* **1a Etapa do Trabalho - Criação do ambiente virtual**

**1- Instale o pip executando o código no terminal:**

sudo apt install python3-pip

**2- Instale o git:**

sudo apt install git

**3- Instalar o pyenv**

curl https://pyenv.run | bash

**4- Instalar o pyenv-virtualenv**

sudo git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyenv root)/plugins/pyenv-virtualenv

**5- Abrir e adicionar no arquivo ~/.bashrc**

xed ~/.bashrc

#Para o pyenv

export PYENV\_ROOT="$HOME/.pyenv"

[[ -d $PYENV\_ROOT/bin ]] && export PATH="$PYENV\_ROOT/bin:$PATH"

eval "$(pyenv init -)"

# Para ativar automaticamente os envs

eval "$(pyenv virtualenv-init -)"

#Para o prompt do pyenv aparecer no terminal

export PYENV\_VIRTUALENV\_DISABLE\_PROMPT=1

pyenvVirtualenvUpdatePrompt() {

RED='\[\e[0;31m\]'

GREEN='\[\e[0;32m\]'

BLUE='\[\e[0;34m\]'

RESET='\[\e[0m\]'

[ -z "$PYENV\_VIRTUALENV\_ORIGINAL\_PS1" ] && export PYENV\_VIRTUALENV\_ORIGINAL\_PS1="$PS1"

[ -z "$PYENV\_VIRTUALENV\_GLOBAL\_NAME" ] && export PYENV\_VIRTUALENV\_GLOBAL\_NAME="$(pyenv global)"

VENV\_NAME="$(pyenv version-name)"

VENV\_NAME="${VENV\_NAME##\*/}"

GLOBAL\_NAME="$PYENV\_VIRTUALENV\_GLOBAL\_NAME"

# non-global versions:

COLOR="$BLUE"

# global version:

[ "$VENV\_NAME" == "$GLOBAL\_NAME" ] && COLOR=""

# virtual envs:

[ "${VIRTUAL\_ENV##\*/}" == "$VENV\_NAME" ] && COLOR="$GREEN"

if [ -z "$COLOR" ]; then

PS1="$PYENV\_VIRTUALENV\_ORIGINAL\_PS1"

else

PS1="($COLOR${VENV\_NAME}$RESET)$PYENV\_VIRTUALENV\_ORIGINAL\_PS1"

fi

export PS1

}

export PROMPT\_COMMAND="$PROMPT\_COMMAND pyenvVirtualenvUpdatePrompt;"

**6- Para poder instalar o Python com o pyenv, execute no terminal o código abaixo:**

feche e abra o cmd antes de rodar

sudo apt update && sudo apt install -y build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev curl libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev llvm

pyenv install 3.11.5

pyenv install 3.7.17

**7- Criar um ambiente virtual com pyenv**

pyenv virtualenv <versaoPython> <nomeDoEnv>

mkdir projetoX

cd projetoX

pyenv virtualenv 3.11.5 envprojetoC

pyenv local envprojetoC

python3 --version

pip install django

* Instalação PostgreSQL via Docker no Linux

1. **Instalar o Docker**

sudo apt install -y apt-transport-https ca-certificates curl software-properties-common gnupg lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu jammy stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update && sudo apt install -y docker-ce docker-compose-plugin

1. **Instalar o Portainer via Docker**

sudo docker volume create portainer\_data

sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer-ce:latest

* 1. **Criar um shortcut no desktop para rodar o Portainer (se quiser)**

Clique com o botão direito do mouse no desktop → Create a new launcher here...

Chrome

Name: Portainer

Command: /usr/bin/google-chrome-stable %U https://localhost:9443

No ícone use o do Chrome em /usr/share/icons/Mint-Y/apps/48/google-chrome.png

Firefox

Name: Portainer

Command: /usr/bin/firefox https://localhost:9443

No ícone use o do Chrome em /usr/share/icons/Mint-Y/apps/48/firefox.png

Na 1a execução você vai cadastrar a senha do *admin*, use *123456789123* para não esquecer depois

Selecione o botão com o desenho da baleia com containers em cima do lado esquerdo

1. **Instalar o PostgreSQL via Docker**
   1. **Crie um diretório chamado postgresql (por exemplo dentro de Programas)**

cd ~/Programas

mkdir postgresql

cd postgresql

* 1. **Crie um arquivo chamado *Dockerfile* com o seguinte conteúdo:**

xed Dockerfile

# Version: 1.0

FROM postgres:latest

ENV REFRESHED\_AT 2022-07-30

RUN apt-get update && apt-get install -y locales

RUN touch /usr/share/locale/locale.alias

ENV LANG pt\_BR.UTF-8

ENV LANGUAGE pt\_BR:pt

ENV LC\_ALL pt\_BR.UTF-8

RUN sed -i '/pt\_BR.UTF-8/s/^# //g' /etc/locale.gen && locale-gen && update-locale LANG=pt\_BR.UTF-8

ENV TZ=America/Sao\_Paulo

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

* 1. **Criar a imagem do PostgreSQL que usaremos**

sudo docker build -t="godoi/postgresql" .

* 1. **Criar o container do PostgreSQL**

sudo docker network create rede\_postgresql

sudo docker volume create --name postgresql\_data

sudo docker run -i -t -d --name postgresql --net=rede\_postgresql-p 5432:5432 \

-e POSTGRES\_USER=postgres \

-e POSTGRES\_PASSWORD=12345 \

--volume postgresql\_data:/var/lib/postgresql/data \

godoi/postgresql

1. **Instalar o pgagmin4 como aplicação desktop**

curl -fsS https://www.pgadmin.org/static/packages\_pgadmin\_org.pub | sudo gpg --dearmor -o /usr/share/keyrings/packages-pgadmin-org.gpg

sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg] https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/jammy pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list && apt update'

sudo apt update && sudo apt install -y pgadmin4-desktop

Só para consulta

Linux Mint 21, 21.1 → Ubuntu jammy

Linux Mint 20.3 → Ubuntu focal

-Adicione o servidor do PostgreSQL clicando em *Add Server:*

*General → Name: PostgreSQL Container*

*Connection → Host name/address: localhost*

*Connection → Username: postgres*

*Connection → Password: 12345*

1. **Instalando Django**
   1. **obs:** Caso não tenha conseguido instalar o Pyenv na máquina, não será possível ter continuidade para a instalação do Django.
   2. Atualizar a Lista de Pacotes:

Abra o terminal e atualize a lista de pacotes do seu sistema com o seguinte comando:

* sudo apt update
  1. Instalar o Django:

Você pode instalar o Django usando o pip. O comando abaixo instalará a versão mais recente do Django:

* pip install Django
* Configurando bibliotecas no pyenv

**1-** **Instale as bibliotecas referente ao python**

pip install fastapi

pip install "uvicorn[standard]"

pip install psycopg2-binary

pip install psycopg2

pip install django-bootstrap4

pip install django-crispy-forms

pip install django-extensions

pip install dj-database-url

pip install dj-static

pip install djangorestframework

pip install template

pip install postgre

pip install fastapi

pip install "uvicorn[standard]"

pip install psycopg2-binary

pip install psycopg2

pip install django-bootstrap4

pip install django-crispy-forms

pip install django-extensions

pip install dj-database-url

pip install dj-static

pip install djangorestframework

pip install template

pip install postgre

pip install annotated-types

pip install anyio

pip install asgiref

pip install beautifulsoup4

pip install certifi

pip install charset-normalizer

pip install click

pip install decouple

pip install defusedxml

pip install diff-match-patch

pip install dj-database-url

pip install dj-static

pip install Django

pip install django-bootstrap-form

pip install django-bootstrap-icons

pip install django-bootstrap4

pip install django-crispy-forms

pip install django-extensions

pip install django-import-export

pip install djangorestframework

pip install et-xmlfile

pip install fastapi

pip install h11

pip install httptools

pip install idna

pip install Jinja2

pip install jmespath

pip install MarkupPy

pip install MarkupSafe

pip install odfpy

pip install openpyxl

pip install pip

pip install postgres

pip install psycopg2-binary

pip install psycopg2-pool

pip install pydantic

pip install pydantic\_core

pip install pytz

pip install PyYAML

rpip install equests

pip install setuptools

pip install sniffio

pip install soupsieve

pip install sqlparse

pip install starlette

pip install static3

pip install tablib

pip install template

pip install templates

pip install toml

pip install typing\_extensions

pip install urllib3

pip install uvicorn

pip install uvloop

pip install watchfiles

**pip install websockets**