Criando Página pelo Flask

1 Passo:

Abrir o Pycharm e criar um arquivo .py

Em seguida , Iremos inserir o seguinte código **: from flask import Flask**

Após fazer isso iremos dar Alt + f12 e irá abrir o terminal

Após abrir o terminal iremos instalar o Flask com o seguinte código **: pip install flask == 2.0.2**

Após o termino da instalação , iremos utilizar o seguinte código :

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

**@app.route(‘/inicio’)**

**def Olá():**

Após isso sera criar uma nova pasta chamada Templantes com um arquivo jogoteca.py e iremos criar um arquivo Html

Iremos inserir o seguinte código

<!DOCTYPE html>  
<html>  
 <head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Jogoteca</title>  
 </head>  
 <body>  
 <div class="container">  
 <div class="page-header">  
 <h1>{{ Titulo }}</h1>  
 </div>  
 <table class="table table-striped table-responsive table-bordered">  
 <thead class="thead-default">  
 <tr>  
 <th>Nome</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <td>God of War</td>  
 </tr>  
 <tr>  
 <td>Skyrim</td>  
 </tr>  
 <tr>  
 <td>Valorant</td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
 </body>

Iremos voltar na nosso arquivo principal e inserir a palavra:

from flask import Flask, render\_template

Iremos Acrescentar no nosso código abaixo **: return\_templante(‘lista.html’)**

@app.route('/Inicio')  
def Olá():  
 return render\_template('lista.html'

Se quisermos modificar algo no no html iremos usar {{ qualquer mensagem aqui }}

No caso acrescentamos titulo no nosso html que ira ser = Jogo

@app.route('/Inicio')  
def Olá():  
 return render\_template('lista.html', Titulo = 'Jogos')

E para testar se está tudo certo no Html precisamos sempre colocar no final do código: app.run()

app.run()

Para adicionar jogos ao nosso site , iremos fazer criar uma variável , com o nome: **lista = [‘nome do jogos’]**

lista = ['Tetris', 'Skyrim', 'Crash']

Para dar tudo certo , iremos ter que ir no lista.html e deixar o seguinte código igual da imagem abaixo:

{% for jogo in jogos %}  
 <tr>  
 <td>{{ jogo }}</td>  
 </tr>  
 {% endfor %}

Após isso iremos facilitar o preenchimento dos nossos jogos , iremos criar uma classe dar as suas funções:

class Jogo:  
 def \_\_init\_\_(self, nome , categoria , console):  
 self.nome = nome  
 self.categoria = categoria  
 self.console = console

E depois de criadas, iremos dar as funções delas:

jogo1 = Jogo('Tetris', 'Puzzle', 'Atari')  
jogo2 = Jogo ('God of War', 'Rack n Stash', 'PS2')  
  
lista = [jogo1, jogo2]

Após isso , iremos ter que modificar no html também iremos ter que criar três **th** e três **td**

<th>Nome</th>  
<th>categoria</th>  
<th>console</th>

<td>{{ jogo.nome }}</td>  
<td>{{ jogo.categoria }}</td>  
<td>{{ jogo.console }}</td>

Após isso , queremos colocar mais abas no nosso site , então iremos criar um novo arquivo html na paste template, copiar e cola o código:

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Jogoteca</title>  
</head>  
<body>  
 <div class="container">  
 <div class="page-header">  
 <h1>{{ titulo }}</h1>  
 </div>  
 <form>  
 <fieldset>  
 <div class="form-group">  
 <label for="nome">Nome</label>  
 <input type="text" id="nome" name="nome" class="form-control">  
 </div>  
 <div class="form-group">  
 <label for="categoria">Categoria</label>  
 <input type="text" id="categoria" name="categoria" class="form-control">  
 </div>  
 <div class="form-group">  
 <label for="console">Console</label>  
 <input type="text" id="console" name="console" class="form-control">  
 </div>  
 <button type="submit" class="btn btn-primary btn-salvar">Salvar</button>  
 </fieldset>  
 </form>  
 </div>  
</body>  
</html>

E iremos criar um novo def no nosso arquivo principal, seguindo os mesmos padrões do outro

@app.route('/Novo')  
def novo():  
 return render\_template('Novo.html', Titulo = 'Novo Jogo')

Após isso iremos no html e acrescentar um código no **<form action = “/criar” method = ...**

<h1>{{ titulo }}</h1>  
</div>  
<form action="/criar" method="post">

Iremos voltar para o nosso arquivo principal , e adicionar o **request** em nossa rota de importação

from flask import Flask, render\_template, request

E depois iremos criar a rota dele:

@app.route('/criar')  
def criar():  
 nome = request.form['nome']  
 categoria = request.form['categoria']  
 console = request.form ['console']  
 jogo = Jogo(nome, categoria, console)

Após esse processo , iremos utilizar o: **lista.append(jogo)** e embaixo iremos adicionar também : return render\_template**(‘lista.html’, títulos=’jogos’,jogos=lista)**

return render\_template('lista.html',titulos='Jogos',jogos=lista)

E iremos mover as informações que estava no **app.route inicio** , para **Class jogo: (Informações que estão em vermelho , são as que transferimos para a class Jogo:**

class Jogo:  
 def \_\_init\_\_(self, nome , categoria , console):  
 self.nome = nome  
 self.categoria = categoria  
 self.console = console  
jogo1 = Jogo('Tetris', 'Puzzle', 'Atari')  
jogo2 = Jogo('God of War', 'Rack n Stash', 'PS2')  
lista = [jogo1, jogo2]

E só colocar para rodar a 1 parte do nosso projeto 😊

2 Passo

Aaaa , faltou mais um código bonitininho... Nós iremos adicionar **, Methods=[‘POST’,]**

@app.route('/criar',methods=['POST',])  
def criar():

Para não ficar salvando toda hora , dando crtl + s , iremos acrescentar algo no código do app.run() **, app.run(debug=True)**

app.run(debug=True)

Após isso , iremos modificar 1 codigo por outro no pycharm

Código antigo: return render\_template('lista.html',titulos='Jogos',jogos=lista)

Código novo:

return redirect

E iremos adicionar redirect no import

from flask import Flask, render\_template, request, redirect

E por ultimo iremos adicionar :

return redirect('/')

**Depois disso , iremos no site do bootstratp , baixar a versão 5.1 e baixar o compiled css and js**

**Após o download iremos abrir a pasta e copiar o arquivo bootstrap.css**

**Depois iremos no nosso pycharm , criar uma pasta chamada : static e clicar com o botão direto sobre a pasta e clicar na opção : Paste**

**Após isso , iremos dar ok e está criado nosso arquivo Css**

**Após feito isso , iremos no nosso arquivo html e iremos criar o seguinte código embaixo de <tittle>**

<title>Jogoteca</title>  
 <link rel="stylesheet" href="../static/bootstrap.css">

**E iremos fazer isso também , no nosso outro html**

<title>Jogoteca</title>  
<link rel="stylesheet" href="../static/bootstrap.css">

**Iremos criar um novo arquivo html com o nome template.html e iremos adicionar esse código:**

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Jogoteca</title>  
 <link rel="stylesheet" href="../static/bootstrap.css">  
</head>  
<body>  
 <div class="container">  
 <div class="page-header">  
 <h1>{{ titulo }}</h1>  
 </div>  
 {% block conteudo %} {% endblock %}  
 </div>  
 </body>  
</html>

**Após isso iremos nos outros dois html e adicionar a seguinte função**

{% extends "template.html" %}  
{% block conteudo %}  
 <table class="table table-striped table-responsive table-bordered">  
 <thead class="thead-default">  
 <tr>  
 <th>Nome</th>  
 <th>categoria</th>  
 <th>console</th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for jogo in jogos %}  
 <tr>  
 <td>{{ jogo.nome }}</td>  
 <td>{{ jogo.categoria }}</td>  
 <td>{{ jogo.console }}</td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
{% endblock %}

{% extends "template.html" %}  
{% block conteudo %}  
 <form action="/criar" method="post">  
 <fieldset>  
 <div class="form-group">  
 <label for="nome">Nome</label>  
 <input type="text" id="nome" name="nome" class="form-control">  
 </div>  
 <div class="form-group">  
 <label for="categoria">Categoria</label>  
 <input type="text" id="categoria" name="categoria" class="form-control">  
 </div>  
 <div class="form-group">  
 <label for="console">Console</label>  
 <input type="text" id="console" name="console" class="form-control">  
 </div>  
 <button type="submit" class="btn btn-primary btn-salvar">Salvar</button>  
 </fieldset>  
 </form>  
{% endblock %}

Agora para ficar mais fácil , iremos modificar o caminho do bootsratp no template:

<meta charset="utf-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Jogoteca</title>  
<link rel="stylesheet" href="{{ url\_for('static', filename='bootstrap.css') }}">

Esse vai ser o novo caminho ( print acima)

**3 -Como criar um ambiente de login e Senha:**

Primeiro iremos criar um ambiente de html na templates com o nome **login.html**

Depois iremos utilizar o seguinte código:

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name = "viewport" content="width=device-width, initial-scale=1">  
 <title>Jogoteca</title>  
 <link rel ="stylesheet" href="{{ url\_for('static', filename='bootstrap.css') }}">  
</head>  
<body>  
<div class="container">  
 <h1>Faça seu login</h1>  
 <form method="POST" action="/autenticar">  
 <p><label>Nome do Usuario:</label><input class="form-control" type="text" name="usuario" required></p>  
 <p><label>Senha:</label><input class="form-control" type="password" name="senha" required></p>  
 <p><button class="btn btn-primary" type="submit">Entrar</button> </p>  
 </form>  
</div>  
</body>  
</html>

Depois disso , iremos voltar para o menu principal e criar a rota dele:

Assim ira ficar a rota do login:

@app.route('/login')  
def login():  
 return render\_template('login.html')  
  
@app.route('/autenticar', methods=['POST',])  
def autenticar():  
 if 'alohomora' == request.form['senha']:  
 return redirect('/')  
 else:  
 return redirect('/login')

**4 – Guardando Dados na sessão**

Iremos primeiro importa o dado **session**

from flask import Flask, render\_template, request, redirect, session

Em seguida iremos adicionar os códigos abaixo em **azul**:

@app.route('/autenticar', methods=['POST',])  
def autenticar():  
 if 'alohomora' == request.form['senha']:  
 session['usuario\_logado'] = request.form['usuario']  
 flash(session['usuario\_logado'] + 'logado com sucesso!')  
 return redirect('/')  
 else:  
 flash('Usuario não logado!')  
 return redirect('/login')

E depois iremos importa o **flash**:

from flask import Flask, render\_template, request, redirect, session, flash

Em seguida iremos adicionar uma chave de segurança para o nosso site

app = Flask(\_\_name\_\_)  
app.secret\_key = 'bubu'

Agora iremos nas paginas **html** e iremos adicionar na pagina **login** e na pagina templates o seguinte código

{% with messages = get\_flashed\_messages() %}  
 {% if messages %}  
 <ul id="messages" class="list-unstyled">  
 {% for message in messages %}  
 <li class="alert alert-sucess">{{ message }}</li>  
 {%endfor %}  
 </ul>  
 {%endif %}  
 {% endwith %}

5 tema **-> Como Deslogar do sistema**

Iremos por enquanto apenas criar a rota de deslogar

@app.route('/logout')  
def logout():  
 session['usuario\_logado'] = None  
 flash('Logout efetuado com sucesso!')  
 return redirect('/')

**6 – Como usar o Programa , só estando logado**

@app.route('/Novo')  
def novo():  
 if 'usuario\_logado' not in session or session['usuario\_logado'] == None:  
 return redirect('/login')  
 return render\_template('novo.html', titulo='Novo Jogo')

**07 – Melhorando o fluxo de login**

**Iremos acrescentar**

@app.route('/login')  
def login():  
 proxima = request.args.get('proxima')  
 return render\_template('login.html', proxima=proxima)

**E iremos criar na pagina login.html**

<input type="hidden" name="proxima" value="{{ proxima }}">