

WebMAC

Com FastAPI e HTMX

Aula 4

FastAPI

Semanas do Curso

1. HTML, CSS e Responsividade

2. Javascript

3. FastAPI

4. SQL

5. HTMX e UI

6. Docker

Objetivos das semanas

1. Criar páginas web
2. Tornar páginas dinâmicas
- 3. Criar um back-end**
4. Gerenciar dados
5. Criar interfaces modernas
6. Rodar com containers

Canal do Codelab no Youtube



CodeLab

@CodeLabBR · 1,47 mil inscritos · 138 vídeos

O CodeLab é um grupo de extensão inter-universitário que tem como objetivo estimular a ...mais

uclab.xyz/site e mais 5 links

Inscrever-se

Início **Vídeos** Shorts Ao vivo Playlists

Mais recentes

Em alta

Mais antigos



Deployment - Aula5 (Webdev 2025)

56 visualizações · há 3 meses



HTMX - Aula4 (Webdev 2025)

65 visualizações · há 3 meses



SQL e Django Models - Aula3 (Webdev 2025)

78 visualizações · há 4 meses



Django - Aula2 (Webdev 2025)

242 visualizações · há 5 meses



JavaScript - Aula1 (Webdev 2025)

234 visualizações · há 5 meses



HTML e CSS - Aula0 (Webdev 2025)

156 visualizações · há 5 meses



Promises - Aula 6 - WEBDEV 23

103 visualizações · há 2 anos

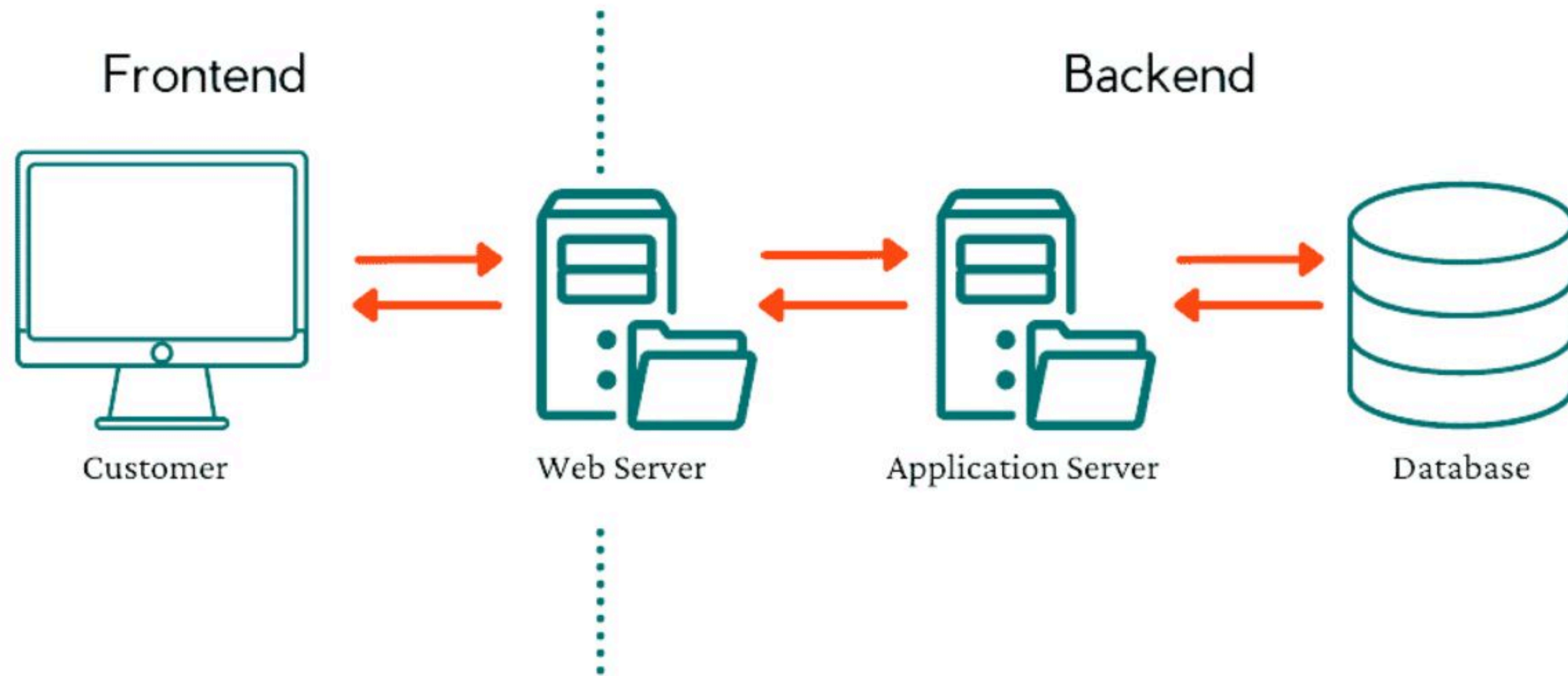


Objetos e Arrays - Aula 5 - WEBDEV 23

89 visualizações · há 2 anos

Front-end x Back-end

How to Connect Frontend and Backend Applications



<https://www.getwidget.dev/blog/connect-frontend-to-backend/>

- HTTP

- Protocolo fundamental da Web
- Conecta clientes (como navegadores) e servidores
- Modelo de requisição e resposta

- Requisição HTTP
 - Métodos HTTP (ex: GET, POST)
 - Mensagens divididas em três partes:
 - Linha de requisição
 - Cabeçalhos (Headers)
 - Corpo da mensagem

- Requisição HTTP
 - Métodos HTTP (ex: GET, POST)
 - Mensagens divididas em três partes:
 - Linha de requisição
 - Cabeçalhos (Headers)
 - Corpo da mensagem (em POST, PUT)

- Requisição HTTP

- Exemplos:

- GET / HTTP/1.1

- Metadados como Host, Content-Type

- {"id":1, "title": "something, ..."}

- Respostas HTTP
 - Em sucesso, o servidor retorna o conteúdo requisitado e um código status (200, 403, 404, 500, ...)

HTTP/1.1 200 OK

Content-Type: application/json

...



FastAPI

- Framework Web para APIs
- Performance rápida (comparável à NodeJS e Go)
- Rápido para codar (type hints :D)
- Uso fácil

main.py

```
from fastapi import FastAPI
```

```
app = FastAPI()
```

```
@app.get("/")
```

```
async def root():
```

```
    return {"message": "Hello World!"}
```

fastapi dev

Type hints

● ● ● example.py

```
name: str = "John"
```

```
age: int = 18
```

```
def greet(name: str) -> str:
```

```
    return f"Hello, {name}"
```



FastAPI