**Comandos:**

Criar pasta: **Mkdir** **“nome da pasta”**

Acessar pasta: **Cd “nome da pasta”**

Sair da pasta: **cd ..**

Ver a pasta: **ls**

Para limpar o terminal: **clear**

**Ativando:**

1. Criar uma pasta para o projeto.
2. Abra um terminal em **“New Terminal”**.
3. **Cd “nome da pasta”**.
4. Cria um ambiente em python: **py –m venv projeto**.
5. **Cd projeto**.
6. **Cd Scripts**.
7. **(Opcional)** Código de liberar o Windows: **Set-ExecutionPolicy –ExecutionPolicy RemoteSigned –Scope CurrentUser**
8. Use “**.\activate”** para ativar o código. Ou .\projeto\Scripts\Activate.ps1
9. Volte para **projeto**(cd ..)
10. Depois use **pip install flask**.

**Arquivos:**

1. Volte para a pasta do projeto e crie um arquivo **“pagina.py”** na pasta **projeto**.
2. Digite no arquivo python:

from flask import Flask, render\_template, request

import mysql.connector

app = Flask (\_\_name\_\_)

@app.route("/")

def index():

    return render\_template("index.html")

@app.route("/about")

def mostraAbout():

    return render\_template("about.html")

@app.route("/contact")

def mostraContact():

    return render\_template("contact.html")

app.run()

1. Em **\projeto** crie uma pasta chamada **templates (DESTE JEITO)** para colocar os arquivos HTML.
2. Depois crie os arquivos HTML para cada uma das rotas.
3. Vá para **\projeto** e no terminal digite: **python.exe -u .\pagina.py** para upar o site.

**HTML:**

*Dica:* Criar um arquivo HTML chamado **base.html** para usar de modelo para outras páginas, neste arquivo pode-se criar um:

{% block conteudo %}

{% endblock %}

Nele pode-se instanciar “miolos” das outras páginas em cima do molde de **base.html**.

<!DOCTYPE *html*>

<html *lang*="pt-br">

<head>

    <meta *charset*="UTF-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Site</title>

</head>

<body>

    {% block conteudo %}

    {% endblock %}

</body>

</html>

Nos outros arquivos use:

{% extends 'base.html' %}

{% block conteudo %}

<h1>

    Página do About:

    <p>

        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Modi vero ut non fuga? Temporibus nihil nulla sed aperiam provident error quis atque! Similique, eum neque necessitatibus mollitia nam vitae excepturi!

    </p>

</h1>

{% endblock %}

Faça estes “miolos” para colocar no **base.html**.

Para colocar a resposta do python no HTML:

{{res\_imc}}

Fazendo IFs no HTML:

 {% if res\_classificacao == "IMC magro" %}

        <br>

        <img *src*="static/magrao.jpg">

    {% else %}

        {% if res\_classificacao == "IMC sobrepeso" %}

            <br>

            <img *src*="static/gorda.webp" *alt*="">

        {% endif %}

    {% endif %}

**Criando links para navegação:**

Crie um link no HTML com **<a>** e linkar a rota da página:

<a *href*="/paginadois">pagina 2</a>

**Linkando o HTML com CSS:**

1. Criar uma pasta chamada **static** para colocar o CSS e fotos.
2. Linkando o CSS do BootStrap:

<link *rel*="stylesheet" *type*="text/css" *href*="static/bootstrap.css">

**Get e post:**

1. O método get deixa na URL os atributos e valores e o post não.

Post para coletar os dados:

@app.route("/cadastro\_usuario", *methods* =['POST'])

def cadastro():

    nome = request.form['txt\_nome']

    cpf = request.form['txt\_cpf']

    email = request.form['txt\_email']

    senha = request.form['txt\_senha']

    db = mysql.connector.connect(*host* = '201.23.3.86',

*port*= 5000,

*user*= 'usr\_aluno',

*password*= 'E$tud@\_m@1$',

*database*= 'aula\_fatec')

    mycursor = db.cursor()

    query = 'INSERT INTO Hugo\_IIItbusuario ( nome, cpf, email, senha) VALUES (%s, %s, %s, %s)'

    values = (nome, cpf, email, senha)

    mycursor.execute(query, values)

    db.commit()

    return 'gravou'

Get:

@app.route("/caduser")

def lista\_user():

    db = mysql.connector.connect(*host* = '201.23.3.86',

*port* = 5000,

*user* = 'usr\_aluno',

*password*= 'E$tud@\_m@1$',

*database*= 'aula\_fatec')

    mycursor = db.cursor()

    query = 'select nome, cpf, email from Hugo\_IIItbusuario'

    mycursor.execute(query)

    resultado = mycursor.fetchall()

    return render\_template('cadastro.html', *usuarios* = resultado)

**HeidiSQL**

Servidor/IP: 201.23.3.86

Usuário: usr\_aluno

Senha: E$tud@\_m@1$

Porta: 5000

Bancos de dados: aula\_fatec

**Para linkar o python com sql**

* Entrar na pasta Scripts, usar **.\activate** e **pip install mysql-connector-python** no terminal.

**Criando uma tabela:**

**CREATE** **TABLE** Hugo\_IIItbusuario (

nome **VARCHAR**(80),

CPF **CHAR**(14) PRIMARY key,

email **VARCHAR**(50),

senha **VARCHAR**(30)

)

--------------------------------------------------------------

**CREATE** **TABLE** Hugo\_IIItbusuario (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

nome **VARCHAR**(80),

CPF **CHAR**(14),

email **VARCHAR**(50),

senha **VARCHAR**(30)

)

--------------------------------------------------------------

**DESC** Hugo\_IIItbusuario (Mostra os atributos da tabela)

**( \* ) Significa selecione todas as tabelas (print)**

**SELECT** \* **FROM** Hugo\_IIItbusuario (Mostra tudo)

**SELECT** nome, CPF **FROM** Hugo\_IIItbusuario (Mostra só nome e cpf)

**SELECT** nome, CPF **FROM** Hugo\_IIItbusuario **WHERE** nome = 'Hugo' (Mostra só o ‘hugo’)

**Inserindo valores:**

**INSERT** **INTO** Hugo\_IIItbusuario (nome, CPF, email, senha)

**VALUES** ('Hugo III', '123.345.456-12', 'hugo@gmail.com', '123456')

**UPDATE** Hugo\_IIItbusuario **SET** nome = 'juse', CPF = '987.654.321-12'

**WHERE** nome = 'Maria'

**Deletando**:

**DELETE** **FROM** Hugo\_IIItbusuario **WHERE** nome = 'Hugo' (Deleta o “Hugo”)

**DROP** **TABLE** Hugo\_IIItbusuario (MATA TUDO)