

WebMAC

Com FastAPI e HTMX

Aula 8

Introdução ao HTMX

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

Resumo da Aula:

Tópicos:

- CRUD, REST e HATEOAS;
- MPA e SPA;
- Introdução ao HTMX;

Conceitos Iniciais

- **CRUD**: Sigla para as operações básicas de um banco de dados.

(SQL)

Create



(HTTP)

POST

Read



GET

Update



PUT

Delete



DELETE

Conceitos Iniciais

- **REST** (Representational State Transfer): Estilo arquitetural para comunicação entre sistemas e aplicações web.

O conceito de **REST** é baseado no fundamento de **Interface Uniforme**

Conceitos Iniciais

- **Interface Uniforme** possui as seguintes características:
 1. Identificação de Recursos
 2. Manipulação de recursos por representação
 3. Mensagens auto-descritivas
 4. Hypermedia as the engine of application state

Conceitos Iniciais

- **Interface Uniforme:** Exemplo

www.exemplo.com/contatos/37

```
<body>
  <section>
    <p>
      Name: Fulano
    </p>
    <p>
      Email: fulano@usp.br
    </p>
    <p>
      <a href="/contatos/37/editar">Editar Contato</a>
      <a href="/contatos/37/email">Acessar Email</a>
      <a href="/contatos/37/arquivo">Arquivar</a>
    </p>
  </section>
</body>
```

1. Identificação de Recursos

2. Manipulação de recursos por representação

3. Mensagens auto-descritivas

4. HATEOAS

Conceitos Iniciais

- REST

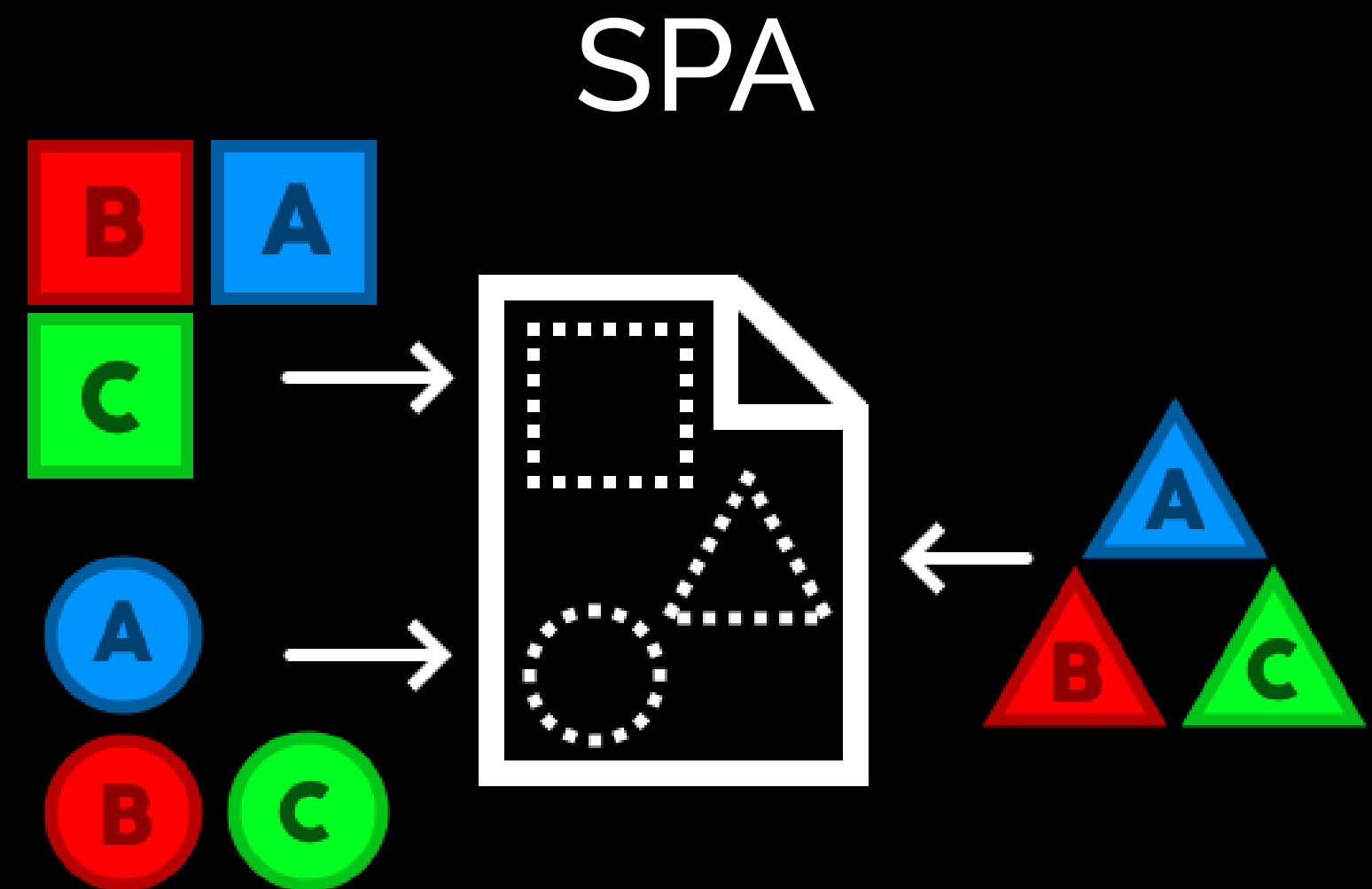
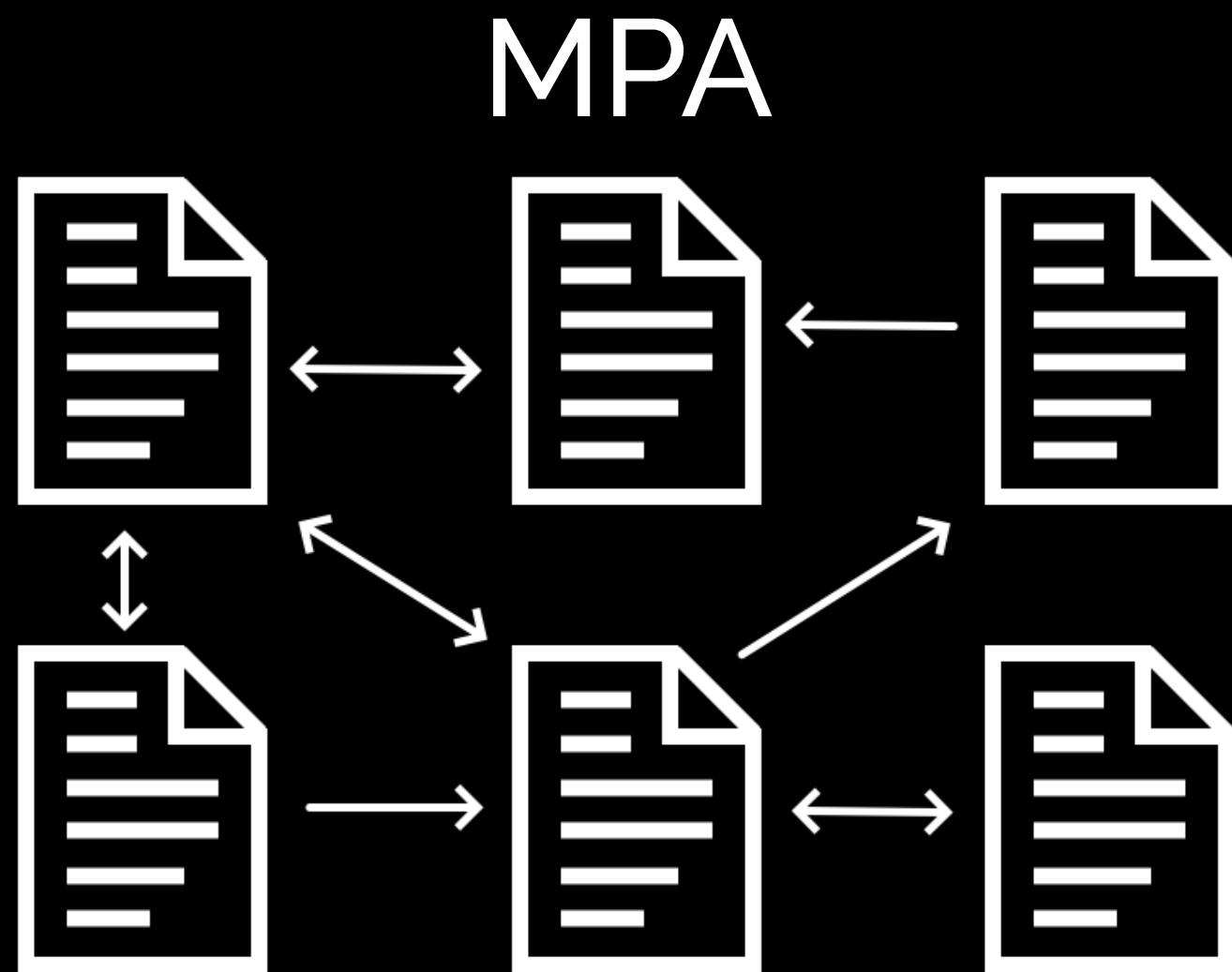
REST NÃO É API DE JSON

```
{  
  "nome" : "Fulano",  
  "email" : "fulano@usp.br"  
}
```

1. Identifica recursos?
2. Manipula por representação?
3. Auto descritiva?
4. HATEOAS?

Conceitos Iniciais

- **MPA e SPA:** Multi-Page Application e Single Page Application



Conceitos Iniciais

- **AJAX** (Asynchronous JavaScript and XML):

```
document.addEventListener('DOMContentLoaded', function() {
  const button = document.getElementById('fetchHtml');
  const result = document.getElementById('result');

  button.addEventListener('click', async function() {
    try {
      const response = await fetch('/server-endpoint');
      if (!response.ok) {
        throw new Error('Erro ao buscar o HTML');
      }
      const html = await response.text();
      result.innerHTML = html; // Insere o HTML retornado no elemento result
    } catch (error) {
      result.innerHTML = `

${error.message}</p>`;
    }
  });
});


```

HTMX

Biblioteca de JS que estende e generaliza a ideia central do HTML como hipertexto, abrindo muitas possibilidades diretamente na linguagem:

```
<button hx-post="/clicked"  
        hx-trigger="click"  
        hx-target="#parent-div"  
        hx-swap="outerHTML">  
    Clique Aqui!  
</button>
```

HTMX

- Qualquer elemento pode fazer requisições
- Qualquer evento pode acionar requisições
- Qualquer verbo HTTP pode ser usado
- Qualquer elemento pode ser o alvo para atualização pela requisição

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`
- ...