



INTRODUÇÃO AO FLASK



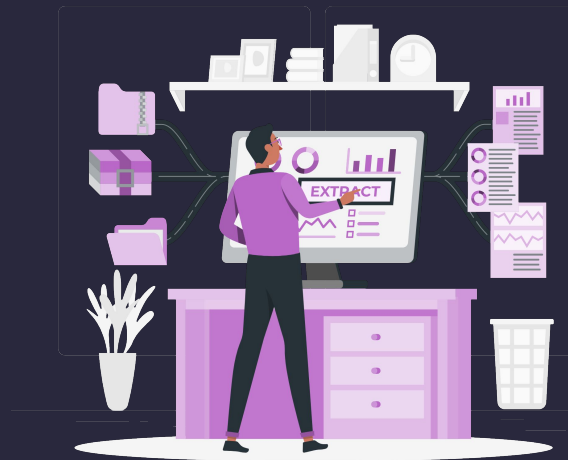
[Framework]



@mrafaelbatista



messiasbatista





01

FLASK

O QUE É UM FRAMEWORK?



FLASK

Flask é um **microframework** web escrito em Python.

Ele é classificado como "micro" porque não requer ferramentas ou bibliotecas particulares, permitindo aos desenvolvedores adicionar apenas o que precisam.



O QUE É UM FRAMEWORK?

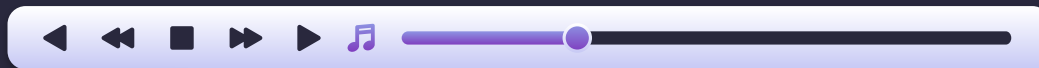


@mrafaelbatista



messiasbatista

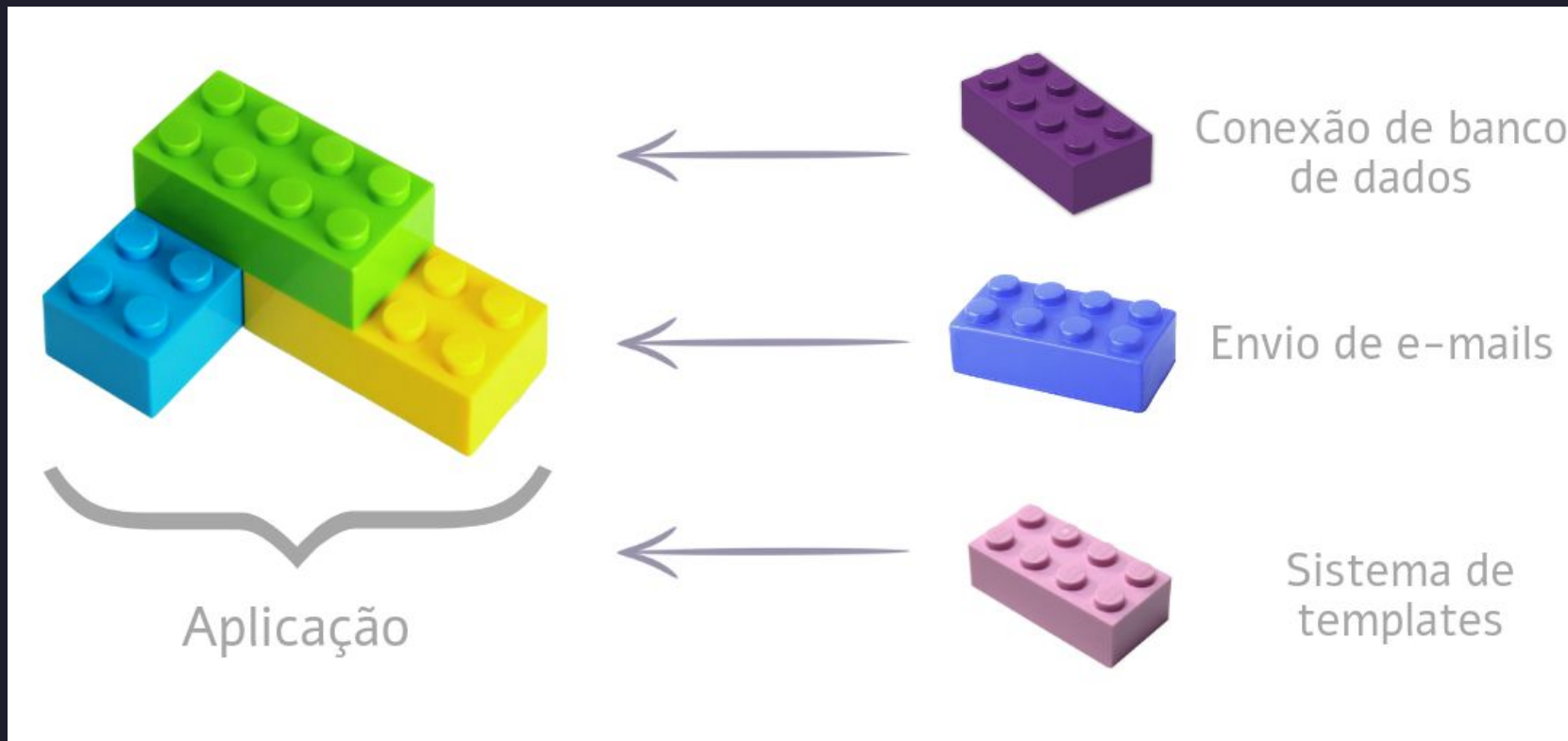
www.mrafaelbatista.dev



Um framework é uma estrutura ou conjunto de ferramentas pré-escritas que oferece uma fundação para desenvolver e organizar projetos de software.

FRAMEWORK

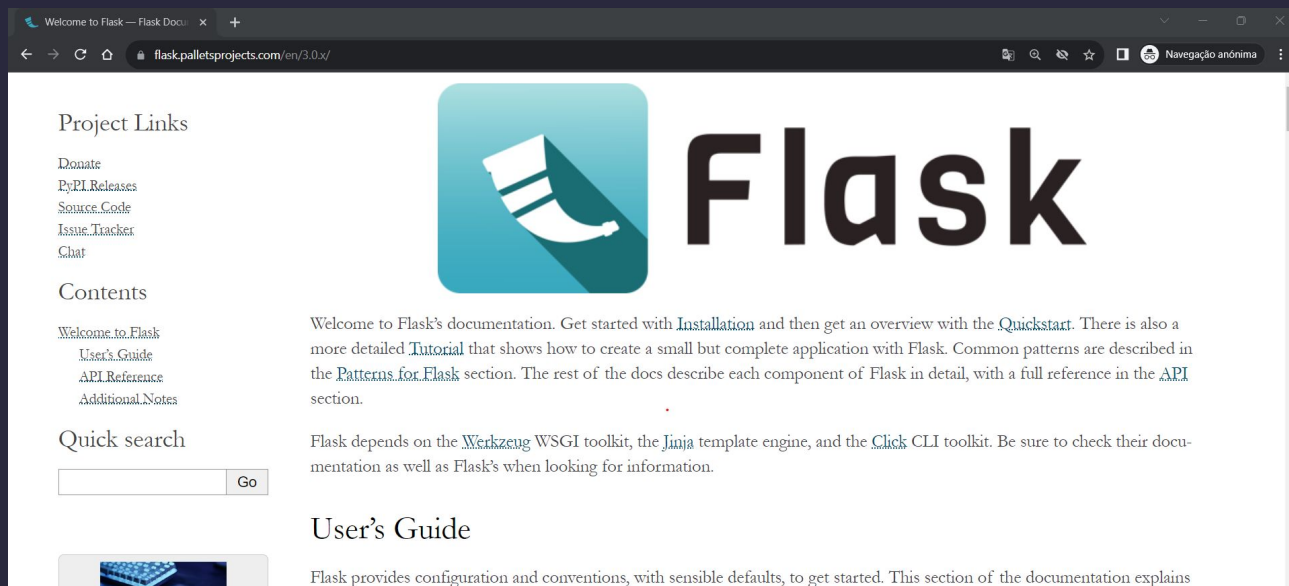
Fonte da imagem





FLASK

<https://flask.palletsprojects.com/>

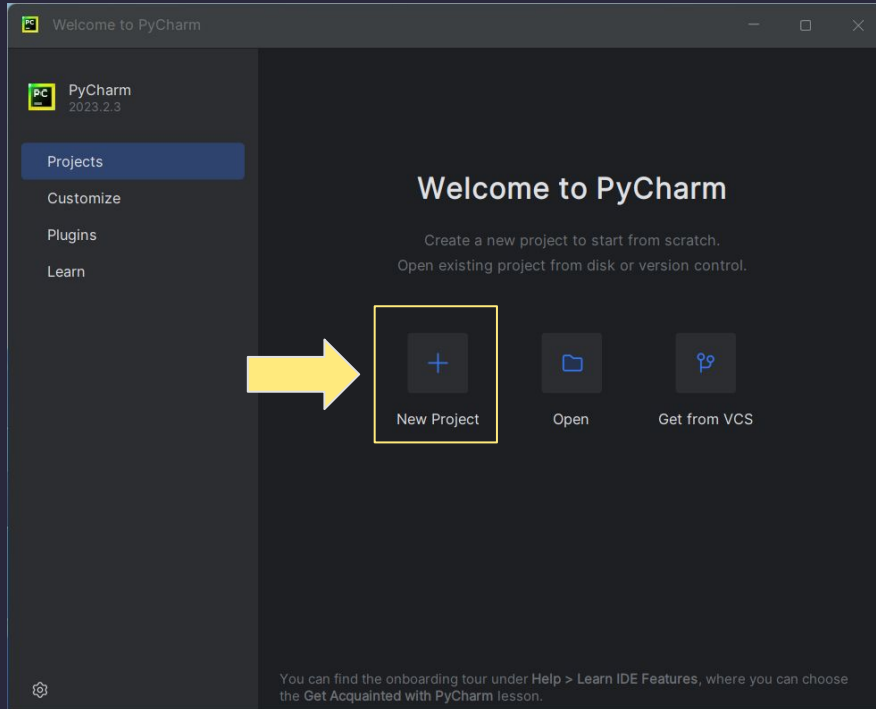


02 INSTALAÇÃO E CONFIGURAÇÃO DO FLASK

LINHA DE COMANDO

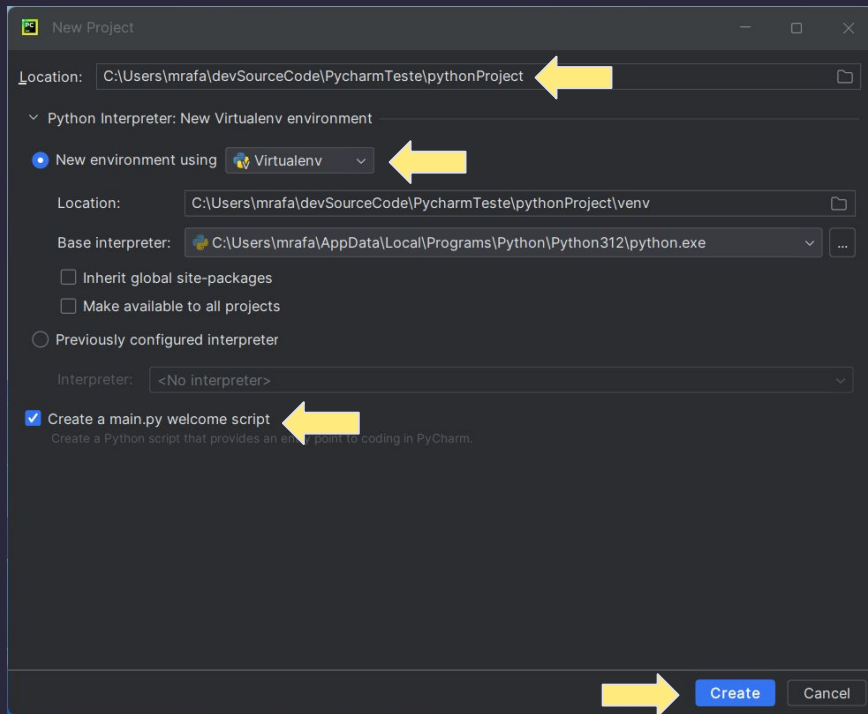


INICIE UM PROJETO NO PYCHARM



1. Inicie o Pycharm
2. Clique em New Project

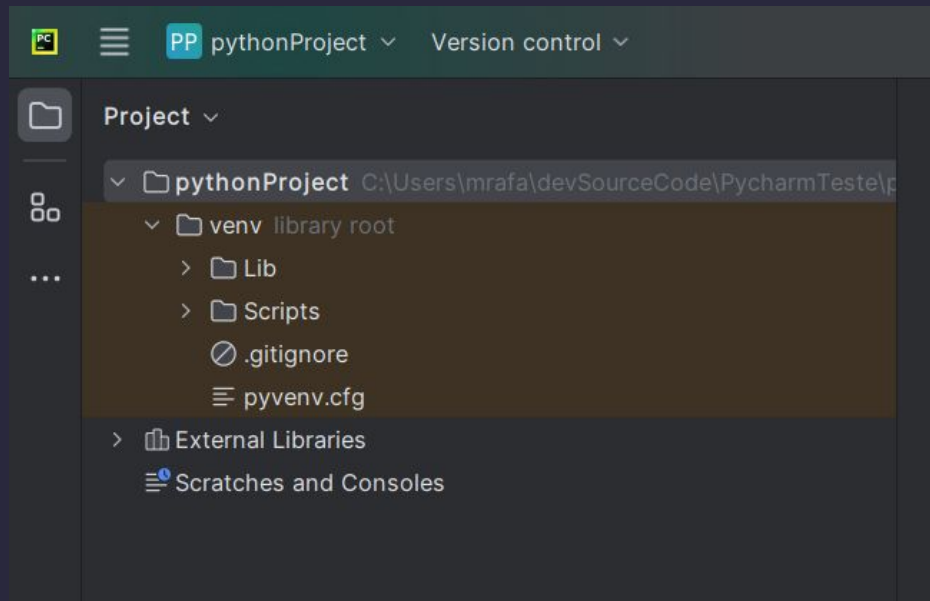
CONFIGURE UM PROJETO NO PYCHARM



1. **Location:** selecione uma pasta vazia para salvar o projeto;
2. **New enviroment using:** deixe marcado para criar uma nova variável de ambiente;
3. Desmarque **Create a main.py welcome script**;
4. Clique em **Create**.

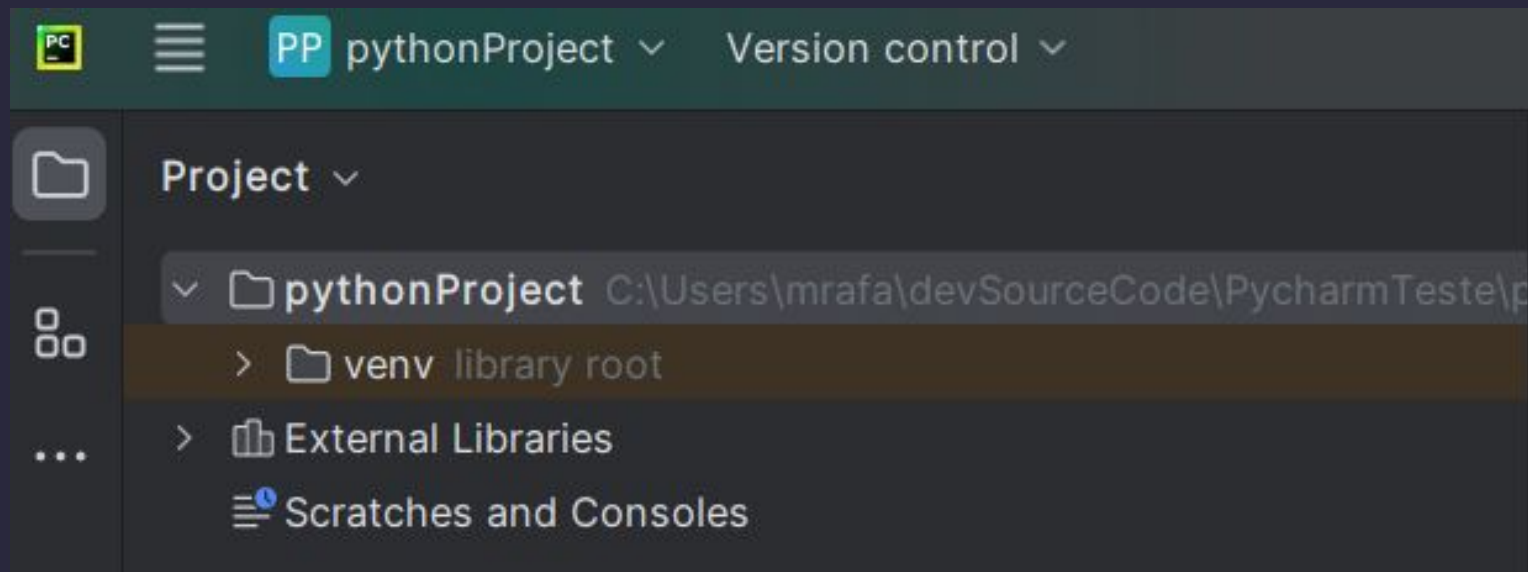


ESTRUTURA DE UM PROJETO NO PYCHARM




- Verifique a estrutura do seu projeto;
- `pythonProject` é o nome do projeto que você colocou.
- Venv é uma pasta que pode ser minimizada;

ESTRUTURA DE UM PROJETO NO PYCHARM



INSTALANDO O FLASK NO PROJETO DO PYCHARM

1. Abra o terminal dentro do Pycharm:

- Atalho para Windows (Alt+F12)
- Ou, acessando o ícone 

2. Atualize o pip da venv


- `python -m pip install --upgrade pip`

3. Instale o Flask

- `Pip install Flask`

INSTALANDO O FLASK NO PROJETO DO PYCHARM

1. Abra o terminal dentro do Pycharm:

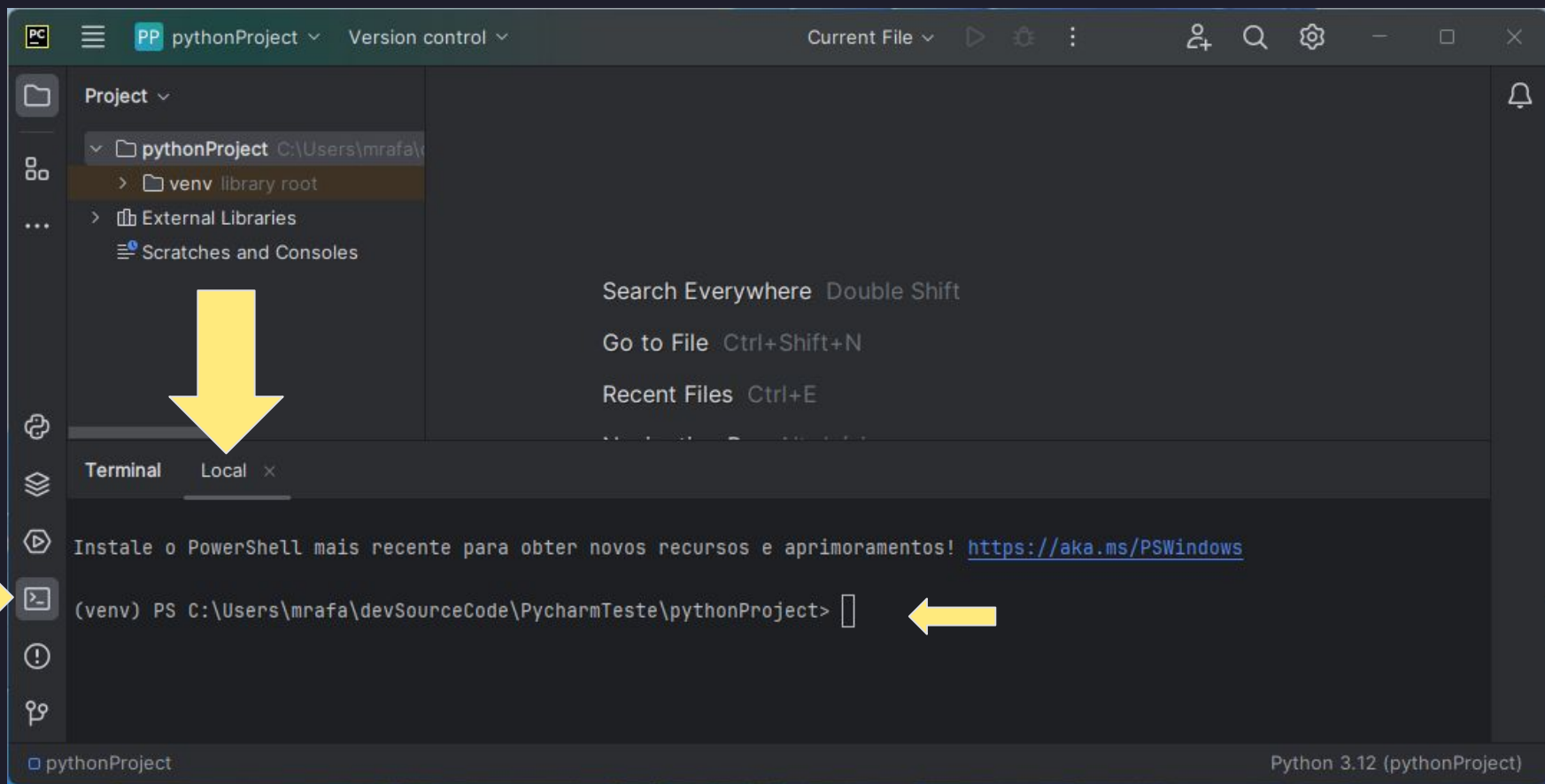
- Atalho para Windows (Alt+F12)
- Ou, acessando o ícone 

2. Atualize o pip da venv

- `python -m pip install --upgrade pip`


3. Instale o Flask

- `Pip install Flask`



INSTALANDO O FLASK NO PROJETO DO PYCHARM

1. Abra o terminal dentro do Pycharm:

- Atalho para Windows (Alt+F12)
- Ou, acessando o ícone 

2. Atualize o pip da venv

- `python -m pip install --upgrade pip`

3. Instale o Flask

- `Pip install Flask`



Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! <https://aka.ms/PSWindows>



(venv) PS C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject> python -m pip install --upgrade pip



python -m pip install --upgrade pip

Terminal Local x + v


Collecting pip
Obtaining dependency information for pip from <https://files.pythonhosted.org/packages/47/6a/453160888fab7c6a432a6e25f8afe6256d0d9f2cbd25971021da6491d899/pip-23.3.1-py3-none-any.whl.metadata>
Using cached pip-23.3.1-py3-none-any.whl.metadata (3.5 kB)
Using cached pip-23.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
Attempting uninstall: pip
Found existing installation: pip 23.2.1
Uninstalling pip-23.2.1:
Successfully uninstalled pip-23.2.1
Successfully installed pip-23.3.1
(venv) PS C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject>

Pip atualizado!



INSTALANDO O FLASK NO PROJETO DO PYCHARM

1. Abra o terminal dentro do Pycharm:

- Atalho para Windows (Alt+F12)
- Ou, acessando o ícone 

2. Atualize o pip da venv

- `python -m pip install --upgrade pip`

3. Instale o Flask

- `pip install Flask`



```
Terminal Local ×  
(venv) PS C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject> pip install Flask
```

pip install Flask



```
Terminal Local × + v
(venv) PS C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject> pip install Flask
Collecting Flask
  Downloading flask-3.0.0-py3-none-any.whl.metadata (3.6 kB)
Collecting Werkzeug>=3.0.0 (from Flask)
  Downloading werkzeug-3.0.0-py3-none-any.whl.metadata (4.1 kB)
Collecting Jinja2>=3.1.2 (from Flask)
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 kB 2.0 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2 (from Flask)
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3 (from Flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from Flask)
  Downloading blinker-1.6.3-py3-none-any.whl.metadata (1.9 kB)
Collecting colorama (from click>=8.1.3->Flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->Flask)
  Downloading MarkupSafe-2.1.3-cp312-cp312-win_amd64.whl.metadata (3.0 kB)
Downloading flask-3.0.0-py3-none-any.whl (99 kB)
    99.7/99.7 kB ? eta 0:00:00
Downloading blinker-1.6.3-py3-none-any.whl (13 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
    97.9/97.9 kB ? eta 0:00:00
Downloading werkzeug-3.0.0-py3-none-any.whl (226 kB)
    226.6/226.6 kB ? eta 0:00:00
Downloading MarkupSafe-2.1.3-cp312-cp312-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, Flask
Successfully installed Flask-3.0.0 Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.0 blinker-1.6.3 click-8.1.7 colorama-0.4.6 itsdangerous-2.1.2
(venv) PS C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject> 
```



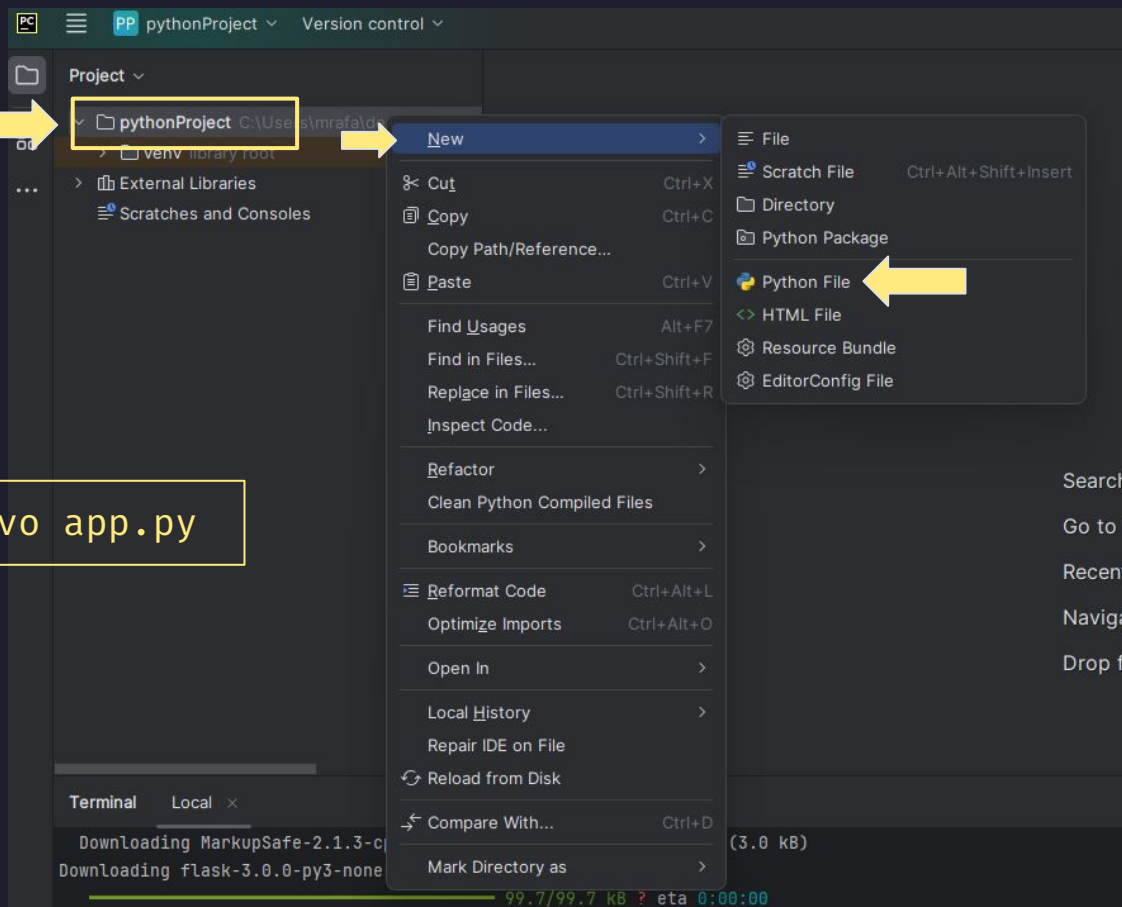
Flask
instalado!



CRIAÇÃO DO ARQUIVO APP.PY

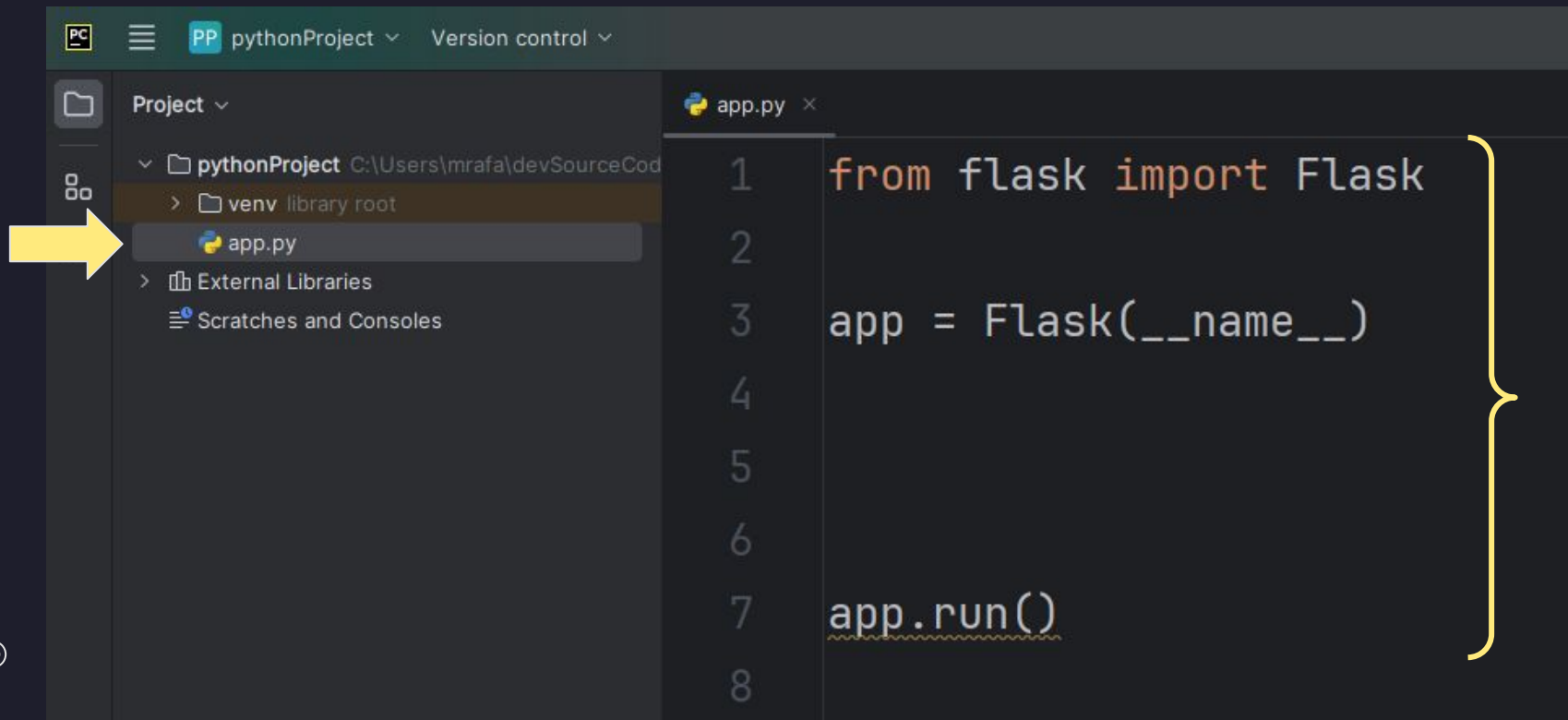
1. No diretório do projeto crie o arquivo app.py com o seguinte código:

```
from flask import Flask  
  
app = Flask(__name__)  
  
app.run()
```



Crie o arquivo app.py







DÚVIDAS?



@mrafaelbatista

messiasbatista

03

HELLO WORLD NO FLASK

O QUE É UM FRAMEWORK?



ALTERANDO O ARQUIVO APP.PY

```
from flask import Flask
```



Importação da biblioteca do Flask

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def ola():
```

```
    return '<h1>Olá, mundo!</h1>'
```

```
app.run()
```

ALTERANDO O ARQUIVO APP.PY

```
from flask import Flask
```

```
app = Flask(__name__)
```



Iniciando o Flask dentro da variável app

```
@app.route('/')
```

```
def ola():
```

```
    return '<h1>Olá, mundo!</h1>'
```

```
app.run()
```

ALTERANDO O ARQUIVO APP.PY

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def ola():
    return '<h1>Olá, mundo!</h1>'

app.run()
```



Criando uma rota no site.

A função `ola()` define o que será realizado ao acessar esta rota.

ALTERANDO O ARQUIVO APP.PY

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def ola():
    return '<h1>Olá, mundo!</h1>'

app.run()
```



```
app.run()
```

Executa o servidor em que o Flask
será executado.



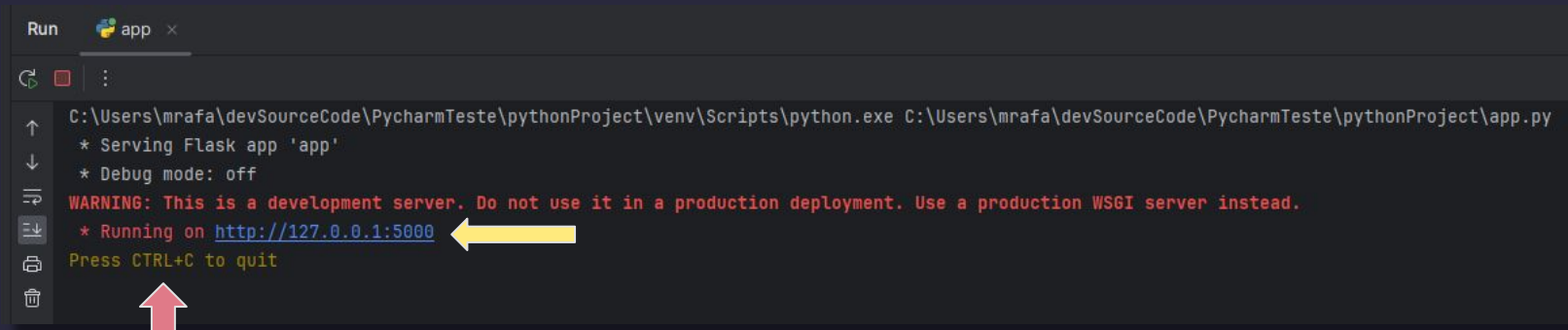
EXECUTE O ARQUIVO APP.PY



ACESSANDO O SITE

O servidor é executado em <http://127.0.0.1:5000>

Caso precise **parar** o servidor, aperte CTRL+C

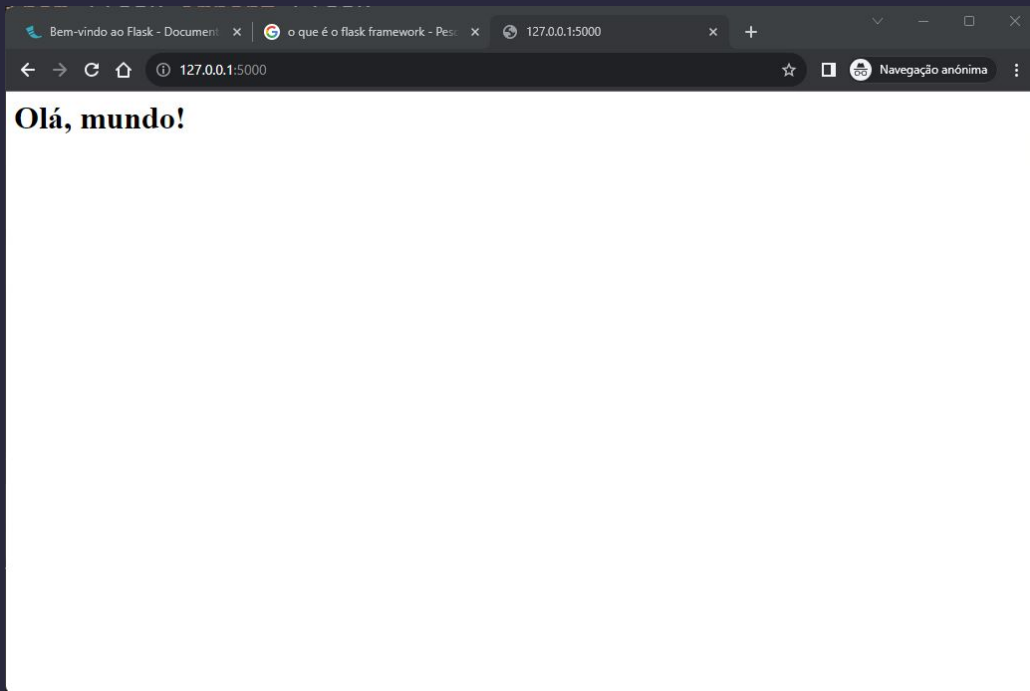


```
Run  app x
C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject\venv\Scripts\python.exe C:\Users\mrafa\devSourceCode\PycharmTeste\pythonProject\app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



ACESSANDO O OLÁ, MUNDO!

Ao acessar o endereço
<http://127.0.0.1:5000>,
espera-se que você
visualize esta página
com a mensagem, **Olá,
mundo!**



04

TEMPLATES NO FLASK

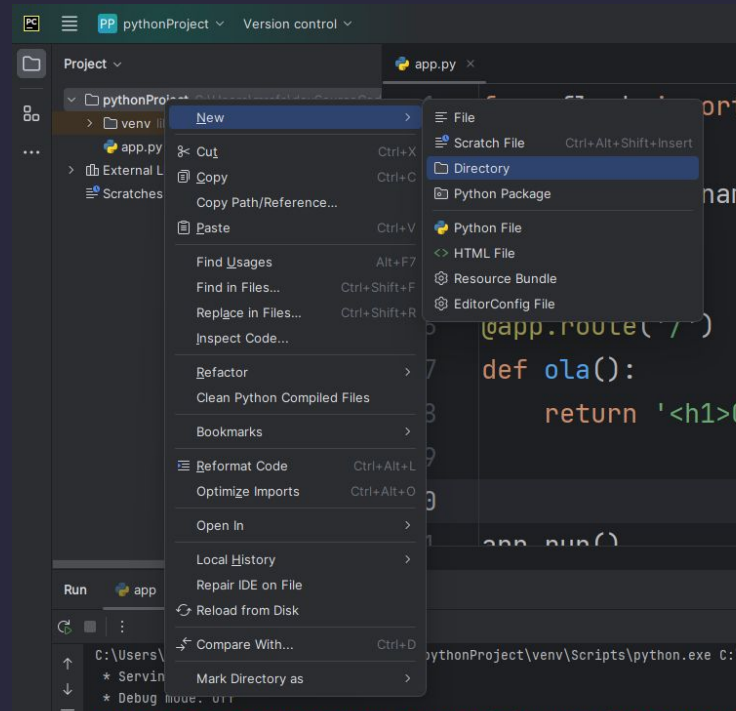
MARAVILHA!!!

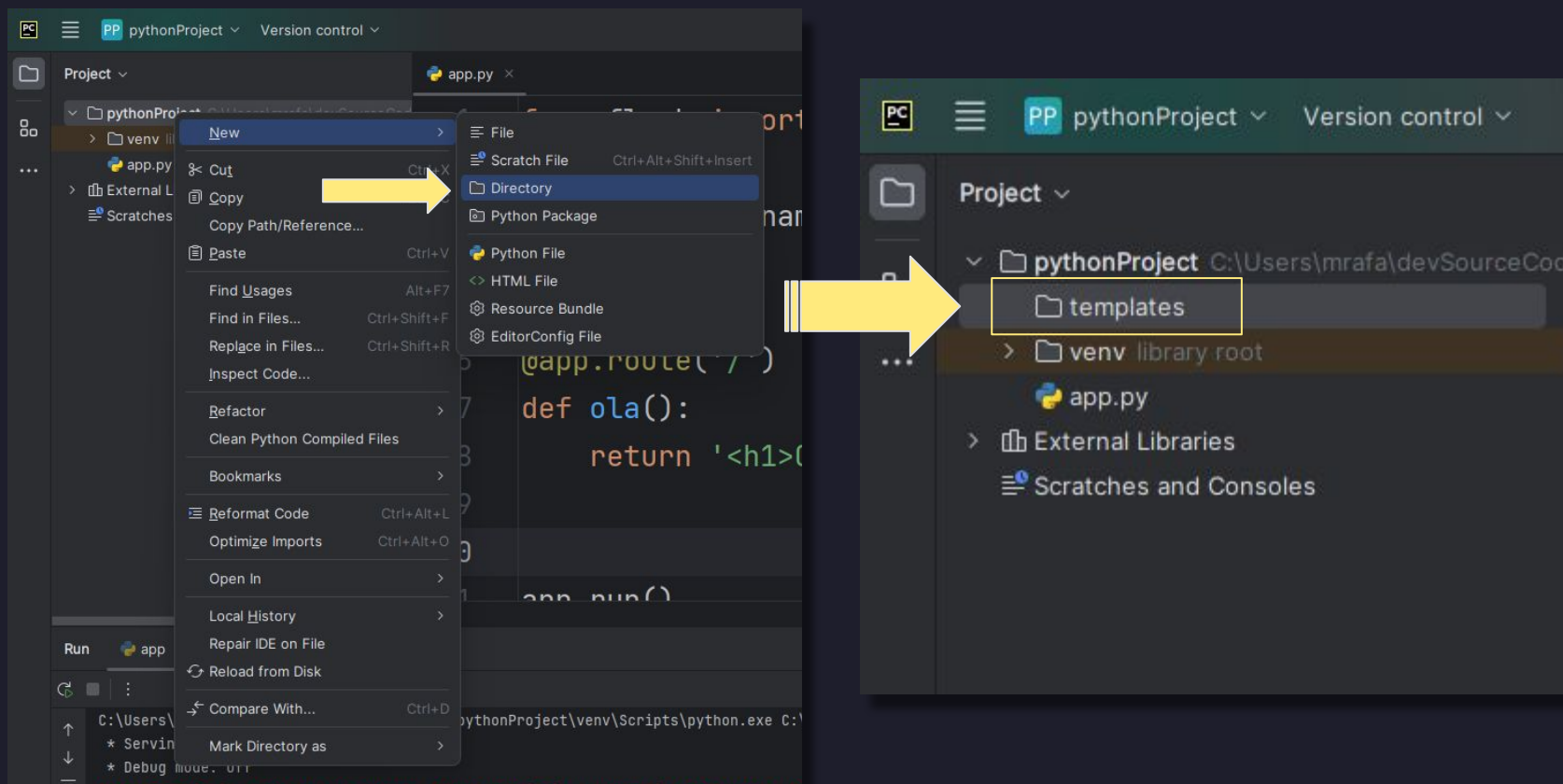


PASTA TEMPLATES

Os templates no Flask são organizados na pasta templates.

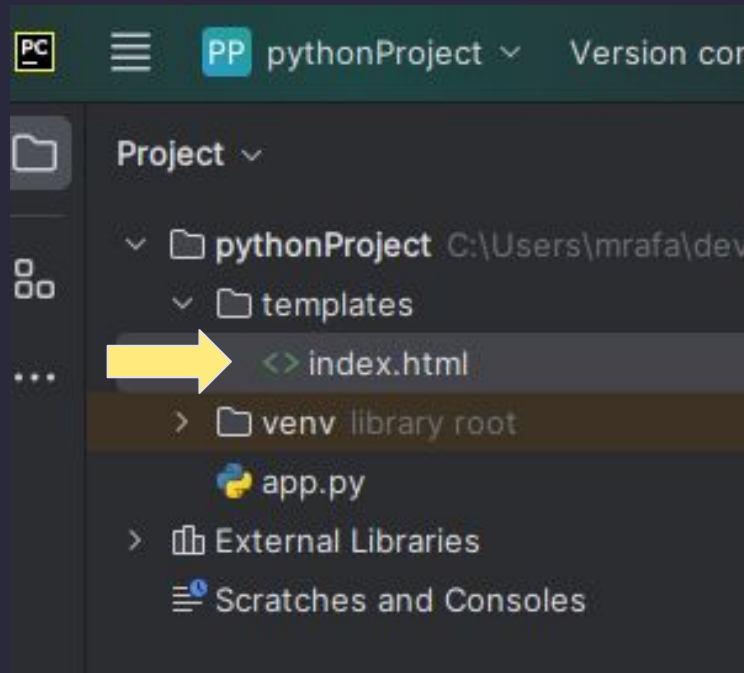
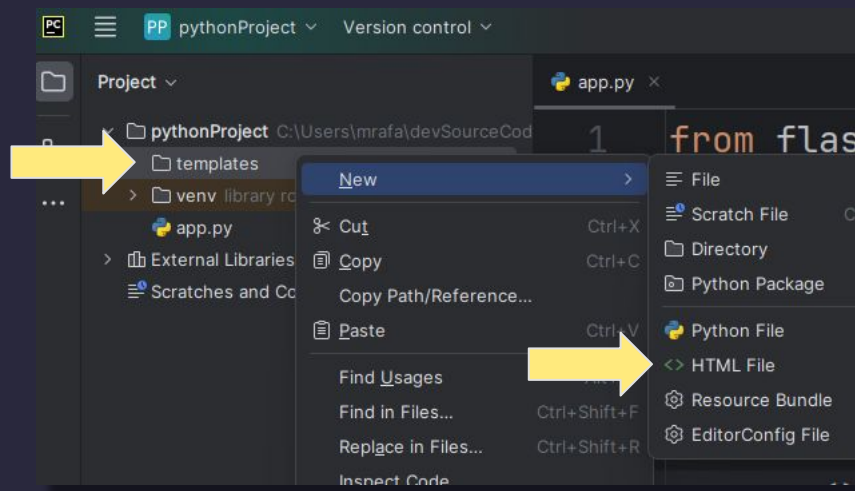
Crie a pasta templates no seu diretório principal.





CRIANDO MINHA PÁGINA PRINCIPAL

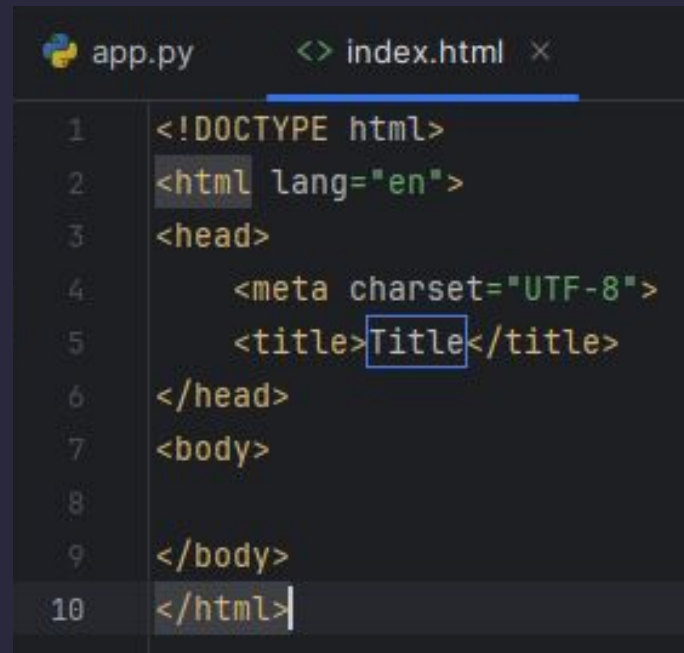
Na pasta templates crie o arquivo **index.html**



CRIANDO MINHA PÁGINA PRINCIPAL

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>

</body>
</html>
```



```
app.py  index.html ×
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8
9  </body>
10 </html>|
```

[ATIVIDADE] PÁGINA PRINCIPAL

Desenvolva a página principal para nosso site.

Contexto:

Site para organização dos estudos com organizador de tarefas e glossário de termos técnicos.



USE SUA CRIATIVIDADE!



@mrafaelbatista



messiasbatista

05

RENDERIZAR A PÁGINA PRINCIPAL

MUITO ARRETADO!



RENDER TEMPLATE

O **render_template** é uma função do Flask que permite renderizar um template HTML.

1. Devemos importar o `render_template`
2. Retornar o `render template` com nosso arquivo `html`;



app.py

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def ola():
    # return '<h1>Olá, mundo!</h1>'
    return render_template('index.html')

app.run()
```



app.py

Agora você sabe como
criar rotas e
templates html.

Aprimore seu site!

E compartilhe seus
avanços...



```
app.py x <> index.html
1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5
6  @app.route('/')
7  def ola():
8      # return '<h1>Olá,  mundo! </u></h1>'
9      return render_template('index.html')
10
11
12  app.run()
```

INTRODUÇÃO AO FLASK

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

