CENTRO UNIVERSITÁRIO CARIOCA

CIÊNCIA DA COMPUTAÇÃO

RAPHAEL FIGUEIRA LOPES
HERBERT DE OLIVEIRA VELOSO

Desenvolvimento front-end em React

Rio de Janeiro

2023

RAPHAEL FIGUEIRA LOPES
HERBERT DE OLIVEIRA VELOSO

Desenvolvimento front-end em React

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Centro Universitário Carioca, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Alberto Tavares da Silva

Rio de Janeiro

2023

L864d Lopes, Raphael Figueira.
Didática de Desenvolvimento web em React / Raphael Figueira Lopes, Herbert de Oliveira Veloso – Rio de Janeiro, 2023.
38 f.

Orientador: Alberto Tavares da Silva e André Luiz Avelino Sobral.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário UniCarioca, Rio de Janeiro, 2023.

1. Desenvolvendo web. 2. Básico da linguagem web. 3. React na Prática. 4. Estudo de Caso. I. Veloso, Herbert de Oliveira. II. Silva, Alberto Tavares da. III. Sobral, André Luiz Avelino. IV. Título.

CDD 005

RAPHAEL FIGUEIRA LOPES
HERBERT DE OLIVEIRA VELOSO

TEMA: Desenvolvimento front-end em React

Banca Examinadora

_____
Prof. Alberto Tavares da Silva, D.Sc. - Orientador
Centro Universitário Carioca

_____
Prof. Sérgio Assunção Monteiro D.Sc. - Professor Convidado
Centro Universitário Carioca

_____
Prof. Manuel Martins Filho, D.Sc. - Professor Convidado
Centro Universitário Carioca

Rio de Janeiro

2023

AGRADECIMENTOS

Agradecemos primeiro a Deus por nos ter mantido na trilha certa durante esteprojeto de pesquisa com saúde e forças para chegar até o final. Agradecemos à nossa Universidade UniCarioca por nos proporcionar umambiente criativo e amigável para os estudos. Aos professores, pelas correções e ensinamentos que nos permitiram apresentar um melhor desempenho no nosso processo de formação profissional aolongo do curso. Agradecemos especialmente ao nosso professor e orientador Alberto Tavares da Silva por aceitar conduzir o nosso trabalho de pesquisa. Obrigado poresclarecer tantas dúvidas e ser tão atencioso e paciente. Agradecemos aos familiares e amigos que nos apoiaram ao longo da nossa jornada acadêmica, e principalmente a nossas mães que sempre estiveram ao nossolado nessa difícil caminhada.

DEDICATÓRIA

Dedico este trabalho aos nossos familiares, que nos motivara em toda caminhada academica, que sempre acreditaram em nossos esforços e tenho certeza de que estão orgulhosos ao acompanhar este trabalho. As palavras deles sempre serviram de motivação para chegarmos onde chegamos.

RESUMO

Neste trabalho serão apresentados os conceitos de React, passo a passo, desde a instalação do Node ao desenvolvimento de um site, com o adicional de um jogo em JavaScript. Utilizando o editor de códigos Visual Studio Code, a biblioteca React e React.js (react-dom) para sua implementação.

Palavras-chave: Desenvolvendo web, React, JavaScript e Projeto prático.

ABSTRACT

This work will present the concepts of React, step by step, from the installation of Node to the development of a website, with the addition of a game in JavaScript. Using the Visual Studio Code editor, the React library and the React.js framework for its implementation.

Key Words: Developing web, React, JavaScript and Practical project.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

1 INTRODUÇÃO

Com o crescimento do uso da *World Wide Web* (www) a partir da década de 90, começou a ser necessário novas técnicas de desenvolvimento Web. Uma dessas técnicas que surgiram é o *Single Page Application* (SPA), que vamos abordar através do React. O React teve seu início em 2011, criado por programadores do Facebook com a necessidade de um código altamente flexível e escalável.

Este trabalho tem como objetivo apresentar uma solução didática para o aprendizado eficaz da biblioteca React, por meio da criação de um jogo em JavaScript que utiliza essa tecnologia. A ideia é difundir o uso de React no dia a dia dos desenvolvedores web que estão estudando ou trabalhando na área.

A difusão de uma tecnologia no mercado pode trazer diversas vantagens para os futuros desenvolvedores. Uma delas é a possibilidade de solucionar problemas de forma mais simplificada, pois, com mais pessoas utilizando a tecnologia, a busca por soluções na internet se torna mais fácil e rápida. Além disso, a difusão de uma tecnologia pode aumentar sua usabilidade no mercado, o que pode torná-la mais valorizada e requisitada pelas empresas.

Um bom exemplo disso é a persistência do uso da linguagem de programação Java no mercado. Devido à sua ampla difusão e ao seu alto nível de usabilidade, o Java é uma das linguagens mais utilizadas na área de desenvolvimento de software. Isso significa que profissionais que dominam essa tecnologia possuem uma vantagem competitiva em relação aos demais.

Por isso, a criação de uma solução didática para o aprendizado de React pode ser extremamente benéfica para os desenvolvedores que estão buscando se destacar no mercado. Além disso, a criação de um jogo em JavaScript pode tornar o aprendizado mais dinâmico e divertido, o que pode aumentar o interesse dos estudantes pela tecnologia.

Ao longo deste trabalho, serão apresentados os principais conceitos de React, bem como as etapas de desenvolvimento do jogo em JavaScript. Também serão discutidos os benefícios da utilização dessa tecnologia no dia a dia dos desenvolvedores web, tanto do ponto de vista técnico quanto profissional. Ao final, espera-se que este trabalho possa contribuir para a difusão de React no mercado e para o sucesso dos futuros desenvolvedores que utilizam essa tecnologia.

1.1 Objetivo do Projeto

O objetivo deste projeto é desenvolver uma solução didática para o aprendizado eficaz de React, por meio da criação de um jogo em JavaScript que utilize essa tecnologia. A proposta é que essa solução facilite o aprendizado de React e aumente a sua difusão no mercado de desenvolvimento web, contribuindo para que os futuros desenvolvedores possam solucionar problemas de forma mais simplificada com uso dessa tecnologia.

Para atingir esse objetivo, serão realizados estudos teóricos sobre a biblioteca React e o desenvolvimento de jogos em JavaScript. A partir disso, será desenvolvido um jogo didático que permita aos estudantes experimentar na prática os conceitos aprendidos sobre React.

Espera-se que esse projeto possa contribuir para a formação de profissionais qualificados em React e para a difusão dessa tecnologia no mercado de desenvolvimento web, aumentando a sua usabilidade e consolidando a sua presença como uma das principais bibliotecas utilizadas pelos desenvolvedores.

1.2 Justificativa da Escolha do Tema

Queremos abordar a importância da difusão da biblioteca React no mercado Front-end. Para isso é necessidade de uma solução didática que facilite o aprendizado dessa tecnologia. A criação de um jogo em JavaScript que utiliza React pode tornar o aprendizado mais dinâmico e divertido, aumentando o interesse dos estudantes pela tecnologia. Além disso, a difusão de uma tecnologia no mercado pode trazer diversas vantagens para os futuros desenvolvedores, como a possibilidade de solucionar problemas de forma mais simplificada e a permanecia do seu uso no mercado.

1.3 Estrutura do Trabalho

O trabalho apresenta em sequência as tecnologias biblioteca React e suas principais ferramentas. No capítulo 2 é apresentado conceitos da biblioteca React, mostrando a informação por trás do uso do React. Neste capítulo ainda é apresentado conceitos básicos da programação front-end e as principais aplicações.

No capítulo 3 é apresentado as principais ferramentas utlizadas par montar o projeto, entre elas o Node.js, npm, vite, visual code studio e os pacotes utilizados.

No capítulo 4 o estudo de caso e os diagramas de classes UML no qual o projeto se baseia. Finalmente o capítulo 5 com conclusão do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta e descreve os conceitos básicos sobre desenvolvendo web, para melhor compreensão do projeto.

2.1 Desenvolvimento web

Desde o início da web e o seu crescimento exponencial cujo seus primeiros sites constituídos com apenas a linguagem de marcações chamada HTML, grande parte deles só apresentavam conteúdo estático, como imagens e textos. Com o passar dos anos foram surgindo outras tecnologias para contribuir na construção de novos desenvolvimentos de websites.

O ambiente web hoje em dia abandonou a simplicidade e adotou uma plataforma de aplicações e com isso tivemos diversas formas de sistemas web existentes, tornando o desenvolvimento de tal muito mais complexo. Com o aumento nos números de usuários e de serviços existentes, a exigência por aplicações melhores e mais robustas foram aumentados exponencialmente, assim exigindo cada vez mais profissionais no desenvolvimento das aplicações web.

2.2 Básico sobre desenvolvimento web

A *World Wide Web* (www), mais conhecida como web, é um sistema de documentos que permite o acesso ao conteúdo em forma de hipertexto, que por sua vez está disposta na Internet, a rede mundial de computadores.

A Internet faz uso do modelo *Transmission Control Protocol/ Internet Protocol* (TCP/IP) para fazer comunicação na rede, logo a web por estar disposta na Internet faz uso de alguns seus recursos.

A web é baseada na arquitetura cliente-servidor, sendo assim, uma máquina hospeda os arquivos de um site e responde toda vez que algum deles é solicitado, essa máquina chamamos de servidor, e qualquer máquina que solicita algum desses arquivos é chamada de cliente. Na maioria das vezes, esta solicitação é feita através de uma requisição HTTP a partir de um navegador no computador ou celular. Quando conexão com o servidor é aceita, é realizada a execução do protocolo TCP, que possui a garantia que os pacotes vão ser entregues e apresentados ao cliente de forma organizada e sem alterações.

Figura 1 - Conexão cliente ao servidor

Fonte: http://high5devs.com/2015/05/aplicacao-web-http/

2.2.1 Endereço IP e portas

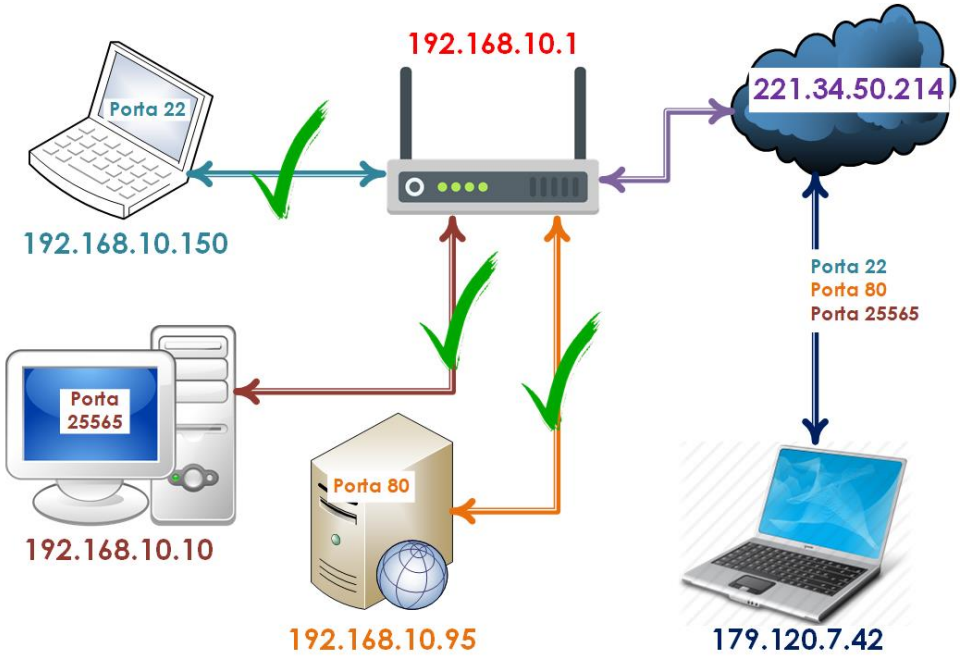
Toda máquina conectada à rede mundial de computadores possui um endereço um endereço *Internet Protoco*l (IP) sendo estruturado somente por números, há também um grande número de portas nas quais as aplicações e processos fazem a comunicação da máquina.

As portas podem denominar algum tipo de serviço, exemplo disso são as aplicações que realizam a entrega de e-mail usando porta 25 utilizando-se do protocolo *Simple Mail Transfer Protocol* (SMTP), ou então o serviço de transferência de arquivos que utiliza a porta 21 utilizando o protocolo *File Transfer Protocol* (FTP), por último e mais importante a porta 80 para serviços de entrega de páginas web

2.2.2 DNS

O servido *Domain Name System* (DNS) é responsável por associar um *Unifor Resource Locator* (URL) a um endereço IP em que está localizado os documentos do site acessado pela URL, assim todas as máquinas ligadas uma rede é configurada para o acessar servidor DNS.

Figura 2 - Endereço IP e portas



Fonte: https://www.iperiusbackup.net/pt-br/aprenda-a-redirecionar-portas-para-a-internet/

2.2.3 Protocolo HTTP e o HTML

Através da padronização do uso do *HyperText Transfer Protocol* (HTTP) fazemos a comunicação dos servidores e os browsers(cliente). Ele também é responsável por especificar o que o cliente pode enviar e o servidores responderão sendo obedecida o protocolo (TANENBAUM, 2003). Tanenbaum (2003) e o modelo referência Open *System Interconnection* (OSI) sendo adotado para analisar projetos de protocolos, assim admitindo a interconexão de sistemas não são fechados, permite comunicação sistemas abertos e/ou proprietários.

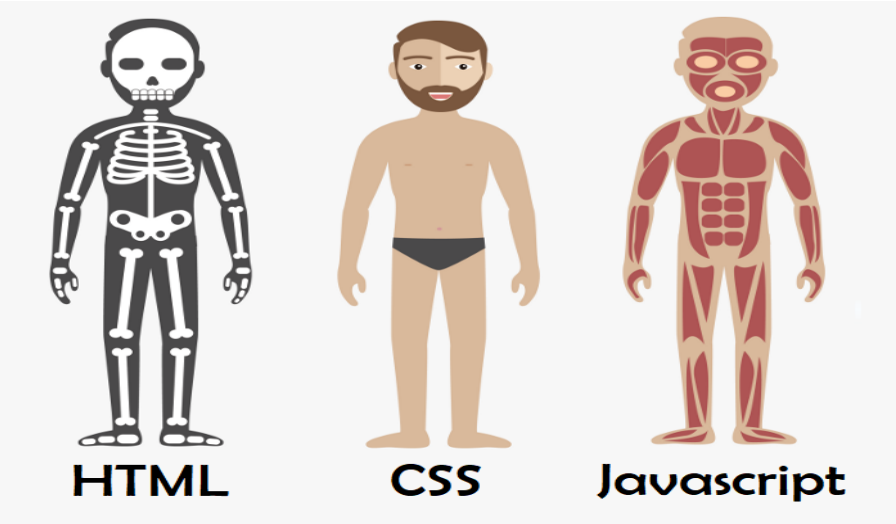
O *HyperText Markup Language* (HTML) e a uma linguagem utilizada para marcação de web, não confunda com programação. Quando cliente recebe uma resposta do servidor por suas requisições, e disponibilizado um documento HTML, então o browser fica responsável por analisar o conteúdo dele, para realizar novas requisições para exibir a página web, adquirindo todos os recursos a página web é exibida.

2.2.4 CSS e JavaScript

Atualmente o HTML faz parte do conjunto de tecnologias essenciais para desenvolvimento de websites, sendo formadas por *Cascading Style Sheet* (CSS) e Javascript. A linguagem de estilo (CSS) é usada para formato da exibição, ou seja, a estética do site.

Já a linguagem de script (Javascript) é usada para interações dentro do site. É necessário saber que esses 3 elementos são necessários para montar um site, porém para o desenvolvimento de sistemas complexos precisaremos aumentar a complexidade, com isso entramos nos conceitos de Front-end e o Back-end.

Figura 3 – Abstração da estrutura com HTML, CSS E JavaScript



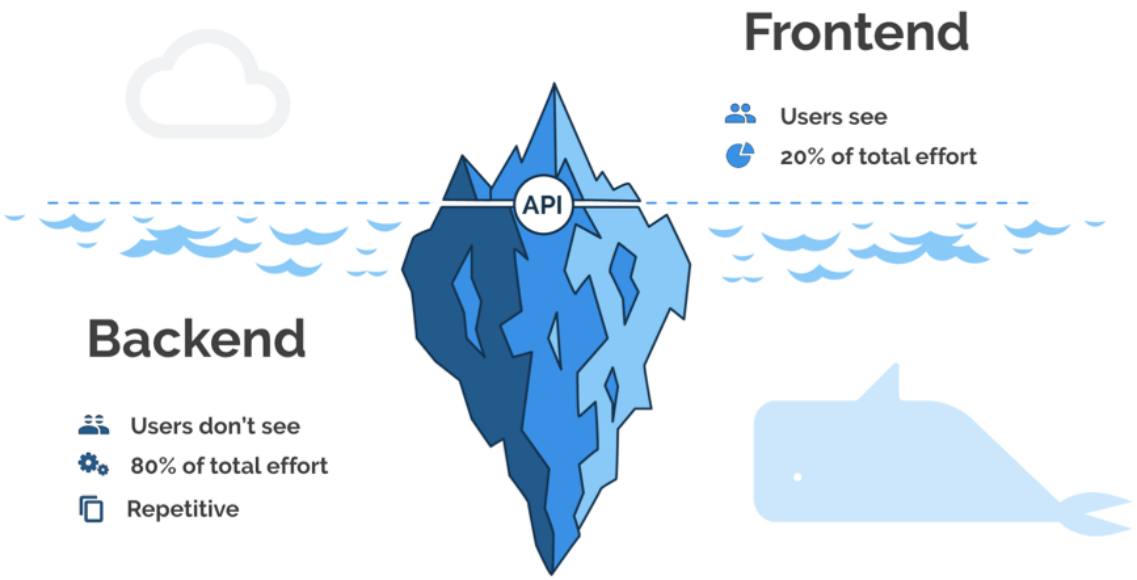
Fonte: https: https://www.alura.com.br/artigos/html-css-e-js-definicoes

2.2.5 Front-end e Back-end

Front-end é responsável pela interface que é exibida para o usuário e toda a sua interatividade, desde animações a coleta de dados e ações, essa e a parte geralmente e imaginada por profissionais de UI e UX para melhor experiência do usuário. Os componentes mais utilizados para este fim são HTML, CSS, JavaScript.

Back-end é responsável pelo trabalho do sistema que não é visível pelo usuário, ou seja, por tratar dos dados captados pelo front-end, lidar com o banco de dados e garantir as regras de negócio da aplicação. Está área trata principalmente banco de dados e linguagens de programação, como PHP, Java, Python e JavaScript (Node), entre outras questões como segurança da informação.

Figura 4 – Diferenças entre Front-end e Back-end



Fonte: https://blog.back4app.com/pt/lista-de-ferramentas-para-backend/

2.3 React

Com seu surgimento em meados de 2011 os engenheiros de software do Facebook criaram o React para uso exclusivo, mas em 2013 foi lançado como uma ferramenta JavaScript open source. React é uma biblioteca JavaScript de código aberto para *interfaces gráficas do usuário* (GUI), melhorando não só a experiência do usuário, como facilitando a vida dos desenvolvedores de interfaces do usuário (UI), o React cria uma virtualização da *Document Object Model* (DOM) do HTML para otimizar a manipulação de elementos exibidos, conceito chamado de Virtual DOM.

Uma das grandes dificuldades da equipe foi com performance, o tamanho das aplicações, previsibilidade e a segurança feitas nos sites, com isso a equipe de engenharias desenvolveu One-way data binding, sendo esse uma tecnologia que permitia mudar ou implementar dados sem alterar o sem alterar o existente assim separando o model e a view de modo seguro, para que o estado sofra alteração e preciso usar função setState.

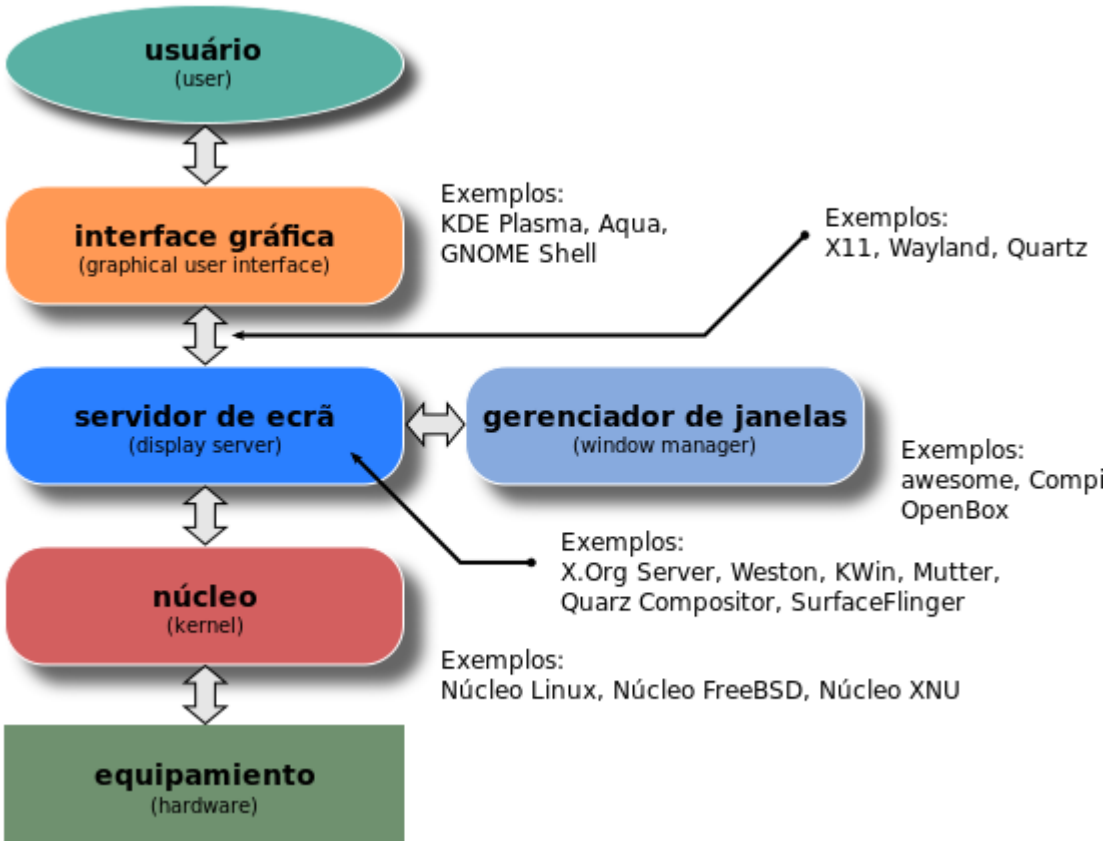
Uma tecnologia que deve ser mencionada ligada a React e o Search Engine Optimization (SEO), ferramenta responsável pelos motores de busca.

2.3.1 GUI

A interfaces gráficas do usuário (GUI) no react, pode ser mais bem compreendido como componente visual interativo que apresenta um comportamento em relação as interações do usuário como interagir com ícones mudando de cor, tamanho entre outras ações.

Essa tecnologia foi desenvolvida em 1981 pela empresa Xerox PARC pelos desenvolvedores Alan Kay, Douglas Engelbart e outros pesquisadores com pequenas colaborações.

Figura 5 – Esquema de Camadas das interfaces gráficas



Fonte: https://pt.wikipedia.org/

2.3.2 DOM

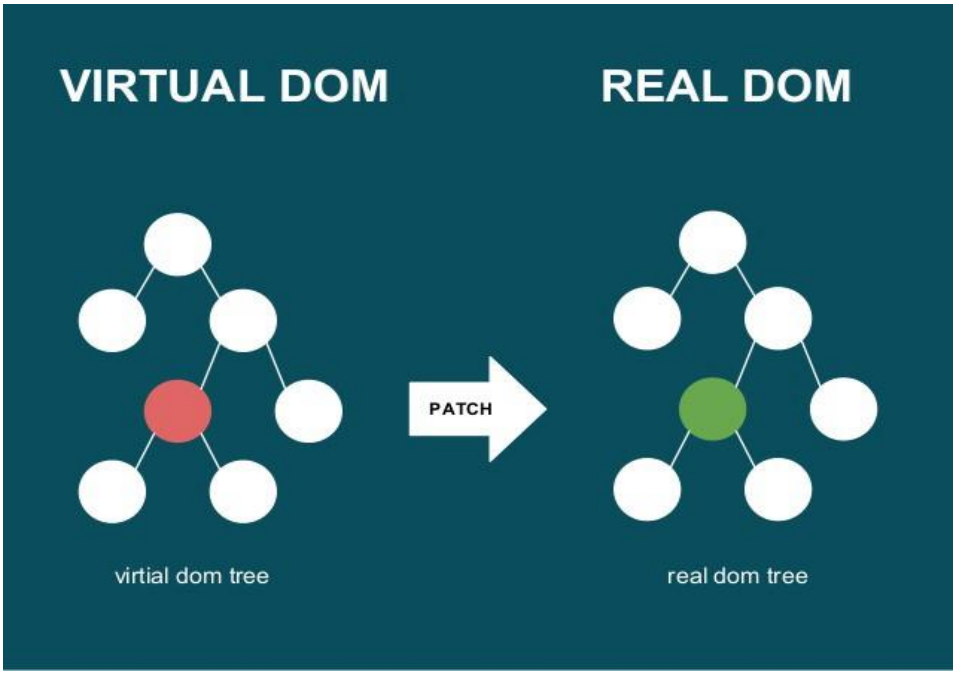
O *Document Object Model* (DOM) teve seu início quando W3C (*World Wide Web*) convenceu a Netacape e Microsoft a criarem uma liguem padronizada em 1997 criando a ECMAScript, quem no futuro seria mais conhecido como nosso querido JavaScript.

DOM e conhecido por representar uma página web no formato de um conceito conhecido como árvore, ele também é uma interface de programação para documentos HTML e *eXtensible Markup Language* (XML), desenvolvimento front-end moderno, sua alteração é feita com o uso de JavaScript, quando alterado pode levar mudanças na página web junto e é uma das operações extremamente lenta. Com a criação do react ouve a necessidade de melhorar a experiência do uso do DOM criando o Virtual DOM.

2.3.2 Virtual DOM

Virtual DOM foi criado pelo Facebook, com uso do react nessa tecnologia ele copia todos os nodes do DOM que replicam o código em Javascript. Com isso as aplicações que eram lentas, por serem feitas no DOM "original" passaram serem feitas no Virtual DOM que são rápidas, pois quando as alterações são feitas no Virtual DOM e são aplicadas somente quando necessárias, ou seja, elas são repassadas para o DOM "Original.".

Figura 6 – Arvore de Virtual DOM de implementação



Fonte: https://tableless.com.br/guia-completo-react-ecossistema/

2.3.3 Lifecycle

Lifecycles não é nada mais do que criação de componentes mais complexos e alguns métodos adicionados na (interface de programação de aplicação) API dos componentes React, assim podendo controlar quando um componente vai ser criando, destruído, atualizado e outros eventos a mais. Temos os seguintes métodos de lifecycle de forma simplificada a seguir:

O *componentWillMount* é pouco usado no dia a dia, seu uso mais comum e no construtor de classe, ele e executado antes do primeiro render. O render por sua vez tem a função de fazer a exibição do código HTML.

O *componentDidMount* executado logo após o primeiro render, sendo o método mais usado, podendo ser usado até mesmo na manipulação do DOM.

O *componentWillReceiveProps* é executado quando as props do componente são atualizadas, geralmente usados quando precisam reagir aos eventos externos.

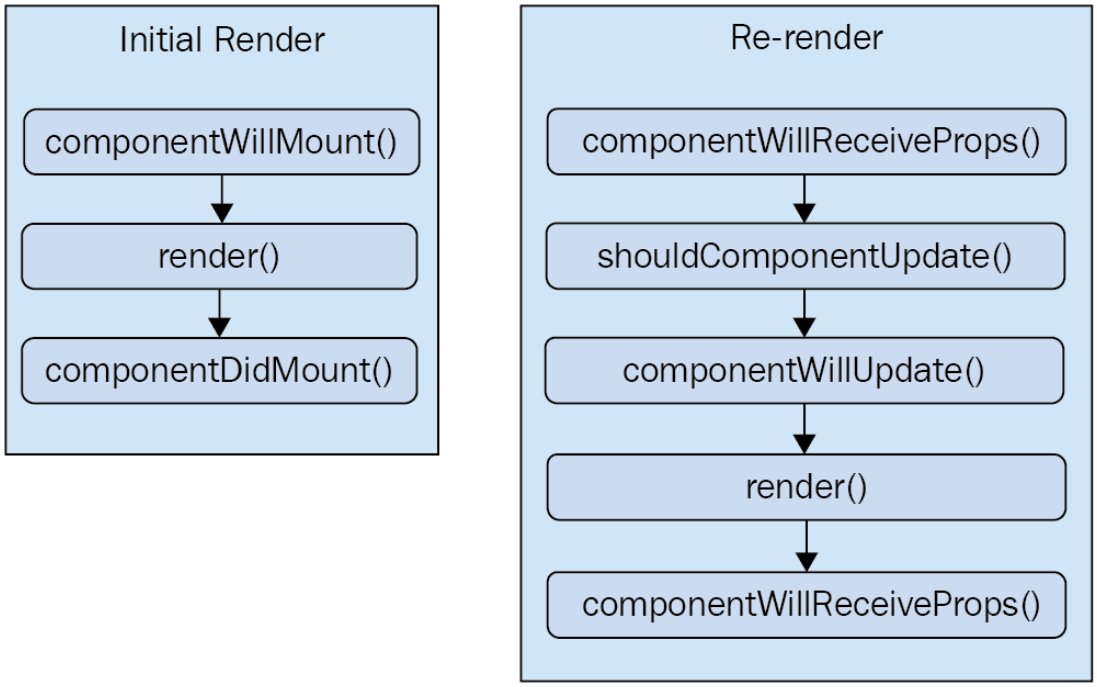
O *componentWillUpdate* é igual o comando WillMount, só que é executado antes da atualização de um componente.

O componentDidUpdate é igual o comando DidMount, só que é executado logo depois da atualização de um componente.

O *componentWillUnmount* é executado quando o ciclo de vida de um componente termina e ele vai ser removido do DOM.

O *shouldComponentUpdate* deve retornar true/false dizendo se ele um componente deve ser atualizado ou não, com base em certos parâmetros sendo usado assim para resolver questões de performance.

Figura 7 – Lifecycle React

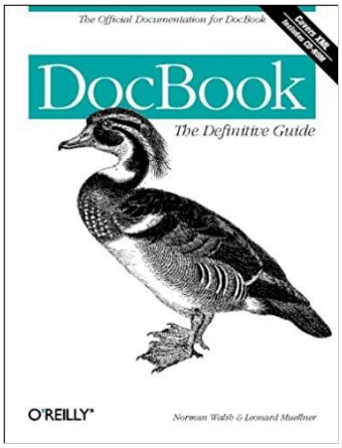


Fonte: https://hub.packtpub.com/react-component-lifecycle-methods-tutorial/

2.3.4 Docbook

Antes de falar de XML temos que falar sobre DocBook é uma linguagem de marcação semântica para documentação técnica, sua criação foi feita para criação de documentos na área da informática dentro dela temos SGML, XML E DSSSL.

Figura 8 – DocBook, The Definitive Guide



Fonte: https://www.amazon.com.br/DocBook-Definitive-Guide-Norman-Walsh/dp/1565925807

2.3.4.1 XML

O *eXtensible Markup Language* (XML) é um subtipo da SGML e uma linguagem de marcação recomendada e criada em 1996 pela (*World Wide Web Consortium*) W3C, que combina a flexibilidade da SGML com uso do HTML, mais conhecida por ser a versão digital da nota fiscal. Atualmente substituída pelo JSON (JavaScript *Object Notation*) que utiliza estrutura de dados em formato de texto para troca de dados entre aplicações simples

Figura 9 -Ícones XML, SGML E JSON

Fonte: https://en.wikipedia.org/

2.3.4.2 SGML E DTD

O *Standard Generalized Markup Language* (SGML) e uma metalinguagem através da qual se pode definir as linguagens de marcações de documentos, seu objetivo e fazer com que os documentos codificados de acordo com a regras possam ser transportados sem perda de informação. Criado e fazendo parte da ISO 8879.

DTD (*Document Type Definition*) e responsável por fornecer os meios de validação dos arquivos XML, especificando o conjunto de regras que controlam o XML, assim todo XML pode fazer referência ao arquivo DTD. O DTD analisa os erros dentro do aquivo XML e aponta onde ocorreu o erro.

2.3.4.3 DSSL

Document Style Semantics and Specification Language (DSSSL) é um padrão internacional desenvolvido para fornecer folhas de estilo para documentos SGML que foi feita na ISO/IEC 10179 publicada em 1996

2.3.5 SEO

O *Search Engine Optimization* (SEO) é um motor de otimização de busca criado para facilitar uso do WordPress que por sua vez é um Content Management System (CMS), que é usado para criar, gerenciar e publicar um site.

O SEO pode ser mais bem compreendido através de sites de pesquisa, que geralmente ranqueiam os sites mais acessados usando um mecanismo de chamado *Search Engine Result Page* (SERP), existem vários tipos de Search Engine Results (SERPs). Com isso SEO busca ajudar os mecanismos de busca, aumenta chance de um site ter um melhor ranqueamento nas páginas de resultado, esse movimento acaba causando um tráfego orgânico para os sites, tráfego orgânico ocorre quando um usuário chega em um site desejado através de suas pesquisas, esse evento muita das vezes dá impressão de que o mecanismo de busca está sempre atualizado.

2.3.5.1 Crawling

Mecanismo de SERP que usa bots para escanear a internet atra de conteúdos novos.

2.3.5.2 Indexação

Mecanismo de SERP que organiza e salva as páginas coletadas no banco de dados.

2.3.5.3 Ranqueamento

Mecanismo de SERP que ranqueia as páginas correlacionadas com a pesquisa de usuário, usando as páginas que foram indexadas anteriormente, a posição pode variar de acordo com os fatores utilizados (tempo de carregamento, número de acessos, estrutura do site e etc.).

2.3.6 Props e State

Properties (Props) bem comum durante o desenvolvimento de software em react ele server para fazer a passagem de parâmetros. Bom exemplo disso pode ser uma mensagem personalizada de forma dinâmica ao usuário como:

Seja bem-vindo “Fulano”!!!, com o uso dele podemos fazer validações com a interação de evitar bugs, com a aplicação PropTypes, que é usado para documentar os tipos de pretendidos de propriedades a serem passadas aos componentes.

State responsável pela forma de apresentar os dados, de forma transformada ou não, podemos usar uma função nativa do React para alterar o estado dos componentes, conhecida como setState. Com isso falaremos sobre componente do state, um deles é o Container trabalha com apresentação, transformação e logica dados (statefull), porém seus pontos fracos são que ele deve ser uma classe e não uma função.

Temos o Presentational esse por usa vez só se importa somente com a apresentação podendo ser usado em uma função, porém não possui estado (stateless).

2.3.7 React JS

React JS é uma biblioteca voltado para desenvolvimento web, lançado em 2013 pelo Facebook, com objetivo de facilitar a vida dos desenvolvedores, fazendo uso de um componente chamado JavaScript eXtension (JSX) que permite desenvolvimento que combina JavaScript com e Linguagem de Marcação de Hipertexto (HTML), assim permitindo escrever HTML dentro do JavaScript, graças a isso podemos desenvolver aplicações e componentes alto rendimento, tornado esse uma das suas principais características.

2.3.7.1 JSX

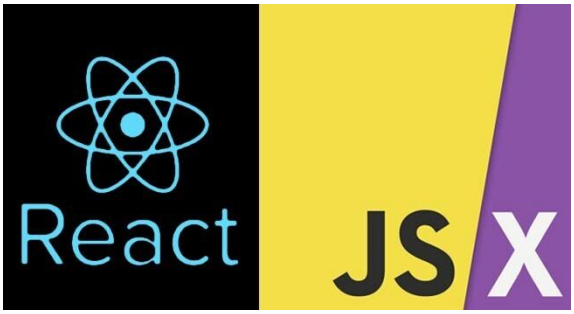
No React o JavaScript *eXtension* (JSX) pode servir como extensão útil de *eXtensible Markup Language* (XML) para construir um esquema de componentes conjunto de nós em XML, tornado mais fácil compreender o DOM. O XML explicando de maneira simplificada nada mais é que uma linguagem de marcação, ela define regras para codificar documentos. O JSX permite simplificar a associação de manipulação de eventos e propriedades com atributos XML.

O JSX logo que chegou quebrando um dos princípios básicos para desenvolvimento de software da época o Separation of Concerns, que se baseava no Princípio da Responsabilidade Única, para evitar que o código tivesse de se preocupar em fazer múltiplas ações fazendo assim separação de interesses para evitar tal evento.

2.3.8 React Native

React Native foi criado em 2015 pelo Facebook, sendo um framework para aplicativos móveis multiplataforma (Android e iPhone operating system (iOS)), convertendo todo código para linguagem nativa do sistema operacional assim podendo ser desenvolvido qual quer um deles (Windows, macOS ou Linux), é liberado por uma licença de Massachusetts Institute of Technology (MIT), uma licença que permite uso em software livre quanto a em software proprietário, possui interface que permite uso de JavaScript e possui a base de conhecimento compartilhado com desenvolvimento mobile e front-end.

Figura 10 -React e JSX



Fonte:

3 PROJETO PRATICO

Este capítulo demonstra como podemos implementar em React, criando de forma simples. Para tal será utilizado Node, NPM, React, Vite e o Visual Studio Code.

3.1 Pré-requisitos e Configuração de ambiente

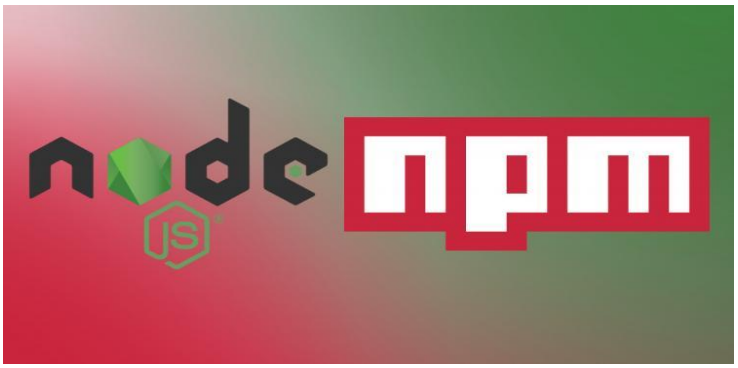
3.1.1 Node.js

Este Node.js é um software de código aberto baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript, para instalar basta acessar https://nodejs.org/en/download/ e escolher o mais adequado ao ambiente. (Node.js, 2009?)

3.1.2 NPM

NPM (Node *Package Manager*) é um gerenciador de pacotes para o Node.JS para instalar e utilizar os comandos npm basta abrir o prompt de comando como administrador e executar o comando **npm install**. (SCHLUETET, 2009).

Figura 11 -Node.Js e npm



Fonte: https://www.linuxinsider.gr/forum/programmatismos/9585/prota-bimata-sto-web-development-ti-einai-npm-kai-pos-hrisimopoioyme-gia

3.1.3 Vite

Vite é uma ferramenta de compilação que visa fornecer uma experiência de desenvolvimento mais rápida e enxuta para projetos da Web modernos. É composto por duas partes principais:

- Um servidor de desenvolvimento que fornece aprimoramentos de recursos avançados em módulos ES nativos, por exemplo, Hot Module Replacement (HMR) extremamente rápido.

- Um comando de compilação que agrupa seu código com Rollup, pré-configurado para produzir ativos estáticos altamente otimizados para produção.

Figura 12 -Vite

Fonte: https://vitejs.dev/

3.1.4 Visual Code Studio

Visual Studio Code é uma IDE de código aberto gratuita mantida pela Microsoft, a qual utilizaremos para desenvolver a aplicação. Para instalar basta acessar o site do link: https://code.visualstudio.com/download/ e escolher a versão mais adequada ao ambiente

Figura 13 -Visual Studio

Fonte: https://www.tabnine.com/blog/visual-studio-vs-visual-studio-code/

3.1.5 The Movie Database API

The Movie Database (TMDB) é um banco de dados de filmes e programas de TV construído pela comunidade, que disponibiliza uma API pública para consumo desses dados. Iniciado em 2008 e com o foco internacional, tem uma amplitude de dados incomparável que é disponibilizada gratuitamente.plementação

3.1.6 Pacotes

Para esse projeto utilizaremos os pacotes:

- React Router (react-router-dom): para criação e navegação entre rotas do site.
- Styled Components: para estilização dos elementos diretamente pelo Javascript.
- React Icons: uma biblioteca de ícones integrada ao React

3.2 Implementação

Construiremos uma plataforma de catálogo de filmes para exemplificação de tudo que foi mostrado até agora.

Para isso iniciaremos a configuração do nosso projeto iniciando um novo projeto em React através do Vite, para isso basta abrir a pasta de sua preferência no terminal e digitar:

npm create vite@latest

Caso ele peça para instalar os pacotes do vite basta digitar y e pressionar “enter”.

Ele irá pedir algumas informações para configuração do projeto, basta preencher conforme solicitado e pressionar “enter” para avançar.

Após a configuração inicial do vite, precisaremos criar uma conta no site do The Movie Database para conseguirmos uma API Key para quando formos usar API para consultar os dados para plataforma.

Acesse o site themoviedb.org, vá até “Junte-se ao TMDB” e faça o seu cadastro preenchendo as informações solicitadas.

Após a realização do cadastro será solicitada a confirmação do seu e-mail, basta você clicar no link que foi enviado para o seu e-mail e sua conta estará ativa.

Acesse a plataforma e vá em configurações, API, solicitar chave de API. Nesta seção haverá o texto “To generate new API Key, click here”, basta clicar onde solicitado, selecionar a opção “Developer” e aceitar os termos. Ele pedirá informações sobre você e sobre o tipo de uso que irá fazer, basta preenchê-las e enviar o formulário para receber a sua API Key.

Abra o VS Code e selecione a pasta em que configuramos o Vite anteriormente, vá em terminal, novo terminal e assim que o terminal for aberto, digite:

npm install

Após finalizar a instalação, digite o seguinte código para instalar todas as dependências que usaremos:

npm install react-router-dom styled-components react-icons

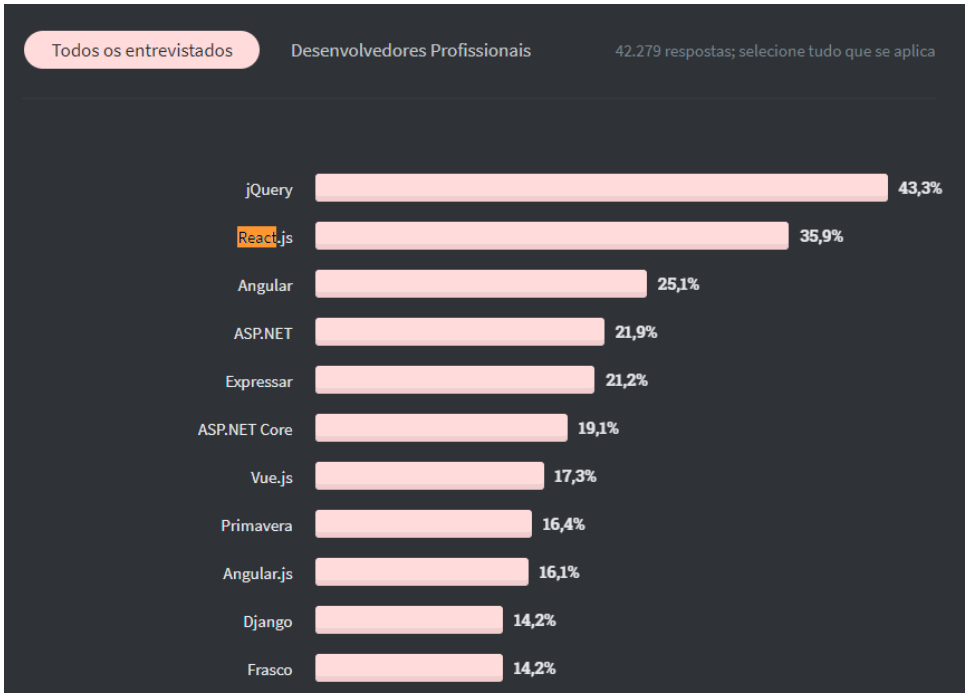
Com isso já temos o mínimo necessário para construção da plataforma.

Para consultar todo o código da plataforma visite o GitHub do projeto em https://github.com/HerbertVeloso/tcc-web, e para visualizar o projeto implementado acesse https://tcc-preview.netlify.app/.

4 ESTUDO DE CASO

O React é uma das bibliotecas JavaScript mais populares atualmente, utilizada para a criação de interfaces de usuário escaláveis e flexíveis. No entanto, muitos desenvolvedores enfrentam dificuldades no aprendizado dessa tecnologia devido à sua curva de aprendizado íngreme. O objetivo deste trabalho é apresentar uma solução didática para o aprendizado eficaz de React, junto de um jogo em JavaScript. Difundindo assim a biblioteca React no dia a dia dos desenvolvedores web que estão estudando ou exercendo a profissão.

Figura 14 – Porcentagem de usuários de para cara biblioteca Javascript



Fonte: https://www.treinaweb.com.br/blog/5-motivos-para-estudar-react

Além disso, a difusão da tecnologia no dia a dia dos desenvolvedores pode levar a soluções de problemas de forma simplificada (busca no Google facilitada) e aumento de usabilidade no mercado, como observado com o Java.

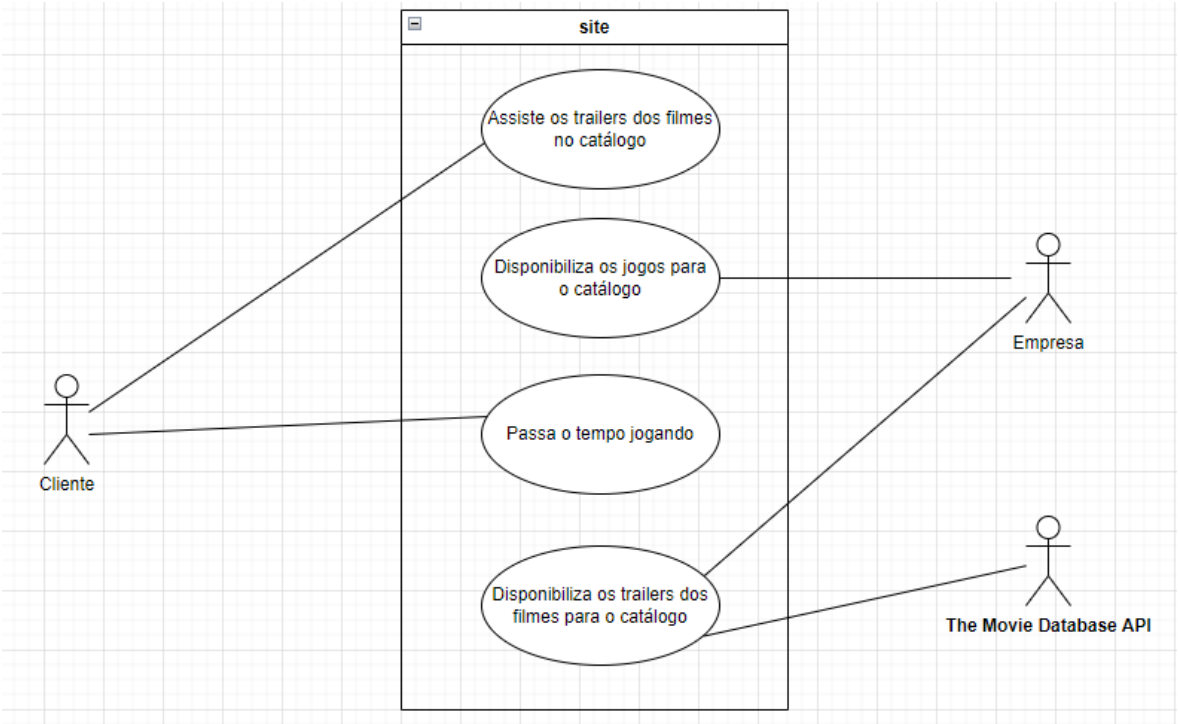
4.1 Modelo de Casos de Uso

Para descrever as funcionalidades utilizadas neste projeto, foi elaborado um diagrama de Casos de Uso e um Diagrama de Classes, esse para definição da estrutura estática do sistema proposto.

A figura 15 apresenta o diagrama de Casos de Uso, onde todos os atores são ligados às suas respectivas tarefas no código de extração de dados, processamento e geração do modelo de previsão.

O dono do site pediu a construção de um site aberto sem cadastro, onde os amantes de filmes possam ver quais filmes foram lançados atualmente, tambem pediu para montarmos uma aba com jogos retros, para que as pesssoas possam jogar online e quem sabe no futuro monetizar o site, vamos ver o comportamente dos atores na figura 15.

Figura 15 – Modelos do caso de uso

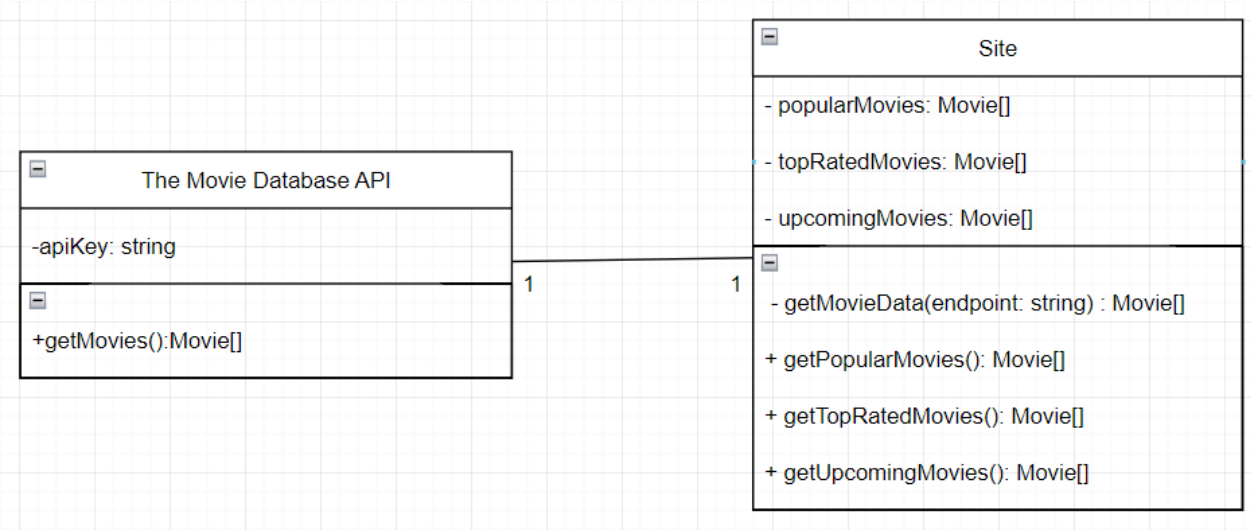


Fonte: Elaborado pelo autor

4.2 Modelo de Classes

A figura 16 apresenta o diagrama de Classes onde veremos melhor relacionamento entre o site e seus respectivos atores, como a The Movie Database API que é uma base de dados gratuita *open source* (código aberto) que é usada pelo nosso site para disponibilizar o catálogo para o cliente que acessá-lo.

Figura 16 – Modelos do classes

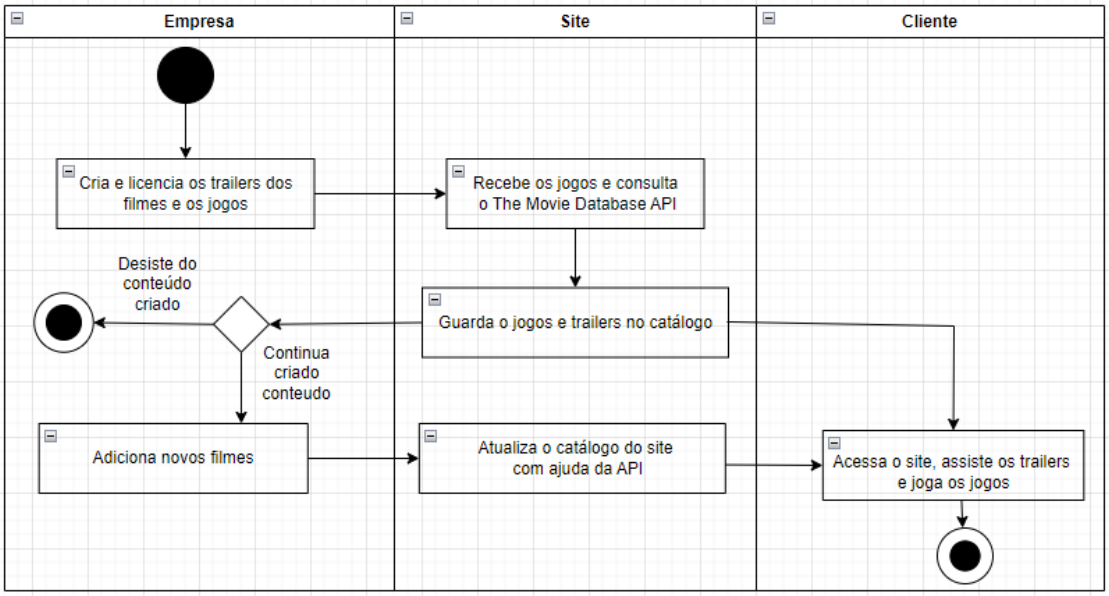


Fonte: Elaborado pelo autor

4.3 Modelo de Interação

O Modelo de Interação abaixo (figura 17) apresenta todo o processo que será abordado neste capítulo, que tem o objetivo de extrair os dados, realizar as previsões e divulgar os resultados no catálogo. Onde podemos ver melhor a The Movie Database API funciona junto a aplicação do nosso projeto.

Figura 17 – Modelo de Interação exibição dos dados

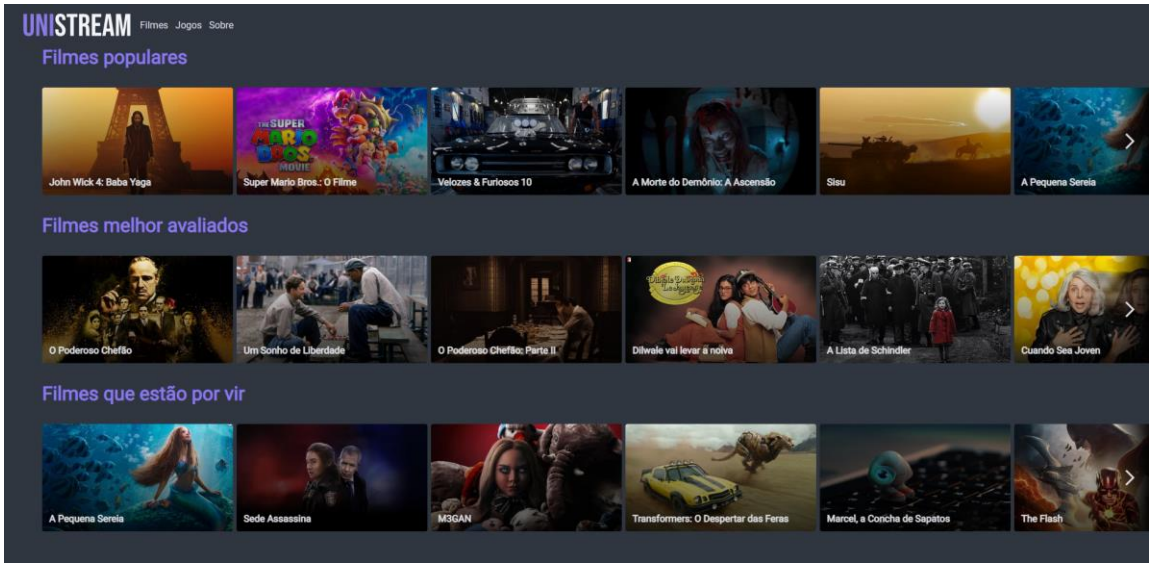


Fonte: Elaborado pelo autor

4.4 Interfaces do Sistema

Abaixo (figura 18) apresenta uma tela baseada no Netflix usando The Movie Database API, para criar o catálogo de filmes do nosso projeto em react.

Figura 18 – Tela do catálogo de filmes



Fonte: Elaborado pelo autor

A tela abaixo (figura 19), temos um clássico da primeira era dos vídeos games, um antigo jogo de fliperamas, o Snake, essa tela é baseado no clássico jogo de celular do Nokia.

Figura 19 – Tela de jogo Snake

Fonte: Elaborado pelo autor

Na tela ilustrada na figura 20, temos outro clássico dos jogos de fliperama, o Pong, feito com base no curso da Alura usando ferramenta P5 JS. Novamente como não usamos essa plataforma, foi criado uma solução para os objetos na tela, desdá sua colisão da bola com as raquetes a movimentação dos elementos em tela.

Figura 20 – Tela de jogo Pong

Fonte: Elaborado pelo autor

5 CONCLUSÃO

O objetivo deste trabalho buscou demonstrar, de forma prática, a montagem de uma aplicação front-end com jogos em JavaScript utilizando a biblioteca React. Através da criação de um minimundo de site aberto, sem a necessidade de cadastro de clientes, e a implementação de uma aba de jogos retrô, foi possível apresentar uma solução completa e envolvente para os usuários.

Ao construir uma plataforma que integra um catálogo de filmes e jogos para jogar, conseguimos exemplificar todos os conceitos e técnicas discutidos ao longo do trabalho. Através da interface desenvolvida, os usuários têm acesso a uma variedade de opções de entretenimento, que atendem tanto aos amantes de filmes quanto aos entusiastas de jogos retrô.

A utilização da biblioteca React se mostrou fundamental para a construção dessa aplicação. Através dos componentes reutilizáveis e da arquitetura baseada em fluxo de dados unidirecional, foi possível desenvolver uma aplicação modular, escalável e de fácil manutenção. Além disso, a flexibilidade e a performance proporcionadas pelo JavaScript foram essenciais para a criação de uma experiência interativa e fluida para os usuários.

Durante o processo de desenvolvimento, enfrentamos desafios e aprendemos lições valiosas. A importância de um planejamento cuidadoso, a atenção aos detalhes e a realização de testes foram aspectos que se mostraram cruciais para o sucesso do projeto. Além disso, a colaboração e a comunicação eficiente entre os membros da equipe garantiram um progresso contínuo e a superação de obstáculos.

É importante ressaltar que este trabalho representa apenas um ponto de partida no vasto campo do desenvolvimento front-end. A área de aplicações web está em constante evolução, e é necessário acompanhar as tendências e novas tecnologias para se manter atualizado. Aprofundar os conhecimentos em React, explorar outras bibliotecas e frameworks, e estar sempre aberto a aprender e aprimorar as habilidades são atitudes essenciais para se destacar nesse campo.

Além disso, é fundamental compreender as necessidades e preferências dos usuários ao criar aplicações interativas e cativantes. A união de elementos como design intuitivo, usabilidade, velocidade de carregamento e uma ampla seleção de conteúdo contribuem para uma experiência positiva e envolvente.

Por fim, a construção desta aplicação front-end com jogos em JavaScript usando a biblioteca React reforça a importância do aprendizado prático e da aplicação dos conceitos teóricos no desenvolvimento de projetos reais. Através desse trabalho, pudemos comprovar a viabilidade e a eficácia dessa abordagem, além de adquirir experiência valiosa que certamente será aplicada em futuros desafios e oportunidades profissionais.

Em suma, a implementação bem-sucedida dessa aplicação front-end reafirma a importância da combinação de conhecimentos técnicos, criatividade e a capacidade de adaptação às necessidades dos usuários. Através da aplicação prática dos conceitos aprendidos, pudemos criar uma plataforma que proporciona entretenimento e diversão aos usuários, demonstrando todo o potencial da tecnologia JavaScript.

Como trabalho futuro sugerimos a implementação de um back end, para os seguintes requisitos: cadastro do login na plataforma, criação de listas de filmes favoritos e a elaboração de um meio de monetização do site.

BIBLIOGRAFIA

Redes VSAT: Redes de Computadores. **Teleco**, 2022. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialredesvsat/pagina_2.asp#:~:text=Tanenbaum%20(2003)%20define%20o%20modelo,sistemas%20abertos%20e%2Fou%20propriet%C3%A1rios.> Acesso em: junho, 2023.

NODE.JS. About Node.js. entre 2009 e 2022. Disponível em:<https://nodejs.org/en/about/> Acesso em: Abril, 2023.

JavaScript. About JavaScript. entre 2011 e 2022. Disponível em:< https://www.alura.com.br> Acesso em: Abril, 2023.

VITE. About vitejs. entre 2009 e 2022. Disponível em:< https://vitejs.dev/Acesso em: Abril, 2023.

Cisco. About Introduction to Networks. Getting Started entre 2002 e 2022. Disponível em: <https://www.cisco.com/> Acesso em: abril, 2023.

GOOGLE. **Google.** 2017. Disponível em: <https://www.google.com.br/>. Acesso em: 15 julho, 2023.

Melo, Diego. O que é XML? [Guia para iniciantes]. tecnoblog. São Paulo, 2021. Disponível em: < https://tecnoblog.net/responde/o-que-e-xml-guia-para-iniciantes/> Acesso em: 10 jun. 2023.

REACT NATIVE. Getting Started entre 2015 e 2022 Disponível em: <https://reactnative.dev/docs/getting-started> Acesso em julho, 2023

REACT. Getting Started entre 2013 e 2022 Disponível em:<https://reactjs.org/> Acesso em junho, 2023

Computer Hope. About GUI entre 2021 e 2022 Disponível em:<https://www.computerhope.com/jargon/g/gui.htm> Acesso em junho, 2023

KRAUSE, Jörg. **Introducing Web Development.** 2016. Disponível em: <https://books.google.com.br/books/about/Introducing_Web_Development.html?id=EKAcvgAACAAJ&redir_esc=y>. Acesso em: 18 jun. 2023.

SOUZA, Vitor Estevão Silva. **FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web.** 2007. Disponível em: <https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/frameweb__um_metodo_baseado_em_frameworks_para_o_projeto_de_sistemas_de_informacao_web_2007.pdf&usg=AFQjCNG5A50iN_ESSwIiPXCklJAXQH02XA>. Acesso em: 10 abril. 2023.

VICENTINI, Luiz Atilio; MILECK, Luciângela Slemer. **DESENVOLVIMENTO DE SITES NA WEB EM UNIDADES DE INFORMAÇÃO: METODOLOGIAS, PADRÕES E FERRAMENTAS.** 2000. Disponível em: Seminário Nacional de Bibliotecas Universitárias, 11 maio. 2000, Florianópolis. Anais... Florianópolis: UFSC, 2000.

WEBCOMPONENTS.ORG. **Web Components.** 2017. Disponível em: <https://www.webcomponents

.org/>. Acesso em: 16 jun. 2023.

GINIGE, Athula; MURUGESAN, San. **Web Engineering: An Introduction.** 2001. Disponível em: <https://www.computer.org/csdl/mags/mu/2001/01/u1014.pdf&usg=AFQjCNHqIm6AXSt3A5Mvh1U_dak_MVpLMQ>. Acesso em: 10 jun. 2023.

CANDELIBAS. duck-hunt Setembro, 2009 Disponível em:<https://github.com/npm/cli/commit/4626dfa73b7847e9c42c1f799935f8242794d020 > Acesso em: Outubro, 2023.

IBM. DTDs (Document Type Definitions) - Visão Geral. **IBM**, 2021. Disponível em < https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=dtds-document-type-definitions-overview >. Acesso em 06 de Julho de 2023.

OVERSON, Jarrod; STRIMPEL, Jason. **Developing Web Components.** 2015. Disponível em: <https://books.google.com.br/books?id=7mvIBgAAQBAJ&printsec=frontcover&dq=developing+web+components&hl=pt-BR&sa=X&ved=0ahUKEwiG8qrVsNzUAhXHxpAKHX-3Bf4Q6AEIKjAA#v=onepage&q=developing web components&f=false>. Acesso em: 26 jul. 2023.