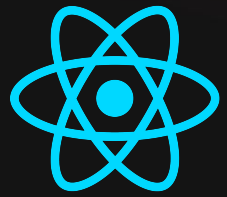


React

Criando aplicações frontend

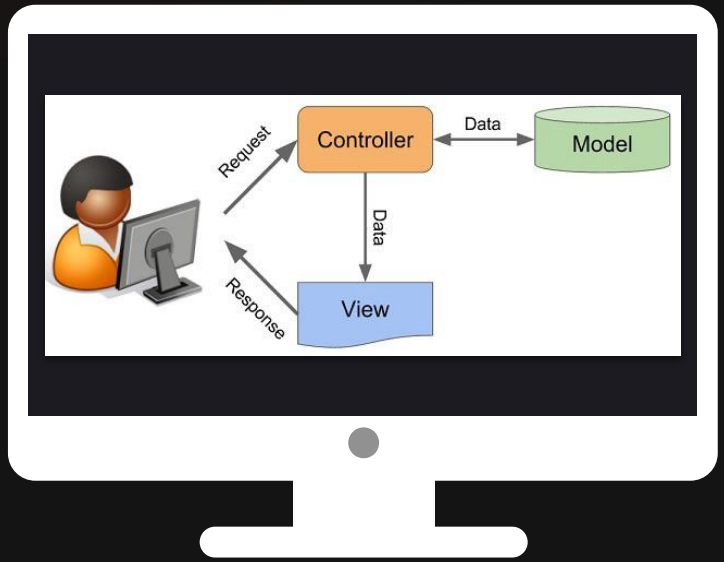
Jefferson Lima

Frontend developer

TABLE OF CONTENTS

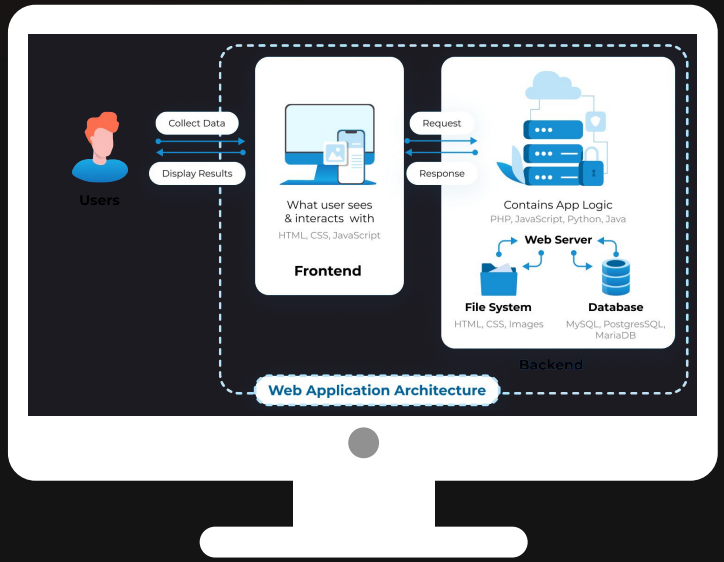
Arquitetura de aplicação

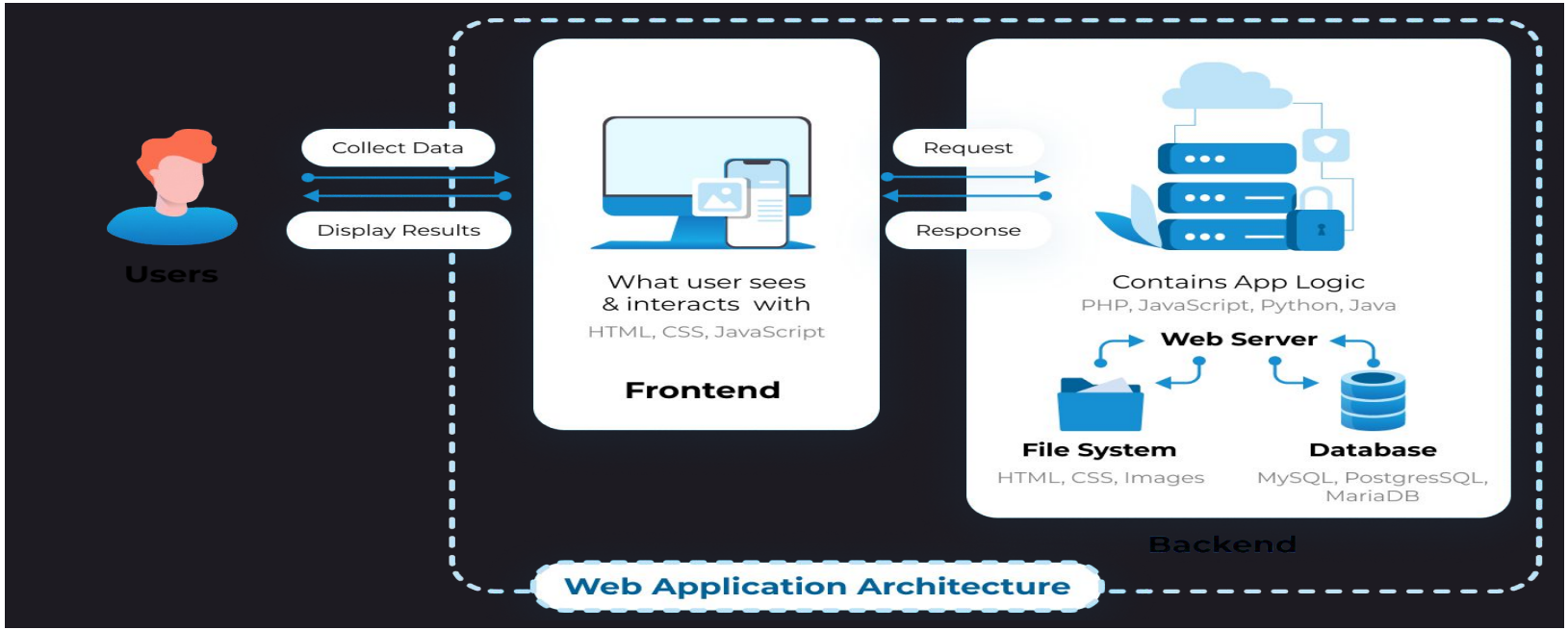
Existem diversos tipos de arquitetura de software que servem para aplicações/contextos específicos



Arquitetura de aplicação

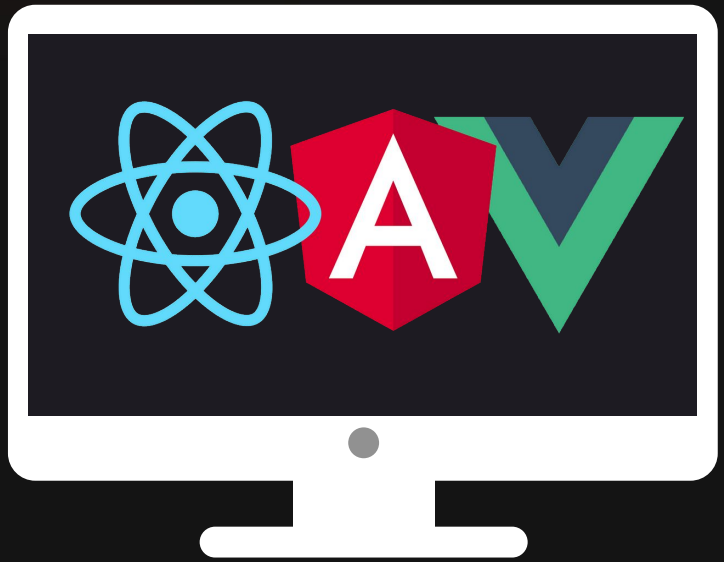
Atualmente em aplicações web, é comum haver separação entre a camada de apresentação (frontend) e as demais camadas (backend)




Users
Collect Data
Display Results
What user sees
& interacts with
HTML, CSS, JavaScript
Frontend
Request
Response
Contains App Logic
PHP, JavaScript, Python, Java
Web Server
File System
HTML, CSS, Images
Database
MySQL, PostgresSQL,
MariaDB
Backend
Web Application Architecture

Frontend

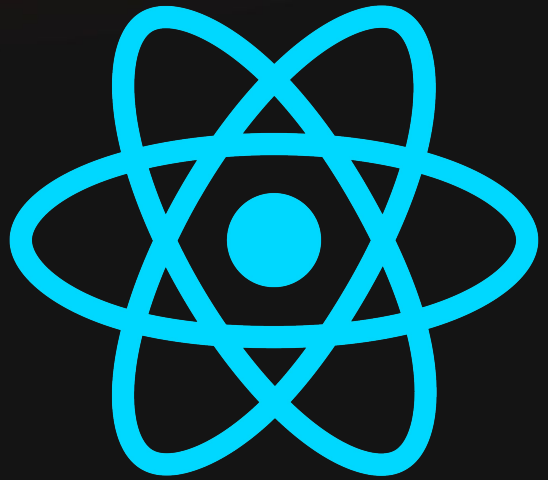
No mercado existem várias ferramentas (frameworks e bibliotecas) para se desenvolver frontend



React

O que é o React?

O reactJS é uma biblioteca Javascript criada pelo Facebook para desenvolvimento de aplicações frontend

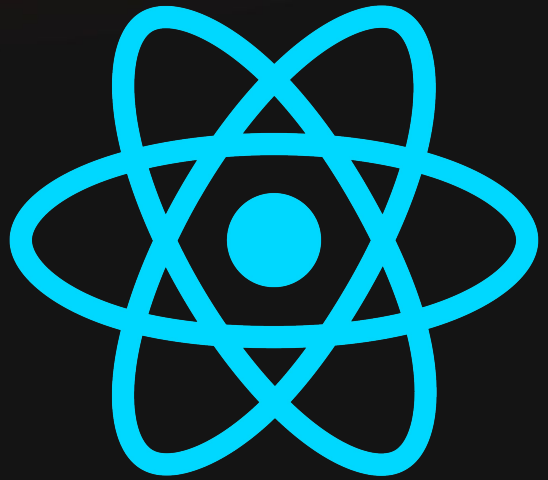


React

Por que react?

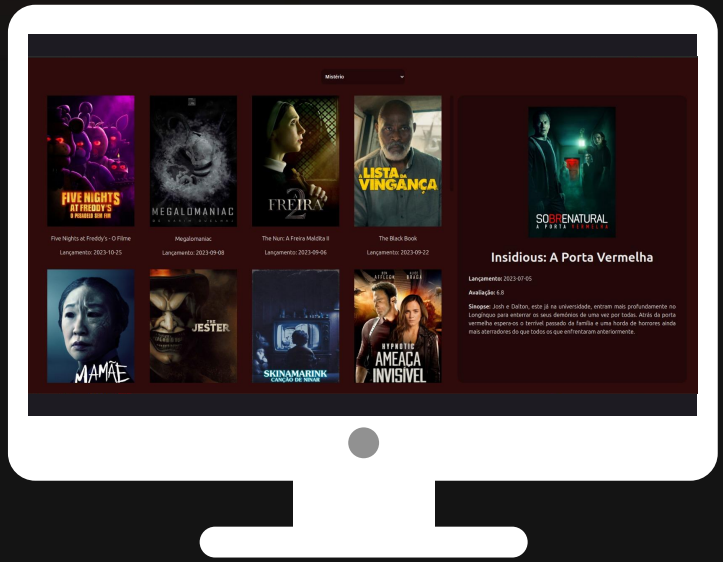
ReactJS tem como premissa trazer soluções para:

- Gerenciamento de estado
- Componentização e reuso de código
- Ligação entre dados e elementos



Mãos no código

Iremos desenvolver uma catálogo de filmes, consumindo a api do **themoviedb.org**



Mãos no código - Criando o projeto

Existem alguns frameworks react que podemos utilizar para a trabalhar com React, para o nosso exemplos utilizaremos o vite. Siga o passo a passo

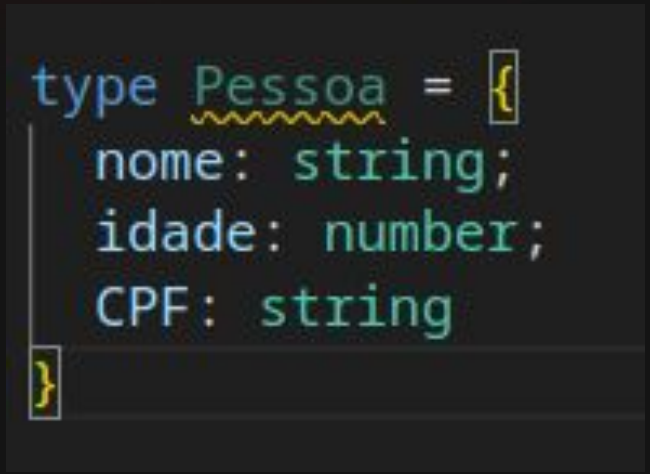
- Abra o prompt de comando e digite o seguinte comando: npm create vite@latest e pressione enter
- Pressione enter novamente na tela de confirmação
- Dê nome ao seu projeto
- Na seleção de framework, selecione a opção React
- No passo seguinte, selecione a opção typescript
- Seu projeto está criado, entre na pasta do projeto utilizando o comando: cd <Nome da pasta>
- Execute o comando para abrir o seu projeto no vscode: code .
- Execute o comando npm install
- Para executar nosso projeto, utilize o comando: npm run dev

Mãos no código - Entendendo o projeto

O comando executado criou os arquivos necessários para começarmos a trabalhar e nos deu uma página de exemplo, vamos entender alguns desses arquivos

Typescript

Typescript é um superset da linguagem Javascript que provê tipagem ao JS



JSX

JSX é a forma que o react utiliza para escrever HTML, permitindo “misturar” HTML e Javascript

```
<button onClick={() => setCount((count) => count + 1)}>
  count is {count}
</button>
```

Gerenciamento de estado

Em react, o estado contém todos os dados de um componente. O estado pode mudar com o tempo ou com as ações do usuário na página. O componente é remontado cada vez que o estado muda.

Para trabalhar com estado, utilizamos o hook useState, que nos provê uma variável e uma função para alterar o estado

```
const [count, setCount] = useState(0)
```

Hooks

Hooks são funções que permitem gerenciar estado e trabalhar com os ciclos de renderização do react.

- Sempre começam com a palavra ‘use’
- O react possui uma série de hooks, sendo os principais o useState e o useEffect
- O desenvolvedor também pode criar seus próprios hooks

useState

Hook que adiciona uma variável de estado ao seu componente

Sintaxe:

[example, setExample] = useState(“Valor inicial”)

- example é o nome na nossa variável
- setExample é a função que devemos chamar para alterar essa variável

useEffect

useEffect é um hook que permite trabalhar executar funções de acordo com o ciclo renderização de um componente. Com ele é possível por exemplo executar um comando a cada vez que um estado for alterado ou na primeira vez que o componente é carregado, por exemplo

Trabalhando com APIs

No desenvolvimento frontend é muito comum se trabalhar com APIs, seja enviando dados (formulários de cadastro, posts, curtidas) e consumindo dados

É possível consumir APIs de forma nativa, utilizando funções do próprio javascript ou utilizar bibliotecas prontas que facilitam o trabalho (axios, react-query)

Para trabalhar com APIs, é necessário entender alguns conceitos de comunicação cliente-servidor

No nosso projeto, iremos utilizar a API do https://developer.themoviedb.org/docs

HTTP

- HTTP (Hypertext Transfer Protocol) é a forma que a internet utiliza para comunicação entre cliente e servidor
- O cliente inicia a comunicação, enviando uma requisição (request) e recebe uma resposta (response)
- A requisição/resposta contém cabeçalhos (headers) que possuem informações adicionais sobre a mesma
- A requisição possui obrigatoriamente um método:
  - GET - Utilizado para buscar dados
  - POST - Utilizado para enviar dados
  - PUT - Utilizado para alterar dados
  - DELETE - Utilizado para deletar dados
- Requisições também podem conter dados ou url params

Axios

Axios é uma biblioteca Javascript para trabalhar com o protocolo HTTP

Para instalar, vamos executar o seguinte comando: npm install axios

Vamos configurar o axios, passando para ele a chave da nossa API - disponível em : https://acesse.one/reactifrn