

**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA BAIANO - CAMPUS
GUANAMBI**

GABRIEL SOARES ROCHA

**E-BOOK AUXILIADOR PSW
VOL 1**

**GUANAMBI-BA
2019**

GABRIEL SOARES ROCHA

E-BOOK PSW

VOL 1

Feito unicamente na intenção de ajudá-los(las) na disciplina de Programação de Sistemas Web do prof. Cleyton.

GUANAMBI-BA

2019

SUMÁRIO

1. Lista de passos para as primeiras criações e instalações:.....	4
2. Criação de uma página simples com o flask:.....	4
3. Programas a serem instalados:.....	5
4. Lista de comandos para Linux:.....	6
5. Browser – Banco de Dados:.....	6
6. Criando um banco com o flask-sqlalchemy:.....	7
7. Templates:.....	8
8. Criando o Blog (simples):.....	8
9. Anexos:.....	10

1. Lista de passos para as primeiras criações e instalações:

1. Criar um diretório com seu nome para trabalhar o projeto;
2. Dentro dele abra o terminal e digite “sudo apt install virtualenv” para instalar a “virtualenv” caso não tenha;
3. Logo após digite “virtualenv -p python3 env” para criar uma MV(Máquina Virtual);
 1. nomes padrões “env” ou “venv”;
4. Criada a MV, ative-a com o comando “source env/bin/activate”, assim você terá ela pronta para uso;
 1. Ou o nome que você deu a MV, no caso acima “env”;
5. Depois você irá instalar o FLASK(mini framework), com este comando “pip install flask” dentro da env ativada;
 1. Para conferir a instalação digite “pip freeze”;
6. E para desativar a env use o comando “deactivate”.

2. Criação de uma página simples com o flask:

1. Abra um editor de texto, do próprio Linux ou um programa com os mesmos fins;
2. Agora você irá construir a estrutura da página com códigos em python, salvando o arquivo como o nome que quiser, com a extensão “.py”;
 1. Padrão de nome -”app.py”;
 2. Visualizar (Anexo_1);

3. Feito isso, no terminal dentro da env, se você fez todo direito digite python3 e o nome do seu arquivo.py, ex: "python3 site.py";
4. Como esse comando será gerado um link, com o botão esquerdo click nele e escolha a opção "abrir link" e sua página logo será aberta.

3. Programas a serem instalados:

1. Abra o terminal, de preferência em sua pasta (a criada para o projeto com o seu nome);
2. Agora com os seguintes comandos instale os programas;
 1. "sudo apt install sqlite3" - SQLite é uma biblioteca em linguagem C que implementa um banco de dados SQL embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo SGBD separado, segundo o Wikipédia;
 2. "sudo apt install sqlitebrowser" - É o programa executável que se acessa ao procurá-lo em atividades e digitar "DB Browser for SQLite";
 3. "sudo snap install code" - com este você instala o Visual Studio Code, caso não funcione tente "sudo snap install code --classic";
 4. "pip install flask-sqlalchemy" - Este será instalado na env ativada. O Flask-SQLAlchemy é uma extensão do Flask que adiciona suporte ao SQLAlchemy ao seu aplicativo. O objetivo é simplificar o uso do SQLAlchemy com Flask, fornecendo padrões úteis e auxiliares extras que facilitam a realização de tarefas comuns, segundo o site documental oficial do flask-sqlalchemy;
 5. "pip install ipython" - Assim como o comando acima este também é dentro da env. Ipython é um interpretador interativo para várias linguagens de

programação, mas especialmente focado em Python. Ipython oferece "type introspection", "rich media", syntax shell, completção por tab e edição auxiliada por histórico de comando, segundo o Wikipédia sobre o ipython, e para usá-lo basta digitar "ipython" e para sair "exit", na env.

4. Lista de comandos para Linux:

1. Guia_de_500_comandos_Linux. Disponível em:
<https://www.linuxpro.com.br/dl/guia_500_comandos_Linux.pdf>. Acesso em: 13 de set. 2019.
2. Com este guia você poderá aprender alguns comandos que podem lhe ser útil na hora de programar.

5. Browser – Banco de Dados:

1. Após a instalação do "sqlite3" e do "sqlitebrowser" com os comandos que você viu nos conteúdos **3.2.1** e **3.2.2**, iremos a sua utilização;
2. Primeiramente abra o programa "DB Browser for SQLite" ao digitá-lo em "Atividades". No programa aberto vá em "Novo banco de dados", depois escolha um diretório pra salvá-lo, com um nome e a extensão ".db";
 1. Exemplo, "meuBanco.db";
 2. Visualizar (Anexo_2);
3. Feito isso, abrirá uma tela na qual você irá defini um nome para sua tabela (Anexo_3);
 1. exemplo (Anexo_4);
4. Depois você irá em "Adicionar campo", que adicionará um atributo a tabela criada, na qual pode-se alterar o nome, tipo(integer, text, blob, real e numeric), além das opções Não nulo, PK(Primary Key), AI(automático) e U(Único);

1. Visualizar (Anexo_5);
2. Exemplo (Anexo_6);
5. Feito os campos, selecione a tabela criada (Anexo_7), e vá em “Navegar dados”, nessa área pode-se atribuir valores aos campos criados (Anexo_8);
 1. exemplo (Anexo_9);

6. Criando um banco com o flask-sqlalchemy:

1. Para esse processo você precisará já ter instalado todos programas do item 3, e ter a sua env;
2. No diretório que contém sua **env** crie um arquivo “.py” com um editor de texto contendo os seguintes códigos, visualizar(Anexo_10);
 1. exemplo “DataBase.py”;
3. Depois abrir no terminal a sua env , ative-a. Acesse o “ipython” e insira os dados comandos;
 1. from “nome_do_arq.py” import db;
 1. ex: from DataBase.py import db;
 2. Repita o primeiro - “from “nome_do_arq.py” import db”;
 3. db_create_all();
 4. visualizar (Anexo_11);
4. Após todo este processo você terá em seu diretório juntamente com a sua env um arquivo chamando “banco1.db” e uma pasta “__pycache__”, na qual você pode está acessando pelo DB Browser for SQLite no mesmo esquema do item **5.5**;
 1. OBS: O banco criado recebe o nome de “banco1”, pois está definido no código visualizado no (Anexo_10), no qual você pode alterá-lo com o nome que preferir, sem esquecer de colocar a extensão “.db” ao final.

7. Templates:

1. Dentro do diretório com sua **env** crie um nova pasta com o nome “templates”;
 1. É um nome padrão;
2. Nessa pasta serão armazenados arquivos “.HTML” e outros, que para serem acessados, é inserido ao “`arq.py`” o módulo “`render_template`”;
 1. Este “`arq.py`” pode-se tomar como exemplo o (Anexo_1);
 2. A inserção será feito no seguinte esquema: “`from flask import Flask, render_template`”;
 3. Visualizar (Anexo_12);
3. Feito isso, agora você criará uma rota que ligará o “`arq.html`” dentro da pasta “templates” a página web, no “`arq.py`” na qual está vinculada;
 1. Visualizar (Anexo_13);
 2. Isso após ter criado um “`arq.html`” de preferência com algum conteúdo no diretório “templates”, como no (Anexo_13), “`index.html`”.

8. Criando o Blog (simples):

1. O primeiro passo, é fazer todos os processos do item 1. Feito isso, crie um diretório templates e logo após instale o Virtual Studio Code, caso este não esteja instalado(para a instalação visualizar item 3.2.3);
2. Em seguida, no terminal, ative a sua env e abra o Virtual Studio Code com o comando “`code .`”. O programa será aberto e você visualizará todos os arquivos em seu diretório: Pasta env, pasta templates e `arq.py`;
 1. adicione na primeira linha de código do seu `arq.py` o módulo “`render_template`”, para as criações de rotas de arquivos “.html” da pasta templates, para que possam ser acessadas quando forem chamadas no navegador;
 2. Visualizar (Anexo_14);
3. Posterior, ainda no “code”, vá em seu diretório templates e crie um arquivo, “`base.html`”. No qual servirá de base para os futuros arquivos. Nela digite

“html”, desse modo pode lhe aparecer a opção “HTML:5”, selecione-a, mas se não apenas construa a estrutura básica de uma página HTML;

1. Visualizar (Anexo_15);

4. Dentro do seu arquivo base.html, entre as tags “<title>” e “<body>”, adicione os comandos do “Jinja”. No primeiro escreva “{% block title %}{% endblock title %}”, e no segundo “{% block conteudo %} {% endblock conteudo %}”

1. Visualizar (Anexo_16);

5. Ainda na pasta templates crie outro arquivo, com o nome “blog.html”. Nele você iniciará a primeira linha com o código “{% extends 'base.html' %}”, que fará um ponte com o outro arquivo, no qual importará as funcionalidades dos códigos que forem sendo acrescentados nele, além dos códigos já presentes;

6. Agora com os códigos encontrados no item acima, **8.4**, adicione um título e um conteúdo ao seu arquivo.html (blog.html);

1. Visualizar (Anexo_17);

7. Após a criação e inserção dos devidos códigos no seu arquivo.html (blog.html), vá para o “arq.py” e declare uma rota para sua página;

1. Visualizar (Anexo_18);

8. Para finalizar, teste-o para confirmar sua funcionalidade, vá ao terminal com sua env ativada e digite “python3 nome_do_arquivo.extensão” (python3 arq.py), se aparecer as seguintes telas, seu blog simples estará funcionando;

1. OBS: não esqueça de escrever ao final do link a rota “/blog”, para o acesso da página;

2. Tela01 visualizar (Anexo_19), e tela02 visualizar (Anexo_20).

9. Anexos: