Estruturando o Projeto e Criando Rotas do CRUD

> https://fastapidozero.dunossauro.com/02/

Objetivos dessa aula:

- Entendimento dos verbos HTTP, JSON e códigos de resposta
- Compreender a estrutura de um projeto FastAPI e como estruturar rotas CRUD (Criar, Ler, Atualizar, Deletar)
- Aprender sobre a biblioteca Pydantic e sua utilidade na validação e serialização de dados
- Implementação de rotas CRUD em FastAPI
- Escrita e execução de testes para validar o comportamento das rotas

O que é uma API?

Acrônimo de Application Programming Interface (Interface de Programação de Aplicações), uma API é um conjunto de regras e protocolos que permitem a comunicação entre diferentes softwares.

As APIs servem como uma ponte entre diferentes programas, permitindo que eles se comuniquem e compartilhem informações de maneira eficiente e segura.

A arquitetura

O que queremos dizer com cliente servidor

Existe uma aplicação que "Serve" e uma que é cliente de quem serve

O que é HTTP?

HTTP, ou Hypertext Transfer Protocol, é o protocolo fundamental na web para a transferência de dados e comunicação entre clientes e servidores.

Por exemplo, quando chamamos a rota *ou endpoint* `/` da nossa aplicação no teste

```
def test_root_deve_retornar_200_e_ola_mundo():
    client.get('/')
```

Tipos de requisição HTTP

Os verbos HTTP indicam a ação desejada a ser executada em um determinado recurso. Os verbos mais comuns são:

- GET: Recupera recursos
- POST: Cria um novo recurso
- PUT: Atualiza um recurso existente
- PATCH: Atualiza parte de um recurso
- DELETE: Exclui um recurso

Tipos de resposta HTTP

As respostas dadas pela API no HTTP vem com códigos para explicar o que aconteceu

- 200 OK: A solicitação foi bem-sucedida
- 201 Created: A solicitação foi bem-sucedida e um novo recurso foi criado
- 404 Not Found: O recurso solicitado não pôde ser encontrado

```
def test_root_deve_retornar_200_e_ola_mundo():
    response = client.get('/')
    assert response.status_code == 200
```

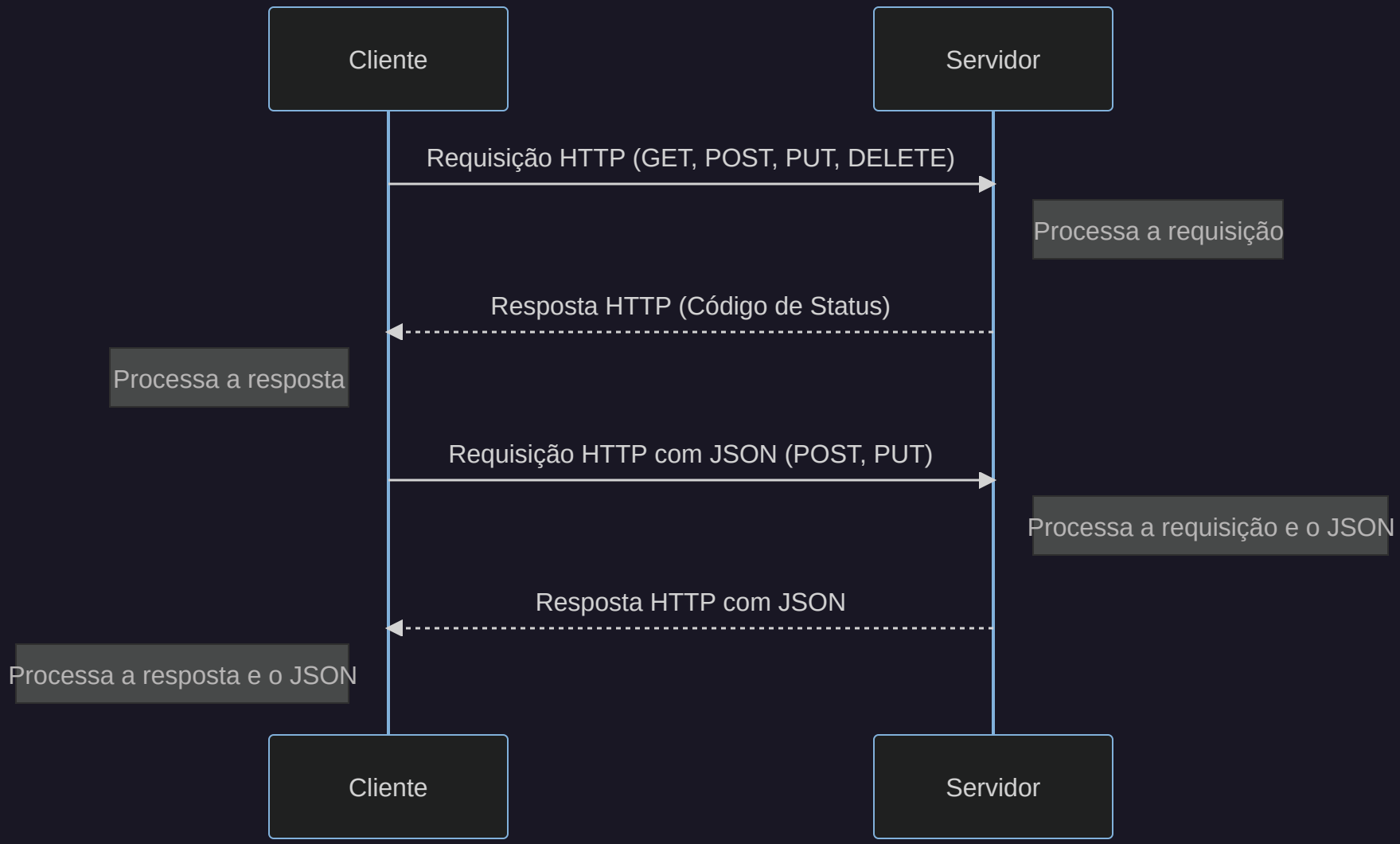
Os dados

Quando estamos falando de APIs modernas, estamos quase sempre falando sobre JSON

```
def test_root_deve_retornar_200_e_ola_mundo():
    response = client.get('/')
    assert response.status_code == 200
    assert response.json() == {'message': 'Olá Mundo!'}
```

Um formato bastante parecido com um dicionário do python

De uma forma geral



O que vamos criar nessa aula?

Endpoints para cadastro, recuperação, alteração e deleção de usuários

Quando a API e o Banco de dados se encontram

Quando falamos de operações de banco de dados, temos um acrônimo para essas operações chamado CRUD:

- **C**reate (Criar): Adicionar novos registros ao banco de dados. No HTTP, essa ação geralmente é associada ao verbo POST.
- **R**ead (Ler): Recuperar registros existentes do banco de dados. No HTTP, essa ação geralmente é associada ao verbo GET.
- **U**pdate (Atualizar): Modificar registros existentes no banco de dados. No HTTP, essa ação geralmente é associada ao verbo PUT ou PATCH.
- **D**elete (Excluir): Remover registros existentes do banco de dados. No HTTP, essa ação geralmente é associada ao verbo DELETE.

Começaremos pelo Create / Post

Criar um endpoint com fastapi que crie um usuário!

```
{
    "username": "dunossauro",
    "email": "dunossauro@email.com",
    "password": "senha-do_dunossauro"
}
```