Testes e Integração Contínua

Bruno Mendes
Mairieli Wessel

POR QUE TESTAR?

Termos

O ser humano está sujeito a cometer um **erro** (engano)

ERRO

Que produz um **defeito** (bug) no código ou documento

DEFEITO

Se um defeito no código for executado, o sistema irá **falhar**

FALHA

O QUE É **TESTE DE SOFTWARE?**

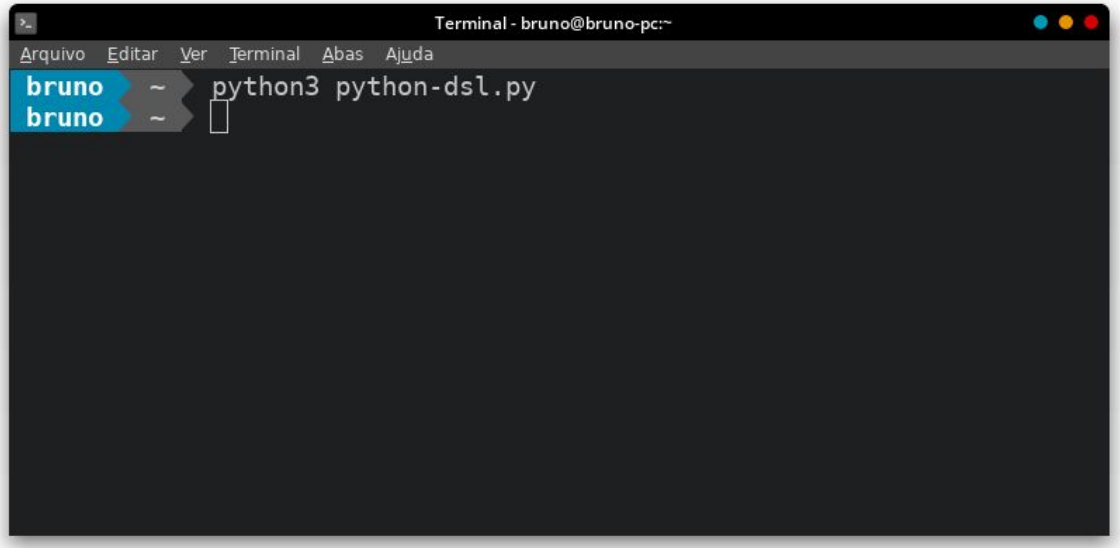
Níveis de Teste de Software

- Teste de **Unidade**
- Teste de **Integração**
- Teste de **Sistema**
- Teste de **Aceitação**

Teste de **Unidade**

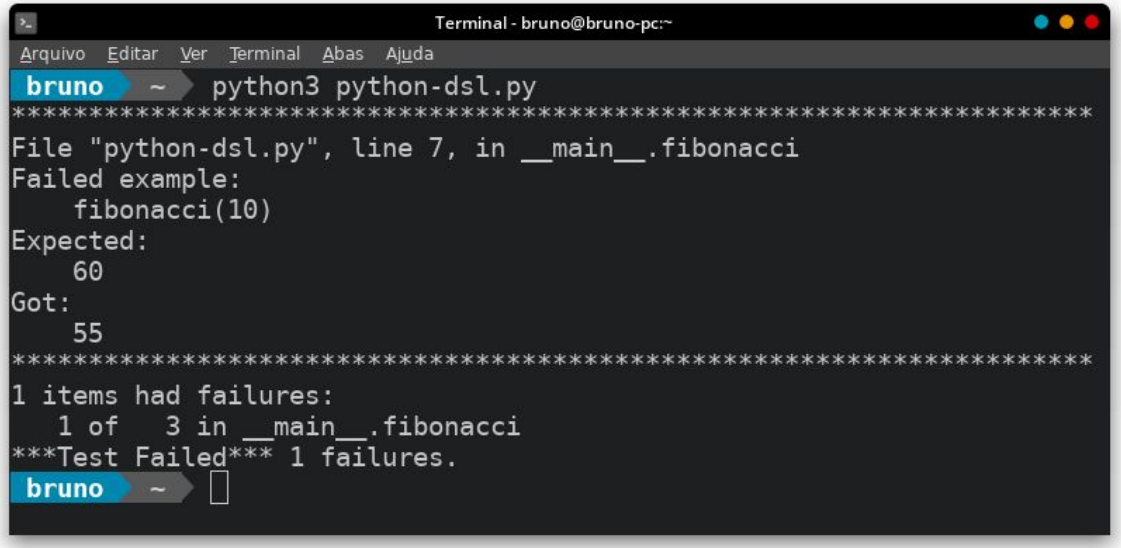
- Explorar a menor unidade do projeto
- Usa-se a descrição do projeto no nível da unidade como guia para os testes
- Os erros estarão nos limites destas unidades
- Pode ser conduzido em paralelo para diversas unidades

```python
def fibonacci(n):
    """
    >>> fibonacci(0)
    0
    >>> fibonacci(1)
    1
    >>> fibonacci(10)
    55
    """
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

import doctest
doctest.testmod()
```



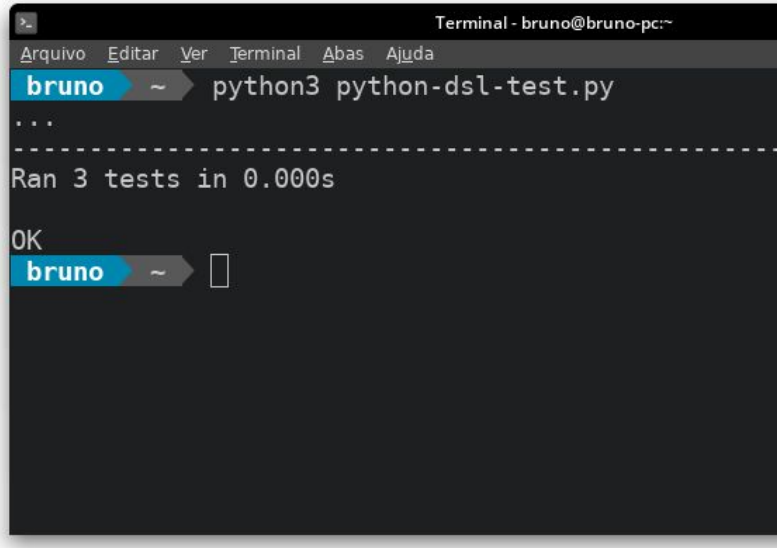
Doctest

```python
def fibonacci(n):
    """
    >>> fibonacci(0)
    0
    >>> fibonacci(1)
    1
    >>> fibonacci(10)
    60
    """
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

import doctest
doctest.testmod()
```



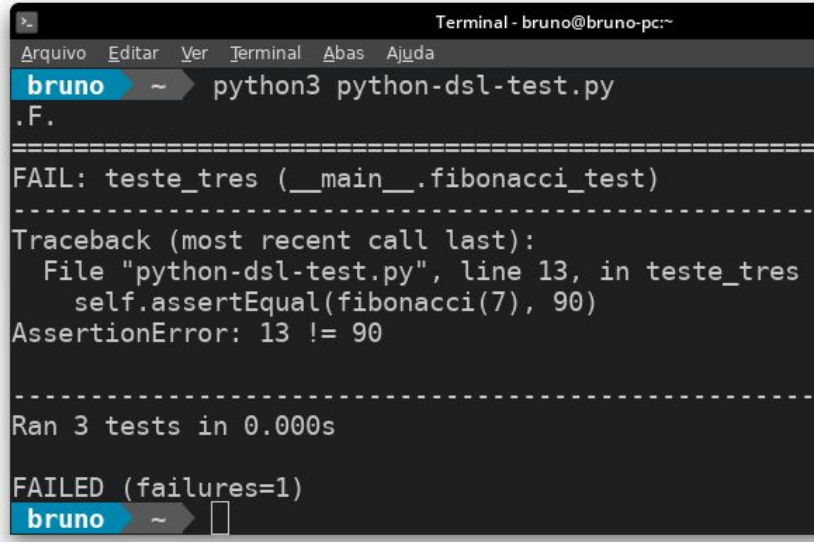
Doctest

```python
import unittest
from fibonacci import fibonacci

class fibonacci_test(unittest.TestCase):

    def teste_um(self):
        self.assertEqual(fibonacci(0), 0,

    def teste_dois(self):
        self.assertEqual(fibonacci(1), 1)

    def teste_tres(self):
        self.assertEqual(fibonacci(7), 13)

unittest.main()
```



Unittest

```python
import unittest
from fibonacci import fibonacci

class fibonacci_test(unittest.TestCase):

    def teste_um(self):
        self.assertEqual(fibonacci(0), 0)

    def teste_dois(self):
        self.assertEqual(fibonacci(1), 1)

    def teste_tres(self):
        self.assertEqual(fibonacci(7), 90)

unittest.main()
```



Unittest

Teste de **Integração**

- É a fase do teste de software em que módulos são combinados e testados em grupo.

Ferramentas de Teste

CODESHIP

circleci

Bamboo

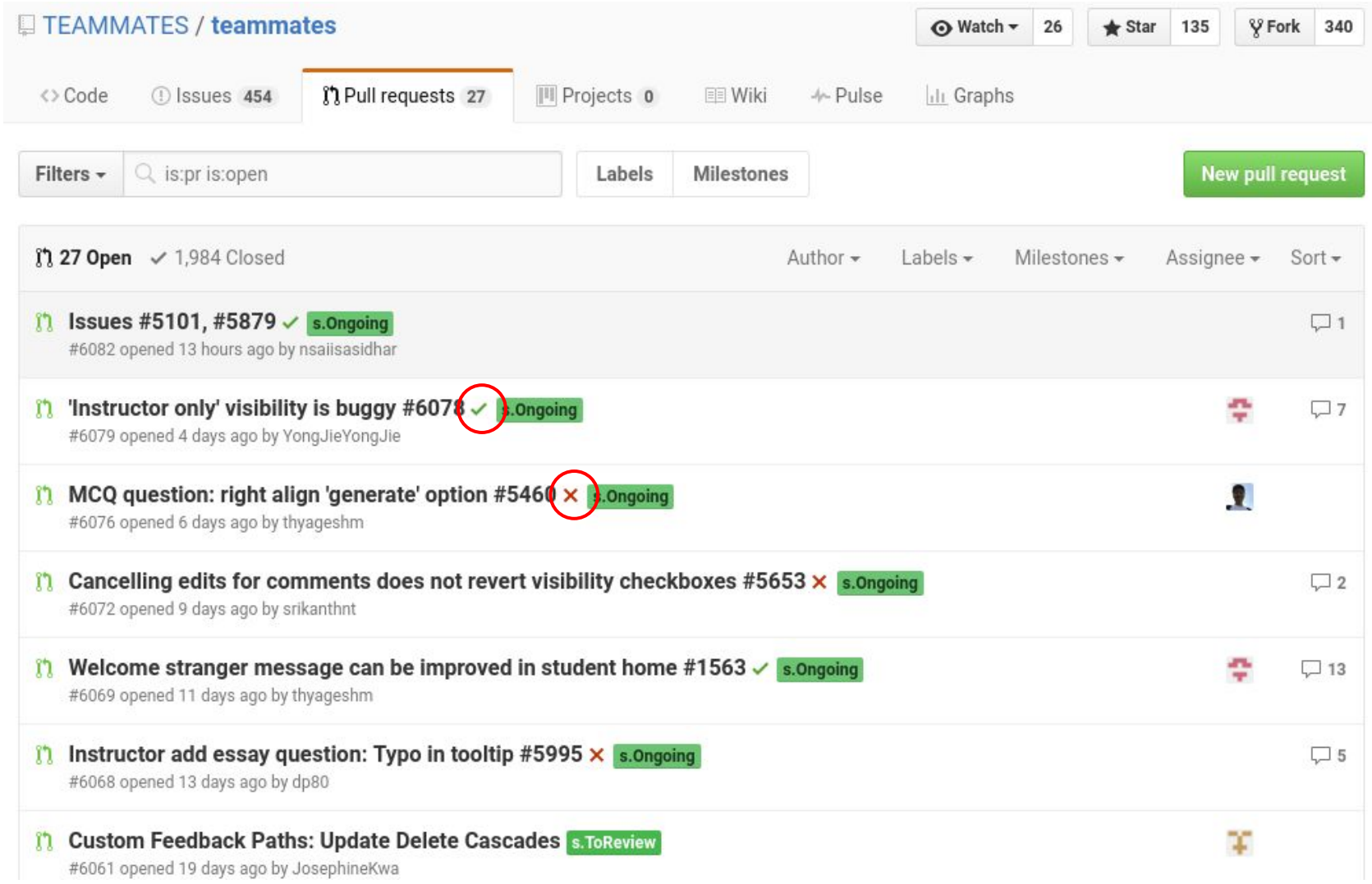
Solano Labs

snap

semaphore

AppVeyor

drone.io


TEAMMATES / teammates
Watch 26
Star 135
Fork 340
Code
Issues 454
Pull requests 27
Projects 0
Wiki
Pulse
Graphs
Filters
is:pr is:open
Labels
Milestones
New pull request
27 Open
1,984 Closed
Author
Labels
Milestones
Assignee
Sort
Issues #5101, #5879 s.Ongoing
#6082 opened 13 hours ago by nsaiisasidhar
1
'Instructor only' visibility is buggy #6078 s.Ongoing
#6079 opened 4 days ago by YongJieYongJie
7
MCQ question: right align 'generate' option #5460 s.Ongoing
#6076 opened 6 days ago by thyageshm
Cancelling edits for comments does not revert visibility checkboxes #5653 s.Ongoing
#6072 opened 9 days ago by srikanthnt
2
Welcome stranger message can be improved in student home #1563 s.Ongoing
#6069 opened 11 days ago by thyageshm
13
Instructor add essay question: Typo in tooltip #5995 s.Ongoing
#6068 opened 13 days ago by dp80
5
Custom Feedback Paths: Update Delete Cascades s.ToReview
#6061 opened 19 days ago by JosephineKwa

CODE COVERAGE

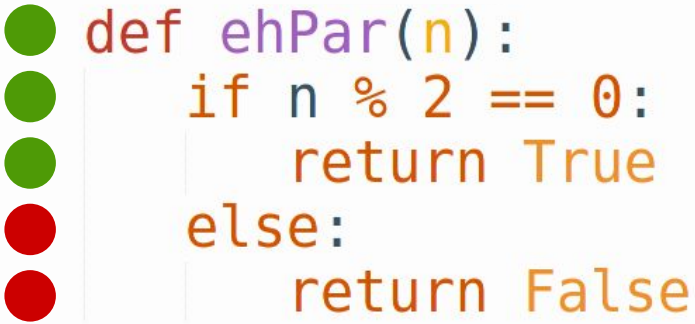
CODE COVERAGE

```
def ehPar(n):
    if n % 2 == 0:
        return True
    else:
        return False
```

CODE COVERAGE

```
import unittest
from parimpar import ehPar

class ehPar_test(unittest.TestCase):

    def teste(self):
        self.assertEqual(ehPar(10), True)

unittest.main()
```

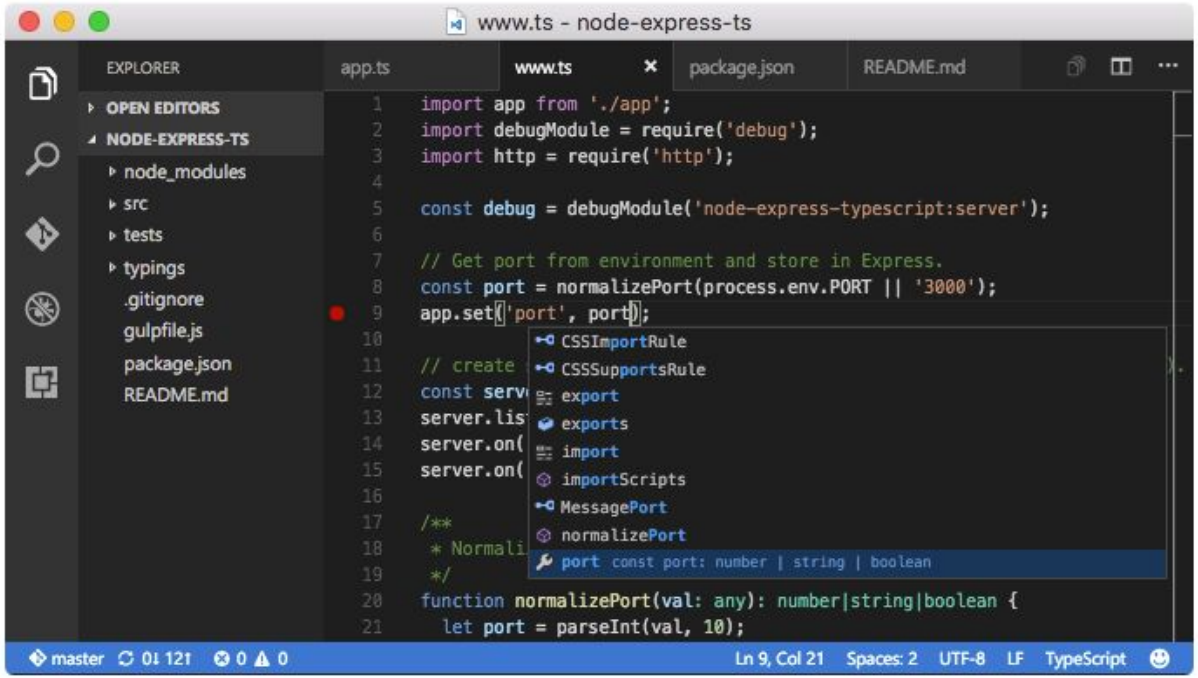
CODE COVERAGE



Visual Studio Code - Open Source

build passing | build pending | coverage 61%

VS Code is a new type of tool that combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. Code provides comprehensive editing and debugging support, an extensibility model, and lightweight integration with existing tools.



mairieli commented 22 hours ago

- Update the 'Example' section README to show this change #93

Fixes #94

Show refactored Example in the README 081c34f

codecov-io commented 22 hours ago

Current coverage is 100% (diff: 100%)

Merging #96 into master will not change coverage

```
@@          master   #96   diff @@
==================================
  Files           1     1
  Lines         103   103
  Methods         0     0
  Messages        0     0
  Branches        0     0
==================================
  Hits          103   103
  Misses          0     0
  Partials        0     0
```

Powered by Codecov. Last update 3f647af...081c34f

Pull Request Review is now available. Learn more. Add your review

All checks have passed
3 successful checks

Hide all checks

- codecov/patch — Coverage not affected when comparing 3f647af...081c3... Details
- codecov/project — 100% (+0.00%) compared to 3f647af Details
- continuous-integration/travis-ci/pr — The Travis CI build passed Details

This branch has conflicts that must be resolved
Only those with write access to this repository can merge pull requests.

ANÁLISE ESTÁTICA

Code Complexity No Patterns	Code Style 78%
Compatibility 100%	Documentation No Patterns
Error Prone 90%	Performance 100%
Security 100%	Unused Code 92%

Avoid variables with short names like i

```
49  Integer i = Integer.parseInt(parametro.substring(parametro.length() - 1));
```

GitLab Community Edition

Version Control on your Server. See https://gitlab.com/gitlab-org/gitlab-ce and the README for more information

★ Star 2322 HTTPS https://gitlab.com/gitlab-or

Files (1,000 MB) Commits (37,010) Branches (475) Tags (400) Changelog MIT License Contribution guide CI configuration

passed 3bf0d143 Merge branch 'fix-todo-mobile-view' into 'master' · about an hour ago by Annabel Dunstone Gray

GitLab

build success coverage 89% code climate 3.4 cii best practices passing 100%

Canonical source

The canonical source of GitLab Community Edition is hosted on GitLab.com.

Gihub Integrations

Code Climate

The open static analysis platform for automated code review

Codecov

Code coverage done right.

Codacy

Automated code reviews to help developers ship better software,

Hound

Hound comments on style violations in pull requests

Coveralls

Deliver code confidently with test coverage history and statistics

QuantifiedCode

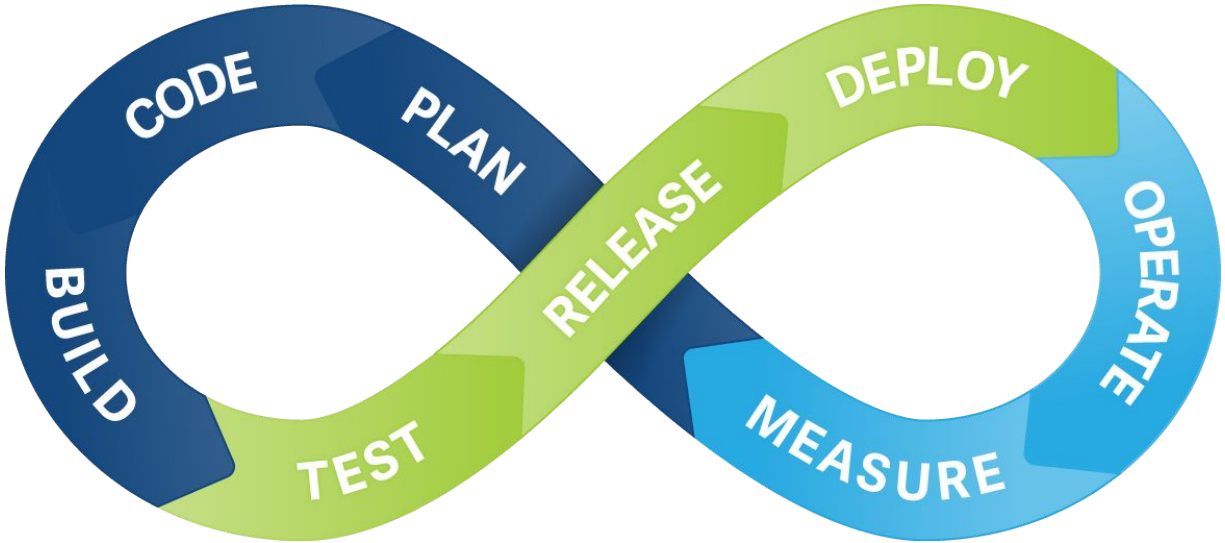
Automated code review and repair for Python

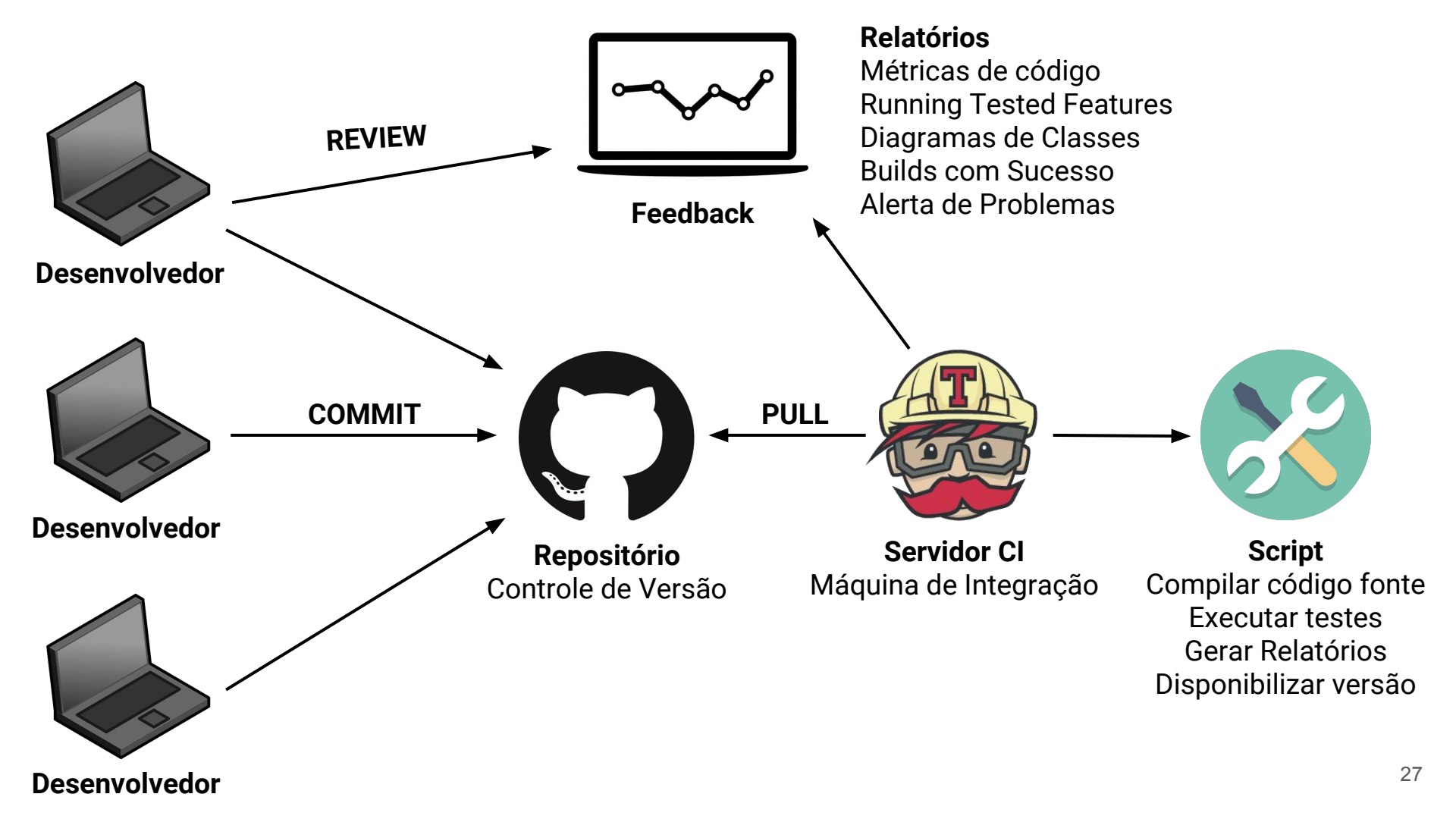
https://github.com/integrations

INTEGRAÇÃO CONTÍNUA

“Integração Contínua é uma prática de desenvolvimento de software onde os membros de um time integram seu trabalho frequentemente, geralmente cada pessoa integra pelo menos diariamente – podendo haver múltiplas integrações por dia. Cada integração é verificada por um build automatizado (incluindo testes) para detectar erros de integração o mais rápido possível.”

Martin Fowler




Relatórios
Métricas de código
Running Tested Features
Diagramas de Classes
Builds com Sucesso
Alerta de Problemas
REVIEW
Feedback
Desenvolvedor
COMMIT
PULL
Desenvolvedor
Repositório
Controle de Versão
Servidor CI
Máquina de Integração
Script
Compilar código fonte
Executar testes
Gerar Relatórios
Disponibilizar versão
Desenvolvedor

Por que Integração Contínua?

Integrar é difícil e o esforço aumenta exponencialmente com:

- Número de componentes
- Número de bugs
- Tempo desde a última integração

Práticas da Integração Contínua

- Manter um único repositório de código.
- Automatize a Build
- Faça uma Build ser auto-testável
- Cada um lança suas modificações todos os dias
- Cada commit deve atualizar o repositório principal em uma máquina de integração

Práticas da Integração Contínua

- Mantenha a Build rápida
- Torne fácil para qualquer um ter o último executável.
- Todos podem ver o que está acontecendo.
- Automatize a Implantação do Sistema

Práticas da Integração Contínua

- Monitorar automaticamente a qualidade do código
- Monitorar métricas de cobertura de código

Benefícios da Integração Contínua

- Redução de Riscos, fornecendo feedbacks rápidos.
- Em todo tempo você sabe onde você está, o que funciona, o que não funciona e os bugs pendentes que você tem em seu sistema.
- Integração Contínua não nos livra dos bugs, mas os tornam mais fáceis de encontrar e remover.

Benefícios da Integração Contínua

- As métricas de qualidade de código publicamente visíveis podem encorajar os desenvolvedores a melhorar seus códigos

Integração Contínua

Entrega Contínua

FERRAMENTAS DE INTEGRAÇÃO CONTÍNUA

CODESHIP

circleci

Bamboo

Solano Labs

snap

semaphore

AppVeyor

drone.io

Travis CI

O que é Travis CI?

- Serviço de hospedagem de integração contínua
- Pode ser integrado ao GitHub
- Gratuito para repositórios públicos e pago para repositórios privados

O que o Travis CI pode fazer?

- Monitorar projetos do GitHub
- Executar testes
- Prover feedback
- Build
- Verificar qualidade do software
- Deploy

Vantagens do Travis CI?

- Fácil para configurar e usar
- Integra com GitHub e cloud services
- Livre de manutenção
- Tempo para realizar a configuração

Desvantagens do Travis CI?

- Não é possível gerar build manual
- Menos extensível que o Jenkins

```
def ehPar(n):
    if n % 2 == 0:
        return True
    else:
        return False
```

```python
import unittest
from ehPar import ehPar

class ehPar_test(unittest.TestCase):

    def teste(self):
        self.assertEqual(ehPar(10), True)


unittest.main()
```

Travis CI Blog Status Help Bruno Mendes

Bruno Mendes
Repositories 9

Token:

Organizations

You are not currently a member of any organization.

Is an organization missing?
Review and add your authorized organizations.

Bruno Mendes

Sync account

We're only showing your public repositories. You can find your private projects on travis-ci.com.

Flick the repository switch on

Add .travis.yml file to your repository

Trigger your first build with a git push

brnomendes/algoritmo-genetico

brnomendes/brnomendes.github.io

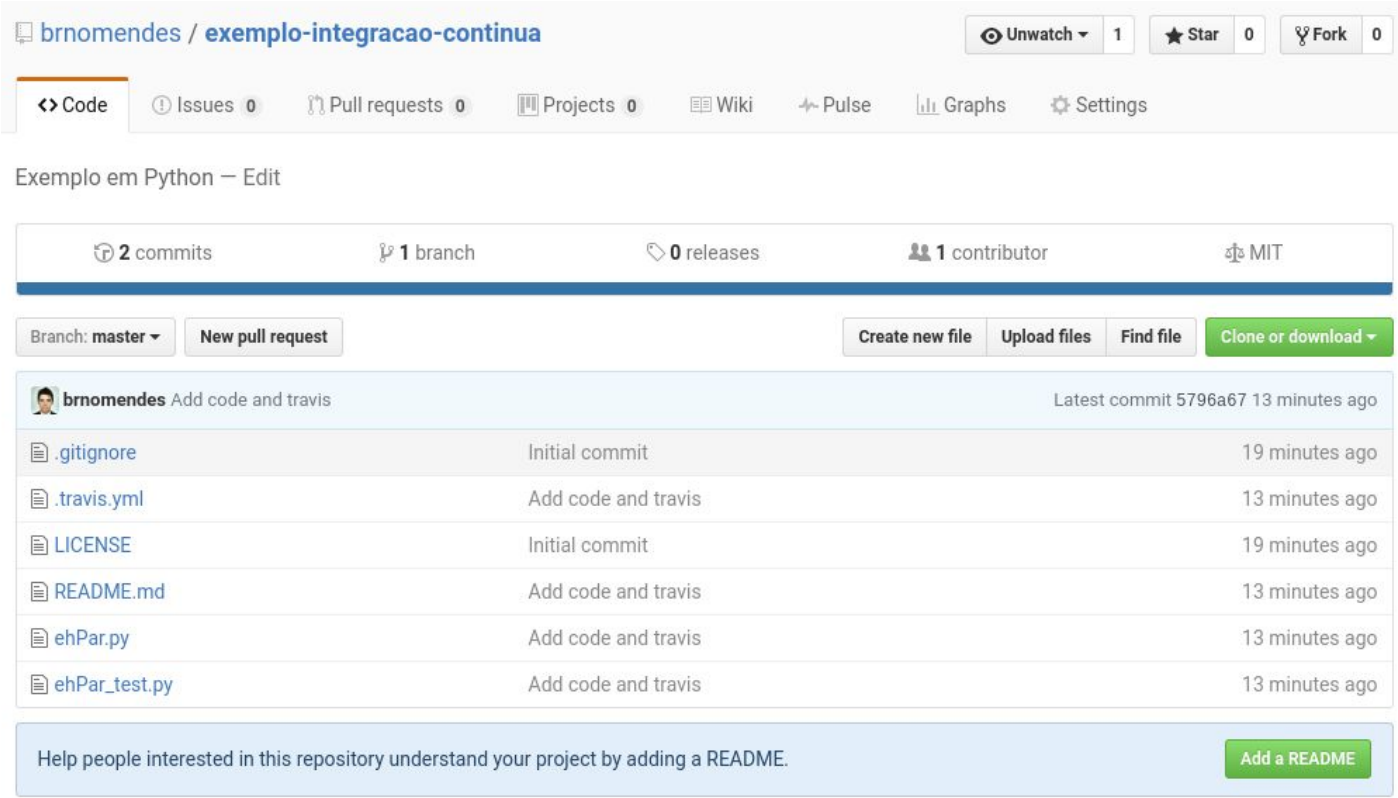
brnomendes/exemplo-integracao-continua

brnomendes/gitlab-gamification

brnomendes/Hemodialise

.travis.yml

```
language: python
python:
  - "3.5"
script: python3 ehPar_test.py
```


brnomendes / exemplo-integracao-continua
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
Exemplo em Python — Edit
2 commits
1 branch
0 releases
1 contributor
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
brnomendes Add code and travis
Latest commit 5796a67 13 minutes ago
.gitignore Initial commit 19 minutes ago
.travis.yml Add code and travis 13 minutes ago
LICENSE Initial commit 19 minutes ago
README.md Add code and travis 13 minutes ago
ehPar.py Add code and travis 13 minutes ago
ehPar_test.py Add code and travis 13 minutes ago
Help people interested in this repository understand your project by adding a README.
Add a README


brnomendes / exemplo-integracao-continua
build passing
Current
Branches
Build History
Pull Requests
More options
master
pass
#2 passed
30 sec
Bruno Mendes
6fa2b03
3 minutes ago
master
Add code and travis
#1 passed
32 sec
Bruno Mendes
5796a67
8 minutes ago

```python
import unittest
from ehPar import ehPar

class ehPar_test(unittest.TestCase):

    def teste(self):
        self.assertEqual(ehPar(10), False)


unittest.main()
```

brnomendes / exemplo-integracao-continua

Current | Branches | Build History | Pull Requests | More options

✗ master	you shall not pass	#3 failed	35 sec
Bruno Mendes		948489a	3 minutes ago
✓ master	pass	#2 passed	30 sec
Bruno Mendes		6fa2b03	13 minutes ago
✓ master	Add code and travis	#1 passed	32 sec
Bruno Mendes		5796a67	17 minutes ago

```
124 If you require sudo, add 'sudo: required' to your .travis.yml
125 See https://docs.travis-ci.com/user/workers/container-based-infrastructure/ for details.
126 $ source ~/virtualenv/python3.5/bin/activate                                    0.00s
127
128 $ python --version
129 Python 3.5.2
130 $ pip --version
131 pip 8.1.2 from /home/travis/virtualenv/python3.5.2/lib/python3.5/site-packages (python 3.5)
132 Could not locate requirements.txt. Override the install: key in your .travis.yml to install dependencies.
133 $ python3 ehPar_test.py                                                         0.07s
134 F
135 ======================================================================
136 FAIL: teste (__main__.ehPar_test)
137 ----------------------------------------------------------------------
138 Traceback (most recent call last):
139   File "ehPar_test.py", line 7, in teste
140     self.assertEqual(ehPar(10), False)
141 AssertionError: True != False
142
143 ----------------------------------------------------------------------
144 Ran 1 test in 0.001s
145
146 FAILED (failures=1)
147
148
149 The command "python3 ehPar_test.py" exited with 1.
150
151 Done. Your build exited with 1.
```

Top ▲

Broken: brnomendes/exemplo-integracao-continua#3 (master - 948489a)

Entrada x

Travis CI <builds@travis-ci.org> 18:20 (Há 2 minutos)

para brunosouza

inglês > português Traduzir mensagem Desativar para: inglês ×

brnomendes / exemplo-integracao-continua (master)

Build #3 was broken. 35 seconds

Bruno Mendes **948489a Changeset →**

you shall not pass

Want to know about upcoming build environment updates?

Would you like to stay up-to-date with the upcoming Travis CI build environment updates? We set up a mailing list for you! Sign up here.

Documentation about Travis CI

Jenkins

O que é Jenkins?

- O Jenkins é um servidor de integração contínua open source e é feito em Java, que disponibiliza mais de 1000 plugins para suportar construção (build) e testes de qualquer tipo de projeto.

O que o Jenkins pode fazer?

- Extensível
- Plugins
- Distribuído
- Build pipeline

Vantagens do Jenkins?

- Extensível
- Plugins

Desvantagens do Jenkins?

- Manutenção do servidor em que o Jenkins está instalado
- Tempo para realizar a configuração


Vs

Travis CI **vs** Jenkins

Travis	**Jenkins**
Comercial	Open source
Serviço	Aplicação
Convenção	Configuração
Fácil de usar	Flexível

Testes e Integração Contínua

Bruno Mendes
Mairieli Wessel