

# Aula 05:

Construção de uma home -page

# Instalação e execução:

```
pip install uvicorn
```

```
pip install jinja2
```

```
pip install aiofiles
```

Depois de executar o script python executar o comando

```
uvicorn fastapi_exemplo_1:app --reload
```

# Construção da HomePage:

```
from fastapi import FastAPI
from starlette.requests import Request
from starlette.staticfiles import StaticFiles
from starlette.templating import Jinja2Templates
app = FastAPI()
app.mount("/static", StaticFiles(directory="static"), name="static")
templates = Jinja2Templates(directory="templates")
@app.get("/items/{RG}")
async def read_item(request: Request, RG: str):
    return templates.TemplateResponse("item.html", {"request": request,
"RG": RG})
```

# Script:

```
fastapi1.py X  item.html  # style1.css  # style.css ... \css  # style.css static
fastapi1.py > read_item
1  from fastapi import FastAPI
2  from starlette.requests import Request
3  from starlette.staticfiles import StaticFiles
4  from starlette.templating import Jinja2Templates
5
6  app = FastAPI()
7
8  app.mount("/static", StaticFiles(directory="static"), name="static")
9
10
11  templates = Jinja2Templates(directory="templates")
12
13
14  @app.get("/items/{RG}")
15
16  async def read_item(request: Request, RG: str):
17      return templates.TemplateResponse("item.html", {"request": request, "RG": RG})
```

# Diretórios:

- Criar um diretório chamado templates e um arquivo chamado item.html
- Criar um diretório chamado static e um arquivo chamado style.css

# Template padrão html:

```
<html>
```

```
<head>
```

```
  <title>RG</title>
```

```
  <link href="{{ url_for('static', path='/style.css') }}" rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
  <h1>O RG de Maria é: {{ RG }}</h1>
```

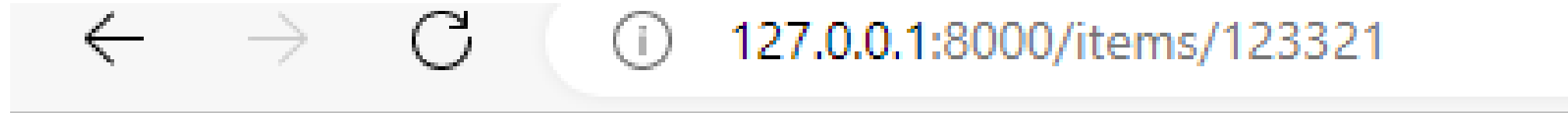
```
</body>
```

```
</html>
```

# Template padrão css:

```
h1 {  
    color: green;  
}
```

Home-Page:



**O RG de Maria é: 123321**



# Melhorando a home-page:

```
h1 {  
    border: 5px solid red;  
    color: black;  
    background-color: deepskyblue;  
    text-align: center;  
    height: 50px;  
    width: 500px;  
    margin: left;}
```



127.0.0.1:8000/items/123321

**O RG de Maria é: 123321**

# Vamos além!

- Enviar um valor de um formulário.

# Script Python:

```
from fastapi import FastAPI, Request, Form
from fastapi.templating import Jinja2Templates
app = FastAPI()
templates = Jinja2Templates(directory="templates")

@app.get('/')
def read_form():
    return 'Olá galera. Estou construindo home-page em FastAPI!'
@app.get("/form")
def form_post(request: Request):
    result = "0"
    return templates.TemplateResponse('form.html', context={'request': request, 'result': result})

@app.post("/form")
def form_post(request: Request, num: int = Form(...)):
    result = (num)
    return templates.TemplateResponse('form.html', context={'request': request, 'result': result})
```

fastapi1.py

fastapi2.py X

form.html

item.html

# style.css ...css

# style.css static



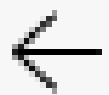
fastapi2.py > form\_post

```
1  from fastapi import FastAPI, Request, Form
2  from fastapi.templating import Jinja2Templates
3
4  app = FastAPI()
5  templates = Jinja2Templates(directory="templates")
6
7
8  @app.get('/')
9  def read_form():
10     return 'Olá galera. Estou construindo home-page em FastAPI!'
11
12  @app.get("/form")
13  def form_post(request: Request):
14     result = "0"
15     return templates.TemplateResponse('form.html', context={'request': request, 'result': result})
16
17
18  @app.post("/form")
19  def form_post(request: Request, num: int = Form(...)):
20     result = (num)
21     return templates.TemplateResponse('form.html', context={'request': request, 'result': result})
```

13 THE FASTAPI-TEMPLATING  
14 Jinja2Templates(directory="templates")  
15 Jinja2Templates(directory="templates")  
16 Jinja2Templates(directory="templates")  
17 Jinja2Templates(directory="templates")  
18 Jinja2Templates(directory="templates")  
19 Jinja2Templates(directory="templates")  
20 Jinja2Templates(directory="templates")  
21 Jinja2Templates(directory="templates")

# Template html:

```
<html>
<head>
  <title>Formulário</title>
</head>
<body>
  <form method="post">
    Enviando um valor: <input type="number" name="num" value="submit">
    <input type="submit">
  </form>
  <p>Resultado: {{ result }}</p>
</body>
</html>
```



127.0.0.1:8000/form

Enviando um valor:

Resultado: 0

# Home Page para o cálculo do IMC:

```
from fastapi import FastAPI, Request, Form
from fastapi.templating import Jinja2Templates

app = FastAPI()
templates = Jinja2Templates(directory="templates")

@app.get('/')
def read_form():
    return 'Olá galera. Home-page para calcular o IMC em FastAPI!'

@app.get("/form")
def form_post(request: Request):
    result = "0"
    return templates.TemplateResponse('form3.html', context={'request': request, 'result': result})

@app.post("/form")
def form_post(request: Request, peso: int = Form(...), altura: int = Form(...)):
    result = round((peso/((altura/100)**2)),1)
    return templates.TemplateResponse('form3.html', context={'request': request, 'result': result})
```



# Script python:

```
fastapi1.py fastapi2.py fastapi3.py X form.html form3.html
fastapi3.py > form_post
1  from fastapi import FastAPI, Request, Form
2  from fastapi.templating import Jinja2Templates
3
4  app = FastAPI()
5  templates = Jinja2Templates(directory="templates")
6
7
8  @app.get('/')
9  def read_form():
10     return 'Olá galera. Home-page para calcular o IMC em FastAPI!'
11
12  @app.get("/form")
13  def form_post(request: Request):
14     result = "0"
15     return templates.TemplateResponse('form3.html', context={'request': request, 'result': result})
16
17
18  @app.post("/form")
19  def form_post(request: Request, peso: int = Form(...), altura: int = Form(...)):
20     result = round((peso/((altura/100)**2)),1)
21     return templates.TemplateResponse('form3.html', context={'request': request, 'result': result})
```

# Template:

```
<html>
<head>

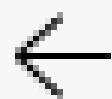
  <title>Formulário</title>
</head>
<body>
<form method="post">
  <p>Enviando o peso (kg): <input type="number" name="peso" value="submit"></p>
  <p>Enviando altura (cm): <input type="number" name="altura" value="submit"></p>
  <p><input type="submit"></p>
</form>

<p>Resultado:imc={{ result }}</p>

</body>
</html>
```

# Template:

```
fastapi1.py fastapi2.py fastapi3.py form.html form3.html X
templates > form3.html > html
1  <html>
2  <head>
3  |
4  |     <title>Formulário</title>
5  </head>
6  <body>
7  <form method="post">
8  |     <p>Enviando o peso (kg): <input type="number" name="peso" value="submit"></p>
9  |     <p>Enviando altura (cm): <input type="number" name="altura" value="submit"></p>
10 |     <p><input type="submit"></p>
11 </form>
12
13 <p>Resultado:imc={{ result }}</p>
14
15 </body>
16 </html>
```



127.0.0.1:8000/form

Enviando o peso (kg):

Enviando altura (cm):

Enviar

Resultado:imc=24.5

# Usando um condicional no template (jinja2):

```
<html>
<head>
  <title>Formulário IMC</title>
</head>
<body>
<form method="post">
  <p>Enviando o peso (kg): <input type="number" name="peso" value="submit"></p>
  <p>Enviando altura (cm): <input type="number" name="altura" value="submit"></p>
  <p><input type="submit"></p>
</form>
```

# Continuação do template:

```
<p>Resultado: Seu imc é:{{ imc }}</p>
<p>Sua faixa de classificação do IMC é:</p>
{% if imc < 18.5 %}
<p> Magreza </p>
{% elif 18.5 < imc <= 24.9 %}
<p> Saudável </p>
{% elif 24.9 < imc <= 29.9 %}
<p> Sobrepeso </p>
{% else %}
<p> Obesidade </p>
{% endif %}
</body>
</html>
```

# Template:

```
fastapi1.py fastapi2.py fastapi3.py fastapi4.py form.html form4.html X
templates > form4.html > html > body > ? > ?
1  <html>
2  <head>
3  |
4  |   <title>Formulário</title>
5  | </head>
6  <body>
7  <form method="post">
8  |   <p>Enviando o peso (kg): <input type="number" name="peso" value="submit"></p>
9  |   <p>Enviando altura (cm): <input type="number" name="altura" value="submit"></p>
10 |   <p><input type="submit"></p>
11 </form>
12
13 <p>Resultado: Seu imc é:{{ imc }}</p>
14
15 <p>Sua faixa de classificação do IMC é:</p>
16 {% if imc < 18.5 %}
17 <p> Magreza </p>
18 {% elif 18.5 < imc <= 24.9 %}
19 <p> Saudável </p>
20 {% elif 24.9 < imc <= 29.9 %}
21 <p> Sobrepeso </p>
22 {% else %}
23 <p> Obesidade </p>
24 {% endif %}
25
26 </body>
27 </html>
```

# Script python:

```
from fastapi import FastAPI, Request, Form
from fastapi.templating import Jinja2Templates
app = FastAPI()
templates = Jinja2Templates(directory="templates")
@app.get('/')
def read_form():
    return 'Olá galera. Home-page para calcular o IMC em FastAPI!'
```



# Continuação do Script:

```
@app.get("/form")
def form_post(request: Request):
    imc = "0"
    return templates.TemplateResponse('form3.html', context={'request':
request, 'imc': imc})

@app.post("/form")
def form_post(request: Request, peso: int = Form(...), altura: int = Form(...)):
    imc = round((peso/((altura/100)**2)),1)
    return templates.TemplateResponse('form4.html', context={'request':
request, 'imc': imc})
```



127.0.0.1:8000/form

Enviando o peso (kg):

Enviando altura (cm):

Enviar

Resultado: Seu imc é:24.5

Sua faixa de classificação do IMC é:

Saudável