

## Tutorial para desenvolvimento de projeto em Flask e MySQL

### 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: Configuração do Ambiente Virtual

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

```
sudo apt install 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: Criação de Banco de Dados

1. Inicie o servidor MySQL:

```
sudo service mysql start
```

2. Acesse o shell do MySQL:

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

3. Execute o seguinte comando para melhorar a segurança do MySQL:

```
sudo mysql_secure_installation
```

4. Dentro do shell MySQL, crie um novo banco de dados e um usuário para o projeto:

```
CREATE DATABASE meu_db;
```

```
CREATE USER 'meu_usuario'@'localhost' IDENTIFIED BY 'minha_senha';
```

```
GRANT ALL PRIVILEGES ON meu_db.* TO 'meu_usuario'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

5. Instalando o MySQL Workbench

Acesse o terminal e execute o seguinte comando para instalar o MySQL Workbench:

```
sudo apt install mysql-workbench # Para sistemas baseados em Debian (Ubuntu)
```

```
sudo yum install mysql-workbench # Para sistemas baseados em Red Hat (CentOS)
```

6. Configurando o MySQL Workbench

Abra o MySQL Workbench após a instalação.

Clique em "New Connection" para configurar uma nova conexão. Preencha os detalhes como o nome da conexão, nome do host (geralmente "localhost"), nome de usuário (geralmente "root") e senha que você definiu durante a instalação do MySQL.

Clique em "Test Connection" para verificar se a conexão está funcionando corretamente.

Após testar com sucesso, clique em "OK" para salvar a conexão.

## Passo 6: Configuração do Flask para MySQL

1. Crie um arquivo de configuração para o seu aplicativo Flask (por exemplo, config.py) e configure a conexão com o banco de dados MySQL:

```
SQLALCHEMY_DATABASE_URI = 'mysql://meu_usuario:minha_senha@localhost/meu_db'
```

## Passo 7: Criando o Aplicativo Flask

1. Crie uma estrutura de diretórios básica para o seu projeto:

```
mkdir app # criando a pasta principal do projeto
```

```
cd app # entrando na pasta principal do projeto
```

```
mkdir static
```

 # local onde você vai armazenar seus arquivos estáticos, como CSS, JavaScript e imagens.

```
mkdir templates
```

 # local onde você vai armazenar seus arquivos HTML de template.

`touch __init__.py` # configuração de inicialização do projeto

`touch routes.py` # arquivos com definição de rotas do projeto

`touch models.py` # arquivo com modelos de tabelas do banco de dados

2. No arquivo `routes.py`, adicione um exemplo de rota para testar a conexão com o banco de dados:

`from flask import Flask, render_template` # `render_template` é uma biblioteca para renderizar arquivos `.html`

`from flask_sqlalchemy import SQLAlchemy`

`app = Flask(__name__)`

`app.config.from_pyfile('../config.py')`

`db = SQLAlchemy(app)`

`@app.route('/')`

`def index():`

`return "Hello World!"`

`if __name__ == '__main__':`

`app.run(debug=True)`

## Passo 8: Executando o Aplicativo

1. Certifique-se de estar no diretório onde você criou o ambiente virtual e o projeto.
2. Ative o ambiente virtual caso não esteja ativo:

`source venv/bin/activate`

3. Execute o aplicativo Flask:

`python app/routes.py`