

**HTML+CSS+JS**



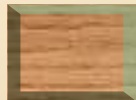
**Requisição Fetch**



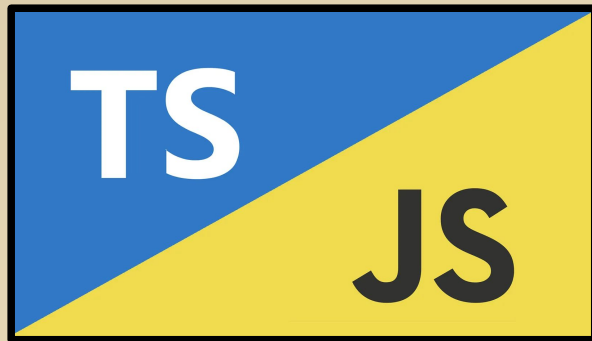
**Requisição Axios**



**REACT**



**DEBUG Requisição**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# JAVASCRIPT + HTML + CSS

HTML CSS JS

- Estrutura
  - HTML define a estrutura da página
  - CSS define o estilo visual
  - JavaScript manipula o comportamento e a lógica





# JAVASCRIPT + HTML + CSS

- Características:

- Manipulação direta do DOM  
(document.getElementById, innerHTML, etc.)
- Separação por arquivos: .html, .css, .js
- Funciona em qualquer navegador sem frameworks
- Pouca reutilização de código
- Difícil manutenção em aplicações grandes

JS  
CSS  
HTML





# JAVASCRIPT + HTML + CSS

- Limitações:

- Código mais difícil de escalar
- Falta de reatividade automática
  - DOM manual = mais propenso a erros
  - Você precisa interagir diretamente com o DOM usando funções como `getElementById`, `innerHTML`, `createElement`, etc.
  - Mudanças nos dados não se refletem automaticamente na interface. Você precisa atualizar o DOM

HTML  
CSS  
JS





# Introdução

## Ambiente

```
//mudar diretorio  
cd users\aluno\vscode  
//criar aplicação reatapp  
npx create-react-app pblc01-aula17
```

```
//abrir pasta no vscode  
c:\users\aluno\vscode\pblc01-aula16  
//iniciar o servidor react  
npm start  
//abrir browser  
http://localhost:3000
```



# Ambiente

```
▼ PBLC01-AULA16
  > node_modules
  > public
  ▼ src
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    logo.svg
    JS reportWebVitals.js
    JS setupTests.js
    .gitignore
    {} package-lock.json
    {} package.json
    ⓘ README.md
```

```
{
  "name": "pblc01-aula16",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/dom": "^10.4.0",
    "@testing-library/jest-dom": "^6.6.3",
    "@testing-library/react": "^16.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^19.1.0",
    "react-dom": "^19.1.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
}
```

```
import React from 'react';
import './App.css';
class App extends React.Component {
  handleSearch = async () => {
    const input = document.getElementById('pokemon-input');
    const container = document.getElementById('pokemon-result');
    const errorMsg = document.getElementById('error-message');
    const pokemonName = input.value.toLowerCase().trim();
    if (!pokemonName) return;
    try {
      const response = await
        fetch(`https://pokeapi.co/api/v2/pokemon/${pokemonName}`);
      if (!response.ok) {
        throw new Error('Pokémon não encontrado');
      }
      const pokemon = await response.json();
      errorMsg.textContent = '';
    }
  }
}
```

App.js

```

container.innerHTML =
  `

## Pokédex



App.js

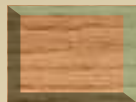

```



# App.css

```
.container {
  text-align: center;
  padding: 2rem;
  font-family: sans-serif;
}
input {
  padding: 0.5rem;
  font-size: 1rem;
}
button {
  padding: 0.5rem 1rem;
  margin-left: 1rem;
  font-size: 1rem;
  cursor: pointer;
}
.card {
  margin-top: 2rem;
  padding: 1rem;
  border: 2px solid #ddd;
  border-radius: 10px;
  background-color: #f8f8f8;
}
.card img {
  width: 150px;
}
.types {
  margin-top: 1rem;
}
```

```
.type {
  padding: 0.25rem 0.5rem;
  margin: 0.25rem;
  display: inline-block;
  border-radius: 5px;
  color: white;
}
.type.fire { background-color: #f08030; }
.type.water { background-color: #6890f0; }
.type.grass { background-color: #78c850; }
.type.electric { background-color: #f8d030; }
.type.ice { background-color: #98d8d8; }
.type.fighting { background-color: #c03028; }
.type.poison { background-color: #a040a0; }
.type.ground { background-color: #e0c068; }
.type.flying { background-color: #a890f0; }
.type.psychic { background-color: #f85888; }
.type.bug { background-color: #a8b820; }
.type.rock { background-color: #b8a038; }
.type.ghost { background-color: #705898; }
.type.dragon { background-color: #7038f8; }
.type.dark { background-color: #705848; }
.type.steel { background-color: #b8b8d0; }
.type.fairy { background-color: #ee99ac; }
.error {
  color: red;
  margin-top: 1rem;
}
```



**HTML+CSS+JS**



**Requisição Fetch**



**Requisição Axios**



**REACT**



**DEBUG Requisição**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano

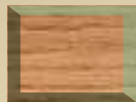


# Requisição com Fetch

- API moderna do JavaScript para fazer requisições HTTP
- Já incluído em todos os navegadores modernos
- Baseado em Promises
- Limitações:
  - Precisa chamar `.json()` manualmente para ler os dados
  - Não lança erro automaticamente em status HTTP como 404/500
  - Sem suporte embutido para interceptadores, timeout, cancelamento etc.

HTML  
CSS  
JS

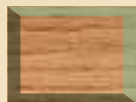




**HTML+CSS+JS**



**Requisição Fetch**



**Requisição Axios**



**REACT**



**DEBUG Requisição**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



HTML CSS JS

# Requisição com Axios

- Biblioteca externa para fazer requisições HTTP
- Baseado em Promises, como o fetch
- Oferece funcionalidades extras para facilitar o desenvolvimento
- Converte JSON automaticamente (res.data)
- Lança erro para status HTTP automaticamente
- Suporte nativo a:
  - Interceptadores de request/response
  - Timeout e cancelamento
  - Headers customizados com facilidade





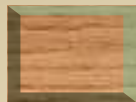
React

# Requisição com Axios

```
//instalar a dependência  
npm install axios
```

```
//ajustar chamada da api no App.js  
const response = await  
  axios.get(`https://pokeapi.co/api/v2/pokemon/${pokemonName}`)  
;  
const pokemon = response.data;
```

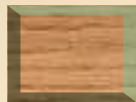




**HTML+CSS+JS**



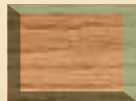
**Requisição Fetch**



**Requisição Axios**



**REACT**



**DEBUG Requisição**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



React

# REACT

- Características
  - Componentes reutilizáveis (function App() { ... })
  - Reatividade automática com useState, useEffect, etc.
  - DOM é gerenciado pelo Virtual DOM
  - Estilo pode ser local ou global







# REACT

React

- Vantagens:
  - Código mais organizado e modular
  - Atualizações automáticas da interface
  - Melhor performance com Virtual DOM
  - Fácil integração com APIs modernas



```
import React, { useState } from 'react';
import axios from 'axios';
import './App.css';
function AppReact() {
  const [pokemonName, setPokemonName] = useState('');
  const [pokemonData, setPokemonData] = useState(null);
  const [error, setError] = useState('');
  const handleSearch = async () => {
    if (!pokemonName.trim()) return;
    try {
      const response = await
axios.get(`https://pokeapi.co/api/v2/pokemon/${pokemonName.toLowerCase()}`);
      setPokemonData(response.data);
      setError('');
    } catch (err) {
      setPokemonData(null);
      setError('Pokémon não encontrado!');
    }
  };
};
```

# AppReact.js

```

return (
  <div className="container">
    <h1>Pokédex (React Moderno)</h1>
    <input
      type="text"
      value={pokemonName}
      onChange={(e) => setPokemonName(e.target.value)}
      placeholder="Digite o nome do Pokémon"/>
    <button onClick={handleSearch}>Buscar</button>
    {error && <p className="error">{error}</p>}
    {pokemonData && (
      <div className="card">
        <h2>{pokemonData.name.toUpperCase()}</h2>
        <img src={pokemonData.sprites.front_default} alt={pokemonData.name} />
        <div className="types">
          {pokemonData.types.map((typeObj) => (
            <span key={typeObj.type.name} className={`type ${typeObj.type.name}`}>
              {typeObj.type.name}
            </span>
          ))}
        </div>
      </div>
    )}
  </div>
);
export default AppReact;

```

## Pokédex (React Moderno)



# AppReact.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import AppReact from './AppReact';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <div className="comparison-container">
      <div className="column">
        <h2>HTML + JS Puro</h2>
        <App />
      </div>
      <div className="column">
        <h2>React Moderno</h2>
        <AppReact />
      </div>
    </div>
  </React.StrictMode>
);
```



# Index.js



# React



```
.comparison-container {  
  display: flex;  
  justify-content: space-between;  
  align-items: flex-start;  
  gap: 2rem;  
  padding: 2rem;  
}  
  
.column {  
  flex: 1;  
  min-width: 300px;  
  border: 1px solid #ddd;  
  border-radius: 8px;  
  padding: 1rem;  
  background: #f9f9f9;  
}  
  
.column h2 {  
  text-align: center;  
  font-size: 1.25rem;  
  margin-bottom: 1rem;  
}
```

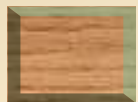
## Index.css



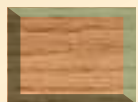
**HTML+CSS+JS**



**Requisição Fetch**



**Requisição Axios**



**REACT**



**DEBUG Requisição**



**HTML e CSS**

Prof. Enzo  
Seraphim

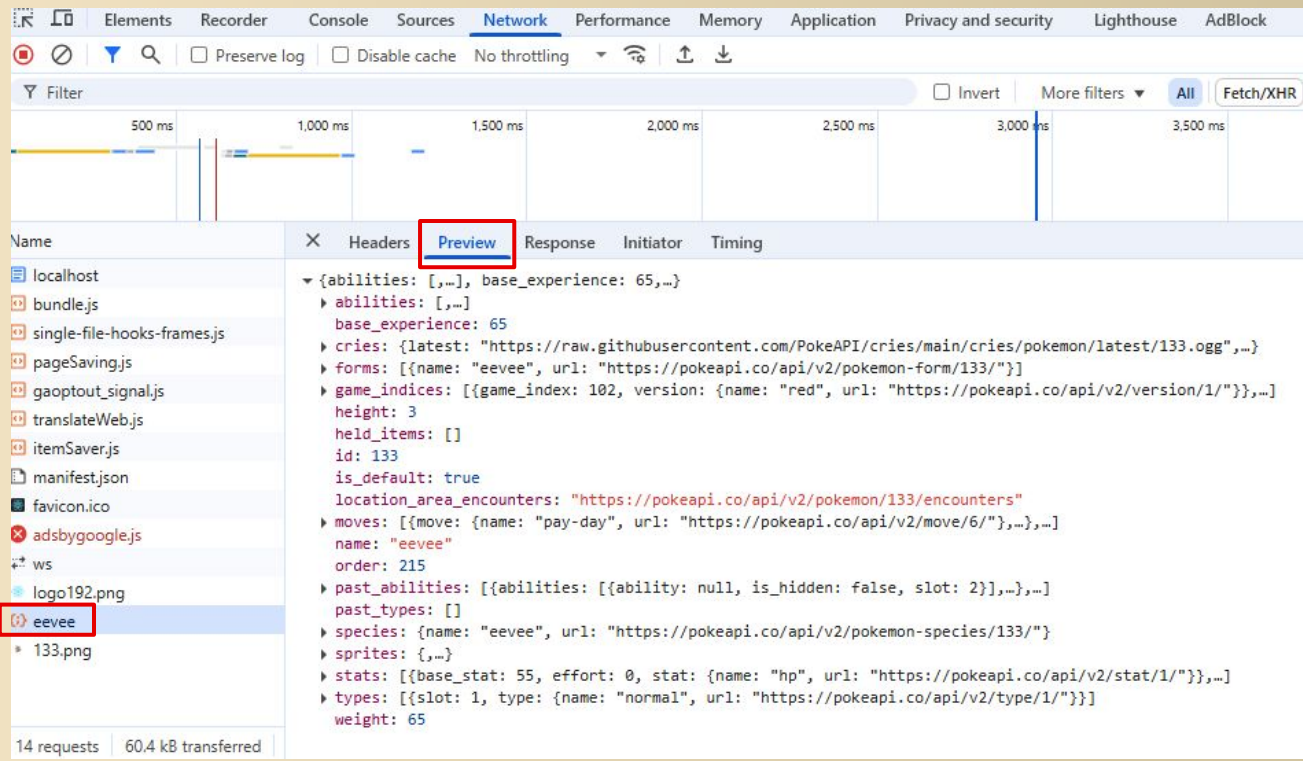
Profa. Bárbara  
Pimenta Caetano



# Para debugar

- `console.log(response.data)`
- `console.table(response.data)`

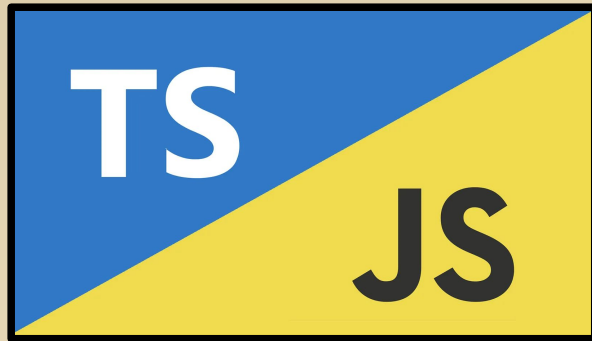
OU F12:



**Prof. Enzo  
Seraphim**

**Profa. Bárbara  
Pimenta Caetano**

Os logotipos, marcas comerciais e nomes de produtos citados nesta publicação tem apenas o propósito de identificação e podem ser marcas registradas de suas respectivas companhias.



**JSON Web Token TS**