



PUC Minas  
Virtual

# Controle de Versão e CI

Renan Santos Mendes



PUC Minas  
Virtual

# Resumindo

- **Controle de Versão:** é a prática de gerenciar código por meio de versões acompanhando as revisões e o histórico de alterações para facilitar a revisão e a recuperação do código.
- **Integração contínua:** é uma prática de desenvolvimento de software em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central.



PUC Minas  
Virtual

# Git e GitHub

# Instalação Git

Para continuarmos façam as seguintes instalações:

- Conheça e instale o Git - <https://git-scm.com/>
- Conheça e crie uma conta no GitHub - <https://github.com/about>

# O que é Git?



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

[Link para download](#)

*Imagem via git-scm.com*

# Características do Git

Sobre o Git:

- **Branching and Merging;**
- **Pequeno e rápido;**
- Distribuído;
- **Staging Area;**
- **Gratuito e Aberto.**

# Características do GitHub

Sobre o GitHub:

- **Colaborativo;**
- **Automações e CI/CD;**
- Segurança;
- Gerenciamento de Projetos;
- Administração de Times;
- **Comunidade;**





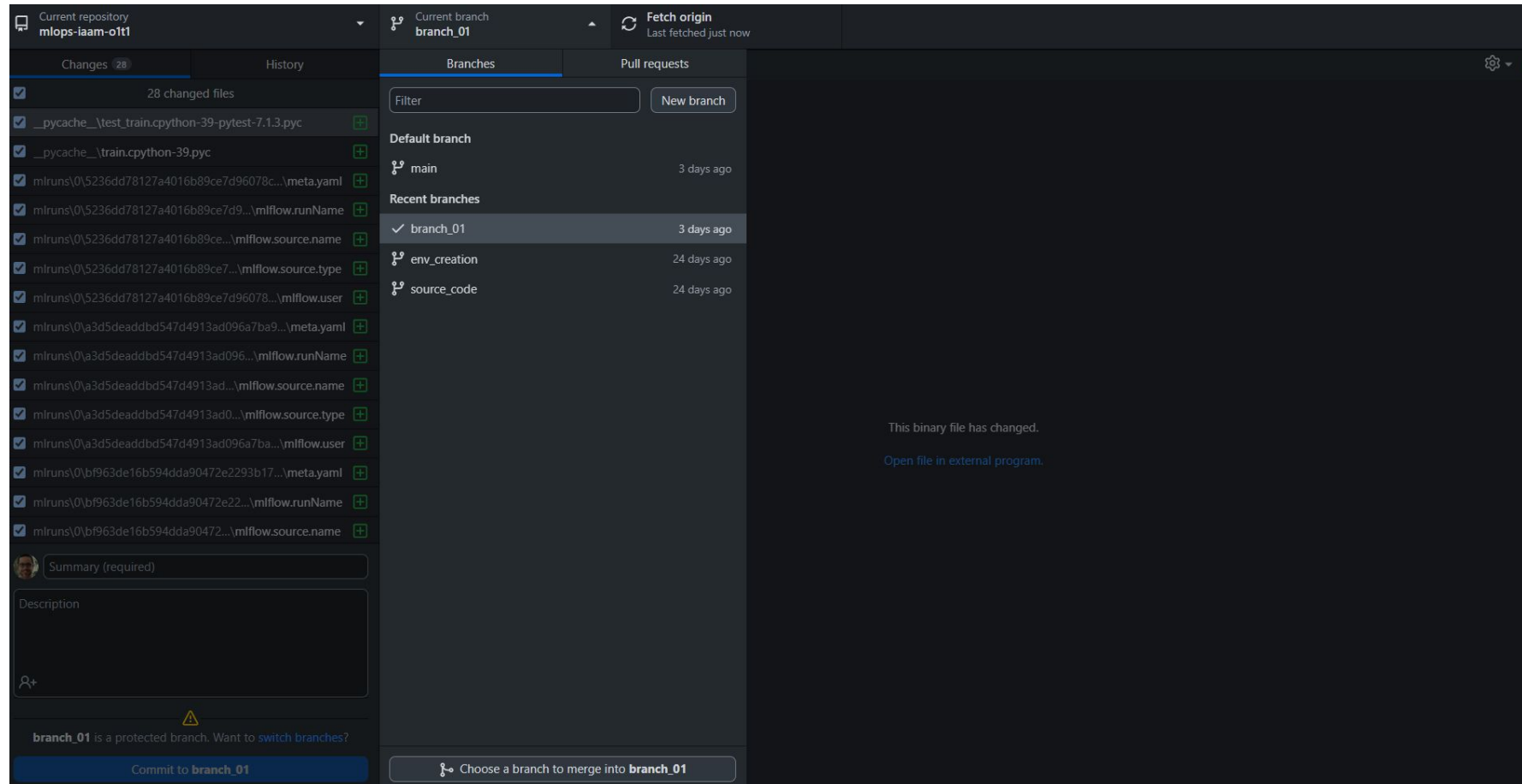
PUC Minas  
Virtual

# GitHub Desktop

# GitHub Desktop

- [Link](#) para download;
- Interface **intuitiva**;
- Controle de versão **simplificado**;
- **Colaboração** eficiente;
- Gerenciamento de **conflitos**;
- Histórico e visualização de alterações;
- Disponibilidade **multiplataforma**.

# GitHub Desktop



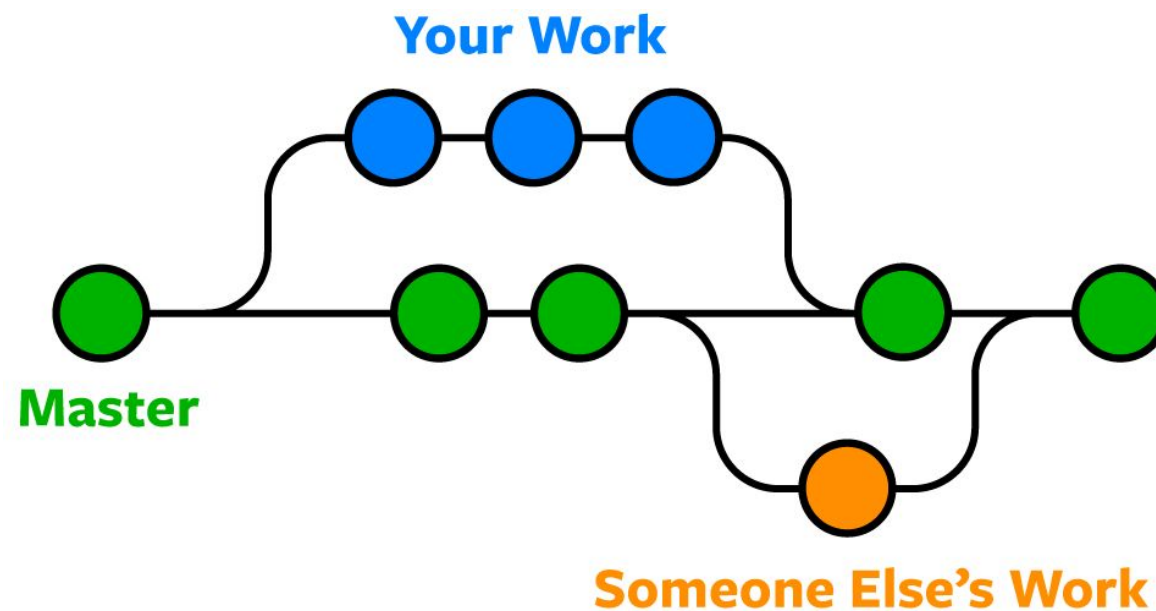


PUC Minas  
Virtual

# Funcionamento do Git

# Funcionamento do Git

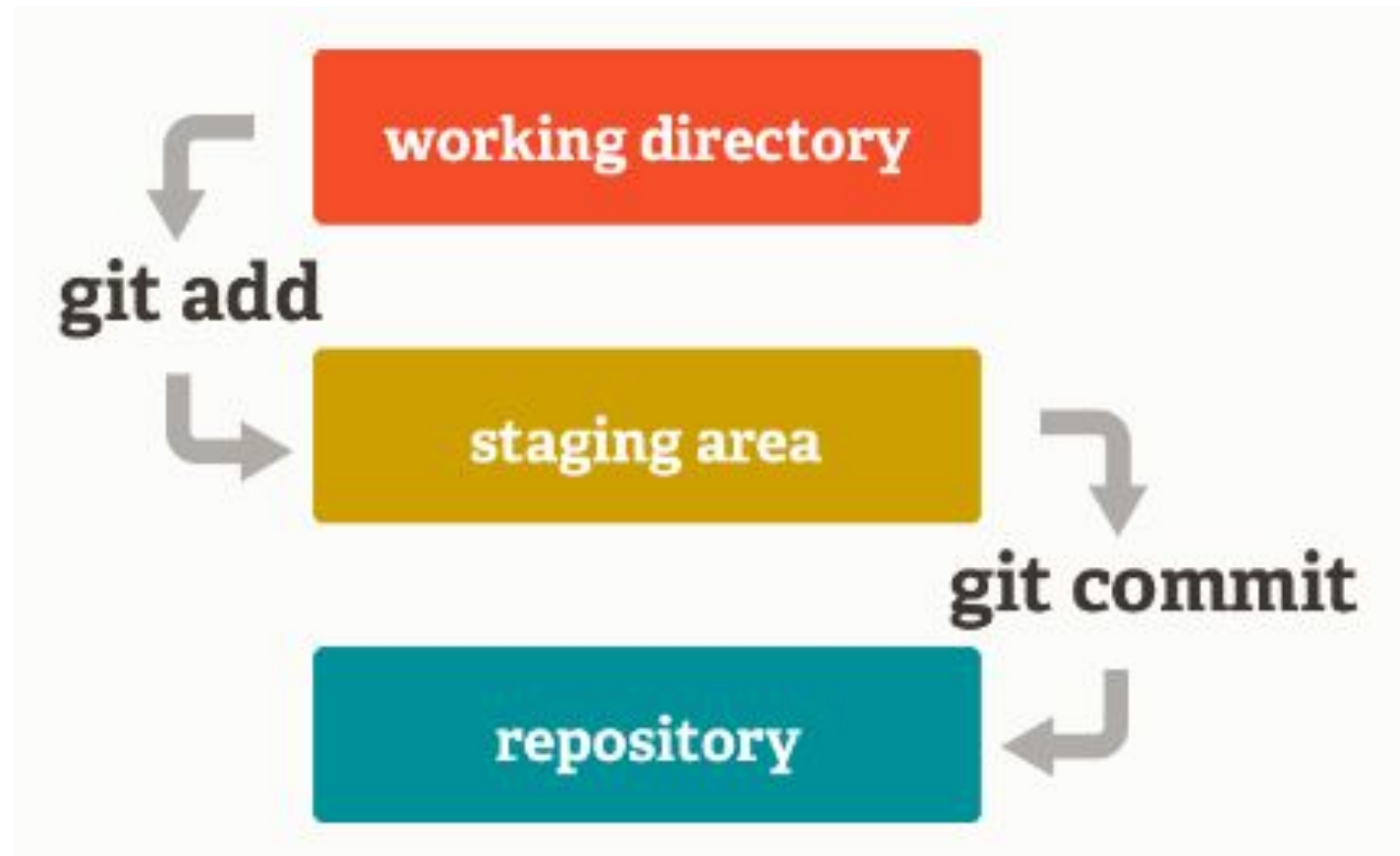
- Um esquema básico de funcionamento do Git



*Imagem via git-scm.com*

# Adicionando alterações no repositório

# Estrutura do Git



*Imagem via git-scm.com*





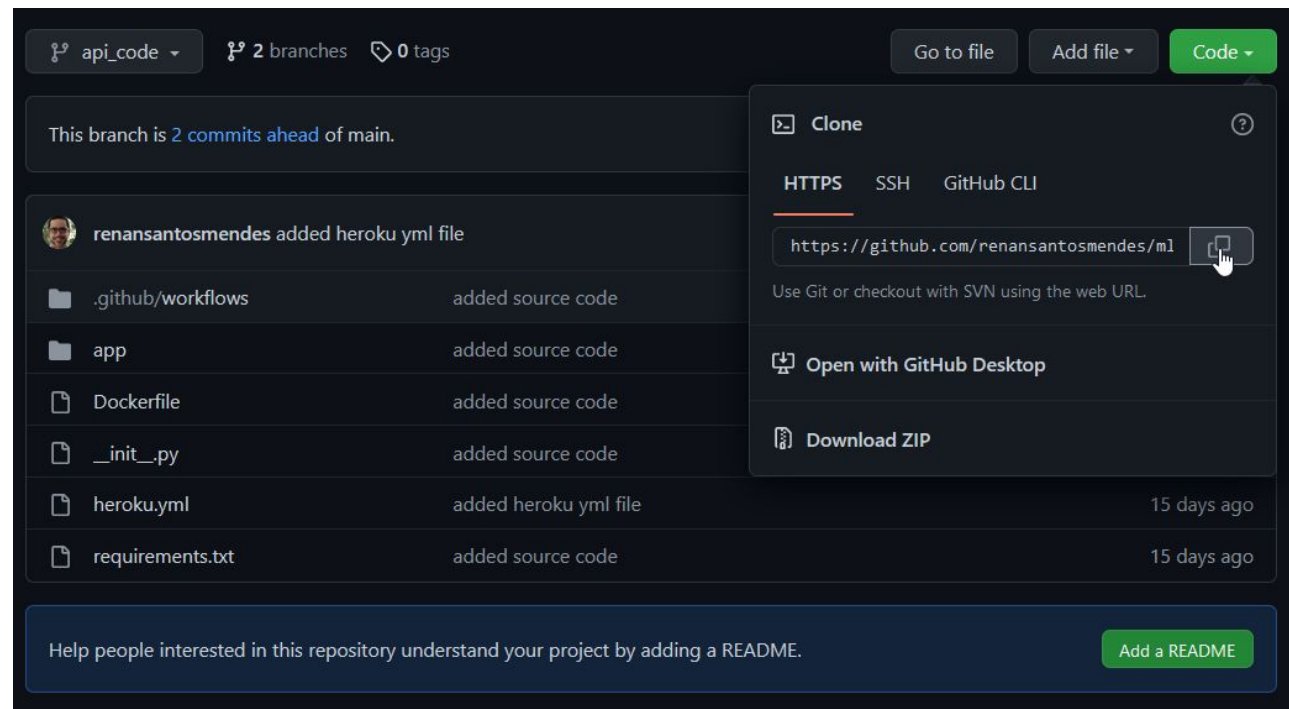
PUC Minas  
Virtual

# Alguns comandos no Git



# git clone

git clone <url\_do\_repositorio>



```
PS C:\PUC\03 - MLOps\Aula 01> git clone https://github.com/renansantosmendes/mlospuc.git
Cloning into 'mlospuc'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 17 (delta 1), reused 12 (delta 1), pack-reused 0
Unpacking objects: 100% (17/17), 2.42 KiB | 70.00 KiB/s, done.
```

# git checkout

git checkout -b <nome\_do\_ramo>

```
PS C:\01 - Estudo\PUC\ML0ps\Aula01\puc_mlops> git checkout -b env_creation  
Switched to a new branch 'env_creation'  
PS C:\01 - Estudo\PUC\ML0ps\Aula01\puc_mlops> |
```

# Adicionar alterações - git add + git commit

- git status
- git add
- git commit -m <mensagem>
- git push

```
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git status
On branch env_creation

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        environment.yml
        requirements.txt
```

nothing added to commit but untracked files present (use "git add" to track)

```
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git add .\environment.yml .\requirements.txt
```

```
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git status
```

On branch env\_creation

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

```
new file:   environment.yml
new file:   requirements.txt
```

```
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git commit -m 'added enviroment files'
[env_creation (root-commit) 3ecf0c0] added enviroment files
 2 files changed, 11 insertions(+)
 create mode 100644 environment.yml
 create mode 100644 requirements.txt
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git push
fatal: The current branch env_creation has no upstream branch.
To push the current branch and set the remote as upstream, use
```

```
git push --set-upstream origin env_creation
```

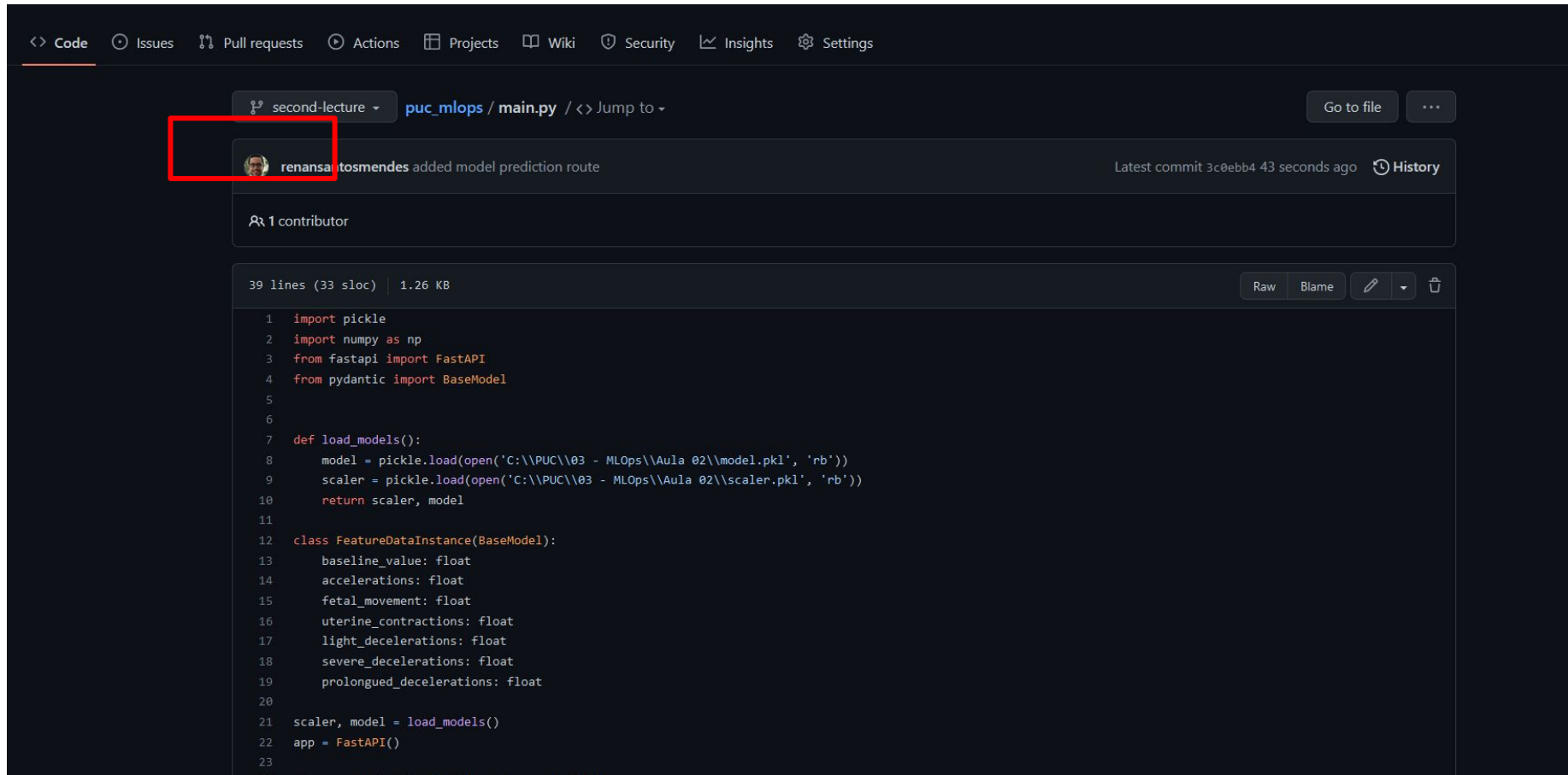
```
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> git push --set-upstream origin env_creation
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 424 bytes | 424.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/renansantosmendes/puc_mlops.git
 * [new branch]      env_creation -> env_creation
Branch 'env_creation' set up to track remote branch 'env_creation' from 'origin'.
PS C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> |
```

# Abrindo um PR ou MR

# Abrindo um *Pull Request*

- O processo de enviar o **código de um ramo para o código principal** é denominado **Pull Request**, ou conhecido como **PR**;
- No Git pode ser feito de duas formas:
  - Comando `git request-pull`;
  - Pela interface;

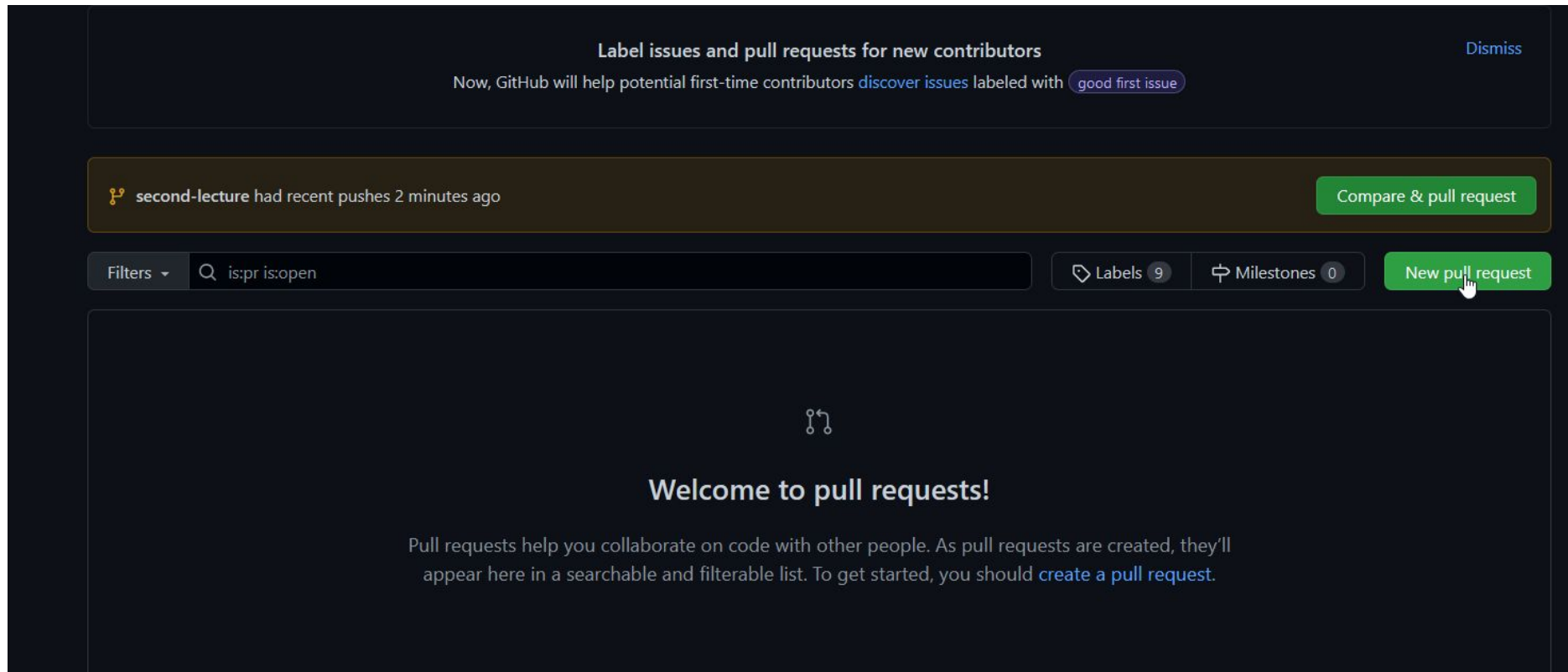
# Abrindo um *Pull Request*



The screenshot shows a GitHub Pull Request interface. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below these, the repository name 'puc\_mlops' and the file path '/ main.py' are displayed. A red box highlights the pull request title 'second-lecture' and the contributor's profile picture and name 'renansantosmendes'. The commit message 'added model prediction route' is visible, along with the latest commit hash '3c0ebb4' and the time '43 seconds ago'. The code editor shows the following Python code:

```
1 import pickle
2 import numpy as np
3 from fastapi import FastAPI
4 from pydantic import BaseModel
5
6
7 def load_models():
8     model = pickle.load(open('C:\\PUC\\03 - MLOps\\Aula 02\\model.pkl', 'rb'))
9     scaler = pickle.load(open('C:\\PUC\\03 - MLOps\\Aula 02\\scaler.pkl', 'rb'))
10    return scaler, model
11
12 class FeatureDataInstance(BaseModel):
13     baseline_value: float
14     accelerations: float
15     fetal_movement: float
16     uterine_contractions: float
17     light_decelerations: float
18     severe_decelerations: float
19     prolonged_decelerations: float
20
21 scaler, model = load_models()
22 app = FastAPI()
23
24 @app.get("/")
25 def root():
26     return {"message": "Hello World"}
```


# Abrindo um *Pull Request*



# Abrindo um *Pull Request*

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

 base: env\_creation

←

compare: second-lecture

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request


- 3 commits



2 files changed

1 contributor


Commits on Oct 12, 2022



added uvicorn package install

 renasantosmendes committed 1 hour ago


 f758b80 



removed unused imports

 renasantosmendes committed 1 hour ago

 9760f85 

added model prediction route

 renasantosmendes committed 5 minutes ago

 3c0ebb4 

PUC Minas **Virtual**



# Abrindo um *Pull Request*

Showing 2 changed files with 33 additions and 3 deletions.

Split Unified


35 main.py

```
... @@ -1,10 +1,39 @@
1 1 import pickle
2 - import sklearn
3 - from joblib import load
4 + import numpy as np
5 3 from fastapi import FastAPI
6 + from pydantic import BaseModel
7
8 +
9 + def load_models():
10 +     model = pickle.load(open('C:\\PUC\\03 - MLOps\\Aula 02\\model.pkl', 'rb'))
11 +     scaler = pickle.load(open('C:\\PUC\\03 - MLOps\\Aula 02\\scaler.pkl', 'rb'))
12 +     return scaler, model
13 +
14 + class FeatureDataInstance(BaseModel):
15 +     baseline_value: float
16 +     accelerations: float
17 +     fetal_movement: float
18 +     uterine_contractions: float
19 +     light_decelerations: float
20 +     severe_decelerations: float
21 +     prolonged_decelerations: float
22 +
23 + scaler, model = load_models()
24
25 app = FastAPI()
```

# Abrindo um *Pull Request*

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).




base: env\_creation

←

compare: second-lecture

✓ Able to merge. These branches can be automatically merged.



Second lecture

Write

Preview

H

B

I

≡

<>

🔗

☰

☷

🔍

👤

🔄

↩

In this PR.....

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

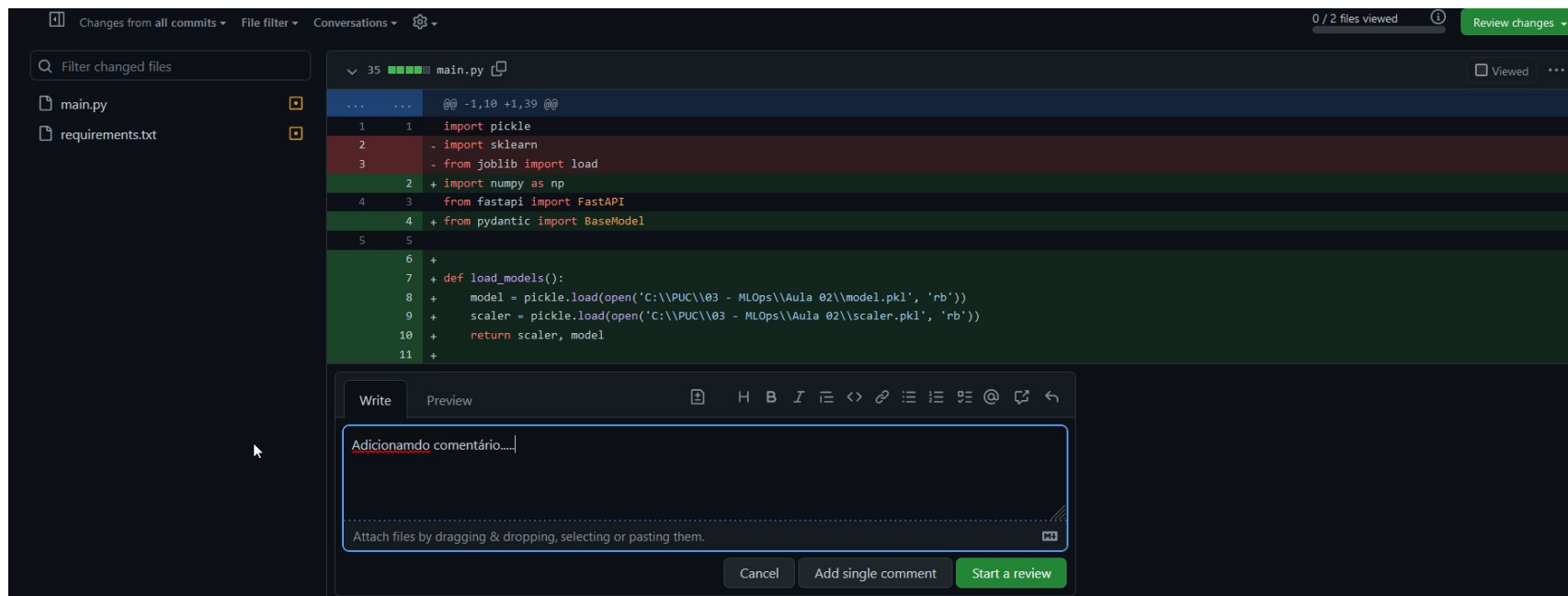
Projects

None yet

Milestone

No milestone

# Abrindo um *Pull Request*





**PUC Minas**  
**Virtual**



PUC Minas  
Virtual

# Ambientes de Desenvolvimento

Renan Santos Mendes





PUC Minas  
Virtual

# Anaconda e Python

# Python e Anaconda

- **Python** é uma das linguagens de programação mais populares do mundo
  - Análise e Ciência de Dados
  - Inteligência Artificial e Aprendizado de Máquina
  - Desenvolvimento Web & Server

# Python e Anaconda

- **Anaconda** é uma distribuição de Python para programação científica que facilita o gerenciamento de pacotes e a implantação em múltiplos sistemas operacionais.
  - Python
  - Conda <https://www.anaconda.com/>
- **PyCharm**: Ferramenta para desenvolvimento de código;





PUC Minas  
Virtual

# Arquivos de ambiente

# Arquivos de ambiente

- Criação de ambientes permite o uso de diferentes versões de bibliotecas;
- A troca de ambientes pode ser feita de forma **rápida e simples**;
- Ambiente virtual = ambiente isolado;
- Existem diversas formas de criar os ambientes:
  - venv
  - pyenv
  - pipenv
  - **conda env**

# Arquivos de ambiente

```
name: mlops-env
dependencies:
  - python=3.9
  - pip
  - pip:
    - -r requirements.txt
```

```
fastapi==0.85.0
mlflow==1.29.0
scikit-learn==1.1.2
pandas==1.5.0
matplotlib==3.6.1
```

# Arquivos de ambiente

Comandos para a criação e ativação de um **ambiente**:

- `conda env create -f environment.yml`
- `conda activate mlops-env`
- `conda deactivate`

Comandos para a criação de um **ambiente vazio**:

- `conda create --name myenv python=3.9`
- `conda activate myenv`
- Ex.: `pip install "tensorflow<2.11"`

```

Selecionar Anaconda Prompt (Anaconda3)
Collecting prometheus-client
  Using cached prometheus_client-0.14.1-py3-none-any.whl (59 kB)
Collecting charset-normalizer<3,>=2
  Using cached charset_normalizer-2.1.1-py3-none-any.whl (39 kB)
Collecting certifi>=2017.4.17
  Downloading certifi-2022.9.24-py3-none-any.whl (161 kB)
----- 161.1/161.1 kB 9.4 MB/s eta 0:00:00
Collecting idna<4,>=2.5
  Using cached idna-3.4-py3-none-any.whl (61 kB)
Collecting greenlet!=0.4.17
  Downloading greenlet-1.1.3.post0-cp39-cp39-win_amd64.whl (101 kB)
----- 101.9/101.9 kB 5.7 MB/s eta 0:00:00
Collecting sniffio>=1.1
  Using cached sniffio-1.3.0-py3-none-any.whl (10 kB)
Collecting smmap<6,>=3.0.1
  Using cached smmap-5.0.0-py3-none-any.whl (24 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp39-cp39-win_amd64.whl (17 kB)
Building wheels for collected packages: databricks-cli
  Building wheel for databricks-cli (setup.py): started
  Building wheel for databricks-cli (setup.py): finished with status 'done'
  Created wheel for databricks-cli: filename=databricks_cli-0.17.3-py3-none-any.whl size=139084 sha256=8a176d1bb66807b7b588373a7fcb900bb8a8be5b08ef9c58cb42bccccd57c184
  Stored in directory: c:\users\renan\appdata\local\pip\cache\wheels\7b\ef\c5\85718fa9e66dec117e42d8b4d7b8a2e40ebdec17232935615f
Successfully built databricks-cli
Installing collected packages: pywin32, pytz, zipp, websocket-client, waitress, urllib3, typing-extensions, threadpoolctl, tabulate, sqlparse, sniffio, smmap, six, pyyaml, pyparsing, pyjwt, protobuf, prometheus-
client, pillow, oauthlib, numpy, MarkupSafe, kiwisolver, joblib, itsdangerous, idna, greenlet, fonttools, entrypoints, cyclr, colorama, cloudpickle, charset-normalizer, certifi, Werkzeug, sqlalchemy, scipy, req
uests, querystring-parser, python-dateutil, pydantic, packaging, Mako, Jinja2, importlib-metadata, gitdb, contourpy, click, anyio, starlette, scikit-learn, pandas, matplotlib, gitpython, Flask, docker, databrick
s-cli, alembic, prometheus-flask-exporter, fastapi, mlflow
Successfully installed Flask-2.2.2 Jinja2-3.1.2 Mako-1.2.3 MarkupSafe-2.1.1 Werkzeug-2.2.2 alembic-1.8.1 anyio-3.6.1 certifi-2022.9.24 charset-normalizer-2.1.1 click-8.1.3 cloudpickle-2.2.0 colorama-0.4.5 contour
py-1.0.5 cyclr-0.11.0 databricks-cli-0.17.3 docker-6.0.0 entrypoints-0.4 fastapi-0.85.0 fonttools-4.37.4 gitdb-4.0.9 gitpython-3.1.29 greenlet-1.1.3.post0 idna-3.4 importlib-metadata-4.13.0 itsdangerous-2.1.2
joblib-1.2.0 kiwisolver-1.4.4 matplotlib-3.6.1 mlflow-1.29.0 numpy-1.23.3 oauthlib-3.2.1 packaging-21.3 pandas-1.5.0 pillow-9.2.0 prometheus-client-0.14.1 prometheus-flask-exporter-0.20.3 protobuf-4.21.7 pydanti
c-1.10.2 pyjwt-2.5.0 pyparsing-3.0.9 python-dateutil-2.8.2 pytz-2022.4 pywin32-304 pyyaml-6.0 querystring-parser-1.2.4 requests-2.28.1 scikit-learn-1.1.2 scipy-1.9.2 six-1.16.0 smmap-5.0.0 sniffio-1.3.0 sqlalchemy-1.4.41
sqlparse-0.4.3 starlette-0.20.4 tabulate-0.9.0 threadpoolctl-3.1.0 typing-extensions-4.4.0 urllib3-1.26.12 waitress-2.1.2 websocket-client-1.4.1 zipp-3.9.0

done
#
# To activate this environment, use
#
#     $ conda activate mlops-env
#
# To deactivate an active environment, use
#
#     $ conda deactivate

```

# Arquivos de ambiente

```
(base) C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops>conda activate mlops-env  
(mlops-env) C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops> conda deactivate  
(base) C:\01 - Estudo\PUC\MLOps\Aula02\puc_mlops>
```





**PUC Minas**  
**Virtual**

# Pipelines para Treino

Renan Santos Mendes





PUC Minas  
Virtual

# Pipelines

# Pipelines

- Uma pipeline é uma **sequência automatizada de processos** no desenvolvimento de software.
- A pipeline é composta por várias etapas;
- A pipeline **automatiza** e **facilita** o desenvolvimento e a entrega;
- Ela é altamente **personalizável** e pode ser adaptada às necessidades específicas de uma equipe ou projeto;
- As pipelines ajudam a garantir a **qualidade**, detectar e **corrigir problemas** e melhorar a **colaboração**.



PUC Minas  
Virtual

# GitHub Actions

# GitHub Actions

- Uma pipeline no GitHub Actions é um **fluxo automatizado de etapas** (ou jobs)
- São executadas em resposta a eventos específicos no seu repositório do GitHub.
- Uma pipeline é definida em um arquivo chamado **workflow**, armazenado no diretório **.github/workflows** do seu repositório.

# GitHub Actions

- O arquivo workflow contém instruções para o GitHub Actions executar as etapas desejadas em resposta a eventos como push de código ou abertura de pull request.
- A pipeline é composta por **um ou mais jobs**, que podem ser executados em **paralelo ou em sequência**, dependendo da configuração.

# GitHub Actions

## Vantagens

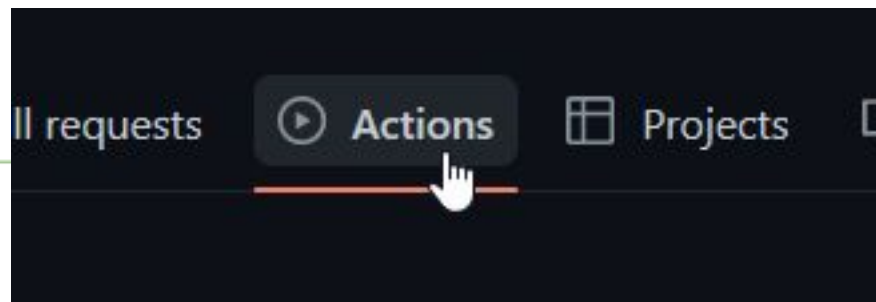
- Simples
- Integrado ao GitHub
- Funciona para CI & CD
- Flexível
- “Marketplace” de elementos prontos
- <https://docs.github.com/en/actions>

# GitHub Actions

The screenshot shows the GitHub interface for the repository 'renansantosmendes / kaggle\_datasets'. The repository is public and has 1 branch and 0 tags. The main content area displays a commit history table with the following entries:

Commit Message	Commit Hash	Date	Commits
renansantosmendes removed covid dataset	0a40b34	on 10 Aug	6 commits
KaggleV2-May-2016.csv added sus dataset		3 months ago	
fetal_health.csv added fetal health dataset		3 months ago	
heart_failure_clinical_records_dataset.csv added heart failure dataset		3 months ago	
insurance.csv added insurance dataset		3 months ago	

Below the commit history, there is a prompt to 'Add a README' to help people understand the project. On the right side, the 'About' section shows 0 stars, 1 watching, and 1 fork. The 'Releases' section indicates that no releases have been published.





# GitHub Actions

## Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

🔍 Search workflows

### Suggested for this repository

#### Simple workflow

By GitHub

Start with a file with the minimum necessary structure.

Configure

### Deployment

[View all](#)

#### Deploy Node.js to Azure Web App



By Microsoft Azure

Build a Node.js project and deploy it to an Azure Web App.

Configure

Deployment

#### Deploy to Amazon ECS



By Amazon Web Services

Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.

Configure

Deployment

#### Build and Deploy to GKE



By Google Cloud

Build a docker container, publish it to Google Container Registry, and deploy to GKE.

Configure

Deployment

#### Terraform



By HashiCorp

Set up Terraform CLI in your GitHub Actions workflow.

Configure

Deployment

#### Deploy to Alibaba Cloud ACK



By Alibaba Cloud

#### Deploy to IBM Cloud Kubernetes Service



By IBM

#### Tencent Kubernetes Engine



By Tencent Cloud

#### OpenShift



By Red Hat

Build a Docker-based project and deploy

# GitHub Actions

The screenshot shows the GitHub Actions interface for the repository 'renansantosmendes / deploy'. The top navigation bar includes links for Code, Issues, Pull requests, Actions (selected), Projects, Wiki, Security, Insights, and Settings. On the right, there are buttons for Pin, Unwatch (1), Fork (0), and Star (0). The left sidebar shows the 'Actions' tab with a 'New workflow' button and a list of workflows, including 'FastAPI-GitHub-Heroku'. The main area displays 'All workflows' with a search bar 'Filter workflow runs'. Below this, a table shows the last 6 workflow runs:

	Event	Status	Branch	Actor
✓ added api route FastAPI-GitHub-Heroku #6: Commit 60a1487 pushed by renansantosmendes	source_code	36 minutes ago 34s		...
✓ fixed test file FastAPI-GitHub-Heroku #5: Commit a74c3b5 pushed by renansantosmendes	source_code	40 minutes ago 30s		...
✗ updated requirements file FastAPI-GitHub-Heroku #4: Commit c4b7729 pushed by renansantosmendes	source_code	42 minutes ago 23s		...
✗ updated pipeline	source_code	1 hour ago		...

# GitHub Actions

The screenshot shows the GitHub Actions interface for a workflow run. The top section indicates the workflow is successful with a green checkmark and the title 'fixed test file FastAPI-Git... #5'. Below this, a summary card shows the run was triggered via push 40 minutes ago by user 'renansantosmendes' with commit 'a74c3b5'. The status is 'Success', the total duration is '30s', and there are no artifacts. The left sidebar lists the jobs 'ci\_pipeline' and 'cd\_pipeline', both of which are successful. The 'Run details' section shows the workflow file 'cicd-pipeline.yml' with the trigger 'on: push'. A visual pipeline diagram shows two steps: 'ci\_pipeline' (10s) and 'cd\_pipeline' (6s), both marked as successful.

fixed test file FastAPI-Git... #5

Summary

Jobs

- ci\_pipeline
- cd\_pipeline

Run details

- Usage
- Workflow file

Triggered via push 40 minutes ago

renansantosmendes pushed [a74c3b5](#) [source\\_code](#)

Status	Total duration	Artifacts
Success	30s	—

cicd-pipeline.yml

on: push

ci\_pipeline 10s

cd\_pipeline 6s

# Criando uma pipeline manualmente

# GitHub Actions

Para se criar um arquivo de workflow, uma sequência de passos deve ser feita:

- Criar um arquivo .yml no subdiretório workflows dentro de uma pasta .github;
  - Ficaria da seguinte forma: Projeto/.github/workflows/file.yml
- Dentro do arquivo .yml devem ser criados pelos menos os seguintes passos:
  - Configuração inicial
  - Jobs
  - Steps dentro dos jobs

# GitHub Actions

```
name: FastAPI-GitHub-Heroku

env:
  HEROKU_API_TOKEN: ${ secrets.HEROKU_API_TOKEN }
  HEROKU_APP_NAME: ${ secrets.HEROKU_APP_NAME }

on:
  push:
    branches:
      - main
      - source_code
  pull_request:
    branches:
      - main

jobs:
  ci_pipeline:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v1
        with:
          fetch-depth: 0

      - name: Set up Python 3.9
        uses: actions/setup-python@v1
        with:
          python-version: 3.9
```



**PUC Minas**  
**Virtual**