

LINK DO GITHUB: https://github.com/ARNNETO/Alcebiades_Neto_DR3_TP1

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`.

```

PROGRESS BAR  100%  CONTROL DE DESARROLLO  TERMINAL  100%
metanetmap:~/Documents/Ciencia de datos/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python3 pyenv/virtualenvs/3.12.0R3 TP1
* metanetmap:~/Documents/Ciencia de datos/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python3 pyenv/activate R3P1
(R3P1 TP1) metanetmap:~/Documents/Ciencia de datos/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python3 pip install fastapi unicorn
Collecting fastapi
  Downloading fastapi-0.115.3-py3-none-any.whl.metadata (27 KB)
Collecting fastapi-0.115.3-py3-none-any.whl.metadata (27 KB)
Collecting unicorn
  Downloading unicorn-2.1.1-py2.py3-none-manylinux1_x86_64_manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (1.3 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 in /home/neto/.local/lib/python3.10/site-packages (from fastapi) (2.5.2)
Requirement already satisfied: typing-extensions<4.8.0,>=4.7.0 in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (from fastapi) (4.12.2)
Requirement already satisfied: httpx<0.28.0,>=0.27.0 in /home/neto/.local/lib/python3.10/site-packages (from pydantic<1.8.1,>=1.8.1,<2.0.0 in /home/neto/.local/lib/python3.10/site-packages (from 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 in /home/neto/.local/lib/python3.10/site-packages (from 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: httpx<0.28.0 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: httpx<0.28.0 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)
Downloading fastapi-0.115.3-py3-none-any.whl (94 KB)
Downloading unicorn-2.1.1-py2.py3-none-manylinux1_x86_64_manylinux_2_17_x86_64_manylinux2014_x86_64.whl (16.2 MB)
  26.2/16.2 MB 7.0 MB/s eta 0:00:00
Downloading starlette-0.41.0-py3-none-any.whl (73 KB)
Downloading anyio-4.6.2.post1-py3-none-any.whl (90 KB)
Using cached sniffio-1.3.1-py3-none-any.whl (10 KB)
Installing collected packages: unicorn, sniffio, anyio, starlette, fastapi
Successfully installed anyio-4.6.2.post1 fastapi-0.115.3 sniffio-1.3.1 starlette-0.41.0 unicorn-2.1.1
(R3P1 TP1) metanetmap:~/Documents/Ciencia de datos/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python3

```

```

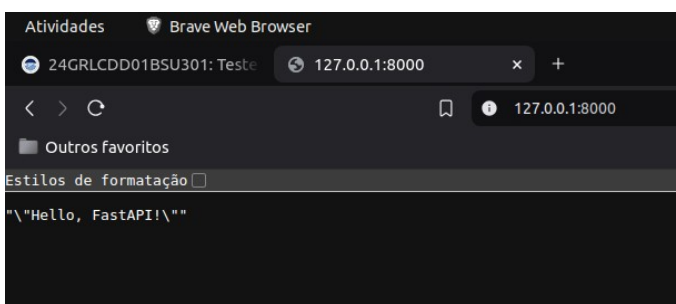
PROBLEMAS  SAÍDA  CONSOLE DE DEPUÇÃO  TERMINAL  PORTAS
• (DR3 TP1) netomneto:~/Documentos/Ciência de dados/4 SEMESTRE/Desenvolvimento De Data Driven Apps Com Python$ pip install fastapi uvicorn
Requirement already satisfied: fastapi in /home/neto/.pyenv/versions/3.10.14/lib/python3.10/site-packages (0.115.3)
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,>=1.8.1,>=2.0.0,!=2.0.1,!=2.1.0,!=2.1.1,!=2.1.2,!=2.1.3,!=2.1.4,!=2.1.5,!=2.1.6,!=2.1.7,!=2.1.8,!=2.1.9,!=2.1.10,!=2.1.11,!=2.1.12,!=2.1.13,!=2.1.14,!=2.1.15,!=2.1.16,!=2.1.17,!=2.1.18,!=2.1.19,!=2.1.20,!=2.1.21,!=2.1.22,!=2.1.23,!=2.1.24,!=2.1.25,!=2.1.26,!=2.1.27,!=2.1.28,!=2.1.29,!=2.1.30,!=2.1.31,!=2.1.32,!=2.1.33,!=2.1.34,!=2.1.35,!=2.1.36,!=2.1.37,!=2.1.38,!=2.1.39,!=2.1.40,!=2.1.41,!=2.1.42,!=2.1.43,!=2.1.44,!=2.1.45,!=2.1.46,!=2.1.47,!=2.1.48,!=2.1.49,!=2.1.50,!=2.1.51,!=2.1.52,!=2.1.53,!=2.1.54,!=2.1.55,!=2.1.56,!=2.1.57,!=2.1.58,!=2.1.59,!=2.1.60,!=2.1.61,!=2.1.62,!=2.1.63,!=2.1.64,!=2.1.65,!=2.1.66,!=2.1.67,!=2.1.68,!=2.1.69,!=2.1.70,!=2.1.71,!=2.1.72,!=2.1.73,!=2.1.74,!=2.1.75,!=2.1.76,!=2.1.77,!=2.1.78,!=2.1.79,!=2.1.80,!=2.1.81,!=2.1.82,!=2.1.83,!=2.1.84,!=2.1.85,!=2.1.86,!=2.1.87,!=2.1.88,!=2.1.89,!=2.1.90,!=2.1.91,!=2.1.92,!=2.1.93,!=2.1.94,!=2.1.95,!=2.1.96,!=2.1.97,!=2.1.98,!=2.1.99,!=2.1.100,!=2.1.101,!=2.1.102,!=2.1.103,!=2.1.104,!=2.1.105,!=2.1.106,!=2.1.107,!=2.1.108,!=2.1.109,!=2.1.110,!=2.1.111,!=2.1.112,!=2.1.113,!=2.1.114,!=2.1.115,!=2.1.116,!=2.1.117,!=2.1.118,!=2.1.119,!=2.1.120,!=2.1.121,!=2.1.122,!=2.1.123,!=2.1.124,!=2.1.125,!=2.1.126,!=2.1.127,!=2.1.128,!=2.1.129,!=2.1.130,!=2.1.131,!=2.1.132,!=2.1.133,!=2.1.134,!=2.1.135,!=2.1.136,!=2.1.137,!=2.1.138,!=2.1.139,!=2.1.140,!=2.1.141,!=2.1.142,!=2.1.143,!=2.1.144,!=2.1.145,!=2.1.146,!=2.1.147,!=2.1.148,!=2.1.149,!=2.1.150,!=2.1.151,!=2.1.152,!=2.1.153,!=2.1.154,!=2.1.155,!=2.1.156,!=2.1.157,!=2.1.158,!=2.1.159,!=2.1.160,!=2.1.161,!=2.1.162,!=2.1.163,!=2.1.164,!=2.1.165,!=2.1.166,!=2.1.167,!=2.1.168,!=2.1.169,!=2.1.170,!=2.1.171,!=2.1.172,!=2.1.173,!=2.1.174,!=2.1.175,!=2.1.176,!=2.1.177,!=2.1.178,!=2.1.179,!=2.1.180,!=2.1.181,!=2.1.182,!=2.1.183,!=2.1.184,!=2.1.185,!=2.1.186,!=2.1.187,!=2.1.188,!=2.1.189,!=2.1.190,!=2.1.191,!=2.1.192,!=2.1.193,!=2.1.194,!=2.1.195,!=2.1.196,!=2.1.197,!=2.1.198,!=2.1.199,!=2.1.200,!=2.1.201,!=2.1.202,!=2.1.203,!=2.1.204,!=2.1.205,!=2.1.206,!=2.1.207,!=2.1.208,!=2.1.209,!=2.1.210,!=2.1.211,!=2.1.212,!=2.1.213,!=2.1.214,!=2.1.215,!=2.1.216,!=2.1.217,!=2.1.218,!=2.1.219,!=2.1.220,!=2.1.221,!=2.1.222,!=2.1.223,!=2.1.224,!=2.1.225,!=2.1.226,!=2.1.227,!=2.1.228,!=2.1.229,!=2.1.230,!=2.1.231,!=2.1.232,!=2.1.233,!=2.1.234,!=2.1.235,!=2.1.236,!=2.1.237,!=2.1.238,!=2.1.239,!=2.1.240,!=2.1.241,!=2.1.242,!=2.1.243,!=2.1.244,!=2.1.245,!=2.1.246,!=2.1.247,!=2.1.248,!=2.1.249,!=2.1.250,!=2.1.251,!=2.1.252,!=2.1.253,!=2.1.254,!=2.1.255,!=2.1.256,!=2.1.257,!=2.1.258,!=2.1.259,!=2.1.260,!=2.1.261,!=2.1.262,!=2.1.263,!=2.1.264,!=2.1.265,!=2.1.266,!=2.1.267,!=2.1.268,!=2.1.269,!=2.1.270,!=2.1.271,!=2.1.272,!=2.1.273,!=2.1.274,!=2.1.275,!=2.1.276,!=2.1.277,!=2.1.278,!=2.1.279,!=2.1.280,!=2.1.281,!=2.1.282,!=2.1.283,!=2.1.284,!=2.1.285,!=2.1.286,!=2.1.287,!=2.1.288,!=2.1.289,!=2.1.290,!=2.1.291,!=2.1.292,!=2.1.293,!=2.1.294,!=2.1.295,!=2.1.296,!=2.1.297,!=2.1.298,!=2.1.299,!=2.1.300,!=2.1.301,!=2.1.302,!=2.1.303,!=2.1.304,!=2.1.305,!=2.1.306,!=2.1.307,!=2.1.308,!=2.1.309,!=2.1.310,!=2.1.311,!=2.1.312,!=2.1.313,!=2.1.314,!=2.1.315,!=2.1.316,!=2.1.317,!=2.1.318,!=2.1.319,!=2.1.320,!=2.1.321,!=2.1.322,!=2.1.323,!=2.1.324,!=2.1.325,!=2.1.326,!=2.1.327,!=2.1.328,!=2.1.329,!=2.1.330,!=2.1.331,!=2.1.332,!=2.1.333,!=2.1.334,!=2.1.335,!=2.1.336,!=2.1.337,!=2.1.338,!=2.1.339,!=2.1.340,!=2.1.341,!=2.1.342,!=2.1.343,!=2.1.344,!=2.1.345,!=2.1.346,!=2.1.347,!=2.1.348,!=2.1.349,!=2.1.350,!=2.1.351,!=2.1.352,!=2.1.353,!=2.1.354,!=2.1.355,!=2.1.356,!=2.1.357,!=2.1.358,!=2.1.359,!=2.1.360,!=2.1.361,!=2.1.362,!=2.1.363,!=2.1.364,!=2.1.365,!=2.1.366,!=2.1.367,!=2.1.368,!=2.1.369,!=2.1.370,!=2.1.371,!=2.1.372,!=2.1.373,!=2.1.374,!=2.1.375,!=2.1.376,!=2.1.377,!=2.1.378,!=2.1.379,!=2.1.380,!=2.1.381,!=2.1.382,!=2.1.383,!=2.1.384,!=2.1.385,!=2.1.386,!=2.1.387,!=2.1.388,!=2.1.389,!=2.1.390,!=2.1.391,!=2.1.392,!=2.1.393,!=2.1.394,!=2.1.395,!=2.1.396,!=2.1.397,!=2.1.398,!=2.1.399,!=2.1.400,!=2.1.401,!=2.1.402,!=2.1.403,!=2.1.404,!=2.1.405,!=2.1.406,!=2.1.407,!=2.1.408,!=2.1.409,!=2.1.410,!=2.1.411,!=2.1.412,!=2.1.413,!=2.1.414,!=2.1.415,!=2.1.416,!=2.1.417,!=2.1.418,!=2.1.419,!=2.1.420,!=2.1.421,!=2.1.422,!=2.1.423,!=2.1.424,!=2.1.425,!=2.1.426,!=2.1.427,!=2.1.428,!=2.1.429,!=2.1.430,!=2.1.431,!=2.1.432,!=2.1.433,!=2.1.434,!=2.1.435,!=2.1.436,!=2.1.437,!=2.1.438,!=2.1.439,!=2.1.440,
```

#####

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!"}
```



#####

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

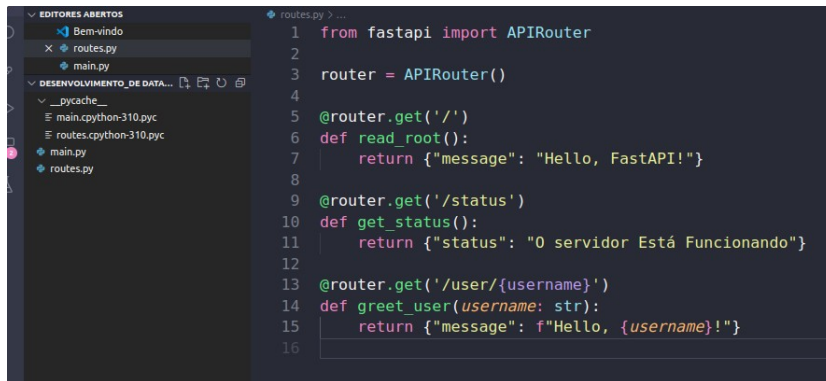
Estruture seu projeto FastAPI de forma organizada, separando o código em múltiplos arquivos:

→ 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.



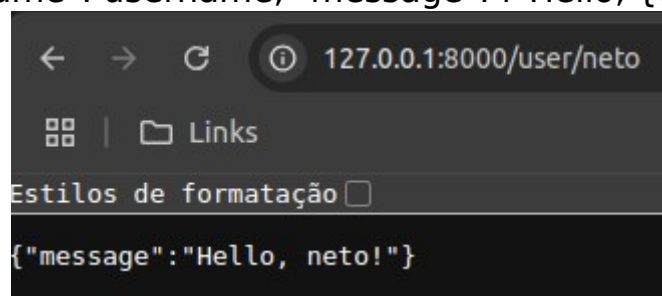
```
1 from fastapi import APIRouter
2
3 router = APIRouter()
4
5 @router.get('/')
6 def read_root():
7     return {"message": "Hello, FastAPI!"}
8
9 @router.get('/status')
10 def get_status():
11     return {"status": "O servidor Está Funcionando"}
12
13 @router.get('/user/{username}')
14 def greet_user(username: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

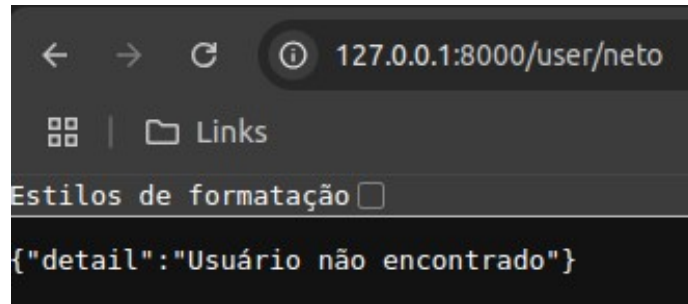
Implemente o tratamento de erro para a rota `/user/{username}`, retornando uma mensagem de erro 404 se o nome de usuário não for encontrado. Utilize o modelo de exceção HTTP:

```
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")
```

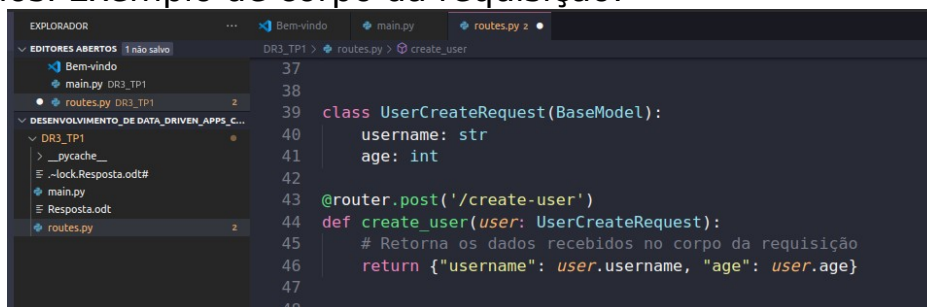
```
return {"username": username, "message": users_db[username]}
```



#####

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.



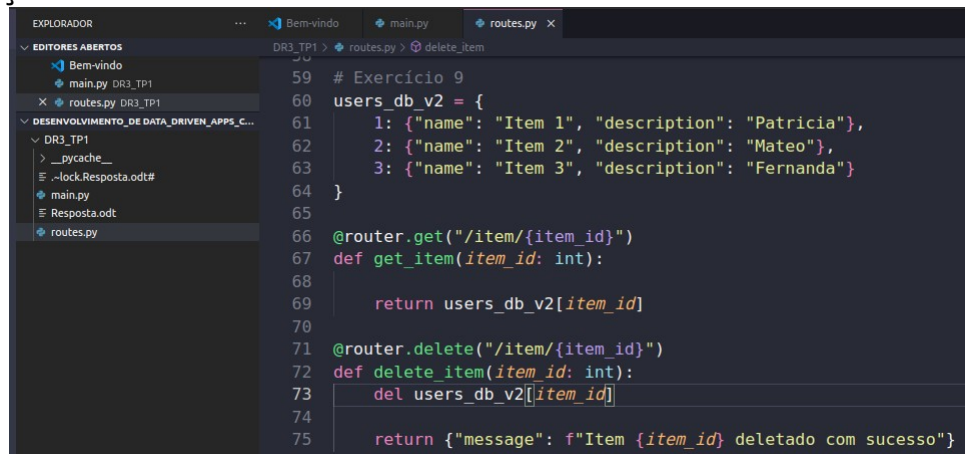
#####

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.

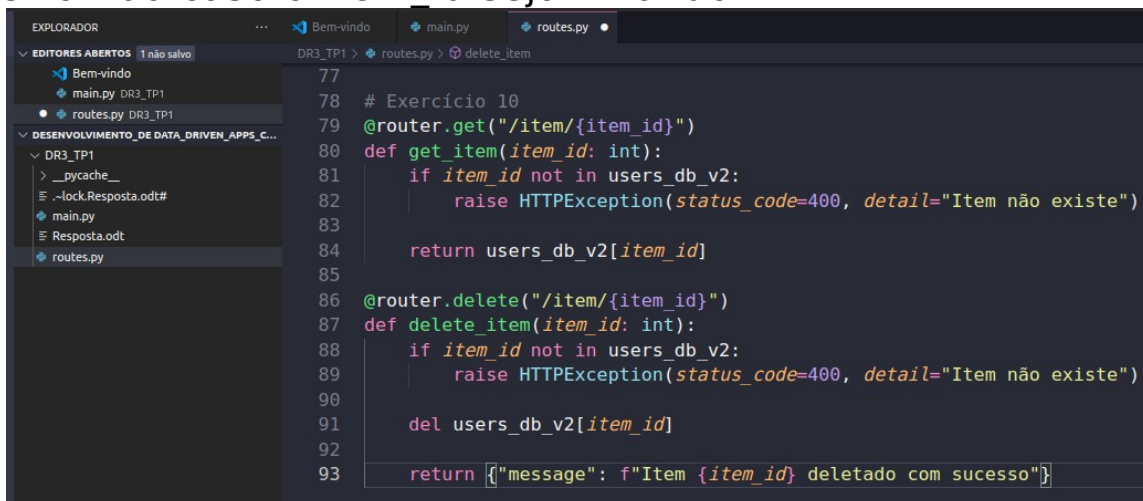


```
59 # Exercício 9
60 users_db_v2 = {
61     1: {"name": "Item 1", "description": "Patricia"},
62     2: {"name": "Item 2", "description": "Mateo"},
63     3: {"name": "Item 3", "description": "Fernanda"}
64 }
65
66 @router.get("/item/{item_id}")
67 def get_item(item_id: int):
68
69     return users_db_v2[item_id]
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.

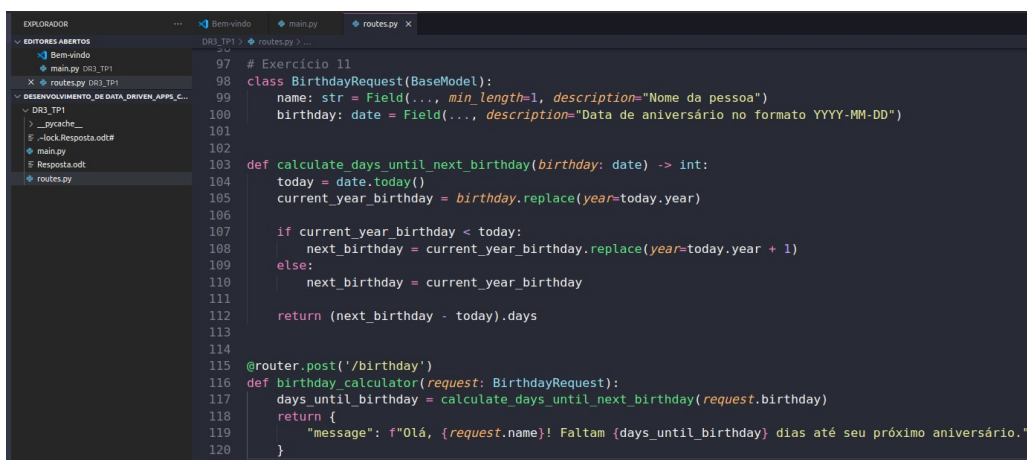


```
77
78 # Exercício 10
79 @router.get("/item/{item_id}")
80 def get_item(item_id: int):
81     if item_id not in users_db_v2:
82         raise HTTPException(status_code=400, detail="Item não existe")
83
84     return users_db_v2[item_id]
85
86 @router.delete("/item/{item_id}")
87 def delete_item(item_id: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

FastAPI 0.100.0 OAS 3.1

swagger-ui

default ^

GET	/	Read Root	▼
GET	/status	Get Status	▼
GET	/user/{username}	Greet User	▼
POST	/create-user	Create User	▼
GET	/item/{item_id}	Get Item	▼
DELETE	/item/{item_id}	Delete Item	▼
POST	/birthday	Birthday Calculator	▼

Schemas ^

BirthdayRequest > Expand all object

HTTPValidationError > Expand all object

UserResponse > Expand all object

ValidationError > Expand all object

UserCreateRequest > Expand all object

UserCreateRequest > Expand all object