

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
ENGENHARIA DE COMPUTAÇÃO**



PLATAFORMA DE EDIÇÃO/AUTOMAÇÃO PARA TRABALHOS ACADÊMICOS

HIGOR FERREIRA ALVES SANTOS

**GOIÂNIA - GO
2023**

HIGOR FERREIRA ALVES SANTOS

PLATAFORMA DE EDIÇÃO/AUTOMAÇÃO PARA TRABALHOS ACADÊMICOS

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador:

Prof. M.E.E.Marcelo Antônio Adad

Banca examinadora:

Prof. Dra. Miriam Gusmão

HIGOR FERREIRA ALVES SANTOS

PLATAFORMA DE EDIÇÃO/AUTOMAÇÃO PARA TRABALHOS ACADÊMICOS

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia de Computação, em: ____/ ____/ ____

Orientador: Prof. M.E.E.Marcelo Antônio Adad

Orientador1: Prof. Dra. Miriam Gusmão

Orientador2:

DEDICATÓRIA

AGRADECIMETOS

EPÍGRAFE

RESUMO

ABSTRACT

Lista de ilustrações

Figura 1 – Passo a passo para criar um documento na plataforma	2
Figura 2 – Divisão de blocos em uma imagem	4
Figura 3 – Etapas do processo de Parsing	5
Figura 4 – Pilares da plataforma, (mapa mental)	7
Figura 5 – Logotipo do React	13
Figura 6 – Logotipo do NextJs	15
Figura 7 – Logotipo do EditorJs	17

Lista de tabelas

Tabela 1 – Diferenças entre o JavaScript e o JSON	12
---	----

LISTA DE SIGLAS

ABNT Associação Brasileira de Normas Técnicas. 1, 2

API Application Programming Interface. 16, 17

CSS Cascading Style Sheet. 10, 16

ECMA European Computer Manufacturers Association. 11

IES Instituição de Ensino Superior. 1, 2

NBR Norma Brasileira Regulamentadora. 1

PDF Portable Document Format. 2–5, 11

SSR Server Side Rendering. 14

SVG Scalable Vector Graphics. 10

TCC Trabalho de Conclusão de Curso. 2

UFPB Universidade Federal da Paraíba. 2

XML eXtensible Markup Language. 10

LISTA DE ABREVIATURAS

CERN Conseil Européen pour la Recherche Nucléaire. 9

HTML HyperText Markup Language. 9, 10, 17

JS JavaScript. 11, 12, 16

JSON JavaScript Object Notation. 5, 12, 17

MathML Mathematical Markup Language. 10

PHP Hypertext Preprocessor. 13, 14

TS TypeScript. 11, 12

W3C World Wide Web Consortium. 9, 10

Web World Wide Web. 8–11, 13, 15

XHTML eXtensible HyperText Markup Language. 10

XSS Cross-Site Scripting. 13

Sumário

Lista de ilustrações	i
Lista de tabelas	ii
Sumário	v
1 INTRODUÇÃO	1
1.1 Objetivo	2
1.2 Fluxo do documento	3
1.2.1 Escrita em blocos	3
1.2.2 Bloco	3
1.2.3 Parsing	4
1.3 Ambiente de desenvolvimento	5
1.3.1 Lista de tecnologias do ambiente de desenvolvimento	6
2 FUNDAMENTAÇÃO TEÓRICA	7
2.1 Do Front-End	8
2.1.1 Tecnologias Web	8
2.1.1.1 <u>Linguagem de Marcação de Hipertexto, HTML</u>	9
2.1.1.2 <u>Funcionamento do HTML</u>	9
2.1.1.3 <u>HTML versão 5</u>	10
2.1.1.4 <u>Folhas de Estilo em Cascata, (CSS)</u>	10
2.1.1.5 <u>JavaScript</u>	11
2.1.1.6 <u>TypeScript</u>	11
2.1.1.7 <u>JavaScript Object Notation, JSON</u>	12
2.1.2 Bibliotecas e Frameworks	13
2.1.2.1 <u>ReactJs</u>	13
2.1.2.2 <u>NextJs</u>	15
2.1.2.3 <u>EditorJs</u>	17
2.2 Do Back-End	18
2.3 O processo de Parsing	18
REFERÊNCIAS	19

1 INTRODUÇÃO

Escrever um trabalho científico pode ser uma tarefa desafiadora. (SEVERINO, 2017) destaca a complexidade e o rigor necessários na elaboração de trabalhos científicos, que não apenas envolvem o domínio do conteúdo específico, mas também a aderência às normas técnicas para apresentação formal e formatação correta.

A Associação Brasileira de Normas Técnicas (ABNT) , é a entidade responsável por, dentre outras, fornecer as normas que regulam o processo de criação de trabalhos acadêmicos. A Norma Brasileira Regulamentadora (NBR) Nº 14724, por exemplo: Especifica os princípios gerais para a elaboração de (teses, dissertações e outros), visando sua apresentação à instituição (banca, comissão examinadora de professores, especialistas designados e/ou outros) (ABNT, 2021).

Ademais, ainda com respeito aos trabalhos acadêmicos, não somente a regulamentação da NBR 14724 deve ser observada. Há ainda a NBR 6023 que trata a respeito da elaboração de referências e a NBR 10520, que diz respeito às citações em documentos.

(CASTRO, 2011), adverte que: "Em ciência, não pode haver uma separação entre forma e conteúdo. Trata-se de uma separação fictícia, pois fica se conhecendo o conteúdo pela forma." Ou seja: A forma do trabalho, sua apresentação, sua formatação e todo o seu arranjo gráfico é tão importante quanto seu conteúdo. (MEDEIROS, 2012) vai complementar essa visão, afirmando que a apresentação gráfica "[...] contribui para a consecução de um trabalho capaz de atingir seu objetivo. Monografia realizada sem a preocupação gráfica, em geral, acaba malsucedida."

Em seu artigo, (SILVA; VITORIA, 2014) vão analisar as percepções e dificuldades dos alunos de um curso superior em Tecnologia de Gestão em Recursos Humanos. Dentre suas dificuldades, (dos alunos em questão), é destacada a questão da formatação do trabalho acadêmico. Há também o fato de que as bancas avaliam os trabalhos baseadas em critérios da própria Instituição de Ensino Superior (IES), critérios estes que não estão necessariamente presentes nas normas da ABNT, ou seja, há uma subjetividade presente que não é comum a todas às IES quanto a questão da formatação. Essa subjetividade contribui para a confusão dos alunos, pois a IES avaliará de acordo com aquilo que julga apropriado, o que muitas vezes pode obscurecer o direcionamento do aluno ao redigir/formatar seu trabalho."

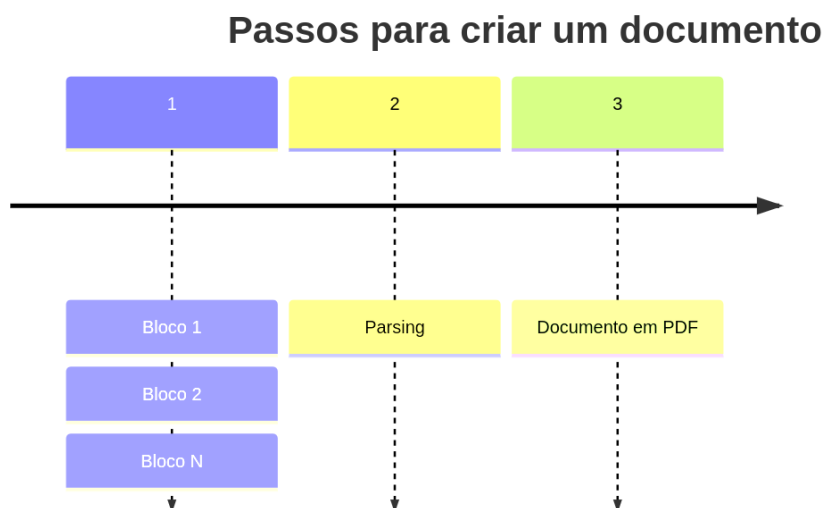
(SANTOS, 2020) em seu Trabalho de Conclusão de Curso (TCC) , também analisa as dificuldades encontradas por egressos, desta vez do curso de Ciências Contábeis da Universidade Federal da Paraíba (UFPB). Em sua pesquisa é destacado que "Quanto a formatação do trabalho com as normas da ABNT, [...], 60% teve alguma dificuldade, inclusive 32% teve muita dificuldade.", ou seja, a formatação do trabalho é um grande desafio presente na vida de boa parte dos estudantes em processo de escrita.

1.1 Objetivo

Levando em consideração os problemas que os alunos de diversas instituições de ensino enfrentam ao elaborar seus respectivos trabalhos (conforme apresentado acima), o objetivo deste instrumento é desenvolver uma plataforma web de alta interatividade¹ e inteligibilidade², de modo que o discente possa se preocupar apenas com o conteúdo. Os detalhes de formatação, de acordo com os padrões da ABNT e da IES, ficarão a cargo da própria plataforma.

A criação de um trabalho de TCC se dará basicamente por três passos básicos: Escrita em blocos; *Parsing*³ e Documento em PDF . A Figura1 ilustra esse fluxo na linha do tempo.

Figura 1 – Passo a passo para criar um documento na plataforma



Fonte: Autoria própria

¹ Refere-se à capacidade de um sistema, aplicação ou interface de responder às ações do usuário de maneira eficaz e intuitiva

² Refere-se à clareza e compreensibilidade da interface, documentação e feedback fornecidos pelo sistema. Um software inteligível facilita o entendimento do usuário sobre como utilizá-lo e quais são os resultados de suas ações.

³ O termo Parsing, (do inglês: análise), será utilizado no sentido de analisar e transformar algo em outra coisa.

O usuário interagirá com a aplicação escrevendo blocos que serão transformados no documento final em PDF. A este processo daremos o nome de Parsing. Após este, bastará enviar o download do PDF ao usuário com todo o padrão de formatação. Os trabalhos desenvolvidos nesta plataforma terão então duas versões: A versão de blocos, (sem formatação e interativa); e a versão final já formatada em PDF.

1.2 Fluxo do documento

1.2.1 Escrita em blocos

A escrita se dará de modo em que tudo será considerado um bloco. A escrita em blocos consiste numa abordagem em que o texto vai sendo escrito em seu fluxo natural, porém blocos podem ser adicionados à escrita. Um bloco é um elemento adicionado ao fluxo de trabalho que desempenha um papel que o diferencia dos demais blocos. Por exemplo: Uma imagem pode ser considerada um bloco nesta abordagem, uma vez que não é um texto mas tem o objetivo de fornecer informações visuais. O próprio corpo do texto em si será considerado um bloco, denominado parágrafo. Um título será um bloco textual cujo objetivo será separar sessões do texto coesas. Uma lista será um bloco para enumerar itens e assim por diante. O documento será basicamente uma composição de diversos blocos dispostos de forma a formar uma unidade coesa final, que será o trabalho propriamente dito.

1.2.2 Bloco

Um bloco é uma unidade lógica no documento que desempenha um papel especializado que nenhum outro bloco o faz. Por exemplo: O bloco mais importante da plataforma⁴ será o de texto, (denominado bloco parágrafo), pois sem texto, não há trabalho. Sem texto não há tão pouca comunicação que transmita informação de caráter acadêmico-científico.

Semelhante ao bloco de texto, diversos outros blocos adjacentes auxiliarão na construção do documento acadêmico. O bloco de imagem, por exemplo, ajuda a exibir informações de forma ilustrativa e auxilia bastante em exemplos que estão sendo dados em determinado contexto do texto.

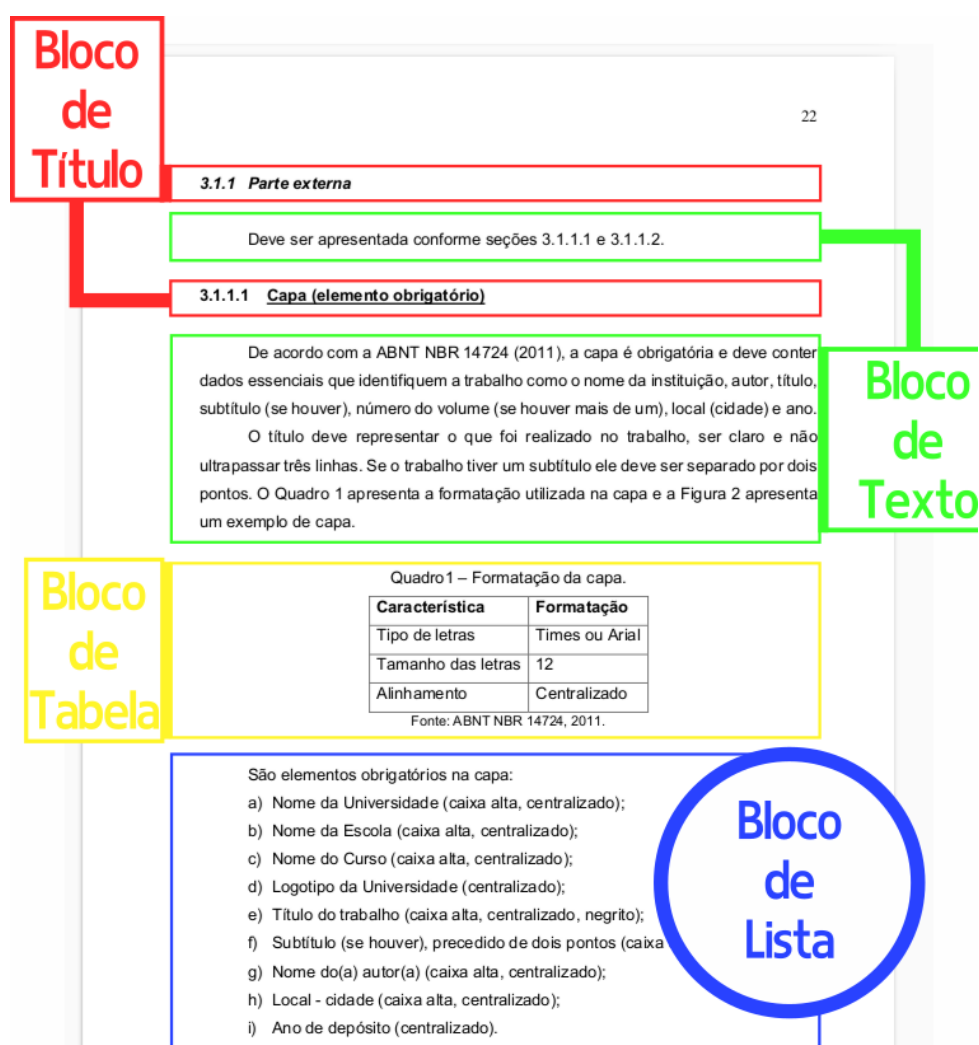
A maior parte dos blocos contará com uma espécie de submenu, (em termos de aplicação), que os permita personalizar. A personalização de blocos é importante para editar configurações e dar autonomia ao usuário em determinar mais precisamente o papel daquele bloco no texto. Por exemplo: Um bloco de título ajuda a separar o texto em unidades coesas. Porém, existem diversos tipos de títulos: Existe o título, o subtítulo, e até o subtítulo do subtítulo.

⁴ O termo plataforma será utilizado de forma intercambiável e como sinônimo de aplicativo; sistema web; ou aplicativo da web

O submenu será a configuração que o usuário fará no bloco após escolhê-lo. No caso do título, por exemplo: Após o usuário escolher este bloco, poderá configurar o nível de título desejado. Nível este que varia do 1 ao 4, sendo 2; 3 e 4 espécies de subtítulos. No caso de uma imagem, o submenu funcionará para que possa ser definida a imagem, bem como seu título de sua descrição.

A imagem abaixo ilustra a composição de um trabalho com seus respectivos blocos:

Figura 2 – Divisão de blocos em uma imagem



Fonte: Adaptado de (PUC-GO, 2022)

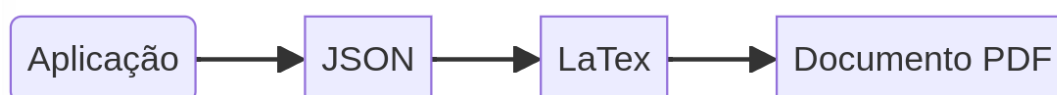
1.2.3 Parsing

O processo de Parsing é o processo que acontecerá sempre que o usuário desejar ver o *layout*⁵ da versão final de seu trabalho. Ele usa o código intermediário gerado pelos blocos para montar o PDF final.

⁵ Do inglês: Disposição, ou esboço. Esta palavra geralmente está associada ao desenho ou visual de algo.

Este processo é, em termos simples, uma espécie de análise a ser aplicada no código gerado pelos blocos da aplicação. A plataforma gerará um código JSON⁶ como resultado das interações do usuário, que posteriormente serão convertidos em código latex⁷. Só então, finalmente será utilizado um utilitário que converterá o código latex em um documento PDF. A Figura3 ilustra esse processo:

Figura 3 – Etapas do processo de Parsing



Fonte: Autoria própria.

1.3 Ambiente de desenvolvimento

O ambiente de desenvolvimento é de extrema importância para que todas as ferramentas utilizadas possam funcionar em perfeita harmonia em suas respectivas integrações e colaborações. Muitas vezes, problemas de compatibilidade podem afetar o funcionamento das mesmas e impedir que o programa final seja executado corretamente, causando *bugs*⁸ e outros imprevistos impeditivos tanto para a correta execução, quanto para a experiência de desenvolvimento. A lista abaixo diz respeito às ferramentas e ao ambiente onde este *software* foi desenvolvido, bem como todas as suas respectivas versões:

⁶ Ver (sessão que trata do JSON)

⁷ Ver (sessão que trata do latex)

⁸ Do inglês: Inseto. Esta palavra é muito utilizada no contexto de desenvolvimento de aplicativos para se referir a problemas que afetam o funcionamento dos mesmos

1.3.1 Lista de tecnologias do ambiente de desenvolvimento

Atender aos requisitos mínimos de hardware e software é fundamental para garantir uma experiência de usuário satisfatória e evitar problemas de desempenho ou compatibilidade com o aplicativo da plataforma. A seguir é listado o ambiente mínimo com seus respectivos softwares necessários para rodar o aplicativo da plataforma:

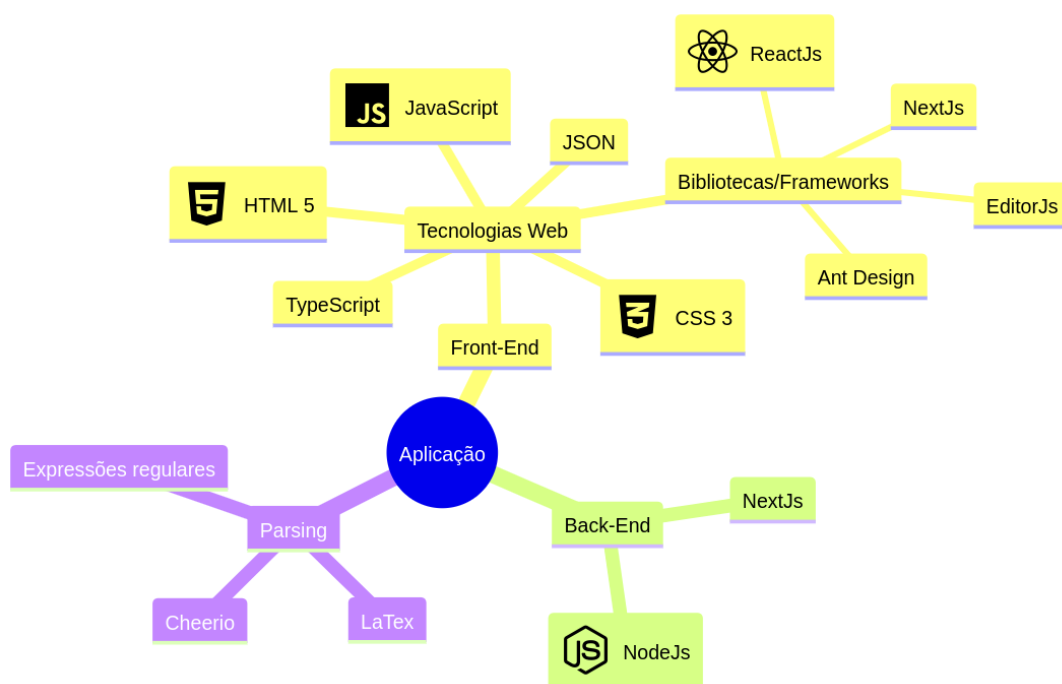
- Npm 10.2.3
- Yarn 1.22.19
- NodeJs 20.10.0
- TypeScript 5.3.3
- kpathsea version 6.3.4/dev
- Sistema Operacional: Ubuntu 20.04
- makeglossaries (Utilitário latex)
- BibTeX 0.99d (TeX Live 2022/dev/Debian)
- pdfTeX 3.141592653-2.6-1.40.22 (TeX Live 2022/dev/Debian)

2 FUNDAMENTAÇÃO TEÓRICA

A plataforma será construída sob alguns pilares fundamentais indispensáveis a seu funcionamento. São estes pilares que garantirão o sucesso e o correto funcionamento da aplicação, afim de que todo o objetivo discutido até o presente momento seja atingido.

A Figura4 mostra em forma de mapa mental todos os principais pilares sobre os quais o aplicativo será contruído. Estes pilares são formados por diversas tecnologias, bibliotecas, *frameworks*¹ e conceitos que deverão trabalhar de forma integrada.

Figura 4 – Pilares da plataforma, (mapa mental)



Fonte: Autoria própria.

Estes pilares estão subdivididos em três grandes subcategorias, a saber: Front-End; Back-End e Parsing. Cada qual com seus respectivos conceitos e tecnologias.

¹ Uma framework é como um kit de ferramentas pré-pronto que fornece uma gama de funcionalidades pré-construídas e testadas afim de facilitar o processo de desenvolvimento. (AMAZON; AWS,)

2.1 Do Front-End

O Front-End é, basicamente, a "linha de frente". É a parte da aplicação que interagirá diretamente com o usuário. Ao profissional que codifica e desenvolve esta parte do projeto, damos o nome de Desenvolvedor Front-End. A interface do usuário, que é onde o mesmo realiza suas interações com o sistema, normalmente é desenhada por um *designer*², ficando a cargo do desenvolvedor o papel de adaptar o *design*³ ao código afim de obter os efeitos desejados. (TOTVS, 2021)

2.1.1 Tecnologias Web

As tecnologias Web desempenham um papel crucial na criação de experiências digitais interativas, permitindo que os usuários se envolvam com o conteúdo de maneira mais dinâmica e significativa. A incorporação da *Internet*⁴ na vida diária resultou em mudanças significativas, marcada por um ritmo de evolução e aprimoramento sem precedentes, além da distribuição de conteúdo em massa. Juntamente com essas mudanças, surgiram novas tecnologias, variando de *softwares*⁵ a *hardwares*⁶, aprimorando a experiência de navegação na Web (MOLGADO, 2016).

A Internet, que teve origem nos Estados Unidos em 1969, foi inicialmente utilizada por universidades, governos e instituições financeiras antes de se expandir globalmente. No início, a internet era uma via de mão única onde os usuários consumiam informações e se comunicavam de maneira privada. A evolução começou com a introdução de sistemas de busca avançados, destacando-se o lançamento do Google em 1998, que democratizou o acesso à informação. (VITORIANO, 2019).

A grande reviravolta na internet aconteceu em 1999, com o surgimento do *Blogger*, marcando o início da Web 2.0, onde a comunicação tornou-se bidirecional. Os usuários passaram a gerar conteúdo e se relacionar publicamente com marcas, empresas e pessoas por meio de comentários, além de consumir informação. A evolução da tecnologia móvel, em conjunto com o surgimento de redes sociais como *Fotolog*, *MySpace*, *Orkut*, *Facebook*, *YouTube* e *Twitter*, ampliou o conceito de Web 2.0, permitindo o compartilhamento de fotos, vídeos e textos em uma escala maior. (VITORIANO, 2019).

A forma como se interage com a internet também evoluiu ao longo do tempo. Passou-se de sites estáticos para interativos e animados, chegando até aos sites total-

² Profissional que atua com design.

³ Do inglês: Desenho.

⁴ Rede mundial de computadores, (BRASIL, 2014).

⁵ O software é o conjunto de instruções dadas a um computador, de modo que ele execute determinada tarefa. Podemos dizer que o software é a parte lógica do sistema computacional. (MACHADO,).

⁶ Com hardware, compreende-se o equipamento físico de um sistema computacional. Suas unidades Lógicas de Processamento, memórias e unidades de armazenamento são hardware. (MACHADO,).

mente responsivos⁷ e adaptáveis de hoje. Isso foi possível devido ao desenvolvimento de novos gadgets e ao surgimento de novas linguagens de programação. Atualmente, a Web Moderna é composta por várias técnicas, metodologias, linguagens e ferramentas que permitem o desenvolvimento de aplicações conectadas e interativas, oferecendo diversas formas de interação com interfaces digitais. (VITORIANO, 2019).

2.1.1.1 Linguagem de Marcação de Hipertexto, HTML

A Linguagem de Marcação de Hipertexto, do inglês: HyperText Markup Language (HTML) foi criada por Tim Berners-Lee enquanto trabalhava na Organização Europeia para a Pesquisa Nuclear (CERN), o laboratório de física de partículas na Suíça, no final dos anos 1980 e início dos anos 1990. O objetivo era criar uma maneira de compartilhar documentos e informações em um ambiente de rede. A primeira versão do HTML tinha apenas 18 elementos de marcação, permitindo a formatação básica de texto e a inclusão de *links*⁸, imagens e listas. (W3C, 2023).

O HTML rapidamente ganhou popularidade e passou por várias iterações, cada uma adicionando novos elementos e funcionalidades. O HTML4, lançado em 1997, trouxe uma série de melhorias, incluindo mais controle sobre a aparência das páginas web, a introdução de folhas de estilo em cascata (HTML) e melhor suporte a *scripts*⁹. (W3C, 2023).

Finalmente, o HTML5, lançado oficialmente em 2014 pelo World Wide Web Consortium (W3C), trouxe uma série de novas funcionalidades, incluindo suporte nativo para vídeo e áudio; novos elementos semânticos; gráficos e animações; geolocalização; armazenamento local e muito mais. (W3C, 2023).

2.1.1.2 Funcionamento do HTML

O HTML funciona como uma linguagem de marcação, o que significa que ele usa "*tags*"¹⁰ para definir diferentes partes de um documento. Essas tags informam ao navegador como exibir o conteúdo da página. Por exemplo, a tag <p> é usada para definir um parágrafo, enquanto a tag <h1> é usada para definir um cabeçalho de primeiro nível. (W3C, 2023).

As páginas HTML são estruturadas usando uma combinação de elementos de bloco, (que formam a estrutura principal da página), e elementos *inline*¹¹, (que forma-

⁷ A responsividade é a capacidade de uma página da Web em se adaptar a diferentes dispositivos e tamanhos de tela. (MDN, 2023)

⁸ Do inglês: Ligação. Também chamado de hiperlink, é uma referência a um documento eletrônico que, quando clicado, leva o usuário para outro recurso ou documento.

⁹ Do inglês: Roteiro. Aqui usado no sentido de código fonte, que nada mais são do que um conjunto de instruções que o computador seguirá de modo interpretativo.

¹⁰ Do inglês: Marcação.

¹¹ Do inglês: Dentro da Linha. São elementos que podem ser escritos sem quebra de linha.

tam o conteúdo dentro desses blocos). Os elementos são aninhados dentro de outros elementos para criar a estrutura hierárquica da página. (W3C, 2023).

2.1.1.3 HTML versão 5

Com o HTML5, os desenvolvedores podem criar jogos online, reproduzir vídeos e áudios diretamente no navegador, tudo isso sem a necessidade de instalação de plugins externos. Isso resultou em uma melhor experiência geral do usuário, com carregamento mais rápido e maior compatibilidade entre os navegadores. (W3C, 2023).

Além disso, o HTML5 também trouxe recursos avançados de armazenamento local, como o *Web Storage*¹² e o *IndexedDB*¹³. Esses recursos permitem que os sites armazenem dados localmente no navegador do usuário, possibilitando a criação de aplicativos web *offline* e sincronização de dados em tempo real. (W3C, 2023).

Outra contribuição importante do HTML5 é o suporte a tecnologias de geolocalização e acesso aos recursos do dispositivo. Isso permite que os desenvolvedores acessem informações de localização do usuário, câmera, microfone e acelerômetro, abrindo possibilidades para o desenvolvimento de aplicativos web que utilizam esses recursos de forma integrada. (W3C, 2023).

Ao longo de sua história, o HTML tem evoluído constantemente para acompanhar as demandas e os avanços tecnológicos da web. O HTML5 é um marco significativo nessa evolução, trazendo recursos semânticos, multimídia e interativos para a criação de páginas da web modernas. (W3C, 2023).

2.1.1.4 Folhas de Estilo em Cascata, (CSS)

Folhas de Estilo em Cascata, ou *Cascading Style Sheets* (CSS), em tradução livre para o português, é uma linguagem de estilo altamente eficaz e amplamente utilizada. Sua principal função é definir a apresentação de documentos escritos em HTML ou XML. Isso inclui uma série de linguagens baseadas em XML, como SVG, MathML e XHTML. O CSS é responsável por descrever a forma como os elementos são apresentados em diferentes mídias, seja na tela do computador, em papel impresso, por meio de dispositivos de fala ou em outras formas de mídia. (MDN, 2023a).

Considerado uma das principais linguagens da Open¹⁴ Web, o CSS tem uma grande importância na padronização dos navegadores Web. Essa padronização é feita de acordo com as especificações estabelecidas pela W3C, a organização que lidera a Web mundial. O desenvolvimento do CSS é feito em níveis distintos: o CSS1, que hoje é considerado obsoleto; o CSS2.1, que atualmente é uma recomendação; e o CSS3, que está sendo dividido em pequenos módulos e caminha para a sua padronização. (MDN, 2023a).

¹² O termo Web Storage pode ser entendido, em tradução livre, como: Armazém da Web

¹³ Termo abreviado de "Indexed DataBase", (Base de Dados Indexada).

¹⁴ Do inglês: Aberto. Neste contexto, refere-se ao padrão "Aberto" da Web

2.1.1.5 JavaScript

JavaScript é uma linguagem de programação notavelmente versátil que, apesar de ser comumente conhecida pela sua utilização em páginas Web, vai muito além disso. Frequentemente abreviada para JS, essa linguagem é leve, interpretada e orientada a objetos com funções de primeira classe. Graças à sua flexibilidade, o JavaScript se expandiu para uma variedade de ambientes que não são navegadores, incluindo Node.js¹⁵, Apache CouchDB¹⁶ e Adobe Acrobat¹⁷, demonstrando sua adaptabilidade e eficácia em diversos contextos. (MDN, 2023b).

Com sua estrutura baseada em protótipos, o JavaScript é uma linguagem dinâmica que suporta múltiplos paradigmas de programação. Isso significa que, além de ser orientada a objetos, ela também suporta estilos de programação imperativos e declarativos, como a programação funcional. Essa capacidade de suportar diferentes estilos de programação torna o JavaScript uma ferramenta poderosa e flexível para os desenvolvedores. (MDN, 2023b).

O padrão para JavaScript é o ECMAScript. Desde 2012, todos os navegadores modernos oferecem suporte completo ao ECMAScript 5.1. Mesmo os navegadores mais antigos fornecem suporte, pelo menos, ao ECMAScript 3. A sexta versão do ECMAScript, oficialmente chamada de ECMAScript 2015 e inicialmente conhecida como ECMAScript 6 ou ES6, foi publicada pela ECMA International em 17 de junho de 2015. Desde então, as especificações do ECMAScript são lançadas anualmente, demonstrando o desenvolvimento contínuo e o avanço dessa linguagem padrão. (MDN, 2023b).

2.1.1.6 TypeScript

O TypeScript, as vezes abreviado como TS, é uma linguagem fortemente tipada construída em cima do JavaScript, (TYPESCRIPT,). Typescript traz uma sintaxe adicional para o JavaScript de modo que o mesmo possa suportar checagem de tipos estática. Sem o TS, fica difícil saber com quais tipos de dados estar-se a trabalhar durante o processo de desenvolvimento, pois o JavaScript é uma linguagem fracamente tipada. Os parâmetros das funções e variáveis não possuem nenhuma informação, forçando os desenvolvedores a recorrerem a todo momento à documentação ou intuir sobre as tipagens. Typescript resolve esse problema, permitindo tipar o código de modo que erros possam ser reportados quando a tipagem estiver incorreta, por exemplo: ao tentar-se passar uma string¹⁸ para uma função que espera um número,

¹⁵ Ver sessão que trata do Node.js

¹⁶ Base de dados que utiliza o JSON nativamente. Veja mais em:
<https://couchdb.apache.org/#about>

¹⁷ Software que lê e converte arquivos em formato PDF. Veja mais em:
<https://www.adobe.com/br/acrobat.html>

¹⁸ Do inglês: Corda, barbante ou fio. No contexto de programação, é usado como termo para cadeia de caracteres. O caractere é, na maioria das linguagens de programação, um tipo de dado. E textos

TypeScript lançará um erro. O JavaScript, por outro lado, permitirá a execução deste código podendo gerar erros de tempo de execução. (W3SCHOOLS, 2024).

O TypeScript possui um compilador, que nada mais é do que um transpilador. Este transpilador é responsável por transformar o código TS em JS. Desta forma, o código JavaScript resultante da transpilação pode ser rodado em praticamente qualquer navegador ou ambiente que suporte o JavaScript.

2.1.1.7 JavaScript Object Notation, JSON

JavaScript Object Notation, (Notação de Objeto JavaScript), popularmente chamado de JSON. É uma sintaxe para a serialização de objetos do javascript. Com objetos do javascript, compreende-se seus tipos de dados e valores, como: objetos; matrizes; números; strings; booleanos; *null*¹⁹ e *undefined*²⁰. Apesar de baseado na sintaxe do JavaScript, distingue-se desta no sentido da forma de escrita. A serialização é o processo de converter dados estruturados, (ou objetos), em um formato que pode facilmente ser armazenado e transmitido pela rede. O JSON basicamente converte os objetos JavaScript em strings. Uma característica deste, é que JSON é legível tanto por humanos, quanto por máquinas, (na maioria dos casos). (MDN, 2024).

A Tabela1 mostra as principais diferenças entre o JavaScript e o JSON.

Tabela 1 – Diferenças entre o JavaScript e o JSON

Tipos e valores JavaScript	Diferença para o JSON
Objetos e Arrays	Os nomes das propriedades devem ser strings com aspas duplas; as vírgulas à direita são proibidas.
Números	Zeros à esquerda são proibidos; um ponto decimal deve ser seguido por pelo menos um dígito.
Strings	Apenas um conjunto limitado de caracteres pode ser escapado; certos caracteres de controle são proibidos; o separador de linha Unicode (U+2028) e o separador de parágrafo (U+2029) caracteres são permitidos; strings devem ter aspas duplas.

Fonte: (MDN, 2024).

são formados por estas cadeias denominadas strings,
¹⁹ Do inglês: Nulo. Neste contexto é um valor especial do JavaScript para representar a nulidade de um objeto/variável.
²⁰ Do inglês: Indefinido. Neste contexto é um tipo de dado do JavaScript para variáveis indefinidas.

2.1.2 Bibliotecas e Frameworks

2.1.2.1 ReactJs

Figura 5 – Logotipo do React



Fonte: React Dev, disponível em: <https://react.dev/>

"Biblioteca para interfaces de usuário Web e nativas". O React é uma biblioteca de JavaScript criada pelo Facebook para solucionar desafios de manutenção e escalabilidade em suas aplicações. No início de 2011, a equipe de desenvolvedores do Facebook enfrentava dificuldades em lidar com o crescimento da aplicação de anúncios, que estava se tornando cada vez mais complexa e difícil de ser mantida. O aumento no número de membros da equipe e de funcionalidades estava afetando negativamente os processos da empresa. Com tantas atualizações em cascata, a aplicação estava se tornando lenta e difícil de ser atualizada sem falhas. (MORAIS, 2021).

Para resolver esses problemas, Jordan Walke, engenheiro do Facebook, propôs uma solução inovadora. Ele sugeriu levar o XHP, uma versão do PHP, para o navegador usando JavaScript. O XHP era uma tecnologia desenvolvida para minimizar ataques de Cross-Site Scripting (XSS) em aplicações Web dinâmicas. No entanto, ele não era capaz de lidar com o grande número de requisições necessárias para esse tipo de aplicação. Com o apoio de sua equipe de gerenciamento, Jordan Walke conduziu um experimento de seis meses para explorar essa ideia. O resultado desse experimento foi o surgimento do ReactJS. (MORAIS, 2021).

O ReactJS revolucionou o desenvolvimento de interfaces de usuário ao introduzir o conceito de componentes reutilizáveis e a abordagem de renderização virtual. Com a utilização de componentes, os desenvolvedores podiam criar e reutilizar peças de interface independentes e isoladas, o que simplificava o desenvolvimento e manutenção do código. Além disso, a renderização virtual permitia atualizações de interface eficientes, otimizando o desempenho da aplicação. O ReactJS foi lançado como um

software de código aberto em 2013, permitindo que desenvolvedores de todo o mundo o utilizassem em seus projetos. (MORAIS, 2021).

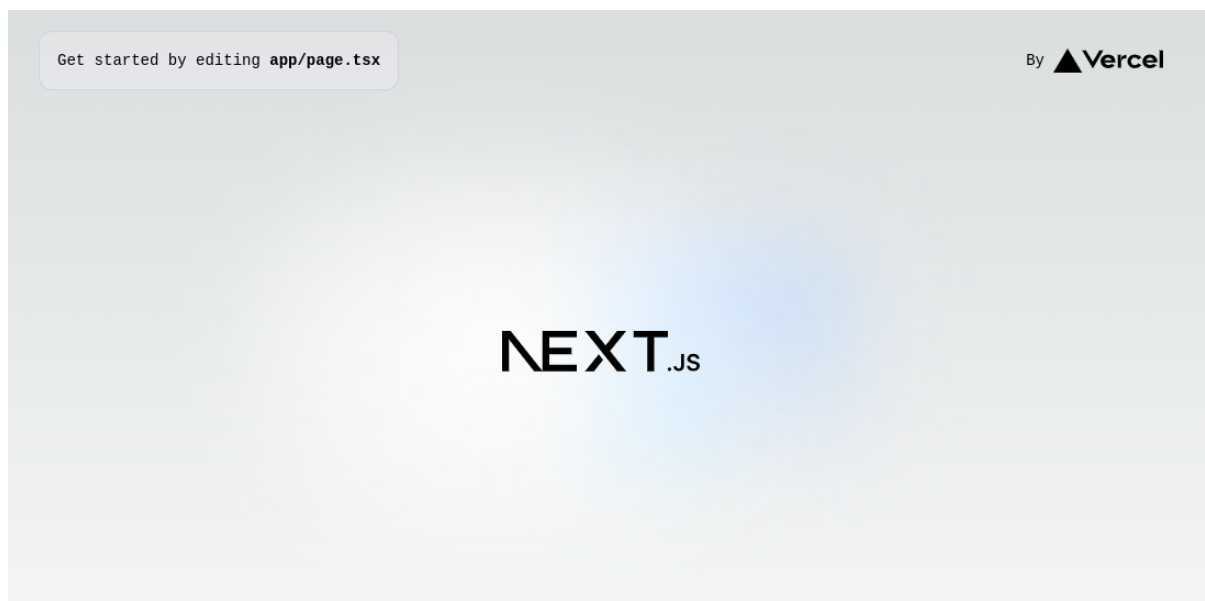
Desde então, o React ganhou uma imensa popularidade e se tornou uma das principais ferramentas para o desenvolvimento de interfaces de usuário em aplicações web. Sua abordagem declarativa, que permite descrever como a interface deve ser exibida com base no estado da aplicação, simplifica a construção de interfaces complexas. Além disso, a capacidade de reutilização de componentes economiza tempo e esforço durante o desenvolvimento. O React também influenciou o desenvolvimento do React Native, uma versão da biblioteca voltada para a criação de aplicativos móveis multiplataforma. Com a ajuda de uma grande comunidade de desenvolvedores e empresas, o ecossistema do React continua a evoluir e fornecer soluções inovadoras para o desenvolvimento de interfaces de usuário modernas e eficientes. (MORAIS, 2021).

O React teve seus primeiros sinais em 2010, quando o Facebook introduziu o XHP na sua stack de PHP, permitindo a criação de componentes compostos. Em 2011, Jordan Walke criou o FaxJS, protótipo inicial do React, que foi desenvolvido para resolver os desafios de suporte aos anúncios do Facebook. Em 2012, o Instagram foi adquirido pelo Facebook e expressou interesse em adotar o React. Isso levou o Facebook a dissociar o React da empresa e torná-lo open source. Em 2013, ocorreu o lançamento oficial do React, mas inicialmente enfrentou resistência da comunidade de desenvolvedores. No entanto, uma "turnê do React" foi realizada para conquistar os não adeptos. (MORAIS, 2021).

No ano seguinte, o React começou a ganhar reputação e confiança. O *React Developer Tools*²¹ e o *React Hot Reloader*²² foram lançados, trazendo melhorias no desenvolvimento e na experiência do usuário. Em 2015, o React se estabeleceu como uma tecnologia estável, com empresas como Netflix e Airbnb adotando-o. O Redux, responsável pelo gerenciamento de estado, foi lançado, e o React Native expandiu-se para o desenvolvimento de aplicativos móveis para Android. (MORAIS, 2021).

Atualmente, o React continua evoluindo, com o lançamento de novas funcionalidades e recursos para melhorar o desenvolvimento de aplicações. Iniciativas de SSR (*Server Side Rendering*)²³, e o foco em componentes funcionais são algumas das áreas de desenvolvimento. O React permanece como uma biblioteca consolidada no mercado de Front- End, sendo amplamente adotado por grandes empresas em todo o mundo. (MORAIS, 2021).

Figura 6 – Logotipo do NextJs



Fonte: Next.Js, disponível em: <https://nextjs-template.vercel.app/>

2.1.2.2 NextJs

O NextJs é uma framework em ReactJs voltada à construção de aplicações Web tando na parte do Front-End quanto no Back-End. Com NextJs, utiliza-se os componentes em React para construir as interfaces de usuário, com o NextJs provendo recursos adicionais e otimizações. (NEXTJS, 2024).

NextJs também se encarrega de todas as configurações necessárias do React, como o processo de enpacotamento²⁴, compilação e etc... Permitindo ao desenvolvedor apenas focar no desenvolvimento da aplicação em si. (NEXTJS, 2024).

Devido à natureza desta framework, O NextJs é um pilar que aparece tanto no Back-End quanto no Front-End. Estas duas frentes serão abordadas com a utilização desta ferramenta, aproveitando ao máximo os recursos fornecidos pela mesma. Os principais recursos oferecidos pelo NextJs são:

- Roteamento: O NextJs provê um roteamento, (que é basicamente a navegação por páginas dentro do app), baseado no sistema de arquivos do Sistema Operacional. Os arquivos em pastas do projeto são mapeados para links, que fornecem os componentes de servidor com suporte a layouts²⁵, rotas aninhadas,

²¹ Ferramentas de Desenvolvimento React

²² Carregamento e recarregamento ultra rápido React

²³ Do inglês: Rederização do Lado do Servidor

²⁴ Em inglês: Bundling. Um Bundle para a Web, por exemplo, junta todos os códigos e recursos em um pacote otimizado para ser distribuído.

²⁵ Layouts são como templates que são comuns às páginas roteadas. Ajudam no processo de reaproveitamento de componentes pois eles podem ser estendidos às páginas, que herdam características destes layouts.

estados de carregamento, manipulação de erros, entre outros...

- **Renderização:** NextJs fornece renderização do lado do cliente e do lado do servidor com componentes de cliente, e componentes de servidor.
- **Busca de dados:** Há um processo de busca de dados simplificado com o uso de *async/await*²⁶ nos componentes de servidor, além de uma API²⁷ expandida para a memorização das requisições, *caching*²⁸ de dados e revalidação.
- **Estilização:** Suporte para os métodos preferidos de estilização. Com inclusão de: Módulos CSS, *Tailwind* CSS, e CSS-in-JS.
- **Otimizações:** Otimizações de scripts, imagens e fontes são também fornecidos para aprimorar o núcleo do aplicativo e a experiência de usuário.
- **TypeScript:** Suporte total ao TypeScript, com uma melhor checagem de tipos e compilação eficiente.

(NEXTJS, 2024).

²⁶ Recurso do JavaScript para lidar com a execução de código assíncrono.

²⁷ Do inglês: Interface de Programação de Aplicações. É uma forma na qual dois ou mais aplicativos ou componentes de de computador se comunicam entre si. É uma interface de software que oferece um serviço para outras partes do mesmo ou de outros softwares. (REDDY, 2011).

²⁸ O processo de caching é o ato de armazenar informações que são acessadas frequentemente de maneira que seu acesso se torne mais rápido. Neste contexto, o resultado de uma requisição pode ser armazenado em cache para que não seja necessário consultar o servidor novamente quando a mesma informação for requisitada.

2.1.2.3 EditorJs

Figura 7 – Logotipo do EditorJs



Fonte: Editor.Js, disponível em: <https://editorjs.io/>

"Editor livre em blocos com saída universal em JSON". O EditorJs é um rico editor de texto em blocos que oferece uma experiência de edição intuitiva e versátil. Tudo o que é feito no EditorJs no fim é transformado em um arquivo JSON ao invés de um documento de marcação em HTML. Essa abordagem deixa o processo mais simples para os desenvolvedores no sentido de projetarem suas próprias integrações. Assim, o EditorJs pode ser aplicado a diversas plataformas. (EDITORJS,).

São recursos do EditorJs:

- Dados de saída limpos
- API baseada em plugins
- Código aberto

O espaço de trabalho do EditorJs consiste em blocos separados, como: Parágrafos; títulos; listas; etc... Cada um deles independentes entre si, com muitos outros recursos como: Copiar e colar; seleção de vários blocos; e entre outros que funcionam de forma familiar a outras ferramentas. (EDITORJS,).

O conceito chave do EditorJS é sua API, na qual todas as unides funcionais do editor são providas através de plugins externos que fazem uso da mesma. Assim, o núcleo do EditorJs fica sendo mais abstrato e poderoso, de modo que o desenvolvedor

possa implementar diversos desafios com a criação de seus próprios plugins. (EDITORJS,).

2.2 Do Back-End

2.3 O processo de Parsing

REFERÊNCIAS

ABNT. *Quem somos? ABNT - Associação Brasileira de Normas Técnicas*. 2021. [Online; acessado em 15-Maio-2023]. Disponível em: <<http://www.abnt.org.br/institucional/sobre>>.

AMAZON; AWS. *O que é uma framework em programação e engenharia?* [Online; acessado em 25-Maio-2024]. Disponível em: <<https://aws.amazon.com/pt/what-is/framework/>>.

BRASIL. Lei nº 12.965, de 23 de abril de 2014. dispõe sobre os direitos civis na internet. *Diário Oficial da União*, 2014.

CASTRO, C. M. *Como redigir e apresentar um trabalho científico*. [S.l.]: São Paulo: Pearson Prentice Hall, 2011.

EDITORJS. Base concepts. [Online; acessado em 30-Maio-2024]. Disponível em: <<https://editorjs.io/base-concepts/>>.

MACHADO, E. Hardware e software. Enciclopédia Significados. Disponível em: <<https://www.significados.com.br/hardware-e-software/s>>.

MDN. Design responsivo. 2023. [Online; acessado em 28-Maio-2024]. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/CSS/CSS_layout/Responsive_Design>.

MDN. Json. 2024. [Online; acessado em 30-Maio-2024]. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON>.

MDN, M. D. N. Css. 2023. [Online; acessado em 16-Maio-2023]. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>.

MDN, M. D. N. Javascript. 2023. [Online; acessado em 16-Maio-2023]. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>.

MEDEIROS, J. B. *Redação científica: a prática de fichamentos, resumos, resenhas*. 11st. ed. [S.l.]: São Paulo: Atlas, 2012.

MOLGADO, V. *A Evolução da Web: linha do tempo interativa da História da internet*. 2016. [Online; acessado em 15-Maio-2023]. Disponível em: <<https://labvis.eba.ufrj.br/a-evolucao-da-web-linha-do-tempo-interativa-da-historia-da-internet/>>.

MORAIS, P. A história do react! Medium, 2021. [Online; acessado em 29-Maio-2024]. Disponível em: <<https://medium.com/@ppternunes/a-hist%C3%B3ria-do-react-ba346c416fe1>>.

NEXTJS. Introduction. 2024. [Online; acessado em 29-Maio-2024]. Disponível em: <<https://nextjs.org/docs>>.

PUC-GO. Coordenação de tcc. manual para elaboração de trabalho de conclusão de curso. Pontifícia Universidade Católica de Goiás, 2022.

REDDY, M. *API Design for C++*. [S.l.]: Elsevier Science, 2011. ISSN 9780123850041.

SANTOS, I. R. As dificuldades na construção do trabalho de conclusão de curso: Percepção de estudantes egressos do curso de ciências contábeis. 2020. 69. trabalho de conclusão de curso (graduação) - ciências contábeis. Universidade Federal da Paraíba, João Pessoa, 2020.

SEVERINO, A. J. *Metodologia do trabalho científico*. 24st. ed. [S.l.]: Cortez, 2017.

SILVA, M. O. da; VITORIA, M. I. C. A experiência de escrita no trabalho de conclusão de curso: Percepções de alunos de um curso superior de tecnologia em gestão em recursos humanos. CAMINE: Caminhos da Educação, Franca, 2014. ISSN 2175-4217.

TOTVS. *Front end: O que é, como funciona e qual a importância*. 2021. [Online: acessado em 25-Maio-2024]. Disponível em: <<https://www.totvs.com/blog/developers/front-end/>>.

TYPESCRIPT. *TypeScript*. [Online: acessado em 29-Maio-2024]. Disponível em: <<https://www.typescriptlang.org/>>.

VITORIANO, D. *O que é a Web Moderna*. 2019. Medium. [Online: acessado em 15-Maio-2023]. Disponível em: <<https://blog.danvitoriano.com.br/o-que-%C3%A9-a-web-moderna-b01e4df9a565>>.

W3C, W. W. W. C. *HTML*. 2023. [Online: acessado em 16-Maio-2023]. Disponível em: <<https://html.spec.whatwg.org/multipage>>.

W3SCHOOLS. *TypeScript Introduction*. 2024. [Online: acessado em 29-Maio-2024]. Disponível em: <https://www.w3schools.com/typescript/typescript_intro.php>.