

UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS
Faculdade de Computação e Engenharia Elétrica
Bacharelado em Sistemas de Informação

COMPUTER INSTRUCTOR: UM APLICATIVO MOBILE PARA AUXILIAR O
APRENDIZADO DA INFORMÁTICA

Brayan Alves Martins Ferreira

Marabá-PA

2021

Brayan Alves Martins Ferreira

**COMPUTER INSTRUCTOR: UM APLICATIVO MOBILE PARA AUXILIAR O
APRENDIZADO DA INFORMÁTICA**

Trabalho de Conclusão de Curso, apresentado à Universidade Federal do Sul e Sudeste do Pará, como parte dos requisitos necessários para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador(a):

Profa. Dra. Zenaide Carvalho da Silva

Marabá-PA

2021

Este trabalho é dedicado à minha família e amigos que contribuíram direta e indiretamente no desenvolvimento do mesmo.

AGRADECIMENTOS

Inicialmente, gostaria de agradecer aos meus pais, Silvio e Irenice, e meu irmão, Ckemerom, que sempre me apoiaram com tudo que eu precisei durante a minha vida.

Aos amigos que a vida me deu, principalmente Bianca Assunção, Henryque Augusto, Maria Eduarda e Vitor Novaes, que de alguma forma fizeram parte dessa jornada, sempre presentes nos filmes ruins, acompanhado do tradicional balde de frango.

Aos amigos da faculdade, em especial dois membros do "Time reg", companheiros de trabalhos, de jogos, de estudos, Andréia Letícia e Gustavo Felipe, e ao membro fantasma, Gustavo Carvalho, assim como também Evander Oliveira, Michel Victor e José Vinícius.

A todos os Professores da Faculdade de Computação e Engenharia Elétrica, em especial minha Orientadora Zenaide Carvalho, pela sua sabedoria, e por ser atenciosa e paciente.

”O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.”

(José de Alencar)

RESUMO

O constante avanço das tecnologias, ocasionou uma revolução nos meios de aprendizagem, possibilitando que as pessoas busquem o conhecimento de diferentes maneiras. Dessa forma, abordagens inovadoras quanto a aprendizagem, podem contribuir para um estudo **simplificado**, permitindo um melhor aprendizado. Na educação, a informática traz benefícios aos alunos que têm acesso a tecnologias e ferramentas, pois contribui na vida escolar e no acesso à informação. Nesse contexto, o presente projeto tem por objetivo trazer uma abordagem diferenciada para aprendizagem dos estudantes de informática, tornando possível consultas rápidas de acordo com necessidades ou questionamentos momentâneos relacionados ao tema informática através de um aplicativo. Portanto, como resultado desse trabalho foi desenvolvido um aplicativo denominado *Computer Instructor* para ajudar no processo de aprendizagem da informática básica para iniciantes, visando permitir mais agilidade e facilidade na busca de conhecimento dessa área, juntamente a um sistema *Web* e um Banco de dados, responsáveis pela administração do conteúdo.

Palavras-chave: Aplicativo *Mobile*. Sistema *web*. Banco de dados.

ABSTRACT

The constant advancement of technologies has brought about a revolution in the means of learning, enabling people to seek knowledge in different ways. Thus, innovative approaches to learning can contribute to a simplified study, allowing for better learning. In education, information technology brings benefits to students who have access to technologies and tools, as it contributes to school life and access to information. In this context, this project aims to bring a differentiated approach to the learning of computer science students, making it possible for quick consultations according to needs or momentary questions related to the computer theme through an application. Therefore, as a result of this work, an application called Computer Instructor was developed to help in the learning process of basic information technology for beginners, aiming to allow more agility and ease in the search for knowledge in this area, together with a responsible Web system and a Database. by content management.

Keywords: Mobile Application, Web System, Database.

LISTA DE ILUSTRAÇÕES

Figura 1 – Processo de Criação do aplicativo	16
Figura 2 – Modelagem do Back-end	27
Figura 3 – Computer Instructor - Modelagem	28
Figura 4 – Wireframe - telas iniciais	30
Figura 5 – Wireframe - telas de módulos	30
Figura 6 – Wireframe - telas de unidades	31
Figura 7 – Wireframes - Mobile	31
Figura 8 – Back-end Computer Instructor	32
Figura 9 – Diagrama Entidade-relacionamento	34
Figura 10 – Mobile - Página inicial	35
Figura 11 – Mobile - Página de informações	36
Figura 12 – Mobile - Página de módulos	37
Figura 13 – Mobile - Página de unidades	38
Figura 14 – Mobile - Página de unidades 2	39
Figura 15 – Mobile - Página de conteúdo	40
Figura 16 – Mobile - Página de conteúdo 2	41
Figura 17 – Mobile - Página de conteúdo 3	42

LISTA DE TABELAS

Tabela 1 – Análise relacionada aos trabalhos correlatos.	24
Tabela 2 – Requisitos Funcionais	26
Tabela 3 – Requisitos Não Funcionais	26
Tabela 4 – Caso de Uso Expandido	29

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i>
APP	Aplicativo
BPMN	<i>Business Process Model and Notation</i>
CRUD	<i>Create, Read, Update and Delete</i>
ER	Entidade-Relacionamento
HQ	História em quadrinhos
HTTP	<i>Hypertext Transfer Protocol</i>
IHC	Interação Humano-Computador
OO	Orientação a Objetos
PID	Projeto de Inclusão Digital
REST	<i>REpresentational State Transfer</i>
TIC	Tecnologia da Informação e Comunicação
UML	Linguagem de Modelagem Unificada
UNIFESSPA	Universidade Federal do Sul e Sudeste do Pará
WEB	<i>World Wide Web</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	13
1.2	Objetivo Geral	14
1.3	Objetivos Específicos	14
1.4	Estrutura do trabalho	14
2	MATERIAIS E MÉTODOS	16
2.1	Wireframes	16
2.2	Ferramentas Utilizadas	17
2.2.1	Figma	17
2.2.2	Visual Studio Code	17
2.2.3	Insomnia	17
2.2.4	Node.js	17
2.2.5	JavaScript	18
2.2.6	React Native	18
2.2.7	Expo	18
2.2.8	LucidChart	18
2.2.9	Heflo	18
3	REVISÃO DE LITERATURA	19
3.1	Engenharia de software	19
3.1.1	Engenharia de Requisitos	19
3.1.2	Modelagem de Sistemas	19
3.2	Diagramas UML	20
3.3	Interface Humano Computador	20
3.4	Tecnologia Móvel	21
4	TRABALHOS CORRELATOS	23

5	APLICATIVO MOBILE COMPUTER INSTRUCTOR	25
5.1	Coleta de requisitos	25
5.2	Modelagem de requisitos	26
5.2.1	Modelagem Back-end	26
5.2.2	Modelagem Computer Instructor	27
5.2.3	Casos de Uso Expandido	28
5.2.4	Prototipagem	29
5.2.4.1	Front-end	29
5.2.4.2	Mobile	31
5.3	Arquitetura do Computer Instructor	31
5.4	Implementação do Computer Instructor	33
5.4.1	Back-end	33
5.4.2	Mobile	34
6	CONSIDERAÇÕES FINAIS	43
6.1	Trabalhos Futuros	43
	REFERÊNCIAS	44

1 INTRODUÇÃO

O constante avanço das tecnologias, ocasionou em uma revolução nos meios de aprendizagem, possibilitando que as pessoas busquem o conhecimento de diferentes maneiras. Dessa forma, abordagens inovadoras quanto a aprendizagem, como o *M-Learning*, um ambiente para alunos e professores que possibilita a aprendizagem através do auxílio de dispositivos móveis, podendo acarretar em um estudo **simplificado**, permitindo um melhor aprendizado.

O ambiente escolar tradicional geralmente se caracteriza por um aprendizado exaustivo e pouco interativo. De forma diferente, às aplicações móveis podem construir o conhecimento despertando o interesse das crianças, jovens e adultos sobre determinado assunto (ABREU; SOUSA; LACERDA, 2017).

Modernizar os métodos de ensino é um desafio muito grande para os professores. Acompanhar o mundo tecnológico requer certo conhecimento de utilização dessas ferramentas. Entretanto, com foco, persistência e direcionamento, é certo que qualquer pessoa passa a ter o domínio dessas tecnologias (ANDRADE; OLIVEIRA, 2017).

Em uma sociedade complexa como a nossa, cujas práticas **sócias** têm sido influenciadas, cada vez mais, pelas novas tecnologias de informação, comunicação e expressão, um dos desafios é ser o gestor do próprio tempo (SILVA, 2019).

Cada vez mais, um maior número de instituições educacionais emprega *softwares* educativos como facilitadores do processo de ensino e aprendizagem (BRAGA, 2006). Nesse contexto, foi desenvolvido um aplicativo para ajudar no processo de aprendizagem da informática básica para iniciantes, visando permitir mais agilidade e facilidade na busca de conhecimento dessa área.

O aplicativo tem como característica a simplicidade, cujo objetivo é propiciar facilidade na usabilidade, possibilitando assim, uma praticidade sem a necessidade de uma introdução sobre o funcionamento do mesmo, tornando-o intuitivo.

1.1 Motivação

Um estudo realizado através de entrevistas com 40 professores e 10 diretores, aponta que 85% dos educadores entrevistados consideram que é importante o uso das novas tecnologias na Educação, pois contribui na sua vida escolar, no acesso à informação e para o futuro como cidadão (ANDRADE; OLIVEIRA, 2017).

Na educação, a informática vem beneficiando os alunos que têm acesso a suas ferramentas, pensando nisso, o projeto de extensão informática educativa no ensino fundamental da rede pública, foi criado com o intuito de transferir conhecimentos sobre informática básica e lógica de programação para os alunos do ensino fundamental (ALVES

et al., 2019).

Diante disso, a partir de estudos relacionados às citações anteriores, surgiu a motivação para sustentar o presente projeto, originando a ideia da necessidade de explicação de processos ou atividades simples dentro da área de informática, tendo em vista que os mesmos podem gerar atraso ou desmotivação na etapa de aprendizagem.

Dessa maneira, surgiu a ideia de desenvolver um aplicativo para servir de guia de bolso, trazendo soluções rápidas e de fácil compreensão para o usuário, relacionados a conceitos e funcionalidades voltados a informática, proporcionando assim uma abordagem diferente de aprendizado, com o intuito de gerar estímulos por busca de novos conhecimentos.

1.2 Objetivo Geral

Desenvolver uma aplicação mobile que possibilita agilidade e facilidade do aprendizado do estudante de informática, apresentando conceitos e funcionalidades, proporcionando ao mesmo um maior entusiasmo em busca de novos conhecimentos relacionados a essa área de tecnologia.

1.3 Objetivos Específicos

- Realizar a modelagem e desenvolvimento do aplicativo proposto.
- Realizar a Modelagem dos *Wireframes* do aplicativo.
- Apresentar o processo de desenvolvimento da aplicação e banco de dados.

1.4 Estrutura do trabalho

O presente trabalho está estruturado em 6 capítulos organizados da seguinte maneira:

- **Capítulo 2:** Apresenta os materiais e métodos empregados no desenvolvimento deste trabalho, assim como seu planejamento.
- **Capítulo 3:** Apresenta uma revisão de literatura com os principais conceitos envolvendo a temática do trabalho em desenvolvimento de um sistema voltado à educação.
- **Capítulo 4:** Apresenta alguns trabalhos relacionados com a proposta do trabalho, destacando suas principais contribuições e temáticas abordadas.

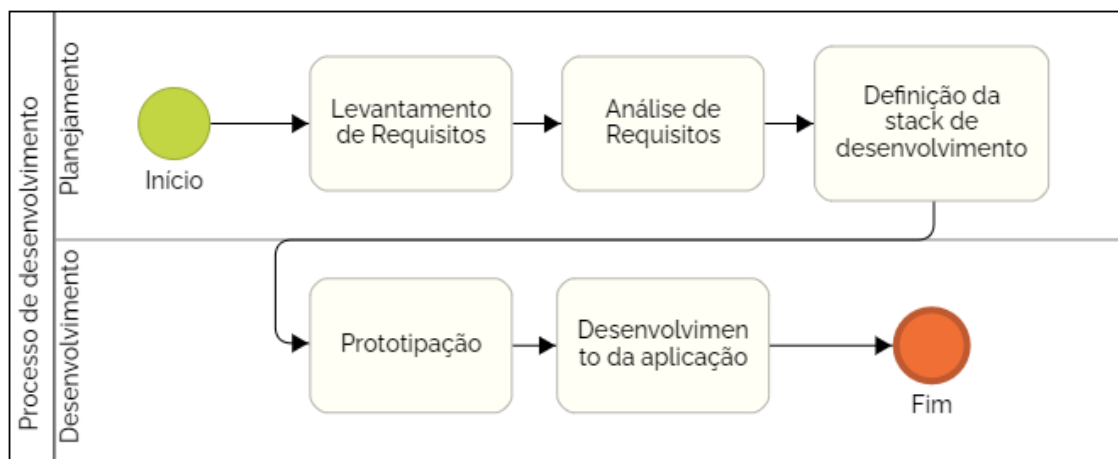
- **Capítulo 5:** Discorre sobre a proposta criada, fazendo pontuações sobre a forma como o sistema está estruturado, e sobre a forma que as informações interagem na aplicação.
- **Capítulo 6:** Apresenta as considerações finais referentes a este trabalho.

2 MATERIAIS E MÉTODOS

Para o processo de desenvolvimento de sistemas ou aplicativos, inicialmente, deve-se realizar o planejamento, etapa onde é formada as ideias, resultando assim, em uma base para os levantamentos dos requisitos, possibilitando uma análise de viabilidade, onde será definido tudo o que será realizável, que por fim tornará possível o fechamento do escopo, originando uma percurso a ser seguido nesse processo.

Nesse contexto, este capítulo apresentará as ferramentas e métodos utilizados para a realização deste trabalho, incluindo os processos de desenvolvimento, bem como as estruturas utilizadas, além das especificações técnicas. A **imagem a seguir** descreve o fluxo do processo de criação utilizado no desenvolvimento do aplicativo, ilustrando sobre a etapa de planejamento e desenvolvimento.

Figura 1 – **Processo de Criação do aplicativo**



HEFLO

Fonte: O Autor (2021).

A Seção 2.1 descreve sobre a etapa de prototipação da interface do sistema e da estratégia utilizada para tanto. E por último, a Seção 2.2 discorre sobre tecnologias que foram utilizadas na criação da ferramenta.

2.1 Wireframes

Wireframes são esboços simples de telas de produtos digitais, como sites e aplicativos. O intuito é estruturar e validar ideias, por isso os *wireframes* não contam com detalhes como cores, fontes, ícones e imagens (AELA.IO, 2019).

Dessa forma, os *wireframes* conseguem demonstrar de forma direta a arquitetura final de uma interface, posicionando os elementos de forma simples e organizada. Portanto,

o *wireframe* reflete apenas o necessário da proposta de uma interface digital (AELA.IO, 2019).

Contudo, foi utilizado esta ferramenta para criar esboços de telas tanto quanto do sistema de cadastro, onde é possível controlar o conteúdo do mesmo, quanto do aplicativo, contendo todas suas funcionalidades.

2.2 Ferramentas Utilizadas

Nas seções a seguir são apresentadas as ferramentas de *software* utilizadas no processo de desenvolvimento do sistema.

2.2.1 Figma

O *Figma*¹ é uma ferramenta de design de interface na qual todo o trabalho é feito através do navegador, logo ela é compatível com *Windows*, *Linux*, *Chrome* e *Mac* (INTERATIVA, 2019). Essa ferramenta foi utilizada para desenvolver os protótipos em forma de *wireframes*.

2.2.2 Visual Studio Code

O *Visual Studio Code*² é um editor de código-fonte leve, mas poderoso, que roda em sua área de trabalho e está disponível para *Windows*, *MacOS* e *Linux* (CODE, 2021). Esta ferramenta foi utilizada no desenvolvimento do sistema, possibilitando a codificação do banco de dados, interface e funcionalidades.

2.2.3 Insomnia

O *Insomnia*³ é um aplicativo gratuito para *desktop* que combina uma interface fácil de usar com funcionalidades avançadas, como auxiliares de autenticação, geração de código e variáveis de ambiente (INSOMNIA, 2021). Foi utilizado para testes e implementações no banco de dados, também foi utilizado para realizar o gerenciamento de conteúdo.

2.2.4 Node.js

*Node.js*⁴ é uma tecnologia usada para executar código *JavaScript* fora do navegador. Com ele podemos construir aplicações *web* em geral, desde *websites* até APIs e microserviços (DEVMEDIA, 2021a). Essa ferramenta foi utilizada no desenvolvimento do banco de dados e do aplicativo.

¹https://medium.com/@Sirius_/figma-uma-nova-ferramenta-para-design-de-interface-que-est%C3%A1-ganhando-o-mercado-sirius-interativa-2e78e0905b44

²<https://code.visualstudio.com/learn>

³<https://support.insomnia.rest/article/157-getting-started-with-insomnia>

⁴<https://www.devmedia.com.br/guia/node-js/40312#introducao-ao-node-js>

2.2.5 JavaScript

O *JavaScript*⁵ é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de *script* para páginas *Web*, mas usada também em vários outros ambientes sem *browser*, tais como *Node.js*, *Apache CouchDB* e *Adobe Acrobat* (MOZILLA, 2021). Essa linguagem foi usada no desenvolvimento do sistema.

2.2.6 React Native

O *React Native*⁶ é um *framework JavaScript* criado para construir a interface do usuário em aplicativos móveis. Baseado no *React*, o *React Native* é a solução para criar aplicativos nativos tanto para *Android* quanto para *iOS* (DEVMEDIA, 2021b). Essa ferramenta foi utilizada na produção da aplicação *mobile*.

2.2.7 Expo

*Expo*⁷ é uma estrutura e uma plataforma para aplicações *React* universais. É um conjunto de ferramentas e serviços criados em torno de plataformas *React Native* e nativas que ajudam a desenvolver, construir, implantar e iterar rapidamente (EXPO, 2021). Essa ferramenta foi utilizada na produção da aplicação *mobile*.

2.2.8 LucidChart

*LucidChart*⁸ é um aplicativo descrito como uma ferramenta de criação de diagramas inteligentes (LUCID, 2021). Essa ferramenta foi utilizada na elaboração dos diagramas do sistema.

2.2.9 Heflo

*Heflo*⁹ é solução BPM para Modelagem e Automatização de Processos de Negócio (VENKI, 2021). Essa ferramenta foi utilizada na criação dos diagramas de processos do sistema.

⁵<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁶<https://www.devmedia.com.br/react-native/>

⁷<https://docs.expo.io/>

⁸<https://lucid.co/pt/produto/lucidchart>

⁹<https://www.venki.com.br/ferramenta-bpm/heflo/>

3 REVISÃO DE LITERATURA

Neste capítulo serão apresentados conceitos que estão associados ao entendimento e desenvolvimento do presente trabalho. A Seção 3.1 irá explicar os principais conceitos da Engenharia de *Software*, assim como sua importância no desenvolvimento de uma aplicação. A Seção 3.2 discorre sobre Linguagem de modelagem unificada, bem como conceitos dos diagramas utilizados. A Seção 3.3 trata sobre Interface Humano Computador, assim como alguns princípios de usabilidade. Já a Seção 3.4 discorrerá sobre as especificações relacionadas à Tecnologia Móvel.

3.1 Engenharia de software

Segundo Pressman (2009), a Engenharia de *Software* é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de *software*. Para Sommerville (2011), a Engenharia de *Software* tem foco em todos os aspectos da produção de *software*, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado. Desse modo, pode-se entender a engenharia como uma série de processos e métodos que possibilitam o gerenciamento e desenvolvimento de *software*.

Nesse sentido, o presente trabalho fundamenta-se nos preceitos estabelecidos na engenharia de *software*. Dentre os diversos fundamentos existentes, foram acordados a Engenharia de Requisitos retratado na Seção 3.1.1 juntamente a Modelagem de Sistemas especificado na Seção 3.1.2.

3.1.1 Engenharia de Requisitos

Segundo Sommerville (2011), engenharia de requisitos é o processo de descobrir, analisar, documentar e verificar as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Desse modo, pode-se compreender que a Engenharia de Requisitos é responsável por identificar e transformar necessidades dos clientes em algo benéfico, em prol de validar os objetivos de acordo com a finalidade determinada.

Assim sendo, a engenharia de requisitos no presente trabalho tem como objetivo garantir a qualidade final do sistema, através do levantamento dos requisitos de maneira inteligível.

3.1.2 Modelagem de Sistemas

Sommerville (2011) descreve a modelagem de sistema como processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão

ou perspectiva, diferente do sistema. Assim sendo, a Modelagem de Sistemas é responsável por contribuir com a compreensão das funcionalidades por meio de ilustrações.

Dessa forma, o desenvolvimento da solução proposta fundamenta-se nas concepções relacionadas a Engenharia de *Software*. Assim sendo, a modelagem do sistema proporciona a criação de uma ferramenta que atenda as necessidades das partes interessadas.

3.2 Diagramas UML

A Linguagem de **modelagem unificada** (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de *software* complexos, tanto estruturalmente quanto para comportamentos (LUCIDCHART, 2021b).

A UML usa os pontos fortes destas três abordagens para apresentar uma metodologia mais consistente e mais fácil de usar. Além disso, representa as melhores práticas para desenvolver e documentar aspectos diferentes da modelagem de *software* e sistemas de negócios (LUCIDCHART, 2021b).

Como muitos sistemas são concebidos a partir da aplicação de práticas e técnicas de OO, a elaboração de documentos modelando os componentes esperados é feita atualmente a partir de diagramas UML (GROFFE, 2013).

Diante deste contexto, foram utilizados os seguintes tipos de UML:

- **Diagrama de Caso de Uso:** Utilizado para descrever as principais funções do aplicativo, bem como a relação dos usuários com o sistema.
- **Diagrama de Caso de Uso Expandido:** Utilizado para descrever o fluxo do caso de uso detalhadamente.
- **Diagrama Entidade-relacionamento:** Utilizado na projeção do banco de dados da aplicação para descrever os objetos envolvidos e seus relacionamentos.

3.3 **Interface Humano Computador**

Segundo Hewett et al. (1992 apud BARBOSA; SILVA, 2010), IHC é uma disciplina interessada no projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos relacionados a esse uso.

Nesse contexto, a IHC visa qualidade através da usabilidade, que está ligada a facilidade de aprendizado, juntamente a satisfação provida através do uso, além da acessibilidade que objetiva remover barreiras na usabilidade por parte do usuário de um sistema, assim como também está ligado comunicabilidade, que tem como propósito

viabilizar uma interface intuitiva, onde o usuário consiga ver e compreender o funcionamento do sistema.

Nielsen (1994 apud BARBOSA; SILVA, 2010) definiu engenharia de usabilidade como um conjunto de atividades que devem ocorrer durante todo o ciclo de vida do produto, ressaltando que muitas delas ocorrem nos estágios iniciais do projeto, antes que a interface com usuário em si seja projetada.

Dessa forma, Nielsen (1994 apud BARBOSA; SILVA, 2010) propõe um conjunto de atividades em seu ciclo de vida, dentre os quais foram utilizados:

- **Conheça seu usuário:** Essa etapa é vista como primeiro passo, que consiste em compreender as necessidades do usuário, ou seja, entender as atividades que o mesmo realiza e identificar pontos de carência a serem resolvidos.
- **Defina as metas de usabilidade:** A definição das metas de usabilidade implica em estipular objetivos de funcionalidades, que serão utilizados de parâmetros para o controle de qualidade.
- **Faça o design coordenado da interface como um todo:** Essa atividade resume-se ao controle do protótipo de interface, realizado por quem tem o conhecimento do projeto em sua totalidade.
- **Faça protótipos:** Nessa atividade ocorre de fato a prototipação de sistemas, de modo que possa ser acordado com os usuários a versão final do mesmo, visando oferecer a melhor opção de acordo com as expectativas das partes interessadas, seja na parte financeira ou na usual.

3.4 Tecnologia Móvel

Alcantara e Vieira (2011) definem a Tecnologia Móvel como a forma de acessar a internet e outros recursos computacionais por meio de dispositivos móveis, tais como, celulares, *iPhone*, *iPod*, *iPad*, *notebooks*, *smart pads*, dentre outros.

De acordo com Turner (2021 apud PESSOA, 2020), os números de utilizadores de *smartphones* têm aumentado exponencialmente ao longo dos anos, verificando-se em 2016 a existência de 2,5 bilhões de utilizadores e estimando-se que este número tenha crescido até aos 3,5 bilhões no ano de 2020.

Desse modo, é possível perceber que a tecnologia móvel tem evoluído cada vez mais ao passar do tempo, criando a oportunidade para o desenvolvimento de diversas ferramentas com diversas funcionalidades. Uma área que pode ser citada dentro das vastas existentes é o *M-Learning* (*mobile learning*), que segundo Telefônica (2021), é uma metodologia de

ensino que proporciona um novo ambiente para alunos e professores, usando dispositivos móveis como plataformas para viabilizar o aprendizado a distância.

Nesse sentido, a solução proposta, tem o *M-Learning* como um de seus maiores fatores, permitindo o acesso a uma gama de conteúdos de uma forma ágil, possibilitando novos aprendizados ou revisões em curtos períodos de tempo.

4 TRABALHOS CORRELATOS

Existem diversos estudos sobre o uso de ferramentas na educação, que podem possibilitar um melhor aprendizado. Dessa maneira, foram feitas buscas por trabalhos para serem utilizados de base que ajudaram no desenvolvimento deste presente trabalho. Dentre estes destacam-se:

- ***Geography History***

Netto et al. (2016) aborda em seu trabalho o desenvolvimento de um aplicativo para o auxílio à disciplina de Geografia, visando proporcionar interatividade entre aluno e disciplina ampliando o acesso aos conteúdos de sala de aula. Netto et al. (2016) detalha em seu trabalho as características do aplicativo *Geography History*, que tem como finalidade fornecer informações como climas, vegetações, relevos, hidrografia, população, moeda, língua oficial, entre outros conhecimentos do país selecionado.

- ***QRFlora***

Abreu, Sousa e Lacerda (2017) discorrem em seu trabalho o desenvolvimento de um aplicativo a ser utilizado como instrumento de auxílio à educação ambiental, podendo ser utilizado para auxiliar professores em disciplinas relacionadas à biologia e em aulas de campo. No aplicativo *QRFlora* é possível identificar espécies arbóreas brevemente cadastradas, sendo necessário realizar a leitura de um *QRcode*, logo após é levado às informações sobre a espécie.

- ***HistoryIcó***

Rolim, Alencar e Costa (2018) dissertam em seu trabalho a criação de um aplicativo instrumento de auxílio à educação patrimonial da cidade Icó-Ceará apresentando curiosidades, contos e a história de 23 patrimônios. No aplicativo *HistoryIcó* é possível acessar as informações através do mapa da cidade, onde possui pontos flutuantes que quando selecionados levam a informação do patrimônio.

- ***m-tutorial***

Rachid e Ishitani (2012) retratam em seu trabalho o desenvolvimento de um aplicativo para criar objetos de aprendizagem, que pode ser realizado em um aparelho celular, mesmo estando em modo *off-line*. O *m-tutorial* pode criar tutoriais que podem conter textos, imagens ou vídeos. No aplicativo também é possível criar tutoriais *off-line*, armazenando o conteúdo internamente, sendo feito a sincronização com um servidor *web* quando houver uma conexão com a internet.

- ***Mobile HQ***

Mühlbeier et al. (2014) desenvolve em seu trabalho a criação de uma ferramenta para aplicação educacional no idioma da língua portuguesa que utiliza HQ e a temática

sobre o Novo Acordo Ortográfico. No aplicativo *Mobile HQ* o usuário pode acessar uma história clicando sobre a mesma, logo após é redirecionado leitura das histórias em quadrinho que acontece quadro a quadro, é possível utilizar botões próximo e anterior para alterar a leitura.

Visando relacionar as funcionalidades de cada ferramenta, foi realizado uma comparação entre as mesmas, incluindo o sistema *Computer Instructor*, ferramenta desenvolvida ao longo deste trabalho, sendo ilustrada na Tabela 1. As características tratadas na tabela foram:

- **Mobile:** Plataforma utilizada utilizada pelo sistema.
- **Off-line:** Capacidade do sistema operar mesmo sem conexão com a internet.
- **Online:** Capacidade do sistema operar conectado à internet, possibilitando controle de conteúdo em tempo real.
- **Controle de conteúdo:** Capacidade do sistema de operar as informações armazenadas, ou seja, realizar cadastros e atualizações de novos conteúdos ou até mesmo a possibilidade de apagar.

Tabela 1 – Análise relacionada aos trabalhos correlatos.

Ferramenta	Mobile	Off-line	Online	Controle de conteúdo
Geography History	X		X	
QRFlora	X		X	X
HistoryIcó	X	X		
m-tutorial	X	X	X	X
Mobile HQ	X	X		
Computer Instructor	X		X	X

Fonte: O Autor (2021).

5 APLICATIVO MOBILE COMPUTER INSTRUCTOR

O aplicativo *Computer Instructor* surgiu a partir da experiência como instrutor no Projeto de Inclusão Digital (PID), onde foi possível perceber que grande parte dos alunos possuíam dúvidas simples que poderiam ser sanadas rapidamente, sem a necessidade da consulta do instrutor, sendo feita uma pesquisa simples e rápida, nesse sentido, foi desenvolvido o aplicativo com o intuito de auxiliar os usuários nessas questões. Dessa forma, a proposta do aplicativo foi disponibilizar conceitos e dicas cadastrados pelo professor.

A Seção 5.1 descreve a etapa de coleta e análise de requisitos do sistema, elaborada de acordo com os pontos de carência identificados. Já a Seção 5.2 irá discorrer sobre a etapa de modelagem através de representações gráficas dos requisitos coletados. A arquitetura do sistema será mostrada na Seção 5.3 que explicará a forma em que o sistema foi estruturado. E por último a Seção 5.4 explana sobre a implementação da ferramenta, mostrando as funcionalidades que foram criadas.

5.1 Coleta de requisitos

Coletar os Requisitos é o processo de determinar, documentar e gerenciar as necessidades e requisitos das partes interessadas a fim de cumprir os objetivos (PMI, 2017). Assim sendo, esta etapa é o processo de entendimento e reconhecimento das carências que as partes interessadas esperam ser solucionadas pelo sistema que será desenvolvido, definindo a funcionalidade do sistema.

Por meio da experiência adquirida como instrutor no Projeto de Inclusão Digital da UNIFESSPA, foi feito o levantamento dos requisitos do sistema, com o intuito de auxiliar na aprendizagem do usuário. Dessa forma, foram feitos levantamentos dos requisitos dividindo-os em Funcionais e Não Funcionais.

Segundo Sommerville (2011), Requisitos Funcionais podem ser entendidos como serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações, já os Requisitos Não Funcionais são restrições aos serviços ou funções oferecidos pelo sistema, muitas vezes, aplicam-se ao sistema como um todo.

A Tabela 2 lista os requisitos funcionais coletados. Dentre os requisitos listados, foram implementados na solução proposta deste trabalho todos os requisitos, por serem requisitos desejáveis.

Tabela 2 – Requisitos Funcionais

ID	Descrição
Req 1	O professor poderá gerenciar o conteúdo do aplicativo.
Req 2	O usuário poderá ver informações sobre o aplicativo.
Req 3	O usuário poderá acessar módulos dos conteúdos.
Req 4	O usuário poderá acessar unidades de cada módulo.
Req 5	O usuário poderá acessar o conteúdo de cada unidade.

Fonte: O Autor (2021).

Os Requisitos Não Funcionais coletados estão relacionados conforme a Tabela 2. Dentre os requisitos listados, foram implementados na solução proposta deste trabalho os requisitos de 1 a 3, por serem requisitos desejáveis, a não implementação do requisito 4 não compromete o funcionamento da ferramenta, já que mesmo sem a interface desenvolvida, ainda é possível realizar o gerenciamento de conteúdo através de outros Apps. A Tabela 3 lista os Requisitos Não Funcionais.

Tabela 3 – Requisitos Não Funcionais

ID	Descrição
Req 1	O sistema deverá estar disponível a todo momento.
Req 2	O Sistema deverá disponibilizar o conteúdo para dispositivos móveis.
Req 3	Deverão ser seguidas as principais heurísticas de Interface Homem Computador de forma a garantir a usabilidade do sistema.
Req 4	O sistema deverá ter uma interface para o gerenciamento do conteúdo.

Fonte: O Autor (2021).

5.2 Modelagem de requisitos

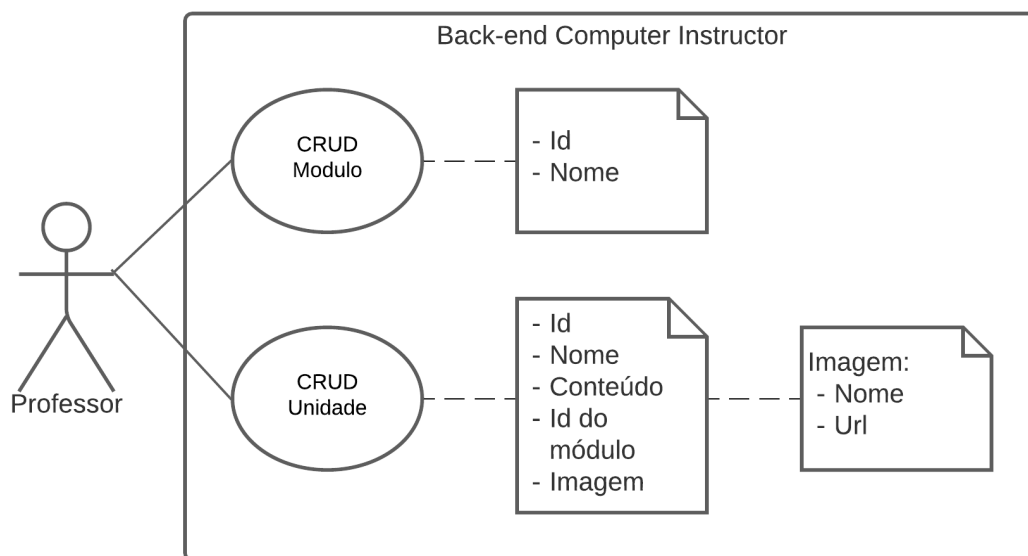
Na linguagem de modelagem unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema (LUCIDCHART, 2021a). Dentro do contexto do *Computer Instructor*, foram criados dois diagramas, o primeiro ilustra o sistema do banco de dados, o segundo demonstra a usabilidade do mesmo.

5.2.1 Modelagem Back-end

Dentro do contexto do *Back-end*, existe o seguinte ator:

- **Professor:** É o ator que representa um professor no sistema, ele possui a capacidade de criar, ler, atualizar ou deletar (CRUD) as informações do App.

Figura 2 – Modelagem do Back-end



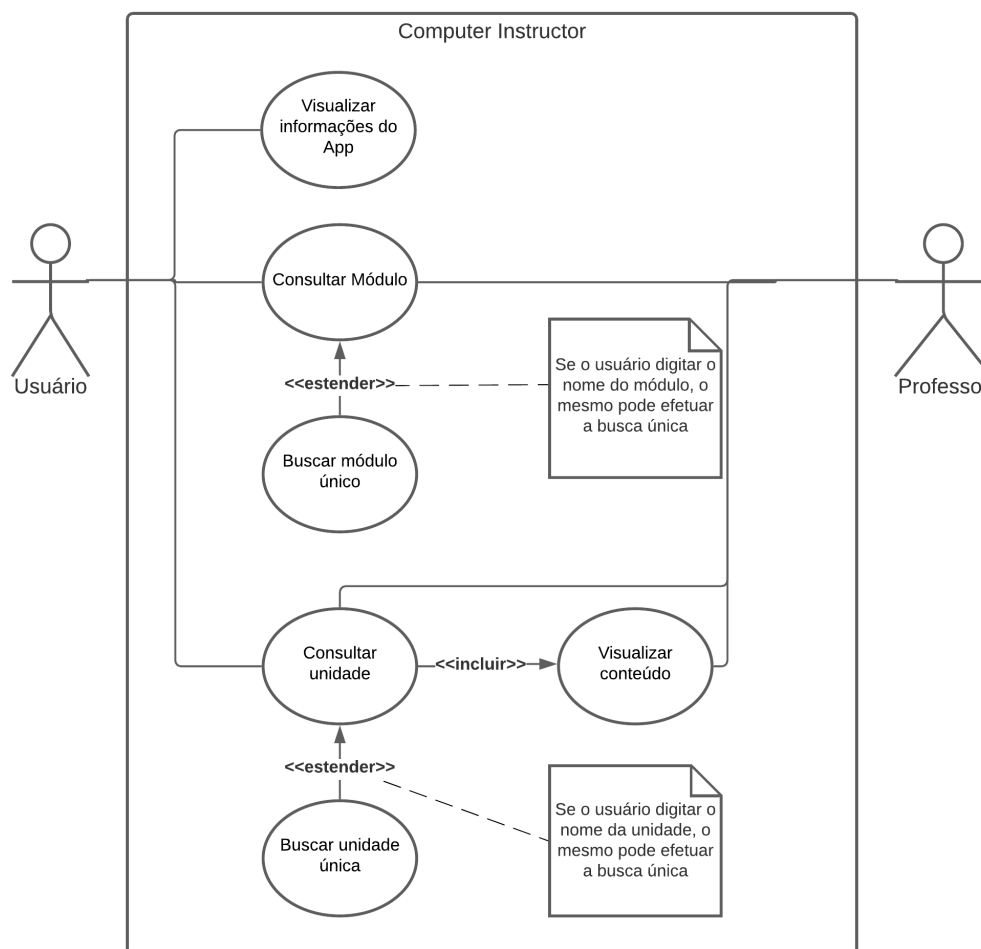
Fonte: O Autor (2021).

5.2.2 Modelagem Computer Instructor

Dentro do contexto do *Computer Instructor*, existem os seguintes atores:

- **Usuário:** É o ator que representa um usuário do App, ele possui a capacidade de visualizar as informações do App, consultar módulos, as unidades do mesmo, e por fim o conteúdo final contido em cada unidade. Também é possível realizar um filtro através da busca única para módulos e unidades.
- **Professor:** Representa as funções que o sistema disponibiliza para o App, listar módulos e unidades que possuem informações.

Figura 3 – Computer Instructor - Modelagem



Fonte: O Autor (2021).

5.2.3 Casos de Uso Expandido

Os casos de uso também podem ser expandidos textualmente em uma sequência de passos presentes dentro dos mesmos, descrevendo detalhadamente o fluxo do caso de uso. A Tabela 4 descreve detalhadamente os casos de uso do sistema.

Tabela 4 – Caso de Uso Expandido

Caso de uso:	Gerenciar conteúdo
Ator:	Professor
Fluxo principal:	O professor realiza o gerenciamento do conteúdo usando as quatro operações básicas: CRUD (Create, Read, Update, Delete).
Caso de uso:	Visualizar informações do App
Ator:	Usuário
Fluxo principal:	O usuário visualiza as informações do App, um breve resumo do mesmo.
Caso de uso:	Consultar módulo
Ator:	Usuário
Fluxo principal:	O usuário visualiza os módulos do App e acessa seus conteúdos.
Caso de uso:	Buscar módulo
Ator:	Usuário
Fluxo principal:	O usuário realiza a busca do módulo desejado através da barra de pesquisa.
Caso de uso:	Consultar unidade
Ator:	Usuário
Fluxo principal:	O usuário visualiza as unidades do App e acessa seus conteúdos.
Caso de uso:	Buscar unidade
Ator:	Usuário
Fluxo principal:	O usuário realiza a busca da unidade desejada através da barra de pesquisa.
Caso de uso:	Visualizar conteúdo
Ator:	Usuário
Fluxo principal:	O usuário acessa o conteúdo da unidade desejada.

Fonte: O Autor (2021).

5.2.4 Prototipagem

Para auxiliar na síntese das ideias, foi elaborado um esboço das telas através de *Wireframes*, utilizando o *Figma* para auxiliar este processo. Nas Sub-subseções 5.2.4.1 e 5.2.4.2 seguem os resultado das prototipações criadas para direcionar o projeto.

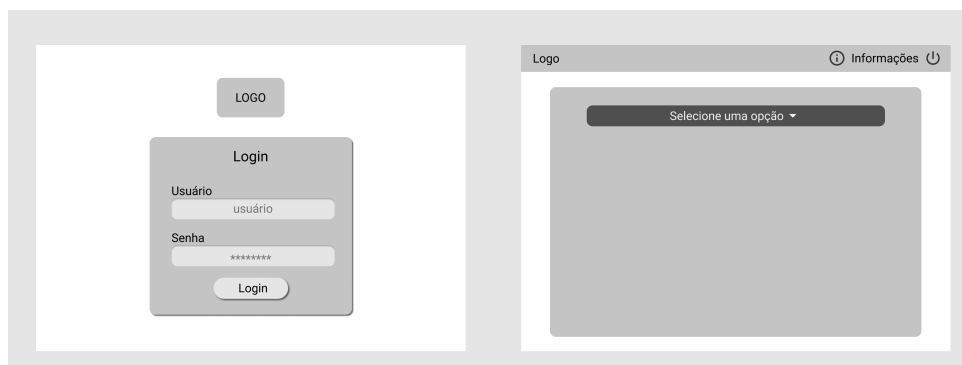
5.2.4.1 Front-end

O esboço do *Front-end* foi criado para facilitar o gerenciamento do conteúdo do *Computer Instructor*, possibilitando criar, ler, atualizar ou deletar as informações. Dessa forma, foram divididos 3 componentes para uma melhor experiência do usuário, visando aplicar conceitos de Interação Humano-Computador (IHC), como facilidade do aprendizado, eficiência, facilidade de memorização e satisfação. Assim sendo, segue abaixo o resultado da prototipagem que foi dividido em três partes, sendo elas as Figuras das telas iniciais, telas de módulos e telas de unidades.

Na Figura 4, está presente o esboço das telas iniciais, onde é possível realizar o

login por parte do usuário brevemente cadastrado, logo após o mesmo é redirecionado para tela *Home*, que concede o gerenciamento da aplicação através das telas de módulos e unidades.

Figura 4 – Wireframe - telas iniciais



Fonte: O Autor (2021).

A Figura 4, exibe o resultado do protótipo das telas de gerenciamento de módulos, onde é possível cadastrar, editar e apagar os mesmos.

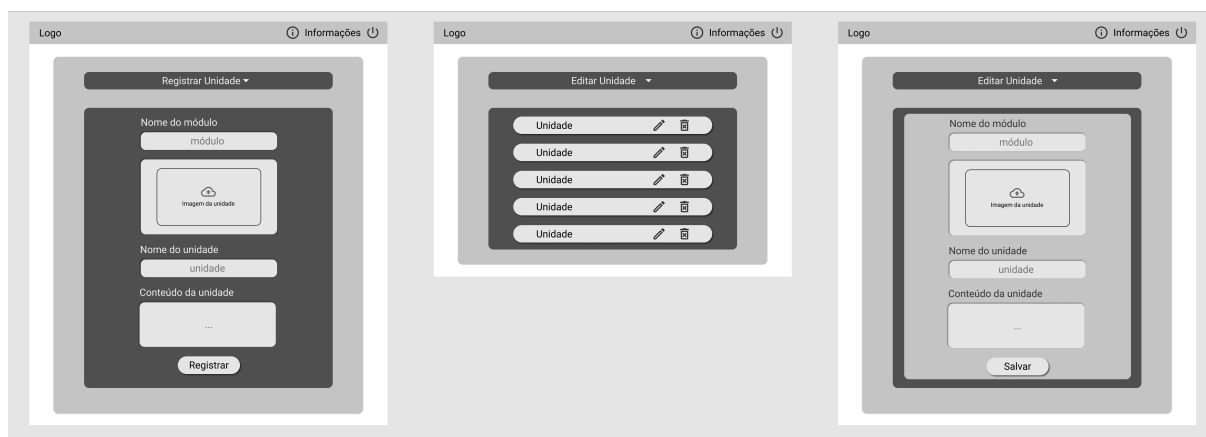
Figura 5 – Wireframe - telas de módulos



Fonte: O Autor (2021).

A Figura 6, apresenta o resultado da prototipagem das telas de gerenciamento de unidades, onde é possível cadastrar, editar e apagar os mesmos, sendo essa a parte mais importante, onde está presente o conteúdo final.

Figura 6 – Wireframe - telas de unidades



Fonte: O Autor (2021).

5.2.4.2 Mobile

O esboço do App foi criado visando a simplicidade, implementando os requisitos para proporcionar agilidade e facilidade no aprendizado do usuário. Nesse sentido, foi desenvolvido a prototipagem do *Computer Instructor* em forma de *wireframes*. Desse modo, a Figura 7 mostra o resultado da prototipagem, onde é possível ver todas as funcionalidades da aplicação, além do fluxo de funcionamento, que possibilita ver informações do mesmo e acessar os módulos cadastrados que possuem unidades com seus respectivos conteúdos.

Figura 7 – Wireframes - Mobile



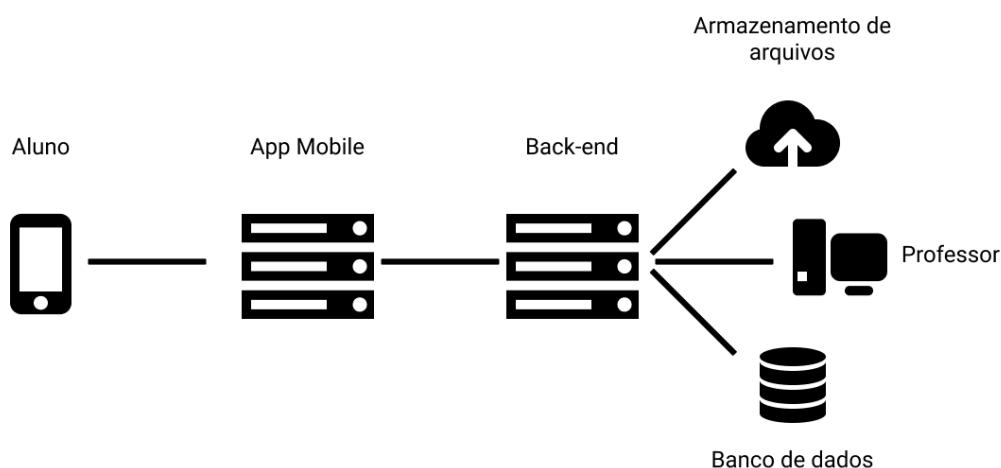
Fonte: O Autor (2021).

5.3 Arquitetura do Computer Instructor

A Figura 8 mostra a arquitetura que foi utilizada na criação do aplicativo *Computer Instructor*. O Banco de Dados é responsável por guardar as informações do sistema, como

os módulos, suas unidades e seus conteúdos, já o Armazenamento de Arquivos tem o intuito de armazenar as mídias cadastradas no App, sendo estes vídeos ou imagens, e todo o controle do conteúdo é feito pelo professor diretamente no *Back-end*.

Figura 8 – Back-end Computer Instructor



Fonte: O Autor (2021).

O *Back-end* vem da ideia do que tem por trás de uma aplicação (SOUTO, 2019). Dessa forma, funciona como uma API (*Application Programming Interface*), um conjunto de funções estabelecidas por um software, que funcionam como uma interface intermediária, para a utilização de funcionalidades deste software por aplicações externas (IRIAS, 2019).

O *Mobile* por sua vez, se trata de um *software* desenvolvido para ser instalado em *smartphones* e *iPads*. Como pode ser visto na figura anterior o App não possui interação com o Banco de Dados ou o Armazenamento de Arquivos, pois todo gerenciamento é feito pelo professor, adicionando, alterando ou removendo os conteúdos através da API, o que garante que as informações gerenciadas passem pelo *Back-end* que é onde estão as regras de negócio, isso garante que o sistema possua integridade nos dados armazenados.

Os Alunos acessam o sistema através do App *mobile*, utilizando a interface provida. Dessa forma, não há interação direta dos alunos com o *Back-end*, Banco de Dados ou Armazenamento de Arquivos.

A estratégia de divisão das responsabilidades do sistema entre o *Back-end* e o *Front-end* é utilizada para garantir uma melhor manutenibilidade, separando a parte lógica, onde ocorre o processamento dos dados, da interface gráfica do sistema, a parte usual,

onde é possível interagir com uma aplicação. Além disso, também foi utilizado a estratégia da utilização de um sistema de banco de dados separado do armazenamento de arquivos com o intuito de conseguir um melhor desempenho do banco de dados da aplicação, por não precisar manipular arquivos extensos.

5.4 Implementação do Computer Instructor

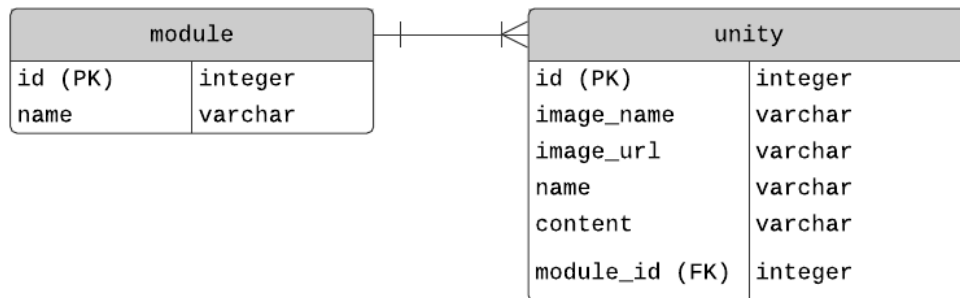
A implementação da aplicação foi feita utilizando a linguagem de programação *Javascript* tanto no *Back-end* quanto no App *mobile*, que por sua vez foi desenvolvido utilizando como base o *framework React Native*. O desenvolvimento da interface foi feito idealizando uma solução que fosse funcional, amigável e minimalista, seguindo o conceito *Affordance*, que segundo Máquina (2014), é o potencial de um objeto de ser usado como foi projetado para ser usado. Nas Subseções 5.4.1 e 5.4.2 serão descritos as criações do *Back-end* e do aplicativo *mobile*, respectivamente.

5.4.1 Back-end

O *Back-end* da aplicação foi desenvolvido utilizando a linguagem de programação *Javascript*, utilizando o modelo *REST (REpresentational State Transfer)*, que representa nada mais que uma “nova” possibilidade para a criação de *web services*, utilizando a semântica dos métodos HTTP (*GET, POST, PUT e DELETE*), na leveza dos pacotes de dados transmitidos na rede e na simplicidade, o que torna desnecessária a criação de camadas intermediárias (DEV MEDIA, 2018). Dessa forma o *Back-end* da aplicação é responsável por dispor as funções através das rotas presentes nele, desempenhando o papel de ser a parte lógica do sistema, operando a base de dados e com o armazenamento de arquivos.

A Figura 9 mostra como as entidades se relacionam entre si dentro do sistema, através de um diagrama entidade-relacionamento (ER). Na imagem é possível perceber que o banco de dados é constituído de duas entidades, uma chamada *module*, que consiste em representar os módulos que possuem unidades, outra entidade, denominada *unity*, cuja mesma engloba o conteúdo final do App.

Figura 9 – Diagrama Entidade-relacionamento

Diagrama Computer Instructor

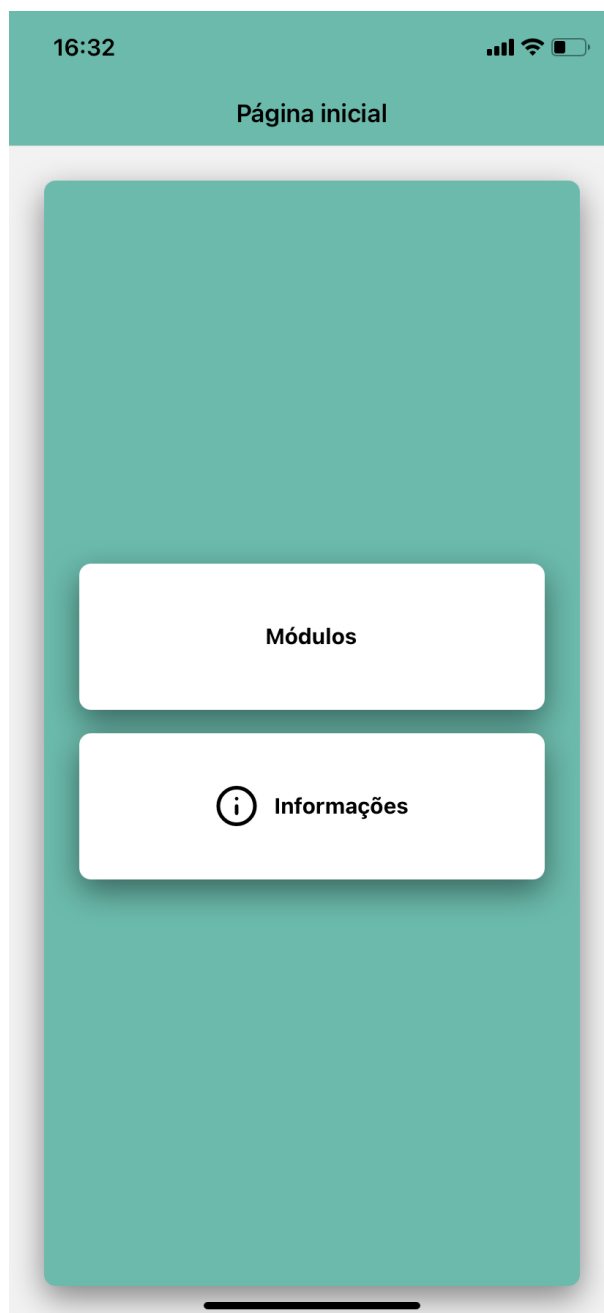
Fonte: O Autor (2021).

5.4.2 Mobile

O aplicativo *mobile* foi desenvolvido utilizando a linguagem de programação *Javascript*, utilizando o *framework React Native*, com o intuito de escrever um único código para funcionar tanto no *Android* quanto no *iOS* de forma nativa, criando assim um App multiplataforma.

A Figura 10 mostra a tela inicial da aplicação, onde de maneira objetiva, há a opção de ver o conteúdo, ou informações sobre a criação do mesmo.

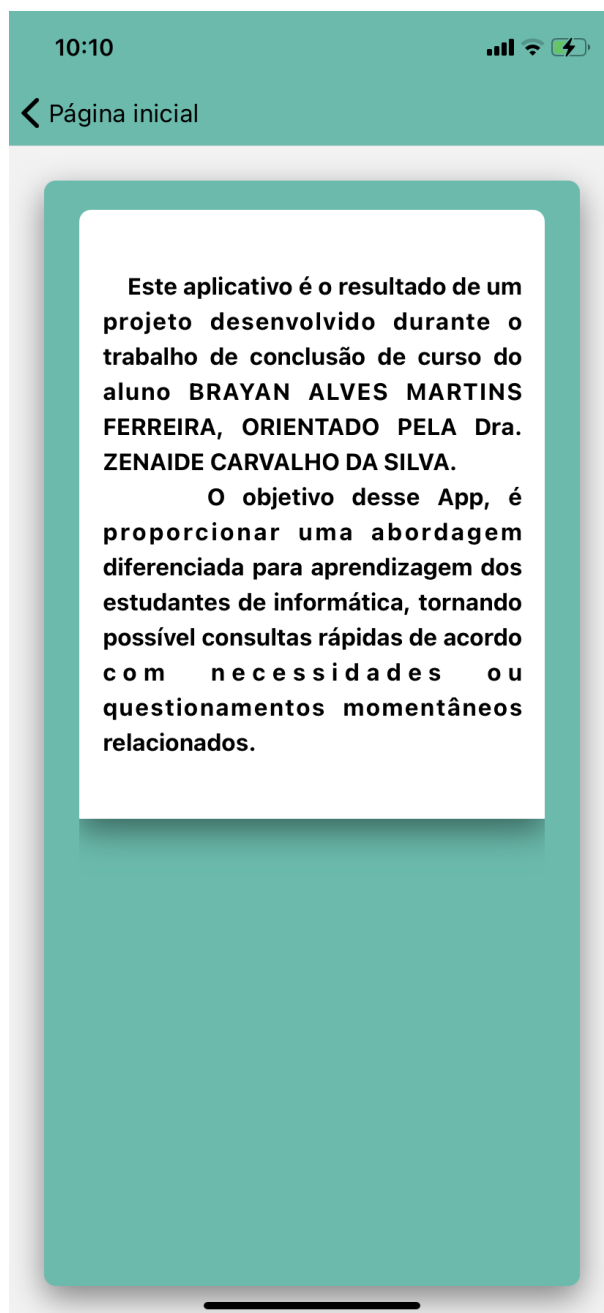
Figura 10 – Mobile - Página inicial



Fonte: O Autor (2021).

Na Figura 11 pode-se ver a tela de informações, onde contém de forma sucinta o propósito e o objetivo do aplicativo.

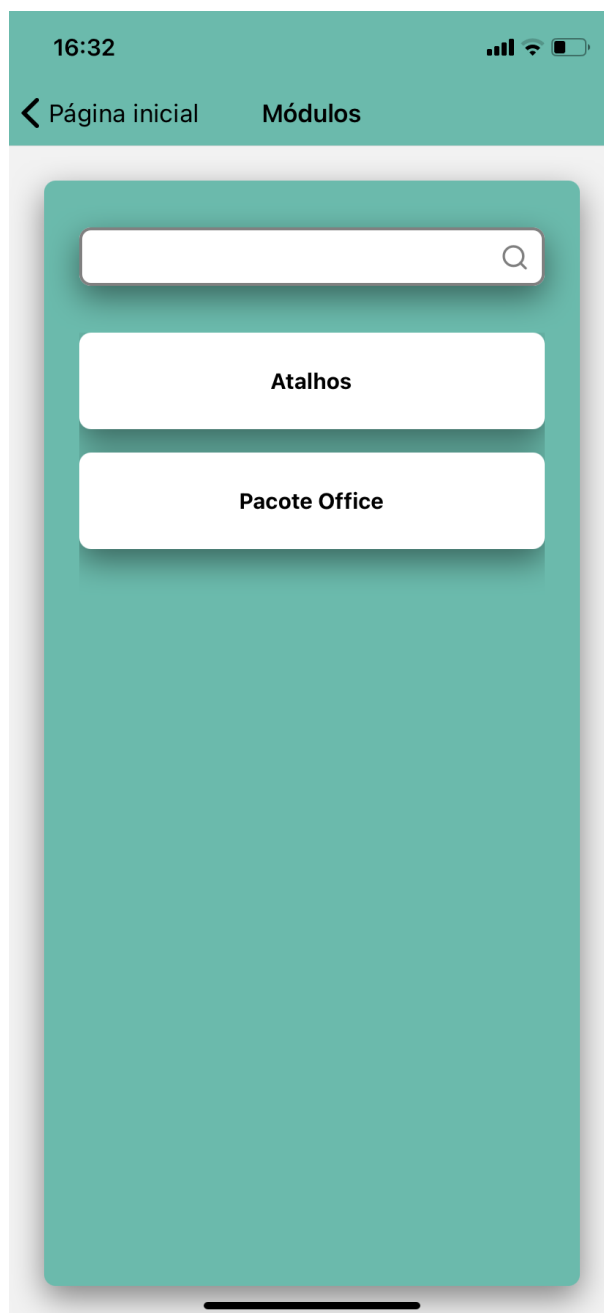
Figura 11 – Mobile - Página de informações



Fonte: O Autor (2021).

Na Figura 12 pode-se ver a tela dos módulos que dividem os conteúdos do aplicativo, também é possível efetuar um filtro para encontrar de maneira mais ágil o módulo desejado. Cada módulo pode conter várias unidades, proporcionando uma diversidade de conteúdo.

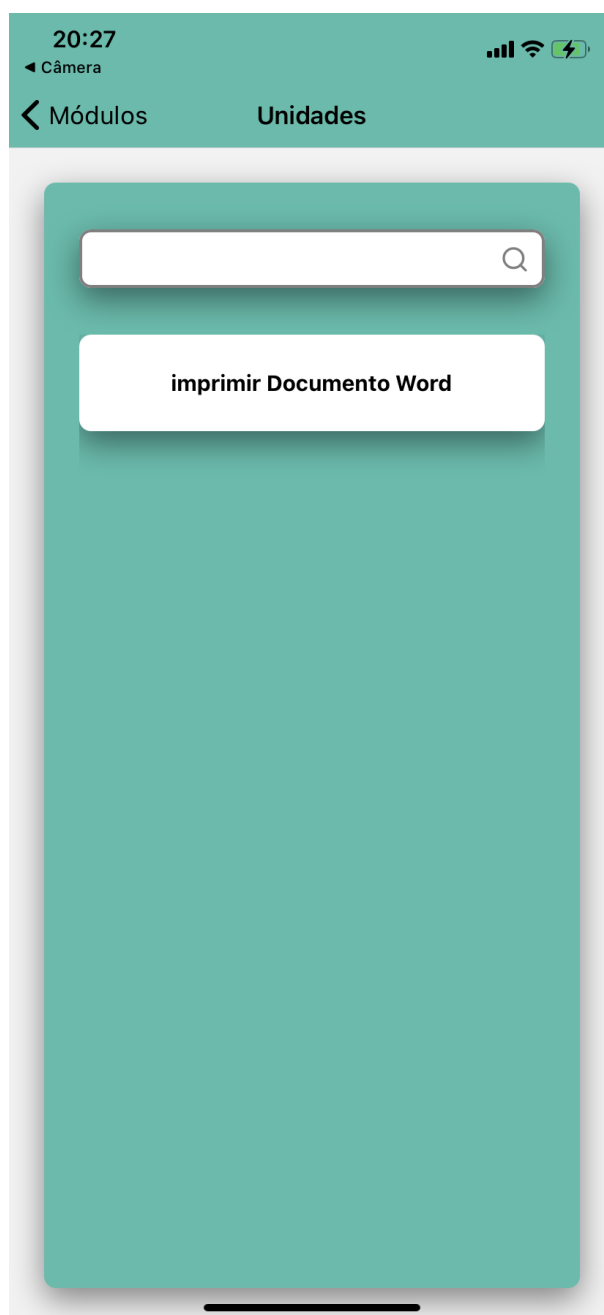
Figura 12 – Mobile - Página de módulos



Fonte: O Autor (2021).

As Figuras 13 e 14 mostram as telas relacionadas às unidades de cada módulo, ilustrando 2 exemplos de diferentes unidades contidas no mesmo, onde cada um possui conteúdo, também é possível efetuar um filtro para encontrar de maneira mais ágil a unidade desejada.

Figura 13 – Mobile - Página de unidades



Fonte: O Autor (2021).

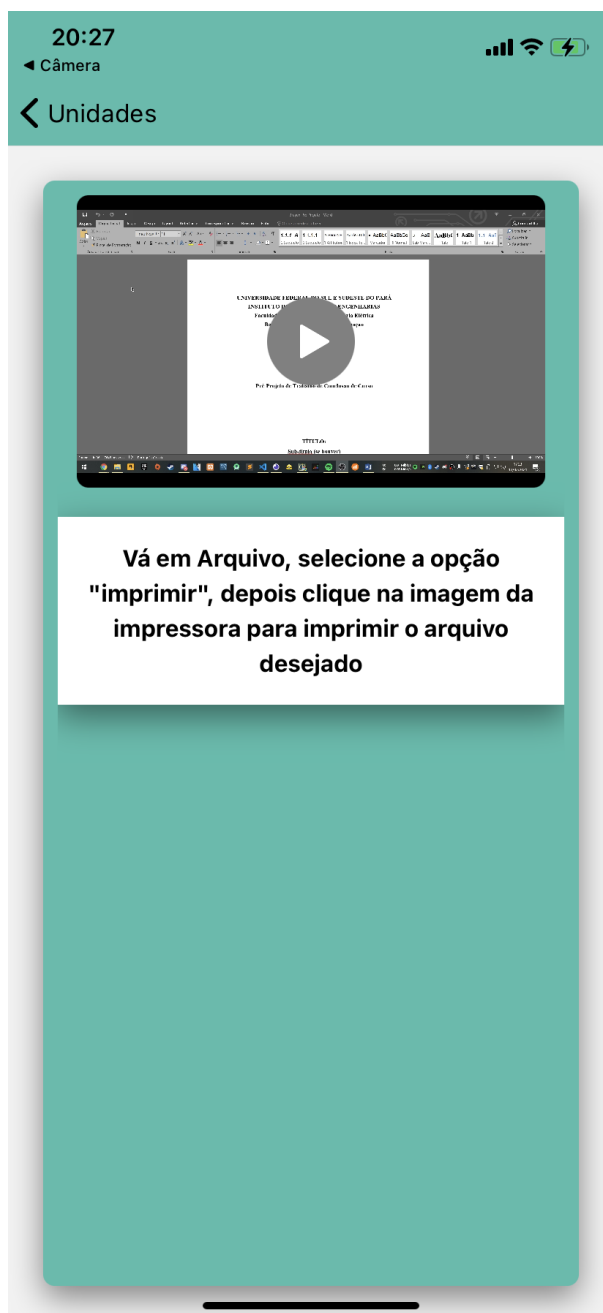
Figura 14 – Mobile - Página de unidades 2



Fonte: O Autor (2021).

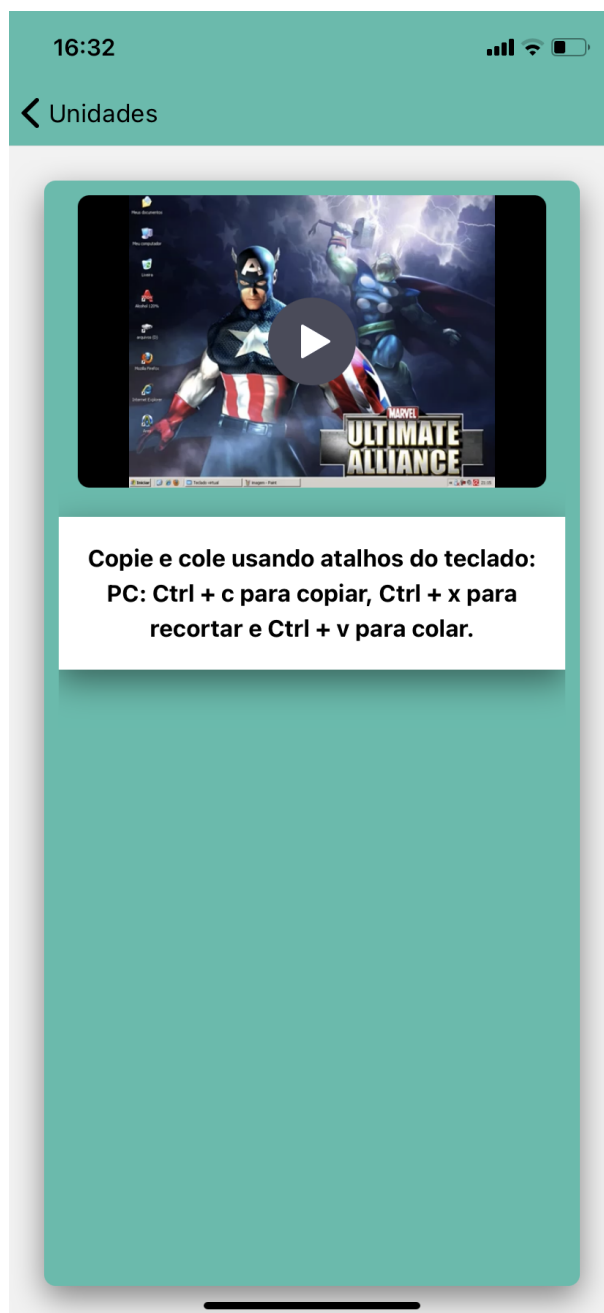
Nas Figuras 15, 16 e 17 é ilustrado a tela final, onde contém o conteúdo final de cada unidade, na mesma é possível ver além de uma explicação textual, uma ilustração em mídia, seja em imagem, *gif*, ou até mesmo vídeo, com o intuito de ser bem simples e objetivo, tornando-o prático.

Figura 15 – Mobile - Página de conteúdo



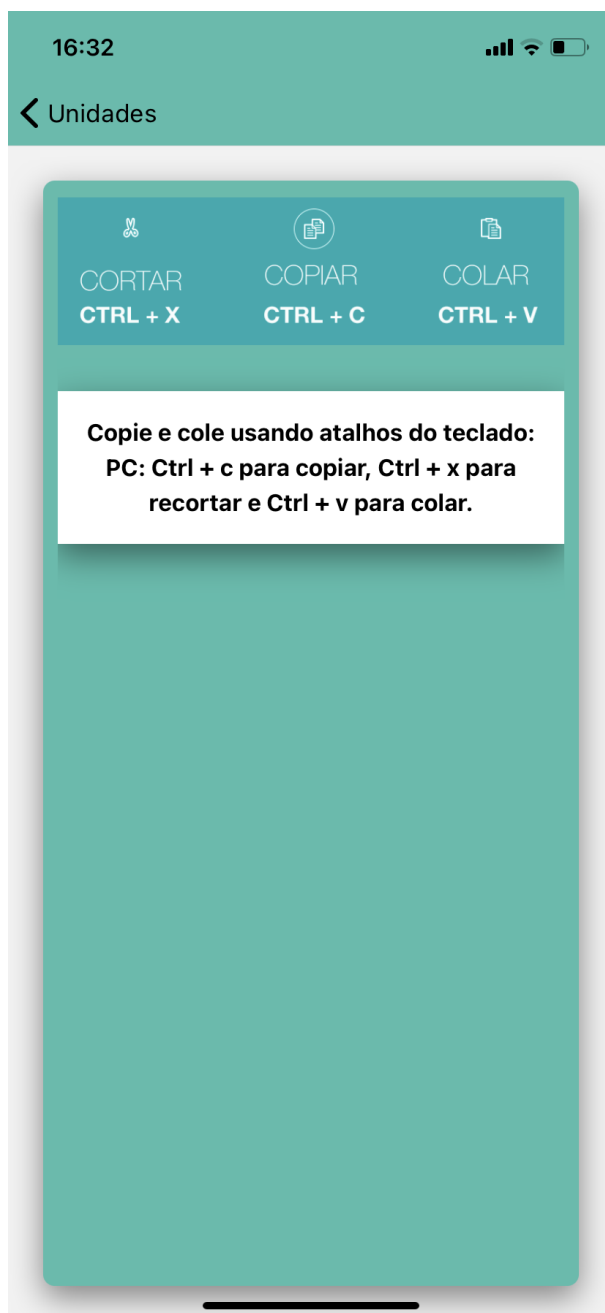
Fonte: O Autor (2021).

Figura 16 – Mobile - Página de conteúdo 2



Fonte: O Autor (2021).

Figura 17 – Mobile - Página de conteúdo 3



Fonte: O Autor (2021).

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido o sistema *Computer Instructor*, criado com o intuito de auxiliar o processo de aprendizagem de estudantes de informática. Desse modo, seu desenvolvimento teve como objetivo a simplicidade, agilidade e funcionalidade, com qualidade, utilizando os fundamentos da engenharia de software.

O desenvolvimento inicial deste trabalho teve origem a partir da análise de pontos de carências no aprendizado de estudantes de informática, baseado na experiência como instrutor, desse modo, foi realizado o levantamento de requisitos para o sistema. Logo após este processo, foi elaborada a modelagem destes requisitos, com o intuito de criar um fluxo de execução do sistema, delimitando as funcionalidades. Em seguida, foi produzido os protótipos de interface do sistema, com a finalidade de esboçar graficamente o funcionamento do sistema, gerando uma base para o desenvolvimento do mesmo. Logo após, deu-se início ao processo de desenvolvimento do sistema, seguindo as modelagens e prototipagens elaboradas. Vale ressaltar que a principal dificuldade encontrada no desenvolvimento do presente trabalho foi organizar as ideias, baseadas nas experiências adquiridas como instrutor, visando um resultado que agregasse valor aos usuários.

Por meio do uso do sistema *Computer Instructor*, o professor poderá ser responsável pelo gerenciamento do conteúdo do mesmo, sendo capaz de cadastrar diversos módulos, com diversas unidades abordando diversos conteúdos. Já por parte dos alunos, através do aplicativo, será possível acessar todo o conteúdo disponibilizado pelo professor, tendo a possibilidade de fazer rápidas consultas aos diversos assuntos.

6.1 Trabalhos Futuros

Conforme abordado ao longo do trabalho, o desenvolvimento do sistema em questão engloba um sistema onde é possível fazer o gerenciamento e apresentação de conteúdos voltados aos alunos de informática, assim, como trabalhos futuros pretende-se realizar:

- Realizar o desenvolvimento do sistema web anteriormente prototipado.
- Implementar modo *off-line*
- Aplicar e avaliar a usabilidade do aplicativo com usuários.

REFERÊNCIAS

- ABREU, J.; SOUSA, J. E. de; LACERDA, M. Um aplicativo móvel para educação ambiental. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2017. v. 28, n. 1, p. 1736. Citado 2 vezes nas páginas 13, 23.
- AELA.IO, E. Wireframe: O que é e como criar um para seus projetos de ux design? **Aela.io**, oct 2019. Citado 2 vezes nas páginas 16, 17.
- ALCANTARA, C. A. A.; VIEIRA, A. L. N. Tecnologia móvel: Uma tendência, uma realidade. **arXiv e-prints**, p. arXiv-1105, 2011. Citado na página 21.
- ALVES, M. R. F. et al. Análise do projeto de extensão de inclusão digital e informática educativa no ensino fundamental da rede pública. In: **Anais do Workshop de Informática na Escola**. [S.l.: s.n.], 2019. v. 25, n. 1, p. 1214-1218. Citado na página 14.
- ANDRADE, G. de O.; OLIVEIRA, V. C. de. Informática na educação.: Um olhar sobre a utilização das novas tecnologias no processo de ensino-aprendizagem. **Educação, Cultura e Comunicação**, v. 8, n. 15, 2017. Citado na página 13.
- BARBOSA, S. D. J.; SILVA, B. S. da. **INTERAÇÃO HUMANO-COMPUTADOR**. [S.l.]: Elsevier Editora Ltda., 2010. Citado 2 vezes nas páginas 20, 21.
- BRAGA, M. da M. **Design de software educacional baseado na teoria dos campos conceituais**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2006. Citado na página 13.
- CODE, V. S. **Set up Visual Studio Code**. [S.l.], 2021. Disponível em: <<https://code.visualstudio.com/learn>>. Acesso em: 16 de Mar. de 2021. Citado na página 17.
- DEVMEDIA. Rest tutorial. **DevMedia**, 2018. Acesso em: 07 de Jul. de 2021. Citado na página 33.
- DEVMEDIA. **Guia de Node.js**. [S.l.], 2021. Disponível em: <<https://www.devmedia.com.br/guia/node-js/40312#introducao-ao-node-js>>. Acesso em: 16 de Mar. de 2021. Citado na página 17.
- DEVMEDIA. **Guia de React Native**. [S.l.], 2021. Disponível em: <<https://www.devmedia.com.br/react-native/>>. Acesso em: 16 de Mar. de 2021. Citado na página 18.
- EXPO. **Introduction to Expo**. [S.l.], 2021. Disponível em: <<https://docs.expo.io/>>. Acesso em: 16 de Mar. de 2021. Citado na página 18.
- GROFFE, R. J. Modelagem de sistemas através de uml: uma visão geral. **DevMedia**, 2013. Citado na página 20.
- HEWETT, T. T. et al. **ACM SIGCHI curricula for human-computer interaction**. [S.l.]: ACM, 1992. Citado na página 20.
- INSOMNIA. **Getting Started with Insomnia**. [S.l.], 2021. Disponível em: <<https://support.insomnia.rest/article/157-getting-started-with-insomnia>>. Acesso em: 16 de Mar. de 2021. Citado na página 17.

- INTERATIVA, S. **Figma: uma nova ferramenta para design de interface que está ganhando o mercado | Sirius Interativa**. [S.l.], 2019. Disponível em: <https://medium.com/@Sirius_/figma-uma-nova-ferramenta-para-design-de-interface-que-est%C3%A1-ganhando-o-mercado-sirius-interativa-2e78e0905b44>. Acesso em: 16 de Mar. de 2021. Citado na página 17.
- IRIAS, A. **API HTTP + REST – Conceito e exemplo em Node.js**. [S.l.], 2019. Disponível em: <<https://imasters.com.br/back-end/api-http-rest-conceito-e-exemplo-em-node-js>>. Acesso em: 06 de Jul. de 2021. Citado na página 32.
- LUCID. **O aplicativo de diagramação inteligente para todas as equipes**. [S.l.], 2021. Disponível em: <<https://lucid.co/pt/produto/lucidchart>>. Acesso em: 31 de Jul. de 2021. Citado na página 18.
- LUCIDCHART. **Diagrama de caso de uso UML: O que é, como fazer e exemplos**. [S.l.], 2021. Citado na página 26.
- LUCIDCHART. **O que é um diagrama UML?** [S.l.], 2021. Acesso em: 18 de Mar. de 2021. Citado na página 20.
- MOZILLA. **JavaScript Tutoriais**. [S.l.], 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 16 de Mar. de 2021. Citado na página 18.
- MÜHLBEIER, A. A. R. K. et al. Mobile hq: o uso de softwares educativos na modalidade m-learning. **Revista de Informática Aplicada**, v. 10, n. 1, 2014. Citado na página 23.
- MÁQUINA, H. **O que é Affordance e como projetar pensando nisso**. [S.l.], 2014. Disponível em: <<https://www.homemmaquina.com.br/o-que-e-affordance/>>. Acesso em: 07 de Jul. de 2021. Citado na página 33.
- NETTO, D. et al. Geography history: Um aplicativo para auxílio ao ensino de geografia. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 1321. Citado na página 23.
- NIELSEN, J. **Usability engineering**. [S.l.]: Morgan Kaufmann, 1994. Citado na página 21.
- PESSOA, R. R. S. **Pesquisa e desenvolvimento de funcionalidade de reconhecimento de imagens para aplicações mobile**. Tese (Doutorado) — IPL Escola Superior de Tecnologia e Gestão, Leiria IP, 2020. Citado na página 21.
- PMI. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PM-BOK)/Project Management Institute**. 6. ed. [S.l.]: Newtown Square, PA: Project Management Institute, 2017. Citado na página 25.
- PRESSMAN, R. S. **Engenharia de Software**. 7. ed. [S.l.]: Amgh Editora, 2009. Citado na página 19.
- RACHID, C. L.; ISHITANI, L. m-tutorial: ferramenta de autoria para desenvolvimento de tutoriais voltados para o m-learning. **Revista Brasileira de Informática na Educação**, v. 20, n. 1, p. 17, 2012. Citado na página 23.
- ROLIM, T. V.; ALENCAR, F. T. F.; COSTA, M. R. G. da. Historyicó-um aplicativo como instrumento de auxílio à educação patrimonial. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2018. v. 7, n. 1, p. 365. Citado na página 23.

SILVA, C. A. d. Crianças e computadores: um estudo exploratório sobre a informática na educação infantil no distrito federal. Universidade de Brasília-unb, 2019. Citado na página 13.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. [S.l.]: Pearson Education do Brasil, 2011. Citado 2 vezes nas páginas 19, 25.

SOUTO, M. **O que é front-end e back-end?** [S.l.], 2019. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>>. Acesso em: 06 de Jul. de 2021. Citado na página 32.

TELEFÔNICA, E. D. **O que é mobile learning? Vá do conceito à prática!** [S.l.], 2021. Disponível em: <https://www.telefonicaeducaciondigital.com/tendencias/-/asset_publisher/G0LheSHQiynX/content/o-que-e-mobile-learning-va-do-conceito-a-pratica->. Acesso em: 02 de Ago. de 2021. Citado na página 21.

TURNER, A. **HOW MANY SMARTPHONES ARE IN THE WORLD?** [S.l.], 2021. Disponível em: <<https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>>. Acesso em: 02 de Ago. de 2021. Citado na página 21.

VENKI. **HEFLO BPM**. [S.l.], 2021. Disponível em: <<https://www.venki.com.br/ferramenta-bpm/heflo/>>. Acesso em: 02 de Ago. de 2021. Citado na página 18.