WebMAC

Com FastAPI e HTMX

Aula 9

Integração de Front e Back-end

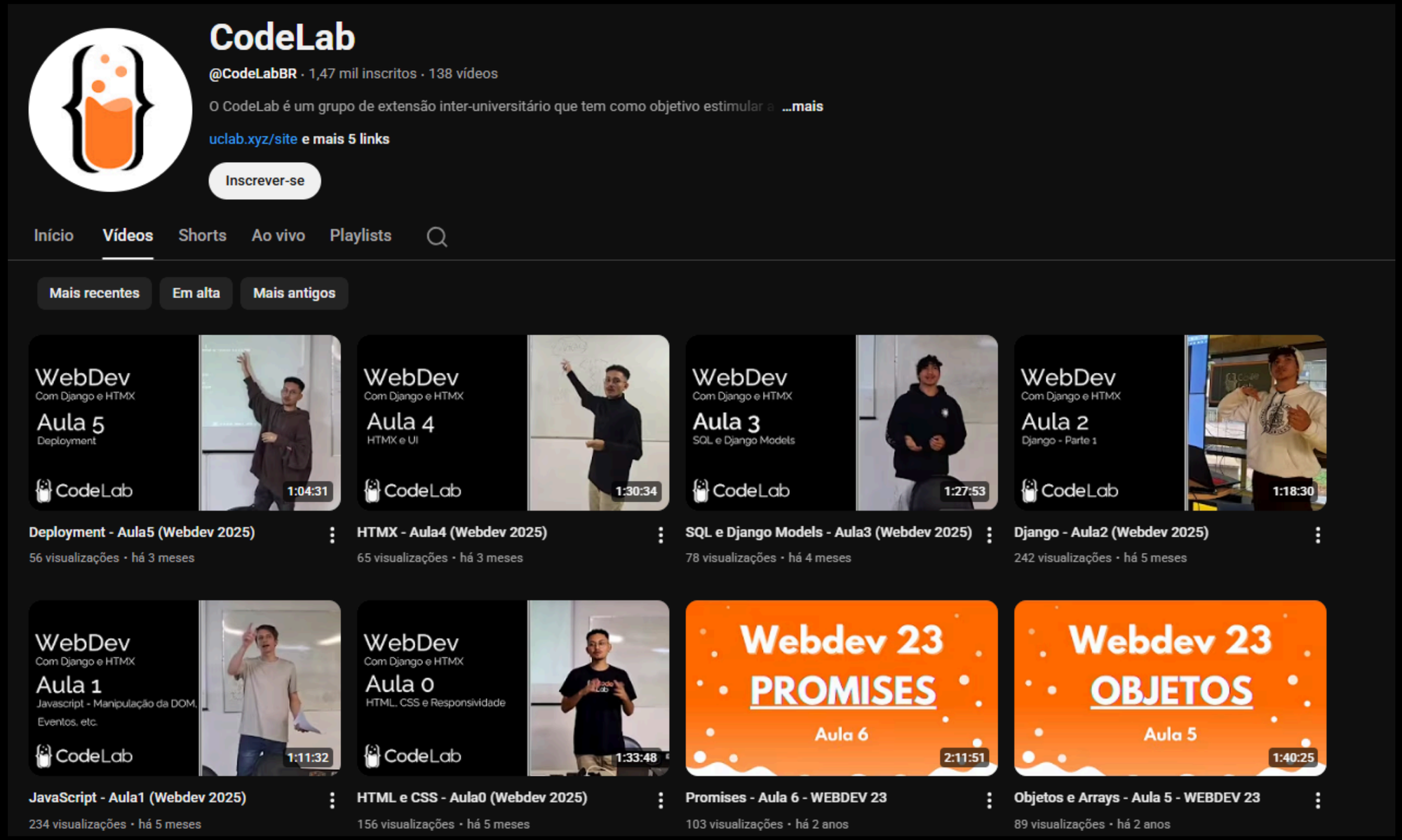
Semanas do Curso

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



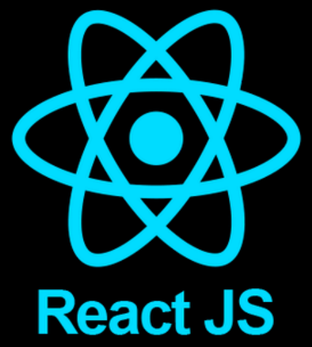
Resumo da Aula:

Tópicos:

- Frameworks de Front-end;
- Integração de BD com front usando HTMX;
- UI e Animações;

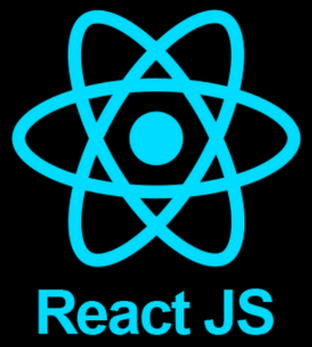
Frameworks de Front-end

- Voltados para o cliente ao invés do servidor
- Atualização da página em tempo real
- Client-side Rendering
- Muito JS para que o próprio browser cosntrua a página



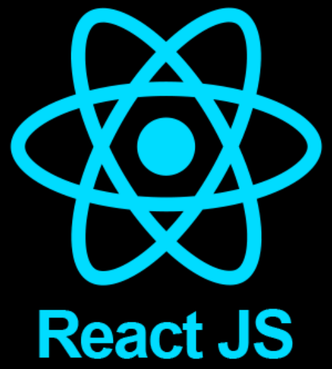
Frameworks de Front-end

Os benefícios de Frameworks de Front-end se assemelham aos de SPAs quando comparados aos de Back-end



Frameworks de Front-end

É comum o uso de ambos os tipos de frameworks no mesmo projeto!



(UI)

+

(Dados)

Frameworks de Front-end

Então, por que não ensinamos nenhum Framework Front-end?

Frameworks de Front-end

- Exemplo da aula passada em REACT

```jsx
import React, { useState, useEffect } from 'react';

const ExemploRequisicao = () => {
  const [conteudo, setConteudo] = useState(null);
  const [carregando, setCarregando] = useState(false);

  const buscarDados = async (idPagina) => {
    setCarregando(true);
    try {
      const response = await fetch(`https://localhost:8000/home/${idPagina}`);
      const data = await response.json();
      setConteudo(data);
    } catch (error) {
      console.error("Erro ao buscar dados:", error);
      setConteudo({ title: "Erro", body: "Não foi possível carregar a página." });
    } finally {
      setCarregando(false);
    }
  };

  useEffect(() => {
    buscarDados('pagina1');
  }, []);
```

```jsx
  return (
    <div>
      <h1>Meu Site</h1>
      <div>
        <button onClick={() => buscarDados('pagina1')}>Página 1</button>
        <button onClick={() => buscarDados('pagina2')}>Página 2</button>
      </div>
      <hr/>

      {carregando ? (
        <p>Carregando...</p>
      ) : conteudo ? (
        <div>
          <h2>{conteudo.title}</h2>
          <p>{conteudo.body}</p>
        </div>
      ) : (
        <p>Nenhum conteúdo disponível.</p>
      )}
    </div>
  );
};

export default ExemploRequisicao;
```

Frameworks de Front-end

- Exemplo da aula passada com HTMX

```
<DOCTYPE html>
<html lang="pt-br">
    <head>
        <title>Meu Site</title>
        <style>...</style>
        <script src="HTMX"></script> <!-- Import do HTMX -->
    </head>
    <body>
        <h1>Meu Site</h1>
        <p>Bem vindo ao meu site!</p>
        <nav>
            <button hx-get="/home/pagina1"
                    hx-target="main">
                Página 1
            </button>
            <button hx-get="/home/pagina2"
                    hx-target="main">
                Página 2
            </button>
        </nav>
        <hr>
        <main>
            <h2>Pagina 1</h2>
            <p>Conteúdo da página 1</p>
        </main>
    </body>
</html>
```

HTMX

Alguns atributos

- hx-get
- hx-post
- hx-put
- hx-delete
- hx-target
- hx-select
- hx-trigger
- hx-swap
- hx-confirm
- hx-push-url
- hx-indicator
- . . .