

Exercício 1) Configurar ambiente de desenvolvimento para uso de FastAPI

Instale o FastAPI e o servidor Uvicorn no seu ambiente local. Apresente um print da instalação bem-sucedida.

Dica: Utilize o comando `pip install fastapi uvicorn`.

```
PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS
neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$ pyenv virtualenv 3.12 DR3_TP1
neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$ pyenv activate DR3_TP1
(DR3_TP1) neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$ pip install fastapi uvicorn
Collecting fastapi
  Downloading fastapi-0.115.3-py3-none-any.whl.metadata (27 kB)
Collecting uvicorn
  Downloading uvicorn-0.32.0-py3-none-any.whl (6.6 kB)
Collecting starlette<0.42.0,>=0.40.0 (from fastapi)
  Downloading starlette-0.41.0-py3-none-any.whl.metadata (6.0 kB)
Requirement already satisfied: pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in /home/neto/.local/lib/python3.10/site-packages (from fastapi) (2.5.2)
Requirement already satisfied: typing-extensions<=4.8.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from fastapi) (4.12.2)
Requirement already satisfied: annotated-types<=0.4.0 in /home/neto/.local/lib/python3.10/site-packages (from pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (0.6.0)
Requirement already satisfied: pydantic-core<=2.14.5 in /home/neto/.local/lib/python3.10/site-packages (from pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (2.14.5)
Collecting anyio<5,>=3.4.0 (from starlette<0.42.0,>=0.40.0->fastapi)
  Downloading anyio-4.6.2.post1-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: idna<=2.8 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi) (3.7)
Collecting sniffio<=1.1 (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi)
  Using cached sniffio-1.3.1-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: exceptiongroup<=1.0.2 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi) (1.2.2)
Download fastapi-0.115.3-py3-none-any.whl (94 kB)
Download uvicorn-0.32.0-py3-none-any.whl (6.6 kB)
Download starlette-0.41.0-py3-none-any.whl (72 kB)
16.2/16.2 MB 7.0 MB/s eta 0:00:00
Installing collected packages: uvicorn, sniffio, anyio, starlette, fastapi
Successfully installed anyio-4.6.2.post1 fastapi-0.115.3 sniffio-1.3.1 starlette-0.41.0 uvicorn-0.32.0
(DR3_TP1) neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$
```

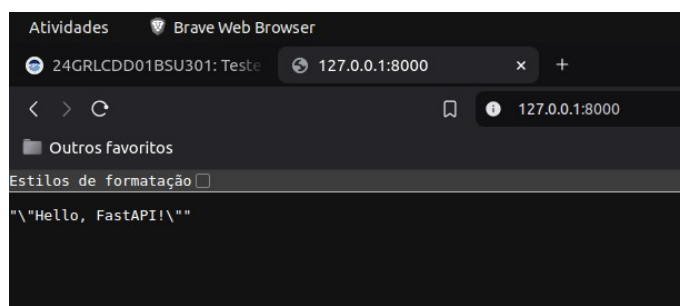
```
PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS
(DR3_TP1) neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$ pip install fastapi uvicorn
Requirement already satisfied: fastapi in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (0.115.3)
Collecting uvicorn
  Downloading uvicorn-0.32.0-py3-none-any.whl.metadata (6.6 kB)
Requirement already satisfied: starlette<0.42.0,>=0.40.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from fastapi) (0.41.0)
Requirement already satisfied: pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in /home/neto/.local/lib/python3.10/site-packages (from fastapi) (2.5.2)
Requirement already satisfied: typing-extensions<=4.8.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from fastapi) (4.12.2)
Requirement already satisfied: click<=7.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from uvicorn) (8.1.7)
Requirement already satisfied: h11<=0.8 in /home/neto/.local/lib/python3.10/site-packages (from uvicorn) (0.14.0)
Requirement already satisfied: annotated-types<=0.4.0 in /home/neto/.local/lib/python3.10/site-packages (from pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (0.6.0)
Requirement already satisfied: pydantic-core<=2.14.5 in /home/neto/.local/lib/python3.10/site-packages (from pydantic<1.8.1,>=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi) (2.14.5)
Requirement already satisfied: anyio<5,>=3.4.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from starlette<0.42.0,>=0.40.0->fastapi) (4.6.2.post1)
Requirement already satisfied: idna<=2.8 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi) (3.7)
Requirement already satisfied: sniffio<=1.1 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi) (1.3.1)
Requirement already satisfied: exceptiongroup<=1.0.2 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from anyio<5,>=3.4.0->starlette<0.42.0,>=0.40.0->fastapi) (1.2.2)
Download uvicorn-0.32.0-py3-none-any.whl (6.6 kB)
Installing collected packages: uvicorn
Successfully installed uvicorn-0.32.0
(DR3_TP1) neto@neto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$
```

#####

Exercício 2) Construir uma aplicação simples usando FastAPI

Crie uma aplicação básica em FastAPI que exiba a mensagem "Hello, FastAPI!" na rota principal (/). Execute a aplicação e envie um print da resposta no navegador.

```
1 from fastapi import FastAPI
2
3 app = FastAPI()
4
5 @app.get('/')
6 def read_root():
7     return {"message": "Hello, FastAPI!"}
8
```



#####

Exercício 3) Criar rotas em uma aplicação FastAPI

Adicione mais duas rotas na sua aplicação:

→ Uma rota `/status` que retorna uma mensagem indicando que o servidor está funcionando.

```
from fastapi import FastAPI
```

```
app = FastAPI()
```

```
@app.get('/status')
```

```
def get_status():
```

```
    return {"status": "Servidor Está Funcionando"}
```

```
from fastapi import FastAPI

app = FastAPI()

@app.get('/status')
def get_status():
    return {"status": "Servidor Está Funcionando"}
```

→ Uma rota `/user/{username}` que receba um parâmetro na URL e retorne uma saudação personalizada para o usuário.

```
from fastapi import FastAPI
```

```
app = FastAPI()
```

```
@app.get('/user/{username}')
```

```
def greet_user(username: str):
```

```
    return {"message": f"Hello, {username}!"}
```

```
from fastapi import FastAPI

app = FastAPI()

@app.get('/user/{username}')
def greet_user(username: str):
    return {"message": f"Hello, {username}!"}
```

#####

Exercício 4) Organizar projetos com FastAPI

Estruture seu projeto FastAPI de forma organizada, separando o código em múltiplos arquivos:

→ Um arquivo para as rotas (`routes.py`).

→ Um arquivo principal para a execução do app (`main.py`).

→ Explique brevemente como essa estrutura ajuda na organização e manutenção do projeto.

Além de tornar o projeto mais organizado e claro, uma estrutura bem definida em um projeto FastAPI facilita a manutenção e melhora a compreensão do código.

```
EDITORES ABERTOS
Bem-vindo
X routes.py
main.py
DESENVOLVIMENTO_DE_DATA...
__pycache__
main.python-310.pyc
routes.python-310.pyc
main.py
routes.py

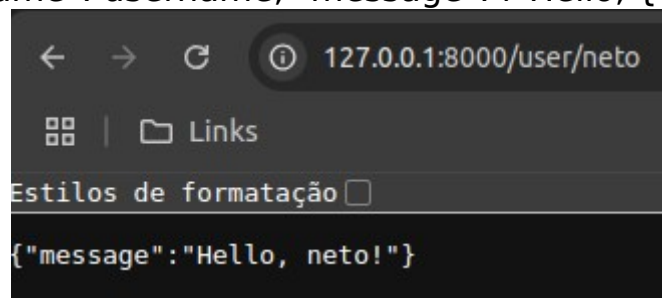
routes.py
1 from fastapi import APIRouter
2
3 router = APIRouter()
4
5 @router.get('/')
6 def read_root():
7     return {"message": "Hello, FastAPI!"}
8
9 @router.get('/status')
10 def get_status():
11     return {"status": "O servidor Está Funcionando"}
12
13 @router.get('/user/{username}')
14 def greet_user(username: str):
15     return {"message": f"Hello, {username}!"}
16
```

#####

Exercício 5) Manipular as respostas usando Response Models

Crie um Response Model para a rota /user/{username} que retorne um JSON com o nome do usuário e uma mensagem. Exemplo de resposta:

```
@router.get('/user/{username}', response_model=UserResponse)
def greet_user(username: str):
    return {"username": username, "message": f"Hello, {username}!"}
```



#####

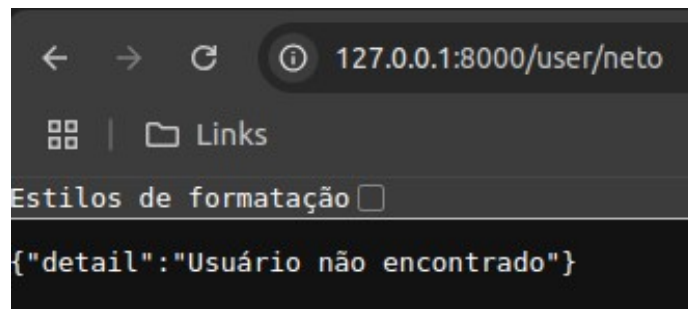
Exercício 6) Tratar erros através de exceções HTTP

Implemente o tratamento de erro para a rota /user/{username}, retornando uma mensagem de erro 404 se o nome de usuário não for encontrado. Utilize o modelo de exceção HTTP:

```
from fastapi import HTTPException
users_db = {
    "Patricia": "Olá, Patricia!",
    "Mateo": "Olá, Mateo!"
}

@router.get('/user/{username}', response_model=UserResponse)
def greet_user(username: str):
    if username not in users_db:
        raise HTTPException(status_code=404, detail="Usuário não encontrado")

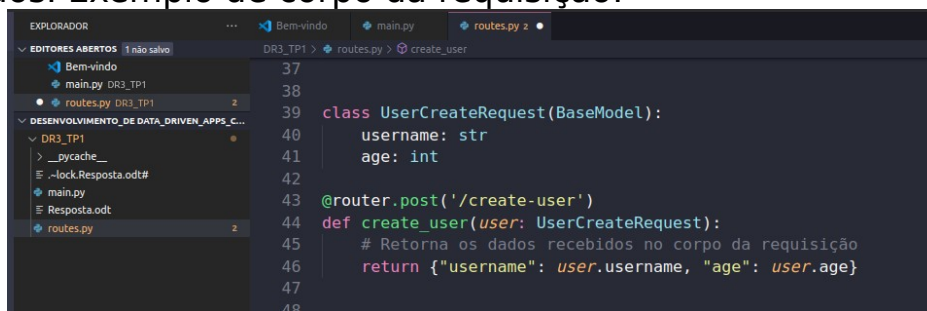
    return {"username": username, "message": users_db[username]}
```



#####

Exercício 7) Criar rotas POST em FastAPI

Adicione uma rota POST chamada /create-user que receba um nome de usuário e idade via JSON no corpo da requisição, e retorne os dados recebidos. Exemplo de corpo da requisição:



#####

Exercício 8) Validar dados de entrada com Pydantic

Utilize o Pydantic para validar a entrada de dados na rota /create-user. O nome de usuário deve ser uma string e a idade um número inteiro positivo. Caso os dados sejam inválidos, a API deve retornar um erro apropriado.



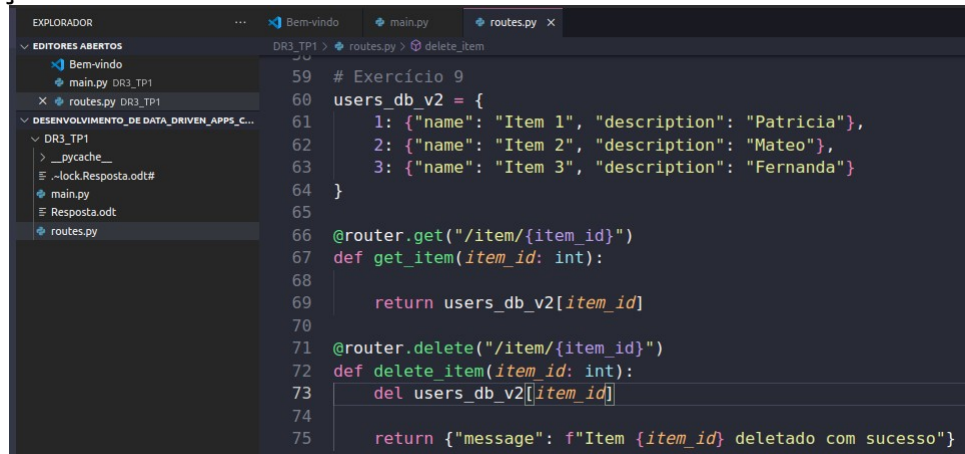
#####

Exercício 9) Criar múltiplos métodos HTTP para uma rota

Crie uma rota `/item/{item_id}` que suporte os métodos GET e DELETE:

GET: Retorna o item correspondente ao `item_id`.

DELETE: Remove o item do sistema e retorna uma mensagem de confirmação.

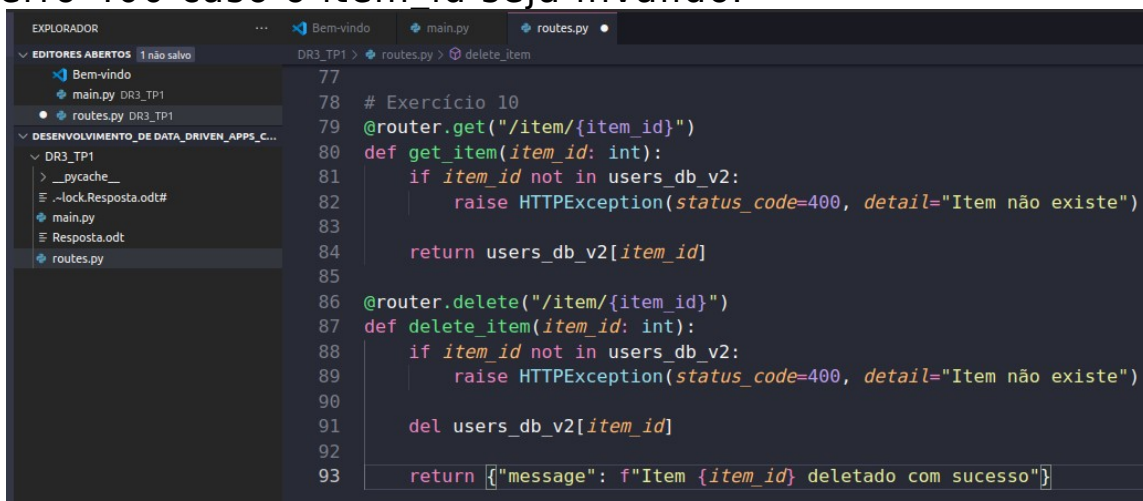


```
59 # Exercício 9
60 users_db_v2 = {
61     1: {"name": "Item 1", "description": "Patricia"},
62     2: {"name": "Item 2", "description": "Mateo"},
63     3: {"name": "Item 3", "description": "Fernanda"}
64 }
65
66 @router.get("/item/{item_id}")
67 def get_item(item_id: int):
68
69     return users_db_v2[item_id]
70
71 @router.delete("/item/{item_id}")
72 def delete_item(item_id: int):
73     del users_db_v2[item_id]
74
75     return {"message": f"Item {item_id} deletado com sucesso"}
```

#####

Exercício 10) Tratar erros de validação com exceções personalizadas

Implemente uma exceção personalizada para tratar erros de validação na rota `/item/{item_id}`, retornando uma mensagem de erro 400 caso o `item_id` seja inválido.

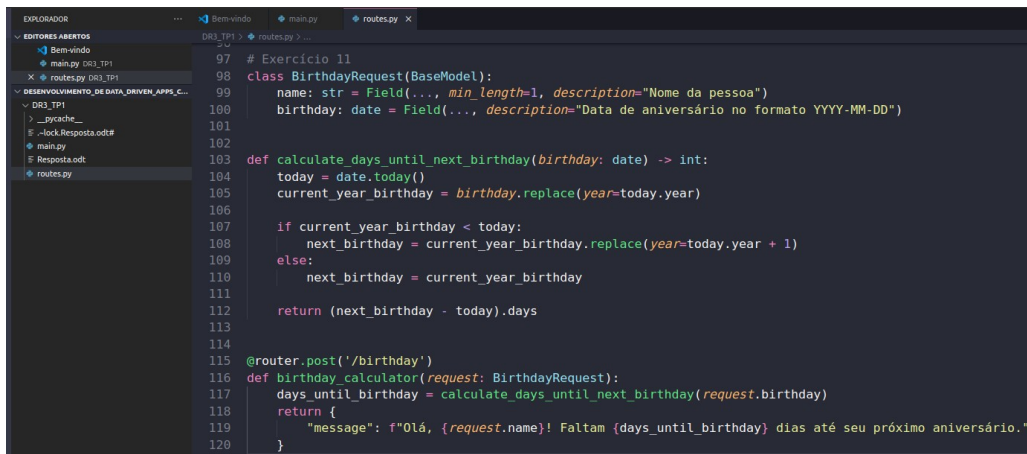


```
77
78 # Exercício 10
79 @router.get("/item/{item_id}")
80 def get_item(item_id: int):
81     if item_id not in users_db_v2:
82         raise HTTPException(status_code=400, detail="Item não existe")
83
84     return users_db_v2[item_id]
85
86 @router.delete("/item/{item_id}")
87 def delete_item(item_id: int):
88     if item_id not in users_db_v2:
89         raise HTTPException(status_code=400, detail="Item não existe")
90
91     del users_db_v2[item_id]
92
93     return {"message": f"Item {item_id} deletado com sucesso"}
```

#####

Exercício 11) Explicar a importância de APIs no contexto da ciência de dados

Crie uma rota POST chamada /birthday que receba um nome e uma data de aniversário (no formato "YYYY-MM-DD") via JSON no corpo da requisição. A rota deve calcular quantos dias faltam até o próximo aniversário da pessoa e retornar uma mensagem personalizada.



```
97 # Exercício 11
98 class BirthdayRequest(BaseModel):
99     name: str = Field(..., min_length=1, description="Nome da pessoa")
100     birthday: date = Field(..., description="Data de aniversário no formato YYYY-MM-DD")
101
102
103 def calculate_days_until_next_birthday(birthday: date) -> int:
104     today = date.today()
105     current_year_birthday = birthday.replace(year=today.year)
106
107     if current_year_birthday < today:
108         next_birthday = current_year_birthday.replace(year=today.year + 1)
109     else:
110         next_birthday = current_year_birthday
111
112     return (next_birthday - today).days
113
114
115 @router.post('/birthday')
116 def birthday_calculator(request: BirthdayRequest):
117     days_until_birthday = calculate_days_until_next_birthday(request.birthday)
118     return {
119         "message": f"Olá, {request.name}! Faltam {days_until_birthday} dias até seu próximo aniversário."
120     }
```

#####

Exercício 12) Criar documentação automática com FastAPI

Use o Swagger UI integrado ao FastAPI para gerar automaticamente a documentação das rotas criadas neste exercício. Envie um print da interface gerada pelo Swagger.

