

Julho, 2022

Discover.
Learn.
Grow.
Repeat.

vnt/school

powered by  venturus

Onde tem conhecimento tem vnt/school



Apresentação

Facilitadores



Artur Braga Pinheiro



Caique de Castro Soares



Guilherme Ferreira de Souza



Hélio Victor Apolinário Soares



João Vitor Aleixo Dias



Raphael De Souza Ramalho Victor Ferreira

**Agora, a gente quer conhecer
um pouco sobre vocês!!!**





Código 4797 8597



<https://www.menti.com/6sphxwewcm>

Organização do Curso

Semana 11/07 - 15/07

Segunda	Terça	Quarta	Quinta	Sexta
<ul style="list-style-type: none">• WEB• HTTP• HTML• SPA• React	<ul style="list-style-type: none">• React• Typescript• CSS• Next.js	<ul style="list-style-type: none">• Projeto	<ul style="list-style-type: none">• Projeto	<ul style="list-style-type: none">• Projeto

Programação do dia

1 **Conceitos básicos**

2 **Web e HTML**

3 **== Intervalo ==**

4 **React**

5 **Revisão**

6 **Dúvidas**



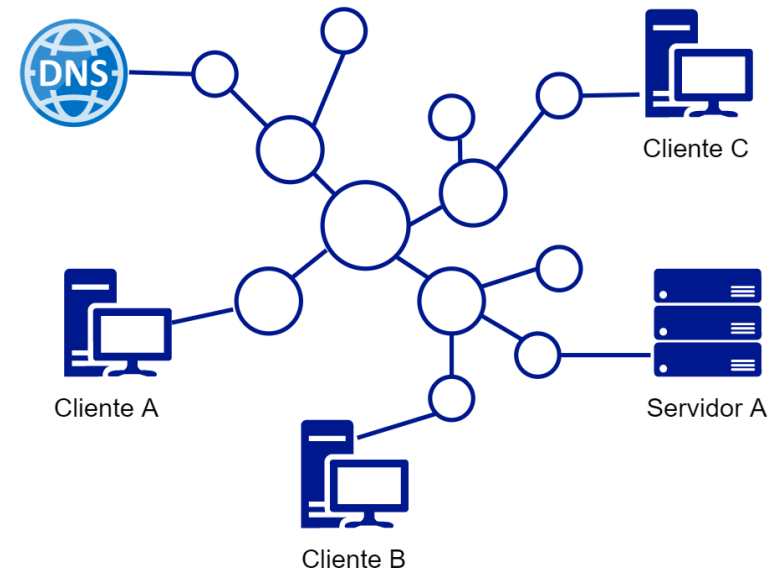
Conceitos básicos



Como funciona a web e o navegador?

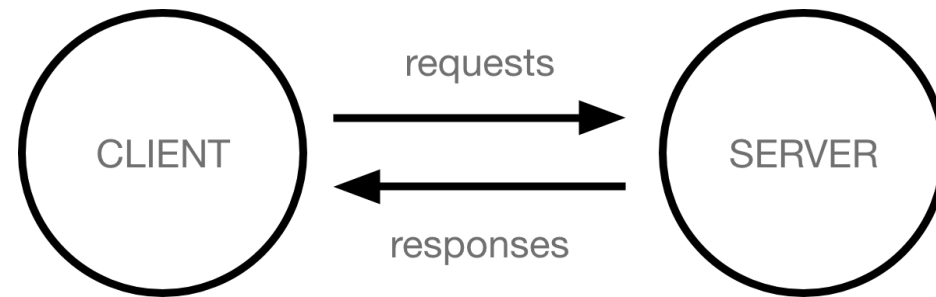
WEB e navegadores

- A web é uma grande rede formada por computadores trocando informação.
- Quando um usuário acessa um site, ele solicita uma cópia de um arquivo (o site alvo) que está armazenado em outro computador, chamado de servidor.



Clientes e servidores

- Computadores conectados à web são chamados de clientes ou de servidores a depender da função que estão exercendo.



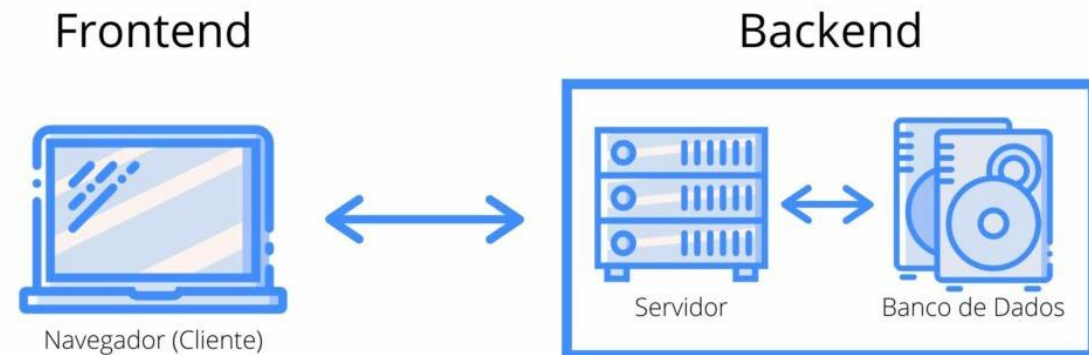
- Os que solicitam informações de outros dispositivos (requests), como textos, imagens, vídeos e outros serviços, são chamados de clientes.
- Computadores que possuem essas informações e as distribuem à outros quando solicitados (responses), são os servidores.



Front-end / Back-end

Definição

- Front-end
 - Responsável pelo código que será processado no cliente, pela construção dos elementos visuais da aplicação e pela boa experiência dos usuários.
- Back-end
 - Responsável pelo código processado no servidor e pelo banco de dados. Essa área garante que as informações sejam organizadas, armazenadas e distribuídas corretamente entre os clientes.



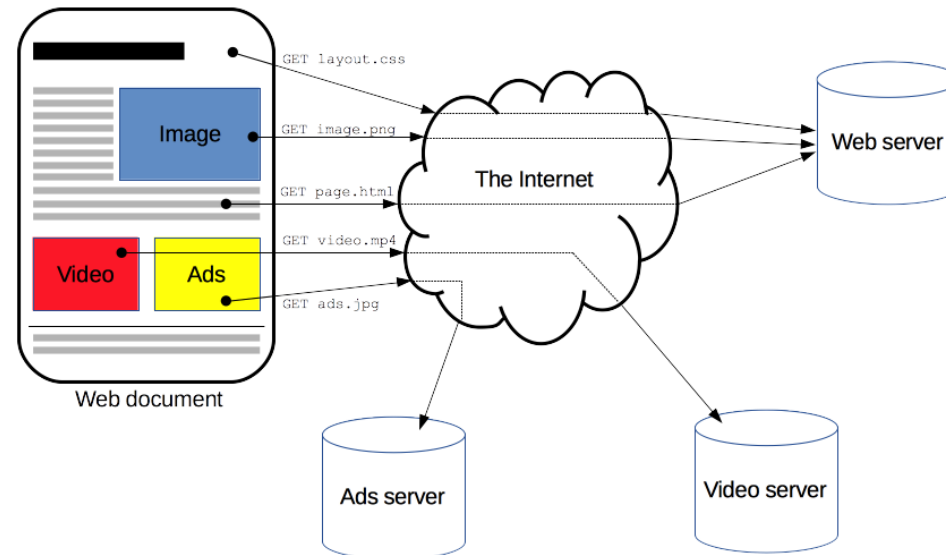


HTTP

Hypertext Transfer Protocol

HTTP

- HTTP é um protocolo que permite a obtenção de recursos;
- É a base de qualquer troca de dados na Web e um protocolo cliente-servidor;
- As requisições são iniciadas pelo destinatário, ex: navegador da Web. Um documento completo é reconstruído a partir dos diferentes sub-documentos obtidos, como por exemplo texto, descrição do layout, imagens, vídeos, scripts e muito mais.





HTTP

Tipos de requisição

HTTP

- GET
 - Tipo mais comum. Usado para solicitar um dado do servidor
- POST
 - Empregado quando deseja-se criar ou processar alguma informação no servidor
- PUT
 - Normalmente utilizado para atualizar alguma informação já existente no servidor
- DELETE
 - Serve para deletar um dado no servidor

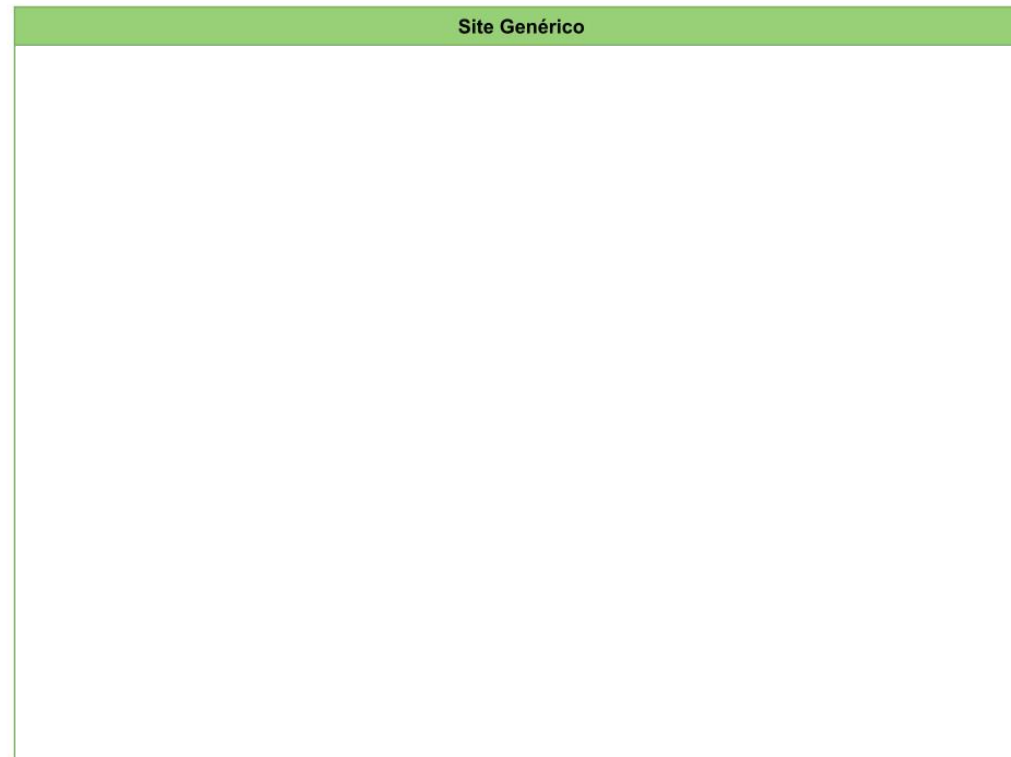


HTTP

Funcionamento

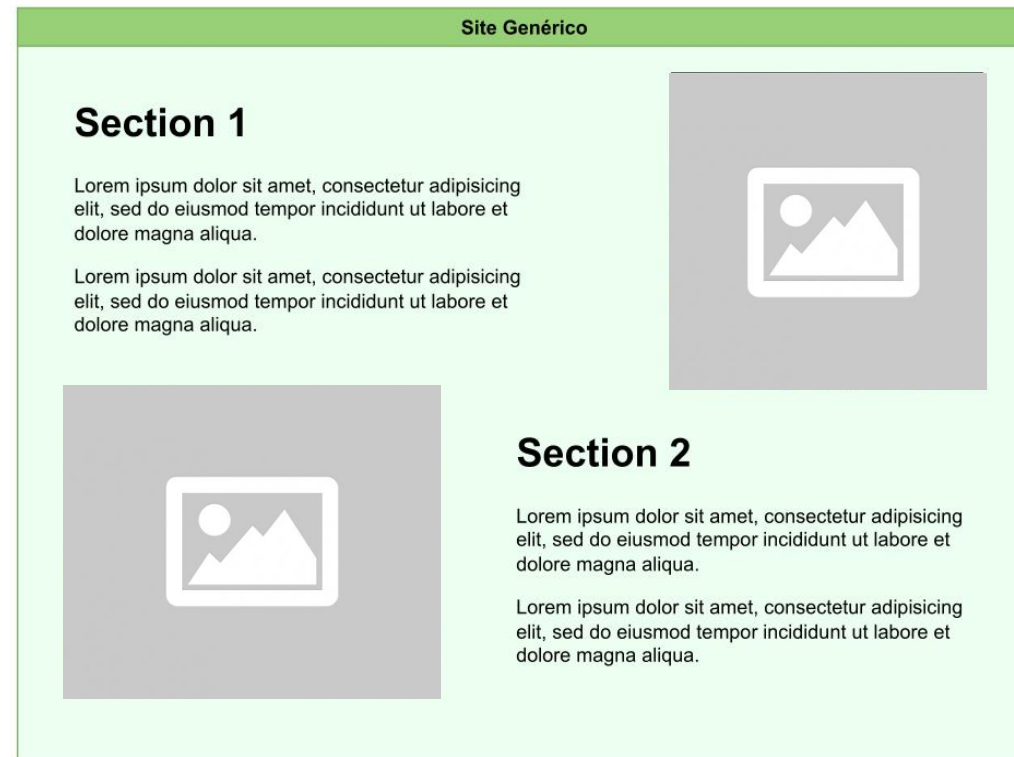
HTTP

- Usuário digita uma url no navegador
- Navegador faz uma requisição ao servidor, solicitando um HTML



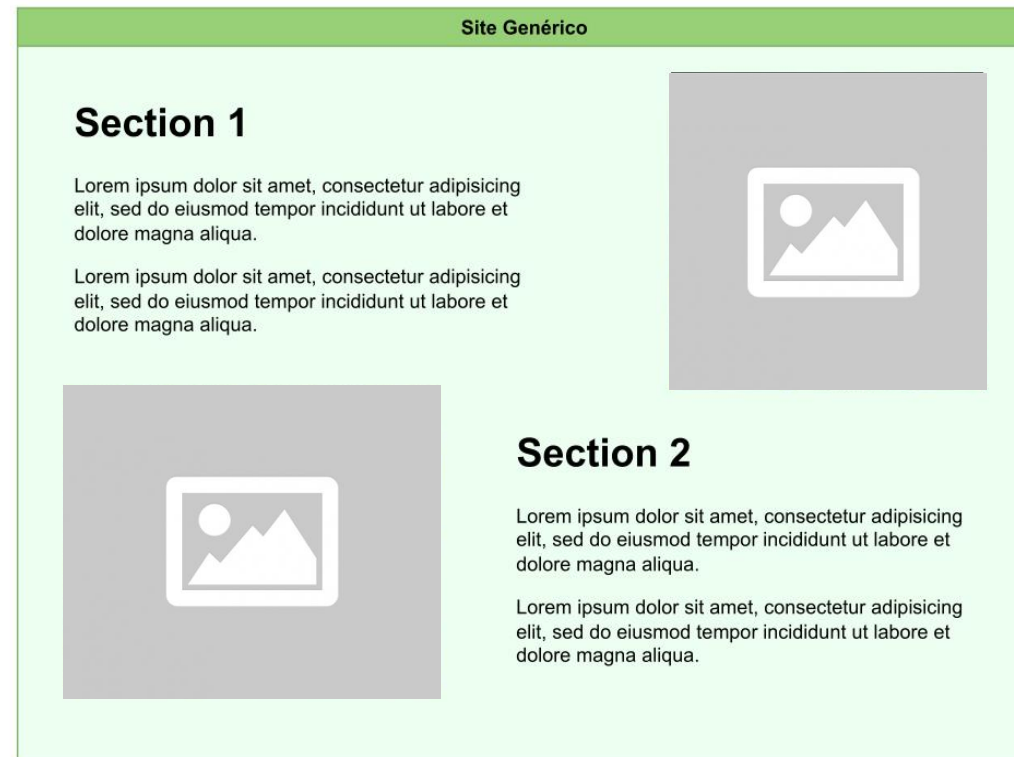
HTTP

- Servidor identifica o HTML solicitado em seu banco de dados e devolve ao cliente, uma cópia em sua resposta



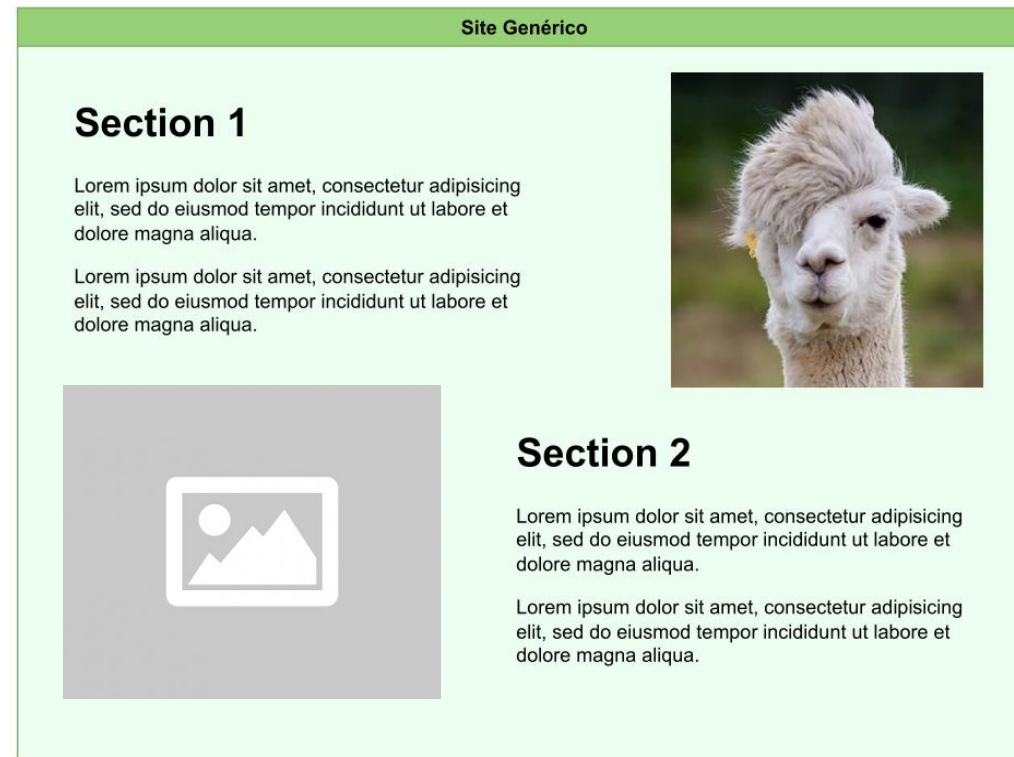
HTTP

- O navegador processa o HTML recebido, identifica que uma imagem precisa ser importada do servidor e realiza um novo request para baixar essa imagem.



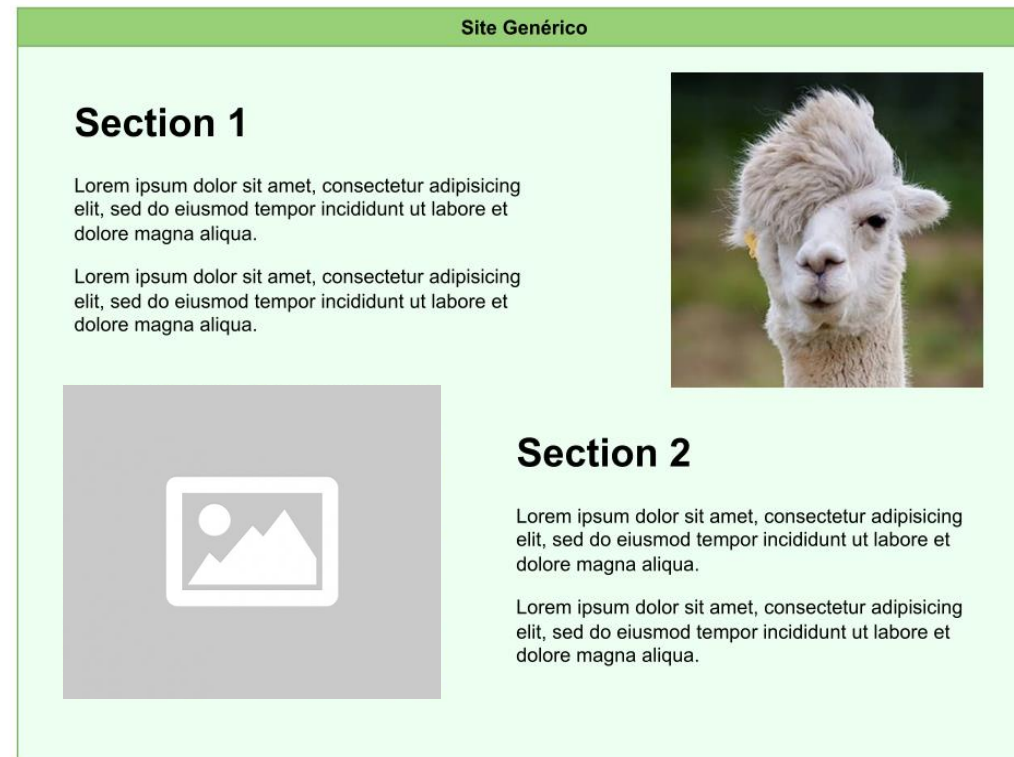
HTTP

- O servidor identifica a requisição do cliente por um imagem específica, procura em seu banco de dados por esse arquivo e o envia para o cliente.



HTTP

- O cliente identifica que um novo vídeo precisa ser importado para o HTML. Uma nova requisição é feita ao servidor.



HTTP

- O servidor identifica a requisição e devolve o vídeo como resposta.





Site Genérico

Section 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.





Section 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Apresentação ao DevTools



VSCode instalado?



Configurar VSCode

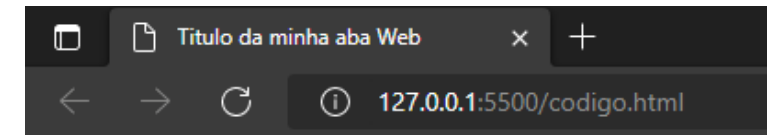


HTML: Linguagem de Marcação de Hipertexto

HTML

- HTML é o bloco de construção mais básico da web.
- Ele usa tags semânticas para indicar ao navegador quais conteúdos serão exibidos na tela.

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Título da minha aba Web</title>
8 </head>
9 <body>
10  <h1>Título principal</h1>
11  <h2>Título secundário</h2>
12  <p>Parágrafo</p>
13  <ol>
14    <li>Item A</li>
15    <li>Item B</li>
16    <li>Item C</li>
17  </ol>
18 </body>
19 </html>
```



Título principal

Título secundário

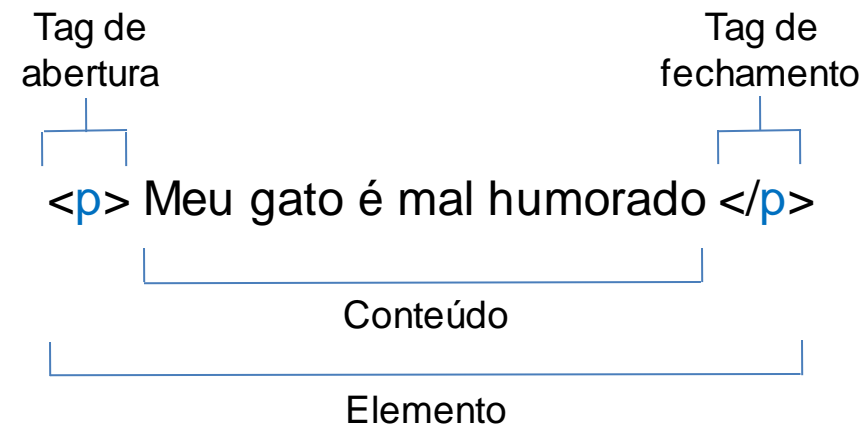
Parágrafo

1. Item A
2. Item B
3. Item C

HTML

Elementos

- Elementos HTML são compostos por tag de abertura, conteúdo e tag de fechamento.
- Tags de abertura são formadas pelo nome do elemento entre "<" e ">", enquanto nas de fechamento, o nome do elemento fica entre "</" e ">"




- A divisão do conteúdo total do site em elementos menores, a partir das tags, auxilia o desenvolvedor à controlar o comportamento, a posição e a exibição de cada parte visual de sua aplicação

HTML

Atributos

- A tag de abertura também podem conter atributos
- Atributos servem para adicionar características extras ao elemento

`<p class="anotacao"> Meu gato é mal humorado </p>`



Atributo

The diagram shows a horizontal line with vertical end caps, positioned under the text 'class="anotacao"' in the code above. A vertical line descends from the center of this horizontal line to the word 'Atributo' below.

- A partir das tags e de seus atributos, o desenvolvedor é capaz de interagir com as informações exibidas no site, podendo personalizar o visual de um trecho específico ou até mesmo, modifica-lo dinamicamente.

HTML

Estrutura

- O HTML de um site é formado por uma série de elementos envolvidos por outros elementos.
- Os elementos envolvidos são chamados de filhos, enquanto os que envolvem, são chamados de pais. No exemplo abaixo, o elemento `` é filho do `<body>`, pai do `` e irmão do `<p>`.

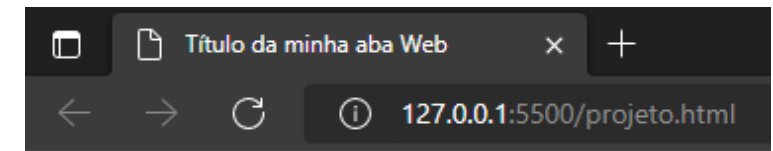
```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Título da minha aba Web</title>
8 </head>
9 <body>
10   <h1>Título principal</h1>
11   <h2>Título secundário</h2>
12   <p>Parágrafo</p>
13   <ol>
14     <li>Item A</li>
15     <li>Item B</li>
16     <li>Item C</li>
17   </ol>
18 </body>
19 </html>
```

HTML

Estrutura

- O parentesco entre elementos é chamado de hierarquia.
- Ao utilizar elementos dentro de outros elementos, o código se torna mais simples e com menos estruturas repetidas, pois algumas características dos elementos pais podem ser herdadas pelos elementos filhos

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Título da minha aba Web</title>
8   </head>
9   <body>
10    <h1 style="color: red">Título principal</h1>
11    <h2 style="color: red">Titulo secundário</h2>
12    <p style="color: red">Parágrafo</p>
13    <ol>
14      <li style="color: red">Item A</li>
15      <li style="color: red">Item B</li>
16      <li style="color: red">Item C</li>
17    </ol>
18  </body>
19 </html>
```



Título principal

Titulo secundário

Parágrafo

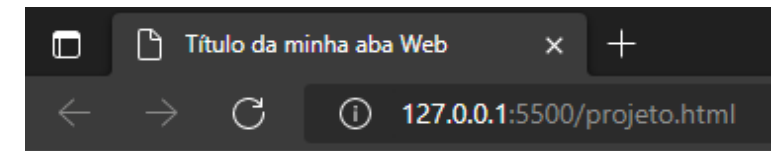
1. Item A
2. Item B
3. Item C

HTML

Estrutura

- No exemplo, o objetivo era colocar todo o texto em vermelho, então ao invés de atribuir a cor vermelha a cada um dos elementos da tela, foi possível simplificar o trabalho atribuindo apenas ao elemento pai. O `<h1>`, `<h2>` e demais, herdaram essa característica por serem filhos do `<body>`.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Título da minha aba Web</title>
8    </head>
9    <body style="color: red">
10     <h1>Título principal</h1>
11     <h2>Título secundário</h2>
12     <p>Parágrafo</p>
13     <ol>
14       <li>Item A</li>
15       <li>Item B</li>
16       <li>Item C</li>
17     </ol>
18   </body>
19 </html>
```



Título principal

Título secundário

Parágrafo

1. Item A
2. Item B
3. Item C

HTML

<head> e <body>

- Duas tags muito importantes e presentes em quase todos os códigos HTML são as tags <head> e a <body>

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <link rel="stylesheet" href="estilo.css">
8      <script src="logica.js"></script>
9      <title>Título da minha aba Web</title>
10 </head>
11 <body>
12     <h1>Título principal</h1>
13     <h2>Título secundário</h2>
14     <p>Parágrafo</p>
15     <ol>
16         <li>Item A</li>
17         <li>Item B</li>
18         <li>Item C</li>
19     </ol>
20 </body>
21 </html>
```


HTML

<head> e <body>

- `<head>`
 - Tag responsável pelas configurações gerais do documento, incluindo título da aba, links para scripts e folhas de estilo (css).

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7    <link rel="stylesheet" href="estilo.css">
8    <script src="logica.js"></script>
9    <title>Título da minha aba Web</title>
10 </head>
11 <body>
12   <h1>Título principal</h1>
13   <h2>Título secundário</h2>
14   <p>Parágrafo</p>
15   <ol>
16     <li>Item A</li>
17     <li>Item B</li>
18     <li>Item C</li>
19   </ol>
20 </body>
21 </html>
```

HTML

<head> e <body>

- `<body>`
 - Tag responsável pelo conteúdo visível ao usuário. Todas as informações presentes na tela de um site estarão posicionadas dentro desta tag.

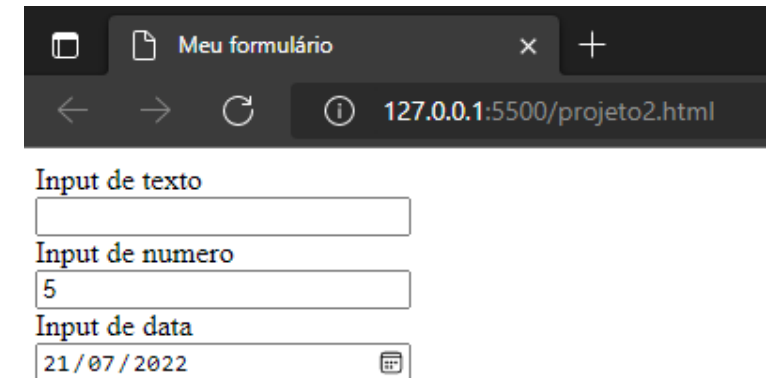
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7    <link rel="stylesheet" href="estilo.css">
8    <script src="logica.js"></script>
9    <title>Título da minha aba Web</title>
10 </head>
11 <body>
12   <h1>Título principal</h1>
13   <h2>Título secundário</h2>
14   <p>Parágrafo</p>
15   <ol>
16     <li>Item A</li>
17     <li>Item B</li>
18     <li>Item C</li>
19   </ol>
20 </body>
21 </html>
```

HTML

Formulários

- Formulários são a forma de o usuário inserir informações como texto, data, números no site para que essas informações sejam capturadas e processadas pelo próprio cliente ou enviadas para processamento no servidor.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Meu formulário</title>
8   </head>
9   <body>
10    <div style="display: flex; flex-direction: column; width: 200px">
11      <label for="texto">Input de texto</label>
12      <input id="texto" type="text" />
13
14      <label for="numero">Input de numero</label>
15      <input id="numero" type="number" />
16
17      <label for="data">Input de data</label>
18      <input id="data" type="date" />
19    </div>
20  </body>
21 </html>
```



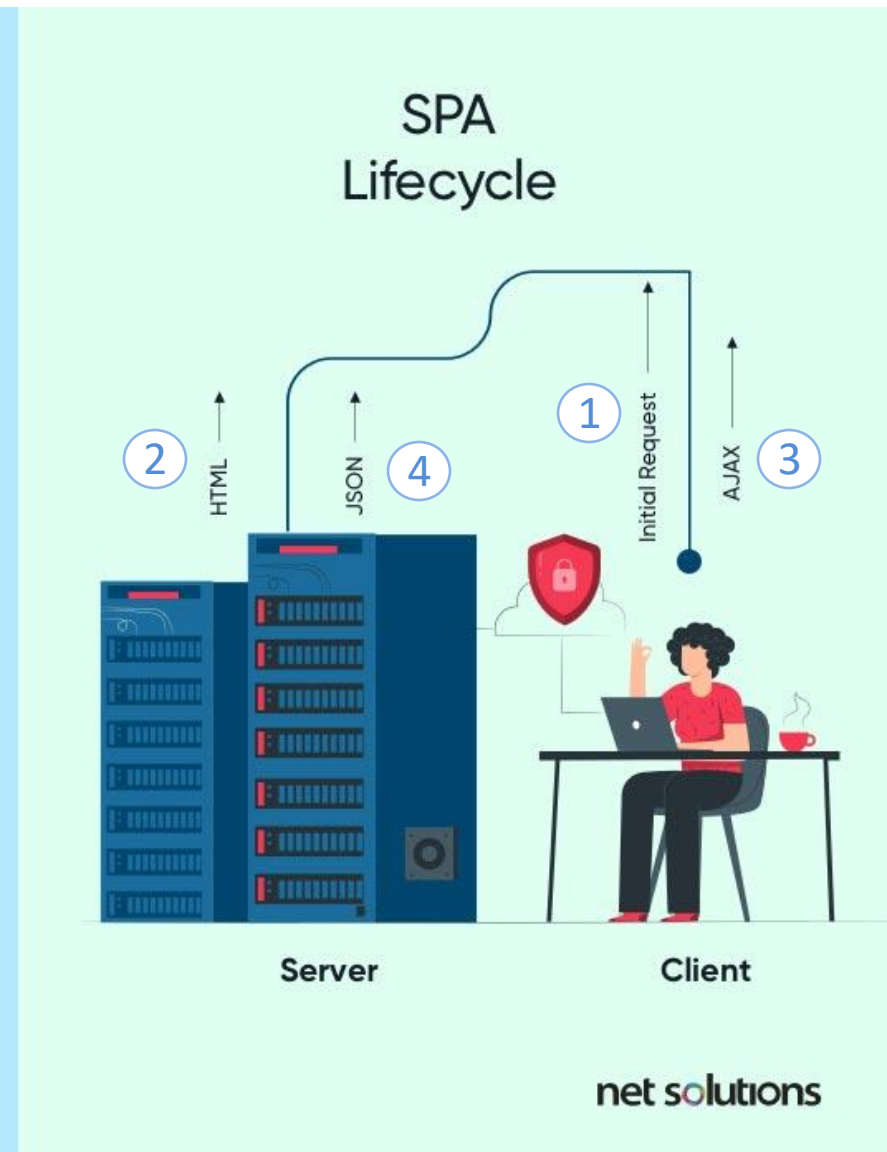
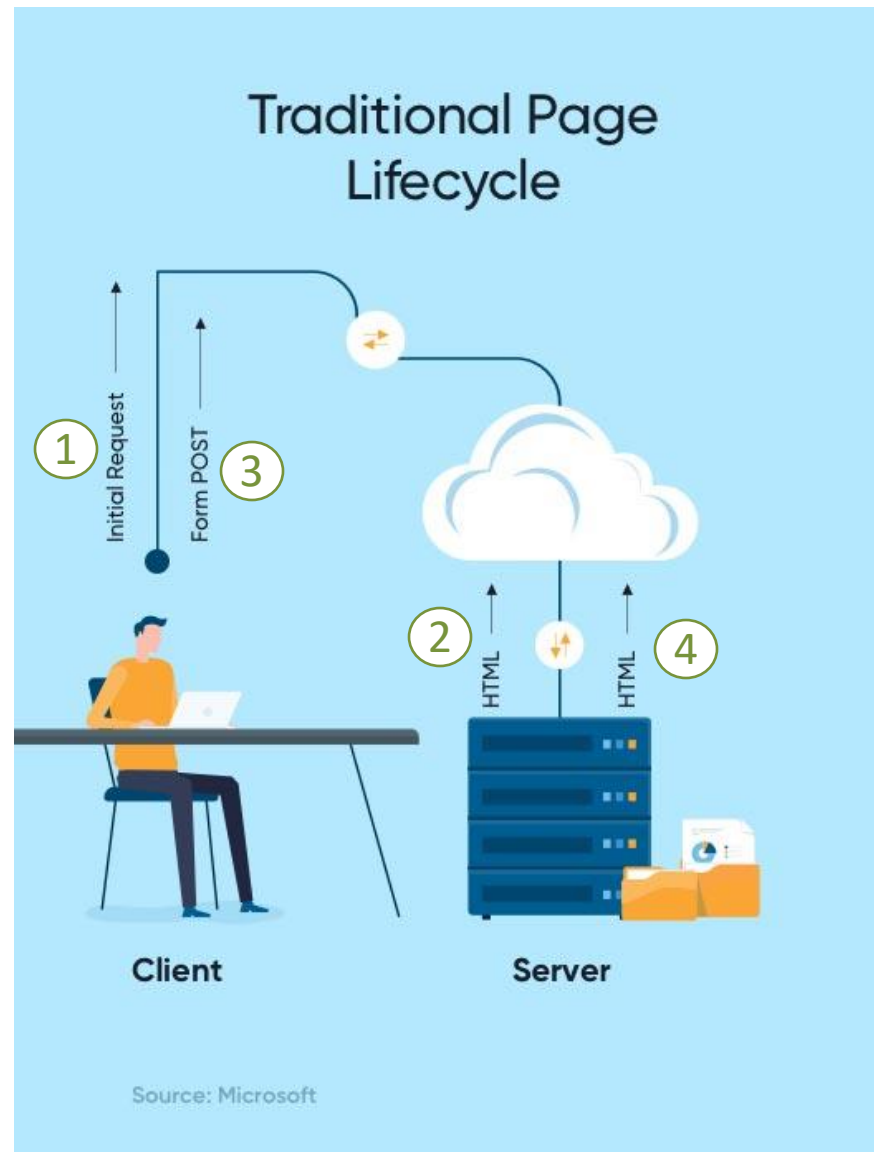


HTML: Primeiras linhas de código



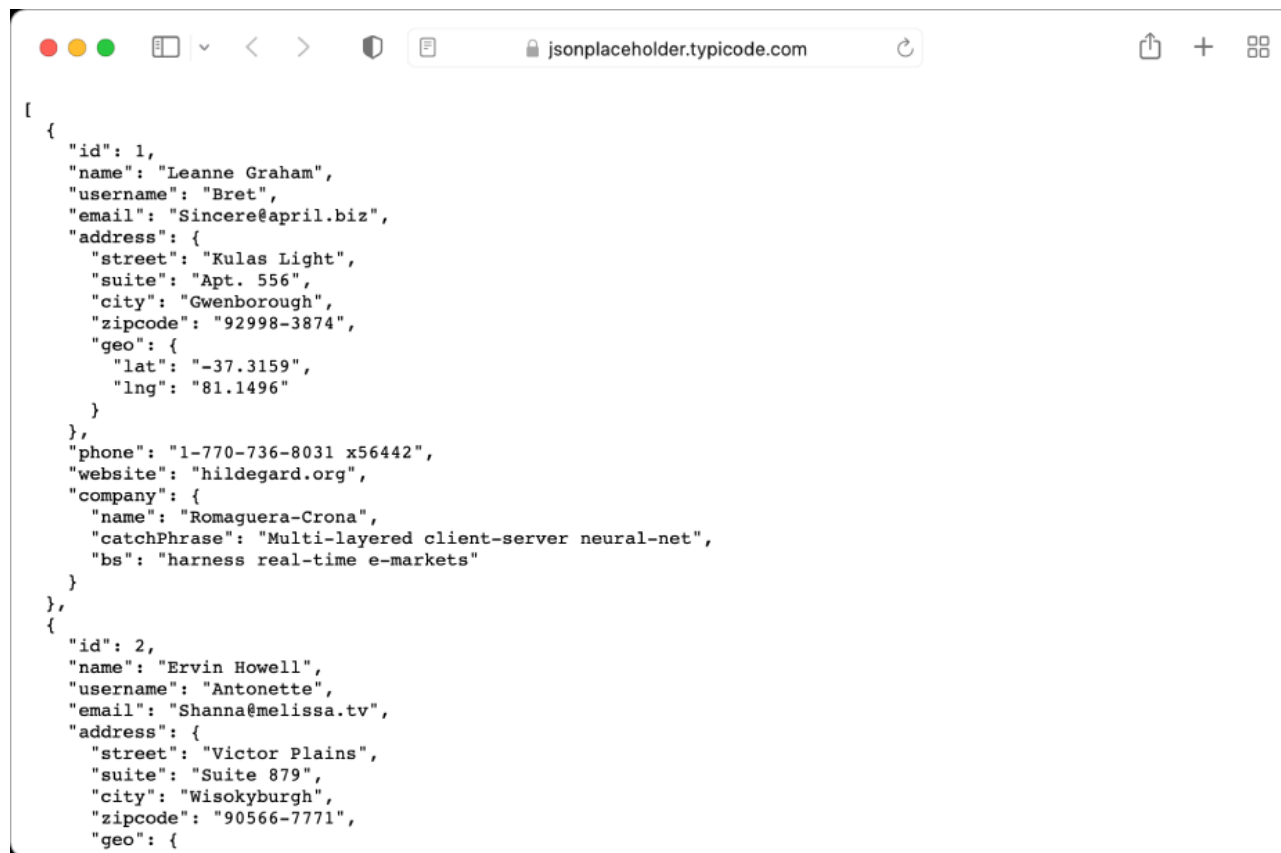
SPA – Single Page Application

Fluxo de uma SPA



Fluxo de uma SPA

JSON



```
{
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  },
  {
    "id": 2,
    "name": "Ervin Howell",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
    "address": {
      "street": "Victor Plains",
      "suite": "Suite 879",
      "city": "Wisokyburgh",
      "zipcode": "90566-7771",
      "geo": {
        "lat": "-43.9469",
        "lng": "-120.3345"
      }
    }
  }
}
```

Fluxo de uma SPA

- O conceito SPA (single page application ou aplicação de página única) foi pensado junto com Frameworks Javascript, pois havia uma necessidade de uma melhor performance na renderização.
- O fluxo de *multi-pages (tradicional)* cada página que precisa ser carregada se traduz em uma nova requisição para o servidor, nesta requisição o browser necessita do HTML, CSS e o JS para apresentar a tela ao usuário.
- Por isso, em várias aplicações web, vemos o browser levando um tempo de carregamento para ver a nova página, ou vemos a nova página ficar em branco inicialmente para ser carregada em seguida

Fluxo de uma SPA

- Já com o SPA não há a necessidade de se fazer requisições para carregamento de novas páginas.
- A aplicação seria “carregada” por inteiro na primeira requisição, onde todo o HTML, CSS e JS necessários seriam carregados de uma vez.
- A partir deste momento, quando novas páginas precisassem ser carregadas, estas seriam carregadas através de condições JS, retirando a necessidade de requisições para o servidor com a finalidade de obter o novo conteúdo a ser renderizado.

React

O que é?

Pra quê serve?



- React é uma biblioteca Javascript focada na criação de interfaces de usuário utilizando o conceito de Single Page Application (SPA).
- Com o React é possível ter uma melhor organização na criação de componentes visuais.

```
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.tsx</code> and save to reload.
12        </p>
13        <a
14          className="App-link"
15          href="https://reactjs.org"
16          target="_blank"
17          rel="noopener noreferrer"
18        >
19          Learn React
20        </a>
21      </header>
22    </div>
23  );
24 }
25
26 export default App;
```

O que é?

Pra quê serve?

JSX: uma extensão do JS muito semelhante ao HTML

```
1 import React from "react";
2 import logo from "./logo.svg";
3 import "./App.css";
4
5 function App() {
6   const userName = "Vinicius";
7
8   return (
9     <div className="App">
10       <header className="App-header">
11         <img src={logo} className="App-logo" alt="logo" />
12         <p style={{ color: "white", fontWeight: 700, fontSize: 20 }}>
13           {userName}
14         </p>
15       </header>
16     </div>
17   );
18 }
19
20 export default App;
21
```



Vinicius





Ambiente de desenvolvimento

Instalação



<https://nodejs.org/>

O que é NPM?

- NPM (Node Package Manager) é um gerenciador de pacotes para Node.js
- Muito útil para compartilhar ferramentas, instalar módulos e gerenciar diversas dependências em aplicações que usam Javascript.
- Existem outras opções para gerenciamento de pacotes (Ex: yarn, pnpm, etc.). No entanto, o NPM é o mais utilizado atualmente no mercado.





Primeiro Contato

Iniciando um projeto React

Iremos utilizar o Create React App: uma maneira muito conhecida para gerar uma aplicação React.
Iremos rodar os seguintes comandos no Terminal:



```
$ npx create-react-app my-first-app --template typescript
```

```
$ cd my-first-app
```

```
$ npm start
```

Revisão

Revisão

- Funcionamento da Web e dos navegadores
- Front-end / Back-end
- HTTP
- DevTools
- HTML
 - Atributos
 - Estrutura
 - Formulários
- SPA
- React
 - Overview
 - Create React App



Dúvidas?

