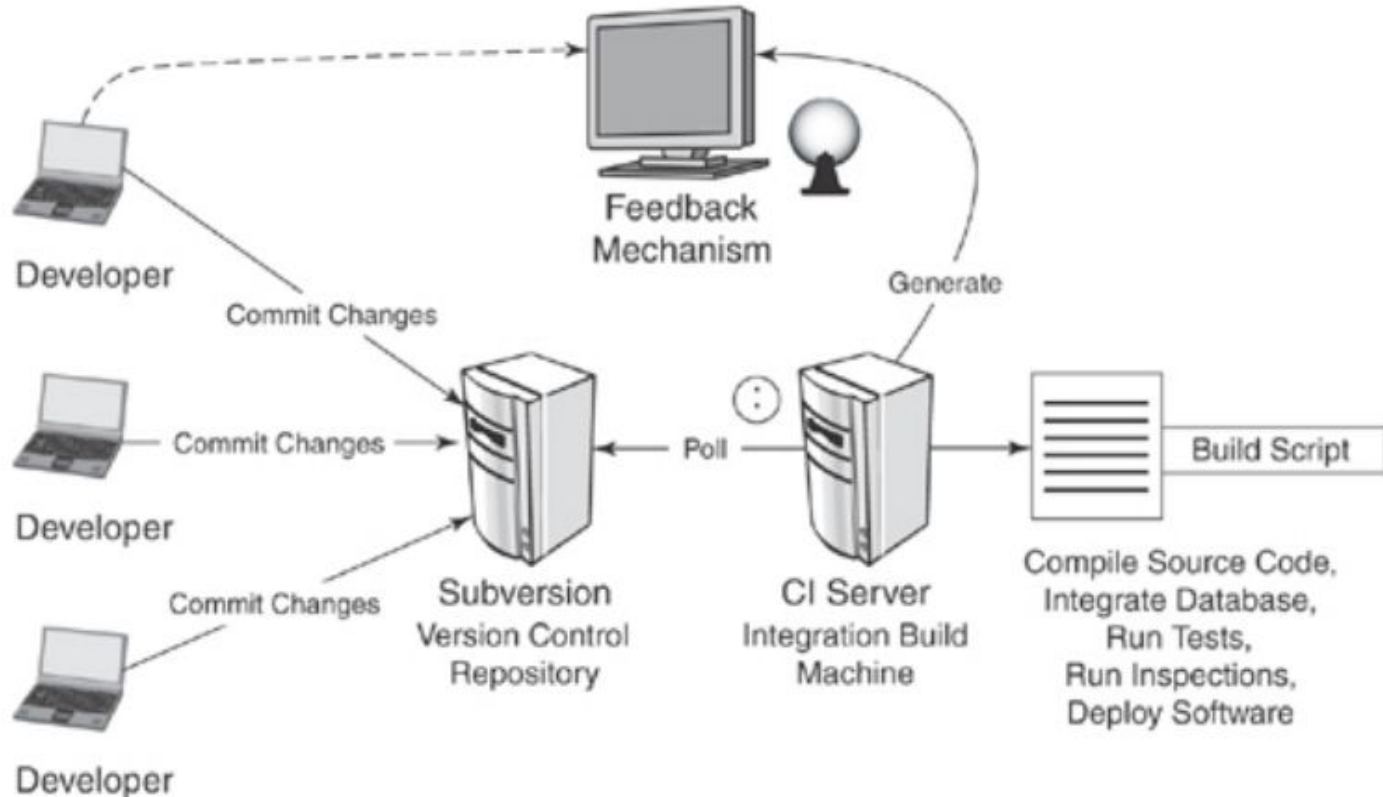
Vamos imaginar um caso



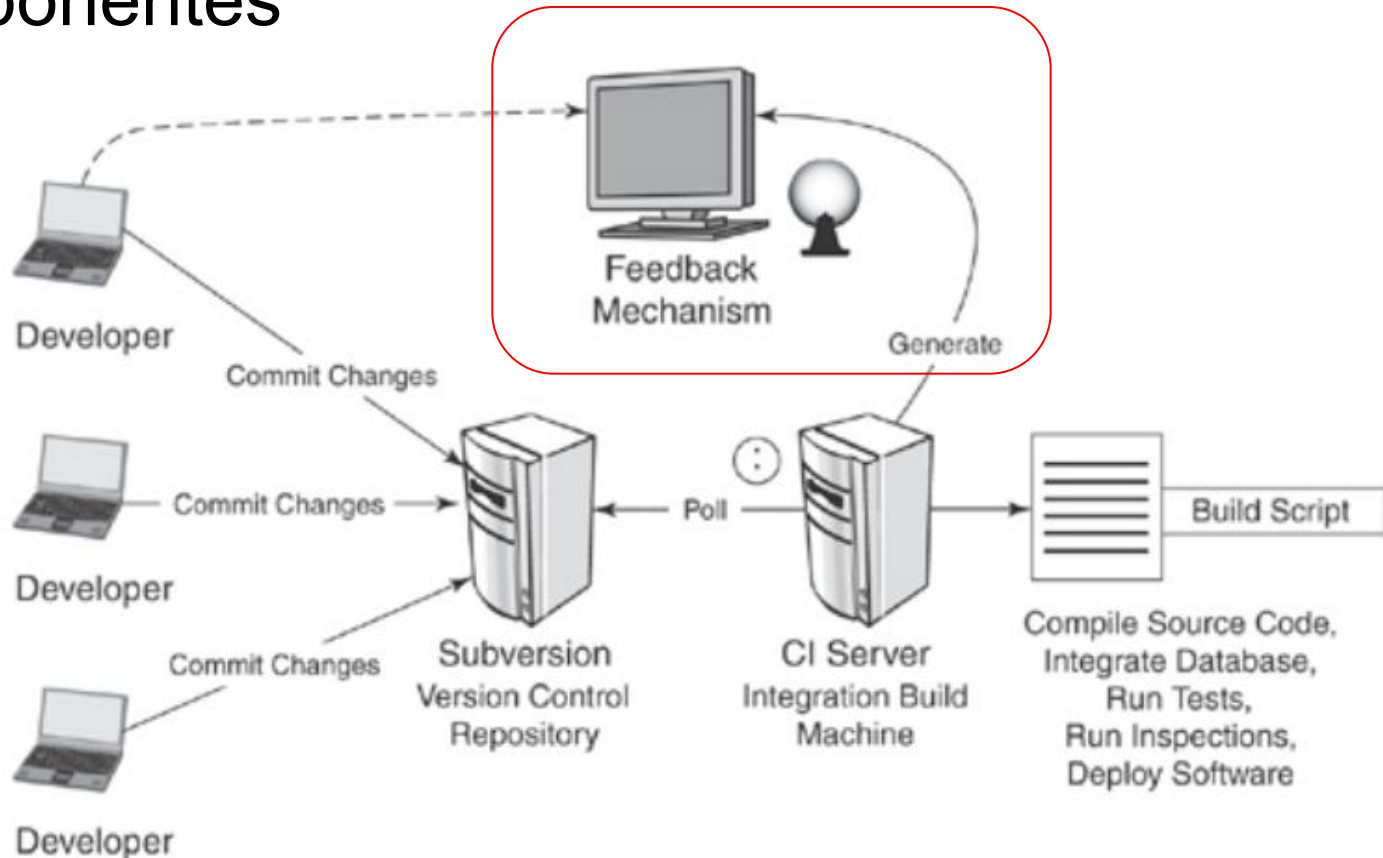
```
def soma(x, y):  
    return x + y
```

~~def soma(x, y):
 return x + y~~

Componentes



Componentes



Critical Alerts

2

via OpsDash
Last updated at 12:11

Warning Alerts

3

via OpsDash
Last updated at 12:11

Servers Down

1

via OpsDash
Last updated at 12:11

Buzzwords

Paradigm shift	17
Turn-key	9
Pivoting	14
Leverage	12
Streamlininess	16
Web 2.0	16
Exit strategy	21
Enterprise	13
Synergy	19

of times said around the office
Last updated at 12:11

Hello

This is your shiny new dashboard.

Protip: You can drag the widgets around!

Synergy



Last updated at 12:11

Reduzindo premissas

Reduzindo premissas

Todos os envolvidos no projeto

- Rodam testes
- Escrevem código pouco complexo
- Escrevem novos testes
- Seguem os guias de estilo da linguagem
- Escrevem documentação

Reduzindo premissas

Você pode ver o CI como algo unicamente capaz de integrar código.

Mas ele é o **componente central da qualidade do que é feito.**

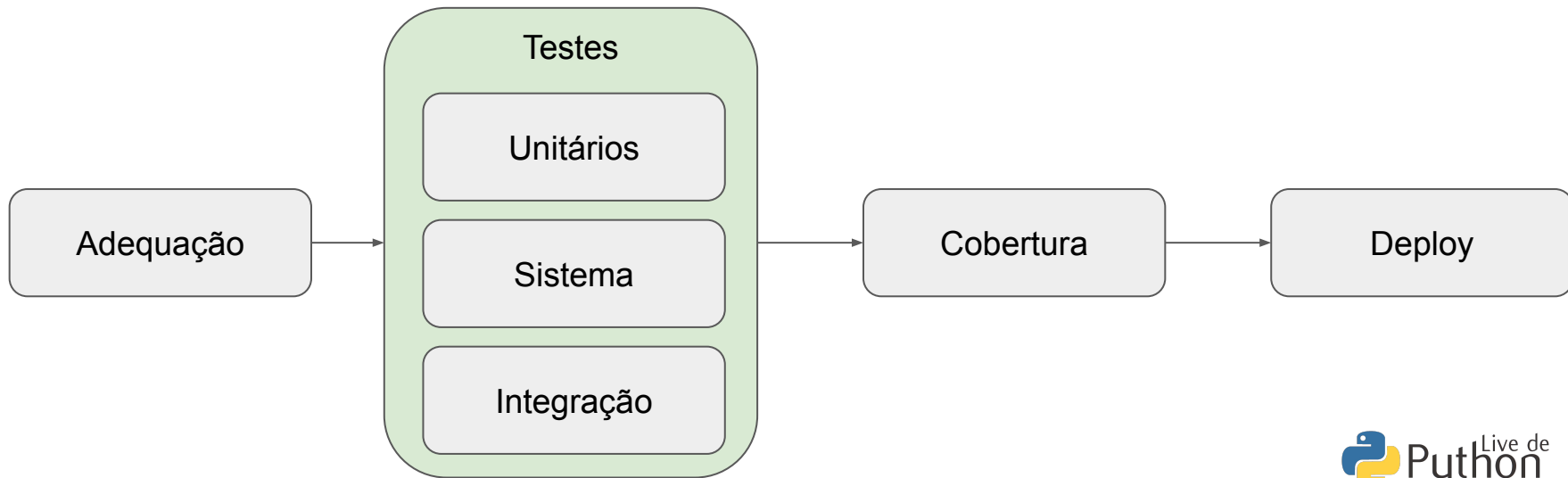
Qual o valor gerado pelo CI?

- Redução de riscos
 - Garante que tudo é integrável
 - Que passa nos testes
 - Que funciona
- Reduz processos manuais chatos
 - Builds, deploys, rodar testes
- Garante que todo código é ENTREGÁVEL
- Gera uma confiança maior entre o time e o produto

O script de build

Um bom script de integração conta com vários passos (steps) ou estágios (stages). Em um vocabulário mais moderno, podemos chamar isso de **Pipeline**.

Onde a cada commit no repositório, vamos ter uma série de coisas acontecendo.



Ferramentas



Código

Repositório



bitbucket

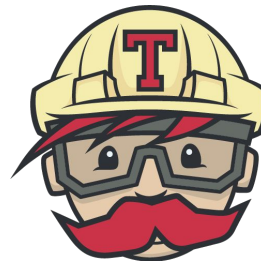


gitlab



github

Integração Contínua



Travis-CI



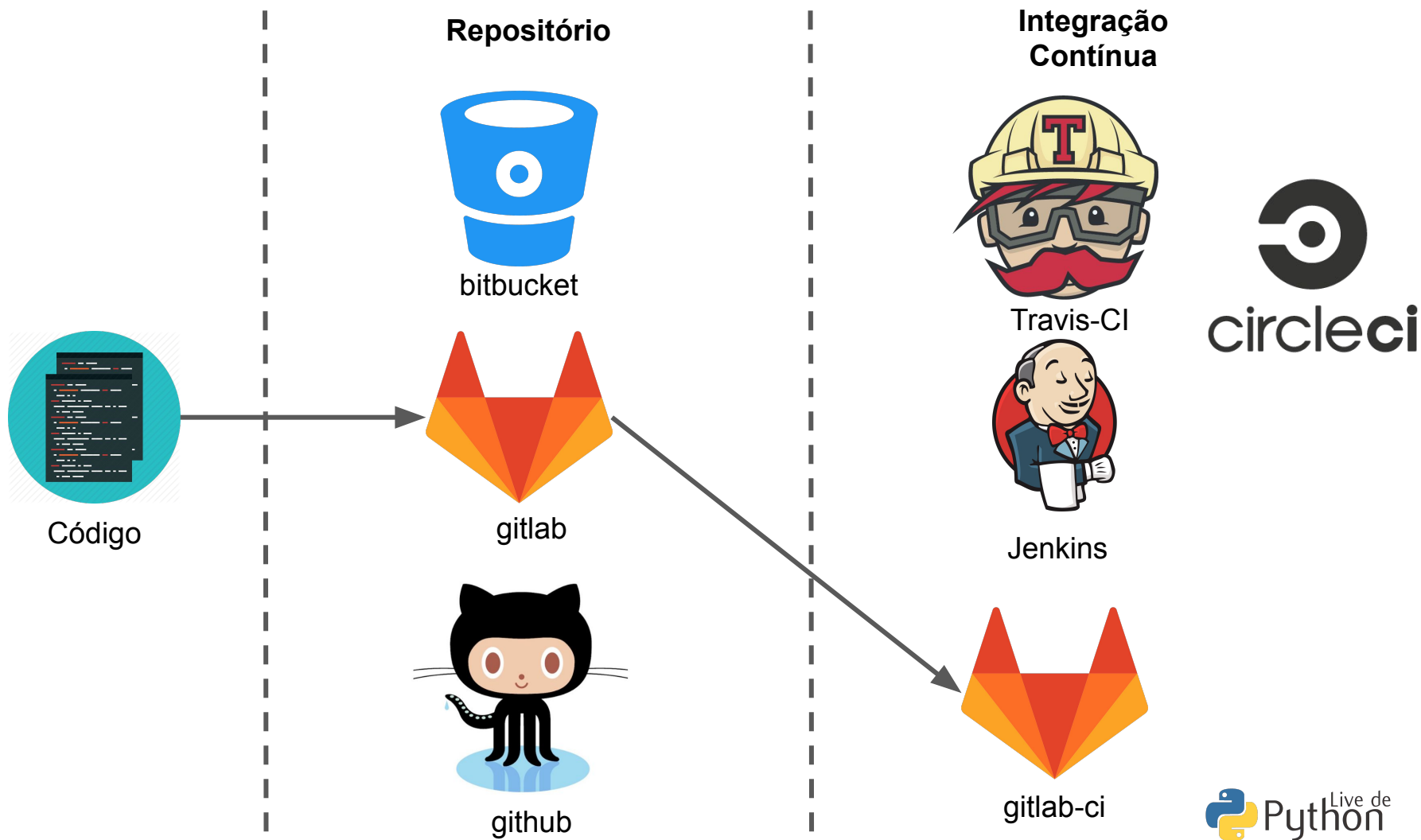
Jenkins



gitlab-ci



circleci



GITLAB-CI

