

## Tutorial de instalação de linguagens, frameworks, ferramentas e banco de dados

### Passo 1: Preparação Inicial

1. Abra um terminal no seu sistema Linux.

2. Certifique-se de que você tenha o Python e o pip instalados. Geralmente, o Python já está instalado em sistemas Linux modernos.

Você pode verificar digitando:

```
python3 --version
```

Se não estiver instalado, você pode instalá-lo com:

```
sudo apt update
```

```
sudo apt install python3 python3-pip
```

### Passo 2: Instalação do Ambiente Virtual

1. Instale a ferramenta pyenv e pyenv-virtualenv para criar ambientes virtuais isolados:

```
curl https://pyenv.run | bash
```

Na raiz de pyenv/ na pasta /plugins faça o clone

```
git clone https://github.com/pyenv/pyenv-virtualenv.git /plugins/pyenv-virtualenv
```

2. Crie um novo diretório para o seu projeto Flask e navegue até ele:

```
mkdir meu_projeto_flask
```

```
cd meu_projeto_flask
```

3. Crie um ambiente virtual:

```
virtualenv venv
```

4. Ative o ambiente virtual:

```
source venv/bin/activate
```

### Passo 3: Instalação do Flask e do Flask-MySQL

1. Com o ambiente virtual ativado, instale o Flask e o Flask-MySQL:

```
pip install Flask Flask-MySQL
```

### Passo 4: Configuração do MySQL

1. Instale o MySQL no seu sistema:

```
sudo apt install mysql-server
```

2. Durante a instalação, você será solicitado a definir uma senha para o usuário **root** do MySQL.

### Passo 5: Inicializando o Banco de Dados

1. Inicie o servidor MySQL:

```
sudo service mysql start
```

2. Confira o status servidor MySQL:

```
sudo service mysql status
```

3. Acesse o shell do MySQL:

```
mysql -u root -p
```

## Passo 6: Configuração do Docker

1. Instale o Docker:

```
sudo apt update
```

```
sudo pip install docker
```

```
sudo apt install docker-ce docker-ce-cli containerd.io
```

2. Inicie o Docker:

```
sudo service docker start
```

3. Confira o status servidor Docker:

```
sudo service docker status
```

## Passo 7: Arquivo “Dockerfile” de configuração da imagem docker

```
# Use the official Python image as the base image
FROM python:3.9

# Set environment variables
# app name
ENV FLASK_APP=run
ENV FLASK_ENV=production
ENV FLASK_RUN_HOST=0.0.0.0

# Set the working directory
WORKDIR /app

# Copy the requirements file to the container
COPY requirements.txt requirements.txt

# Install any needed packages specified in requirements.txt
RUN pip install -r requirements.txt

# Expose the port on which your Flask app will run
EXPOSE 5000

# Copy the rest of your application code into the container
COPY . .

# Start the Flask application
CMD ["flask", "run"]
```

## Passo 8: Arquivo “requirements.txt” de configuração das dependências do docker

Flask==3.0  
SQLAlchemy==2.0.21  
mysqlclient==2.2.0  
Werkzeug==3.0

## Passo 9: Inicializar o projeto usando container docker

### 1. Criando imagem Docker:

**docker** build -t myflaskapp .

### 2. Start no App usando docker:

**docker** run -p 5000:5000 myflaskapp

## Arquitetura do Projeto

```
myflaskapp/ # Diretório raiz do projeto
├── app/ # Diretório do aplicativo Flask
│   ├── __init__.py # Arquivo de inicialização do aplicativo
│   ├── routes.py # Rotas e lógica da aplicação
│   ├── models.py # Definição de modelos de banco de dados
│   ├── templates/ # Diretório de templates HTML
│   └── static/ # Diretório de arquivos estáticos (CSS, JS, imagens)
├── migrations/ # Diretório de migrações do banco de dados (para o Flask-Migrate)
├── config.py # Configuração do aplicativo
├── requirements.txt # Lista de dependências do Python
├── Dockerfile # Arquivo de configuração do Docker para criar a imagem
├── run.py # Script para executar o aplicativo
└── .env # Arquivo de variáveis de ambiente (opcional)
```