



CRIANDO
SOLUÇÕES
TECNOLÓGICAS
COM A ENGENHARIA
DE COMPUTAÇÃO

VOLUME 4

EDILSON CARLOS SILVA LIMA
ELDA REGINA DE SENA CARIDADE
JONATHAN ARAUJO QUEIROZ
MARCOS JOSÉ DOS PASSOS SÁ
LEONARDO DE CASTRO MESQUITA
YONARA COSTA MAGALHÃES
(Organizadores)

CRIANDO SOLUÇÕES TECNOLÓGICAS COM A ENGENHARIA DE COMPUTAÇÃO

VOLUME 4

EDITORA PASCAL

2024

2024 - Copyright© da Editora Pascal

Editor Chefe: Prof. Dr. Patrício Moreira de Araújo Filho

Edição e Diagramação: Eduardo Mendonça Pinheiro

Edição de Arte: Marcos Clyver dos Santos Oliveira

Bibliotecária: Rayssa Cristhália Viana da Silva – CRB-13/904

Revisão: Os autores

Conselho Editorial

Dr. Othon Carvalho Bastos Filho

Dr. André Leonardo Demaison Medeiros Maia

Dr. Moisés dos Santos Rocha

Dr^a. Sandra Imaculada Moreira Neto

Dr. Wellinton de Assunção

Dr. Saulo José Figueiredo Mendes

Dr. Raimundo Luna Neres

Dr. Fabio Antonio da Silva Arruda

Dados Internacionais de Catalogação na Publicação (CIP)

L732c

Coletânea Criando Soluções Tecnológicas com a Engenharia de Computação / Edilson Carlos Silva Lima et al. (Orgs). São Luís - Editora Pascal, 2024.

240 f. : il.: (Criando Soluções Tecnológicas com a Engenharia de Computação; v. 4)

Formato: PDF

Modo de acesso: World Wide Web

ISBN: 978-65-6068-040-1

D.O.I.: 10.29327/5391075

1. Engenharia de programa de computador. 2. Tecnologias. 3. Ferramentas. 4. Soluções computacionais. I. Lima, Edilson Carlos Silva. II. Caridade, Elda Regina de Sena. III. Queiroz, Jonathan Araújo. IV. Sá, Marcos José dos Passos. V. Mesquita, Leonardo de Castro. VI. Magalhães, Yonara Costa. VII. Título.

CDU: 004.41

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores.

2024

APRESENTAÇÃO

O livro em questão, mostra desenvolvimento de trabalhos do curso de Engenharia de Computação da Universidade CEUMA, apresenta uma diversidade de temas relevantes e inovadores no campo da tecnologia. Os capítulos abordam uma gama de tópicos, começando pelo desenvolvimento de aplicações móveis utilizando Flutter para a parte de front-end. Em seguida, exploram a construção de APIs Rest com Nest JS, destacando a importância dessa tecnologia na integração de sistemas. A obra também se aprofunda em Internet das Coisas (IoT), apresentando conceitos e aplicações práticas.

Um dos pontos altos do livro é a discussão sobre sustentabilidade e inovação, evidenciado pelo protótipo de uma Máquina de Produção de Filamento 3D utilizando reciclagem de garrafa PET. Essa abordagem demonstra o compromisso dos autores com soluções tecnológicas que impactam positivamente o meio ambiente. Além disso, o livro traz uma perspectiva inclusiva ao discutir formas de noticiar para deficientes auditivos e/ou visuais, promovendo a acessibilidade e a igualdade na disseminação de informações.

A obra também se destaca ao explorar o uso da gamificação e o desenvolvimento em Flutter para o ensino de História e Geografia do Estado do Maranhão, demonstrando como a tecnologia pode ser aplicada de maneira inovadora na educação. Por fim, o livro encerra com um olhar para o futuro, abordando a implementação da tecnologia 5G no mercado, especificamente nos provedores de internet locais, evidenciando a relevância desse avanço para a conectividade e a comunicação rápida.

Dessa forma, o livro oferece uma visão abrangente e atualizada das tendências e inovações no campo da Engenharia de Computação, destacando a aplicação prática de conceitos teóricos em soluções concretas e impactantes.

ORGANIZADORES

Edilson Carlos Silva Lima



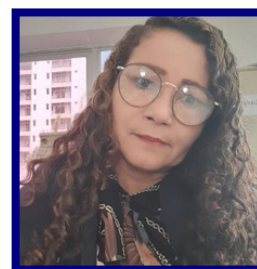
Mestrando em Engenharia da Informática com ênfase na área de Sistemas e Tecnologias de Informação na Universidade Fernando Pessoa na Cidade do Porto em Portugal (com previsão de término em 2023), pós-graduado em Análise e Projeto de Sistemas (UFMA, 2009.1), graduado em Sistema de Informação pela Universidade CEUMA (2008.2) e Tecnólogo em Tecnologia de Informática pela Universidade CEUMA (2003.1), atuou como Programador, Analista de Sistema, Gerente de TI, é professor desde 2012.

LATTES: <https://lattes.cnpq.br/3633743402684029>

ORCID: <https://orcid.org/0000-0002-2301-8006>

Elda Regina de Sena Caridade

Bacharel em Ciência da Computação pela Universidade Federal do Maranhão (1998) e Mestra em Engenharia de Computação e Sistemas pela Universidade Estadual do Maranhão-UEMA. Atuando na área de docência do ensino superior desde 2002 na Universidade Ceuma e em outras Instituições de Ensino Superior. Atualmente coordena os cursos de Engenharia de Computação e Sistemas de Informação da Universidade CEUMA. Tem experiência na área de Engenharia de Computação, Análise e Desenvolvimento de Sistemas.



LATTES: <http://lattes.cnpq.br/8833973908569237>

ORCID: <https://orcid.org/0000-0003-2243-0477>

Jonathan Araujo Queiroz



Possui graduação em matemática licenciatura pela Universidade Federal do Maranhão (2012), especialista em Métodos Estatísticos Aplicados pela Universidade Estadual do Maranhão (2016), mestrado em Engenharia de Eletricidade pela Universidade Federal do Maranhão (2016), doutorado em Engenharia Elétrica pela Universidade Federal do Maranhão (2018) e Pós-Doutorado pela Universidade Federal do Maranhão (2020). É colaborador da Sociedade Brasileira de Engenharia Biomédica e revisor de IEEE Access, IEEE Engineering in Medicine and Biology Society, Journal of Cardiac Disorders and Therapy (JCDDT), e Electric Power Components and Systems Journal.

LATTES: <http://lattes.cnpq.br/7145102625820184>

ORCID: <https://orcid.org/0000-0001-8006-6242>

Marcos José dos Passos Sá

Possui graduação em Tecnólogo em Desenvolvimento de Sistemas pela Universidade Ceuma (2008), Especialização em MBA Governança de TI pela Universidade Ceuma (2013) e Mestrado em Engenharia de Computação e Sistemas pela Universidade Estadual do Maranhão (2020). Atualmente é Professor Mestre da Universidade Estadual do Maranhão e Professor da Universidade Ceuma. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, Cloud Computer e Telecomunicações.



LATTES: <http://lattes.cnpq.br/2111990319894447>

ORGANIZADORES

Leonardo de Castro Mesquita



Possui graduação em Gestão da Tecnologia da Informação pela Universidade Paulista (2014), Especialização em MBA Profissional em Redes de Computadores pela Escola Superior Aberta do Brasil – ESAB (2015), Graduação em licenciatura em Geografia pela Universidade Cruzeiro do Sul – UNICSUL (2022), Especialização em Geografia Regional Brasileira pelo Instituto Facuminas EAD LTDA – FACUMINAS (2022). Analista de Sistemas na Faculdade de Medicina - UFMA, trabalhando com projetos de realidade aumentada, realidade virtual, objetos de aprendizagem e simuladores avançado de alta performance que apresenta realismo elevado e múltiplos modelos de fisiologia humana. Foi Coordenador Tecnológico do Núcleo de Educação a Distância - NEAD da Universidade Federal do Maranhão – UFMA. É pesquisador do Laboratório de Inovações Tecnológicas Aplicadas à Saúde – LITAS/UFMA. É professor da Universidade do CEUMA e Coordenador do Núcleo de Tecnologias Educacionais – NUTED/CEUMA. Professor de Informática no Instituto Estadual de Educação, Ciência e Tecnologia do Maranhão – IEMA. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, Cloud Computer Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, Cloud Computer, Desenvolvimento de Software, Gestão de TI e TIC's na educação.

LATTES: <http://lattes.cnpq.br/4641659171222606>

ORCID: <https://orcid.org/0009-0009-1643-1562>

Yonara Costa Magalhães

É bacharel em Ciência da Computação pela Universidade Federal do Maranhão (1998) e mestre em Engenharia de Eletricidade pela Universidade Federal do Maranhão (2002). É professora titular da Universidade do CEUMA (2011) e professora contratada na Universidade Estadual do Maranhão (UEMA/2022). Foi docente no Centro Universitário Euro-Americano (UNIEURO- DF 2006/2011), tendo atuado como: Pesquisadora Institucional, membro do NDE de Sistemas de Informação e de CST em Redes de Computadores, Coordenadora dos Cursos de Sistemas de Informação, CST em Redes de Computadores e CST em Gestão da Tecnologia da Informação e Coordenadora Pedagógica. Foi Coordenadora do Curso de CST em Análise e Desenvolvimento de Sistemas no CEUMA. É pesquisadora do NuSTI/CnPQ (Núcleo de Pesquisa em Sistemas e Tecnologia da Informação) do CEUMA e membro da Sociedade Brasileira de Computação (SBC).



LATTES: <http://lattes.cnpq.br/8188763596503654>

ORCID: <https://orcid.org/0000-0001-5502-9634>

SUMÁRIO

CAPÍTULO 1	10
ESTRATÉGIAS DE OTIMIZAÇÃO DE DESEMPENHO EM APLICAÇÕES WEB COM O GOOGLE LIGHTHOUSE	
<i>Mateus da Rocha Rodrigues</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Jonathan Araújo Queiroz</i>	
D.O.I.: 10.29327/5391075.1-1	
CAPÍTULO 2	27
DESENVOLVIMENTO DE APP PARA GESTÃO MÉDICA EM NEUROLOGIA COM FRONT END EM FLUTTER E API REST COM SPRING BOOT	
<i>Alanna Rocha Monteiro</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Yonara Costa Magalhães</i>	
D.O.I.: 10.29327/5391075.1-2	
CAPÍTULO 3	44
CRIAÇÃO DE UM APLICATIVO PARA CONDOMÍNIOS UTILIZANDO FLUTTER E REST API VOLTADOS PARA A UTILIZAÇÃO DO MORADOR E PORTEIRO	
<i>Pedro Henrique Silva de Sousa Carneiro</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Leonardo de Castro Mesquita</i>	
D.O.I.: 10.29327/5391075.1-3	
CAPÍTULO 4	60
PLATAFORMA WEB DESENVOLVIDA EM Node.JS E REACT PARA AUXÍLIO NA APRENDIZAGEM DE CIÊNCIAS EXATAS	
<i>Nikaelson Passos Pereira</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Yonara Costa Magalhães</i>	
D.O.I.: 10.29327/5391075.1-4	
CAPÍTULO 5	77
UTILIZAÇÃO DA GAMIFICAÇÃO PARA O ENSINO DA LINGUAGEM NORDESTINA	
<i>André Kleysson Gois de Carvalho</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Leonardo de Castro Mesquita</i>	
D.O.I.: 10.29327/5391075.1-5	

CAPÍTULO 6.....	93
AMPLIAÇÃO DE APLICATIVO MOBILE COM FLUTTER: FORNECENDO APOIO PEDAGÓGICO E PSICOSSOCIAL AO CORPO DISCENTE E DOCENTE	
<i>Ada Kezia Freitas Ribeiro Rodrigues</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Yonara Costa Magalhães</i>	
D.O.I.: 10.29327/5391075.1-6	
CAPÍTULO 7.....	104
DESENVOLVIMENTO INTEGRADO DE FRONT-END E BACK-END PARA APRIMORAR A EXPERIÊNCIA DO USUÁRIO	
<i>Kaio Eduardo Pereira Cruz</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Elda Regina de Sena Caridade</i>	
<i>Jonathan Araújo Queiroz</i>	
D.O.I.: 10.29327/5391075.1-7	
CAPÍTULO 8	120
PLATAFORMA WEB DE INSERÇÃO DE QUESTÕES E GRUPOS NO LUDO EDUCACIONAL	
<i>Kelve Antonio Oliveira Lima</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Yonara Costa Magalhães</i>	
D.O.I.: 10.29327/5391075.1-8	
CAPÍTULO 9.....	135
OcupApp: CONTROLE INTELIGENTE DE ESPAÇOS	
<i>Marcos Levy de Freitas Coelho</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Jonathan Araújo Queiroz</i>	
D.O.I.: 10.29327/5391075.1-9	
CAPÍTULO 10	151
O PAPEL DO SPRING FRAMEWORK NO DESENVOLVIMENTO DO APLICATIVO CITY CONNECT	
<i>Mayron Sergio Santos</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Elda Regina de Sena Caridade</i>	
D.O.I.: 10.29327/5391075.1-10	

CAPÍTULO 11	168
DESENVOLVIMENTO DA PLATAFORMA GUARANI USANDO VUE.JS E SPRING BOOT PARA DIVULGAÇÃO DE PESQUISAS CIENTÍFICAS NA WEB	
<i>Bruno Emanuel Santos Lima</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Yonara Costa Magalhães</i>	
D.O.I.: 10.29327/5391075.1-11	
CAPÍTULO 12	182
DESENVOLVIMENTO DE APLICATIVO PARA GERENCIAMENTO DE FINANÇAS PESSOAIS	
<i>Thiago Vinicius Araujo de Souza</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Jonathan Araújo Queiroz</i>	
D.O.I.: 10.29327/5391075.1-12	
CAPÍTULO 13	202
CONECTANDO ATLÉTICAS ACADÊMICAS COM FLUTTER	
<i>Carlos Magno Souza dos Santos</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Leonardo de Castro Mesquita</i>	
D.O.I.: 10.29327/5391075.1-13	
CAPÍTULO 14	219
CRIAÇÃO DE UM APLICATIVO PARA CONDOMÍNIOS UTILIZANDO FLUTTER E REST API VOLTADOS PARA A UTILIZAÇÃO DO MORADOR E PORTEIRO	
<i>Pedro Henrique Silva de Sousa Carneiro</i>	
<i>Edilson Carlos Silva Lima</i>	
<i>Leonardo de Castro Mesquita</i>	
D.O.I.: 10.29327/5391075.1-14	
AUTORES	235



ESTRATÉGIAS DE OTIMIZAÇÃO DE DESEMPENHO EM APLICAÇÕES WEB COM O GOOGLE *LIGHTHOUSE*

1

*Performance Optimization Strategies in Web Applications with
Google Lighthouse*

Mateus da Rocha Rodrigues
Edilson Carlos Silva Lima
Jonathan Araújo Queiroz

D.O.I.: 10.29327/5391075.1-1



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Em um mundo digital onde a rapidez e a interatividade são fundamentais, a otimização do desempenho em aplicações *web* tornou-se crucial para proporcionar boas experiências aos usuários. Este artigo discute as estratégias que impulsionam a eficiência das aplicações, destacando a importância das métricas de desempenho do *Core Web Vitals* presentes na ferramenta de análise *Google Lighthouse*. Essas métricas oferecem indicadores essenciais para avaliar e aprimorar a qualidade da experiência na *web*. Para embasar este artigo, serão empregadas as metodologias quantitativas e estudos de casos múltiplos em três aplicações para explorar como essas métricas se relacionam, evidenciando a importância da otimização do desempenho em um ambiente digital em constante evolução.

Palavras-chave: Otimização de desempenho, *Core Web Vitals*, *Google Lighthouse*.

Abstract

In a digital world where speed and interactivity are fundamental, optimizing performance in web solutions has become crucial to providing better experiences for users. This article discusses strategies that boost application efficiency, highlighting the importance of the *Core Web Vitals* performance metrics present in the *Google Lighthouse* analysis tool. These metrics offer essential indicators for evaluating and improving the quality of the web experience. To support this article, experimental methodologies and unique case studies will be used to explore how these metrics relate, highlighting the importance of optimizing performance in a constantly evolving digital environment.

Keywords: Optimizing performance, *Core Web Vitals*, *Google Lighthouse*.

1. INTRODUÇÃO

No contexto dinâmico e exigente das aplicações web, a eficiência do desempenho emergiu como um fator crítico para o sucesso digital. À medida que usuários se tornam cada vez mais exigentes, a experiência proporcionada por uma aplicação é intrinsecamente vinculada à sua performance. Nesse cenário, o desafio para desenvolvedores e gestores reside não apenas na entrega de funcionalidades, mas na garantia de uma resposta ágil e eficiente, capaz de cativar e reter os usuários. Surge, portanto, a necessidade de uma abordagem estratégica e técnica para otimizar o desempenho, superando obstáculos que vão desde a complexidade do desenvolvimento até as expectativas crescentes dos usuários.

Este artigo tem como objetivo abordar a importância do bom desempenho no contexto das aplicações web. Além disso, o estudo busca justificar a necessidade de implementar estratégias de otimização de desempenho, não apenas como um requisito técnico, mas como um diferencial no mundo digital altamente competitivo de hoje. Compreender como o desempenho afeta diretamente a retenção de usuários é fundamental para quem deseja oferecer experiências de alta qualidade na web. Nesse artigo será utilizado as metodologias quantitativas e estudos de casos múltiplos.

Na seção 2, Trabalhos Relacionados, é apresentado três artigos que oferecem referências valiosas para enriquecer o conteúdo deste artigo. Em seguida, na seção 3, Fundamentação Teórica, exploramos aspectos essenciais. Inicialmente, é destacado a importância do desempenho (3.1). Em seguida, é abordado algumas métricas do *Core Web Vitals* que serão utilizadas para a auditoria das aplicações (3.2) e a ferramenta Google *Lighthouse* (3.3), uma poderosa aliada na auditoria para análise de desempenho de aplicações web. Avançando para a seção 4, Resultado e discussão, começamos com os Estudos de casos com análise de cada aplicação (4.1, 4.2, 4.3), na seção 5, Análise de desempenho e na seção 6, Conclusão.

2. TRABALHOS RELACIONADOS

Neste capítulo, é feito uma análise dos trabalhos relacionados que compartilham semelhanças com esse artigo. A seleção desses artigos enriquece a compreensão dos assuntos abordados, proporcionando a capacidade de contextualizar no cenário mais amplo da área de estudo. Os estudos anteriores desempenham um papel fundamental na identificação das contribuições inovadoras e dos avanços para a otimização de aplicações web.

2.1 Otimização de Desempenho Orientado ao Tempo de Carregamento em Aplicações Web Front-end

Este trabalho oferece um guia abrangente para desenvolvedores que desejam otimizar o desempenho de aplicações web, baseado em revisões da literatura especializada e destacando ferramentas e técnicas comprovadas. O trabalho é uma referência fundamental para aprimorar o tempo de carregamento e eficiência das aplicações, incluindo uma aplicação prática de código aberto.

Para futuros trabalhos e contribuições da comunidade, o Guia tem o potencial de evoluir de várias maneiras. Isso inclui a coleta e documentação de casos de sucesso relacionados à refatoração de grandes aplicações com foco na redução do tempo de carregamento,

a expansão do número de técnicas abordadas no Guia, como CSS e Fontes, e a introdução de mais ferramentas para oferecer diagnósticos mais completos e precisos aos desenvolvedores.

2.2 A new approach in quantifying user experience in web-oriented applications

Este artigo propõe uma abordagem para quantificar a experiência do usuário em aplicações *web*. Ele se concentra nos fatores que compõem os *Core Web Vitals*, destacando sua importância como indicadores da qualidade da experiência do usuário em sites. Embora a *web* seja influenciada por diversos fatores que podem afetar essa experiência, o estudo se concentra em uma análise aplicada em um exemplo de site de teste. Esses indicadores refletem principalmente a qualidade do trabalho dos desenvolvedores *web* envolvidos na criação dos *sites*, ressaltando a relevância de direcionar a atenção à experiência do usuário durante o desenvolvimento.

Os esforços futuros continuarão a se concentrar na implementação de fatores adicionais e no aprimoramento da experiência do usuário em aplicações *web*. Esta pesquisa representa um passo importante na compreensão e melhoria da experiência do usuário na *web*, com potencial para influenciar positivamente o trabalho dos desenvolvedores e a qualidade geral dos aplicativos voltados para a *web*.

2.3 Towards representative web performance measurements with Google Lighthouse

O trabalho em questão é uma revisão da literatura que analisa o uso do *Google Lighthouse* como ferramenta para medir o desempenho de sites na *web*. O objetivo do estudo é entender como a comunidade de pesquisa tem abordado o desafio da variabilidade nas medições de desempenho ao usar essa ferramenta e demonstrar a estratégia de executar várias medições e agregá-las em um resultado único.

Como trabalhos futuros, os autores sugerem a realização de estudos adicionais para mitigar medições de desempenho não representativas. Além disso, eles destacam a importância de pesquisas que investiguem estratégias para lidar com a variabilidade nas medições de desempenho e fornecer recomendações mais robustas para o uso do *Google Lighthouse*.

2.4 Proposta

Um diferencial deste artigo é a abordagem abrangente de aplicações *web*, que inclui uma análise das *Core Web Vitals* que estão incluídas no *Google Lighthouse*, métricas essenciais para avaliar a qualidade da experiência do usuário. O artigo não apenas discute conceitos teóricos, mas também oferece uma visão prática. Essa abordagem proporciona uma compreensão mais sólida de como as métricas de desempenho impactam diretamente a usabilidade e a satisfação do usuário, tornando-o altamente informativo e aplicável.

Além disso, o uso da ferramenta *Google Lighthouse* para conduzir auditorias em análises de medição de desempenho adiciona um nível de rigor técnico ao artigo, além de



abordar sobre a variabilidade que pode ocorrer nas medições, demonstrando um compromisso com a precisão e a confiabilidade dos resultados. A ferramenta é uma referência amplamente reconhecida no campo da otimização *web*, o que aumenta a credibilidade do estudo. Este artigo oferece uma visão detalhada analisando três aplicações usando o *Lighthouse* e proporcionando *insights* sobre como melhorar o desempenho. Esse aspecto estratégico adiciona profundidade ao conteúdo, destacando a relevância do tema em um contexto mais amplo e atual.

3. FUNDAMENTAÇÃO TEÓRICA

A otimização de desempenho em aplicações *web* mudou de uma mera preocupação técnica para se tornar um imperativo estratégico. À medida que a demanda por experiências digitais aumenta, a velocidade, a interatividade e a satisfação do usuário ganham destaque. Nesse contexto, é abordado as estratégias que moldam o panorama da otimização de desempenho. Nas seções subsequentes, serão explorados os diferentes aspectos relacionados à otimização de desempenho em soluções *web*. Na Seção 3.1, é descrito sobre a importância do desempenho *web*. Em seguida, na Seção 3.2, será apresentado uma análise de algumas métricas do *Core Web Vitals* que serão utilizadas na análise das aplicações, destacando a relevância individual de cada uma delas. Por fim, na Seção 3.3, a ferramenta *Lighthouse*, que desempenha um papel essencial na análise do desempenho *web*.

3.1 Importância do Desempenho

O desempenho de um site tem um impacto significativo na sua classificação nos mecanismos de pesquisa, sendo avaliado com base em fatores como a velocidade de carregamento da página, a experiência do usuário e outras métricas de desempenho. Estudos do Google indicam que tempos de carregamento inferiores a 100 milissegundos criam a sensação de resposta instantânea para os visitantes, alinhando-se com o funcionamento do processador visual de memória sensorial no cérebro humano. A cada segundo adicional de carregamento, a capacidade de manter a atenção do usuário diminui, e após 10 segundos, a impaciência e a frustração podem levar os visitantes a abandonar o site, resultando em uma experiência negativa que pode afetar o retorno futuro a sites tão lentos (Kinsta, 2023).

Vasilijević (2020), enfatiza a importância do desempenho do site para fornecer uma experiência de usuário de alta qualidade. A velocidade do site é um dos principais fatores que podem afetar a satisfação do usuário e que o carregamento lento das páginas pode levar à frustração e a uma impressão negativa do site. Ao usar ferramentas como *Lighthouse*, os desenvolvedores podem identificar áreas onde o desempenho do site pode ser melhorado e fazer alterações para garantir que as páginas carreguem de forma rápida e suave. Isso pode ajudar a garantir que os usuários tenham uma experiência positiva ao usar o site e tenham maior probabilidade de retornar no futuro.

O desempenho de um site não apenas influencia a classificação nos mecanismos de pesquisa, mas também impacta diretamente a experiência do usuário. A pesquisa do Google, que relaciona os tempos de carregamento da página ao funcionamento do cérebro humano, revela a conexão entre os aspectos técnicos da *web* e as reações emocionais dos usuários. Isso ressalta a importância de os desenvolvedores e proprietários de sites considerarem esse elo ao priorizar o desempenho, pois isso afeta a satisfação e a retenção dos usuários. Esse insight é valioso para a comunidade de desenvolvimento *web* na busca por experiências online mais eficazes e agradáveis.

3.2 Core Web Vitals

As *Core Web Vitals* são o subconjunto de *Web Vitals* que se aplica a todas as páginas da *web*, que devem ser medidos por todos os proprietários de *sites* e são revisadas por todas as ferramentas do *Google*. Cada uma das *Core Web Vitals* representa uma face-ta distinta da experiência do usuário, é mensurável em campo e reflete a experiência de mundo real de um resultado crítico centrado no usuário, sendo uma iniciativa do *Google* para fornecer orientação unificada em relação a sinais de qualidade essenciais que visam proporcionar uma ótima experiência ao usuário na *web* (Walton, 2020).

De acordo com McQuade (2020), o *Google* também estabeleceu limites recomendados para cada métrica, dividindo-os em categorias, como: “bom”, “precisa melhorar” e “ruim”. O objetivo é que as páginas alcancem a categoria “Bom” em todas as métricas para proporcionar uma experiência de usuário agradável e eficaz. As *Core Web Vitals* fazem parte do esforço do *Google* para melhorar a qualidade da *web*, proporcionando uma maneira padronizada de medir e otimizar a experiência do usuário em termos de desempenho. Elas também podem ter um impacto no *ranking* nos mecanismos de busca, tornando-as ainda mais relevantes para os desenvolvedores e proprietários de *sites*. Segundo (Pícolo, 2021), o objetivo é fornecer aos desenvolvedores uma referência para avaliar e melhorar o desempenho de suas aplicações, garantindo uma experiência de usuário satisfatória.

É interessante como o *Google* estabeleceu limites recomendados e categorias para essas métricas, enfatizando o objetivo de proporcionar uma experiência de usuário agradável. Existem diversas métricas definidas pelo *Google* no *Core Web Vitals*, nesse artigo em questão será abordado as métricas que estão incluídas na ferramenta de análise de desempenho *Google Lighthouse* que será utilizada.

3.2.1 Largest Contentful Pain (LCP)

Largest Contentful Paint (LCP), ou Maior Renderização de Conteúdo, é uma métrica importante e centrada no usuário para medir a velocidade de carregamento percebida porque marca o ponto na linha do tempo de carregamento da página quando o conteúdo principal da página provavelmente já foi carregado (Walton, 2019). Uma LCP rápida ajuda a assegurar ao usuário que a página é útil, (na figura 1 é mostrado o tempo ideal definido pelo *Google*, passando desse tempo requer atenção).



Figura 1. LCP.

Fonte: Walton, 2019.

A métrica LCP informa o tempo de renderização da maior imagem ou bloco de texto visível na janela de visualização *viewport*, em relação a quando a página começou a carregar pela primeira vez (Walton, 2019).

Destaca-se a importância na avaliação da velocidade de carregamento percebida pelos usuários. Ao mencionar que a LCP marca o ponto em que o conteúdo principal da pá-

gina provavelmente já foi carregado, ele ressalta a relevância de garantir uma LCP rápida para proporcionar uma experiência útil ao usuário.

3.2.2 Total Blocking Time (TBT)

O *Total Blocking Time* (TBT), ou tempo total de bloqueio, é uma métrica de laboratório importante para medir a responsividade da carga porque ajuda a quantificar o nível de não-interatividade de uma página antes que ela se torne interativa de forma confiável (Walton, 2019). Um baixo valor de TBT ajuda a garantir que a página seja utilizável, (na figura 2 podemos observar o tempo ideal).



Figura 2. TBT.

Fonte: Crystallize, [s.d.].

Para fornecer uma boa experiência ao usuário, os *sites* devem se esforçar para ter uma *Total Blocking Time* de menos de 200 milissegundos quando testados em um *hardware* móvel típico. (Walton, 2019).

3.2.3 Cumulative Layout Shift (CLS)

Cumulative Layout Shift (CLS), ou Mudança Cumulativa de *Layout*, é uma métrica importante e centrada no usuário para medir a estabilidade visual porque ajuda a quantificar a frequência com que os usuários experimentam mudanças inesperadas de *layout*: uma CLS baixa ajuda a garantir que a página seja agradável (Walton, 2019), (conforme é mostrado na figura 3, sendo o tempo máximo ideal é de 0,1).



Figura 3. CLS.

Fonte: Walton, 2019.

CLS é uma medida da maior explosão de ocorrências de mudança de *layout* para cada mudança de *layout* inesperada que ocorre durante toda a existência de uma página. Uma mudança de *layout*, conhecida como janela de sessão (*session window*), ocorre sempre que um elemento visível muda sua posição de um quadro renderizado para o próximo (Walton, 2019).

No texto é destacado a importância na avaliação da estabilidade visual de uma página *web*. Além de quantificar a frequência das mudanças inesperadas de *layout*, ele introduz o conceito de “janela de sessão”, que é fundamental para entender como o CLS é calculado ao longo da existência da página.

3.2.4 First Contentful Paint (FCP)

A *First Contentful Paint* (FCP), ou primeira renderização de conteúdo, é uma métrica importante e centrada no usuário para medir a velocidade de carregamento percebida porque marca o primeiro ponto na linha do tempo de carregamento da página onde o usuário pode ver alguma coisa na tela (Walton, 2019), (conforme mostra na figura 4): uma FCP rápida ajuda a tranquilizar o usuário que algo está acontecendo.

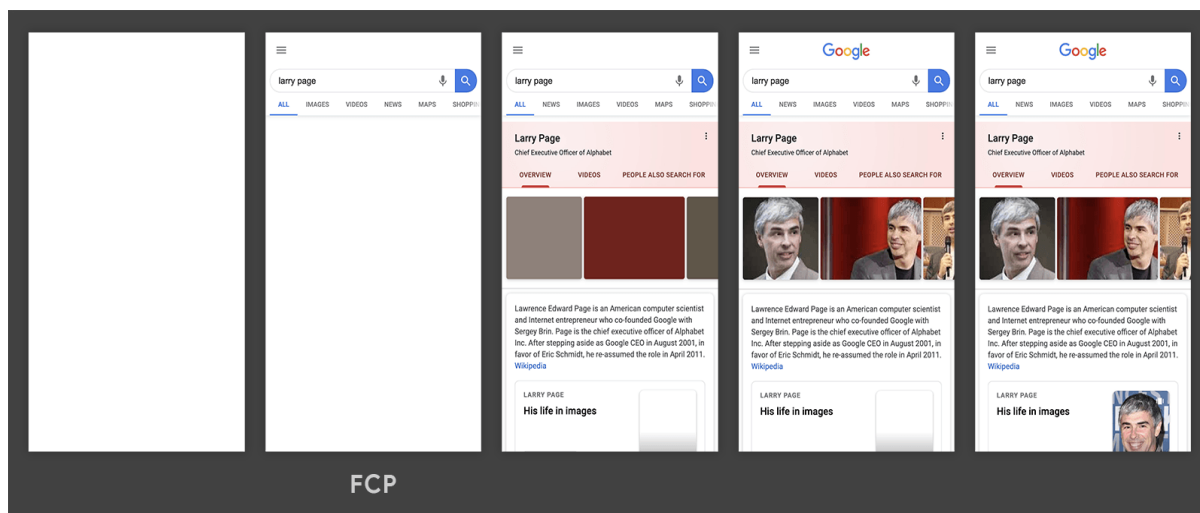


Figura 4. Linha do tempo de carregamento.

Fonte: Walton, 2019.

A FCP mede o tempo desde o início do carregamento da página até o momento em que qualquer parte do conteúdo da página é renderizada na tela (na figura 5 pode-se observar que o tempo máximo sendo “Bom” é de 2 segundos).



Figura 5. FCP.

Fonte: Walton, 2019.

Para fornecer uma boa experiência ao usuário, os *sites* devem se esforçar para ter uma *First Contentful Paint* de 1,8 segundos ou menos. Para garantir que você esteja atingindo essa meta para a maioria de seus usuários, um bom limite para medir é o 75º percentil de carregamentos de página, segmentado através de dispositivos móveis e *desktop* (Walton, 2019).

O texto aborda a importância da velocidade de carregamento percebida de uma página web, sendo essencial ressaltar que a FCP desempenha um papel crítico na satisfação do usuário, pois representa o momento em que algo visível aparece na tela, transmitindo a sensação de progresso.

3.2.5 Speed Index

O *Speed Index* mede a rapidez com que o conteúdo é exibido visualmente durante o carregamento da página. O *Lighthouse* primeiro captura um vídeo do carregamento da página no navegador e calcula a progressão visual entre quadros (Chrome for Developers, 2019), (na figura 6 é mostrado o tempo ideal indicado para o SI).



Figura 6. SI.

Fonte: Baseado em sqlpac, 2020.

Uma boa pontuação SI é de 3,4 segundos ou menos. A qualidade da experiência do usuário é melhor quando o conteúdo acima da dobra aparece mais rápido (Hostinger, 2023).

Essa métrica é crucial para avaliar a rapidez na exibição do conteúdo durante o carregamento da página. O ideal é que essa métrica se mantenha abaixo de 3,4 segundos, já que uma exibição mais rápida do conteúdo acima da dobra proporciona uma experiência do usuário de maior qualidade.

3.3 Lighthouse

O *Lighthouse* é uma ferramenta de código aberto que faz auditorias automáticas para aprimorar a qualidade das páginas da *Web*. Com análises de desempenho, acessibilidade, compatibilidade com *Progressive Web Apps* e muito mais, ele ajuda os desenvolvedores a descobrir o que precisa melhorar nos sites (Google for Developers, 2018).

Segundo o *Web.dev* (2020), citado por Yüksel (2020), o relatório resultante da análise *Lighthouse* contém recomendações detalhadas e documentos sobre como implementar melhorias. Além disso, a conclusão fornece *links* para guias adicionais que os desenvolvedores podem explorar para melhorar ainda mais seu desempenho. Resultados da análise; 0-49 pontos são categorizados como vermelhos e ruins, 50-89 pontos são laranja e precisam ser melhorados e 90-100 pontos são categorizados como bons. De acordo com Yüksel (2020, p. 2016), documentos e guias adicionais compartilhados pelo *Google* mostram que o *Lighthouse* é mais do que apenas uma ferramenta de análise.

De acordo com o *Google for Developers* (2023), a avaliação do desempenho das páginas da *web* é afetada por uma variabilidade intrínseca, proveniente de múltiplos elementos, como a diversidade de redes, *hardware* do usuário e características inerentes do servidor. Essa complexidade desafia a obtenção de medidas consistentes, mesmo quando implementadas práticas de otimização, destacando a importância de estratégias flexíveis de avaliação para lidar com tais variações e garantir uma compreensão mais precisa do desempenho *web*. Isolar fatores externos, executar o *Lighthouse* várias vezes e utilizar valores agregados, como a mediana ou o mínimo, em vez de resultados únicos, pode oferecer uma visão mais estável do desempenho da página.

Lighthouse é muito importante pois o fato de proporcionar recomendações detalhadas e *links* para guias adicionais torna-o uma ferramenta valiosa para otimizar o desempenho

das páginas, e sua ênfase na aplicação de SEO o diferencia como um guia prático para desenvolvedores, sendo muito relevante para os dias atuais.

4. RESULTADO E DISCUSSÃO

A análise do desempenho de aplicações *web* revela métricas desafiadoras que necessitam de atenção. Através dessas abordagens, será analisado três aplicações destacando a importância de realizar análises de desempenho, do qual detalha o que é necessário aprimorar na aplicação, resultando em uma experiência mais satisfatória durante o acesso, utilizando as metodologias quantitativa, quanto à natureza da pesquisa e estudo de caso múltiplo, quanto à escolha do objeto de pesquisa.

4.1 Açaibot

Essa aplicação é um *site* que tem serviços voltados para o *Discord* dedicado a oferecer suporte a jogadores e comunidades de jogos. Ele fornece recursos para a criação de equipes, organização de eventos e interações sociais entre os membros.

Ao avaliar o desempenho desta aplicação *web*, observamos um desempenho geral de 69 pontos (conforme na figura 7).

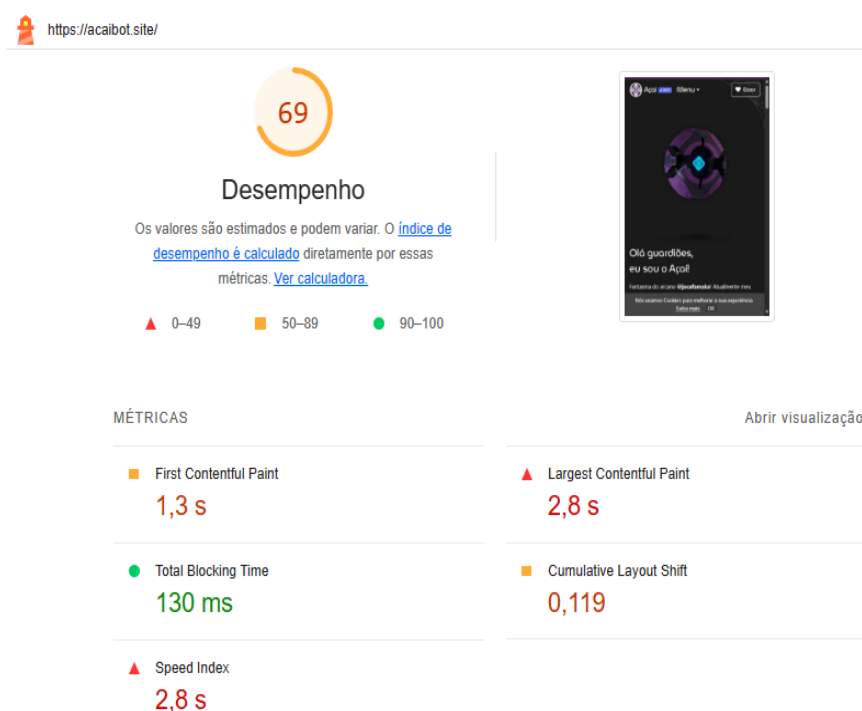


Figura 7. Desempenho da aplicação utilizando o Lighthouse.

Fonte: Autoral, 2023.

Podemos observar que há espaço para melhorias, tendo em vista que o resultado na maioria das métricas esteve por poucos milissegundos em ultrapassar o valor considerado como “Bom”. Vamos analisar as métricas individuais para entender onde as otimizações podem ser aplicadas.

First Contentful Paint (FCP)

Tendo obtido um resultado de 1,3 segundos, uma melhoria nessa métrica pode ser alcançada reduzindo o tempo de resposta inicial do servidor, o que acelera o carregamento da página. Também é importante eliminar recursos que impedem a renderização inicial, evitando bloqueios na primeira exibição da página.

Largest Contentful Paint (LCP)

Essa métrica obteve o pior resultado alcançado 2,8 segundos, sendo 0,3 milisegundos acima do limite. O indicado é usar formatos de vídeo para conteúdo animado em vez de GIFs, que podem ser ineficientes em termos de tamanho de arquivo e desempenho de rede, é recomendado também a utilização de formatos de vídeo como MPEG4 ou WebM para animações, o que economiza *bytes* de rede e melhora a métrica.

Total Blocking Time (TBT)

Apesar de ter obtido um bom resultado de 130 milisegundos, tem um uso excessivo do DOM que está causando um grande uso de memória, o que é uma das principais causas do alto TBT. Para melhorar essa métrica, é importante minimizar o trabalho da *thread* principal, reduzindo a compilação e execução de *JavaScript* sempre que possível.

Cumulative Layout Shift (CLS)

Os elementos de imagem não continham largura e altura explicitamente definidas, pode causar mudanças de *layout*. Definir largura e altura para elementos de imagem pode reduzir essas mudanças e melhorar a métrica CLS que teve um resultado de 0,119.

Speed Index (SI)

O *Speed Index* obteve o resultado de 2,5 segundos, depende da melhoria das outras métricas para obter um melhor resultado geral em toda a aplicação. Para melhorar o *Speed Index*, é fundamental trabalhar nas métricas mencionadas anteriormente, otimizando o carregamento de conteúdo, reduzindo bloqueios da *thread* principal e mantendo a estabilidade visual.

4.2 Estude melhor

Esta aplicação é uma plataforma com cursos de exatas voltados a disciplinas de Matemática, Física, Química e Biologia.

Ao avaliar o desempenho desta aplicação *web*, diversos indicadores de qualidade foram analisados. Começando pelo Tempo de Carregamento, que é avaliado em 71 pontos (conforme a figura 8), indicando uma performance geral sólida.

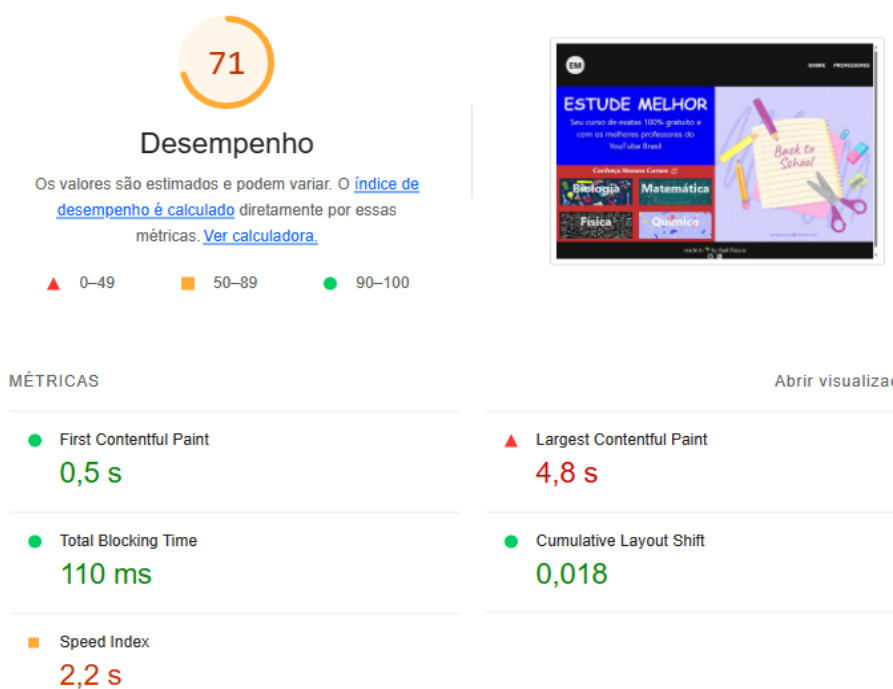


Figura 8. Desempenho da aplicação utilizando o Lighthouse.

Fonte: Autoral, 2023.

No entanto, na análise revela oportunidades de otimização que podem aprimorar a experiência do usuário.

First Contentful Paint (FCP)

Essa métrica obteve uma marca notável, carregando o conteúdo visível em apenas 0,5 segundos. Isso demonstra que o conteúdo inicial é entregue rapidamente aos usuários, o que é crucial para uma impressão inicial positiva.

Largest Contentful Paint (LCP)

No entanto o LCP está ligeiramente atrasado, demorando 4,8 segundos para carregar o maior elemento visível da página. Para melhorar essa métrica, é recomendável pré-carregar a imagem de maior destaque para melhorar o resultado dessa métrica.

Time to Interactive (TTI)

Foi registrado em 110ms, sendo identificado que existem tarefas mais longas na linha de execução principal, o que pode afetar a responsividade da aplicação. Evitar a execução de tarefas críticas na linha principal é fundamental para manter uma experiência de usuário ágil.

Cumulative Layout Shift (CLS)

Essa métrica reportou um valor de 0,018, indicando que a aplicação está propensa a grandes mudanças no *layout*, envolvendo cerca de 5 elementos. Evitar essas alterações drásticas no *design* é crucial para manter a estabilidade visual e garantir que o usuário não seja surpreendido por mudanças bruscas na página.

Speed Index (SI)

Investir em otimizações direcionadas para FCP, LCP e CLS deve, por consequência, melhorar o *Speed Index*, do qual já obteve de 2,2 segundos e que mede a eficiência geral da aplicação.

4.3 Na memória

Blog que tem por objetivo o compartilhamento de experiências do qual o usuário precisa fazer o *login* e automaticamente já tem o acesso para fazer uma publicação descrevendo alguma experiência para uma determinada categoria disponível.

A aplicação está operando de forma razoavelmente eficiente, obtendo uma pontuação de 82 (conforme na figura 9).

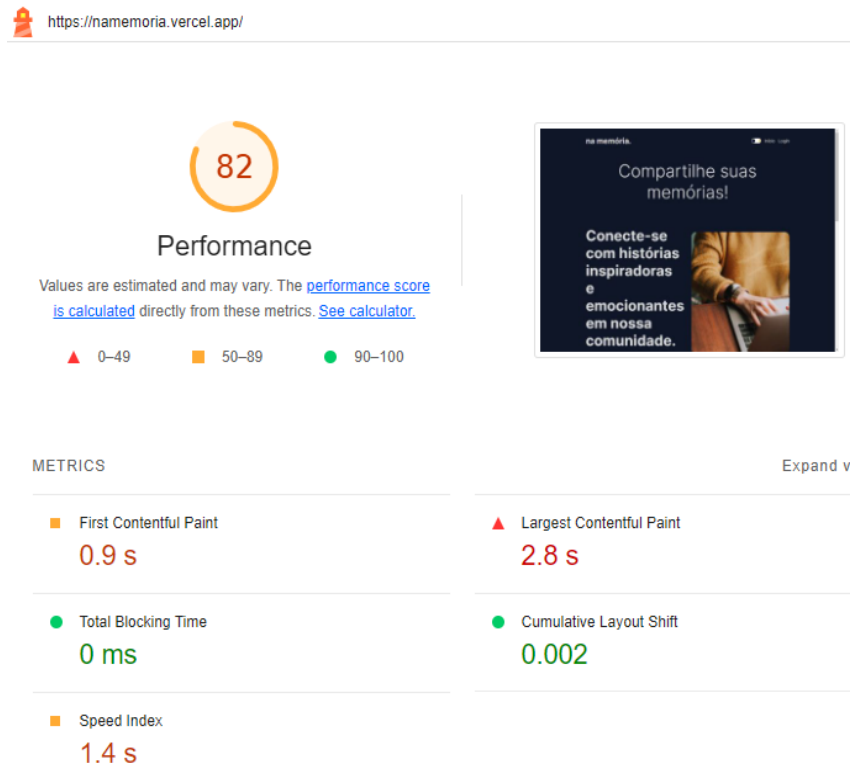


Figura 9. Desempenho da aplicação utilizando o Lighthouse.

Fonte: Autoral, 2023.

No entanto, algumas métricas podem ser melhoradas para proporcionar uma experiência ainda melhor, a seguir analisado o resultado de cada métrica após a análise.

First Contentful Paint (FCP)

Com um tempo de 0,9 segundos, o FCP destaca que o primeiro conteúdo visível está sendo entregue em um período aceitável. No entanto, a recomendação de eliminar recursos de bloqueio de renderização é crucial, pois os recursos estão retardando a primeira visualização da página. Além disso, a sugestão de entregar JavaScript e CSS críticos *in-line* e adiar aqueles não críticos pode resultar em melhorias referente a métrica.

Largest Contentful Paint (LCP)

A métrica LCP, que mede o maior elemento de conteúdo mostrado na janela de visualização, registrou um tempo de 2,8 segundos. Esta é uma área em que melhorias podem ser significativas. O "título central" é apontado como o maior contribuinte, e otimizações direcionadas a esse elemento podem reduzir o LCP.

Total Blocking Time (TBT)

A métrica TBT atingindo 0ms é positiva, pois indica que não há bloqueio da thread principal. Isso é crucial para uma boa experiência do usuário, uma vez que bloqueios na *thread* principal podem prejudicar a interatividade da página.

Cumulative Layout Shift (CLS)

O CLS, medindo mudanças de *layout*, tem um valor de 0.002, o que é excelente. No entanto, a análise ressalta que elementos DOM específicos, como botões do tema, *login* e início, contribuem mais para o CLS da página. Portanto, a sugestão é evitar grandes mudanças de *layout* e manter esses elementos mais estáveis.

Speed Index (SI)

Com um valor de 1,4 segundos indica um bom desempenho na capacidade da página de mostrar conteúdo visível aos usuários rapidamente. Esse é um indicador promissor, já que uma pontuação mais baixa no *Speed Index* está associada a um carregamento mais rápido da página.

5. ANÁLISE DE DESEMPENHO

Neste estudo, foi analisado três aplicações web diferentes usando métricas essenciais do *Core Web Vitals* e a ferramenta *Google Lighthouse*, utilizando a metodologia quantitativa, quanto à natureza da pesquisa e estudo de caso múltiplo, quanto à escolha do objetivo de estudo. Essas aplicações englobam uma variedade de finalidades, desde aplicação com serviços dedicado a comunidades de jogos até uma aplicação de cursos de exatas e um blog. Ao avaliar com base nas métricas de desempenho, cada aplicação teve resultados que precisam de otimização. Além disso, comparamos o desempenho relativo dessas aplicações, analisando por meio de tabelas os resultados obtidos em cada análise com os valores máximo que o *Google* considera como “Bom” para cada métrica e destacando a aplicação que tem o melhor desempenho.

5.1 Açaibot

No Gráfico 1, podemos observar que os resultados das métricas obtidos após análise da aplicação estão dentro do limite considerado como “Bom”, com exceção das métricas LCP e CLS que ultrapassam esse limite.

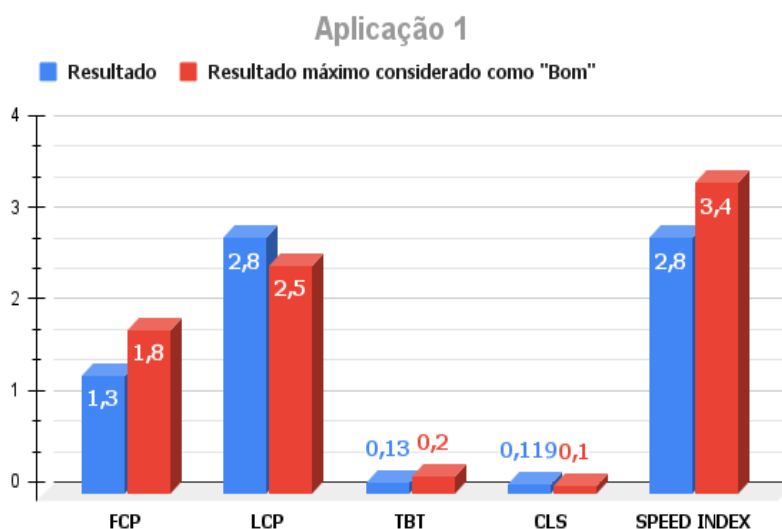


Gráfico 1. Comparativo entre resultados obtidos após análise e valores máximos considerados como “Bom”.

Fonte: Autoral, 2023.

A análise de desempenho revela que existem áreas críticas que requerem atenção e otimização. O tempo de resposta inicial do servidor é uma métrica que precisa ser melhorada para acelerar o carregamento da página. Além disso, a eliminação de recursos que bloqueiam a renderização é essencial para evitar bloqueios na primeira exibição da página. A otimização de imagens também é necessária para acelerar o carregamento. Embora haja espaço para aprimoramentos, as métricas gerais demonstram uma base sólida para futuras melhorias. Comparativamente, essa aplicação se encontra em um estágio intermediário em relação às outras duas.

5.2 Estude melhor

Na Gráfico 2 os resultados das métricas obtidos após análise da aplicação em sua maioria estão dentro do limite considerado como “Bom”, com exceção da métrica *Largest Contentful Paint* (LCP) que ultrapassa muito esse limite e requer uma maior atenção.

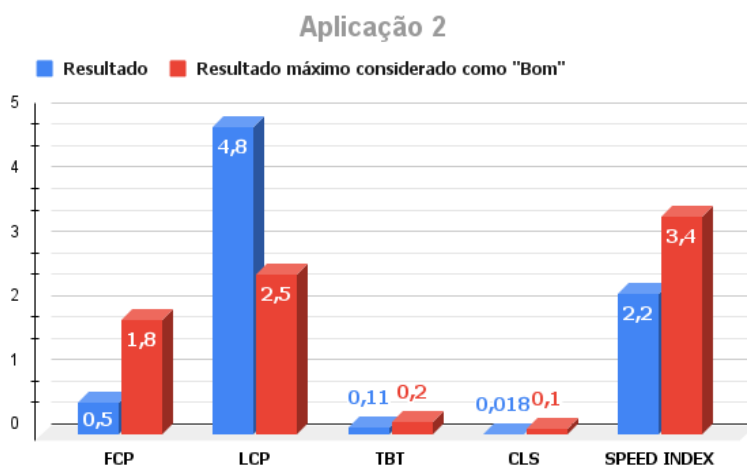


Gráfico 2. Comparativo entre resultados obtidos após análise e valores máximos considerados como “Bom”.

Fonte: Autoral, 2023.

A análise de desempenho mostra um cenário geral aceitável, mas com potencial de otimização. O (FCP) é um destaque positivo muito importante, pois está carregando o conteúdo visível rapidamente. No entanto, o (LCP) requer atenção tendo o tempo de 4,8 segundos, um destaque negativo já que está em um valor muito superior do que deveria e com isso demora mais para carregar o maior elemento visível da página. Melhorar o FCP e o LCP por meio da eliminação de recursos que bloqueiam a renderização e do uso de formatos de imagens mais leves é essencial. As métricas de interatividade e estabilidade visual estão em um nível satisfatório. Comparativamente, esta aplicação apresenta um desempenho intermediário entre as três.

5.3 Na memória

Na Gráfico 3 podemos observar que os resultados das métricas obtidos após análise da aplicação em sua maioria estão dentro do limite considerado como “Bom”, sendo algo positivo para a experiência do usuário, a única exceção fica para métrica LCP que ultrapassou 0,3 milissegundos desse limite.

Aplicação 3

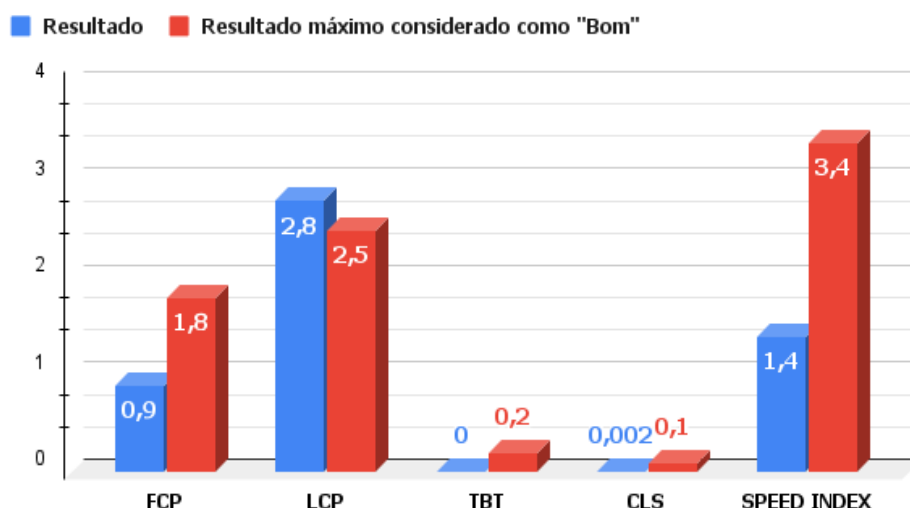


Gráfico 3. Comparativo entre resultados obtidos após análise e valores máximos considerados como “Bom”.

Fonte: Autoral, 2023.

A análise revela que a aplicação possui um desempenho sólido, com algumas métricas, como o LCP, oferecendo espaço para otimizações. Ao eliminar recursos de bloqueio de renderização, entregar recursos críticos *in-line* e minimizar mudanças de *layout*, a experiência do usuário pode ser aprimorada.

Considerando a pontuação geral de desempenho, a aplicação na memória lidera com a pontuação de desempenho mais alta de 82, indicando um desempenho relativamente melhor. Em seguida, a aplicação Estude Melhor apresenta um desempenho intermediário com uma pontuação de 71, enquanto a aplicação Açaibot fica com o desempenho mais fraco, com uma pontuação de 69. O *Largest Contentful Paint* (LCP) é uma métrica crítica que requer melhorias em todas as aplicações, e o *Total Blocking Time* (TBT) varia em eficiência, sendo uma área de destaque nas aplicações na memória e Açaibot. Portanto, todas as aplicações têm margem para melhorias em seus desempenhos gerais e métricas específicas.

O estudo não foca somente nos conceitos teóricos, mas se destaca ao oferecer uma abordagem prática e detalhada para analisar o desempenho de três aplicações web usando as métricas de desempenho presentes no *Google Lighthouse*, identificando áreas em cada aplicação que necessitam de otimização, além de fornecer *insights* práticos para melhorar o desempenho.

6. CONCLUSÃO

Neste artigo, foi realizado uma análise das métricas de desempenho em três aplicações web, avaliando métricas que compõe o *Google Lighthouse*. Através dessa análise, foi revelado os pontos fortes e as áreas que necessitam de melhorias em cada aplicação. Ao destacar a importância do desempenho nas aplicações web, mostramos como as métricas impactam a experiência do usuário e a classificação nos mecanismos de busca.

Olhando para o futuro, vemos uma série de oportunidades para aprimorar ainda mais o desempenho. A inclusão de técnicas de carregamento assíncrono, minificação de código podem elevar as métricas de desempenho a um novo patamar. Além disso, a análise contínua das métricas, bem como o acompanhamento de como essas métricas afetam a

experiência do usuário e as classificações nos mecanismos de busca, permanece crucial. Os trabalhos futuros devem se concentrar na busca contínua da otimização do desempenho, aproveitando as tecnologias e metodologias emergentes.

A necessidade de manter um equilíbrio entre a entrega de conteúdo atraente e a busca por melhorias no desempenho nem sempre é fácil. No entanto, é essa complexidade que torna a otimização de desempenho um campo empolgante e inestimável no desenvolvimento *web*. À medida que novas tecnologias e ferramentas emergem, enfrentamos esses desafios com determinação e perseverança, impulsionados pelo objetivo de oferecer experiências de usuário excepcionais na *web*.

REFERÊNCIAS

Chrome for Developers. **Speed Index**, 2019. Disponível em: <https://developer.chrome.com/docs/lighthouse/performance/speed-index>. Acesso em: Agosto de 2023.

Crystallize. **TBT**. Disponível em: <https://www.crystallize.com/learn/best-practices/frontend-performance/other-web-vitals/total-blocking-time-tbt>, [s.d.]. Acesso em: Agosto de 2023.

Google for Developers. **Lançamento da categoria de auditorias de SEO na extensão do Chrome do Lighthouse**, 2018. Disponível em: <https://developers.google.com/search/blog/2018/02/seo-audit-category-in-lighthouse?hl=pt-br>. Acesso em: Agosto de 2023.

Google for Developers. **Variabilidade do Lighthouse**, 2023. Disponível em: <https://developers.google.com/web/tools/lighthouse/variability?hl=pt-br>. Acesso em: Setembro de 2023.

HERIČKO, T.; ŠUMAK, B.; BRDNIK, S., **Towards representative web performance measurements with Google Lighthouse**, 2021. DOI:10.18690/978-961-286-516-0.9.

MCQUADE, Bryan. **Definindo os limites das métricas Core Web Vitals**, 2020. Disponível em: <https://web.dev/i18n/pt/defining-core-web-vitals-thresholds>. Acesso em: Agosto de 2023.

PÍCOLO, Mariana Nunes. **Otimização de Desempenho Orientado ao Tempo de Carregamento em Aplicações Web Front-end**, 2021. Universidade de Brasília, Faculdade UnB Gama, Brasília. Acesso em: <https://bdm.unb.br/handle/10483/30632>.

Kinsta. **Um guia sobre otimização de velocidade de sites para iniciantes**, 2023. Disponível em: <https://kinsta.com/pt/aprenda/velocidade-de-sites>. Acesso em: Agosto de 2023.

VASILJEVIĆ, V.; KOJIĆ N.; VUGDELIJA N., **A new approach in quantifying user experience in web-oriented applications**. Itema Conference, 2020. DOI: 10.31410/ITEMA.2020.9.

Sqlpac. **Speed Index**, 2020. Disponível em: <http://www.sqlpac.com/en/documents/google-pagespeeds-lighthouse-6-new-features-lcp-clr.html>. Acesso em: Agosto de 2023.

WALTON, Philip. **[Cumulative Layout Shift]**, 2019. Disponível em: <https://web.dev/cls>. Acesso em: Agosto de 2023.

WALTON, Philip. **[First Contentful Paint]**, 2019. Disponível em: <https://web.dev/fcp>. Acesso em: Agosto de 2023.

WALTON, Philip. **[Largest Contentful Paint]**, 2019. Disponível em: <https://web.dev/lcp>. Acesso em: Agosto de 2023.

WALTON, Philip. **[Total Blocking Time]**, 2019. Disponível em: <https://web.dev/tbt>. Acesso em: Agosto de 2023.

WALTON, Philip. **[Web Vitals]**, 2019. Disponível em: <https://web.dev/i18n/pt/vitals>. Acesso em: Agosto de 2023.

YÜKSEL, D.; SÜMBÜLTEPE, K.; TOLON, M., **Arama Motoru Optimizasyonu (SEO) Analiz Skoruiile Sıralama İlişkisi: Google Lighthouse**, 2020. DOI: 10.20491/isarder.2020.965.



2

DESENVOLVIMENTO DE APP PARA GESTÃO MÉDICA EM NEUROLOGIA COM *FRONT END EM FLUTTER* E API REST COM *SPRING BOOT*

*Development of a REST API using Spring boot for a Front End
developed in Flutter*

Alanna Rocha Monteiro
Edilson Carlos Silva Lima
Yonara Costa Magalhães

D.O.I.: 10.29327/5391075.1-2



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Visando uma melhor segurança de dados cadastrais de pacientes em uma Unidade de Terapia Intensiva, leitos e análise da evolução no quadro clínico periodicamente, por meio de médicos e/ou residentes *neurologistas*, o presente artigo tem como objetivo relatar o desenvolvimento de um *APP* para tais fins. A aplicação será consumida pela linguagem de programação *flutter* e uma *API Spring Boot*, com *API* e *Design Patterns*, onde haverá telas de cadastro de pacientes, registrar informações cruciais, como dados pessoais, histórico médico, inclusive, oferecer a capacidade de gerenciar leitos de internação, incluindo a disponibilidade, ocupação e status dos pacientes internados. Além disso, armazenamento de questionários, respondidos por cada usuário referente ao protocolo de avaliação de diagnóstico por serviços de saúde. Este aplicativo representa um avanço significativo no gerenciamento de dados em neurologia, aumentando a eficiência no atendimento ao paciente, reduzindo a probabilidade de erros de documentação e melhorando a qualidade dos cuidados prestados. Sendo assim, será avaliado através de um questionário de 4 perguntas para fins investigativos, qualitativos e quantitativos.

Palavras-chave: *Neurologistas, APP, API Spring Boot, Design Patterns, Flutter.*

Abstract

Aiming at better security of registration data of patients in an Intensive Care Unit, beds and analysis of the evolution in the clinical picture periodically, through physicians and/or neurologist residents, this article aims to report the development of an *APP* for such ends. The application will be consumed by the *flutter* programming language and a *Spring Boot API*, with *API* and *Design Patterns*, where there will be patient registration screens, record crucial information, such as personal data, medical history, including offering the ability to manage hospital beds, including the availability, occupancy, and status of inpatients. In addition, storage of questionnaires answered by each user regarding the diagnostic evaluation protocol by health services. This application represents a significant advance in data management in neurology, increasing efficiency in patient care, reducing the likelihood of documentation errors and improving the quality of care provided. Therefore, be evaluated through a questionnaire of four questions for investigative, qualitative and quantitative purposes.

Keywords: *Neurologists, APP, Spring Boot API, Design Patterns, Flutter.*

1. INTRODUÇÃO

A medicina moderna está cada vez mais incorporando a tecnologia para otimizar o diagnóstico, tratamento e acompanhamento de pacientes. Na especialidade da neurologia, onde o registro preciso de dados clínicos e a avaliação minuciosa de sintomas são cruciais para a tomada de decisões clínicas, a implementação de aplicativos móveis específicos pode desempenhar um papel fundamental na melhoria dos cuidados aos pacientes. Neste contexto, este artigo explora o desenvolvimento de um aplicativo destinado a médicos e/ou residentes neurologistas, projetado para a gestão de informações cadastrais de pacientes, controle de leitos de internação e preenchimento de formulários de avaliação clínica.

Os avanços tecnológicos têm revolucionado a medicina, proporcionando ferramentas que aprimoram a precisão e a eficiência do atendimento médico. Na neurologia, onde o tempo é muitas vezes um fator crítico, a capacidade de acessar rapidamente informações detalhadas sobre pacientes e suas condições clínicas é de extrema importância. Além disso, a especialidade demanda a utilização de escalas de avaliação, testes neurológicos específicos e a compilação de dados precisos para a monitorização de doenças neurológicas crônicas. O desenvolvimento de um aplicativo que atenda a essas necessidades específicas representa um passo significativo em direção à melhoria da qualidade dos cuidados neurológicos.

Para alcançar melhores resultados em conformidade com os objetivos do projeto, foram empregados ao desenvolvimento as seguintes metodologias: Estudo de caso, para que a raiz do problema seja descrita e explicitada, e também entrevista com o público-alvo para ter um *feedback* da funcionalidade e satisfação dos usuários do sistema. Nesse sentido, o aplicativo busca, por meio de formulários, com perguntas discursivas e específicas, fazer com que o controle de pacientes em UTI neurológica seja mais prático e eficiente, através da tecnologia.

Sendo assim, o processo de criação e implementação desse aplicativo, destacando suas funcionalidades, usabilidade e potencial impacto na prática clínica em neurologia é bastante promissor. Diante disso, examina também como a tecnologia móvel pode aprimorar a coleta de dados, promover uma documentação mais precisa e facilitar o acompanhamento longitudinal dos pacientes neurológicos. A integração bem-sucedida dessas inovações na rotina dos profissionais de saúde pode oferecer benefícios substanciais tanto para os médicos quanto para os pacientes, contribuindo para uma neurologia mais eficiente e prática, que será discutida em cada capítulo deste artigo.

2. TRABALHOS RELACIONADOS

Nesta seção, será apresentado uma análise de três artigos relevantes, contextualizando o *app* de gestão de clínicas neurológicas em relação a trabalhos anteriores, destacando em relação aos métodos tradicionais, evidenciando sua abordagem também inovadora, ainda que haja ideias previamente já introduzidas. Esta revisão não apenas sublinha as vantagens do projeto, inclusive justifica a escolha específica desses trabalhos, enriquecendo o entendimento do cenário acadêmico e prático em que o *app* se destaca, dando continuidade ao desenvolvimento.



2.1 Desenvolvimento de uma API REST utilizando Spring boot para um Front End desenvolvido em Flutter

Ainda que o artigo foque também no desenvolvimento de uma API para um aplicativo na área médica neurológica, o projeto atual se destaca ao incorporar funcionalidades de edição de formulários, com o cadastro e edição de perguntas, através de descrição, grupo e em cada uma delas a alteração das opções de respostas. A contribuição diferenciada reside na capacidade de personalização avançada oferecida pelo *app*, destacando-se pela resposta dinâmica às demandas clínicas. A habilidade de personalizar formulários e adicionar perguntas específicas não apenas expande as utilidades, mas também o posiciona como uma solução mais adaptável e abrangente no âmbito da gestão de clínicas neurológicas, além do acesso estendido a residentes, representa um avanço significativo em relação ao projeto mencionado. (SILVA, 2022).

2.2 Desenvolvimento de um Aplicativo Para a Avaliação de Pacientes Neurológicos na UTI Utilizando o Framework Flutter

Em comparação com este que se concentrou no desenvolvimento front-end, o *app* se destaca ao integrar recursos ausentes nesse projeto anterior. Foi adicionado ícones para o cadastro de novas perguntas/formulários, proporcionando uma experiência mais rica no gerenciamento ao usuário. Além disso, diferentemente do projeto mencionado, a aplicação atual é estendida aos residentes, promovendo uma abordagem mais inclusiva, incluindo em andamento a implementação de notificações e alertas e segregação por unidade hospitalar. Esses aprimoramentos significativos solidificam a posição do projeto em relação às demandas práticas da gestão de clínicas neurológicas (CONCEIÇÃO, 2022).

2.3 Importância da Avaliação Neurológica no Paciente Crítico: Um Relato de Experiência

Já o terceiro artigo, de notável relevância, aborda de maneira abrangente a importância crítica da avaliação neurológica em ambientes de Unidade de Terapia Intensiva (UTI), ressaltando a necessidade premente de ferramentas eficazes que aprimorem a gestão e a documentação desse processo crucial. Dessa maneira, a introdução da tecnologia como meio de aprimorar a avaliação neurológica representa um avanço significativo no campo da neurologia e o atual projeto busca não apenas otimizar a documentação, mas também contribuir para avanços na compreensão e gestão de condições neurológicas em ambientes críticos, além de reduzir chances de perda de documentos ou praticidade nas atualizações de quadro clínico. Este enfoque inovador destaca a utilidade prática do aplicativo, inclusive, seu impacto positivo na evolução contínua da prática neurológica em UTI (ROCHA, 2019).

3. FUNDAMENTAÇÃO TEÓRICA

Este tópico, foram discutidas as bases teóricas e conceituais que sustentam o desenvolvimento do aplicativo dedicado à neurologia, destacando as tecnologias e metodologias que orientaram o processo de criação.

3.1 MVC (Model-View-Controller)

O padrão arquitetural Model-View-Controller (MVC), tomando como hábito o estudo de Gamma *et al.* (2007, p.20), é fundamental para o desenvolvimento de aplicações web e móveis robustos e organizadas. Designado para a aplicação em três componentes principais: o Modelo, que lida com a lógica de negócios e os dados, sendo o objeto de aplicação; a Visão, que representa a interface do usuário, apresentação na tela; e o Controlador, responsável por gerenciar a interação entre o Modelo e a Visão. No contexto do aplicativo de neurologia, o MVC foi adotado para garantir uma clara separação de responsabilidades, facilitando a manutenção e escalabilidade do sistema, além de um padrão de arquitetura de software responsável por contribuir na otimização da velocidade entre as requisições feitas pelo comando dos usuários (ZUCHER, 2020).

3.2 UML (Unified Modeling Language)

A Unified Modeling Language (UML) é uma linguagem de modelagem padronizada que oferece um conjunto de diagramas para representar visualmente diferentes aspectos de um sistema de software, como relata Guedes (2018) em seu livro sobre essa abordagem prática: há uma diferença notável entre construir uma casa e construir um prédio de diversos andares, um pedreiro não conseguiria construir um prédio sem um projeto. Dito isso, para o projeto do *app*, foram utilizados diagramas UML, como diagramas de caso de uso e diagramas de classes, para modelar a estrutura, os relacionamentos e os fluxos de trabalho da aplicação. Esses diagramas são cruciais para a compreensão geral do sistema e para a comunicação eficaz entre as necessidades do usuário e o desenvolvimento, que são descritos nos próximos tópicos.

3.2.1 Diagrama de Casos de Uso

Segundo a IBM (2021), na UML, os diagramas de caso de uso representam a dinâmica de um sistema, sendo instrumentais na captura dos requisitos e na identificação das interações entre o sistema e seus agentes, isto é, ainda que abstraído-se dos detalhes internos do funcionamento, elucidam as funcionalidades do sistema e a maneira pela qual os atores interagem com ele.

Dito isso, em relação aos elementos de modelos em diagramas de caso de uso, são eles: caso de uso, que enfatizam a função que o sistema exerce na consecução dos objetivos do usuário; os agentes, expressando as operações do usuário que interagem com o sistema em processo de modelagem; os subsistemas, elemento prototípico que simboliza uma entidade autônoma de comportamento em um sistema e os relacionamentos, estabelecendo a estrutura e a dinâmica entre os elementos do modelo, para, por fim, contextualizar para um melhor entendimento do processo de desenvolvimento do projeto (LEANDRO, 2012).

3.2.2 Diagrama de Classes

Na UML, os diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema. São as cópias do sistema ou subsistema. Você



pode utilizar os diagramas de classe para modelar os objetos que compõem o sistema, para exibir os relacionamentos entre os objetos e para descrever o que esses objetos fazem e os serviços que eles fornecem (IBM, 2021).

Sendo assim, ainda de acordo com a IBM (2021), os diagramas de classes são úteis em muitos estágios do projeto do sistema, começando no estágio de análise, onde auxilia a compreender os requisitos do domínio do problema e a identificação de seus componentes, além de que os diagramas de classes elaborados nas fases iniciais do projeto englobam classes que frequentemente evoluem para objetos e classes de software concretas à medida que o código é desenvolvido. Posteriormente, é possível aprimorar os modelos analíticos e conceituais anteriores através de diagramas de classe, os quais detalham partes específicas do sistema, como a interface do usuário, a lógica de implementação, entre outros aspectos. O diagrama de classes, nesse contexto, transforma-se em um instantâneo preciso que delinea minuciosamente o funcionamento do sistema, as inter-relações entre os componentes em diversos níveis e a estratégia de implementação desses elementos.

3.3 Design Patterns (Padrões de Projeto)

Os Design Patterns, ou Padrões de Projeto, são constituídos a partir de quatro elementos essenciais, segundo Gamma et al. (2007): nome do padrão, o problema, a solução e as consequências, isto é, a partir de padrões para o desenvolvimento de projetos, trazer consigo a facilidade do uso de soluções reutilizáveis para problemas comuns no processo de criação do software, bem como a documentação. Eles fornecem abordagens testadas e comprovadas para resolver desafios de design e arquitetura. Ao aplicar Design Patterns no desenvolvimento do aplicativo, buscamos aprimorar a escalabilidade, manutenção e flexibilidade do sistema, como conferidos ao longo do artigo.

De acordo com Serapião e Ribeiro (2004, p.32), os padrões de criação são cruciais para abstrair o processo de instanciação, tornando um sistema independentemente de como seus objetos são criados, compostos e representados. Nesse sentido, ainda se dividindo entre estruturais, concentra-se na organização de classes e objetos para criar estruturas mais amplas, onde empregam a herança para combinar interfaces ou implementações (SERAPIÃO; RIBEIRO, 2004, p.45). Exemplos desses padrões incluem o *decorator* e o *adapter*. Além disso, encontramos padrões comportamentais, como o *observer* e o *template method*, que, por sua vez, se preocupam com o comportamento e interação dinâmica entre objetos.

3.4 Front End

Segundo Brunno Kriger (2023), o *front-end* é uma área da programação que se dedica a criar a parte visual e interativa de um site, aplicativo ou *software*. É o que o usuário vê e usa quando acessa uma plataforma digital. Nesse sentido, destaca-se pela responsividade, interação do usuário, performance, que serão concretizadas durante o desenvolvimento do projeto.

3.5 Dart

De acordo com Barro (2023), *Dart* é uma linguagem de programação versátil, multi-paradigma, empregada no desenvolvimento de aplicações *web*, móveis e *desktop*. Em-

bora possa não ter a mesma popularidade de algumas linguagens mais estabelecidas, *Dart* possui uma comunidade ativa e crescente, uma eficiência e desempenho, oferecendo a flexibilidade de executar código tanto no lado do cliente quanto no servidor, proporcionando uma abordagem versátil para a criação de soluções em diferentes plataformas, além do gerenciamento eficiente de estado em aplicações, por isso sua ampla utilização.

3.6 Flutter

Segundo Matheus Alberto (2023), *Flutter* é um *framework* (ferramenta) de desenvolvimento com foco multiplataforma em dispositivos móveis. Criado pela *Google*, é bastante utilizado no mercado e, mais recentemente, permite a criação de aplicações para *desktop* (*Linux*, *Windows* e *macOS*). Sendo assim, oferece uma abordagem moderna e eficaz para o desenvolvimento de aplicativos móveis, destacando-se pela capacidade de criar interfaces usuário personalizadas e performáticas em várias plataformas a partir de um único código.

3.7 Framework Spring e API REST

O *framework Spring* é amplamente reconhecido por simplificar o desenvolvimento de aplicativos *Java*, fornecendo um conjunto de ferramentas e bibliotecas para criar aplicativos robustos e eficientes. Dito isso, segundo TechTargetContributor (2021), é uma estrutura de aplicativo de código aberto que fornece suporte de infraestrutura, como uma das estruturas *Java Enterprise Edition* (*Java EE*), ajudando os desenvolvedores a construir aplicativos de alto desempenho usando *POJOs* (*Plain Old Java Objects*). No contexto do aplicativo neurologia, o *Spring Boot* foi escolhido devido à sua capacidade de suportar a criação de serviços web *RESTful*. Por consequente, a arquitetura baseada em *API REST* (*Representational State Transfer*) permite a comunicação eficiente entre o aplicativo móvel e o servidor, possibilitando a recuperação e a atualização de dados de forma simples e segura.

3.7.1 Arquitetura REST

Partindo do princípio de que uma *API* (*Application Programming Interface* ou *Interface de Programação de Aplicação*) pode ser definida todo um conjunto de padrões que possibilita a construção de aplicativos, onde ele conecta aplicações (GUEDES, 2019), isto é, diretrizes que viabiliza a intercomunicação entre plataformas, a arquitetura *REST*, segundo Lima (2020), é um dos modelos de arquitetura que fornece orientação para sistemas distribuídos se comunicarem diretamente usando princípios e protocolos da *Web* existentes, ou seja, um estilo estrutural que fornece um conjunto de princípios e restrições para o design de serviços web, orientando a construção de *APIs*.

Visando um melhor entendimento do funcionamento, ainda de acordo com Lima (2020), existem cinco métodos: o método *GET*, que é amplamente empregado para solicitar recursos do servidor; o método *POST*, projetado para enviar dados de entrada ao servidor, comumente utilizado em formulários *HTML*, já o método *PUT* é utilizado para editar e atualizar documentos no servidor e o método *DELETE*, conforme o próprio nome indica, é responsável pela exclusão de dados ou coleções no servidor, configurando os principais métodos utilizados no projeto.



Nesse sentido, assim como os métodos aplicados, o retorno de cada resposta proveniente de um *app REST*, também é de suma importância, envia um código que especifica o estado da requisição, são eles, os principais: 200 (*OK*), indicando conclusão bem-sucedida da requisição; 201 (*CREATED*), sinalizando criação bem-sucedida de objeto ou recurso; 204 (*SEM CONTEÚDO*), para êxito na exclusão de objeto ou recurso; 400 (*BAD REQUEST*), indicando erro na requisição devido a diversos motivos; 404 (*Não Encontrado*), para indicação de que a rota ou coleção não foi encontrada; e 500 (*Erro de Servidor Interno*), apontando a ocorrência de um erro no servidor, para, por fim, ter um melhor monitoramento, um *feedback* para o desenvolvedor, segurança na aplicação, além do trato adequado de situações de erro (LIMA, 2020).

3.8 Metodologia XP (Extreme Programming)

O *extreme Programming* é uma metodologia ágil de desenvolvimento de *software* voltada para times de pequeno a médio porte, no qual os requisitos são vagos e mudam frequentemente (WILDT, 2015, p.16). Dessa forma, a metodologia de desenvolvimento ágil *Extreme Programming* (XP) enfatiza a colaboração próxima entre os membros da equipe, a entrega frequente de *software* funcional e a capacidade de resposta às mudanças nos requisitos do cliente. No desenvolvimento do aplicativo, adotamos os princípios do XP para garantir um processo iterativo e flexível. Isso permitiu que a equipe de desenvolvimento se adaptasse às necessidades em constante evolução da neurologia clínica, fornecendo um aplicativo que atende de maneira eficaz às demandas dos profissionais de saúde e de seus pacientes.

4. PROCEDIMENTOS METODOLÓGICOS

Levando em consideração que o projeto em questão visa a iteração e melhoria contínua no processo de desenvolvimento do aplicativo, foram adotadas abordagens de cunho investigativo, coletando dados, realizando análises e tomando decisões com base em questionário e entrevistas, voltadas para o funcionamento e praticidade, posteriormente conferidos pelo *app*.

Consoante a isso, além de manter a ética e privacidade dos dados, é aliado ao método qualitativo, para compreender as necessidades e preferências dos usuários, bem como para coletar *feedback* qualitativo durante o processo, para, por fim, relatar em resultados e discussões. Sendo assim, foi desenvolvido uma *API Spring Boot* para ser consumida por uma aplicação mobile, com o objetivo de solucionar problemas de falta de diligência, praticidade e perda de documentação ou erro médico por confusão em dados de pacientes neurológicos internados em UTI's.

Para nortear os resultados do projeto, foram feitas entrevistas com 35 pessoas, dentre elas, médicos e residentes, para que fosse possível a avaliação do aplicativo e se foi bem-sucedida a conclusão da resolução da problemática proposta para o *App*. Nesse sentido, as perguntas feitas se diferenciaram em quatro etapas: a expectativa dos usuários, a responsabilidade do *app*, intuitividade e melhorias a serem posteriormente implementadas, colaborando para os resultados com vantajosa porcentagem de satisfação que serão descritas nos tópicos de resultados e discussões.

Nos tópicos seguintes exploraremos o desenvolvimento prático do aplicativo voltado para médicos e residentes em neurologia, bem como a integração da aplicação com a *API*

que suporta seu funcionamento, abrangendo diversas fases, desde a criação do aplicativo até a análise de dados finais, com o desenvolvimento da *API* e a validação da comunicação entre ela e a aplicação, incluindo feedback do projeto.

5. RESULTADOS E DISCUSSÃO

5.1 Desenvolvimento da Aplicação

Com o intuito de facilitar o dinamismo, a praticidade e a segurança na questão do gerenciamento médico de dados de pacientes, leitos e quadros clínicos através de formulários avaliativos em neurologia, estes são cadastrados pelos médicos, com perguntas específicas sobre o estado de saúde do paciente, afim de ter uma maior praticidade e precisão de diagnóstico, especialmente em Unidade de Terapia Intensiva, além de evitar perda ou trocas de documentação, descreve-se o desenvolvimento de uma *API Spring boot* que será consumida por uma aplicação *Flutter*. Dito isso, a partir do usuário cadastrado (médico ou residente), terão acesso aos dados de seus pacientes, leitos, armazenamento dos questionários de quadro clínico e outras funcionalidades similares, tudo a partir do *app*.

O processo teve início com a delimitação dos requisitos funcionais, sendo eles cadastro do usuário, criar e editar formulários, acompanhar avaliações, a partir dos quais foram desenvolvidos o modelo da interface e os diagramas.

De fundamental importância para a visualização das funcionalidades básicas e requisitos do sistema, bem como a documentação acerca da idealização do projeto, o processo de análise corroborou para a definição de cada ação e relacionamento presente no sistema a ser desenvolvido. Mediante a análise realizada, tornou-se viável a elaboração dos diagramas, servindo como alicerces para a integralidade desse processo de criação.

O Diagrama de Casos de Uso foi o ponto de partida, sendo apresentado de imediato na representação visual, conforme ilustrado na figura 1, com os atores, à esquerda (residente) e logo posteriormente o médico. Por conseguinte, como é descrito no diagrama abaixo, foi especificado cada função ou recurso específico fornecido pelo sistema, através de cada ator, isto é, papel desempenhado por um usuário, anteriormente citados. Nesse sentido, cada médico e residente terão acesso à listagem de pacientes cadastrados, criar e acompanhar avaliações, consultar histórico de pacientes, com alguns diferenciais por parte do médico, por ter sido estendidas as funcionalidades a ele de criar e editar formulários e listagem de residentes.

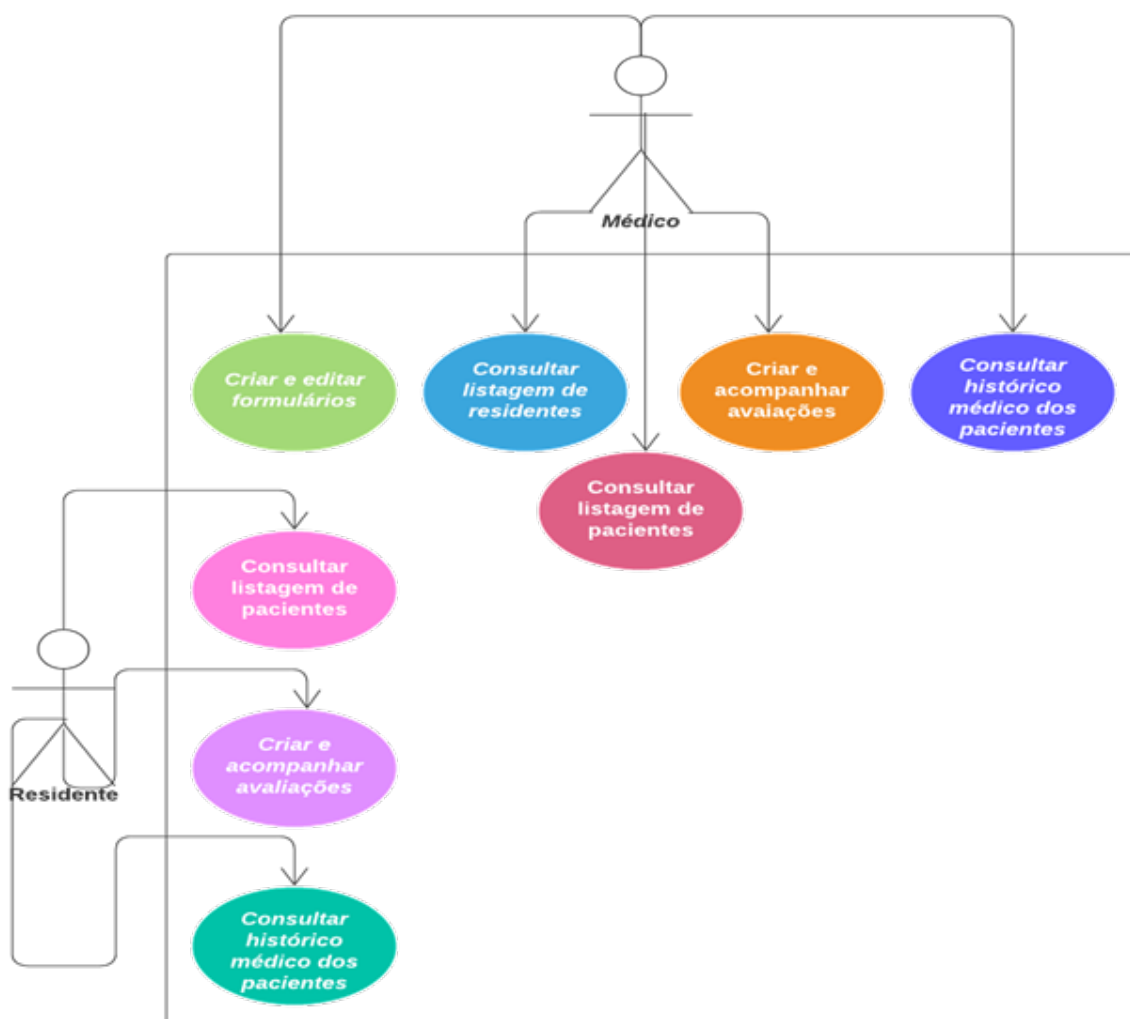


Figura 1. Diagrama de caso de uso

Fonte: Autoral, 2023

A Figura 1 acima, demonstra, assim, a interação dos usuários (atores), representados por figuras humanas, com o sistema, evidenciam as ações dos atores no contexto do diagrama de caso de uso do aplicativo em questão.

Para fins de visualização melhor do funcionamento app, descrevendo um pouco a estrutura do sistema, assim como suas entidades, suas propriedades (atributos), mostrando também como as classes estão relacionadas umas com as outras, será apresentado logo abaixo (figura 2) o diagrama de classe do projeto. Dessa forma, como pode ser observado, resumidamente, existem relacionamentos entre usuário com hospital por meio da tabela UsuarioHospital, onde médicos e residentes terão seu cadastro concluído e *linkado* com dados hospitalares; já os pacientes a serem avaliados estão ligados ao hospital, assim como suas avaliações, com descrição e grupos, que representam os tipos de modelo de perguntas, bem como as opções de cadastro de respostas da avaliação.

(deletar), desempenhando todos um papel crucial de gerenciamento de todas as possíveis solicitações.

```

PerguntaController.java x MedicoResidenteController.java
32     @GetMapping("/{perguntaId}")
33     public ResponseEntity<Pergunta> buscar(@PathVariable Long perguntaId){
34         Pergunta pergunta = perguntaRepository.buscar(perguntaId);
35
36         if (pergunta != null) {
37             return ResponseEntity.ok(pergunta);
38         }
39
40         return ResponseEntity.notFound().build();
41     }
42
43     @PostMapping
44     public ResponseEntity<?> adicionar (@RequestBody Pergunta pergunta){
45         try {
46             pergunta = cadastroPergunta.salvar(pergunta);
47
48             return ResponseEntity.status(HttpStatus.CREATED).body(pergunta);
49         } catch (EntidadeNaoEncontradaException e){
50             return ResponseEntity.badRequest().body(e.getMessage());
51         }

```

Figura 3. Trecho de código

Fonte: Autoral, 2023.

5.3 Consumo da API com o Postman

Visto a necessidade de demonstrar os testes e validação acerca do bom funcionamento da *api*, foi utilizado o *Postman*, plataforma de *API* para desenvolvedores, de modo a conferir e confirmar as premissas já descritas anteriormente acerca de um melhor monitoramento, segurança na aplicação, além do trato adequado de situações de erro.

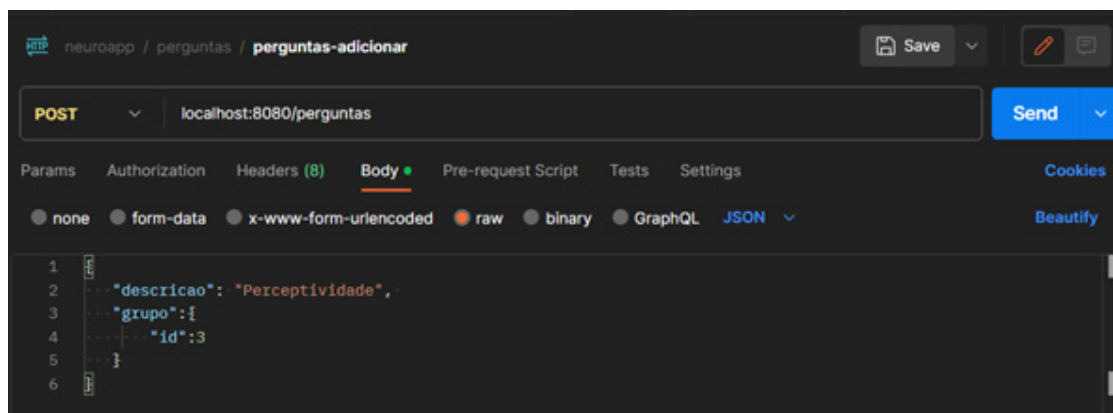


Figura 4. Método POST

Fonte: Autoral, 2023.

Como descrito na figura 4 acima, a utilização do método *POST*, responsável por adicionar valores aos campos, confirmando uma das funcionalidades, que foram expressas

no código da figura 3, trata-se de uma amostra da descrição e, a partir de uma *foreign key* (chave estrangeira), do grupo correspondente para a formulação dos questionários de avaliação do quadro clínico dos pacientes, que serão ilustrados e listados ainda neste tópico.

Por conseguinte, a amostragem de algumas telas para melhor visualização para o consumo da *API*, torna-se mais ilustrativo e didático, levando em consideração também a implementação, inclusive, de mais ideias que, pouco a pouco, estarão sendo implementadas.

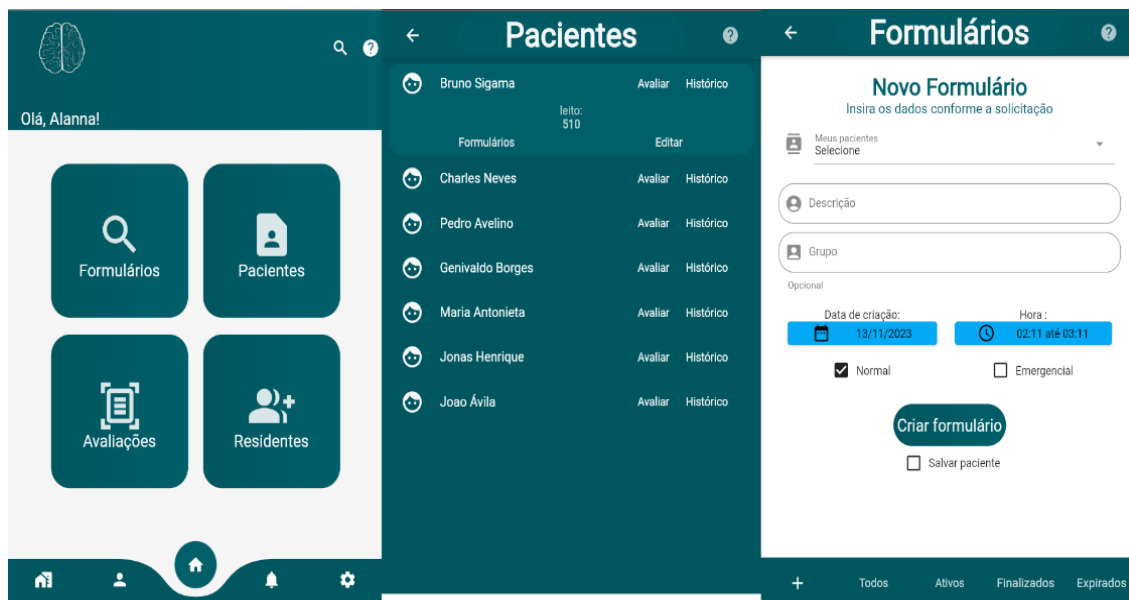


Figura 5. Telas de Home, Pacientes e Cadastro de Formulários

Fonte:Autoral, 2023.

Na figura 5, é demonstrado o desenvolvimento do esboço de telas em *Flutter*, primeiramente, a tela home do app, que se encaixa sob o perfil médico, onde terão acesso aos formulários cadastrados, listagem de pacientes e aba de avaliações dos seus quadros clínicos, bem como os residentes associados. Na sequência, todos os pacientes listados com a opção tanto de avaliação, quanto acesso aos históricos de cada paciente, para, por fim, a tela do cadastro de novos formulários, como descrito anteriormente.

Dando continuidade aos testes e validações, pode-se deixar registrado o retorno do *Postman* acerca do processamento de requisições, dentre elas, a listagem de todos os valores dos campos anteriormente descritos, a partir do trecho do código (figura 3), ainda com as opções de respostas, com o status 200 *OK*, sob o método *GET*, indicando que a solicitação foi processada com sucesso pelo servidor.

```

14 3,
15  {
16      "id": 2,
17      "descricao": "Comando Verbal",
18      "valor": 3,
19      "pergunta": {
20          "id": 1,
21          "descricao": "Nível de Consciência",
22          "grupo": {
23              "id": 1,
24              "nome": "Abertura Ocular"
25          }
26      }
27  },
28  {
29      "id": 3,
30      "descricao": "Orientado",
31      "valor": 5,
32      "pergunta": {
33          "id": 2,
34          "descricao": "Escala de coma de Glasgow",
35          "grupo": {
36              "id": 2,
37              "nome": "Resposta Verbal"
38          }
39      }

```

Figura 6. Método GET

Fonte: Autoral, 2023.

Dessa forma, como exemplificado acima, na figura 6, foram cadastrados alguns modelos de perguntas a serem utilizados na avaliação do quadro clínico de pacientes: por nível de consciência, aliados ao grupo de abertura ocular, que terão a possibilidade de adesão de registro de opções, como, por exemplo, espontânea ou comando verbal; por escala de coma de *Glasgow*, com a resposta verbal, podendo ser orientado, confuso ou desorientado; por resposta motora, ao comando verbal e entre outras possibilidades de formulação, para, por fim, dar seguimento para os resultados e discussões.

5.4 Validação do App

Neste tópico, a fim de discutir acerca da aplicação desenvolvida, foi utilizado um questionário de 4 perguntas com uma amostra de 35 pessoas, sendo 5 médicos e 30 residentes, de modo a investigar acerca das necessidades desses usuários, bem como fazer um levantamento das demandas que foram atendidas pela *app* (desenvolvido através do *framework flutter*) para a rotina de uma UTI, inclusive utiliza-se do *feedback* para a implementação de futuras melhorias.

A aplicação do questionário foi implementada após todos os testes de verificação e validação de todas as funcionalidades almejadas e, a partir do protótipo funcional, visou-se manter na elaboração de questões de teor discursivas para a coleta de dados, evidenciando, para fins mais didáticos, a seleção das respostas, que melhor pontuavam a credibilidade do projeto, mas também as ideias de implementação para futuras melhorias.

5.4.1 Entrevistas

Pergunta 1: Quais seriam as expectativas e/ou necessidades de uma aplicação direcionada para a área neurológica?

Médico 1: Um ambiente intuitivo e acompanhamento detalhado dos pacientes, possibilitando o registro eficiente de dados clínicos relevantes.

Médico 2: Registro preciso e detalhado das avaliações de quadro clínico, além de proporcionar uma visão abrangente do histórico médico.

Residente 1: Otimizar o fluxo de trabalho e melhorar a qualidade do atendimento

Residente 2: Praticidade e facilidade de uso, corroborando para um fluxo de trabalho mais ágil.

Residente 3: Interconexão entre os profissionais e fornecer notificações em tempo real para situações críticas.

Pergunta 2: Acredita que a premissa a qual o app se refere, consegue atender a demanda da rotina contínua em UTI?

Médico 1: A ideia de centralizar dados clínicos, históricos e avaliações de pacientes pode, de fato, atender a essa demanda.

Médico 2: Sim, principalmente pela possibilidade de cadastro e atualização de novos formulários.

Residente 1: Sim, ainda que seria bom nós termos acesso também em relação aos médicos associados.

Residente 2: A capacidade e o dinamismo de registrar e acessar rapidamente dados clínicos, atendem sim,

Residente 3: Sim, mesmo que seria de suma relevância a capacidade de receber notificações em tempo real.

Pergunta 3: Você considera a interface do app intuitiva e de fácil acesso?

Médico 1: Sim, é vital, o app se apresenta amigável e com uma navegação rápida.

Médico 2: Acredito que sim, a disposição dos dados e funcionalidades são boas.

Residente 1: Sim, de fácil interação e entendimento.

Residente 2: Sim, bem direta e com as principais funcionalidades.

Residente 3: Particularmente gostei do design, e, sim, é intuitiva e de fácil acesso.

Pergunta 4: Quais seriam as melhorias que poderiam estar sendo implementadas?

Médico 1: Creio que ferramentas de relatórios e personalizados.

Médico 2: Se estender, posteriormente, a uma gestão maior que abrange resultados de exames.

Residente 1: O gerenciamento de medicamentos seria interessante para um melhor controle de saúde do paciente na UTI.

Residente 2: Customização da interface, de acordo com as preferências individuais dos profissionais.

Residente 3: Notificações em tempo real de mudanças no estado clínico dos pacientes.



Sendo assim, foram exemplificados os resultados obtidos após o desenvolvimento da API, focalizando na avaliação da experiência de cada um dos usuários. Através da análise de uma amostra representativa de 35 participantes, composta por médicos e residentes, foi possível identificar padrões de satisfação e percepções em relação ao aplicativo, assim como sua porcentagem de avaliação de 85,71%, acumulando 30 respostas positivas.

Observou-se um *feedback* expressivamente positivo, indicando uma aceitação robusta e uma apreciação significativa das funcionalidades oferecidas. Os usuários, tanto médicos quanto residentes, destacaram a interface intuitiva, a facilidade de acesso e a utilidade geral do aplicativo. As respostas coletadas revelam uma aceitação do aplicativo como uma ferramenta valiosa para a gestão de informações clínicas na área neurológica. Essa recepção positiva sugere que o aplicativo atende às expectativas e necessidades identificadas, proporcionando um suporte eficaz para profissionais de saúde no ambiente dinâmico da neurologia, por fim, principalmente para a implementação de trabalhos futuros.

6. CONCLUSÃO

Em meio à complexidade do cenário hospitalar, este projeto emergiu como uma solução inovadora e essencial para a gestão eficiente de informações médicas. A aplicação, desenvolvida com base em boas práticas de programação orientada a objetos e empregando o robusto *framework Spring Boot*, *api* com arquitetura *REST*, aliados com o desenvolvimento de telas com o *framework Flutter*, destaca-se não apenas pela sua simplicidade e funcionalidade intrínseca, também pelos diferenciais estratégicos que o colocam à frente na área de gestão clínica, como foram comprovados, seja pelo gerenciamento de pacientes, formulários, avaliações, bem como a antecipação das necessidades emergentes da prática clínica moderna.

Este projeto obteve como resultados uma alta taxa de satisfação por parte dos entrevistados, propondo uma maior praticidade de controle de pacientes e das suas fichas técnicas de diagnóstico e estado clínico, por meio de formulários digitais que são criados e gerenciados pelos profissionais da área neurológica, podendo ser considerado um projeto promissor dentro da área da saúde.

Tendo em vista a solução apresentada, pode ser considerado como uma implementação adequada do projeto, tomando como base o feedback dos usuários, a inclusão de funcionalidades, como notificações e alertas em tempo real, seleção de unidade hospitalar para profissionais da saúde que trabalham em várias localidades, o gerenciamento de medicamentos e o acesso a resultados de exames diretamente no *app*. Por fim, essas melhorias já planejadas são reflexo do comprometimento contínuo em proporcionar uma solução completa e abrangente para os desafios enfrentados na gestão hospitalar, incluindo o empenho quanto ao seu desenvolvimento.

REFERÊNCIAS

- ALBERTO, Matheus. **Flutter: o que é e tudo sobre o framework**, 2023. Disponível em: < <https://www.alura.com.br/artigos/flutter> > Acessado em 12/11/2023
- BARRO, Bruna B., **Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem**, 2023. Disponível em: < <https://www.hostinger.com.br/tutoriais/linguagem-dart> > Acessado em 12/11/23
- BECK, K. **Programação Extrema (XP) Explicada: Acolha as Mudanças**. Porto Alegre: Bookman, 2004. x

- CARVALHO, M.; MACEDO, P. **Metodologias ágeis engenharia de software sob medida**. 1.ed. São Paulo: Érica, 2012 x
- CONCEIÇÃO, Pamela Silva. Desenvolvimento de um Aplicativo Para a Avaliação de Pacientes Neurológicos na UTI Utilizando o Framework Flutter. São Luís, Brasil, 2022.
- GAMMA, Erich et al. **Padrões de Projetos: soluções reutilizáveis de softwares orientados a objetos**. Porto Alegre: Bookman, 2007.
- GUEDES, Gilleanes. **UML 2: Uma Abordagem Prática**. 2018, Novatec Editora, 3ª Edição, capítulo 1.
- GUEDES, Marylene. **O que é uma API?** Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-uma-api?gclid=Cj0KCQjwltGUBhDXARIsAIJx01nvW7efzh_LASFQI-vzQYal-IoTemjz-Btw-THFVctFM0q2-YEJqcXkaAgZnEALw_wcB>. Acessado em: 10/11/2023
- IBM, Documentação. **Diagramas de Caso de Uso**, IBM, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>. Acesso em 10/11/2023
- IBM, Documentação. **Diagramas de Classes**, IBM, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em 12/11/2023
- KRIGER, Bruno; FRONT END: entenda o que é, para que serve, como aprender essa especialidade!, 2023. Disponível em: < <https://kenzie.com.br/blog/front-end/> > Acessado em 12/11/2023
- LEANDRO. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Devmidia, 2012. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>.
- LIMA, Guilherme. **REST: Conceito e fundamentos**. Alura, 2020. Disponível em <https://www.alura.com.br/artigos/rest-conceito-e-fundamentos>. Acesso em: 11/11/2023
- RIBEIRO, H.; SERAPIÃO, T. **Padrões de projeto: design patterns**. TCC (ciência da computação) - Faculdade de Ciência da Computação da Universidade do Vale do Paraíba, Jacareí, 2004.
- ROCHA, Ana Carolyn da Silva et al. **Importância da Avaliação Neurológica no Paciente Crítico: um Relato de Experiência**. In: II CONGRESSO NORDESTINO DE ENFERMAGEM EM CUIDADOS INTENSIVOS - Rio Grande do Norte, Brasil, 2019. Disponível em: <<https://www.doity.com.br/anais/coneci2019/trabalho/95012>>. Acesso em: 09/11/2023
- SILVA, Allicia Sousa da. **Desenvolvimento de uma API REST utilizando Spring boot para um Front End desenvolvido em Flutter**. São Luís, Brasil, 2022.
- SOMMERVILLER, Ian. **Engenharia de software**. 9.ed. São Paulo: Person, 2011. X
- TECHTARGET, Editor. **Contributor Spring Framework**, Techtargget, 2021. Disponível em: <https://www.techtargget.com/searcharchitecture/definition/Spring-Framework>. Acesso em: 08/11/2023
- WILDT, Daniel et al. **eXtreme Programming: Práticas para o dia a dia no desenvolvimento ágil de software**. casa do código, 2015.
- ZUCHER, Vitor. **O que é padrão MVC? Entenda arquitetura de softwares!** Disponível em: < <https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc> > Acessado em: 01/11/ 2023.



3

▶ CRIAÇÃO DE UM APLICATIVO PARA CONDOMÍNIOS UTILIZANDO *FLUTTER* E *REST API* VOLTADOS PARA A UTILIZAÇÃO DO MORADOR E PORTEIRO

*Development of an application for condominiums using Flutter
and a Rest API designed for use by residents and gatekeepers*

Pedro Henrique Silva de Sousa Carneiro

Edilson Carlos Silva Lima

Leonardo de Castro Mesquita

D.O.I.: 10.29327/5391075.1-3



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

A segurança é o principal fator que leva as pessoas a escolherem morar dentro de condomínios fechados; no entanto, no Brasil, há inúmeros fatores de risco que fragilizam até mesmo esse sistema. Um desses fatores é a falta de praticidade e agilidade na entrada de convidados e prestadores de serviço. Portanto, foi planejado criar um aplicativo utilizando Flutter no front-end e desenvolver uma Rest API utilizando o Framework Spring para o back-end. Para atender a esse propósito, serão adotadas as metodologias de entrevista e estudo de caso. Essa aplicação será voltada para proporcionar ao porteiro a capacidade de analisar todos os convites feitos pelos moradores, proporcionando maior praticidade na autorização de pessoas para a entrada no condomínio e reservas dos espaços.

Palavras-chaves: Segurança, Rest API, flutter, back-end, front-end.

Abstract

The main factor driving people to choose living in gated communities is security; however, in Brazil, there are numerous risk factors that even undermine this system. One of these factors is the lack of convenience and speed in admitting guests and service providers. Therefore, it was planned to create an application using Flutter on the front-end and develop a Rest API using the Spring Framework for the back-end. To achieve this goal, the methodologies of interviews and case studies will be employed. This application will be focused on providing the gatekeeper with the ability to analyze all invitations made by residents, offering greater convenience in authorizing people to enter the condominium and to reserve spaces.

Keywords: Security, Rest API, flutter, back-end, front-end.

1. INTRODUÇÃO

A entrada segura e eficiente em condomínios é um aspecto crucial para a tranquilidade dos moradores e a proteção de suas propriedades. Condomínios estão constantemente buscando maneiras de aprimorar o controle de acesso, garantindo que apenas indivíduos autorizados entrem nas instalações. A necessidade de agilidade nesse processo é evidente, uma vez que moradores, visitantes e prestadores de serviços frequentemente enfrentam longas filas nos portões de entrada. Nesse contexto, a tecnologia desempenha um papel fundamental na melhoria da segurança e na otimização da entrada, tornando-a mais rápida e eficaz.

O desenvolvimento de um aplicativo para condomínios voltado para o uso do porteiro requer uma interface de usuário (UI) eficiente e amigável, conhecida como “front end”. O Flutter, uma estrutura de código aberto da Google, tem se destacado como uma escolha excepcional para o desenvolvimento de front end de aplicativos móveis. Sua capacidade de criar interfaces de usuário atraentes e responsivas para dispositivos Android e iOS em um único código-base tornou-o uma opção popular. Ao utilizar o Flutter, podemos garantir que o Tranca seja fácil de usar e proporcione uma experiência de entrada suave para os porteiros.

Por trás da interface do aplicativo do porteiro, a lógica de funcionamento é alimentada pela programação orientada a objetos (POO). A POO é uma metodologia de programação que se concentra na organização de dados e funcionalidades em objetos interconectados, facilitando a manutenção e a expansão do código. O Spring Java é um popular framework de desenvolvimento que oferece suporte à criação de aplicativos baseados em Java, tornando mais eficiente o desenvolvimento de sistemas robustos e escaláveis, permitindo o desenvolvimento de uma REST API (Interface de Programação de Aplicativos) permite que o aplicativo do porteiro se comunique de forma eficaz com outros sistemas e serviços, melhorando a interoperabilidade e a eficiência geral. Tendo em vista as informações anteriores, as metodologias empregadas foram de estudo de caso e entrevista.

Em resumo, o desenvolvimento de um aplicativo para condomínios voltado para o uso do porteiro é uma resposta direta à necessidade de melhorar a segurança e a agilidade na entrada de condomínios. Utilizando tecnologias como o Flutter para o front end e o Spring Java com uma REST API para o back end, é possível criar um sistema que atenda às demandas dos moradores e facilite o trabalho dos porteiros. Essa abordagem inovadora não apenas agiliza a entrada, mas também aumenta a segurança e a eficiência operacional, proporcionando uma solução moderna para os desafios enfrentados pelos condomínios nos dias de hoje.

2. TRABALHOS RELACIONADOS

Neste tópico serão abordados trabalhos que por meio do método de análise bibliográfica foram utilizados como base para o desenvolvimento deste artigo, visando inserir novas soluções que sejam compatíveis com os seguintes trabalhos.

2.1 Análise da utilização do framework Spring em sistema web para o aplicativo de gerenciamento de condomínio.

Este projeto teve como objetivo o desenvolvimento de uma API REST, garantindo relacionamentos de tabelas no banco de dados que são bem estruturadas para o controle de condomínios, fazendo com que sejam cadastrados usuários, condomínios e convites, na conclusão do trabalho foi apresentadas aberturas para trabalhos futuros. Por exemplo: criação de contatos para visitantes frequentes, e reserva de espaços do condomínio.

2.2 O uso de *Clean Architecture* com *Monorepo* para o desenvolvimento de um aplicativo no Flutter consumindo serviços de uma API REST

Este artigo foi o desenvolvimento do front-end de um aplicativo de controle de condomínio desenvolvido em flutter que foi criado telas de criação de convites, convites criados e uma home que mostra várias funcionalidades futuras, sendo elas convites e também a reserva de espaços, este front preparado para o consumo de REST API, que deixou abertura para criação de contatos frequentes e reserva de espaços no condomínio.

2.3 Segurança em condomínio: a importância do controle de acesso

Este artigo foi desenvolvido na intenção de alertar a relevância no controle de acesso de pessoas em condomínio, destacando a importância da fiscalização pessoal, ou seja, sendo feita por alguma pessoa capacitada e treinada na área de segurança para criar uma tranquilidade para os moradores dos condomínios, e frisou que somente o uso de tecnologias, não é suficiente para garantia dessa tranquilidade.

2.4 Diferencial do projeto desenvolvido

Analisando os trabalhos apresentados acima, foram incrementadas ao código da API a funcionalidade de reservar espaços do condomínio, assim como a criação de contatos frequentes para o morador, e para controlar a entrada de convidados e prestadores de serviços, foi desenvolvido o app para manuseio do porteiro, que somente ele possa confirmar a entrada ao condomínio.

3. FUNDAMENTAÇÃO TEÓRICA

Neste tópico serão abordados assuntos que foram tomados como base para a produção do projeto, estes foram adquiridos a partir de pesquisas, além disso também durante o curso de graduação. O objetivo é identificar e aprofundar sobre o que cada assunto trata, e mostrar o porquê de serem aplicáveis nas soluções do projeto.

3.1 Metodologia ágil

Segundo Losnak (2023), A metodologia ágil é uma forma de gerir projetos, em que se busca a otimização dos processos. Em vez de seguir um plano rígido — como na gestão tradicional —, na metodologia ágil, podemos fazer ajustes e melhorias ao longo do desen-

volvimento de projetos. Isso significa identificar e corrigir falhas rapidamente, garantindo resultados mais efetivos. Portanto, essa forma de gestão permite às pessoas gerentes uma abordagem moderna e eficiente para enfrentar desafios do mercado.

3.1.1 Scrum

O Scrum é um framework ágil que visa facilitar a colaboração, a adaptação e a entrega de valor de forma iterativa e incremental. Ele se baseia em três pilares fundamentais: transparência, inspeção e adaptação. O Scrum divide o trabalho em iterações chamadas de Sprints, que geralmente têm duração de duas a quatro semanas e utiliza artefatos como o Product Backlog e o Sprint Backlog, além de cerimônias como as reuniões de planejamento, revisão e retrospectiva (Losnak, 2023).

3.2 Programação orientada ao objeto

Segundo Gillis (2021), a programação orientada a objetos (POO) é um modelo de programação de computador que organiza o design de software em torno de dados ou objetos, em vez de funções e lógica. Um objeto pode ser definido como um campo de dados que possui atributos e comportamento exclusivos.

OOP se concentra nos objetos que os desenvolvedores desejam manipular, e não na lógica necessária para manipulá-los. Esta abordagem de programação é adequada para programas grandes, complexos e ativamente atualizados ou mantidos. Isto inclui programas de fabricação e design, bem como aplicativos móveis; por exemplo, OOP pode ser usado para software de simulação de sistema de fabricação.

A organização de um programa orientado a objetos também torna o método benéfico para o desenvolvimento colaborativo, onde os projetos são divididos em grupos. Os benefícios adicionais da OOP incluem reutilização de código, escalabilidade e eficiência.

- **Encapsulamento:** Encapsulamento refere-se à construção de um objeto de maneira a resguardar o acesso direto aos seus dados internos. Ao encapsular um objeto, agrupamos propriedades e métodos intimamente relacionados dentro do mesmo objeto, possibilitando o acesso a essas propriedades exclusivamente por meio de métodos públicos. Dessa maneira, abordamos questões cruciais, tais como a segurança e a confiabilidade do estado do objeto (Toledo, 2019).
- **Abstração:** Abstração implica em “ocultar” partes da implementação do objeto, revelando apenas uma interface simplificada para sua utilização. Considere, por exemplo, um forno de micro-ondas. Você não precisa compreender toda a complexidade do funcionamento interno, responsável por gerar ondas e produzir calor. Seu desejo é simplesmente pressionar um ou dois botões para obter uma refeição quente. Toda essa complexidade representa detalhes de implementação, os quais não é necessário conhecer (Toledo, 2019).
- **Herança:** A Herança é um método para evitar a repetição de código no qual, conforme o próprio nome indica, um objeto pode herdar características (ou seja, propriedades e métodos) de outra classe, sem a necessidade de reescrever essas mesmas características (Toledo, 2019).
- **Polimorfismo:** Poli significa muitas e Morphos significa forma, então Polimorfismo significa muitas formas. No paradigma de Programação Orientada a Objetos

(POO), o Polimorfismo ocorre quando duas ou mais classes possuem métodos com o mesmo nome, mas podem ter implementações diferentes. Dessa forma, torna-se viável utilizar qualquer objeto que implemente o mesmo método sem a necessidade de preocupação com o tipo do objeto (Toledo, 2019).

3.3 UML

Segundo Grady Booch, James Rumbaugh e Ivar Jacobson, A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.

3.3.1 Diagrama de Caso de Uso:

Na UML, os diagramas de caso de uso modelam o comportamento de um sistema e ajudam a capturar os requisitos do sistema. Os diagramas de caso de uso descrevem funções de alto nível e escopo de um sistema. Esses diagramas também identificam as interações entre o sistema e seus agentes. Os casos de uso e os agentes nos diagramas de caso de uso descrevem o que o sistema faz e como os agentes o usam, mas não como o sistema opera internamente. (IBM, 2021).

3.3.2 Diagrama de Classes

Em UML, diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema. Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classe para modelar um sistema inteiro ou vários diagramas de classe para modelar os componentes de um sistema (IBM, 2021).

3.4 Back end

Segundo Ugo Roveda 2021 back-end é tudo aquilo que está por trás da interface de uma aplicação: seus sistemas, banco de dados, toda parte de segurança de dados, envio e recebimento de informações, armazenamento etc.

O back-end serve para que os sites, aplicativos, softwares ou outros tipos de sistemas de informação tenham todas as suas funcionalidades operando de maneira efetiva e cumprindo seus objetivos.

O back-end também serve para dar suporte aos projetos de tecnologia, unindo uma equipe multidisciplinar com uma visão mais abrangente do que está por trás de todos os processos.

3.4.1 API (*Application Programming Interface*)

Segundo Andrade (2016) uma API - *Application Programming Interface*, ou interface de programação de aplicativos - é o que expõe as funcionalidades de um sistema. Ela é uma interface que permite a troca de informações entre os sistemas, independentemente de suas tecnologias ou detalhes das implementações.

Uma aplicação envia uma requisição à API, informando o que ela deseja fazer. Pode ser uma consulta aos dados do sistema, um cadastro de uma nova informação, uma atualização de dados etc. Por exemplo, em uma API de cadastramento de usuários, poderão ser solicitadas a criação de um novo usuário, uma lista de todos os usuários do sistema, as informações de um usuário etc. A partir daí, a API irá retornar uma resposta à aplicação, indicando se o processamento da requisição foi feito com sucesso ou não, e retornando quaisquer dados relevantes a ela.

3.5 Spring Framework

Segundo Cavalcante (2020), O Spring Framework é um framework desenvolvido para a plataforma Java, fundamentado nos padrões de projeto (Design Patterns), inversão de controle e injeção de dependência. Ele é composto por vários módulos abrangentes, capazes de impulsionar significativamente a aplicação Java.

3.5.1 Injeção de dependências

A injeção de dependência visa evitar o acoplamento de código em uma aplicação. Em termos simples, consiste na provisão de instâncias de classes que um objeto necessita, sem que esse objeto as instancie por si mesmo. Podemos afirmar que a injeção de dependência representa uma maneira de implementar a inversão de controle (Cavalcante, 2020).

3.5.2 Rest architecture

Segundo Lima (2020), REST (Representational State Transfer) é um modelo de arquitetura, não uma linguagem de programação ou tecnologia, que fornece orientação para sistemas distribuídos se comunicarem diretamente usando princípios e protocolos da Web existentes sem a necessidade de SOAP ou outro protocolo complexo

REST requer que o cliente faça uma solicitação ao servidor para enviar ou modificar dados. A solicitação inclui: um verbo ou método HTTP que define o que o servidor fará, um header (cabeçalho) com cabeçalhos de solicitação que transmitem informações sobre a solicitação, Informações no corpo da solicitação, essas informações são opcionais (Lima, 2020).

Na arquitetura REST tem os métodos que são cinco: O método GET é o método mais comumente usados e geralmente é usado para solicitar ao servidor o envio de recursos, o método POST foi projetado para enviar dados de entrada para o servidor. Na prática, é frequentemente usado para dar suporte a formulários HTML, o método PUT edita e atualiza o documento no servidor e método DELETE, como o nome sugere, exclui um dado ou uma coleção do servidor (Lima, 2020).

Cada resposta retornada por um aplicativo REST envia um código que define o status

da solicitação. Por exemplo: 200 (OK), a solicitação foi concluída com sucesso, 201 (CREATED), o objeto ou recurso foi criado com sucesso, 204 (SEM CONTEÚDO), o objeto ou recurso foi excluído com sucesso, 400 (BAD REQUEST), ocorreu um erro na solicitação (podem ser vários os motivos), 404 (Não Encontrado), rota ou coleção não encontrada, 500 (Erro de servidor interno), ocorreu algum erro de servidor (Lima, 2020).

3.6 ORM

Segundo Kriger (2023), ORM (Object Relational Mapping, ou Mapeamento Objeto Relacional em português) é uma técnica de programação que visa facilitar a comunicação entre bancos de dados relacionais e linguagens de programação orientadas a objetos, como Java, Python e JavaScript.

Ela consiste em um conjunto de ferramentas que automatizam a persistência de dados em um banco de dados relacional, fazendo a conversão de dados de objetos para tabelas.

3.6.1 Hibernate

Segundo Kriger (2023), O Hibernate destaca-se como uma das bibliotecas de Mapeamento Objeto-Relacional (ORM) mais amplamente utilizadas para Java. Possuindo suporte para uma vasta gama de bancos de dados relacionais, ele capacita os desenvolvedores a manipular dados de forma mais intuitiva, utilizando objetos Java em vez de recorrer diretamente à escrita de consultas SQL.

Além dessa funcionalidade fundamental, o Hibernate oferece recursos avançados que enriquecem o desenvolvimento de aplicações. Isso inclui a capacidade de realizar mapeamento de herança, suporte eficiente a cache de segundo nível e a habilidade de criar consultas dinâmicas, proporcionando flexibilidade e eficiência no acesso e manipulação de dados.

3.7 LOMBOCK

Segundo Silva (2021) o Lombok é um framework para Java que permite escrever código eliminando a verbosidade, o que permite ganhar tempo de desenvolvimento para o que realmente é importante. Seu uso permite gerar em tempo de compilação os métodos getters e setters, métodos construtores, padrão builder e muito mais.

3.7.1 Anotações importantes do Lombok

@Data: A @Data anotação é provavelmente a anotação usada com mais frequência no conjunto de ferramentas do Project Lombok. Ele combina a funcionalidade de @ToString, @EqualsAndHashCode, @Getter e @Setter.

Essencialmente, usar @Data em uma classe é o mesmo que anotar a classe com um padrão @ToString @EqualsAndHashCode, bem como anotar cada campo com ambos @Getter e @Setter. Anotar uma classe com @Data também aciona a geração do construtor do Lombok. Isso adiciona um construtor público que usa qualquer campo @NonNull ou



final como parâmetro. Isso fornece tudo o que é necessário para um Plain Old Java Object (POJO) (Kimberlin, 2010).

@EqualsAndHashCode: Esta anotação em nível de classe fará com que o Lombok gere métodos equals e hashCode, já que os dois estão ligados intrinsecamente pelo hashCode contrato.

Por padrão, qualquer campo da classe que não seja estático ou transitório será considerado por ambos os métodos. Assim como @ToString, o exclude parâmetro é fornecido para evitar que um campo seja incluído na lógica gerada. Também pode-se usar o of parâmetro para listar apenas os campos que devem ser considerados (Kimberlin, 2010).

3.8 Front end

Segundo Brunno Kriger (2023), o front-end é uma área da programação que se dedica a criar a parte visual e interativa de um site, aplicativo ou software. É o que o usuário vê e usa quando acessa uma plataforma digital

O front-end serve para que os usuários possam interagir com uma aplicação; em outras palavras, para construir a interface de utilização de um site, aplicativo ou software.

3.9 Dart

Segundo Barro (2023), Dart é uma linguagem de programação multi-paradigma utilizada para desenvolvimento de aplicações web, aplicações móveis e de desktop. A linguagem Dart permite que o código criado rode tanto no lado do cliente quanto no servidor.

3.10 Flutter

Segundo Matheus Alberto (2023), Flutter é um framework (ferramenta) de desenvolvimento com ênfase em multiplataforma para dispositivos móveis. Desenvolvido pela Google, tem ampla aceitação no mercado e, mais recentemente, incorporou a capacidade de criar aplicações para desktop (Linux, Windows e macOS).

Na prática, o Flutter é uma ferramenta que viabiliza o desenvolvimento simultâneo de aplicativos para diferentes plataformas - Android e iOS - com um único conjunto de códigos. Assim, seus principais benefícios incluem versatilidade, uma curva de aprendizado reduzida e maior agilidade.

Este framework apresenta vantagens significativas para as empresas, permitindo que um único desenvolvedor crie aplicativos para diferentes sistemas operacionais, ao contrário do modelo nativo, que demanda dois desenvolvedores para cada plataforma.

4. METODOLOGIAS

As metodologias adotadas para a realização desse projeto foram estudo de caso no qual, estudou-se quais as melhores soluções possíveis para a problematização apresentada e também utilizamos entrevistas para poder analisar os resultados do projeto como um todo, que no qual foram propostas perguntas para pessoas que utilizaram os apps que

5. RESULTADOS E DICUSSÃO

Neste tópico serão demonstrados na prática os tópicos abordados anteriormente na fundamentação teórica, como por exemplo o desenvolvimento da API, utilizando o *Lombok* e suas anotações.

5.1 Descrição do sistema

O sistema desenvolvido é uma continuação do aplicativo para gerência de condomínios o Tranca, que tem como objetivo de facilitar a criação de convites para entrada no condomínio (para moradores e prestadores de serviço), assim como também reserva de espaços (quadra, piscina, churrasqueira). Desenvolvido em Scrum ele foi montado tanto no front-end em flutter, quanto a API feita com o spring framework, sendo feito 2 apps diferentes para morador e para o porteiro/administração.

5.2 Scrum

No desenvolvimento do aplicativo Tranca, adotou-se a metodologia Scrum, uma abordagem ágil e iterativa. O Product Backlog, colaborativamente estruturado com o Product Owner, direcionou as prioridades de funcionalidades. Durante as Sprints, as equipes de front-end e back-end, responsáveis pelo desenvolvimento em Flutter e Spring Framework, respectivamente, colaboraram para atender às metas estabelecidas. A realização de reuniões regulares, como o Daily Scrum e Sprint Review, promoveu a sincronização eficaz entre as equipes, permitindo feedback contínuo e aprimoramentos iterativos ao longo do ciclo de desenvolvimento.

5.3 LPOO

A linguagem de programação orientada a objetos, foi utilizada dentro do projeto, no momento de desenvolver a API dentro das modelagens das classes de JAVA.

5.3.1 Encapsulamento e abstração

O encapsulamento pode ser visto nos modificadores de acesso a classes e objetos, por exemplo temos os modificadores `public` e `private` como pode ser observado na Figura 1.

```

18 public class Usuario {
19
20     @EqualsAndHashCode.Include
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     @Column(name = "tr01idusuario")
24     private Long id;
25
26     @Column(name = "tr01nome", nullable = false, length = 45)
27     private String nome;

```

Figura 1. Java Class “Usuario”.

Fonte: Autoral, 2023.



A abstração pode ser encontrada dentro da classe Java que abstrai informações para a modelagem do objeto.

5.4 UML

A UML foi utilizada no desenvolvimento lógico do projeto, no momento de desenvolvimento de diagrama de casos de uso e o diagrama de classes.

5.4.1 Diagrama de casos de uso:

O diagrama de casos de usos é utilizado para diagramar que tipos de ações cada tipo de usuário pode tomar em relação ao sistema proposto, para este projeto foi feita a diagramação conforme a Figura 2.

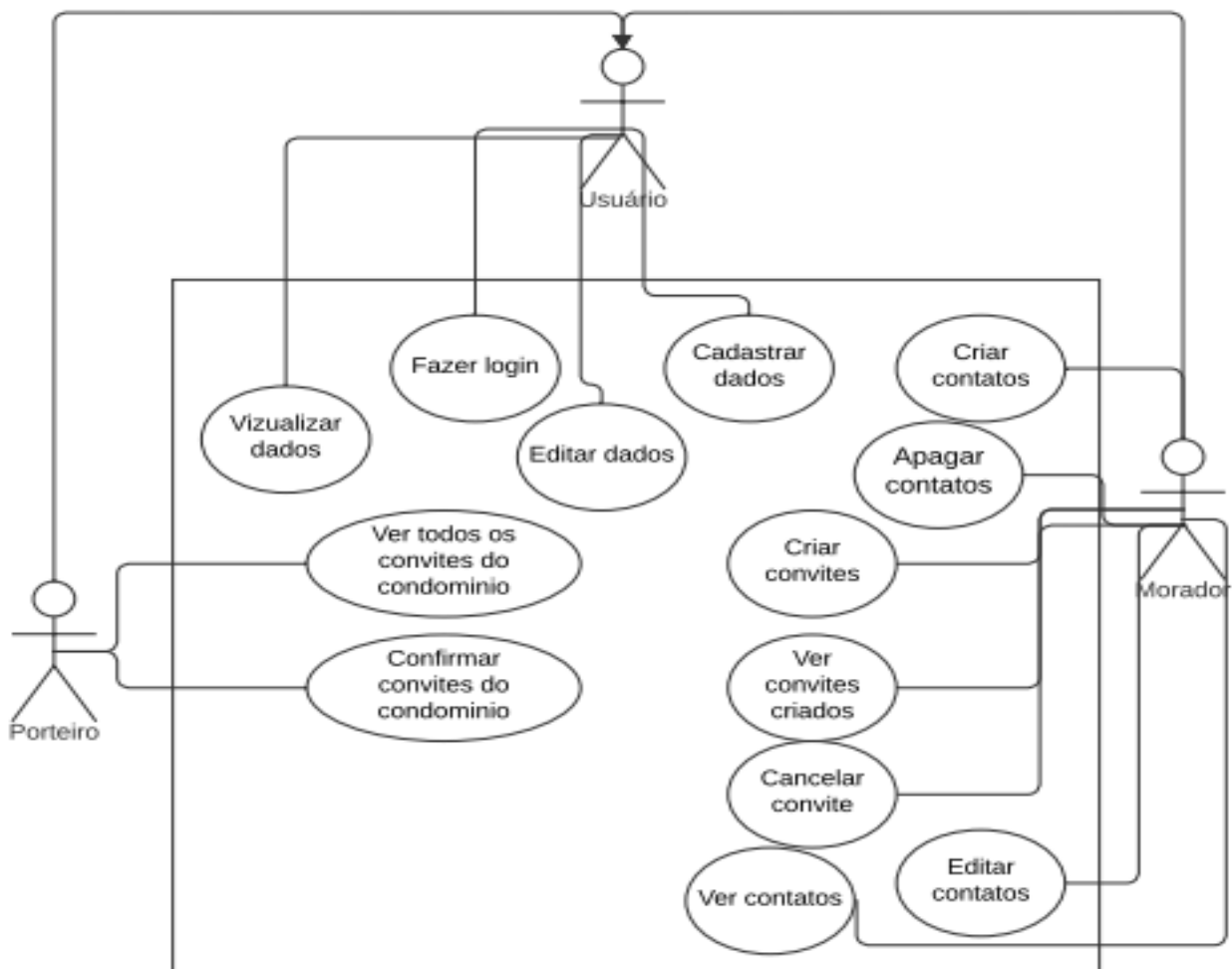


Figura 2. Diagrama de casos de uso do Tranca

Fonte: Autoral, 2023.

De acordo com a imagem acima pode ser visualizado cada ação que cada tipo de usuário é capaz de fazer dentro do condomínio, por exemplo somente o morador consegue adicionar contatos e criar convites e qualquer usuário consegue fazer login.

5.4.2 Diagrama de caso de usos

O diagrama de classes é feito para determinar os relacionamentos entre as classes presentes dentro da estrutura do código da api, para assim determinar a estrutura das tabelas do banco de dados do programa, este modelo pode ser visto exemplificado pela Figura 3.

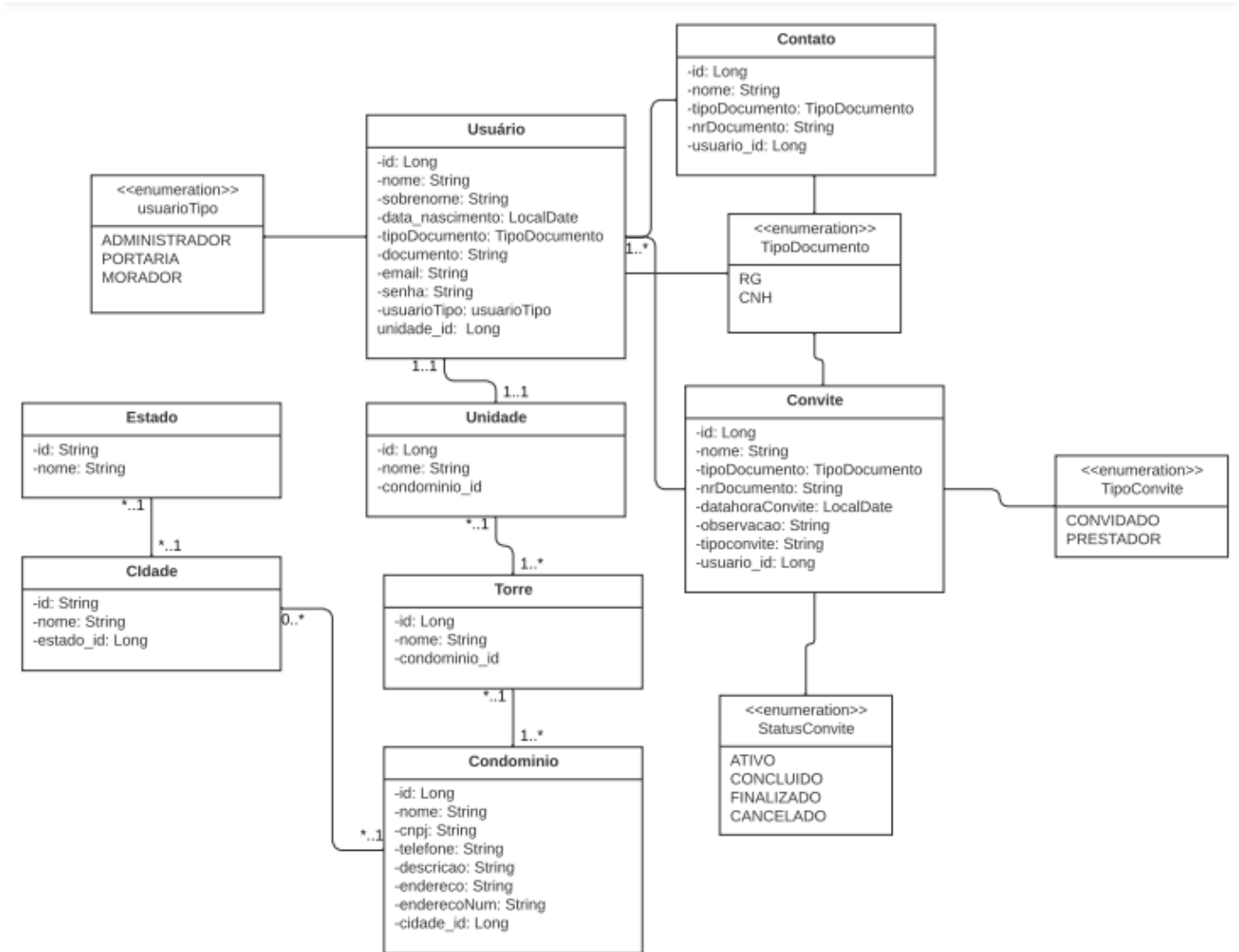


Figura 3. Diagrama de classe do Tranca

Fonte. Autoral, 2023

O diagrama acima mostra cada relacionamento entre essas classes, que são posteriormente, a base fundamental para a construção do banco de dados do aplicativo.

5.5 Implementação do Lombok

O Lombok entra como inovação nesse código com a anotação @Data que permite o desenvolvedor, com essa anotação um autoincremento dos *getters* e *setters* das classes e objetos da API, gerando assim muito mais facilidade e reduzindo a complexidade do desenvolvimento da API, como pode ser observado na Figura 4.

```

@JsonRootName("contato")
@Data
@Entity
@Table(name = "tr08contato")
public class Contato {

    @EqualsAndHashCode.Include
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "tr08idcontato")
    private Long id;

    @Column(name = "tr08nome", nullable = false, length = 60)
    private String nome;
}

```

Figura 4. Uso do @Data annotation

Fonte. Autoral, 2023

Dessa Maneira o @Data cria todos os métodos *getters* e *setters* de todos os atributos dentro dos objetos no código.

5.6 Controllers

Os *controllers* são responsáveis pelos *endpoints* do projeto, que são os métodos de adicionar, deletar, listar e atualizar, estes *endpoints* foram testados no POSTMAN, como pode ser visto na Figura 5.

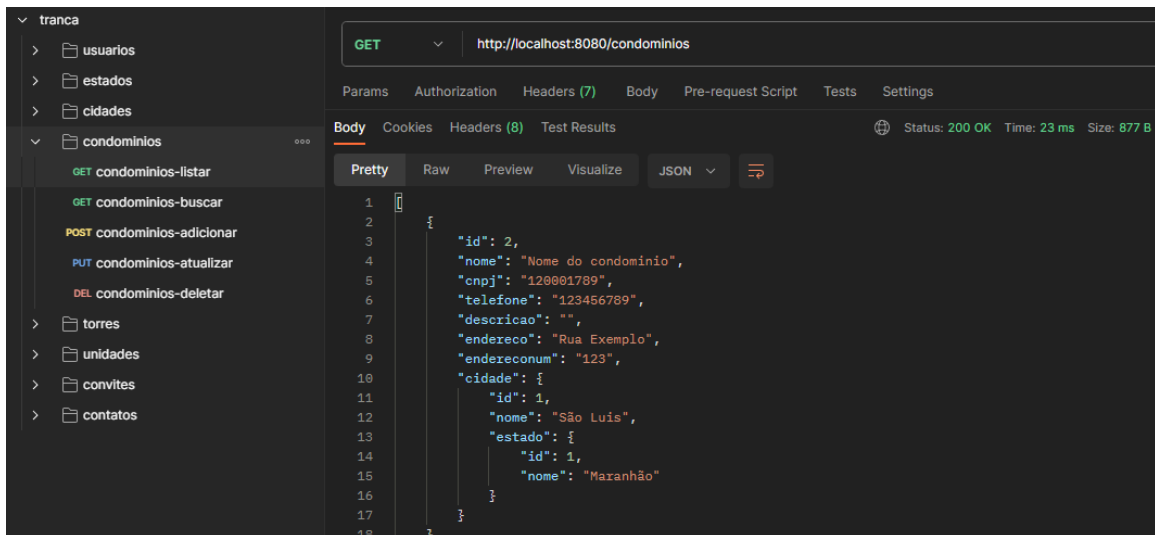


Figura 5. Listando condominios no POSTMAN

Fonte. Autoral, 2023

Na imagem acima é utilizado o método *get* para poder listar todos os condominios inseridos no banco de dados do Tranca.

5.7 Front-end

O desenvolvimento do front-end desse aplicativo foi feito de dois aplicativos, que são eles voltados, um para o morador e o outro para funções administrativas e para o porteiro. As figuras 6.a e 6.b mostram as interfaces de login e do porteiro respectivamente.



Figura 6. a. Tela de login; **b.** Tela inicial porteiro

Fonte. Autoral, 2023

A tela do porteiro serve para apresentar para ele, quais tipos de convites existem, são eles: ativos (foram feitos pelo morador e ainda não foram confirmados pelo porteiro), finalizados (São os que já foram confirmadas as entradas pelo porteiro), expirados (Os quais o porteiro não confirmou dentro do prazo de validade) e por fim os cancelados (que foram cancelados pelo morador que gerou o convite).

5.8 Entrevistas

Para avaliar a eficiência e praticidade do aplicativo foram feitas cinco perguntas sobre a usabilidade e efetividade do Tranca em uma amostra de 50 pessoas, dentre elas moradores e porteiros de condomínio.

5.8.1 Perguntas e respostas

1-Você considera que o processo de criação de convites no Tranca é intuitivo e fácil de usar?

R: 90% (45 pessoas) responderam Sim, 6% (3 pessoas) disseram Não e 4% (2 pessoas) não souberam responder.

2-O Tranca proporcionou maior praticidade e rapidez no momento de realizar reservas de espaços comuns no condomínio?

R: 84% (42 pessoas) responderam Sim, 8% (4 pessoas) disseram Não e 8% (4 pessoas) não souberam responder.

3-Você percebeu benefícios na utilização do recurso de contatos frequentes do Tranca para o controle de acesso?

R: 96% (48 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 2% (1 pessoa) não soube responder.

4-O Tranca agilizou a autorização de entrada de convidados e prestadores de serviço no seu condomínio?

R: 90% (45 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 8% (4 pessoas) não souberam responder.

5-A experiência geral com o Tranca contribuiu para uma interação mais eficiente entre moradores, porteiros e prestadores de serviço?

R: 94% (47 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 4% (2 pessoas) não souberam responder.

6. CONCLUSÃO

O aplicativo Tranca, desenvolvido com Flutter para o front-end e uma REST API baseada no framework Spring para o back-end, emerge como uma solução inovadora para aprimorar a segurança e praticidade em condomínios. Ao incorporar melhorias como a criação de contatos frequentes e a reserva de espaços, o Tranca vai além de simples agilidade na entrada, priorizando também a segurança e eficiência operacional, alinhando-se às demandas contemporâneas. A escolha estratégica de tecnologias como Flutter e Spring, respaldada por conceitos fundamentais como programação orientada a objetos e metodologias ágeis, sustenta a robustez e eficácia do projeto. Com interfaces intuitivas tanto para moradores quanto para o porteiro, o Tranca representa uma resposta moderna e eficiente para os desafios enfrentados pelos condomínios na gestão de acesso.

A fundamentação teórica, que abrange temas como UML, back-end, API, Spring Framework, ORM (Hibernate), e Lombok, fornece alicerces sólidos para a implementação prática do Tranca. Os resultados e discussões destacam a aplicação efetiva desses conceitos, desde a modelagem de classes até a implementação do Lombok para simplificar o código. Os diagramas de classe e caso de uso evidenciam a estrutura lógica do sistema, reforçando a abordagem ágil e eficiente adotada no projeto. Em resumo, o Tranca não apenas atingiu, mas ultrapassou suas metas, emergindo como uma solução tecnológica abrangente e contemporânea para os requisitos dinâmicos dos condomínios, amalgamando eficiência, segurança e praticidade.

REFERÊNCIAS

ALBERTO, Matheus, **Flutter: o que é e tudo sobre o framework**, 2023. Disponível em: < <https://www.alura.com.br/artigos/flutter> > Acessado em 12/11/2023

ANDRADE, V., I: **UM ESTUDO SOBRE PADRÕES E TECNOLOGIAS**

BARRO, Bruna B., **Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem**, 2023. Disponível em : <<https://www.hostinger.com.br/tutoriais/linguagem-dart> > Acessado em 12/11/23

BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML, Guia do Usuário: tradução**; Fábio Freitas da Silva, Rio de Janeiro, Campus ,2012.

CAVALCANTE, Paulo; **Spring Framework: o que é, seus módulos e exemplos!**; 2020; Disponível

em: < <https://blog.geekhunter.com.br/spring-framework/> > Acessado em: 12/11/2023.

CORREIA, Marcelo da Conceição, **O uso de Clean Architecture com Monorepo para o desenvolvimento de um aplicativo no Flutter consumindo serviços de uma API REST**, 2022.

DIAS, E. M., CARDOSO, F. T., GONSALVES, J. D. S., FERREIRA, A. L. P. ; **SEGURANÇA EM CONDOMÍNIO: A IMPORTÂNCIA DO CONTROLE DE ACESSO**, 2016. Disponível em: < <https://www.linkedin.com/pulse/seguran%C3%A7a-em-condom%C3%ADnio-import%C3%A2ncia-do-controle-de-padi-lha-ferreira/> > Acessado em 28/10/2023

FERREIRA, C.E.J., **Análise da utilização do framework Spring em sistema web para o aplicativo de gerenciamento de condomínio**, 2022.

GILLIS, Alexander S. **object-oriented programming (OOP)**, 2021; Disponível em: <<https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP> > acessado 12/11/2023.

IBM, Documentação. **Diagramas de Caso de Uso**, IBM, 2021. Disponível em: < <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case> > acessado em: 12/11/2023.

IBM, Documentação. **Diagramas de Caso de Uso**, IBM, 2021. Disponível em: < <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams> > Acessado em 12/11/2023

KIMBERLIN, M., **REDUCING BOILERPLATE CODE WITH PROJECT LOMBOK**, 2010. Disponível em < <https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok> > Acessado em 12/11/2023

KRIGER, Bruno; **FRONT END: ENTENDA O QUE É, PARA QUE SERVE, COMO APRENDER ESSA ESPECIALIDADE!**, 2023. Disponível em: < <https://kenzie.com.br/blog/front-end/> > Acessado em 12/11/2023

KRIGER, Bruno; **ORM (OBJECT RELATIONAL MAPPER) – SAIBA O QUE É E IMPORTÂNCIA NA PROGRAMAÇÃO**, 2023. Disponível em: < <https://kenzie.com.br/blog/orm/> > Acessado em 12/11/2023

LIMA, Guilherme, **REST: Conceito e fundamentos**, 2020. Disponível em: < <https://www.alura.com.br/artigos/rest-conceito-e-fundamentos> > Acessado em 12/11/2023

LOSNAK, Giulia, **Metodologia Ágil - O que é?**, 2023. Disponível em: < <https://www.alura.com.br/artigos/o-que-e-metodologia-agil> > Acessado em 12/11/2023

PARA O DESENVOLVIMENTO WEB – BACK-END, Rio de Janeiro – RJ, Brasil, Escola politécnica, UFRJ, 2016.

ROVEDA, Ugo, **O QUE É BACK END, PARA QUE SERVE E COMO APRENDER EM 2021**, 2021. Disponível em: < <https://kenzie.com.br/blog/back-end/> > acessado em 12/11/2023

SILVA, Alex, **Como usar o Lombok em projetos Java**, 2021. Disponível em: < <https://www.dio.me/articles/como-usar-o-lombok-em-projetos-java> > Acessado em 12/11/2023

TOLETO, Sergio. **Os 4 pilares da Programação Orientada a Objetos**, Sergiotoleto, 2019. Disponível em: <https://www.sergiotoledo.com.br/artigos/os-4-pilares-da-programacao-orientada-a-objetos>. Acessado em 10/11/2023



PLATAFORMA WEB DESENVOLVIDA EM Node.JS E REACT PARA AUXÍLIO NA APRENDIZAGEM DE CIÊNCIAS EXATAS

4

*Web platform developed in Node.JS and React to aid in learning
exact sciences*

Nikaelson Passos Pereira

Edilson Carlos Silva Lima

Yonara Costa Magalhães

D.O.I.: 10.29327/5391075.1-4



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

A educação é primordial e fundamental em qualquer país, ter uma rede educacional sólida é essencial para promover o engajamento dos alunos. Com foco na área de ciências exatas, este estudo apresenta o desenvolvimento de uma plataforma educacional para auxílio de alunos. O objetivo deste artigo é apresentar uma alternativa acessível e de fácil utilização que visa apoiar o processo de ensino e aprendizagem. Tecnologias como *Typescript*, *Node.JS*, *Prisma*, *React* foram utilizadas para o desenvolvimento da plataforma. Como metodologia, foi realizado um estudo de caso com a aplicação de um questionário de forma qualitativa e uma revisão de literatura sobre os conceitos apresentados neste trabalho. A plataforma foi amplamente aceita pelos alunos, que demonstraram satisfação em utilizá-la.

Palavras-chave: Aprendizagem, Typescript, Node.JS, Prisma, React.

Abstract

Education is paramount and fundamental in any country, having a solid educational network is essential to promote student engagement. Focusing on the area of exact sciences, this study presents the development of an educational platform to help students. The objective of this article is to present an accessible and easy-to-use alternative that aims to support the teaching and learning process. Technologies such as Typescript, Node.JS, Prisma, React were used to develop the platform. As a methodology, a case study was carried out with the application of a qualitative questionnaire and a literature review on the concepts presented in this work. The platform was widely accepted by students, who expressed satisfaction in using it.

Keywords: Learning, Typescript, Node, Prisma, React.

1. INTRODUÇÃO

A educação, indiscutivelmente, desempenha um papel primordial e basilar no desenvolvimento de qualquer nação, moldando mentes e preparando os cidadãos para os desafios da sociedade moderna. Porém, o sistema de educação brasileiro enfrenta muitos problemas que colocam em risco toda a formação de estudantes em todo Brasil, sobretudo nas áreas de ciências exatas e da natureza. Segundo dados do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), em 2019, cerca de 22% das escolas públicas brasileiras enfrentam problemas com a falta de professores em disciplinas específicas, como matemática, física, química e biologia (INEP, 2019).

Essa situação é refletida no baixo desempenho dos estudantes brasileiros em avaliações nacionais e internacionais. O Programa Internacional de Avaliação de Estudantes (PISA), que avalia o desempenho de estudantes de 15 anos em diversas áreas, indicou que o Brasil está abaixo da média em matemática e ciências. De acordo com os resultados do PISA de 2018, o Brasil ocupou a 57ª posição em matemática e a 63ª posição em ciências entre os 79 países avaliados (OCDE, 2018).

Neste sentido, a plataforma ‘Estude Melhor’ foi desenvolvida para mitigar o problema de acesso aos conteúdos de ciências exatas como: matemática, física, química e biologia. Também como auxiliar a professores, que podem usar a plataforma como complemento às suas aulas. Este trabalho constatou por meio de pesquisas em indexadores online que o desenvolvimento de plataformas gratuitas de aulas e questões é escasso no Brasil. A pesquisa nos indexadores usou-se palavras-chave como: “plataforma de educação”, “plataforma de aulas”, “plataforma web de ensino”.

Portanto, este estudo explora o desenvolvimento de uma plataforma educacional focada nas disciplinas de exatas, como citado acima. Foi desenvolvida com foco nos conteúdos do ensino médio, visando colaborar com professores e alunos. A dor que a plataforma busca resolver é a escassez de acesso aos conteúdos de qualidade nessas disciplinas, ajudando os alunos a contornar problemas como a falta de professores em escolas públicas, conteúdo raso, assim como a falta de flexibilidade para estudar. Neste sentido, a plataforma visa mitigar esses desafios, oferecendo aulas e questões de ciências exatas, utilizando métodos pedagógicos inovadores como o e-learning, dando flexibilidade para o aluno e autonomia para estudar mesmo fora da sala de aula. Ou seja, a plataforma usa tecnologia para melhorar o aprendizado e preparar os estudantes para os desafios do século XXI.

A plataforma apresenta aulas e questões, as aulas abordam conteúdos do ensino médio, por exemplo, na disciplina de biologia aborda desde conteúdos iniciais como: origem da vida, até mais avançados como: ecologia. Já as questões são disponibilizadas sobre o tema da aula assistida, estas são questões obtidas da internet, todas com enunciado contextualizado e com cinco asserções, podendo o aluno ver quais questões acertou. A metodologia envolveu pesquisa sobre metodologia XP, planejamento, modelagem UML, estudo de caso e desenvolvimento utilizando Node.js e React, e pesquisa qualitativa junto a alunos de um curso preparatório em São Luís, Maranhão, para avaliação da plataforma ‘Estude Melhor’.

Neste artigo, serão explorados temas valiosos. No capítulo 2, abordaremos a metodologia utilizada. A fundamentação teórica estará presente no capítulo 3, o estudo de caso da plataforma no capítulo 4, enquanto os resultados e discussões serão apresentados no capítulo 5. Por último, o capítulo 6 discorre sobre a conclusão.

2. METODOLOGIA

A metodologia é uma parte fundamental de qualquer trabalho. Uma metodologia sólida é a espinha dorsal de trabalhos acadêmicos, fornecendo a estrutura e o rigor necessários para embasar as conclusões. “Pode-se definir método como caminho para se chegar a determinado fim. E método científico como o conjunto de procedimentos intelectuais e técnicos adotados para se atingir o conhecimento” (Gil, 2008, p. 7).

Neste trabalho buscou-se construir uma plataforma estruturada e alinhada aos princípios educacionais, visando aprimorar o aprendizado dos estudantes em todo o Brasil. Isso é alcançado por meio da aplicação de etapas cuidadosamente planejadas, incluindo:

- Pesquisa sobre a metodologia XP (*Extreme Programming*) e planejamento;
- Planejamento da aplicação e modelagem UML: nesta etapa foram criados o diagrama de caso de uso e o diagrama de classes, a ferramenta utilizada foi o lucidchart.
- Desenvolvimento da aplicação: para o desenvolvimento do *back-end* foram usados o *Node.js* com *Typescript* e *Prisma* como *ORM*. Já no *front-end*, foram usados React e *Typescript*, além da biblioteca *Chakra UI* para estilização e responsividade.
- Pesquisa qualitativa sobre a plataforma: o público-alvo foi um grupo de cerca de cem alunos que frequentam um curso preparatório em São Luís, Maranhão. A pesquisa contou com uma amostra de 20 desses alunos. A pesquisa contou com perguntas acerca do problema e da contribuição da plataforma ‘Estude Melhor’.

3. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo vamos apresentar e conceituar os principais assuntos abordados neste trabalho. Neste sentido, o presente artigo busca estabelecer uma base sólida de conhecimento. Através de uma análise criteriosa, pretende-se situar o leitor no panorama teórico que embasa a pesquisa. Na busca por uma abordagem eficaz no desenvolvimento de software utilizamos a metodologia XP, item 3.1. Como boa prática de programação usamos o TDD, item 3.2. Em busca de uma modelagem visual, o item 3.3 aborda a UML. Abordamos ainda *Back-End* no item 3.4, *Front-End* no item 3.5 e *ORM* no item 3.6.

3.1 Metodologia XP (*Extreme Programming*)

A Metodologia de Programação Extrema (XP) é um enfoque ágil de desenvolvimento de software que enfatiza a colaboração constante entre desenvolvedores, testes frequentes e melhoria contínua. De acordo com Beck (2000), a XP se baseia em valores como comunicação, *feedback*, simplicidade e coragem. Essa abordagem é caracterizada pela escrita de testes automatizados antes da implementação do código, um processo conhecido como Desenvolvimento Orientado a Testes (TDD), promovendo a entrega incremental e frequente de funcionalidades. Além disso, a XP envolve o planejamento semanal, reuniões diárias curtas para alinhamento da equipe e refatoração constante do código para mantê-lo limpo e adaptável (Jeffries; Anderson; Hendrickson, 2000).

No desenvolvimento do projeto, a aplicação da Metodologia XP teve um impacto notável na eficiência e na qualidade do desenvolvimento. A prática de Desenvolvimento Orientado a Testes (TDD) permitiu que a equipe garantisse que as funcionalidades estivessem corretas e que os requisitos fossem atendidos. Além disso, a entrega incremental

de módulos foi facilitada pela abordagem ágil, permitindo uma melhor adaptação às mudanças de requisitos e necessidades. A refatoração constante, outro pilar da XP, contribuiu para manter o código base organizado e de fácil manutenção, permitindo ajustes contínuos e evolução do sistema. Dessa forma, a Metodologia XP não apenas otimizou o processo de desenvolvimento, mas também promoveu uma cultura de colaboração e melhoria constante.

3.2 TDD (Test-Driven Development)

O Desenvolvimento Orientado a Testes (TDD) é uma abordagem de desenvolvimento de software que enfatiza a criação de testes automatizados antes da implementação do código funcional. De acordo com Beck (2022), TDD segue um ciclo iterativo: escrever um teste automatizado que falhe inicialmente, implementar o código mínimo necessário para fazer o teste passar e, finalmente, refatorar o código para melhorar a estrutura. Essa prática promove a criação de software modular, testável e de alta qualidade. Como exemplificado por Jeffries, Anderson e Hendrickson (2000), o TDD não apenas ajuda a identificar problemas precocemente, mas também orienta o design do código, resultando em uma arquitetura mais limpa e manutenível. Como mantra o TDD apresenta codificar, testar e refatorar em um ciclo até apresentar uma alta qualidade, a figura 1 exemplifica esse ciclo.

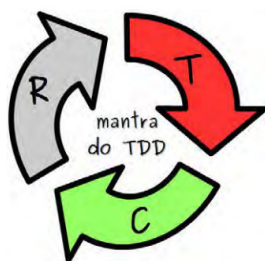


Figura 1. Mantra do TDD: testar, codificar e refatorar

Fonte: Wildt *et al.* (2015, p. 100)

O uso do TDD foi crucial na plataforma, proporcionando benefícios como garantia de funcionamento das partes do sistema, integração contínua e suporte ao desenvolvimento ágil da Metodologia XP. Além de validar a lógica do código, o TDD serviu como documentação viva, facilitando a compreensão das funcionalidades. Essa prática contribuiu para evitar regressões, melhorar a qualidade do código e manter a estabilidade do software ao adicionar novas funcionalidades.

3.3 UML (Linguagem de Modelagem Unificada)

A Linguagem de Modelagem Unificada (UML) desempenha um papel crucial na Engenharia de Software, oferecendo uma notação visual padronizada para modelar sistemas complexos. A UML trata-se de uma linguagem visual empregada na modelagem de softwares que seguem o paradigma de orientação a objetos. A UML é uma linguagem de modelagem de natureza genérica, viável para ser empregada em todos os âmbitos de aplicação (Guedes, 2011). Essa linguagem, proposta por Grady Booch, Ivar Jacobson e James Rumbaugh, transcende barreiras linguísticas e facilita a descrição e documentação de sistemas de forma abrangente.

3.3.1 Diagrama de Casos de Uso

O diagrama de caso de uso, proposto por Jacobson *et al.* (2006), é uma ferramenta poderosa para representar as interações entre atores e um sistema (Jacobson *et al.*, 2006). Tem como objetivo identificar os atores (utilizadores, sistemas externos ou até componentes de hardware específicos) que irão interagir de algum modo com o software, assim como as ações, isto é, as capacidades que o sistema oferecerá aos atores, referidas como cenários neste esquema (Guedes, 2011).

No contexto da plataforma, o diagrama foi crucial para identificar as necessidades dos usuários e orientar o desenvolvimento das funcionalidades. Ele mapeou os cenários de uso da plataforma, desde o cadastro de usuários até a resolução de exercícios. Através desse diagrama, a equipe pôde compreender como os diferentes atores interagiriam com a plataforma e quais seriam suas necessidades, direcionando o desenvolvimento para atender a esses requisitos com precisão.

3.3.2 Diagrama de Classes

Essa ferramenta essencial para a modelagem orientada a objetos permite visualizar de forma clara e organizada as abstrações do mundo real que compõem o sistema, auxiliando no entendimento das interações e hierarquias entre os elementos do software (Rumbaugh; Jacobson; Booch, 2005). No contexto da plataforma, esse diagrama foi fundamental para mapear a estrutura estática do sistema. Ele proporcionou uma visão clara das entidades centrais, como módulos, aulas e exercícios, e suas interações. Isso influenciou diretamente o desenho da arquitetura, permitindo que os desenvolvedores entendessem as entidades fundamentais e suas responsabilidades, resultando em um sistema bem-organizado e compreensível.

3.4 Back-End

O *Back-End* é a parte central de um sistema de software, responsável por gerenciar a lógica de negócios, o armazenamento de dados e a comunicação com o Front-End (Presman, 2016). Ele atua como uma ponte entre o usuário e o banco de dados, processando solicitações, manipulando dados e garantindo a integridade do sistema. Por meio de tecnologias robustas e linguagens eficientes, o *Back-End* sustenta a funcionalidade e a confiabilidade de aplicações complexas, desempenhando um papel crítico no desenvolvimento de software. Para o contexto da plataforma o *Back-End* usou ferramentas importantíssimas para a qualidade da aplicação, como *Node.JS* e *Typescript*.

3.4.1 API (Application Programming Interface)

Uma Interface de Programação de Aplicações (API) é um conjunto de regras e protocolos que permitem a comunicação entre diferentes softwares. Ela define as operações e os formatos de dados que os aplicativos podem utilizar para se interconectarem de maneira eficaz (Sommerville, 2011). As APIs desempenham um papel vital na construção de sistemas escaláveis e modulares, permitindo que diferentes partes do software interajam de forma independente e coordenada. Elas possibilitam a reutilização de funcionalidades, a integração de serviços de terceiros e a construção de interfaces de usuário coesas. Como

mostra a figura 2, diferentes clientes podem fazer o consumo da API e se basearem neste modelo.

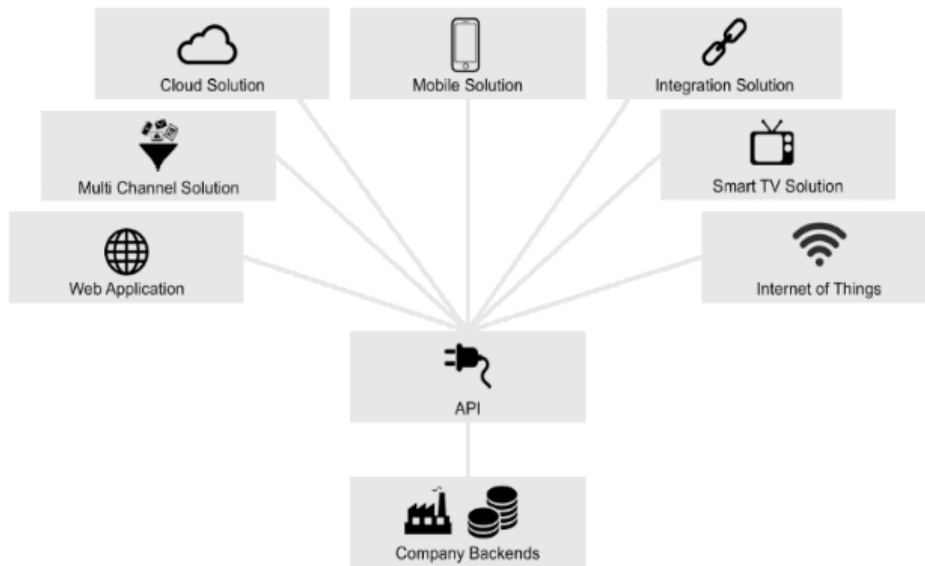


Figura 2. Soluções baseadas em API

Fonte: Biehl (2016, p. 22)

Esse modelo apresenta maior facilidade de integração e escalabilidade e por isso é amplamente usado. Devido a esses fatores ele foi escolhido para o desenvolvimento da plataforma.

3.4.2 Node.JS

O Node.js, desenvolvido por Ryan Dahl, revolucionou o desenvolvimento *Back-End* ao introduzir a capacidade de executar código *JavaScript* no servidor (Dahl, 2009). Com base em um mecanismo de I/O assíncrono e não bloqueante, o *Node.js* permite que aplicações lidem com múltiplas solicitações de forma eficiente e escalável. Sua adoção no desenvolvimento *Back-End* proporciona um ambiente coeso para construir aplicações em tempo real, APIs robustas e servidores escaláveis, estendendo a versatilidade do *JavaScript* para o lado do servidor.

3.4.3 Typescript

O *TypeScript*, uma linguagem de programação desenvolvida pela Microsoft, atua como uma extensão do *JavaScript*, introduzindo a tipagem estática e recursos de orientação a objetos (*TypeScript*, 2023). Ao adicionar essa camada de tipagem ao *JavaScript*, o *TypeScript* oferece um maior nível de segurança e previsibilidade no desenvolvimento *Back-End*. Ele se integra perfeitamente com o *Node.js*, permitindo que os desenvolvedores detectem erros antes da execução e melhorem a manutenção do código. Sua aplicação no *Back-End* reforça a confiabilidade e a escalabilidade das aplicações, ao mesmo tempo que mantém a produtividade no desenvolvimento.

3.5 Front-End

O Front-End é a interface visual com a qual os usuários interagem em um software, englobando elementos como layouts, botões, formulários e animações (Pressman, 2016). É a camada que permite aos usuários interagir com o sistema de maneira intuitiva e agradável, representando a ponte entre a funcionalidade do *Back-End* e a experiência do usuário. O desenvolvimento *Front-End* envolve a criação de interfaces dinâmicas e responsivas, proporcionando a apresentação eficaz dos dados e funcionalidades disponíveis.

3.5.1 React

O *React*, desenvolvido pelo Facebook, é uma biblioteca *JavaScript* de código aberto que simplifica a criação de interfaces de usuário interativas e reativas (React, 2023). Baseado em componentes reutilizáveis, o *React* permite a construção modular de interfaces, facilitando a manutenção e atualização das mesmas. Com seu paradigma de programação declarativa, o *React* permite que os desenvolvedores descrevam como uma interface deve se comportar em diferentes estados, resultando em uma experiência mais fluida para o usuário. Através de sua abordagem inovadora, o *React* tem impactado profundamente o desenvolvimento *Front-End*, proporcionando a construção eficaz de interfaces complexas e dinâmicas.

A utilização do *React* para o desenvolvimento da plataforma trouxe agilidade e confiabilidade. Tornou o processo de desenvolvimento mais rápido por possibilitar que os desenvolvedores se preocupem somente com a aplicação.

3.6 ORM (Mapeamento Objeto-Relacional)

O Mapeamento Objeto-Relacional é uma abordagem que permite a integração entre sistemas orientados a objetos e bancos de dados relacionais, simplificando a interação entre as estruturas de dados do software e as tabelas do banco de dados (Pressman, 2016; Fowler, 2006). Ao eliminar a necessidade de escrever consultas SQL manualmente, o *ORM* oferece uma camada de abstração que agiliza o desenvolvimento, melhora a manutenção e facilita a integração entre o software e a persistência de dados.

Na plataforma desenvolvida, o *ORM* escolhido foi o *Prisma* pela sua facilidade de uso com *Typescript*. Ele faz a ponte dos objetos da aplicação e comunica com o banco de dados, automatizando as interações. A figura 3 mostra como um *ORM* atua.

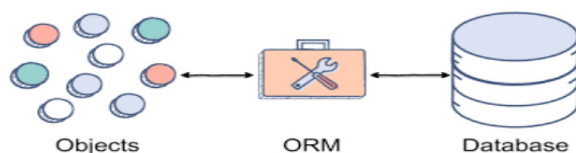


Figura 3. Esquema de funcionamento de um ORM

Fonte: Walker (2022).

Ao mapear os objetos no *Typescript* o *Prisma* automaticamente diminui toda a complexidade de integração com o banco de dados dando maior fluidez ao processo de desenvolvimento.

3.6.1 Prisma

O *Prisma* é um moderno *ORM* que oferece uma experiência eficiente e segura na interação com bancos de dados. Sua abordagem inovadora, que inclui geração de esquemas, migrações automatizadas e segurança nas consultas, possibilita aos desenvolvedores a construção de aplicativos escaláveis e robustos (Prisma, 2023). O *Prisma* é uma ferramenta valiosa para simplificar a complexidade da comunicação com bancos de dados, promovendo maior produtividade e confiabilidade no desenvolvimento.

4. ESTUDO DE CASO PLATAFORMA ESTUDE MELHOR

Nesta parte iremos discorrer sobre a construção do projeto e do estudo de caso da plataforma 'Estude Melhor'. E exemplificar como cada conceito e ferramentas apresentadas na fundamentação foi implementada neste projeto. Começaremos pela modelagem da aplicação, apresentaremos e desenvolvimento do *Back-End* e por último falaremos sobre *Front-End*. As funcionalidades desenvolvidas inicialmente são focadas em disponibilizar o acesso dos alunos às aulas e a realização das questões de fixação. Ao selecionar a matéria escolhida para estudar o seleciona a aula e ao final pode realizar as questões de fixação, estas não possuem tempo limite para conclusão. Todas as questões são obtidas através da internet, possuindo enunciado contextualizado e com cinco asserções, podendo o aluno ver quais questões acertou.

4.1 Planejamento XP

Como já informado anteriormente a plataforma foi desenvolvida usando a metodologia ágil XP. Esta metodologia foi escolhida pela sua simplicidade de aplicação e por ser amplamente usada. A metodologia XP segue um ciclo para o desenvolvimento do projeto, que é: planejamento, projeto, codificação e testes. A seguir, na figura 4 mostramos o que fizemos em cada fase do ciclo.

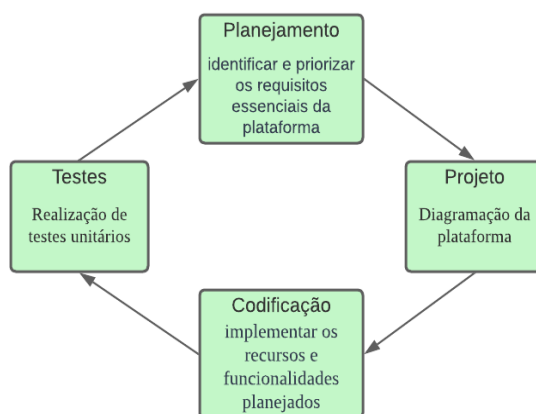


Figura 4. Ciclo XP

Fonte: O autor (2023).

Na fase de planejamento fizemos o levantamento dos requisitos e definimos os essenciais para assim para a fase de projeto. Na fase de projeto fizemos toda a diagramação da plataforma como será mostrado a seguir. Logo após a fase de projeto, foi implementado em código os recursos e funcionalidades planejados. A fase de testes foi feita durante

toda a fase de codificação, a fim de prevenir erros e aumentar a confiabilidade do código.

4.2 Modelagem

Esta etapa consiste em implementar uma modelagem UML, foram usados dois diagramas nessa fase: diagrama de caso de uso e diagrama de classes. A escolha desses diagramas se deu pelo seu aspecto visual e sua precisão em mostrar a interação entre atores e as relações entre classes e atributos.

4.2.1 Diagrama de Casos de Uso

A plataforma é pensada para o aluno agir de maneira autônoma e simplificada, por isso no diagrama apenas aparece o ator 'Aluno' que representa os usuários e o ator 'Administrador' que representa o administrador da plataforma, como mostra a figura 5, a seguir.

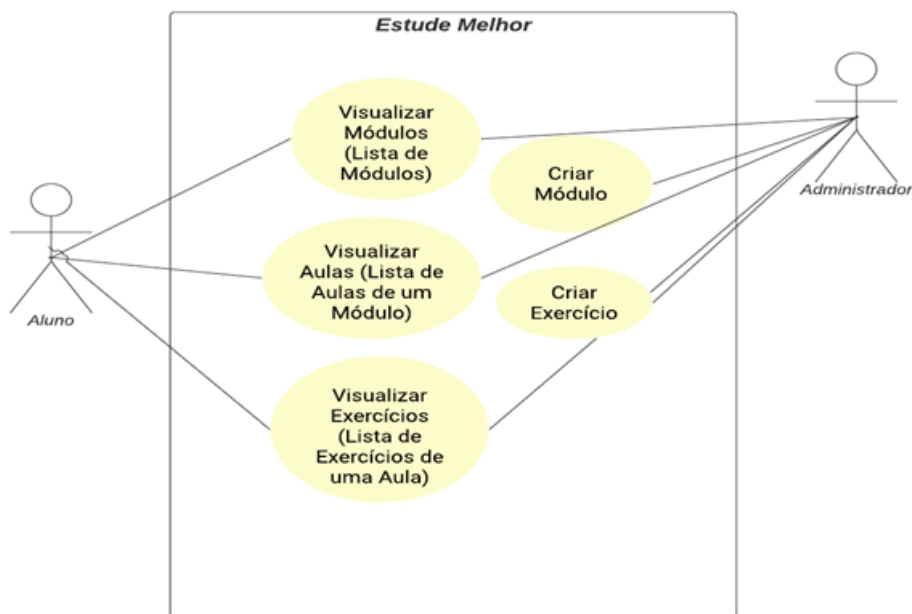


Figura 5. Diagrama de Casos de Uso

Fonte: O autor (2023).

O aluno pode acessar os módulos de cada matéria, assistir as aulas e fazer os exercícios. Já o administrador pode criar módulos e exercícios, assim como acessar tudo que o aluno acessa. Esta simplificação é pensada para haver um maior aproveitamento do tempo de estudo do usuário. Aumentando assim sua produtividade e contribuindo para haver um maior entendimento do conteúdo.

4.2.2 Diagrama de Classes

Este diagrama dentro do sistema define as estruturas das classes e serve de base para outros diagramas dentro da modelagem. A figura 6, a seguir, apresenta as principais classes da modelagem.

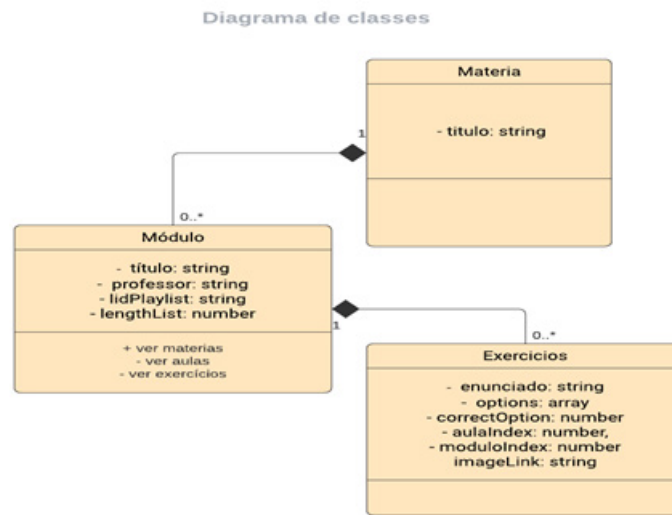


Figura 6. Diagrama de classes

Fonte: O autor (2023).

A classe matéria possui relacionamento com a classe de módulo, sendo esta dependente da anterior. Por sua vez, a classe de exercícios é dependente da classe de módulos.

4.3 Back-End

O *Back-End* da plataforma foi desenvolvido após a modelagem da aplicação é responsável por toda a regra de negócio e fornece os *Endpoints* para o *Front-End* sendo imprescindível o seu funcionamento para as demais partes da aplicação funcionarem de maneira eficiente e eficaz.

4.3.1 API

Para atender as funcionalidades propostas para a plataforma foram desenvolvidos seis *Endpoints*, divididos entre os métodos *GET* e *POST*. O método *GET* é responsável por fazer a leitura de dados no servidor, enquanto o método *POST* é responsável por gravar dados no servidor. A relação de *Endpoints* está descrita figura 7, abaixo:

Método GET	
/cousers/:id	Obtém informações sobre módulos com base no ID do curso. Realiza validação do parâmetro id.
/classes/:idL/lengthList:num	Obtém vídeos de uma playlist com base no ID da playlist e na quantidade de vídeos. Utiliza uma chave de acesso e realiza tratamento dos dados recebidos.
/exercicios/:id/aula_index/:num	Obtém exercícios de um módulo com base no ID do módulo e no índice da aula. Realiza validação dos parâmetros id e num e faz tratamento dos dados antes de enviar a resposta.
Método POST	
/criarMateria	Cria uma nova matéria. Realiza validação dos dados recebidos no corpo da requisição.
/criarModulo	Cria um novo módulo. Valida os dados recebidos no corpo da requisição, verifica a existência do módulo e da matéria associada antes de criar o novo módulo.
/criarExercicio	Cria um novo exercício para um módulo específico. Realiza validação dos dados recebidos e verifica a existência do módulo associado antes de criar o exercício.

Figura 7. Endpoints da API

Fonte: O autor (2023).

A API ficou com seis *endpoints* no total sendo que as três do método *GET* e três do

método *POST*. Todas as rotas possuem validação de parâmetros para evitar o mau funcionamento e assim prejudicar o *Front-End*.

4.3.2 TDD

A escrita de testes é uma etapa muito importante no XP, pois garante a confiabilidade do desenvolvimento além de garantir um software robusto e eficiente. A escrita de testes deve ser feita em todas as etapas do projeto e reescritos a cada modificação. A seguir mostraremos um teste realizado na rota que retorna os módulos. Para isso usamos a biblioteca de testes Javascript *JEST*. Ao total foram inscritos 12 testes para as 6 rotas, totalizando 2 para cada rota, o tipo de teste feito foi o teste de Integração. O teste de integração entre diferentes partes do sistema (o endpoint, o servidor, etc.) garante que, em condições normais e de erro, o sistema se comporte conforme o esperado. A figura 8 a seguir mostra o código.

```

1 // Descrição
2 describe('Testes para o endpoint /cousers/:id', () => {
3   // Teste para verificar se o endpoint retorna status 200 e um array de módulos
4   it('Deve retornar status 200 e uma lista de módulos quando o ID da matéria é válido', async () => {
5     const response = await request(app).get('/cousers/1'); // Supondo um ID de matéria válido
6
7     // Verifique o status da resposta e o tipo do corpo da resposta
8     expect(response.status).toBe(200);
9     expect(Array.isArray(response.body)).toBe(true);
10  });
11
12 // Teste para verificar se o endpoint retorna status 400 quando o ID da matéria é inválido
13 it('Deve retornar status 400 quando o ID da matéria é inválido', async () => {
14   const response = await request(app).get('/cousers/invalid_id'); // Supondo um ID inválido
15
16   // Verifique o status da resposta
17   expect(response.status).toBe(400);
18 });
19 });

```

Figura 8. Endpoints da API

Fonte: O autor (2023).

São feitos dois testes, o primeiro para ver se o endpoint está retornando os dados corretamente, para isso se passa um ID (identificador) válido e espera um status 200 e um array como resposta. No segundo teste, passamos um ID inválido e esperamos o retorno do status 400.

4.3.3 ORM

No desenvolvimento da plataforma usamos o Prisma como ORM, pela sua integração com o Typescript, robustez e confiabilidade. O Prisma além de facilitar a integração de aplicativos com bancos de dados SQL, ele retira a complexidade das operações, o que facilita o desenvolvimento. A figura 9 abaixo mostra a criação de uma tabela no banco de dados.


```

model Modulos {
  id Int @id @default(autoincrement())
  materia_id Int
  title String @db.VarChar(255)
  teacher String @db.VarChar(155)
  idPlaylist String @db.VarChar(255)
  lengthList Int

  materia Materias @relation(fields: [materia_id], references: [id])
  exercicios Exercicios[]
}

```

Figura 9. Mapeamento com ORM

Fonte: O autor (2023).

Repare na facilidade que o Prisma apresenta, em poucas linhas de código podemos fazer a criação de uma tabela no banco, passos todos os campos que queremos, assim como o tipo e relacionamentos, diminuindo a complexidade e otimizando o tempo de resposta.

4.4 Front-End

No Front-End da aplicação foram usadas as tecnologias mais recentes e robustas disponíveis a fim de garantir uma experiência de uso positiva. Essa etapa é muito importante pois interage diretamente com o usuário. Para o desenvolvimento de todo o *Front-End* usamos a biblioteca *React* pela sua facilidade de desenvolvimento, escalabilidade e ampla documentação.

4.4.1 React

A construção das telas foi feita com um *layout* simples e intuitivo visando sempre promover comodidade aos usuários e além de aumentar a concentração. O uso do React para o desenvolvimento trouxe facilidade para o desenvolvimento, além de robustez. A ampla documentação e ferramentas tornaram possível a construção de um layout intuitivo. A plataforma também foi desenvolvida para ser acessada por diferentes tipos de aparelhos, ou seja, totalmente responsiva quanto aos diferentes tipos de telas, além de ser compatível com as versões mais recentes de todos os browsers. A seguir a imagem 10 apresenta a tela de aulas.

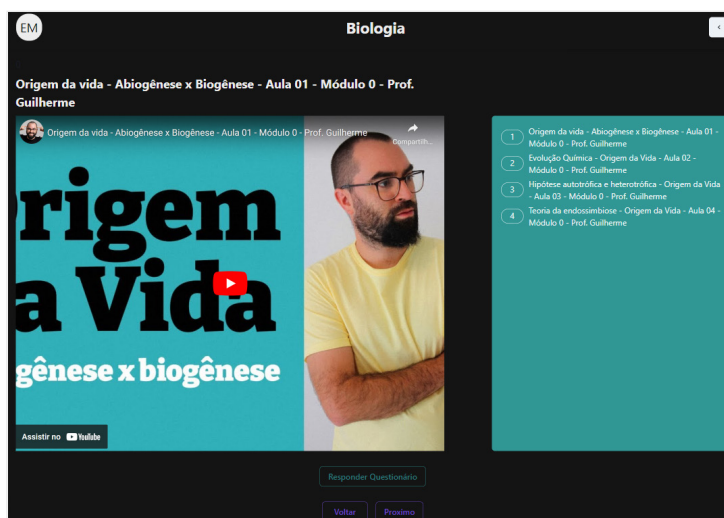


Figura 10. Tela de aulas

Fonte: O autor (2023).

Nesta tela o usuário pode assistir a videoaula, passar para a próxima ou fazer as atividades referente a aula. A adoção de poucas cores serve para haver menos distração e assim maior rendimento na aprendizagem.

Também o aluno pode acessar o questionário referente a cada aula, cada questionário tem um número de questões coletadas da internet de modo a testar o aprendizado dos alunos e promover a fixação do assunto. A figura 11 abaixo mostra uma tela em visão mobile.

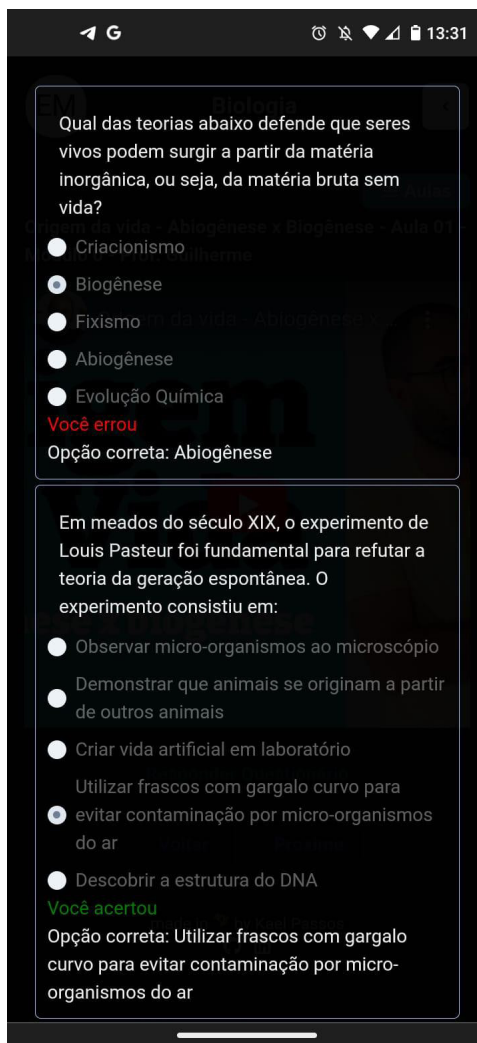


Figura 11. Tela de exercícios

Fonte: O autor (2023).

Como vemos na imagem, a plataforma também faz uma correção automática das respostas, possibilitando assim o aluno verificar se acertou ou errou a questão, feedback importante para o aluno avaliar sua compreensão da aula.

5. RESULTADOS E DISCUSSÃO

A plataforma 'Estude Melhor' é uma plataforma destinada a corroborar com a educação de qualidade gratuita e democrática. O seu desenvolvimento visa ajudar a fortalecer este pilar estratégico para o desenvolvimento do país. Visando a sua validação com o público-alvo, a plataforma com funcionalidades essenciais foi disponibilizada para uso por estudantes de um cursinho em São Luís, Maranhão. Após a interação dos alunos com a plataforma foi feita uma pesquisa pelo Google Forms, abordando aspectos do problema e sobre a

contribuição da plataforma para resolvê-lo. A pesquisa foi realizada entre os meses de outubro e novembro de 2023 na cidade de São Luís, Maranhão. De um público de 100 alunos que frequentam um cursinho preparatório para vestibulares e concursos. Os professores não foram consultados inicialmente devido às funcionalidades desenvolvidas inicialmente estarem focadas no aprendizado autônomo dos alunos, ou seja, de forma extrassala de sala.

A primeira pergunta visa entender o problema pela visão dos alunos. Entender como eles são afetados ajudará a obter uma maior assertividade da plataforma. A pergunta consistia em: **Quais suas dificuldades ao procurar conteúdo de exatas gratuito para estudar?** As respostas são apresentadas no gráfico 1, a seguir.

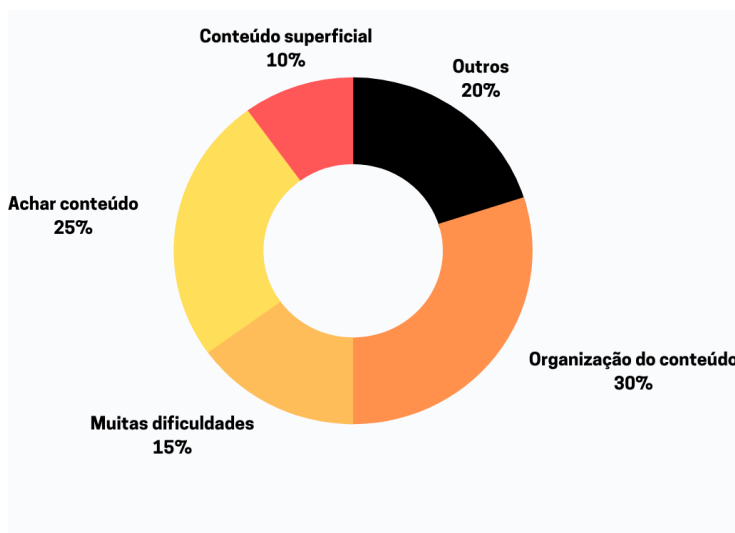


Gráfico 1. Problemas mais frequentes

Fonte: O autor (2023).

Problemas como achar conteúdo, organização e superficialidade representam 55% dos problemas relatados. A plataforma está organizada de forma a mitigar esses problemas, trazendo facilidade no acesso, na organização e complexidade do conteúdo.

A segunda pergunta trata da experiência dos alunos ao interagir com a plataforma e consistiu em: **Qual foi sua experiência na plataforma estude melhor?** A análise das respostas mostra uma avaliação positiva por parte dos alunos, como apresentado no gráfico 2, a seguir.

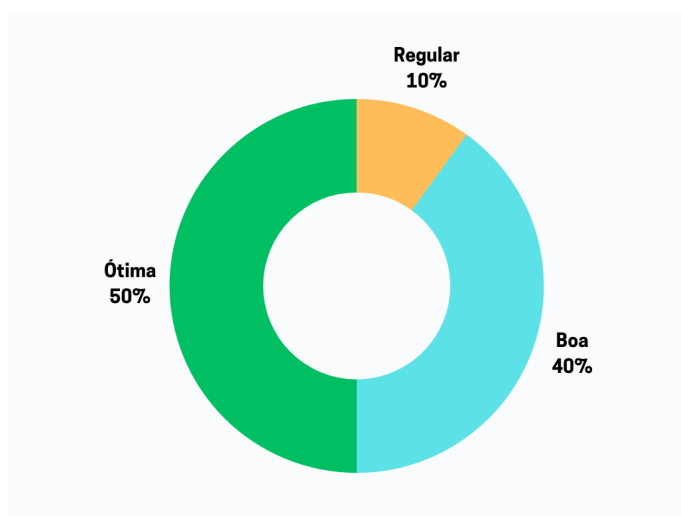


Gráfico 2. Avaliação dos alunos sobre a plataforma

Fonte: O autor (2023).

Destacamos ainda às seguintes respostas:

“É uma excelente plataforma que entrega muita praticidade para os estudos, contendo módulos bem organizados e questionários de fixação”

“Achei a iniciativa muito boa e com certeza quero ver crescer a plataforma”

“Uma Experiência ótima”

Por fim, a terceira pergunta aborda as melhorias que os alunos sugerem para a plataforma. Assim dando um norte de quais melhorias implementar futuramente, a pergunta consistia em: **Quais melhorias você sugere para a plataforma estude melhor?** Os alunos sugeriram diversas sugestões das quais se destacam:

“Suporte ao Estudante: que ofereça suporte ao estudante por meio de *chatbots* ou assistentes virtuais, para responder a perguntas comuns e fornecer orientações instantâneas”

“Que possa ter uma quantidade maior de questões para fixação”

“Adquirir mais conteúdo”

“Possuir cronogramas”

A avaliação dos alunos sobre a plataforma foi positiva, as respostas ressaltaram a importância da iniciativa. Dentre as sugestões de melhorias essas consistiam basicamente em novas funcionalidades e expansão das funcionalidades essenciais já desenvolvidas.

6. CONCLUSÃO

A área de ciências exatas é uma das que mais sofrem e evidenciam as deficiências no acesso de conteúdo de educação de qualidade no Brasil. O cenário educacional marcado por essas lacunas, refletem no baixo desempenho dos estudantes brasileiros em avaliações nacionais e internacionais, impactando o futuro dos estudantes e o desenvolvimento do país.

A fim de mitigar esses problemas, a plataforma ‘Estude Melhor’ se apresentou como uma alternativa inovadora. Até a data de desenvolvimento deste trabalho, não foram encontradas outras plataformas gratuitas semelhantes. A inovação da plataforma também se dá pela sua solução do problema propondo o e-learning como sua contribuição.

Ao focalizar nas disciplinas de exatas e alinhar-se com as urgentes necessidades de melhorias no ensino dessas áreas. Apresentando o conteúdo de forma completa, exercícios para fixação e organização fluida e gradual, se mostra uma aliada poderosa para alunos de todo o país. Os resultados discutidos neste artigo não apenas validam a eficácia da plataforma ‘Estude Melhor’, mas também enfatizam a importância da inovação. Como pontos fortes a plataforma se destaca pela sua fácil usabilidade: plataforma responsiva e design minimalista; conteúdo completo e aprofundado: conteúdo compreende desde a base até mais complexos exigidos no ensino médio; apresenta questões para fixação: questões para fixar o conteúdo estudado.

As sugestões de melhorias dos alunos oferecem orientações para trabalhos futuros, como: implementação de suporte ao estudante por meio de *chatbots* ou assistentes virtuais, criação de cronogramas personalizados, exercícios desenvolvidos por I.A. promovendo maior robustez a plataforma.

REFERÊNCIAS

- BECK, Kent. **Test driven development: By example**. Addison-Wesley Professional. 2022.
- BECK, Kent. **Extreme programming explained: embrace change**. Addison-Wesley Professional, 2000.
- BIEHL, Matthias. **RestFull Api Design: Api you users will love**. API-University Press, 1 ed. 2016.
- DAHL, R. **Node.js: JavaScript on the Server**. 2009. Disponível em: <https://nodejs.org/pt-br/docs/guides>. Acesso em: 15 de ago. 2023.
- FOWLER, M. **Padrões de Arquitetura de Aplicações Corporativas**. Bookman; 1ª ed. 2006.
- GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social** / Antonio Carlos Gil. - 6. ed. - São Paulo : Atlas, 2008.
- GUEDES, Gilleanes T. A. **UML 2 : uma abordagem prática / Gilleanes T. A. Guedes**. 2. ed. São Paulo : Novatec Editora, 2011.
- INEP. **Censo Escolar da Educação Básica 2019**. 2019. Disponível em: <http://portal.inep.gov.br/censo-escolar>. Acesso em 8 ago. 2023.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **O Processo Unificado Explicado**. Campus. 2006.
- JEFFRIES, Ron; ANDERSON, Ann; HENDRICKSON, Chet. **Extreme Programming Installed**. Addison-Wesley Professional, 2000.
- OCDE. **Resultados PISA 2018**. 2018. Disponível em: <https://www.oecd.org/pisa/>. Acesso em 10 ago. 2023.
- PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. AMGH; 8ª ed. 2016.
- PRISMA. **Documentação do Prisma**. Disponível em: <https://www.prisma.io/>. Acesso em: 16 de ago. 2023.
- REACT. **Documentação do React**. 2023. Disponível em: <https://reactjs.org/>. Acesso em: 15 de ago. 2023.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. Addison-Wesley. 2005.
- TYPESCRIPT. **TypeScript: JavaScript Development at Any Scale**. 2023. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 15 de ago. 2023.
- WALKER, Jhony. **ORM - As melhores bibliotecas para JavaScript**. 2022. Disponível em: <https://dev.to/jhonywalkeer/orm-as-melhores-bibliotecas-para-javascript-2pc0>. Acesso em: 18 de ago. 2023.
- WILDT, Daniel; et al. **eXtreme Programming: Práticas para o dia a dia no desenvolvimento ágil de software**. Casa do Código. 2015.



5

UTILIZAÇÃO DA GAMIFICAÇÃO PARA O ENSINO DA LINGUAGEM NORDESTINA

Title of Article in English: Using gamification to teach the Northeastern language

André Kleysson Gois de Carvalho
Edilson Carlos Silva Lima
Leonardo de Castro Mesquita

D.O.I.: 10.29327/5391075.1-5



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

A região nordeste do Brasil é reconhecida por sua notável diversidade linguística, que resulta de uma complexa interação de fatores históricos, culturais, sociais e geográficos. Essa riqueza linguística pode ocasionalmente criar desafios na compreensão mútua, mesmo entre pessoas da mesma região. Para abordar essa questão, foi desenvolvido o jogo “Aventuras pelo Nordestês”, que visa ilustrar as variações linguísticas e culturais presentes no Nordeste brasileiro, proporcionando um meio eficaz de aprender e compreender as diferentes nuances linguísticas dessa região. Essa iniciativa adota uma abordagem metodológica científica baseada em estudos de caso, permitindo uma apreciação mais profunda e significativa das riquezas linguísticas e culturais do Nordeste. E de uma análise qualitativa do resultado da pesquisa sobre o funcionamento do jogo para a aprendizagem e amostra da diversificação linguística nordestina.

Palavras-chave: nordeste, variação, jogo, cultural, estudo de caso.

Abstract

The northeastern region of Brazil is renowned for its remarkable linguistic diversity, which results from a complex interaction of historical, cultural, social and geographical factors. This linguistic richness can occasionally create challenges in mutual understanding, even between people from the same region. To address this issue, the game “Aventuras pelo Nordestês” was developed, which aims to illustrate the linguistic and cultural variations present in the Brazilian Northeast, providing an effective means of learning and understanding the different linguistic nuances of this region. This initiative adopts a scientific methodological approach based on case studies, allowing for a deeper and more meaningful appreciation of the linguistic and cultural riches of the Northeast. And a qualitative analysis of the results of the research into how the game works for learning and sampling linguistic diversification in the Northeast.

Keywords: northeast, variation, game, cultural, case study.

1. INTRODUÇÃO

A região nordeste do Brasil, marcada por sua notável diversidade linguística e cultural, enfrenta uma série de problemáticas complexas. O preconceito linguístico, a perda de identidade cultural, dificuldades no acesso à educação, barreiras na comunicação inter-regional, desafios na preservação da cultura e tradições, desigualdades sociais e econômicas, e a necessidade de programas de educação bilíngue e multilíngue são questões que permeiam a vida dos nordestinos. Valorizar e proteger essa riqueza cultural e linguística é essencial para promover a inclusão, a compreensão e o respeito por todas as manifestações que tornam o nordeste brasileiro tão único e enriquecedor.

Neste artigo, investigamos a aplicação da gamificação como uma poderosa ferramenta educacional para abordar as complexidades linguísticas encontradas nos estados nordestinos do Brasil, destacando as ricas variações culturais e regionais presentes no dialeto nordestino. O nosso propósito não se limita à mera identificação dessas diferenças, mas também se estende a promover uma compreensão mais profunda da riqueza cultural intrínseca a essa região, tornando, assim, o processo de ensino mais envolvente e cativante.

Os jogos desempenham um papel crucial nesse contexto, oferecendo uma experiência interativa que mergulha os participantes em narrativas, diálogos e mecânicas específicas, proporcionando um aprendizado completo e envolvente. A metodologia de estudo de caso será aplicada para analisar a eficácia dessa abordagem inovadora, e este artigo apresenta de forma objetiva a estrutura que será seguida na análise desse método educacional.

2. TRABALHOS RELACIONADOS

Neste capítulo, será realizada uma revisão sobre os seguintes trabalhos: 2.1 Jogos como ferramenta para favorecer a aprendizagem, 2.2 Variação linguística e a representação da identidade nordestina pela linguagem, 2.3 Valorização da variação linguística e da cultura nordestina: uma proposta de intervenção a partir da literatura de Jessier Quirino e 2.4 Aventuras pelo Nordeste.

2.1 Jogos como ferramenta para favorecer a aprendizagem

O artigo em questão fala sobre como os jogos podem ser usados como ferramentas para a aprendizagem. Ele explora as teorias de Piaget e Vygotsky sobre o papel do jogo no desenvolvimento cognitivo e social das crianças, bem como a importância dos jogos educativos e como eles podem ser usados para despertar o interesse dos alunos. O artigo também discute como os educadores podem garantir que os jogos sejam usados de forma eficaz e apropriada em suas salas de aula, e fornece exemplos de jogos educativos que podem ser usados em diferentes áreas do currículo.

O autor destaca que o desafio de implementar o uso de jogos na rotina dos estudantes é grande, mas que se toda a comunidade escolar se dispuser a fazer isso regularmente, é possível alcançar o êxito no processo de ensino-aprendizagem. Ou seja, o autor acredita que o uso de jogos na escola pode ser uma ferramenta poderosa para alcançar os objetivos educacionais que norteiam o trabalho pedagógico.



2.2 Variação linguística e a representação da identidade nordestina pela linguagem

O artigo aborda a relação entre variação linguística e identidade nordestina através da linguagem. Os autores realizam uma análise do livro “Ciço de Luzia”, escrito por Efigênio Moura, explorando como a literatura pode ser uma ferramenta essencial para estudar variações e mudanças linguísticas em uma comunidade. O trabalho também destaca a relevância de examinar questões identitárias, costumes e outros tópicos relacionados à cultura nordestina. A pesquisa evidencia o papel da variação linguística na formação da identidade cultural regional.

O estudo da variação linguística pode oferecer importantes percepções sobre a história, cultura e tradições de uma comunidade, bem como sobre as relações de poder e os sistemas de prestígio social dentro dela. No artigo, também é evidenciado como a variação linguística pode refletir questões de desigualdade social e econômica e por vezes é estigmatizada e considerada inferior por estar associada a grupos marginalizados. Os autores destacam a importância de combater o preconceito linguístico e valorizar a diversidade cultural e linguística do Brasil, promovendo a conscientização e a educação em prol de uma sociedade mais igualitária e justa.

2.3 Valorização da variação linguística e da cultura nordestina: uma proposta de intervenção a partir da literatura de Jessier Quirino

O trabalho apresentado tem como objetivo primordial valorizar a variação linguística e a cultura nordestina no ensino de língua portuguesa, fazendo uso da obra de Jessier Quirino como recurso pedagógico. Por meio de estratégias como leitura, análise e produção de textos, busca-se promover não apenas o entendimento da língua, mas também uma apreciação mais profunda da cultura regional. O projeto enfatiza o letramento crítico e a educação inclusiva, visando fortalecer a interpretação crítica e promover a inclusão social e cultural dos alunos, contribuindo para a valorização da diversidade cultural e linguística, ao mesmo tempo em que combate preconceitos e promove uma educação mais inclusiva e respeitosa.

E enfatiza que há um vasto campo a ser explorado na desconstrução das barreiras do preconceito linguístico, priorizando a valorização da riqueza da diversidade linguística e cultural. Isso implica em combater atitudes preconceituosas e, de forma paralela, impulsionar uma educação que seja não apenas inclusiva, mas também profundamente respeitosa. É essencial compreender que as transformações na língua surgem a partir das necessidades sociocomunicativas dos falantes e estão intrinsecamente ligadas à realidade que os envolve.

2.4 Diferencial do projeto: Aventuras pelo Nordestês

A introdução de um jogo de plataforma que explore a variação linguística nordestina traz um diferencial notável em relação aos textos previamente mencionados. Enquanto os textos anteriores enfatizam a relevância dos jogos na educação e a análise da variação linguística nordestina na literatura, o jogo proporciona uma experiência prática e cativante que transcende a mera teoria.

Ao mergulharem em um ambiente virtual rico em elementos da cultura nordestina e

diferentes sotaques, os jogadores não apenas absorvem conhecimento sobre a diversidade linguística, mas também a experimentam de forma interativa. Essa abordagem dinâmica não somente torna o processo de aprendizagem mais imersivo e memorável, como também promove a valorização da riqueza cultural e linguística do Nordeste brasileiro. Além disso, o jogo oferece uma oportunidade valiosa para combater o preconceito linguístico, pois normaliza a diversidade linguística como algo positivo e enriquecedor, contribuindo, assim, para uma educação inclusiva e respeitosa. Essa abordagem lúdica e inovadora reconhece que as mudanças na língua são intrínsecas à vida cotidiana e podem ser exploradas de maneira construtiva, fortalecendo a compreensão e aceitação das diversas formas de expressão linguística.

3. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será realizada uma revisão referencial dos seguintes tópicos: 3.1 IDE – Game Maker, 3.2 -Metodologia Ágil - Scrum, 3.3 – UML, 3.4 – Padrões de projeto.

3.1 IDE – Game Maker

O GameMaker, uma engine de desenvolvimento de jogos desenvolvida pela YoYo Games, destaca-se como uma escolha popular entre criadores de jogos 2D e, em alguns casos, projetos 3D, devido à sua notável facilidade de uso, possibilitando que tanto iniciantes quanto desenvolvedores experientes concretizem suas visões de jogos de forma eficaz. Com uma interface intuitiva que se baseia em recursos de “arrastar e soltar”, o GameMaker permite a rápida montagem de ativos de jogos, a incorporação de mecânicas personalizadas e o ajuste de elementos visuais e sonoros. “Game maker is ideal for learning game development as it allows you to start making games without having to study a completely new language” (Habgood, 2006). Além disso, o GameMaker oferece uma linguagem de script poderosa chamada GameMaker Language (GML), proporcionando flexibilidade e controle sobre a lógica do jogo, e a capacidade de exportar jogos para diversas plataformas o torna uma escolha versátil para desenvolvedores independentes e estúdios de menor porte. Com uma comunidade ativa e recursos de aprendizado, o GameMaker se destaca no desenvolvimento de jogos, seja para projetos simples ou aventuras complexas, democratizando a criação de experiências interativas.

O GameMaker, como mencionado, é uma ferramenta extremamente versátil e acessível para o desenvolvimento de jogos. Sua facilidade de uso torna a entrada no mundo do desenvolvimento de jogos mais suave para iniciantes, enquanto sua profundidade e a linguagem de script GML oferecem uma plataforma de aprendizado e criação para desenvolvedores mais experientes. O GameMaker também se destaca por fornecer a capacidade de criar jogos para uma ampla variedade de plataformas, o que é essencial para alcançar diferentes públicos. “*In particular, GameMaker was used to create some of the most revolutionary games of the indie scene, like Undertale, Spelunky, Hotline Miami, Gods Will Be Watching, Risk of Rain, and Hyper Light Drifter*” (Cossu, 2019). Sua comunidade ativa e recursos educacionais, como tutoriais e documentação, também facilitam o processo de aprendizado e desenvolvimento. Portanto, o GameMaker continua sendo uma escolha valiosa para aqueles que desejam dar vida a suas ideias e projetos criativos no mundo dos jogos.

3.2 Metodologia Ágil - Scrum

O Scrum, uma metodologia ágil amplamente adotada, é notável por sua eficácia na gestão de projetos, proporcionando estrutura e flexibilidade para equipes que necessitam se adaptar rapidamente às mudanças nas demandas do projeto. Além de incorporar princípios de aprimoramento contínuo e produtos minimamente viáveis, o Scrum permite a obtenção de feedback imediato dos consumidores, o que resulta em ajustes e melhorias constantes no produto. “O que o Scrum faz é promover a união das equipes para criar grandes projetos, e isso exige que todos não apenas vejam o objetivo final, mas que também façam entregas incrementais para atingi-lo” (Sutherland, 2016). A ênfase na comunicação contínua, transparência e responsabilidade compartilhada torna o Scrum uma escolha valiosa para otimizar a eficiência e a qualidade das entregas, promovendo um ambiente de trabalho que valoriza o empoderamento das equipes e a melhoria contínua.

O Scrum é uma metodologia que revolucionou a gestão de projetos, especialmente na era da rápida evolução tecnológica. Sua ênfase na flexibilidade, feedback constante e responsabilidade compartilhada o torna uma abordagem eficaz para equipes que enfrentam desafios dinâmicos. Além disso, a ênfase no empoderamento das equipes e na tomada de decisões descentralizadas impulsiona a criatividade e a colaboração. “Para um time atingir seu objetivo coletivo, é preciso que cada um dentro do time se comprometa individualmente o máximo possível na direção do objetivo coletivo” (Cruz, 2018). No entanto, é fundamental que as equipes se comprometam com a filosofia do Scrum e suas práticas para colher os benefícios completos dessa metodologia. Em resumo, o Scrum continua a ser uma ferramenta valiosa para equipes que buscam se destacar na entrega de projetos eficazes e de alta qualidade.

3.3 UML

A Unified Modeling Language (UML) é uma linguagem visual amplamente empregada no campo da engenharia de software, permitindo que desenvolvedores e arquitetos representem sistemas de forma clara e concisa. “A UML é simplesmente uma notação padrão de diagramação - caixas, linhas etc. A modelagem visual como uma notação comum pode ser uma grande ajuda, mas dificilmente é tao importante quanto saber como projetar e pensar em termos de objetos” (Larman, 2007). Com seus diversos diagramas, a UML facilita a comunicação, documentação e visualização de sistemas complexos, tornando-se uma ferramenta indispensável para o desenvolvimento de software e a compreensão de projetos de alta complexidade.

A seguir, diagramas utilizados na criação do jogo: 3.3.1 – Diagrama de caso de uso, 3.3.2 – Diagrama de classes.

3.3.1 Diagrama de caso de uso

O diagrama de caso de uso, uma ferramenta essencial na engenharia de software, representa as interações entre um sistema e seus atores, desempenhando um papel crítico na identificação e definição dos requisitos funcionais do sistema. “Esses diagramas são importantes principalmente para a organização e a modelagem de comportamentos do sistema” (Booch, 2006). Cada caso de uso ilustra como atores específicos interagem com o sistema para alcançar objetivos definidos, oferecendo uma visão clara e abstrata dos requisitos do sistema. Isso aprimora a comunicação entre a equipe de desenvolvimento e

os stakeholders, ajuda na documentação de funcionalidades cruciais e fornece uma base sólida para testes de aceitação, assegurando que o sistema atenda efetivamente às necessidades dos usuários finais.

Os diagramas de caso de uso são fundamentais na engenharia de software, proporcionando clareza e eficácia na comunicação de requisitos. Além disso, ao destacar a importância da organização lógica e hierárquica dos casos de uso, eles contribuem para uma compreensão abrangente das funcionalidades do sistema, essenciais para projetos iterativos e incrementais. “É possível modelar um sistema complexo com um único diagrama de caso de uso ou ainda criar muitos diagramas de caso de uso para modelar os componentes do sistema” (IBM, 2011). A engenharia de software depende da capacidade de capturar, organizar e comunicar eficazmente os requisitos funcionais, e o diagrama de caso de uso desempenha um papel crucial nesse processo.

3.3.2 Diagrama de classes

O diagrama de classes é uma peça fundamental na linguagem de modelagem UML, amplamente reconhecida na engenharia de software, desempenhando um papel crucial na representação da estrutura estática de sistemas. “O diagrama de classes é uma das ferramentas mais importantes da UML, pois permite representar a estrutura estática do sistema, ou seja, as classes que compõem o sistema, seus atributos e métodos, bem como as relações entre elas” (Guedes, 2018). Ele oferece uma visão abstrata das classes, seus atributos e relacionamentos, e é frequentemente considerado a base para a criação de outros diagramas UML, facilitando a compreensão da estrutura de um sistema, a identificação das classes envolvidas e as associações entre elas. Sua versatilidade torna-o aplicável em diversas fases do desenvolvimento de software, desde a concepção até a implementação e documentação, contribuindo para uma comunicação mais eficaz e uma compreensão mais profunda de sistemas complexos.

O diagrama de classes é uma ferramenta essencial para a modelagem de sistemas de software, proporcionando uma representação visual clara e acessível da estrutura de um sistema. Sua capacidade de servir como base para outros diagramas UML destaca sua importância na engenharia de software, simplificando a identificação de classes-chave, seus atributos e relacionamentos. “Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles” (Fowler, 2005). Esse recurso é valioso para o desenvolvimento de sistemas, ajudando a criar bases sólidas para projetos de software bem-estruturados. Com sua versatilidade e utilidade, o diagrama de classes continua a ser uma ferramenta indispensável na engenharia de software.

3.4 Padrões de Projeto

Padrões de projeto são soluções comuns para problemas frequentes no desenvolvimento de software. “O padrão não é um pedaço de código específico, mas um conceito geral para resolver um problema em particular” (Refactoring Guru, 2014). Eles funcionam como modelos flexíveis, permitindo que os desenvolvedores adaptem as soluções gerais às necessidades específicas de seus programas. Enquanto os algoritmos são como receitas específicas, os padrões são como plantas de construção que fornecem diretrizes gerais, dando liberdade aos desenvolvedores para personalizar a implementação de acordo com seus projetos individuais. Esses padrões tornam o processo de desenvolvimento mais efi-



ciente e organizado.

A seguir, padrões de projeto utilizados na criação do jogo: 3.4.1 – Builder, 3.4.2 – Singleton, 3.4.3 – State.

3.4.1 Builder

O Builder é um padrão de projeto criacional de grande relevância na engenharia de software, especialmente quando se trata da criação de objetos complexos. “Esse padrão de desenvolvimento pode ser utilizado na construção de objetos complexos, com uma abordagem de desenvolvimento “por partes” (Santana, 2020). Sua abordagem de construção passo a passo torna possível dividir o processo de criação em etapas gerenciáveis, permitindo a construção de objetos com múltiplas partes e configurações de forma eficiente. Isso promove a reutilização de código e a flexibilidade ao possibilitar a produção de diferentes tipos e representações do mesmo objeto, simplificando o desenvolvimento de sistemas que demandam variabilidade e personalização.

O Builder, ao simplificar a construção de objetos complexos, torna o código mais legível e modular, facilitando a manutenção e a escalabilidade do sistema. “ Ele permite que você produza diferentes tipos e representações de um objeto usando o mesmo código de construção” (Moreira, 2021). A capacidade de produzir diferentes representações de um objeto com o mesmo código de construção é particularmente valiosa em cenários em que se precisam criar variantes do mesmo objeto, adaptando-se a diferentes necessidades. Isso resulta em um código mais organizado e mais fácil de estender, promovendo uma abordagem mais robusta no desenvolvimento de software orientado a objetos.

3.4.2 Singleton

O padrão Singleton é uma técnica fundamental no design de software, que visa assegurar a existência de apenas uma única instância de uma classe específica em todo o sistema. Essa abordagem é particularmente valiosa em situações em que se deseja controlar o acesso a recursos compartilhados, como conexões de banco de dados, gerenciadores de logs ou configurações globais. “A classe pode garantir que nenhuma outra instância seja criada (pela interceptação das solicitações para criação de novos objetos), bem como pode fornecer um meio para acessar sua única instância” (Gamma, 2000). Para implementar o Singleton, uma classe é projetada de forma que sua instância seja criada apenas uma vez e, em seguida, essa instância é compartilhada e acessada por todas as partes do sistema que a necessitam.

Para atingir esse objetivo, o padrão Singleton propõe a criação de uma classe adicional que serve como o único ponto de acesso à classe original. “Podemos notar que tanto o construtor da classe quanto seu atributo único possuem acesso privado, ou seja, a própria classe será a única responsável por suas instâncias” (Lima, 2019). Isso garante que a instância única da classe seja criada de forma controlada e que todas as solicitações subsequentes de acesso a essa instância sejam redirecionadas para a instância já existente. Dessa forma, o padrão Singleton ajuda a evitar problemas de concorrência e economiza recursos do sistema, tornando-o uma ferramenta poderosa para garantir que recursos críticos sejam gerenciados de forma eficaz em aplicações de software.

3.4.3 State

O padrão State é uma abordagem valiosa no design de software que habilita objetos a ajustarem seu comportamento dinamicamente conforme seu estado interno se modifica, proporcionando uma sensação de que o objeto está mudando de classe. “State pertence ao grupo de padrões comportamentais. Utilizamos ele quando queremos justamente que o comportamento de um objeto mude, de acordo com o seu estado atual” (Bigardi, 2019). Isso é particularmente útil em cenários nos quais um objeto pode se encontrar em diferentes estados e, dependendo de qual estado está ativo, seu comportamento deve ser adaptado de maneira adequada. O State permite que essas transições de estado sejam tratadas de forma organizada, promovendo a clareza do código e a manutenibilidade do sistema.

O padrão State é uma ferramenta poderosa para lidar com objetos que têm comportamentos diferentes em estados distintos. Ele permite que os desenvolvedores modularizem o código relacionado ao estado em classes separadas, tornando-o mais fácil de entender e de estender. Além disso, o State favorece a adição de novos estados e a modificação do comportamento do objeto sem que isso afete outras partes do código. “Quando o estado for alterado, basta trocar a instância utilizada para caracterizar o estado, que consequentemente o comportamento da entidade será alterado” (Guerra, 2014). No entanto, é importante usá-lo com prudência, uma vez que, em casos simples, a complexidade introduzida pelo padrão pode não ser justificada. Portanto, o padrão State é uma ferramenta valiosa para criar objetos flexíveis e adaptáveis, mas deve ser aplicado de acordo com as necessidades específicas do projeto.

4 PROJETO

A concepção de um jogo educacional que explore a riqueza das variações linguísticas do Nordeste do Brasil é, sem dúvida, um projeto instigante e repleto de desafios cativantes. O jogo proposto tem como missão principal transmitir de forma lúdica e envolvente as diversas nuances linguísticas presentes nessa fascinante região do país. Para concretizar essa visão, optou-se por criar um jogo de aventura com o estilo de plataforma, no qual o jogador enfrentará obstáculos que exigem o entendimento e a aplicação das diferentes formas de expressão características do Nordeste. O enredo do jogo se desenrola em um cenário típico da região nordestina, completo com personagens e diálogos autênticos.

Durante o desenvolvimento do jogo, a equipe utilizou a IDE Game Maker, uma escolha que se revelou fundamental para transformar essa visão em realidade. Além disso, o processo de desenvolvimento seguiu um modelo ágil, com reuniões de sprint do framework Scrum realizadas desde o planejamento inicial até a conclusão do projeto. Essas reuniões eram cruciais para determinar quais itens e funcionalidades seriam implementados, garantindo que o jogo atendesse aos objetivos de aprendizado de forma eficaz. Como visto na Figura 1 a seguir uma parte do código de colisão do jogador.



```

3
4 #region Colisao
5 var _col = instance_place(x + velh, y, Obj_chao);
6 #region pixel perfect horizontal
7 if _col {
8     if velh > 0 {
9         x = _col.bbox_left + (x - bbox_right);
10    }
11    if velh < 0{
12        x = _col.bbox_right + (x - bbox_left);
13    }
14    velh = 0;
15 }
16 #endregion
17 #endregion
18
19 x += velh;
20
21 #region pixel perfect vertical
22 var _col = instance_place(x, y + velv, Obj_chao);
23 if _col {
24     if velv > 0 {
25         y = _col.bbox_top + (y - bbox_bottom);
26     }
27     if velv < 0{
28         y = _col.bbox_bottom + (y - bbox_top);
29     }
30     velv = 0;
31 }
32 #endregion

```

Figura 1. Código de colisão do jogador.

Fonte: Autoral, 2023.

O diagrama de Caso de Uso, por sua vez, foi empregado para modelar as interações entre os atores do jogo, ou seja, os próprios jogadores, e o sistema. Esse diagrama permitiu a identificação das principais funcionalidades do jogo, como a movimentação do personagem, interações com NPCs (personagens não jogáveis), e até a resolução de perguntas sobre os estados nordestinos. Como é possível ver na Figura 2. Através desse instrumento, tornou-se mais fácil compreender como o jogador interage com o jogo e como a experiência é moldada.

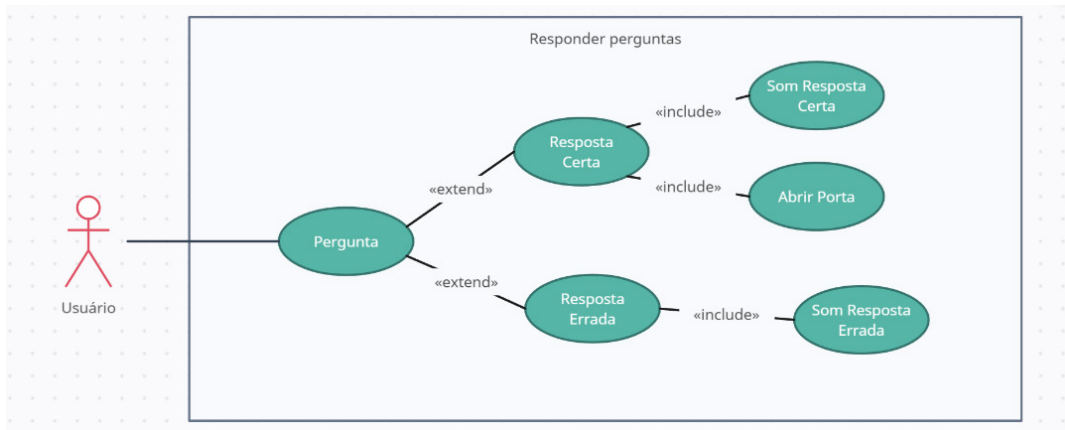


Figura 2. Diagrama de caso, interação com pergunta.

Fonte: Autoral, 2023.

Para modelar a estrutura do jogo, a equipe utilizou o diagrama de Classes, que incluiu classes para personagens, objetos, inimigos, diálogos e outros elementos do jogo. Cada classe foi cuidadosamente projetada com atributos e métodos específicos, o que não apenas organizou o código de maneira eficaz, mas também facilitou a manutenção e a escalabilidade do projeto. A seguir na Figura 3 um diagrama de classe utilizado.

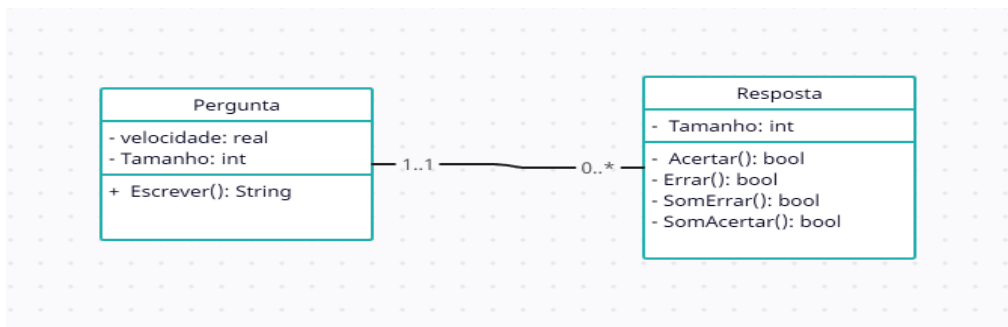


Figura 3. Diagrama de classe, relação perguntas.

Fonte: Autoral, 2023.

Para aprimorar a robustez e flexibilidade do jogo, incorporamos padrões de projeto cruciais, como o Builder, Singleton e State. O Builder desempenhou um papel fundamental na simplificação da criação de elementos complexos, permitindo-nos estruturar componentes do jogo de maneira eficiente e coesa. Aqui citaremos a aplicação do padrão de projeto Builder, o qual foi empregado na criação de diversas representações de um objeto, no caso, o NPC, juntamente com suas variações. Na Figura 4 parte do código dos objetos npc. Todos os objetos npcs tinham a herança de um objeto pai que continha o código da Figura 4, porém na criação dos objetos havia mudanças do texto a ser impresso durante o diálogo com o jogador e no seu sprite (imagem do npc).

```

1 if(!instance_exists(Obj_textbox2)){
2     if place_meeting(x,y,Obj_Player) && keyboard_check_pressed(ord("E")){
3         create_textbox2(text_id);
4     }
5 }
    
```

Figura 4. Código objeto npc_pai.

Fonte: Autoral, 2023.

O Singleton, por sua vez, foi essencial para garantir a existência de uma única instância de recursos compartilhados, como configurações globais, tornando o sistema mais eficiente e seguro. O padrão foi aplicado na criação do objeto “câmera”, conforme ilustrado na Figura 5. O código de implementação do Singleton para a câmera estabelece que, se uma instância da câmera ainda não existir, ela será criada, garantindo que somente uma única instância da câmera exista em qualquer momento. Isso elimina qualquer risco de duplicação indesejada de objetos “câmera” e contribui para um sistema mais robusto e confiável.

```

1 if !instance_exists(Obj_camera){
2     instance_create_layer(x,y,"player",Obj_camera);
3 }
    
```

Figura 5. Código criação do objeto câmera.

Fonte: Autoral, 2023.

Além disso, o padrão State foi empregado com sucesso para gerenciar transições de estados no jogo, proporcionando uma gestão eficaz e dinâmica das mudanças de contexto, o que resultou em um jogo mais organizado e adaptável. O padrão de projeto comportamental chamado “State”, que permite a um objeto alterar seu comportamento em resposta à mudança de seu estado interno. Isso cria a ilusão de que o objeto está mudando de classe. No contexto de um jogo, um exemplo disso pode ser a criação de dois estados para

um jogador: “vivo” e “morto”. Podendo ser observado na Figura 6 quando o estado “morto” é verdadeiro, o jogador é considerado morto, e quando é falso, o jogador é considerado vivo, conforme ilustrado na Figura 6. Isso permite que o jogador se mova quando está vivo e, caso esteja morto, o objeto do jogador entra no estado de morte e reinicia a fase.

```

16 if(morto == false){
17
18     if (global.pause){
19         velh = 0;
20         velv += grav;
21         sprite_index = Spr_player_idle;
22         exit;
23     }
24     //Controle
25
26     //apenas se tiver sem dano
27     if(timer_dano <= 0 and morto == false){
28         velh = (_right - _left) * vel;
29     }else{
30         velh = 0;
31     }
32
33     //jump
34     if _chao{
35         if _jump{
36             velv = -vel_jump;
37             audio_play_sound(Snd_pulo,0,false);
38         }
39         if (velh != 0){
40             sprite_index = Spr_player_run;

```

Figura 6. código objeto jogador.

Fonte: Autoral, 2023.

Essas estratégias bem aplicadas culminaram em um jogo que oferece uma base sólida para a criação de experiências educacionais interativas e envolventes, consolidando a nossa missão educacional de forma eficaz e atraente. Nesse estudo de caso, foi detalhada a concepção e desenvolvimento de um jogo educacional que explora as variações linguísticas do Nordeste do Brasil. A equipe utilizou várias ferramentas e técnicas, incluindo a IDE Game Maker, modelagem de casos de uso, diagrama de classes e padrões de projeto, para criar um jogo que atende aos objetivos educacionais e proporciona uma experiência envolvente e interativa. A integração de diversos elementos e técnicas demonstrou a abordagem de diferentes ângulos usada para resolver os desafios inerentes a esse projeto educacional.

5. RESULTADOS E DISCURSSÃO

A pesquisa qualitativa é uma abordagem valiosa e essencial no campo da investigação social e científica, que se concentra na compreensão aprofundada e contextual de fenômenos complexos. Aqui a utilizaremos para explorar e analisar os resultados de nossa pesquisa sobre o jogo Aventuras pelo nordetês.

A pesquisa envolveu a participação de um grupo do Facebook de viajantes que viajam pelo nordeste composto por um pouco mais de 150 indivíduos, tendo como resultado 32 entrevistados e se concentrou em explorar as seguintes questões:

Como se sentem ao perceber a variação do dialeto entre os estados nordestinos? (Os resultados estão apresentados no Gráfico 1).

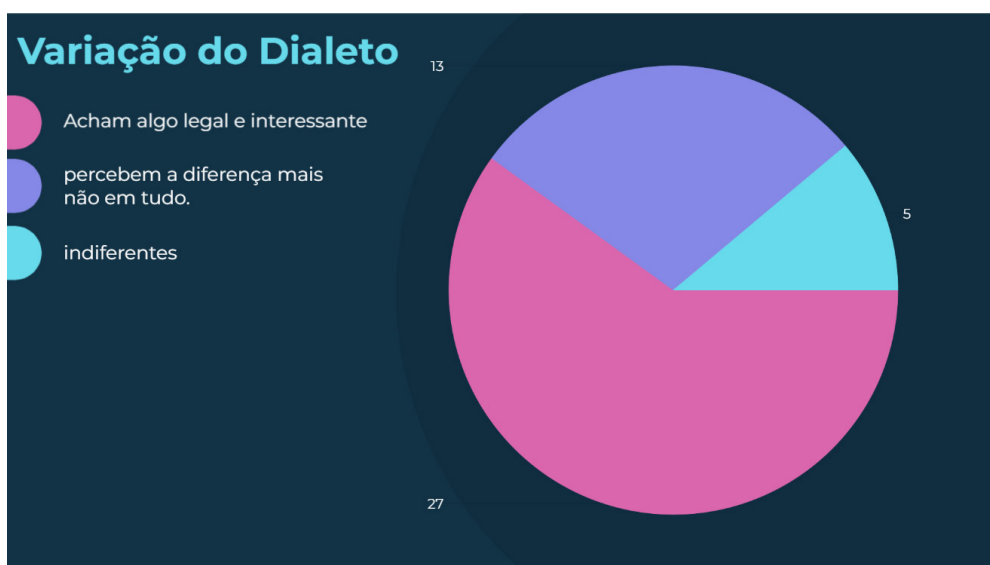


Gráfico 1. Sentimento em relação a variação do dialeto.

Fonte: Autoral, 2023.

A utilização de um aplicativo para o ensino do dialeto nordestino do Brasil é uma estratégia eficaz para promover o aprendizado e a preservação dessa rica variedade linguística. No entanto, com base na experiência com o aplicativo, quais inovações ou melhorias você sugeriria para aprimorar o processo de ensino-aprendizado? (Os resultados estão disponíveis no Gráfico 2).



Gráfico 2. Melhorias Apontadas.

Fonte: Autoral, 2023.

Quais foram os pontos positivos e negativos que você identificou durante a experiência de jogo? (Os resultados são apresentados nas Gráfico 3 e 4).

Onde no Gráfico 3 estão os pontos positivos indicados pelos entrevistados como visto a seguir.

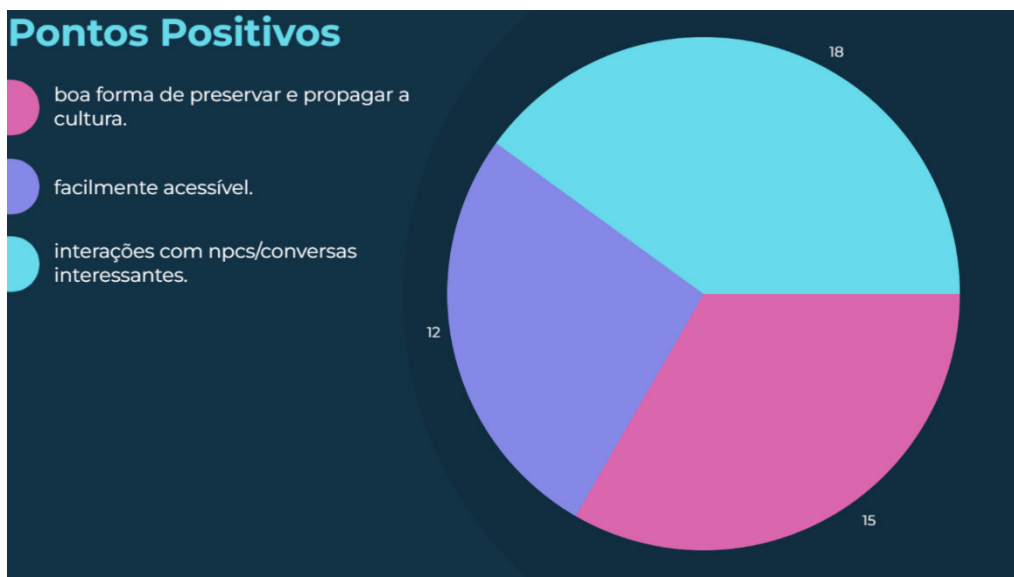


Gráfico 3. Pontos Positivos.

Fonte: Autoral, 2023.

E no Gráfico 4 estão os pontos negativos indicados pelos entrevistados como visto a diante.

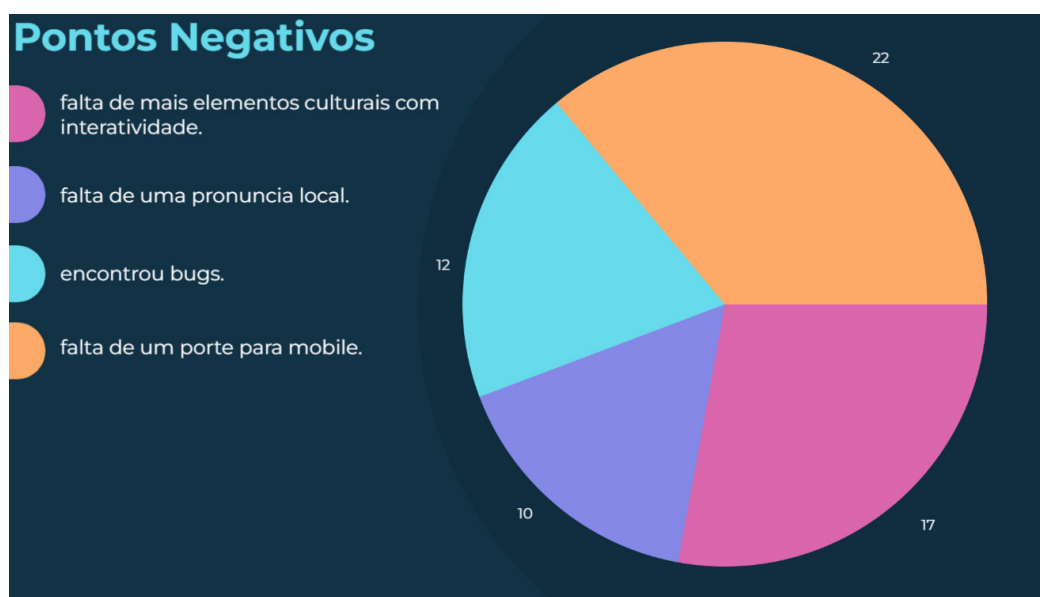


Gráfico 4. Pontos Negativos.

Fonte: Autoral, 2023.

A pesquisa proporcionou percepções valiosas, incluindo a identificação de bugs e a identificação de pontos negativos durante a experiência. No entanto, é importante ressaltar que esses resultados não serão negligenciados, mas sim tratados como áreas de interesse para futuros trabalhos, a fim de aprimorar ainda mais o aplicativo e o processo de ensino-aprendizado do dialeto nordestino do Brasil.

6. CONCLUSÃO

No contexto da valorização e preservação da rica diversidade linguística brasileira, a criação de um jogo que demonstra a variação linguística nordestina representa um marco

significativo. O desafio de apresentar essa riqueza de formas de expressão, sotaques e vocabulários de uma maneira acessível e envolvente foi superado com sucesso. O jogo não apenas entreteve, mas também educou os jogadores, proporcionando uma experiência imersiva na qual eles puderam explorar e apreciar a riqueza da linguagem nordestina.

Para os trabalhos futuros, é essencial continuar a coleta de informações e dados linguísticos para aprimorar a precisão e a autenticidade do jogo. Isso pode envolver a colaboração com linguistas e pesquisadores locais, bem como a realização de estudos aprofundados sobre as diferentes nuances da linguagem nordestina em diferentes regiões e contextos socioculturais. Além disso, é importante considerar a inclusão de elementos de jogabilidade que promovam o aprendizado ativo e a interação social, contribuindo para uma experiência educacional mais rica e envolvente.

A contribuição desse jogo vai além do entretenimento. Ele desempenha um papel crucial na promoção da conscientização sobre a diversidade linguística e na preservação das raízes culturais do Nordeste do Brasil. Ao oferecer uma plataforma divertida e informativa, o jogo ajuda a combater estereótipos e preconceitos linguísticos, fortalecendo o orgulho e a valorização das diferentes variantes do português faladas na região nordestina. Além disso, ao criar interesse e curiosidade sobre a linguagem, o jogo inspira uma apreciação mais profunda da cultura e história nordestina, enriquecendo o patrimônio linguístico e cultural do país como um todo.

REFERÊNCIAS

- BIGARDI, Gustavo. **Arquitetura e Desenvolvimento de software — Parte 16 — State**. <https://gb-bigardi.medium.com/arquitetura-e-desenvolvimento-de-software-parte-16-state-aa998a86277f>, 2019. Acessado em 28/10/2023.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **Uml - Guia do Usuário**. Editora Elsevier, 2006.
- COSSU, Sebastiano. **Game Development with GameMaker Studio 2**. Editora Apress, 2019.
- CRUZ, Fábio. **Scrum e Agile em Projetos**. Editora Brasport, 2018.
- DOMINGOS, José; GUIMARÃES, Éricka de Souza. **Varição linguística e a representação da identidade nordestina pela linguagem**. Brazilian Journal of Development, <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/download/40180/pdf/100589>, 2021.
- Factory Guru. **O que é um padrão de projeto?**. <https://refactoring.guru/pt-br/design-patterns/what-is-pattern>. 2014. Acessado em 20/10/2023.
- FOWLER, Martin. **UML Essencial: Um Breve Guia para Linguagem Padrão de modelagem de objetos**. Editora Bookman, 2005.
- FRANCO, Magda; et al. **JOGOS COMO FERRAMENTA PARA FAVORECER A APRENDIZAGEM**. https://www.editorarealize.com.br/editora/anais/conedu/2018/TRABALHO_EV117_MDI_SAI7_ID7680_07092018192407.pdf, 2021.
- GAMMA, Erich; et al. **Padrões de Projetos: Soluções Reutilizáveis**. Editora Bookman, 2000.
- GUEDES, Gilleanes T. A.. **UML 2 - Uma Abordagem Prática**. Novatec Editora, 2018.
- GUERRA, Eduardo. **Design Patterns com Java**. Editora Casa do código, 2014.
- HABGOOD, Jacob; OVERMARS, Mark. **The Game Maker's Apprentice: Game Development for Beginners**. Editora Apress, 2006.
- IBM. **Diagramas de Caso de Uso**. <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>, 1911. Acessado em 26/10/2023.

LARMAN, Craig. **Utilizando UML e Padrões**. Editora Bookman, 2007.

LIMA, christian. **Padrão Singleton: Como funcionam? Onde vivem? Do que se alimentam?**. <https://medium.com/@christianmellolima/padr%C3%A3o-singleton-como-funcionam-onde-vivem-do-que-se-alimentam-6291fb72b22d>, 2019. Acessado em 24/10/2023.

MOREIRA, Diogo D.. **Padrão Builder**. <https://diogomoreira.gitbook.io/padroes-de-projeto/padroes-gof-criacionais/padrao-builder>, 2021. Acessado em 26/10/2023.

SANTANA, Rodrigo. **Design Patterns com C# Aprenda padrões de projeto com os games**. Editora Casa do Código, 2020.

SILVA, Manuella. **VALORIZAÇÃO DA VARIAÇÃO LINGUÍSTICA E DA CULTURA NORDESTINA: UMA PROPOSTA DE INTERVENÇÃO A PARTIR DA LITERATURA DE JESSIER QUIRINO**. <https://dspace.bc.uepb.edu.br/jspui/bitstream/123456789/25037/1/PDF%20-%20Manuella%20Karen%20Targino%20da%20Silva>, 2021.

SUTHERLAND, Jeff. **A arte de fazer o dobro do trabalho na metade do tempo**. Leya Editorial, 2014.



AMPLIAÇÃO DE APLICATIVO MOBILE COM FLUTTER: FORNECENDO APOIO PEDAGÓGICO E PSICOSSOCIAL AO CORPO DISCENTE E DOCENTE

6

*Expansion of mobile application with Flutter: providing
pedagogical and psychosocial support to the student body and
faculty*

Ada Kezia Freitas Ribeiro Rodrigues

Edilson Carlos Silva Lima

Yonara Costa Magalhães

D.O.I.: 10.29327/5391075.1-6



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Atualmente a tecnologia vem tomando um espaço muito grande no mercado de trabalho no estilo de vida no cotidiano. Entretanto alguns setores ainda não vêm acompanhando toda essa automatização, este projeto foi desenvolvido para o Núcleo de apoio Discente e Docente (NADD) que ainda usam papéis para a criação do seu banco de dados local para armazenar as informações de forma manual, o presente projeto tem o objetivo mostrar uma praticidade e acesso a um App de apoio pedagógico e psicossocial ao discente e docente, o Aplicativo foi desenvolvido em *Flutter*, Foram criado uma interface de usuário e funcionalidades relacionadas aos Eventos do NADD que serão acompanhadas pelo usuário, tornando assim mais rapidez em armazenamento de dados e praticidade para os usuários, Além disso foram adotados uma coleta de informações, foram realizado uma pesquisa com implementação de questionários com quatro questões para fins de análise qualitativa.

Palavras-chave: NADD, Usuário, Praticidade, Discente e Docente.

Abstract

Currently, technology has been taking a very large space in the job market, in lifestyle, in everyday life. However, some sectors have not yet been following all this automation, this project was developed for the Student and Teacher Support Center (NADD) who still use paper to create their local database to store information manually, this project aims to show practicality and access to a pedagogical and psychosocial support App to the student and teacher, the Application was developed in *Flutter*, A user interface and functionalities related to NADD Events were created that will be followed by the user, thus making data storage faster and practicality for users, In addition, an information collection was adopted, a research was carried out with the implementation of questionnaires with four questions for qualitative analysis purposes.

Keywords: NADD, User, Practicality, Student and Teacher.

1. INTRODUÇÃO

Embora estejamos sempre em constante evolução na área Tecnologia este projeto foi desenvolvido para o Núcleo de Apoio ao Docente e Discente (NADD). que usa papeis para a criação do seu banco de dados local para armazenar as informações de forma manual. Este aplicativo tem como o objetivo automatizar esse processo e oferecer um suporte psicossocial e pedagógico de suma importância para o corpo discente e docente da universidade Ceuma.

No desenvolvimento do nosso projeto, adotamos procedimentos metodológicos que foram responsáveis para a realização do nosso projeto como *flutter* no qual fizemos o *Font-end* do nosso projeto *intelliJ IDEA* onde foi realizado o *Beck-end*. todos os outros métodos adotados foi sempre tornando-o mais eficaz e mais prático a interação com os usuários. A coleta de informações abrange diversos aspectos, desde as necessidades e preferências dos usuários até as tendências atuais em tecnologia educacional e apoio psicossocial.

Esta Pesquisa é uma busca constante pela automatização e aprimoramento da experiência de acesso à plataforma do NADD. Nosso objetivo principal é garantir que cada etapa, desde o cadastro até o login e os acessos seja sempre feito com praticidade. Queremos garantir que a interação com a plataforma seja contínua e significativa, permitindo que os usuários se beneficiem plenamente dos serviços e recursos oferecidos.

2. TRABALHOS RELACIONADOS

Neste tópico abordaremos alguns projetos relacionados ao desenvolvimento do NADD, Estes projetos abordam problemas com soluções ótimas e trabalhos futuros sempre visando na automatização de processos e melhor funcionabilidade de sistemas e acesso as determinadas plataformas e melhor percurso em Instituições de Ensino Superior.

No artigo cujo o tema “Desenvolvimento de uma *API REST* utilizando *Spring boot* para um *Front End* desenvolvido em *Flutter*” da autora Allicia Sousa da Silva foi apresentado o seguinte problema Para monitorar o progresso do estado de saúde, diversos profissionais empregaram questionários para avaliar todos os pacientes hospitalizados, permitindo a comparação das respostas a fim de analisar a evolução do quadro clínico, o problema relacionado ao artigo que foi citado foi solucionado com o desenvolvimento de uma *API Spring Boot* projetada para ser utilizada por uma aplicação *Flutter* específica para a avaliação de pacientes neurológicos em Unidades de Terapia Intensiva (UTI), possibilitando o armazenamento de todas as avaliações no banco de dados.

No trabalho apresentado da autora Pamela silva da conceição “*Desenvolvimento de um Aplicativo Para a Avaliação de Pacientes Neurológicos na UTI Utilizando o Framework Flutter*” autora relatou sobre a necessidade de mais tecnologia da medicina, seu App foi desenvolvido especificadamente para a neurologia, realiza todo o controle medico e mostra o quadro clinico dos pacientes internados na (UTI), a aplicação foi feita com êxito e sua aplicação pode ser utilizada em outras áreas como enfermagem, fisioterapia, psicologia etc.

Neste presente artigo “Educação inclusiva no ensino superior: desafios e experiências de um núcleo de apoio discente e docente” de dois centros Universitários de São Paulo da autora Karina Nones Tomelin, o artigo comenta aborda um projeto criado de nome “Projeto Especial” que tem como foco o acolhimento e apoio de estudantes com Deficiência e Pessoas com muita Dificuldade de Aprendizado este projeto consistia em mentorias,



curso e palestras para os estudantes, o projeto desempenhou um papel fundamental em relação à aprendizagem dos envolvidos, e a permanência dos alunos atendidos e a presença ativa dos professores neste processo, o mesmo continua em evolução sempre reavaliado e em constante evolução a cada semestre, mostrando assim que é possível em nível superior uma prática de assistência positiva em redes de ensino superior.

Neste presente artigo aqui apresentado “Ampliação de aplicativo mobile com *Flutter*: fornecendo apoio pedagógico e psicossocial ao corpo docente e discente”, apresentamos uma nova forma de acesso à plataforma NADD uma nova configuração que permite um acesso aos eventos, resultados e atividades do Núcleo de Apoio ao Docente e Discente, permitindo assim uma experiência com mais rapidez, mais praticidade e eficácia ao usuário, projetado especificadamente para melhor experiência do usuário.

3. FUNDAMENTAÇÃO TEÓRICA

Apresentaremos nesta pauta ferramentas que foram usadas para a realização do nosso projeto no *Back-End* e *front-End*. linguagem de programação e funcionalidades e atuação no projeto 3.1 *flutter*, 3.2 *Spring boot*, 3.3 *Workbench*, 3.4 *API rest*.

3.1 Flutter

“O Flutter utiliza a linguagem DART que apresenta semelhanças ao desenvolvimento de linguagens conhecidas, como C e Java, e ainda apresenta documentação ampla” (Vieiralves, 2020, p. 14) *flutter*, foi desenvolvido pelo Google, é um framework de código aberto que possibilita a criação de aplicativos móveis multiplataforma. Com ele, os desenvolvedores podem construir aplicativos para Android e iOS a partir de um único código-fonte. a linguagem a ser utilizada é a linguagem de programação Dart, reconhecida por sua rapidez e eficiência. O Flutter oferece um conjunto completo de widgets personalizáveis que possibilitam aos desenvolvedores criar interfaces de usuário ágeis e responsivas.

O *Flutter* é um *framework* multiplataforma, cujo foco inicial seria entregar a capacidade de que, com o mínimo de alteração de código e com a maior velocidade e fluidez na utilização final da aplicação, ainda seja possível desenvolver uma mesma aplicação para vários ambientes (Vieiralves, 2020, p. 14.)

3.2 Spring boot

“O *Spring Boot* é um projeto da *Spring* que veio para facilitar o processo de configuração e publicação de aplicações *Back-end*., fazendo uso de padrões e recursos que reduzem drasticamente o tempo e complexidade do desenvolvimento” (Silva, 2019, p. 11) O *Spring Boot* é um Projeto de código aberto que simplifica o desenvolvimento de aplicativos Java, com ênfase na produtividade, fornecendo um ambiente de configuração mínimo. Baseado no framework Spring, ele facilita tarefas como configuração e gerenciamento de dependências, permitindo o desenvolvimento rápido de aplicativos independentes prontos para produção.

3.3 MySQL Workbench

“MySQL Workbench, sendo uma ferramenta de desenvolvimento de banco de dados que oferece opções para modelagem, administração, criação e manutenção do banco de dados” (Bronzatto,2019, p. 3) O *Workbench* é Uma Aplicação de gerenciamento de bancos de dados que fornece uma interface gráfica para simplificar a administração e desenvolvimento de bancos de dados. Um dos principais tópicos abordados é o MySQL. MySQL é um sistema de gerenciamento de banco de dados relacionado que é compatível com o Workbench. Isso permite aos usuários acessarem, consultar, gerenciar e desenvolver bancos de dados MySQL de maneira eficaz.

3.4 API REST

Uma API REST (Representational State Transfer) é uma abordagem comum para desenvolver serviços web, REST surge atualmente como uma forma rápida, eficiente e simples de solucionar esse problema de comunicação entre diferentes sistemas construídos utilizando diferentes linguagens de programação (Puluceno, 2012). Ela utiliza princípios padronizados, como métodos HTTP (GET, POST, PUT, DELETE) para operar em recursos identificados por URLs. As APIs REST são conhecidas por sua escalabilidade, flexibilidade e interoperabilidade, permitindo uma comunicação eficaz entre sistemas. Eles retornam dados em JSON ou XML, simplificando a troca de informações. Devido à sua eficiência e simplicidade, as APIs REST são amplamente utilizadas em aplicações web, aplicativos móveis e integrações de sistemas.

3.5 Metodologia

A metodologia abordada no presente artigo quantitativa. A coleta de dados forma feitas através de questionários, buscando a opinião do público alvo afim de coletar as informações, as ferramentas que foram utilizadas para realização do nosso trabalho foram *flutter com a linguagem de programação Dart* auxiliando na realização do *Front-end, Spring boot*, fornecendo toda o pacote em *Spring* com todas as dependencias fornecidas *Workbench*, usamos para a criação do nosso banco de dados, *API rest*. Para programar todo o nosso banco usamos a *intelliJ IDEA* foi usado linguagem Java para a criação da nossa *API*.

4. RESULTADOS E DISCUSSÃO

Neste segmento, mostraremos o desenvolvimento do projeto que irá ser apresentado como foi realizado o font-end, e beck-end., o banco de dados, API, toda a estrutura do nosso projeto.

4.1 Front-end

O *Font-end* é a parte de suma importância do desenvolvimento do aplicativo tem a ligação com o banco de dados e nele podem ser programadas as telas que faram a exibição de elemento desejado pelo programador e podem ser programados em várias linguagens *Java,Python* entre outras. O presente projeto teve seu *Font-end* programado no ambiente



Visual Studio Code um ambiente de fácil manuseio e com inúmeras funcionalidades o *VS (Visual Studio Code)* é um editor de código-fonte desenvolvido pela *Microsoft* uma ótima ferramenta para os desenvolvedores. Como ilustrado na figura 1 mostraremos o projeto em *flutter* com os pacotes e dependências associadas ao projeto e seguiremos apresentando o desenvolvimento.

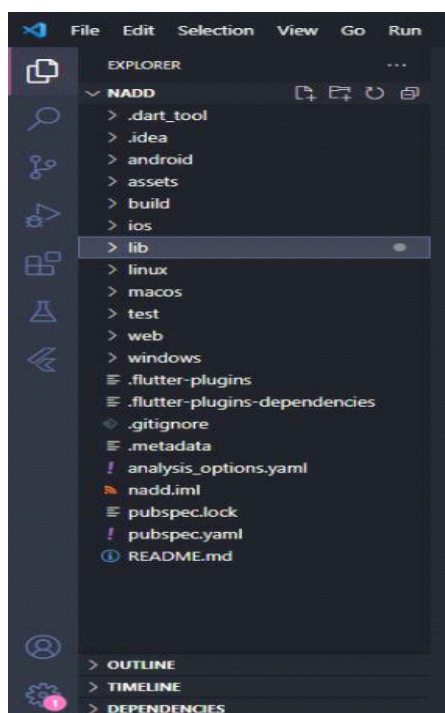


Figura 1. Aplicação *flutter*.

Fonte: Autoral (2023).

Como ilustrado na figura temos apresentado uma aplicação em *flutter*. *Flutter* é um framework que é programado em linguagem *Dart* é utilizado uma linguagem Moderna, o *framework flutter* vem sendo usado em multiplataformas para desenvolver projetos com um código nativo suas aplicações podem ser para *Ios*, *Andoid*, *Web*, em nossa aplicação como ilustrado na imagem temos dentro de cada pacotes funcionalidades que se conectam para melhor funcionamento do projeto.

No Pacote *Lib* temos a principal diretório do projeto com toda a programação da interface do nosso projeto e autenticação das funcionalidades dentro dele temos o *main.dart* que é a origem da nossa aplicação, ainda falando sobre a pasta *Lib* temos as *widgets* que são funções que são os blocos de construção de todas as interfaces dos usuários, configurações de botões de telas todo o esqueleto do nosso projeto. Como mostrado na figura 1 os plugins como *Flutter plugins* e *Flutter plugins dependencies* são pacotes onde pode-se criar um meio de conexão com os sistemas *Andoid* ou *Ios* no qual forem hospedados para utilizar dos recursos do sistema como cameras ou *GPS*, são dependências que estão associadas ao *pubspec.yaml* diretório onde pode-se analisar, acrescentar novas dependências.

O *Gitignore* é um diretório importância grande no projeto nele ficam armazenados os arquivos e melhores versões de um sistema, ele delega quais projetos o *Git* deve ignorar é uma boa pratica esta ação que por conta disso ocorre sempre uma limpa de arquivos que poderiam atrapalhar a ação de compilação do projeto. Já o *Metadata* é um diretório que é criado automaticamente com o projeto eles ficam guardados os metadados do projeto.

Para a Criação do Aplicativo *NADD* em ênfase no usuário foram utilizados alguns requisitos como coleta de informações dos indivíduos como informações como ilustrado na

figura 2 a tela de cadastro na figura 3 ilustraremos todo os eventos do NADD que estarão disponíveis como resultados atividade e atividades projeto onde o usuário pode ter acesso as datas de início e fim de suas atividades e resultado de cada uma delas.

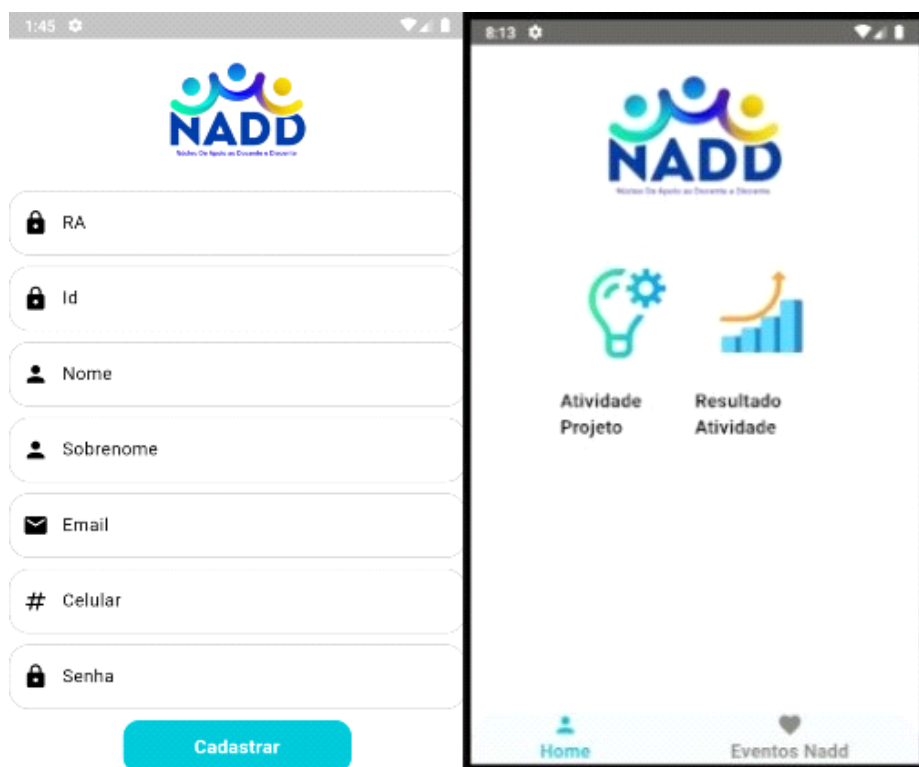


Figura 2. Tela De Cadastro e Tela Home

Fonte: O Autor (2023).

Como ilustrado na figura 2 da Tela de Cadastro onde é recolhido todas as informações necessárias do usuário para seu acesso na plataforma NADD automatizando a rapidez e acesso a plataforma e Tela Home do App onde podem ser visualizadas os eventos do NADD suas Atividades e Resultado das mesmas.

4.2 Workbench

A ferramenta muito utilizada pelos desenvolvedores *Workbench* nele podem ser criados bancos de dados de aplicativos com determinadas ações diferentes, no desenvolvimento do projeto aqui apresentado, foram criado todo o banco de dado no *Workbench* com a visualização de 12 tabelas, colunas e chaves estrangeiras que relacionam informações de tabelas entre- si, mostraremos na figura a seguir figura 3.

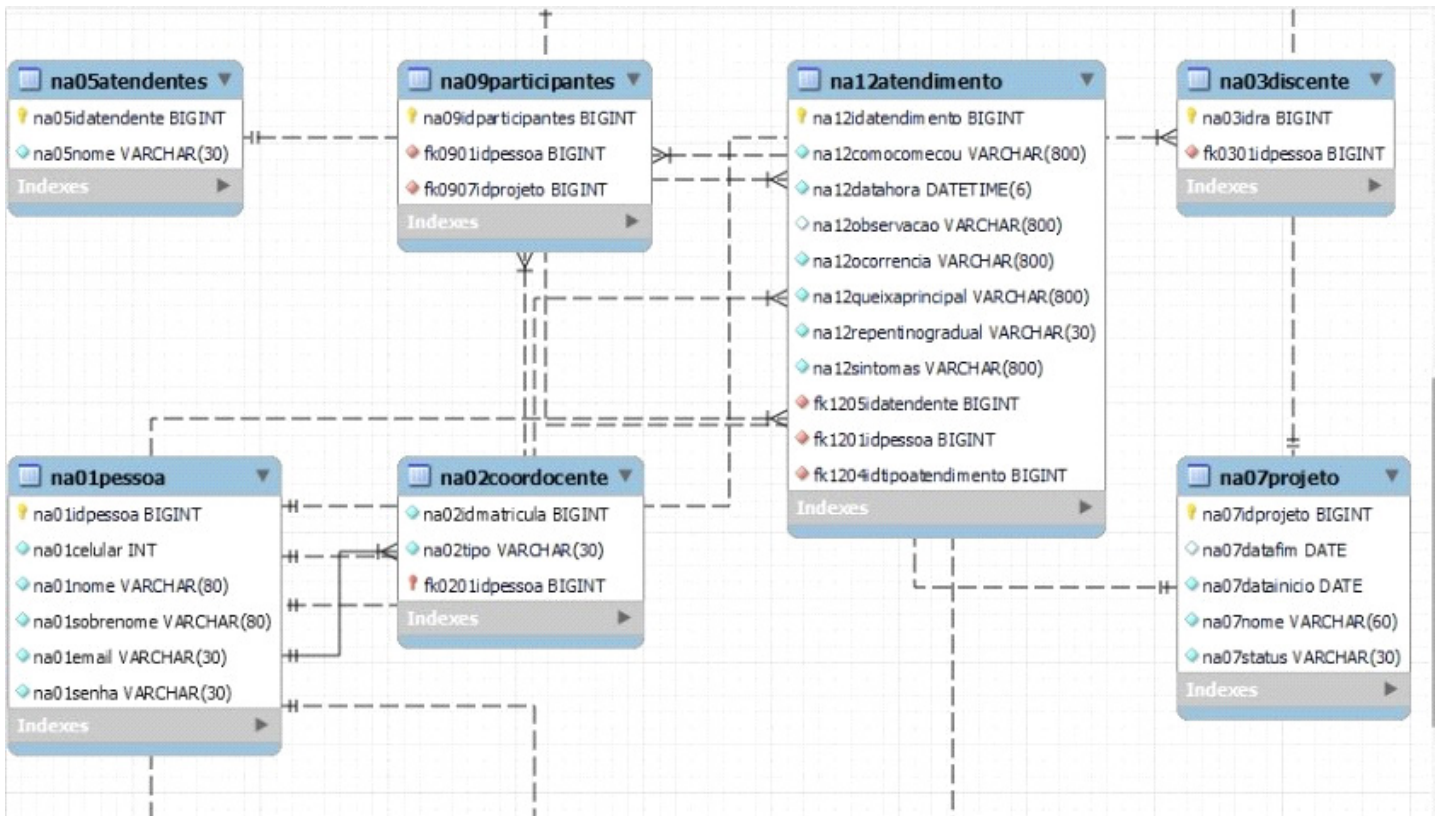


Figura 3. Diagrama de Entidades Relacionadas

Fonte: O Autor (2023).

Como Visualizado na figura 3 temos Diagrama De Entidades Relacionadas são diagramas que mapeiam todo o fluxo do banco de dados do projeto apresentado e fazem a conexão com a *API rest*. e consequentemente visualizando todas as informações no *Font-End*.

4.3 API rest

O *IntelliJ IDEA* é Uma ferramenta que é crucial para o desenvolvimento de software podendo ser programado em *Java,Scala* entre outras linguagens, o projeto apresentado foi desenvolvido na linguagem *Java* no mesmo ambiente de desenvolvimento já citado, foram criado classes , interfaces, *enums*, repositórios todos para o melhor funcionamento do projeto como mostra na imagem 5 apresenta uma classe Resultado atividade com colunas e uma chave estrangeira que relaciona esta tabela apresentada com outra de atividades projeto .


```

1 package br.com.nadd.naddapi.Domain.model;
2
3 > import ...
4
5 28 usages
6
7 @JsonRootName("resultadoatividade")
8 @Data // tem a funcionalidade de: @Getter, @Setter, @EqualsAndHashCode e outros ver documentação
9 @Entity
10 @Table(name = "naddresultadoatividade")
11 public class ResultadoAtividade {
12
13     @EqualsAndHashCode.Include
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     @Column(name = "naddidresultado")
17     private Long id;
18
19     @Column(name = "naddnota", nullable = false, length = 45)
20     private String nota;
21
22     @ManyToOne
23     @JoinColumn(name="fk1108idatividadesprojeto", nullable = false)
24     private AtividadesProjeto atividadesprojeto;
25
26 }
27
28
29
30

```

Figura 4. Classes

Fonte: O Autor (2023).

Na figura acima temos imagem de uma Classe do nosso projeto na linguagem de programação *Java*, esta Classe é Resultados Atividade uma classe que concidentemente aparece em nossa Tela Home como apresentado neste Capítulo acima, esta classe receberá a nota da atividade realizada pelo usuário que terá acesso a plataforma, ela é a presenta de uma chave estrangeira que fará relação de entre está tabela e outra do banco de dados. Portanto será esta informação mostrada nesta classe que será visualizada pelo usuário pela Tela Home.

5. ESTUDO DE CASO: NADD

Neste Tópico apresentaremos sobre um questionário que foi aplicado com 9 alunos da universidade Ceuma no qual foram aplicadas 3 questões objetivas e 1 discursiva como mostraremos nas figuras 6 e 7 e aplicando o questionário mostraremos as o resultado da pesquisa iremos apresentar as perguntas e as respostas dos entrevistados.

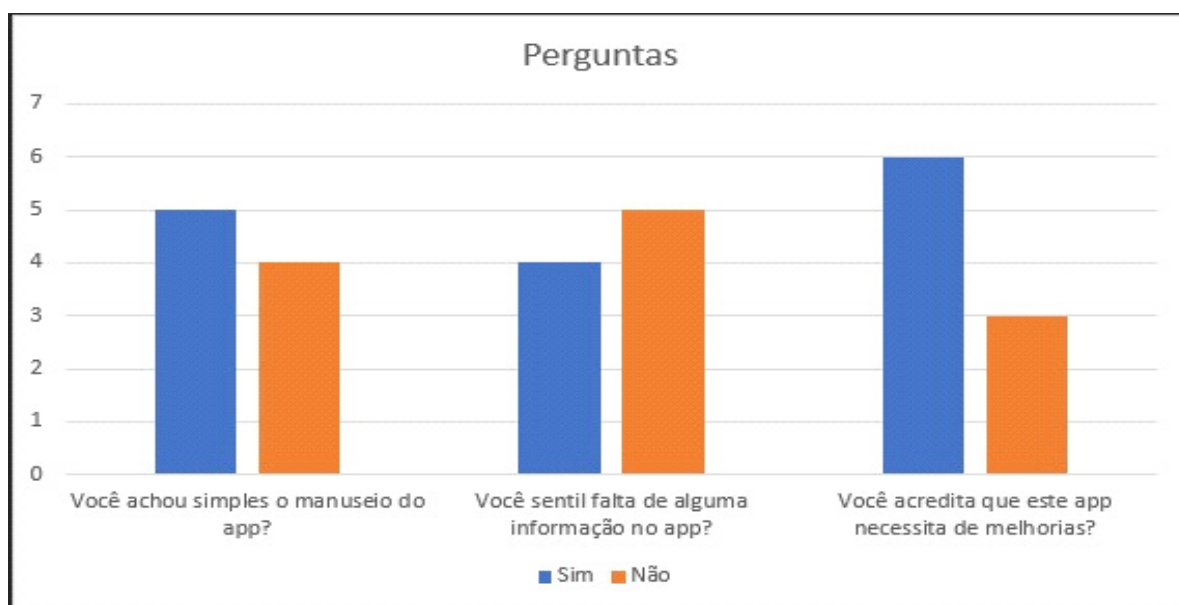


Figura 5. Formulário

Fonte: O Autor (2023).

No gráfico acima temos perguntas feitas de maneira em sucinta e objetiva aos usuários para saber sua opinião sobre a plataforma perguntas Relacionadas a facilidade de manuseamento do Aplicativo, Praticidade no App.

Pergunta	Resposta
Se você respondeu que o aplicativo necessitava de melhorias quais seriam essas?	Creio que seria interessante uma aba de feedbacks para ver a opinião de outras pessoas.
Se você respondeu que o aplicativo necessitava de melhorias quais seriam essas?	Respondi que não havia.

Tabela 1. formulário da questão subjetiva

Fonte: O Autor (2023).

Como mostrado na tabela da pergunta discursiva apresentada é importante frisar que a opinião dos usuários a respeito do App é de suma importância, consideramos cada opinião dada, como algo construtivo para que o aplicativo NADD possa atender as metas desejadas e sempre mostrando, mas melhorias.

6. CONCLUSÃO

No Contexto atual observa-se sempre o aumento da tecnologia sempre facilitando o cotidiano, o aplicativo apresentado neste artigo tem como função automatizar a interação do usuário com o NADD (Núcleo de apoio Discente e Docente), no decorrer deste artigo mostramos como um aplicativo pode melhorar fortalecimento da relação entre NADD e discentes docentes. O aplicativo NADD é uma ferramenta inovadora e indispensável, oferecendo um grande suporte ao fornecer uma plataforma que integra recursos de apoio aos discente e docentes.

O Aplicativo NADD que Fornece Auxílio psicossocial e pedagógico ao discente e docente da universidade Ceuma é um App que automatiza o processo de cadastro de usuário

que muito mais rapidez na plataforma o usuário sempre terá acesso aos eventos do NADD, é um app que suprir a necessidade de rapidez de cadastro do usuário na plataforma de maneira mais eficaz sem a necessidade de execução manual tudo com mais eficácia.

O Projeto fornece auxílio ao discente e docente, contudo mas pode ocorrer sim melhorias nas interfaces nas funcionalidades projeto pode ser automatizado podemos considerar como trabalhos futuros, Entretanto o projeto cumprir suas funcionalidades esperadas e podendo assim ser reutilizado em pesquisa de shows, pesquisa de eventos.

REFERÊNCIAS

BRONZATTO, Marcello. **DESENVOLVIMENTO DE UM SISTEMA DE ENVIO DE NOTAS FISCAIS ELETRÔNICAS DE SERVIÇOS**. Google acadêmico. São Paulo, 2019. 6 p. Disponível em: https://scholar.google.pt/schhp?hl=pt-BR&as_sdt=0,5. Acesso em: 22 nov. 2023.

CONCEIÇÃO, Pamela silva conceição. **Desenvolvimento de um Aplicativo Para a Avaliação de Pacientes Neurológicos na UTI Utilizando o Framework Flutter**". São Luis. 12 p.

PULUCENO, THIAGO . **ESTUDO DE CASO SOBRE UMA API REST UTILIZANDO A ABORDAGEM DE PROGRAMAÇÃO ORIENTADA E EVENTOS COM A PLATAFORMA NODE.Js**. Google acadêmico. FLORIANÓPOLIS - SC, 2012. 74HIAGO VIEIRA PULUCENO p. Disponível em: <https://scholar.google.com.br/?hl=pt>. Acesso em: 22 nov. 2023.

SILVA, ALICE FERNANDES. **GERADOR DE CÓDIGO PARA UMA API REST COM BASE NO FRAMEWORK SPRING BOOT**. Google acadêmico. CAMPINA GRANDE PB, 2019. 11 p. Disponível em: https://scholar.google.pt/schhp?hl=pt-BR&as_sdt=0,5. Acesso em: 22 nov. 2023

SILVA, Allicia Sousa. **Desenvolvimento de uma API REST utilizando Spring boot para um Front End desenvolvido em Flutter**. São Luís. 13 p. Disponível em: . Acesso em: 22 nov. 2023.

VIEIRALVES, LUIZ EDUARDO S. **ANÁLISE DE LINGUAGEM MULTIPLATAFORMA COM FOCO EM FLUTTER**. Google Acadêmico. Novo Hamburgo, 2020. 78 p. Disponível em:<https://scholar.google.com.br/?hl=pt>. Acesso em: 22 nov. 2023.

TOMELIN, Karina Nones . **“Educação inclusiva no ensino superior: desafios e experiências de um núcleo de apoio discente e docente**. Google acadêmico. São Paulo, 2018. 10 p. Disponível em: https://scholar.google.pt/schhp?hl=pt-BR&as_sdt=0,5. Acesso em: 22 nov. 2023.



DESENVOLVIMENTO INTEGRADO DE FRONT-END E BACK-END PARA APRIMORAR A EXPERIÊNCIA DO USUÁRIO

7

*Integrated Front-End and Back-End Development to Enhance User
Experience*

Kaio Eduardo Pereira Cruz

Edilson Carlos Silva Lima

Elda Regina de Sena Caridade

Jonathan Araújo Queiroz

D.O.I.: 10.29327/5391075.1-7



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Sites na internet dedicados a fornecer informações sobre viagens, destinos turísticos, serviços relacionados a viagens e opções de hospedagem. Esses sites são projetados para ajudar os usuários a planejar suas viagens, encontrar acomodações, descobrir atrações locais, obter informações práticas sobre destinos e, muitas vezes, realizar reservas online. Neste contexto, este artigo descreve o desenvolvimento de uma página web que desempenha papel fundamental no projeto Matrip, um aplicativo de viagens criado em *Flutter*. O objetivo principal deste trabalho foi abordar um problema, como a falta de integração em serviços de viagens online. Frequentemente, os usuários enfrentam a dificuldade de acessar informações centralizadas sobre destinos, acomodações e atrações turísticas, além de enfrentar obstáculos ao tentar realizar reservas online de maneira eficiente detectado na promoção do turismo na região do Maranhão, o trabalho buscou, assim, desenvolver uma solução tecnológica, tornando o turismo local mais acessível e atrativo para turistas e colaboradores. Assim, simplificando processo de inclusão dos produtos oferecidos pelo app, demonstrando a importância da tecnologia na gestão. Além disso, o Matrip visa proporcionar aos usuários uma experiência completa na exploração dos principais destinos turísticos do estado do Maranhão, como Carolina, Raposa, Alcântara e Lençóis Maranhenses. A metodologia adotada incluiu as pesquisas bibliográficas extensas para embasar as decisões de desenvolvimento. Os resultados alcançados incluem *back-end* em *Java*, *front-end* em *HTML* e usa um banco de dados *MySQL* para armazenar informações sobre passeios e restaurantes. Em resumo, o Matrip é uma solução inovadora para promover o turismo na região, destacando a importância da tecnologia na divulgação de destinos turísticos e no impulso à economia local.

Palavras-chave: Flutter, MySQL, Front-end, Back-end

Abstract

Websites dedicated to providing information about travel, tourist destinations, travel-related services and accommodation options. These sites are designed to help users plan their trips, find accommodation, discover local attractions, obtain practical information about destinations and, often, make reservations online. In this context, this article describes the development of a web page that plays a fundamental role in the Matrip project, a travel application created in *Flutter*. The main objective of this work was to address a problem such as the lack of integration in online travel services. Users often face the difficulty of accessing centralized information about destinations, accommodations and tourist attractions, in addition to facing obstacles when trying to make online reservations efficiently detected in the promotion of tourism in the Maranhão region, the work sought to develop a solution technology, making local tourism more accessible and attractive for tourists and employees. Thus, simplifying the process of including products offered by the app, demonstrating the importance of technology in management. Furthermore, Matrip aims to provide users with a complete experience in exploring the main tourist destinations in the state of Maranhão, such as Carolina, Raposa, Alcântara and Lençóis Maranhenses. The methodology adopted included extensive bibliographic research to support development decisions. The results achieved include a *Java* back-end, *HTML* front-end and uses a *MySQL* database to store information about tours and restaurants. In summary, Matrip is an innovative solution to promote tourism in the region, highlighting the importance of technology in promoting tourist destinations and boosting the local economy.

Key-words: Flutter, MySQL, Front-end, Back-end



1. INTRODUÇÃO

As crescentes mudanças na relação entre as pessoas e o meio ambiente têm destacado os complexos problemas do turismo em áreas naturais. De acordo com Silva (2013), a natureza passou a ser considerada um produto a ser consumido, resultando em um aumento significativo dos investimentos destinados à conservação desses espaços naturais e, por conseguinte, atraindo um número cada vez maior de turistas que buscam experiências imersivas. Estes desenvolvimentos levantam desafios e oportunidades relacionados com o equilíbrio entre a utilização turística, a conservação da natureza e o desenvolvimento comunitário sustentável.

O turismo desempenha um papel importante na conservação ambiental, enfatizando a necessidade de preservar os recursos naturais e a biodiversidade. Ao mesmo tempo, este tipo de turismo proporciona oportunidades de desenvolvimento econômico em comunidades com paisagens e diferentes tipos de culturas, o que pode contribuir para o bem-estar das comunidades locais. No entanto, é importante compreender os aspectos deste setor, uma vez que o ecoturismo também pode levar a impactos negativos, tais como a degradação ambiental e conflitos socioeconômicos (Anafisco, 2023).

A metodologia desta pesquisa envolve abordagens interdisciplinares, incluindo revisões de literatura sobre turismo em áreas naturais e melhores práticas no desenvolvimento de aplicações móveis e web. Além disso, a pesquisa consiste na implementação do aplicativo Matrip, integrando *front-end* e *back-end*, utilizando tecnologias como *Flutter*, *Java*, *HTML* e *MySQL*. Pesquisas práticas foram realizadas com usuários reais para avaliar como o aplicativo e página web podem ser aplicadas e suas eficácias, essas informações foram analisadas para avaliar a facilidade de uso, a satisfação do usuário.

Utilizamos uma estrutura precisa para abordar esse tema, como trabalhos relacionados, nos quais pesquisamos projetos que pudessem auxiliar no desenvolvimento da aplicação web. Outro tópico de extrema importância foi a fundamentação teórica, a qual embasou nossa pesquisa. Na quarta seção, realizamos análises de estudos de caso como estratégia de pesquisa mais detalhada. Na seção de Resultados e Discussões, apresentaremos os resultados concretos obtidos com o desenvolvimento, destacando a integração bem-sucedida de *front-end* e *back-end*. Nas Considerações Finais, enfatizaremos o impacto da nossa solução tecnológica no setor de turismo regional e as perspectivas para futuros desenvolvimentos. Por fim, na seção de Referências, apresentaremos uma lista bibliográfica que serviu de base para nossa pesquisa e o desenvolvimento do trabalho.

2. TRABALHOS RELACIONADOS

Os projetos mencionados compartilham semelhanças, os projetos abordam áreas específicas, como compra e venda de segunda mão, organização de eventos e criação de catálogos de produtos e todos estão focados no desenvolvimento de plataformas web ou mobile para aprimorar a experiência em diferentes contextos relacionados a viagens e turismo. Para exemplificar e enriquecer a nossa abordagem, esta seção examina projetos semelhantes focados em melhorar a experiência do utilizador na descoberta e na interação. As seções a seguir descreverão os seguintes artigos estudados: “2.1 Desenvolvimento de uma aplicação web, com Front-end e Back-end, para compra e venda de segunda mão.”, no item “2.2 Sistema Web e Mobile para eventos.”, no item “2.3-Desenvolvimento de uma Plataforma Web para Catálogo de Produtos e Realização de Pedidos”, E por fim “2.4

Diferença entre trabalhos.”

2.1 Desenvolvimento de uma aplicação web, com Front-end e Back-end, para compra e venda de segunda mão

O trabalho foi desenvolvido por Matheus Medeiros Gentil e Sarah Izabel Pfaffenzeller Esquivel, o site proposto visa fornecer uma plataforma semelhante ao “Wallapop”. Aplicação mobile com possibilidade de comprar, vender e trocar artigos novos e usados com grande diferenciação geográfica. Os usuários poderão pesquisar produtos com base em sua localização, tornando a experiência de compra e venda mais conveniente e direcionada. O site permitirá que os usuários se cadastrem gratuitamente e comecem a listar e exibir seus produtos. O projeto se concentra na criação de serviços *back-end* que suportam esses processos, bem como no desenvolvimento de diferentes visualizações front-end e na criação de interfaces gráficas fáceis de usar e intuitivas.

A página web baseado no aplicativo móvel oferece vantagem significativa, como facilidade geral de uso e menores custos de desenvolvimento e manutenção, mas esta abordagem também traz desafios, como limitações na implementação de recursos avançados e dependência constante da conectividade com a Internet. Para futuras melhorias nesta plataforma, transformá-la em um *Progressive Web App* (PWA) é uma estratégia promissora, pois combina adaptabilidade web e recursos nativos de aplicativos móveis (incluindo recursos *offline* e notificações *push*).

2.2 Sistema Web e Mobile para eventos

O trabalho foi desenvolvido por Kaue Ribeiro, o principal objetivo do site proposto neste projeto é trazer uma melhor experiência do cliente em eventos como shows e bares, fornecendo soluções inovadoras para os problemas comuns de filas e esperas. Através desta plataforma, os clientes podem fazer o pré-registo, comprar bilhetes e aceder à ementa do restaurante a partir do telefone. Isso elimina a necessidade de fazer fila, pedir e pagar bebidas ou alimentos na chegada. Por meio do aplicativo, o cliente é avisado quando seu pedido estiver pronto e o sistema de código agiliza o processo de retirada do produto sem precisar esperar na fila tradicional.

O projeto visa melhorar a experiência do evento por meio do aplicativo e do site, trazendo benefícios como eliminação de listas de espera, aumento da satisfação dos clientes e pedidos digitais. No entanto, os desafios incluem a necessidade de adquirir tecnologia e formar pessoal. Para melhorias futuras, a integração do pagamento móvel, o feedback do usuário, a funcionalidade expandida e a acessibilidade são considerações importantes. Em suma, o plano visa melhorar a participação, enfrentar desafios e promover o desenvolvimento contínuo para satisfazer as necessidades dos usuários.

2.3 Desenvolvimento de uma Plataforma Web para Catálogo de Produtos e Realização de Pedidos

O trabalho foi desenvolvido por Karen Schmidt Lotthammer, Josiel Pereira, Juarez Bento da Silva e João Bosco da Mota Alves, o principal objetivo do site proposto neste enquadramento é oferecer uma plataforma versátil e flexível para a criação de catálogos de



produtos para diferentes áreas de negócio. A plataforma permitirá que empresas e empreendedores criem categorias e listas de produtos e as disponibilizem para potenciais clientes. Além disso, os clientes poderão visualizar esses produtos, tirar dúvidas e tirar dúvidas diretamente pela plataforma, o que simplificará o processo de compra e tornará mais eficiente a interação entre vendedores e consumidores.

A plataforma oferece muitos benefícios, incluindo apresentação simplificada de produtos, acesso mais fácil aos clientes e maior visibilidade do negócio. Também ajuda a acelerar as vendas e torna o processo de compra mais eficiente. No entanto, algumas desvantagens podem incluir problemas de segurança e gerenciamento de dados, bem como a necessidade de monitorar constantemente as atualizações do produto. Para melhorias futuras, considere implementar recursos de análise de dados para ajudar a tomar decisões de negócios e continuar a focar na segurança para proteger informações confidenciais dos clientes.

2.4 Diferença do projeto desenvolvido

O artigo sobre o desenvolvimento de uma aplicação web para compra e venda de produtos usados concentra-se na criação de uma plataforma de geocomércio para produtos usados. Por outro lado, o Matrip visa melhorar a experiência de registo e marcação de visitas e restaurantes. Embora ambos os projetos possam se beneficiar da mudança para *Progressive Web Apps* (PWA) para fornecer uma experiência mais responsiva, as áreas de aplicação são muito diferentes.

O artigo sobre sistema web e mobile para eventos tem como foco otimizar a experiência do usuário em eventos, eliminando filas e esperas com um aplicativo. Por outro lado, este projeto visa melhorar a gestão de passeios e restaurantes e simplificar o processo de registo e reserva de serviços. Ambos os projetos visam aumentar a satisfação dos clientes, mas as condições operacionais e os desafios associados são diferentes. As melhorias futuras centrar-se em contextos específicos, como a integração e a acessibilidade dos pagamentos móveis.

O terceiro artigo sobre o desenvolvimento de uma plataforma de catálogo de produtos online tem semelhanças com o Matrip, que visa criar uma plataforma para exibir produtos e facilitar a comunicação empresarial com um cliente. Embora o objetivo do artigo seja criar listagens gerais de produtos, este projeto foca em inscrições para passeios e restaurantes. Ambos os projetos têm a vantagem de simplificar a apresentação dos produtos e melhorar a comunicação entre vendedores e consumidores, mas desafios e considerações relacionadas ao seu contexto de utilização, como segurança da informação e atualizações relacionadas aos produtos.

Cada projeto aborda desafios específicos e oportunidades únicas de melhoria, enfatizando a importância da adaptabilidade e personalização na criação de soluções tecnológicas eficazes. Outras melhorias serão baseadas nas necessidades do público-alvo, o que aumentará a satisfação do usuário e a eficiência operacional, refletindo a capacidade da tecnologia de impulsionar a inovação em vários setores.

3. FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de sistemas Web envolve a integração de *back-end* e *front-end*. Para apoiar o desenvolvimento de sistemas Web, é crucial uma base teórica sólida. Os con-

ceitos abordados neste artigo foram cuidadosamente selecionados para explorar soluções contemporâneas amplamente adotadas em cenários de desenvolvimento de sistemas web. Esta base de conhecimento permitirá a criação de um sistema eficiente, interativo e responsivo que atenda às expectativas dos usuários. Esta fundamentação teórica servirá como um guia estratégico para a criação de um sistema web altamente responsivo, interativo e eficiente. As seções a seguir descreverão os seguintes tópicos estudados: “3.1 Metodologia Ágil”, no item “3.2 UML”, no item “3.2.1 Caso de Uso”, no item “3.2.2 Diagramas de Classes”, no item “3.3 Spring Framework”, no item “3.3.1 Spring Boot”, no item “3.3.2 Dependências”, no item “3.3 Front End”, por fim item “3.3.1 Bootstrap”.

3.1 Metodologia Ágil (Scrum)

A metodologia Ágil *Scrum*, caracterizada por princípios como transparência, colaboração e entrega progressiva, oferece uma forma eficaz em desenvolvimento de sistemas. Sua implementação requer a formação de uma equipe multidisciplinar, a nomeação de um *Product Owner* para gerenciar o backlog e a divisão do projeto em múltiplas fases chamadas Sprints. Durante cada Sprint, a equipe se concentra em entregar produtos selecionados do *Product Backlog*, tornando a qualidade uma prioridade máxima.

Scrums diários, revisões de *sprint* e retrospectivas são práticas importantes para impulsionar a melhoria e garantir a melhoria contínua. Como destacado por Aragon Fernandes (2014), o *Scrum* consiste em método iterativo e incremental para o gerenciamento de projetos complexos, cujo objetivo é garantir agilidade nas entregas e maximizar a aderência aos requisitos dos clientes, a cooperação entre os integrantes da equipe e a produtividade de cada participante. É um dos métodos dos denominados ‘ágeis’ mais difundidos no mercado de TI.

3.2 UML

A Linguagem de Modelagem Unificada (UML) desempenha um papel importante na engenharia de software, fornecendo notação padronizada e métodos visuais para definir, especificar, desenvolver e documentar programas de software complexos. UML é uma ferramenta que ajuda desenvolvedores e equipes a compreender, projetar e implementar programas de *software* de forma clara e precisa. Fornece uma linguagem comum que remove barreiras de comunicação entre os membros da equipe e as partes interessadas, contribuindo para o sucesso dos projetos de desenvolvimento de software. Como enfatizaram Booch, Rumbaugh e Jacobson (2000), “UML é uma linguagem projetada para visualizar, especificar, construir e documentar produtos em programas de software complexos”.

3.2.1 Caso de Uso

Segundo Jacobson (1992), podemos dizer que um caso de uso é um “documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um a Freire processo”. Na engenharia de *software* casos de uso são uma ferramenta importante para compreender e descrever a funcionalidade de um sistema de forma clara e acessível, desempenhando um papel essencial na facilitação da comunicação entre equipes de desenvolvimento, assegurando que as necessidades dos usuários sejam compreendidas. Fornecendo uma representação visual e textual das interações entre os



atores (usuários ou sistemas externos), ajudando a capturar requisitos, cenários e fluxos de trabalho.

Além disso, os casos de uso desempenham um papel importante na fase de teste de software. Eles fornecem cenários específicos que podem ser usados para verificar se o sistema está funcionando conforme o esperado. Isso ajuda a identificar problemas e garantir a qualidade do software. Resumindo, os casos de uso são uma ferramenta versátil que não apenas ajuda a comunicar e documentar requisitos, mas também ajuda a garantir a qualidade e o sucesso dos projetos de engenharia de software.

3.2.2 Diagramas de Classes

Segundo Bezerra (2007), diagramas são uma apresentação de coleções de elementos gráficos que tem um significado predefinido. Os diagramas de classes são parte integrante da linguagem de modelagem UML (Unified Modeling Language) e desempenham um papel importante na engenharia de *software*. Essas representações visuais fornecem uma maneira estruturada de descrever a estrutura de um sistema de *software*, identificando as classes que compõem o sistema, seus atributos, métodos e relacionamentos. Os diagramas de classe são utilizados para a modelagem de sistema, permitindo que equipes de desenvolvimento e *stakeholders* entendam a arquitetura e as interações do sistema.

3.3 Spring Framework

Spring Framework é uma estrutura de desenvolvimento de aplicativos Java conhecida por sua flexibilidade. Ele oferece módulos e recursos, incluindo injeção de dependência, controle de transações, segurança, desenvolvimento web e integração de sistemas. A injeção de dependência é um dos principais conceitos do *Spring*, que pode gerenciar com eficácia as dependências entre os componentes do aplicativo e melhorar a capacidade de manutenção e teste do código. O controle de transações simplifica o gerenciamento de transações e garante a integridade dos dados em operações complexas. Para desenvolvimento web, *Spring MVC* fornece uma estrutura extensível e configurável que facilita a criação de aplicações web dinâmicas (Spring Framework Documentation: Overview, 2023).

3.3.1 Spring Boot

Spring Boot é uma extensão do *framework Spring* e representa um grande avanço no desenvolvimento de aplicações Java. Esta tecnologia simplifica muito a criação, configuração e execução de aplicações, proporcionando uma experiência de desenvolvimento mais eficiente e eficaz. *Spring Boot* é conhecido por suas convenções inteligentes, predefinições e uma grande variedade de módulos que aceleram o processo de desenvolvimento (Spring Boot Project, 2023).

A flexibilidade do Spring Boot permite que os desenvolvedores personalizem configurações quando necessário, mantendo ao mesmo tempo a facilidade de uso. Com todas essas características, o Spring Boot se estabeleceu como uma ferramenta indispensável para o desenvolvimento rápido e eficiente de aplicações Java, atendendo às crescentes demandas do mundo da programação.

3.3.2 Dependências

As dependências desempenham um papel fundamental no desenvolvimento de software, atuando como componentes externos ou bibliotecas que um projeto utiliza para estender sua funcionalidade. Em linguagens de programação como Java, o gerenciamento de dependências é de grande importância para integrar recursos de terceiros, economizar tempo e esforço de desenvolvimento e garantir que os projetos sejam executados de maneira eficaz. Algumas das dependências que foram utilizadas no desenvolvimento desse *back-end*:

- **spring-boot-starter-web** Responsável por fornecer suporte para criação de aplicações web, incluindo funcionalidades como controladores, roteamento e tratamento de solicitações e respostas.
- **spring-boot-starter-test**: Fornece recursos para testar aplicativos *Spring Boot*, incluindo classes de teste e bibliotecas para executar testes de unidade e integração.
- **spring-boot-starter-security**: É usado para adicionar recursos de segurança às aplicações *Spring Boot*, permitindo autenticação e autorização do usuário.
- **org.jetbrains:annotations**: Fornece anotações úteis no desenvolvimento Java, permitindo melhor documentação e expressividade do código..
- **spring-boot-starter-data-jpa**: Essa dependência é utilizada para integração com *Spring Data JPA* e facilita o acesso ao banco de dados através do Mapeamento Relacional de Objetos (ORM).
- **mysql-connector-java**: Responsável por integrar a aplicação ao banco de dados *MySQL*, esta dependência fornece o *driver JDBC* para comunicação com o banco de dados.
- **org.projectlombok:lombok**: O projeto *Lombok* fornece recursos que simplificam o desenvolvimento, como a geração automática de métodos *getter* e *setter*, reduzindo assim reduzindo a quantidade de código repetitivo.

3.3 Frond End

O desenvolvimento *front-end* é a criação de interfaces de usuário em aplicações web e móveis, desempenhando um papel central na apresentação visual e na interatividade. Os desenvolvedores *front-end* convertem elementos de design e funcionalidades em código *HTML*, *CSS* e *JavaScript* com o objetivo de fornecer aos usuários uma experiência envolvente e funcional. As tecnologias e estruturas *front-end* desempenham um papel vital neste processo, ajudando a desenvolver interfaces de utilizador modernas e eficazes que realizam operações críticas e desempenham um papel na usabilidade de websites e aplicações (Kriger, 2023).

Além disso, o desenvolvimento front-end anda de mãos dadas com a melhoria da usabilidade de sites e aplicações. A interface do usuário é a primeira impressão que o usuário tem de um aplicativo, e o desenvolvimento bem-sucedido melhora a satisfação do usuário, melhorando a eficiência e simplificando a navegação. Em última análise, o desenvolvimento *front-end* não se trata apenas do que os usuários veem, mas também de como eles interagem com o aplicativo, tornando-o uma disciplina essencial no desenvolvimento de *software*.

3.3.1 Bootstrap

Bootstrap é um framework de design *front-end* amplamente reconhecido e muito utilizado na indústria de desenvolvimento web. Além disso, o *Bootstrap* oferece uma estrutura consistente e flexível, com uma grande coleção de componentes, estilos e utilitários, simplificando a criação de páginas web que se adaptam perfeitamente a diversos dispositivos e tamanhos de tela, tornando o desenvolvimento de interfaces mais atraentes. De acordo com Manteiga, Sánchez e Urroz (1994), uma das aplicações da metodologia *Bootstrap* consiste em obter intervalos de confiança confiáveis.

4. ESTUDO DE CASO

Para atingir este objetivo, foi adotada uma abordagem de estudo de caso para fornecer uma análise aprofundada da implementação conjunta de *back-end* e *front-end*. Esta abordagem visa garantir que a plataforma seja eficaz e consiga suprir às necessidades do da aplicação Matrip, proporcionando aos utilizadores uma melhor experiência.

A abordagem para o desenvolvimento desta página foi criar uma plataforma web funcional e eficiente, projetada para atender às necessidades dos colaboradores do turismo do estado do Maranhão. O processo é dividido em diferentes etapas, incluindo análise aplicada, levantamento de requisitos, implementação do *back-end* no *front-end*, diagramas de caso de uso, entidade de relacionamento e a seleção das melhores tecnologias a serem utilizadas no desenvolvimento.

4.1 Análise da Aplicação

Durante o desenvolvimento do projeto, a escolha estratégica de incorporar o framework Spring foi fundamental para fortalecer a eficiência da aplicação. O Spring proporcionou uma base sólida, introduzindo conceitos poderosos como inversão de controle e injeção de dependência. Durante a fase de análise, foram definidas as funcionalidades específicas a serem implementadas na tela de cadastro, levando em consideração os requisitos do usuário e as necessidades do sistema.

A análise envolveu um rigoroso processo de levantamento de requisitos para identificar as necessidades dos colaboradores do turismo no estado do Maranhão. Além disso, um diagrama de classes é criado para representar a estrutura das classes e seus relacionamentos. Os casos de uso são definidos para mapear interações entre usuários e o sistema, incluindo operações como criação, edição e exclusão de categorias, tours e outros elementos. Essas etapas são fundamentais para orientar o desenvolvimento da aplicação, garantindo que ela atenda efetivamente aos requisitos e seja construída de forma eficiente.

4.1.1 Diagrama de caso de uso

Um caso de uso não é um passo em uma funcionalidade do sistema. Ao contrário, um caso de uso é um relato fim a fim de um dos usos do sistema por um agente externo (Bezerra, 2007). Esse método é frequentemente utilizado em contextos em que a complexidade e a singularidade de um fenômeno exigem uma abordagem mais aprofundada. Na figura 1 podemos ver o diagrama caso de uso utilizado para o desenvolvimento da página web.

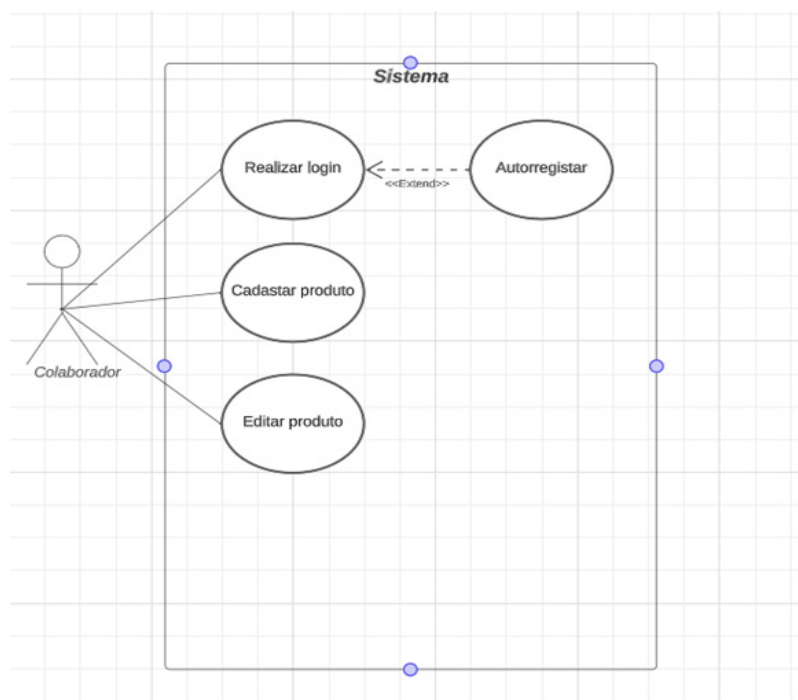


Figura 1. Diagrama de caso de uso

Fonte: Autoral, 2023

Ele oferece a funcionalidade de “Login”, que é a primeira etapa para os colaboradores acessarem suas contas e contribuírem com informações sobre produtos turísticos. Além disso, o “Colaborador” permite “Editar Produtos”, o que é crucial para manter informações precisas e atualizadas na plataforma. Também oferece a capacidade de “Cadastrar Produtos”, facilitando a inclusão de novos itens na plataforma. O “Colaborador” possui uma relação de *extend* com “Realizar Autorregistro”, o que significa que, se um usuário não possui uma conta, ele pode criar uma, tornando o processo de entrada e contribuição mais acessível.

4.1.2 Diagrama de classe

O diagrama de classe, uma ferramenta essencial na engenharia de software, oferece uma visão estruturada e visual das entidades e suas interações em um sistema. No contexto do projeto de cadastro de passeios e restaurantes, o diagrama revela elementos-chave, como a classe “Colaborador”, representando usuários, com métodos claros como Login() e CadastrarProdutos(). A classe “ProdutoTuristico” encapsula informações essenciais sobre os produtos cadastrados, e classes como “CategoriaProduto”, “Cidade” e “País” contribuem para uma organização lógica e geográfica, proporcionando clareza na compreensão e desenvolvimento do sistema.

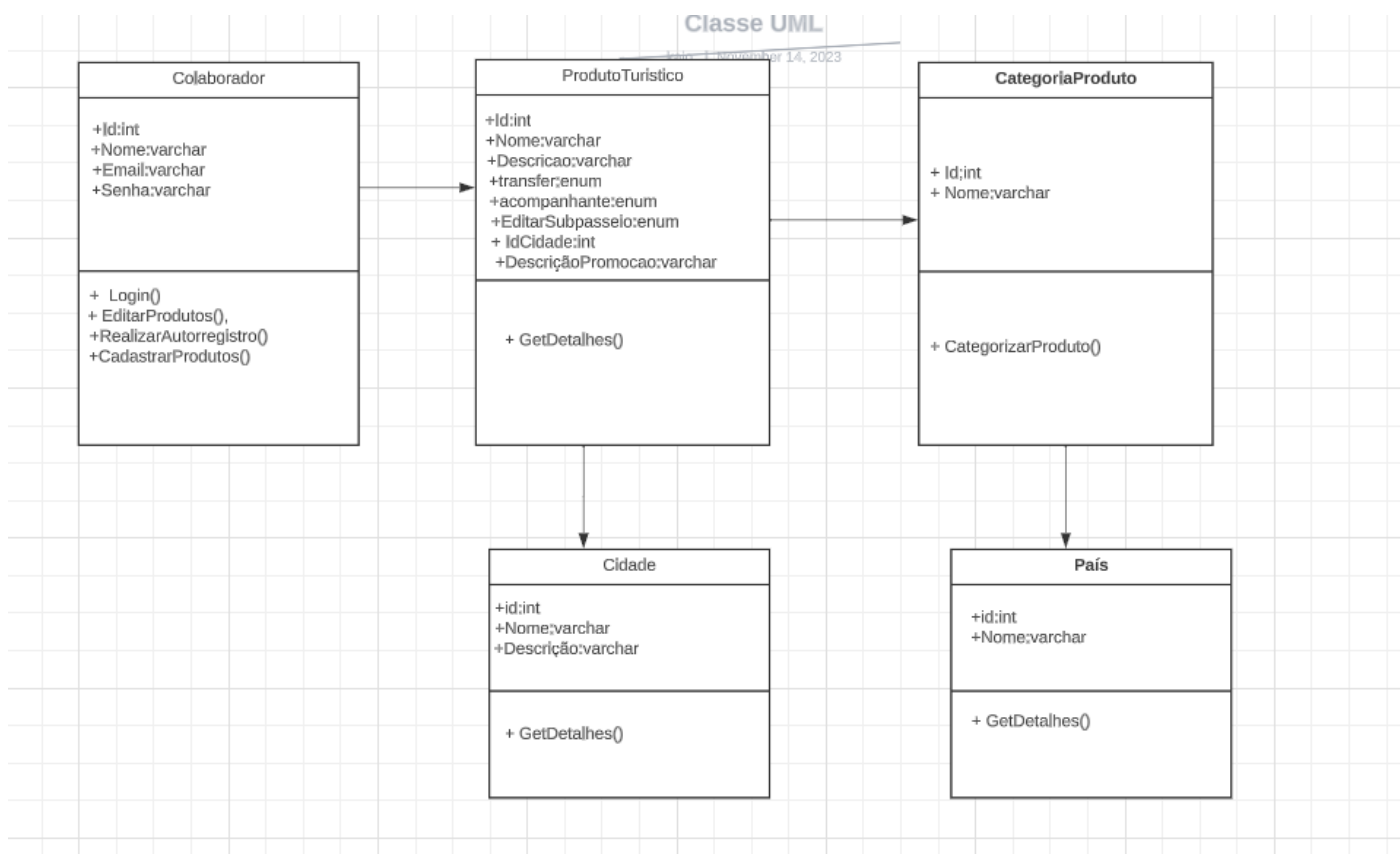


Figura 2. Diagrama de classe.

Fonte: Autoral, 2023

Essa representação visual é impoente para a eficiência e transparência no processo de desenvolvimento de software. O diagrama de classe desempenha um papel fundamental ao modelar as entidades essenciais e suas interações. Observe na figura abaixo o digrama de classe utilizado:

4.3 Levantamento de requisitos

Essa fase de levantamento de requisitos não apenas estabeleceu o rumo do projeto, mas também ajudou a definir os critérios de sucesso para garantir que atendesse às necessidades de seu público-alvo e proporcionasse uma experiência de usuário de alta qualidade dentro do setor de turismo. Neste momento, realizamos uma análise abrangente da situação do turismo no estado do Maranhão, com o objetivo de identificar as necessidades específicas dos turistas e amantes da gastronomia da região. Esse processo envolveu várias estratégias, incluindo pesquisa de mercado, análise de dados demográficos, e a coleta de informações junto a agências de turismo, restaurantes e outros prestadores de serviços locais.

4.4 Ferramentas Utilizadas

No desenvolvimento das páginas web do projeto Matrip, foram aplicadas as ferramentas necessárias para garantir a criação de uma interface atrativa e funcional. *Visual Studio Code* é um editor de código que foi utilizado na criação da estrutura da página em código HTML com eficiência e precisão, fornecendo uma base sólida para interfaces web.

Além disso, o *MySQL*, como sistema gerenciamento de banco de dados, para armazenar e recuperar informações. A API desenvolvida no *IntelliJ IDEA* gerenciar o acesso ao banco de dados *MySQL* para recuperar informações sobre atrações turísticas, restaurantes e reservas de passeios.

4.5 Implementação do Back-End no Front-End

A implementação da API com a página web envolveu a criação de uma interface dinâmica que interage com a API por meio de chamadas assíncronas. A página web apresenta um formulário para cadastro de produtos, onde os dados são enviados para a API por meio de solicitações *POST*. Além disso, a página exibe uma lista de categorias existentes, obtidas por meio de chamadas *GET* à API. Quando o usuário escolhe editar uma categoria, o modal exibe o nome atual da categoria e permite a modificação. A atualização é realizada por meio de uma solicitação *PUT* à API, garantindo que as informações estejam sempre sincronizadas. A Figura 3 apresenta a tela inicial da aplicação web.



The screenshot shows the Matrip web application interface. At the top, there is a dark blue header with the Matrip logo on the left and a user profile on the right showing a globe icon and the text "Olá, Nathalia". Below the header, there is a green button labeled "Novo Produto". The main content area has a dark blue header with the text "Produtos Cadastrados". Below this, there is a table with three columns: "Nome", "Categoria", and "Descrição". Each row in the table has a green "Editar" button on the right side.

Nome	Categoria	Descrição	
Passeio Hilux 4x4	Passeio	Passeio Pelas Dunas Lençois Maranhenses	Editar
Galinhada	Gastronomia	Melhor Comida De Carolina-Ma	Editar
Feira Livre	Artesanato	Feira No Centro Historico De São Luis	Editar

Figura 3. Tela de produtos cadastrados.

Fonte: Autoral, 2023

A tela do *front-end* desenvolvida para colaboradores apresenta diversas funcionalidades essenciais para o gerenciamento de produtos. Inicialmente, oferece a opção “Novo Produto”, que direciona o colaborador para a tela de cadastro, permitindo o registro de novos produtos. Além disso, a tela exibe uma lista dos produtos previamente cadastrados, fornecendo uma visão geral dos itens disponíveis, ao selecionar a opção de editar, o usuário é redirecionado para uma tela onde pode modificar informações específicas de um produto. A Figura 4 exibe a tela destinada ao cadastro de produtos, utilizada pelo colaborador.

Figura 4. Tela de cadastro.

Fonte: Autoral, 2023

A tela de cadastro de produtos permite inserir o nome do produto, incluir uma descrição detalhada para destacar seus atrativos e criar uma frase de promoção envolvente. Além disso, os colaboradores podem escolher se o transporte está incluído com as opções “Sim” ou “Não” e definir a disponibilidade de acompanhantes.

4.6 Conexão com o banco de dados

A configuração da conexão com o banco de dados *MySQL* é essencial para as operações de persistência de dados. As propriedades “spring.datasource.url”, “spring.datasource.username”, e “spring.datasource.driverClassName” especificam, respectivamente, a URL do banco de dados, o nome de usuário, e a classe do driver do *MySQL*. A inclusão de parâmetros adicionais na URL, como desativar a comunicação segura “useSSL=false” e permitir consultas múltiplas em uma única chamada “allowMultiQueries=true”, personaliza a configuração para atender às necessidades específicas. A Figura 5 mostra a tela de conexão com o banco de dados.

```

2  spring.jpa.generate-ddl=true
3  spring.jpa.hibernate-ddl-auto=create
4
5  spring.datasource.url=jdbc:mysql://localhost/db02matrip?useSSL=false&allowMultiQueries=true&serverTimezone=UTC
6  spring.datasource.username=root
7  #spring.datasource.password=#eCsL2212067@
8  spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
9  spring.jpa.show-sql=true
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

```

Figura 5. Conexão com o banco de dados .

Fonte: Autoral, 2023

A inclusão do driver JDBC (spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver) e outras informações essenciais, como a URL e o nome de usuário, completa a

configuração da conexão, garantindo que o Spring Boot interaja com o banco de dados MySQL. Essa abordagem, baseada no Spring Boot e suas convenções de configuração, simplifica a integração de tecnologias e promove um desenvolvimento ágil e eficiente.

4.7 Endpoint

A aplicação utiliza diversos *endpoints* para realizar operações relacionadas à entidade. Esses *endpoints* desempenham funções específicas no ciclo de vida da aplicação, permitindo a interação entre o cliente *front-end* e o servidor *back-end*. A Figura 6 mostra alguns dos *endpoint* utilizados.

```
@DeleteMapping("/{passeioId}")
public ResponseEntity<Passeio> remover(@PathVariable Long passeioId){
    Passeio passeio = passeioRepository.buscar(passeioId);

    try {
        cadastroPasseio.excluir(passeioId);

        return ResponseEntity.noContent().build();
    } catch (EntidadeNaoEncontradaException e){
        return ResponseEntity.notFound().build();
    } catch (EntidadeEmUsoException e){
        return ResponseEntity.status(HttpStatus.CONFLICT).build();
    }
}
```

Figura 6. Endpoint no backend.

Fonte: Autoral, 2023

O *endpoint* de DELETE associado à rota “/passeios/{passeioId}” tem como finalidade remover um passeio específico da base de dados. Ao receber uma solicitação *delete* com o identificador do passeio a ser excluído, a aplicação verifica a existência do passeio com base nesse *id*. Se o passeio for encontrado, ele é removido permanentemente da base de dados, resultando em uma resposta com o código HTTP 204 (No Content), indicando o sucesso da operação. Em caso de tentativa de exclusão de um passeio não existente, a resposta será 404 (Not Found). Adicionalmente, se o passeio estiver sendo referenciado por outras entidades ou em uso, a resposta será 409 (Conflict), indicando uma violação de integridade referencial. A Figura 5 exibe os *endpoints* sendo utilizado no *front-end*.

```
function apagarPasseio(id) {
    api.delete(`/passeios/${id}`)
        .then(() => {
            listarPasseios(); // Atualiza a lista após a exclusão
        })
        .catch(error => {
            console.error(error);
        });
}
```

Figura 7. Endpoint no frontend.

Fonte: Autoral, 2023

Essa função está associada ao *endpoint* de *delete*, o qual é acionado quando um usuário decide remover um passeio específico. No contexto da aplicação, o *endpoint* está vinculado à rota “/passeios/{id}”, onde {id} representa o identificador único do passeio a ser excluído.

5. RESULTADOS E DISCUSSÃO

O projeto obteve sucesso ao desenvolver uma plataforma que integra a criação de produtos de viagens e turismo através de uma aplicação web. A combinação das tecnologias de *back-end* em *Spring Boot* e tecnologias de *front-end* em *HTML* permite o gerenciamento eficiente do conteúdo de viagens. A implementação de um diagrama entidade-relacionamento (DER) organiza os dados de forma estruturada, simplificando a categorização do produto. Porém, manter a precisão dos dados e validar as entradas continuam sendo desafios importantes, portanto, a metodologia de pesquisa bibliográfica adotada neste projeto desempenha um papel fundamental na identificação e resolução contínua dessas questões, garantindo a eficácia da plataforma.

O uso da metodologia de pesquisa experimental permitiu uma análise aprofundada dos resultados obtidos durante testes controlados na plataforma. Diversos experimentos foram conduzidos, visando avaliar o impacto de alterações específicas na eficiência e na experiência do usuário. Em suma, esse projeto demonstra a importância da tecnologia na promoção do turismo e na capacidade de colaboradores gerenciarem eficazmente produtos turísticos, contribuindo para o crescimento econômico regional e a satisfação dos viajantes.

6. CONSIDERAÇÕES FINAIS

O projeto é um exemplo de como a tecnologia está a capacitar a indústria das viagens, capacitando os funcionários para gerir eficazmente os produtos de viagem, mostrar destinos e melhorar a experiência do viajante. Combinação de tecnologia A integração de ferramentas como *IntelliJ IDEA*, *MySQL* e *Visual Studio Code* é fundamental para o desenvolvimento e comunicação *front-end* e *back-end*. Apesar do sucesso do projeto, desafios como a manutenção da precisão dos dados e a validação dos insumos requerem atenção contínua. Mas a flexibilidade da plataforma em termos de funções editoriais, de promoção de viagens e de gestão de produtos de viagens proporciona um ambiente versátil para funcionários e gestores.

No geral, o projeto destaca a importância da tecnologia na promoção do turismo e como soluções inovadoras podem impulsionar as economias locais, atraindo turistas e satisfazendo as suas necessidades. Porém ainda existem melhorias que podem ser aplicadas no futuro, uma das principais áreas de melhoria é a experiência do usuário, que poderia investir em uma interface mais amigável e atrativa para tornar a navegação fácil e agradável, a implementação de eventos com a possibilidade de compra de ingressos pela aplicação pode ser uma extensão lógica para atender a demanda de turistas e colaboradores que desejam promover e participar de eventos locais, como festas, shows, excursões temáticas e outras atividades relacionadas ao turismo. Reconhecemos que há espaço para melhorias, especialmente na experiência do usuário, que se beneficiaria de uma interface mais amigável e atrativa, visando tornar a navegação fácil e agradável. Uma proposta para o futuro envolve a implementação de eventos com a capacidade de compra de ingressos através da aplicação, oferecendo uma extensão lógica para atender às demandas de turistas e colaboradores interessados em promover e participar de eventos locais, como festas, shows e excursões temáticas.

REFERÊNCIAS

BEZERRA, Eduardo. **Princípios de análise e Projeto de sistemas com UML**. 2. ed., Rio de Janeiro: Elsevier, 2007.

Booch, G., Jacobson, I., & Rumbaugh, J. (1999). **The Unified Modeling Language User Guide**. Addison-Wesley.

Cohn, M. (2004). **User Stories Applied: For Agile Software Development**. Addison-Wesley.

FERNANDES, A. A.; ABREU, V.F.: **Implantando a governança de TI**, 4ª Ed.- São Paulo, SP: Editora Brasport Livros e Multimídia Ltda, 2014.

FRONT END: ENTENDA O QUE É, PARA QUE SERVE, COMO APRENDER ESSA ESPECIALIDADE!. 12 jun. 2023. Disponível em: <https://kenzie.com.br/blog/front-end/>. Acesso em: outubro de 2023.

Honey, M. (2008). **Ecotourism and Sustainable Development: Who Owns Paradise?** Island Press.

JACOBSON, Ivar. **Object Oriented Software Engineering: A Use Case Driven Approach**. Addison Wesley, 1992.

Schwaber, K., & Sutherland, J. (2017). **The Scrum Guide**. Scrum.Org. [Guia oficial do Scrum].

Silva, J. G. (2013). **Ecoturismo: Natureza, Sociedade e Desenvolvimento**. Annablume.

Spring Framework Overview: Spring Framework. Disponível em: <https://docs.spring.io/spring-framework/reference/overview.html>. Acesso em: outubro de 2023.

TURISMO NAS CIDADES: IMPACTOS POSITIVOS E NEGATIVOS PARA A ECONOMIA E A SOCIEDADE. 12 jun. 2023. Disponível em: <https://anafisco.org.br/?s=TURISMO>. Acesso em: outubro de 2023.





PLATAFORMA WEB DE INSERÇÃO DE QUESTÕES E GRUPOS NO LUDO EDUCACIONAL

8

*Web Platform for insertion questions and groups in LUDO
Educacional*

Kelve Antonio Oliveira Lima
Edilson Carlos Silva Lima
Yonara Costa Magalhães

D.O.I.: 10.29327/5391075.1-8



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

O problema da queda de interesse dos jovens tem tornado o ensino nas instituições educacionais cada vez mais desafiador, com foco maior nas escolas públicas devido às suas formas limitadas de ensino. Os alunos, no início de sua vida acadêmica, encontram-se desestimulados (JAPIASSU *et al.*, 2020). Tendo em vista os problemas da queda de interesse, foi realizado este projeto, que faz uso das metodologias de Estudo de Caso; Pesquisa de Ação, de cunho Qualitativo. Este artigo trata de uma parte do projeto: uma plataforma web para o cadastro de questões e grupos pelos docentes; questões essas que serão acessadas pelos alunos em uma plataforma web de um jogo semelhante ao LUDO™. Ao invés de jogar os dados, os participantes devem responder a uma pergunta, e suas respostas serão responsáveis por mover as peças de quem acertar, avançando no jogo. Essa abordagem gamificada do ensino serve como um alicerce para novas dinâmicas de aprendizado (ARAUJO *et al.*, 2022). Os resultados iniciais indicam um aumento na motivação e entusiasmo dos usuários.

Palavras-chave: Gamificação, React, Frontend.

Abstract

The issue of declining interest among young people has made education in educational institutions increasingly challenging, with a greater focus on public schools due to their limited teaching methods. Students, at the beginning of their academic journey, find themselves demotivated (JAPIASSU *et al.*, 2020). In light of the problems associated with this decline in interest, this project was undertaken, utilizing the methodologies of Case Study and Action Research, with a qualitative approach. This article addresses a portion of the project: a web platform for the registration of questions and groups by teachers; these questions will be accessed by students on a web platform for a game similar to LUDO™. Instead of rolling dice, participants must answer a question, and their answers will be responsible for moving the pieces for those who answer correctly, advancing in the game. This gamified approach to teaching serves as a foundation for new learning dynamics (ARAUJO *et al.*, 2022). Initial results indicate an increase in motivation and enthusiasm among users.

Keywords: Gamification, React, Frontend.



1. INTRODUÇÃO

Esse trabalho se trata de uma abordagem sobre a queda no interesse dos jovens no ensino, este declínio tem gerado um problema substancial nas instituições educacionais, especialmente nas públicas, como destacado por (JAPIASSU *et al.*, 2020) em sua publicação. E também, como figurado por (ARAUJO *et al.*, 2022), os jogos são uma ótima ferramenta para a melhora do aprendizado, por conta de sua abordagem mais dinâmica de temas e conteúdo, aumentando o interesse e o foco dos alunos.

Este trabalho se trata de uma plataforma gamificada, para isto, esse artigo se vale das seguintes metodologias: Pesquisa em Ação, onde será realizado um questionário sobre questões pertinentes no tocante as questões em que o problema fora pautado; Estudo de Caso, pois o artigo se vale de um enfoque nesta questão do desestímulo dos alunos com o ensino tradicional; e de cunho Qualitativo, pelo enfoque deste artigo em uma plataforma de cadastro de professores formuladores, questões e sessão de ingressão na aplicação.

A aplicação desenvolvida neste artigo refere-se a uma plataforma web onde os professores e docentes no geral, poderão adicionar questões e iniciar sessões, para a partir do seu login e da adição de questões, possa ir para uma sala na aplicação web onde será executada uma partida similar a LUDO™, mas, ao invés de rolar dados, os jogadores terão de responder questões e dependendo do quão certo foi sua resposta, move-se no tabuleiro, desse modo visamos o aumento no foco e no aprendizado dos alunos envolvidos nesta dinâmica.

Neste trabalho, abordaremos seções essenciais da pesquisa. Começaremos com uma análise dos Trabalhos Relacionados para contextualizar o campo. Em seguida, exploraremos o Referencial Teórico para compreender os métodos utilizados. O Projeto será detalhado, destacando os procedimentos adotados. Resultados e Discussão apresentarão insights da aplicação de IA na análise de dados empresariais. Na Conclusão, consolidaremos achados e apontaremos futuras direções. A seção de Referências fornecerá fontes utilizadas, assegurando transparência e validação dos dados.

2. TRABALHOS RELACIONADOS

Os “Trabalhos Relacionados” são estudos anteriores sobre o mesmo tema. Eles ajudam a situar o novo estudo, fornecendo credibilidade, identificando lacunas na pesquisa e evitando repetições. Também servem como fonte de referências para os leitores. Resumidamente, são importantes para entender o contexto e a base do novo estudo. No tópico 2.1 temos o primeiro trabalho relacionado e dentro desse tópico tendo um parágrafo para explicar a problematização do trabalho e um segundo para a sua solução, igual no 2.2 e 2.3, mas obviamente com trabalhos relacionados diferentes, e no 2.4 o diferencial do meu trabalho para estes relacionados.

2.1 Primeiro Trabalho Relacionado

Serviço Web para apoiar a recuperação de objetos de aprendizagem em repositórios e federações; feito por Yeison Aguirre Osorio, Juan Sebastian Espinosa Trejos, Valentina Tabares Morales, Nestor Dario Duque Mendez; no ano de 2016, exposto e publicado na XI Conferência Latino-Americana sobre Objetos de Aprendizagem e Tecnologia (LACLO), 1-6,

no ano de 2016.

Este trabalho problematizou a necessidade de garantir uma experiência adequada aos usuários ao acessarem Objetos de Aprendizagem (AO) por meio de repositórios e federações. O desafio estava na gestão eficiente desses recursos, especialmente no que diz respeito aos metadados, como a descrição do conteúdo e a conformidade com padrões.

A solução proposta foi um serviço web para a avaliação automática dos metadados de AO. Esse serviço classifica os resultados de uma busca de acordo com o nível de qualidade dos metadados, permitindo que os usuários recebam OAs que atendam a requisitos específicos e exibam como prioridade aqueles com melhores condições.

2.2 Segundo Trabalho Relacionado

MeRLÍ, uso de padrões para etiquetar recursos educativos, feito por Aleix Canals Pou, em 2020, publicado em Grup de Desenvolupament i Programari Lliure - Departament de Educació de la Generalitat de Catalunya.

Este trabalho problematizou a necessidade de permitir aos usuários localizar rapidamente recursos educacionais online que atendam às suas necessidades, garantindo a qualidade do catálogo e a adequada descrição dos elementos.

A solução proposta foi o sistema MeRLÍ (Online Resource Metadata), que utiliza a especificação do padrão LOM, um tesouro e uma ontologia do currículo educacional para descrever e classificar os recursos. O ciclo de vida dos recursos no MeRLÍ assegura a qualidade educacional mínima e a adequação dos metadados. Além disso, oferece três formas de acesso: um buscador de recursos, um gerenciador de conteúdo e um Web Service, permitindo que os usuários escolham a opção mais conveniente.

2.3 Terceiro Trabalho Relacionado

Um serviço Werb para gamificação, feito por: José Luiz Vilas Boas, Murilo Augusto Lopes Teixeira, Jacques Duílio Brancher, Armando Maciel Toda, Eduardo Filgueiras Damasceno; proposto na XVIII Simpósio Internacional de Informática Educativa, SIEE, Salamanca, 123-128, do ano de 2016.

Este trabalho problematiza o desafio de implementar e manter a gamificação dentro de um aplicativo, especialmente em relação à conexão entre o conteúdo visualizado e a base de dados, considerando o consumo significativo de espaço, o que pode afetar a velocidade do sistema com o aumento do número de usuários.

A solução proposta é um Web Service que permite aos usuários acessar conquistas, pontos e informações em uma base de dados sem a necessidade de armazenar em seu próprio ambiente. Além disso, um sistema de página de administrador oferece informações das comunidades e usuários, visando monitorar e gerenciar as ações dos indivíduos para melhorar o processo de ensino-aprendizagem de maneira agradável e imersiva.

2.4 Diferencial do Trabalho

A diferença principal entre o trabalho, reside na abordagem específica para lidar com o problema da queda de interesse dos jovens no ensino. Enquanto os trabalhos anteriores

se concentram em questões como a eficiência na localização de recursos educacionais (MeRLÍ) ou na implementação de gamificação em aplicativos educacionais (Web Service para Gamificação), o meu projeto utiliza metodologias como Estudo de Caso e Pesquisa de Ação, de cunho qualitativo.

Tem o seu foco principal no desenvolvimento de uma plataforma web de cadastro de professores e questões, aplicando gamificação como estratégia para enfrentar a desmotivação dos alunos, especialmente nas instituições de ensino público. A ênfase na dinâmica do ensino por meio de jogos é destacada como uma abordagem única para cativar a atenção dos alunos, essa proposta visa enfrentar diretamente o desafio específico do desestímulo dos estudantes, buscando proporcionar uma experiência de aprendizado mais envolvente e estimulante.

3. FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica em um artigo é a parte onde o autor explica as teorias e pesquisas já existentes sobre o tema. É importante porque situa o estudo no contexto certo, mostra que o autor conhece bem o assunto e ajuda na escolha dos métodos de pesquisa. No 3.1 tratou-se sobre UML, com o diagrama de classes e o diagrama de casos de uso, no 3.2 tratou-se sobre a Metodologia Ágil Scrum, no 3.3 tratou-se da API, no 3.4 do Framework Materialize, no 3.5 no React/Next.js e no 3.6 na Interação Homem-Máquina.

3.1 UML (Unified Modeling Language)

UML, ou Linguagem de Modelagem Unificada (do inglês “Unified Modeling Language”), é uma linguagem gráfica padronizada utilizada na engenharia de software para representar, visualizar, projetar e documentar sistemas complexos. Ela oferece um conjunto de diagramas e notações que permitem aos desenvolvedores e equipes de projeto descrever de forma clara e precisa a estrutura, o comportamento e as interações de um sistema de software.

Como dito por (GARDNER; TRACY, 2003, p30), “A Linguagem de Modelagem Unificada (UML) é amplamente utilizada no desenvolvimento de software orientado a objetos e também foi usada, com personalizações, para o desenvolvimento de software baseado em componentes, modelagem de processos de negócios e design de sistemas[...]”.

3.1.1 Diagramas de Casos de Uso

Os diagramas de caso de uso visualizam interações entre usuários e um sistema, identificando casos de uso (ações) e atores (usuários externos). Essa representação gráfica simplifica a compreensão de como os usuários interagem com o sistema, destacando funcionalidades principais e cenários de uso, facilitando a comunicação de requisitos entre desenvolvedores e stakeholders.

3.1.2 Diagrama de Classes

Os diagramas de classe modelam a estrutura estática de um sistema, representando classes, atributos e relacionamentos entre objetos. Essa visualização facilita a compreen-

são da organização interna do software, proporcionando uma base para o design orientado a objetos. Esses diagramas são cruciais para a implementação, manutenção e expansão eficientes do sistema.

3.2 Metodologia Ágil

A metodologia ágil é uma abordagem no desenvolvimento de software que se baseia na colaboração, adaptabilidade e entrega incremental. Ela foca em equipes auto-organizadas que trabalham em ciclos curtos chamados “iterações” ou “sprints”. Durante cada iteração, um conjunto de funcionalidades é desenvolvido, testado e entregue. O cliente e a equipe revisam e ajustam os objetivos conforme o projeto avança (SCHWABER; SUTHERLAND, 2017).

3.2.1 Scrum

O Scrum é uma forma de trabalhar em projetos de forma mais organizada e eficiente. Nele, há papéis como o dono do produto, o líder do time e os desenvolvedores. O trabalho é dividido em “sprints”, períodos curtos de tempo para entregar partes do projeto. Há reuniões diárias para manter todos atualizados. Ao fim de cada sprint, é entregue uma parte do projeto pronta para uso.

Como dito por (SCHWABER; SUTHERLAND, 2017) “Scrum, por sua vez, tem em sua origem o caráter de desenvolvimento de softwares, mas a partir dos anos 90 seu uso foi ampliado mundialmente e começou a ser utilizado para pesquisas e identificações de mercados viáveis, tecnologias e funcionalidades de produtos[...]”.

3.3 API (Application Programming Interface)

Uma API, ou Interface de Programação de Aplicativos, é um conjunto de regras e protocolos que permite que diferentes softwares se comuniquem e interajam entre si. Ela atua como um intermediário, permitindo que uma aplicação solicite e receba informações e também outra essa está facilitando a integração de funcionalidades e o compartilhamento de dados entre sistemas distintos.

De acordo com (ERLE; GIBSON, 2006), “Uma API define uma maneira padrão para um programa para chamar o código contido em outro aplicativo ou biblioteca. A API do Google define um conjunto de objetos JavaScript e métodos que podem ser usados para colocar mapas em páginas Web”.

3.4 Frameworkizek Materialize

Materialize é um framework front-end de código aberto que facilita a criação de interfaces de usuário atraentes e responsivas para aplicações web. O Materialize fornece uma variedade de componentes prontos para uso, como botões, barras de navegação, formulários e muito mais, facilitando o processo de desenvolvimento.

De acordo com Anirudh Prabu e Aravind Shenoy (2016), “Materialize é um framework semelhante ao Bootstrap e ao Foundation, que apresenta componentes de interface do



usuário. No entanto, ele possui funções diferentes, pois o Bootstrap e o Foundation são baseados em dispositivos móveis, enquanto o Materialize segue o Material Design, originado pela Google.”

3.5 React e Next.js

React é a biblioteca JavaScript amplamente utilizada para construir interfaces de usuário interativas, ela permite a criação de componentes reutilizáveis que representam partes específicas da interface. O React utiliza uma abordagem declarativa, facilitando a atualização eficiente da UI quando o estado dos dados muda. React é uma biblioteca JavaScript para a construção de interfaces de usuário (UIs) reativas e dinâmicas. Ele se destaca pela composição de componentes, que facilita a organização e reutilização de código. Ele é especialmente útil para a construção de aplicações web escaláveis e com bom desempenho. Em conjunto, React e Next.js formam uma combinação poderosa para o desenvolvimento de aplicações web modernas e eficientes.

Como dito por Huy e Truong (2020), “O desenvolvimento web em JavaScript se tornou o novo padrão na indústria de tecnologia, sendo que a maioria dos trabalhos exige conhecer React, Node.js[...]”.

Já o Next.js é um framework React de código aberto que simplifica o desenvolvimento de aplicativos web modernos. Ele oferece renderização tanto no lado do servidor quanto no lado do cliente, facilitando a criação de páginas estáticas e dinâmicas. Next.js também inclui recursos como pré-renderização, roteamento automático e suporte para APIs, agilizando o desenvolvimento de aplicações web escaláveis e otimizadas.

3.6 Interação Homem-Máquina

A interação homem-máquina (IHM) desempenha um papel crucial no sucesso de qualquer aplicativo educativo, especialmente quando se trata de gamificação. A experiência do usuário, a usabilidade e a capacidade de envolver os estudantes são fatores críticos. A IHM envolve a criação de interfaces intuitivas e agradáveis que permitam aos usuários interagir facilmente com o sistema, tornando a aprendizagem mais envolvente e eficaz.

Segundo Moraes (2001) IHM são sistemas supervisórios que surgiram da necessidade de uma interface amigável, eficiente e ergonômica entre sistemas de automação complexos e a equipe encarregada da sua operação. Devem, portanto, ser construídas tendo os operadores como usuário final e representar o processo real.

4. RESULTADOS E DISCUSSÃO

A integração da fundamentação teórica com a prática é o ponto de convergência onde o conhecimento se transforma em ação tangível. Esta etapa essencial do projeto implica na aplicação direta das teorias e conceitos previamente discutidos na seção de fundamentação teórica. No 4.1 está o estudo, 4.2 está a Metodologia Ágil, 4.3 está o diagrama de casos de uso, no 4.4 está o diagrama de classes, e no 4.5 está o React, e dentro do React expliquei as telas mais importantes usadas na aplicação do projeto.

4.1 Estudo de Caso

Tivemos como estudo de caso a situação da desmotivação dos alunos no ensino público e privado, principalmente no ensino público, onde os estudantes não possuem outros métodos de ensino, têm acesso apenas ao tradicional e engessado do ensino padronizado, como notado pelos estudos citados que me levou a pensar neste projeto, com uma abordagem a este método de ensino, foram desenvolvidos, pensando nesta lacuna, uma aplicação gamificada para fortalecer a motivação e o empenho dos alunos.

A aplicação gamificada proposta incorpora elementos lúdicos ao ambiente educacional, transformando a aprendizagem em uma experiência mais interativa e atrativa. Ao abordar de maneira específica as limitações enfrentadas pelos alunos no ensino público, o projeto busca democratizar o acesso a métodos de ensino mais dinâmicos e personalizados.

A gamificação não apenas oferece uma abordagem inovadora, mas também representa uma oportunidade de cativar a atenção dos alunos de uma forma que vai além das abordagens comuns e corriqueiras. Este projeto não se limita a suprir a falta de motivação, ela tem foco na inovação do ensino. Não apenas temos foco na resolução de desmotivação, mas sim em uma revolução do ensino usando essas abordagens.

4.2 Metodologia Ágil Scrum

A metodologia Ágil Scrum foi adotada equipe através de uma reunião em que foram definidos também como seria solucionado o problema e ordenação, escolhemos está por conta de sua adaptabilidade, por ter uma abordagem flexível e adaptativa ao desenvolvimento de projetos, o que significa que a equipe pode responder rapidamente a mudanças nas necessidades e no mercado.

4.3 Diagrama de Casos de Uso

A figura 1 apresenta o Diagrama de Casos de Uso se vale da compreensão de como os usuários interagem com o sistema, destacando funcionalidades principais de cada componente de cada cenário de uso, e também facilitando a comunicação de requisitos para os desenvolvedores:

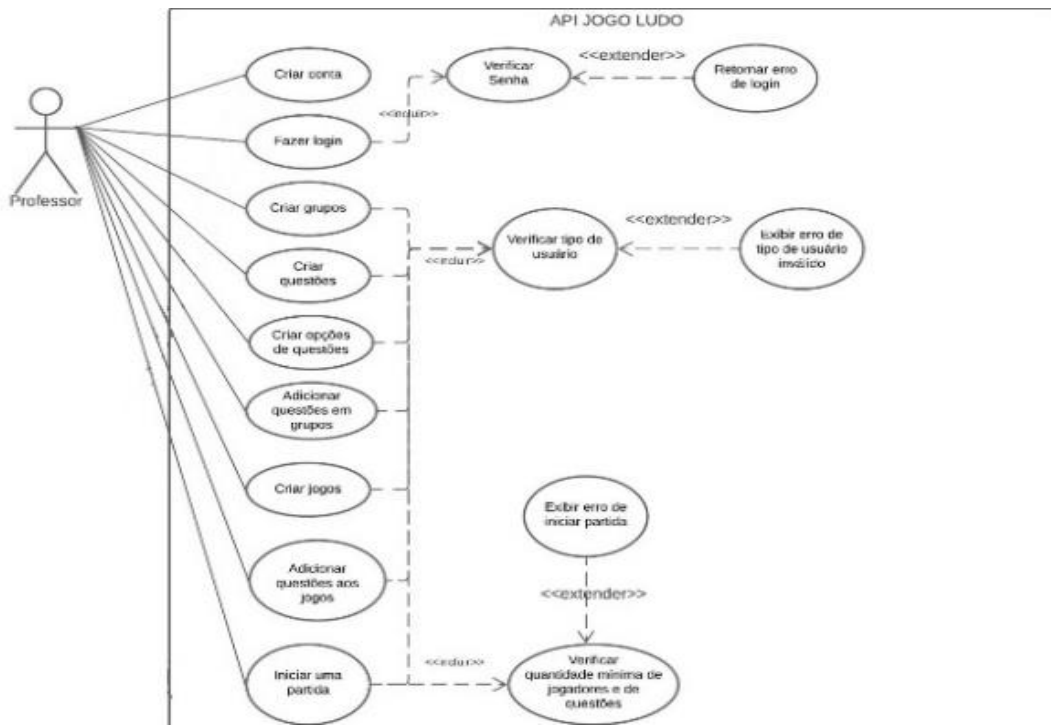


Figura 1. Diagrama de Caso de uso.

Fonte: Autoral, 2023.

No diagrama acima, está demonstrado todas as funções do professor no projeto como um todo, e todas as ramificações posteriores as suas ações, o professor pode criar uma conta, contra está que é usada posteriormente para fazer o login, seu login passa pelo processo de validação de senha, então após ser validado, o professor tem acesso as demais funções da aplicação como criar grupos, criar questões, cria as alternativas das questão, adicionar estas questões aos grupos criados, criar uma sessão de um jogo, então adiciona-se as questões ao jogo, e por fim, dar início a partida do jogo.

4.4 Diagrama de Classes

A figura 2, o Diagrama de Classes, aqui apresentado, concentram-se na estrutura interna do sistema, descrevendo classes, atributos e relacionamentos. Esses diagramas são fundamentais para o design e a implementação detalhada, proporcionando uma base sólida para a construção do software.

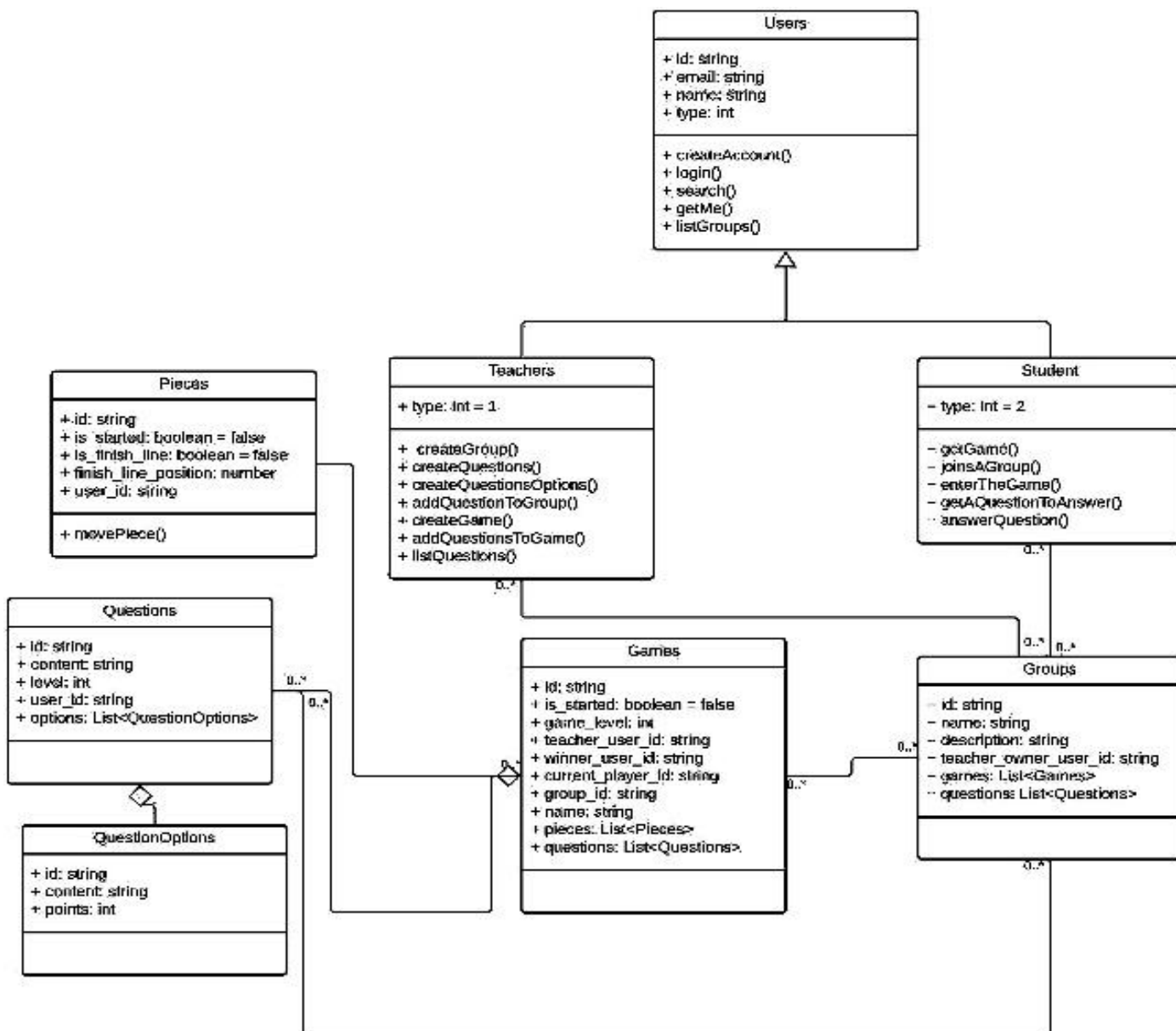


Figura 2. Diagrama de Classes.

Fonte: Autoral, 2023)

No diagrama acima está a descrição de todas as classes e relações entre da base do projeto, como descrito, a entidade Usuário, tem como suas propriedades principais como Id, e-mail, nome e type; esta entidade de Usuário tem relações de herança com a entidade Professor, e com a entidade Estudante, ambas estas entidades de Professor e Estudante herdam da Entidade Grupo.

4.5 Cadastro de Questões

O Cadastro de Questões (figura 3), é uma funcionalidade central no projeto, possibilitando a organização e disponibilização eficiente de material educativo. Os professores podem criar, armazenar e gerenciar questões diversificadas, enquanto os alunos têm acesso a uma variedade de desafios, contribuindo para uma experiência educacional mais rica e personalizada.



Figura 3. Tela de Adição de Questões.

Fonte: Autoral, 2023.

Nesta tela é onde são adicionadas as questões que serão adicionadas aos grupos, como não se possuem os valores de dados como no jogo tradicional do LUDO, é necessário atribuir um valor as questões de 1 a 6 como se fosse um dado, pois como dito, elas fazem esse papel, sua importância para o trabalho é extraordinária, pois é a função de desenvolvimento flexível do projeto.

4.6 Adição de Grupos

A funcionalidade de Criação de Grupos (figura 4), facilita a interação entre alunos, promovendo colaboração e discussões. Os professores podem formar grupos, onde são adicionadas as questões, e através desses grupos, os alunos pode ingressar e então após o início, o jogo então se inicia.



Figura 4. Cadastro de Grupos.

Fonte: Autoral, 2023.

Nesta tela serão adicionados os grupos, tendo como seu campo de cadastro a adição dos nomes dos grupos e logo abaixo uma breve descrição do grupo, nesta descrição é fundamental deixar evidente qual o tipo do grupo para que não haja interações entre grupos de nível alto e com estudantes de nível mais baixo.

4.7 Validação da plataforma

A pesquisa em um artigo é o alicerce do conhecimento acadêmico. Ela não apenas valida as teorias propostas, mas também proporciona uma compreensão mais profunda do tema em questão. Ao oferecer evidências tangíveis, a pesquisa fortalece a credibilidade do artigo e contribui para o avanço contínuo do campo. Além disso, ela cria uma base

sólida para futuras investigações, promovendo um ciclo vital de descoberta e desenvolvimento intelectual.

4.8 Pesquisa-Discursiva

Esta pesquisa foi realizada com 14 participantes, todos estes entrevistados são professores que fazem parte da formação de jovens no ensino de base até a completude do ensino médio, atuam desde o início do Ensino Fundamental II até o 3º Ano do Ensino Médio, foram escolhidos pois foram estes professores que me acompanharam por toda essa jornada acadêmica desta faixa citada; as perguntas foram impressas e entregues no início de Outubro de 2023, porém, por conta de suas atividades escolares, apenas uma fatia de 9 dos 14 responderam a estas perguntas em tempo hábil da confecção deste trabalho.

O método de captação das respostas para a montagem dos gráficos foi simples, após a leitura da resposta discursiva do professor às perguntas, foram analisadas, e após essa análise, as respostas foram classificadas entre 'boa', 'mediana', ou 'péssima', para a construção dos gráficos pizza.

1ª Pergunta: “Qual a sua opinião sobre o nível de entusiasmo e empolgação dos alunos?”

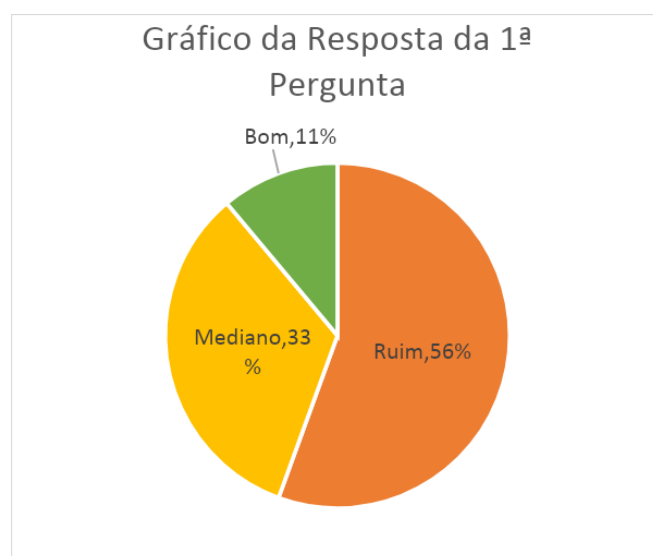


Gráfico 1: 1ª Pergunta.

Fonte: Autoral, 2023.

Como visto no gráfico 1, exposto acima: 56% dos professores consideram que o entusiasmo e a empolgação está *ruim*, as suas respostas para esta pergunta variam em suas causas, mas culminaram em um único fim; 33% dos professores consideram que a empolgação dos seus alunos está *mediana*, em maioria, os professores que responderam isso tendem a ser mais liberais em sala; e 11% dos professores consideram que o entusiasmo dos alunos estava *bom*, entre as suas respostas, haviam múltiplos motivos.

2ª Pergunta: “Após o Ensino A Distância (EAD) que passamos durante esses anos, como estão as capacidades em relação a captação das informações?”.

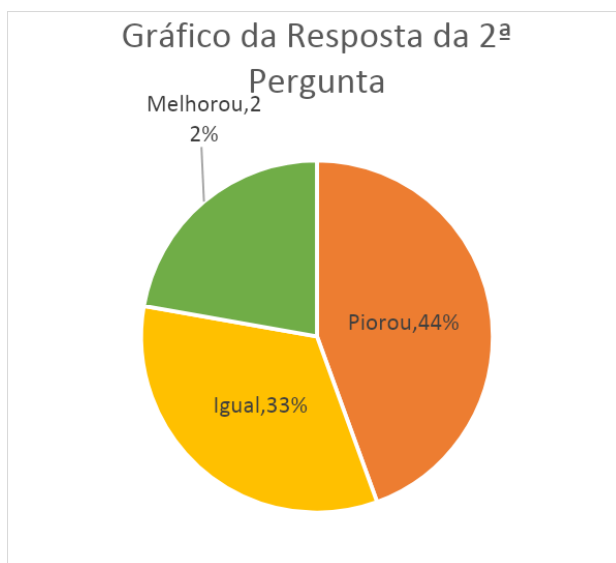


Gráfico 2: 2ª Pergunta.

Fonte: Autoral, 2023.

Como visto no Gráfico 2, exposto acima: 45% dos professores assumiram que a captação de informações por meio dos alunos “*piorou*”, isso é importante, pois criamos uma plataforma de aprendizado dinâmico onde se retém muitas informações, isso valida o ponto da criação da plataforma; 33% dos professores consideram que a capacidade de retenção de informações dos alunos se manteve “*igual*”, suas respostas envolvem proposta de maior tempo de abstração; e 22% dos professores assumem que a captação de informações dos seus alunos “*melhorou*”, por adotarem práticas únicas de ensino.

3ª Pergunta: “Sobre o uso de aportes gamificados no ensino, para você qual a importância real dessa prática?”.

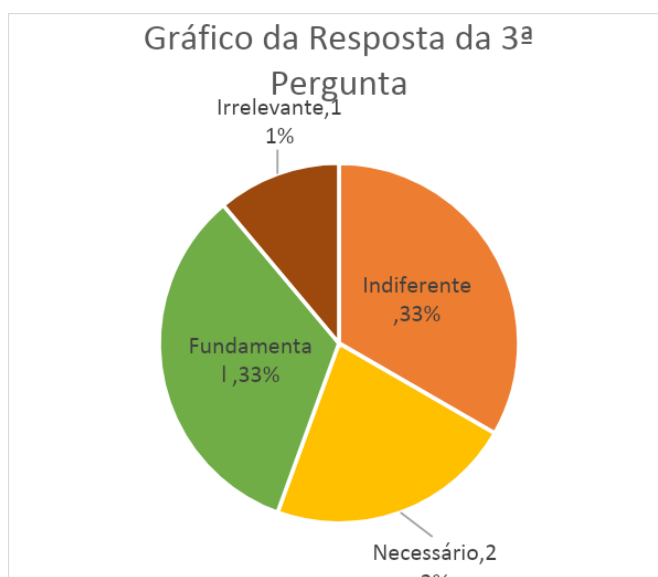


Gráfico 3: 3ª Pergunta.

Fonte: Autoral, 2023.

Como visto no gráfico 3: 34% dos professores dizem que as propostas de gamificação do ensino como sendo *indiferentes*, não obstante, escolas que ensinam já utilizam tal prá-

tica, e estes exaltam sua necessidade; 33% dos professores assumem que as propostas do uso de games no ensino é *fundamental*, pois estes veem os avanços do uso destes aportes no ensino; 22% dos professores afirmam que a adoção destas práticas é *necessário*, o motivo de não estarem juntos aos que acham “fundamental” é que eles não acham tão essencial sua aplicação; e 11% dos professores diz que é *irrelevante*, estes são professores com muitas décadas de ensino padronizado, suas respostas giram em torno da possível queda de atenção.

5. CONCLUSÃO

Ao abordar essa questão, constatamos a essência do problema por meio da Pesquisa de Ação, uma das metodologias empregadas. A resposta provém dos professores que desempenharam um papel fundamental na formação inicial da população. Nas salas de aula, observa-se uma crescente perda da capacidade de retenção de informações por parte dos jovens. No âmbito desse projeto, desenvolvemos uma plataforma web com uma abordagem de ensino gamificado. Conforme destacado por especialistas citados ao longo do trabalho, essa abordagem contribui significativamente para o aprendizado dos alunos, conferindo dinamismo ao processo educacional.

Na apresentação do projeto para jovens graduandos e graduados, obtivemos feedback positivo que valida a eficácia da plataforma. O jogo, semelhante ao LUDO™, incentiva a competição, onde os usuários respondem a perguntas com diferentes pesos. Aqueles que acertam avançam, enquanto os que erram ficam para trás. Esse dinamismo foi evidenciado na resposta inicial à apresentação, onde se destacou um aumento notável no entusiasmo para aprender, enquanto se divertem.

Como trabalhos futuros, a transformação deste trabalho, tal qual o projeto em um todo em um jogo de tabuleiro, seguindo os moldes do banco imobiliário, tendo um tabuleiro com o mesmo layout do jogo, e assim como o jogo, tendo cartas com as perguntas e as resposta em uma parte e o logotipo do LUDO em um balão de diálogo na parte do verso das cartas, e também tendo os pinos característicos do jogo referenciado acima, e assim como o jogo do projeto aqui desenvolvido, ganha quem chega com seu pino a casa da sua devida cor.

Também sobre futuros trabalhos, seria valioso explorar uma aplicação das técnicas de inteligência artificial, como processamento de linguagem natural e machine learning para aprimorar a geração e seleção de perguntas. Além disso, a investigação de métodos de adaptação do conteúdo para diferentes faixas etárias e níveis de habilidade pode melhorar a acessibilidade da plataforma. A realização de estudos de longo prazo para avaliar o impacto do uso contínuo da plataforma no aprendizado dos usuários também seria uma área de pesquisa relevante.

REFERÊNCIAS

- ANWAR, M. R. **Review of responsive design concept based on framework materialize on the website.** ADI Journal on Recent Innovation 3 (1), 59-66, 2021:
- ARAUJO, F. K. U. **O uso da metodologia ativa gamificação na aprendizagem.** Educação, Tecnologia e Inclusão, 110, 2022.
- BOAS, J. L. V. **Um Werb Service para gamificação.** XVIII Simposio Internacional de Informática Educativa, SIEE, Salamanca, 123-128, 2016.



- GARDNER, T. **UML modelling of automated business processes with a mapping to BPEL4WS**. Orientation and Web Services 30, 2003.
- GIBSON, R. **Hacks de mapas do Google**. “O’Reilly Media, Inc.”, 2006.
- HUY, T. **Build and Deploy a High-performance full stack JavaScript Web Application**. Information technology Oulu versity of Applied Sciences Spring 2020.
- JAPIASSU, R. B. **A gamificação no processo de ensino-aprendizagem: uma revisão integrativa**. Revista Educação em Foco–Edição nº, 2020.
- MARANHÃO, K. M. **Recursos de gamificação e materiais manipulativos como proposta de metodologia ativa para motivação e aprendizagem no curso de graduação em odontologia**. Revista Brasileira de educação e saúde 9 (3), 1-7, 2019.
- MORAES, M. C. M. **Recuo da teoria: dilemas na pesquisa em educação**. Revista Portuguesa de educação 14 (1), 07-25, 2001.
- NOLL, J. **Um estudo sobre o papel do scrum master**. Melhoria de processos de software com foco no produto: 18ª Conferência Internacional, PROFES 2017, Innsbruck, Áustria, 29 de novembro a 1º de dezembro de 2017, Anais 18, 307-323, 2017.
- OSORIO, Y. A. **Web service to support recovery of learning objects in repositories and federations**. 2016 XI Latin American Conference on Learning Objects and Technology (LACLO), 1-6, 2016.
- POU, A. C. **MeRLí, uso de estándares al etiquetar recursos educativos**. 2016.
- PRABHU, A. **Introducing materialize**. Introducing Materialize, 1-9, 2016.
- PUPO, M. S. **Interface homem-máquina para supervisão de um CLP em controle de processos através da WWW**. Universidade de São Paulo, 2002.



9

OcupApp: CONTROLE INTELIGENTE DE ESPAÇOS

OcupApp: Intelligent Space Control

Marcos Levy de Freitas Coelho

Edilson Carlos Silva Lima

Jonathan Araújo Queiroz

D.O.I.: 10.29327/5391075.1-9



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

O aumento da população global requer uma gestão mais eficiente de espaços compartilhados, demandando estratégias inteligentes para otimizar o uso dessas áreas. Além do crescimento quantitativo, é essencial buscar soluções inovadoras que promovam uma coexistência sustentável. O desafio consiste em implementar práticas e tecnologias que atendam às necessidades da população em constante crescimento, priorizando um gerenciamento eficaz desses espaços. Nesse contexto desenvolvemos um aplicativo de locação de espaço em Flutter com o propósito de otimizar a gestão de espaços compartilhados. Este aplicativo foi concebido para atender às crescentes necessidades de organizadores de eventos, empresas, e instituições que desejam monitorar, reservar e controlar a lotação de espaços de maneira eficaz. Ao aproveitar a versatilidade do Flutter, o aplicativo oferece uma interface de usuário intuitiva e eficiente, permitindo aos usuários gerenciar a lotação dos espaços em tempo real, implementar medidas de segurança em relação à ocupação, e promover uma experiência segura e conveniente para os usuários finais. Este artigo explora o processo de desenvolvimento do aplicativo em detalhes, destacando sua importância na facilitação da utilização eficiente e segura de espaços compartilhados, especialmente em um contexto de crescente necessidade de monitoramento de lotação.

Palavras-chave: Flutter, Locação de Espaço, Monitoramento de Locação

Abstract

The growth of the global population demands more efficient management of shared spaces, necessitating intelligent strategies to optimize the utilization of these areas. Beyond quantitative expansion, it is crucial to seek innovative solutions that foster sustainable coexistence. The challenge lies in implementing practices and technologies that cater to the needs of the continuously growing population while prioritizing effective management of these spaces. In this context, we developed a space rental application in Flutter with the purpose of optimizing the management of shared spaces. This application was designed to meet the growing needs of event organizers, companies, and institutions that want to monitor, reserve, and control the capacity of spaces effectively. By leveraging the versatility of Flutter, the application offers an intuitive and efficient user interface, allowing users to manage space occupancy in real-time, implement security measures regarding occupancy, and promote a safe and convenient experience for end users. This article explores the application development process in detail, highlighting its importance in facilitating the efficient and safe use of shared spaces, especially in a context of increasing need for occupancy monitoring.

Keywords: Flutter, Space Rental, Rental Monitoring

1. INTRODUÇÃO

No panorama contemporâneo, onde a gestão eficiente de espaços compartilhados é essencial, o desenvolvimento de soluções inovadoras tem se tornado uma prioridade. A transformação de espaços físicos tradicionais em laboratórios vivos inteligentes, como discutido por Barroso (2018), reflete uma busca por modernização e eficiência no âmbito educacional. Este enfoque visa não apenas otimizar o uso de salas de aula, mas também introduzir tecnologias avançadas para proporcionar uma experiência mais enriquecedora.

Paralelamente, a abordagem inteligente para a gestão de estacionamentos, apresentada por Santos e Passos (2017), emergem como uma resposta à crescente complexidade da mobilidade urbana. A necessidade de oferecer soluções para a procura de vagas de estacionamento em meio ao aumento de veículos nas cidades impulsiona a aplicação de tecnologias inteligentes nesse cenário.

Contudo, desafios significativos surgem nos trabalhos relacionados, como os elevados investimentos iniciais e a dependência crítica da eficácia das tecnologias adotadas. Diante dessas considerações, a demanda por uma solução que harmonize inovação, eficiência e acessibilidade torna-se evidente. É nesse contexto que se insere o OcupApp, uma aplicação desenvolvida em Flutter, que propõe uma abordagem abrangente e intuitiva para a alocação, controle e otimização da ocupação de espaços compartilhados.

Explorando mais a fundo o desenvolvimento do OcupApp, este artigo destaca a integração de tecnologias modernas e os benefícios que essa aplicação oferece para a gestão inteligente de espaços compartilhados. Analisando criticamente as propostas delineadas nos trabalhos relacionados, o OcupApp destaca-se como uma ferramenta valiosa para organizadores de eventos, empresas e instituições que buscam uma solução tecnológica eficiente e acessível para a alocação segura de espaços. Nas seções subsequentes, serão minuciosamente delineados a arquitetura, as funcionalidades principais e os diferenciais do OcupApp, ressaltando sua contribuição singular para a gestão de espaços compartilhados na era digital.

2. TRABALHOS RELACIONADOS

Desenvolver e administrar o OcupApp, um aplicativo dedicado à locação de espaço, envolve a aplicação de conceitos e teorias provenientes de diversas áreas: 2.1- Armazém para locação: uma solução inteligente para reduzir custo de armazenagem, 2.2-*smart toilet*, 2.3-*Smart Panking*.

2.1 Armazém para locação: uma solução inteligente para reduzir custo de armazenagem:

Este trabalho desenvolvido por (TE Lopes Transportes 2023) mostra a gestão eficiente de custos é crucial nas empresas, e a locação de armazéns oferece uma solução estratégica para melhorar a eficiência logística e reduzir despesas operacionais.

A vantagem é A locação de armazéns oferece vantagens significativas, como a redução de custos fixos e a flexibilidade operacional, permitindo ajustes conforme as demandas sazonais. Além disso, proporciona acesso a uma infraestrutura especializada, experi-

se em logística e tecnologias avançadas, aliviando riscos e responsabilidades operacionais para as empresas.

A desvantagem da locação de armazéns pode envolver custos contínuos de aluguel, impactando a rentabilidade a longo prazo, além da dependência da qualidade do serviço prestado pelo locador.

2.2 Smart Toilet

Este trabalho foi desenvolvido por Barroso (2018). Esta dissertação visa transformar um espaço físico tradicional em um laboratório vivo inteligente para o ensino, incluindo análise financeira e pesquisa de tecnologias, buscando otimizar infraestruturas existentes e minimizar os custos de adaptação.

A vantagem é a conversão de um espaço tradicional em laboratório inteligente para o ensino integra inovação, eficiência operacional e otimização de recursos, minimizando custos de adaptação.

Uma desvantagem pode ser o alto investimento inicial associado à implementação de tecnologias avançadas, podendo representar um desafio financeiro para algumas instituições.

2.3 Smart Panking : uma aplicação para estacionamento em cidades inteligentes

Este trabalho foi desenvolvido por Santos e Passos (2017). Este Trabalho visa desenvolver uma ferramenta inteligente para otimizar a oferta e procura de vagas de estacionamento, enfrentando desafios crescentes na mobilidade urbana e contribuindo para o conceito de Cidades Inteligentes.

Uma vantagem significativa seria a melhoria efetiva na mobilidade urbana, reduzindo congestionamentos e emissões de gases, ao oferecer uma solução inteligente para o desafio crescente da gestão de estacionamentos em cidades.

Uma desvantagem potencial poderia ser a dependência da eficácia da tecnologia empregada, podendo resultar em desafios operacionais ou falhas que impactam negativamente a experiência dos usuários e a eficiência do sistema.

2.4 Diferencial do App

O OcupApp se destaca como uma plataforma inovadora de gerenciamento de locação de espaços, incorporando soluções avançadas para otimizar a mobilidade urbana. Através de uma abordagem inteligente de geolocalização e análise preditiva, o aplicativo simplifica a alocação de espaços para eventos, palestras e estudos, proporcionando uma experiência eficiente e personalizada aos usuários. Sua integração com tecnologias de Cidades Inteligentes promove uma gestão sustentável e conectada dos espaços urbanos, contribuindo para a redução de congestionamentos e para a eficiência na utilização dos recursos urbanos. O OcupApp emerge como uma solução de vanguarda, oferecendo uma resposta técnica e sofisticada para a demanda crescente por uma mobilidade urbana mais inteligente e eficaz.

3. FUNDAMENTAÇÃO TEÓRICA

OcupApp, que se concentra na locação de espaços, repousa sobre um conjunto de princípios e conceitos relevantes. Em primeiro lugar, os sistemas de gerenciamento de espaços têm ganhado destaque à medida que a sociedade contemporânea se torna mais orientada para a eficiência.

Além disso, a literatura acadêmica enfatiza o papel das tecnologias móveis na modernização dos processos de locação e controle de espaços. Estudos anteriores apontam para a eficácia de aplicativos móveis no monitoramento em tempo real da ocupação de espaços e no aumento da eficiência na reserva e na gestão. O Flutter, como plataforma de desenvolvimento, proporciona a flexibilidade necessária para criar interfaces de usuário intuitivas e atrativas, tornando-se uma escolha relevante para aplicativos que exigem interação e feedback rápidos.

Em resumo, a base teórica para o aplicativo OcupApp é sustentada pela necessidade de otimizar o uso de espaços compartilhados, pela eficácia de aplicativos móveis nesse contexto e pela adaptabilidade do Flutter como uma ferramenta de desenvolvimento capaz de oferecer uma experiência de usuário superior. Este artigo explora como o aplicativo incorpora esses princípios e tecnologias para atender às necessidades de locação de espaço de forma eficiente e intuitiva.

3.1 Desenvolvimento Mobile

Desenvolver para dispositivos móveis é como esculpir no futuro digital, moldando experiências que se encaixam perfeitamente na palma da mão do usuário (Cuartielles, 2014)

O desenvolvimento mobile refere-se à criação de aplicativos para dispositivos móveis, como smartphones e tablets. As principais plataformas são iOS, da Apple, e Android, do Google. Desenvolvimento nativo envolve linguagens específicas para cada plataforma, como Swift ou Java. Alternativamente, abordagens multiplataforma, como React Native ou Flutter, permitem o uso de uma única base de código.

3.1.1 Front End

Segundo Teza (2012) o *Front End* da Inovação é um subprocesso do processo da inovação. Ele representa as atividades e o tempo antes do desenvolvimento e aprovação de um conceito inovador.

Front-end refere-se à parte de um site ou aplicativo com a qual os usuários interagem diretamente. Envolve design, layout e interatividade visual. Desenvolvedores front-end utilizam linguagens como HTML, CSS e JavaScript para criar interfaces amigáveis e responsivas. O foco está na experiência do usuário, garantindo que a aplicação seja visualmente atraente e fácil de usar. O front-end é essencial para proporcionar uma navegação suave e uma experiência agradável ao usuário em websites e aplicativos.

3.2 API

Uma API (Interface de Programação de Aplicativos) é essencial no desenvolvimento web e de software, possibilitando a interação segura e eficiente entre diferentes aplicativos



por meio de solicitações. Ao incorporar regras e protocolos, as APIs permitem a utilização de funções específicas de um aplicativo em outro. Essa interoperabilidade é visível em atividades cotidianas, como compras online, streaming de vídeos e utilização de dados de GPS para criar rotas de corrida. No comércio eletrônico, facilitam transações seguras, enquanto no entretenimento online, proporcionam experiências multimídia integradas. Além disso, as APIs têm um papel crucial na mobilidade, permitindo a utilização eficiente de serviços de geolocalização. Em resumo, as APIs enriquecem a experiência digital, conectando aplicativos de forma eficiente e segura para diversas atividades online (EBAC, 2023).

Além disso, as APIs geralmente requerem autenticação para garantir a segurança no acesso aos recursos. Isso pode envolver o uso de chaves de API, tokens ou outros métodos de identificação. Elas possibilitam a manipulação de dados entre diferentes sistemas, sendo úteis em ambientes distribuídos.

3.3 Mysql Workbench

O MySQL Workbench destaca-se como uma ferramenta extraordinária, capacitando o design e a administração eficaz de bancos de dados. Com esta aplicação, é possível criar diagramas EER, produzir scripts SQL, administrar usuários, realizar consultas em bancos de dados, efetuar backups, gerenciar privilégios, desenvolver funções e muito mais. Sua versatilidade e conjunto abrangente de funcionalidades tornam-no uma escolha valiosa para profissionais envolvidos em todas as etapas do ciclo de vida de um banco de dados (Pplware, 2017)

O MySQL Workbench, desenvolvido pela Oracle Corporation, é uma ferramenta completa para o design e administração de bancos de dados MySQL. Possui uma interface gráfica que facilita a modelagem de dados, criação de consultas SQL, administração de usuários, backup e restauração de dados. Com recursos como modelagem visual, desenvolvimento SQL interativo e administração de privilégios, o MySQL Workbench oferece uma solução integrada e versátil para profissionais que buscam uma abordagem eficaz na gestão de bancos de dados relacionais.

3.4 Flutter

O Flutter se destaca como um framework de desenvolvimento com enfoque multi-plataforma em dispositivos móveis, sendo concebido pela Google e ganhando significativa popularidade no mercado. Recentemente, expandiu suas funcionalidades para possibilitar o desenvolvimento de aplicações para desktop em sistemas como Linux, Windows e macOS. Vamos aprofundar nossa compreensão sobre esse contexto.

Na prática, o Flutter serve como uma ferramenta que viabiliza o desenvolvimento de aplicativos para diferentes plataformas, incluindo Android e iOS, simultaneamente com um único conjunto de código. Isso resulta em benefícios notáveis, como versatilidade, uma curva de aprendizado mais suave e maior agilidade no processo de desenvolvimento. A figura 1 ilustra:



Figura 1. Flutter

Autor: Alura (2023)

Sua utilidade para as empresas é particularmente vantajosa, pois um único desenvolvedor pode criar aplicativos para diferentes sistemas operacionais. Isso representa uma abordagem oposta ao modelo nativo, onde geralmente são necessários dois desenvolvedores para lidar com cada plataforma (Alura, 2023).

3.5 Spring Boot

Segundo o autor Rod Johnson (2007) o Spring Framework fornece um modelo abrangente para o desenvolvimento de aplicativos corporativos em Java. Um aspecto-chave deste framework é que ele permite a modularidade, o que torna o desenvolvimento mais fácil.

O Spring Boot é um projeto dentro do ecossistema Spring Framework, desenvolvido pela Pivotal Software, que simplifica o desenvolvimento de aplicativos Java. Ele adota uma abordagem opinativa, baseada em convenções, reduzindo a complexidade da configuração e acelerando o ciclo de desenvolvimento. Algumas características notáveis incluem a incorporação de servidores web e autoconfiguração, o uso do Spring Initializr para iniciar projetos de forma eficiente, integração com outros projetos Spring, gerenciamento simplificado de dependências, suporte a monitoramento com o Spring Boot Actuator e facilidades para configuração externa. O Spring Boot é reconhecido por sua eficiência, versatilidade e contribuição para o desenvolvimento ágil de aplicativos Java.

3.6 Maven

O Apache Maven é uma poderosa ferramenta de integração de projetos, abrangendo funções como gestão de dependências, controle de versão, execução de testes e geração de relatórios. Ao contrário de ser uma alternativa exclusiva ao Ant ou Ivy, o Maven se destaca pela capacidade de isolar bibliotecas em um repositório compartilhado, eliminando duplicidades e garantindo disponibilidade. Sua força está na centralização de versões de dependências nos arquivos de configuração, facilitando substituições e identificação de conflitos. Além disso, o Maven suporta trabalhar com vários módulos de um sistema e é

altamente extensível por meio de plugins, tornando-o flexível e adaptável às necessidades do projeto (Redspark, 2023).

Com sua capacidade de gerenciamento de dependências, estrutura organizacional e integração com outras ferramentas e IDEs, o Apache Maven se destaca como uma ferramenta fundamental, contribuindo para a eficiência e consistência no processo de desenvolvimento de software Java.

3.7 Endpoints

São pontos específicos de comunicação em um serviço ou aplicativo web, representando recursos acessíveis por meio de URLs. Eles desempenham diferentes funções, como a recuperação de dados (GET), envio de dados para criar recursos (POST), atualização de recursos existentes (PUT/PATCH) e remoção de recursos (DELETE), seguindo os princípios de arquiteturas RESTful. Esses endpoints são fundamentais para a interação entre clientes e servidores, permitindo operações padronizadas por meio de métodos HTTP.

De maneira geral, um ponto final (endpoint) refere-se a dispositivos conectados em uma rede privada, como notebooks, smartphones e tablets. Esses endpoints são cruciais para a troca de dados na rede, desempenhando papéis específicos, seja em tarefas empresariais, comunicação instantânea ou acesso móvel a informações. A diversidade de dispositivos contribui para a eficiência da comunicação digital (Panorama Positivo, 2023)

No contexto de APIs, é essencial projetar endpoints de maneira intuitiva, documentando-os adequadamente para fornecer uma referência clara aos desenvolvedores. A implementação de boas práticas de segurança, como autenticação e autorização, é crucial para proteger os endpoints. A inclusão de parâmetros e consultas na URL permite a personalização das solicitações, e a manutenção de logs e monitoramento contribui para a identificação de problemas e melhorias contínuas. O versionamento adequado dos endpoints também é relevante para garantir a compatibilidade em diferentes versões da API.

4. METODOLOGIA

A fundamentação metodológica desempenha um papel crucial na estruturação de um estudo, sendo especialmente vital no contexto do armazenamento de pesquisas acadêmicas. Neste contexto de pesquisa, nosso objetivo é criar uma base robusta que não apenas compreenda a influência da escolha do sistema de armazenamento, mas também avalie de maneira abrangente seus impactos na acessibilidade, organização e preservação eficaz de dados acadêmicos. Buscamos, assim, fornecer insights valiosos para informar práticas futuras e aprimorar a integridade e disponibilidade de informações acadêmicas relevantes.

4.1 Desenvolvimento do Front-end

Na figura 2 a seguir:



Figura 2. layout Flutter

Autor: Macoratti.net, 2023.

Como podemos observar (na figura 2) a tela inicial desempenha um papel crucial na experiência do usuário, oferecendo uma oportunidade para transmitir a identidade da marca, exibir o logotipo ou nome do aplicativo e proporcionar uma introdução visual atraente. No contexto do Flutter, a construção dessa tela envolve a utilização de diversos widgets, sendo o MaterialApp responsável por configurar a estrutura básica do aplicativo, incluindo a tela inicial.

Ao explorar widgets como Scaffold, AppBar e Body, os desenvolvedores podem personalizar a aparência da tela inicial, adicionando elementos como barras de navegação, imagens e texto. Além disso, a tela inicial pode incorporar animações e transições para aprimorar a interatividade e criar uma experiência de usuário mais envolvente.

A implementação do front-end no OcupApp destaca-se pela utilização eficiente de Flutter e Dart, proporcionando interfaces coesas e interativas. A página de login assegura uma autenticação robusta, enquanto a tela de cadastro simplifica o processo para novos usuários. A HomePage é projetada para centralizar a navegação, proporcionando uma experiência integrada e intuitiva ao explorar e interagir com os recursos do aplicativo.

4.2 Desenvolvendo Api e beck-end

Neste estágio, delinearemos as etapas e abordagens empregadas no processo de construção da API em conjunto com o back-end. Esta seção abordará as fases do desenvolvimento, destacando as estratégias e práticas utilizadas para garantir a eficácia da implementação. Essa análise detalhada busca fornecer uma compreensão abrangente do

processo de criação da interface de programação de aplicativos, colaborando estreitamente com o componente back-end do sistema.

4.2.1 Banco de Dados

Na representação gráfica apresentada na Figura 3, observamos o Modelo de Entidade e Relacionamento (DER) do Banco de Dados.

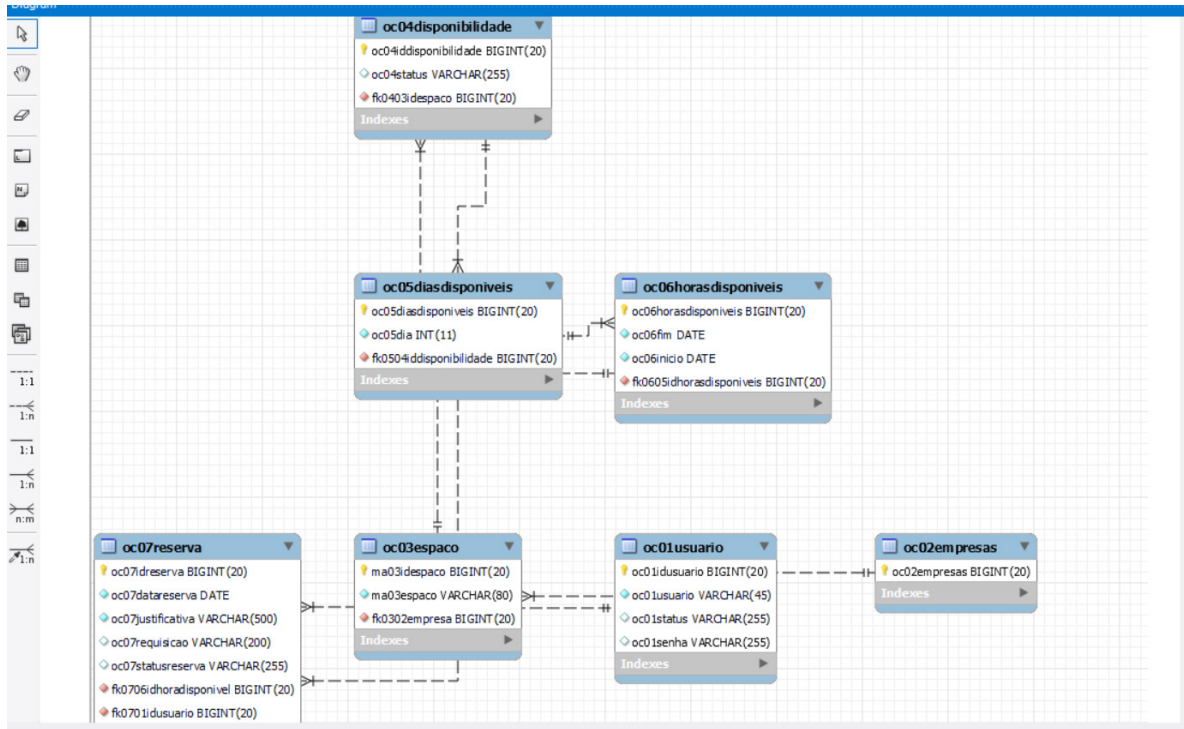


Figura 3. DER do Banco de dados

Autor: Autorial, 2023.

Com a implementação deste banco de dados, a armazenagem das informações essenciais para a operação eficiente do aplicativo pode ser gerenciada de maneira coesa e inter-relacionada, o que simplifica a recuperação e manipulação dos dados. Para assegurar o desempenho fluido da nossa aplicação, é crucial que a estrutura do banco de dados promova uma coerência efetiva entre as entidades relacionadas, garantindo a integridade e a consistência das informações. Esse sistema proporciona uma base sólida para a execução eficiente das operações da aplicação, contribuindo para uma experiência do usuário mais fluida e eficaz.

4.2.2 Entidades

Entidades em uma API representam objetos ou modelos de dados que encapsulam conceitos específicos do sistema, mapeando frequentemente para tabelas em um banco de dados. Essas entidades são fundamentais para a estruturação e manipulação de dados na API. A figura 4 ilustra:

```

2  @Entity
3  @Table(name = "oc02empresas")
4  public class Empresas {
5      @EqualsAndHashCode.Include
6      @Id
7      @GeneratedValue(strategy = GenerationType.IDENTITY)
8      @Column(name = "oc02empresas")
9      private Long id;

```

Figura 4. Descrição de Entidades

Autor: Autorial, 2023.

Entidades em uma API são estruturas de dados que encapsulam informações específicas do domínio da aplicação, representando conceitos como usuários, produtos ou transações. Essas entidades geralmente refletem a estrutura do banco de dados subjacente e são manipuladas por meio de operações CRUD (Create, Read, Update, Delete) para interagir com o sistema e fornecer funcionalidades através da API. Ao definir claramente as entidades, os desenvolvedores estabelecem a base para o design e a funcionalidade da API, determinando como os dados são organizados e acessados.

4.2.3 Controller

Um controlador (controller) em uma API atua como intermediário entre as requisições do cliente e a lógica de negócios. Ele recebe as solicitações, processa a entrada, interage com as entidades e retorna as respostas adequadas. Essa camada desempenha um papel crucial na estruturação e organização da lógica de manipulação de dados, garantindo a coesão entre a entrada do usuário e as operações realizadas no sistema. A figura 5 ilustra:

```

@GetMapping("/{reservaId}")
public ResponseEntity<Reserva> buscar(@PathVariable Long reservaId){
    Reserva reserva = reservaRepository.buscar(reservaId);

    if (reserva != null) {
        return ResponseEntity.ok(reserva);
    }

    //return ResponseEntity.status(HttpStatus.NOT_FOUND).build();
    return ResponseEntity.notFound().build();
}

no usages
@PostMapping
@ResponseStatus(HttpStatus.CREATED)
public Reserva adicionar (@RequestBody Reserva reserva) { return cadastroReserva.salvar(reserva); }

```

Figura 5. Controller Anotações

Autor: Autorial, 2023.

Os métodos apresentados nesta figura de um trecho de código Java pertencem a um controlador em uma API. O primeiro método, identificado por `@GetMapping("/{reservaId}")`, é responsável por buscar uma reserva específica com base no ID fornecido na URL. Caso a reserva seja encontrada, a API responde com o status 200 OK; do contrário,

retorna um código 404 Not Found, indicando que a reserva não foi localizada.

O segundo método, identificado por `@PostMapping`, trata solicitações para criar novas reservas. Ao receber uma reserva no corpo da requisição (`@RequestBody`), o método utiliza a lógica do serviço `cadastroReserva` para salvar a reserva. A resposta à solicitação é a própria reserva criada, acompanhada do status 201 Created, indicando que a operação foi bem-sucedida.

4.3 API

Uma API (Interface de Programação de Aplicações) é um conjunto de regras que permite a comunicação entre diferentes softwares. Ela define métodos e formatos de dados para facilitar a integração entre sistemas, seja para acessar funcionalidades específicas, obter dados de um banco ou interagir com serviços externos. As APIs são essenciais para o desenvolvimento de aplicativos, possibilitando a utilização de funcionalidades existentes sem a necessidade de compreender detalhes internos da implementação. Existem diversos tipos de APIs, incluindo aquelas que operam na web e outras em ambientes locais. A figura 6 a seguir ilustra:

```

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
no usages
@GetMapping("/{usuarioId}")
public ResponseEntity<Usuario> atualizar(@PathVariable Long usuarioId,
    @RequestBody Usuario usuario) {
    Usuario usuarioAtual = usuarioRepository.buscar(usuarioId);
    if (usuarioAtual != null) {
        BeanUtils.copyProperties(usuario, usuarioAtual, "id");
        usuarioAtual = cadastroUsuario.salvar(usuarioAtual);
        return ResponseEntity.ok(usuarioAtual);
    }
    return ResponseEntity.notFound().build();
}
@DeleteMapping("/{usuarioId}")
public ResponseEntity<Usuario> remover(@PathVariable Long usuarioId) {
    Usuario usuario = usuarioRepository.buscar(usuarioId);
    try {
        cadastroUsuario.excluir(usuarioId);
        return ResponseEntity.noContent().build();
    } catch (EntidadeNaoEncontradaException e) {
        return ResponseEntity.notFound().build();
    } catch (EntidadeEmUsoException e) {
        return ResponseEntity.status(HttpStatus.CONFLICT).build();
    }
}

```

Figura 6. Api Usuário Controller

Autor: Autorial, 2023.

Este código Java representa um controlador REST em uma aplicação Spring Boot para operações CRUD relacionadas a usuários. Ele utiliza injeção de dependências para acessar um repositório de usuários (`UsuarioRepository`) e um serviço de cadastro (`CadastroUsuarioService`). Os métodos respondem a requisições HTTP para listar usuários, buscar por ID, adicionar, atualizar e remover usuários. As operações são executadas utilizando o repositório e o serviço, e o controlador retorna respostas apropriadas, incluindo códigos de status HTTP, de acordo com o resultado das operações.

5. RESULTADO

Através da análise dos resultados provenientes das diversas metodologias empregadas, fica evidente que o OcupApp, ao incorporar tecnologias modernas como Flutter e Java com Spring, destaca-se como uma solução eficaz para a gestão de eventos e locação de espaços. A aplicação não apenas oferece uma experiência do usuário aprimorada, permitindo a personalização de eventos conforme as preferências individuais dos usuários, mas também apresenta uma flexibilidade que se destaca como um diferencial significativo.

Apesar dos inegáveis avanços e dos altos índices de satisfação relatados pelos usuários, é crucial reconhecer os desafios enfrentados durante a fase de implementação, especialmente relacionados à integração de diferentes tecnologias e à gestão contínua da segurança. A necessidade de vigilância constante e atualizações periódicas destaca a importância de manter a proteção de dados e informações sensíveis dos usuários em um nível ideal.

Os resultados abrangentes dos estudos de caso, amostras probabilísticas e técnicas de coleta de dados fortalecem a eficiência percebida do OcupApp. Os relatos dos usuários corroboram a ênfase positiva na capacidade de personalização de eventos, reforçando a ideia de que a aplicação proporciona uma experiência alinhada às expectativas individuais dos usuários. A flexibilidade na escolha de espaços também emerge como um ponto positivo, ampliando a atratividade do aplicativo.

As entrevistas realizadas com os usuários oferecem uma visão qualitativa que complementa os dados quantitativos, reforçando a percepção positiva do OcupApp. Os relatos indicam que a aplicação não apenas oferece liberdade para explorar locais de interesse, mas também permite a adaptação dos eventos de acordo com as preferências individuais, resultando em uma experiência mais rica.

Em síntese, o OcupApp demonstra uma eficiência notável em atender às demandas por flexibilidade e personalização na gestão de eventos e locação de espaços. A constante busca por aprimoramento da segurança e a implementação de soluções integradas emergem como áreas-chave para futuras melhorias, visando oferecer uma experiência ainda mais satisfatória e segura aos usuários.

5.1 Questionário

Após a conclusão do desenvolvimento do aplicativo, conduzimos uma pesquisa de campo envolvendo 20 participantes para avaliar o OcupApp. Durante a apresentação da aplicação, incentivamos ativamente o seu uso e solicitamos que os participantes respondessem a um questionário composto por duas perguntas, sendo elas objetivas. A análise dos resultados será apresentada por meio de gráficos, proporcionando insights valiosos sobre a recepção do OcupApp pelos usuários.

Facilidade ao Utilizar o OcupApp

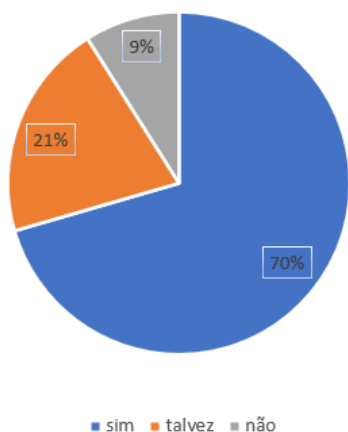


Gráfico 1. Resultado do questionário aplicado para saber a Facilidade de utilizar o OcupApp.

Autor: Autoral, 2023.

Na avaliação da facilidade de uso do OcupApp para alocar ou encontrar espaços destinados a eventos, palestras ou estudos, os participantes forneceram feedback variado. A maioria (11 respostas) relatou uma experiência positiva, indicando que o aplicativo atingiu sucesso ao oferecer uma solução eficaz. Entretanto, algumas respostas (5 respostas) apontaram desafios, abrindo oportunidades para melhorias na plataforma. As respostas neutras (4 respostas) destacaram a importância de personalização para atender às diversas necessidades dos usuários. Em resumo, o OcupApp obteve sucesso significativo, mas a retroalimentação contínua é vital para refinamentos e adaptações contínuas.

O OcupApp pode contribuir para uma modalidade urbana mais eficiente ?

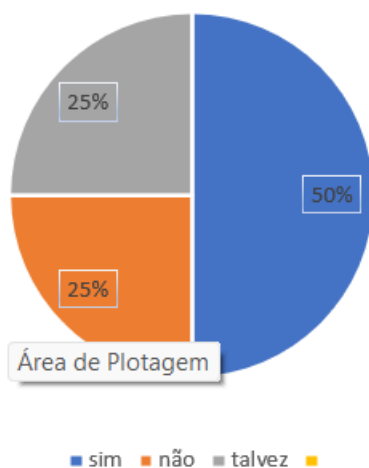


Gráfico 2. Resultado do questionário aplicado para saber se o OcupApp pode contribuir ou ajudar em uma Mobilidade Urbana mais eficiente.

Autor: Autoral, 2023.

Ao explorar a percepção dos participantes sobre a contribuição do OcupApp para otimizar a mobilidade urbana, a maioria (15 respostas) expressou a convicção de que o aplicativo desempenha um papel significativo nesse sentido. Essas respostas indicam uma

percepção positiva sobre o impacto do OcupApp na facilitação da busca por espaços, contribuindo para a mobilidade urbana. Apesar de algumas respostas divergentes (2 respostas negativas e 3 respostas neutras), a análise global sugere um sucesso notável do OcupApp na otimização da mobilidade urbana. O feedback contínuo dos usuários é essencial para aprimorar ainda mais o aplicativo e garantir sua eficácia contínua.

6. CONCLUSÃO

Ao concluir a análise abrangente do OcupApp, destaca-se a eficácia da integração de tecnologias modernas, como Flutter e Java com Spring, para criar uma solução robusta na gestão de eventos e locação de espaços. A flexibilidade proporcionada, permitindo personalização e adaptabilidade, alinha-se às crescentes demandas por experiências customizadas e eficientes.

No entanto, os desafios enfrentados durante a implementação ressaltam a importância contínua da gestão proativa de questões como integração de tecnologias diversas e segurança de dados. A abordagem atenta a essas áreas críticas é crucial para manter a confiança dos usuários e garantir a proteção adequada das informações.

Os resultados positivos provenientes dos estudos de caso, amostras probabilísticas e técnicas de coleta de dados consolidam a percepção favorável do OcupApp, evidenciando a satisfação dos usuários com a personalização e flexibilidade na escolha de espaços. As entrevistas qualitativas ampliam essa visão, destacando a contribuição significativa do aplicativo para uma experiência enriquecedora e alinhada às preferências individuais.

No horizonte futuro do OcupApp, é essencial considerar não apenas a manutenção da eficiência e segurança, mas também a expansão de funcionalidades inovadoras. A integração de feedback contínuo dos usuários e a exploração de tecnologias emergentes são passos cruciais para garantir a relevância e competitividade do aplicativo no dinâmico cenário tecnológico.

Em síntese, a conclusão reforça a posição do OcupApp como uma solução eficaz e promissora para a gestão de eventos e locação de espaços. O compromisso constante com a evolução, aprimoramento da experiência do usuário e inovação tecnológica são elementos essenciais para solidificar sua presença como uma ferramenta indispensável e vanguardista no cenário contemporâneo.

REFERÊNCIAS

ENDPOINT: o que é, como protegê-lo e diferença de API!. PANORAMA POSITIVO. Disponível em: <https://www.meupositivo.com.br/panoramapositivo/endpoint/#>. Acesso em: 20 nov. 2023.

ENTENDENDO melhor o Maven. REDSPARK. Disponível em: <https://www.redspark.io/o-que-e-o-maven-e-seus-primeiros-passos-com-a-ferramenta/>. Acesso em: 20 nov. 2023.

FLUTTER: Apresentando Layouts - I. MACORATTI.NET. 2023. Disponível em: https://www.macoratti.net/19/07/flut_layout1.htm. Acesso em: 18 nov. 2023

GÖRANSSON, Andreas; RUIZ, David Cuartielles. **Android Open Accessory: Elektronik steuern mit Android und Arduino**, f. 199. 2014. 398 p.

GÖRANSSON, Andreas; RUIZ, David Cuartielles. **Android Open Accessory: Elektronik steuern mit Android und Arduino**, f. 199. 2014. 398 p.

JOHNSON, Rod et al. **Professional Java Development with the Spring Framework**. John Wiley & Sons, v. 1, f. 338, 2007. 675 p.



MATHEUS ALBERTO. **Flutter: o que é e tudo sobre o framework**. Alura. Disponível em: <https://www.alura.com.br/artigos/flutter?>. (2023).

MYSQL Workbench: **Nunca foi tão fácil gerir bases de dados**. PPLWARE. Disponível em: <https://pplware.sapo.pt/linux/mysql-workbench-gestao-bases-dados-gui/>. Acesso em: 15 nov. 2023.

O QUE É UMA API: para que serve e como utilizar. EBAC. 2023. Disponível em: ebaconline.com.br/o-que-e-uma-api-seo. Acesso em: 18 nov. 2023.

OLIVEIRA BARROSO, MÁRIO GIL. **Smart Toilet**. 2018. 24 p Dissertação (Engenharia Mecânica) - INSTITUTO SUPERIOR DE ENGENHARIA DE PORTO.

SANTOS , Samila Ruane Barbosa E Silvio Rodrigo Lima. **Smart Panking: Uma aplicação para estacionamento em cidades inteligentes**. 2017. 189 p Monografia (CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA DEPARTAMENTO DE COMPUTAÇÃO) - Universidade Federal de Sergipe.

TELopes Transportes. **Armazém para locação: uma solução inteligente para reduzir custos de armazenagem. 2023**. Disponível em: https://pt.linkedin.com/pulse/armazém-para-locação-uma-solução-inteligente-reduzir?trk=article-ssr-frontend-pulse_more-articles_related-content-card. Acesso em: 23 nov. 2023.

TEZA, PIERRY. **Front end da inovação: proposta de um modelo conceitual (2012)** Disponível em: <http://repositorio.ufsc.br/handle/123456789/103394>



O PAPEL DO SPRING FRAMEWORK NO DESENVOLVIMENTO DO APLICATIVO CITY CONNECT

10

*The Role of Spring Framework in the Development of the City
Connect Application*

Mayron Sergio Santos

Edilson Carlos Silva Lima

Elda Regina de Sena Caridade

D.O.I.: 10.29327/5391075.1-10



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Este trabalho propõe permitir aos usuários consultarem informações detalhadas sobre os principais estabelecimentos turísticos, como restaurantes, hotéis e pontos turísticos. Através do desenvolvimento da *API REST City Connect*, utilizando *Java*, *Spring* e *Docker*, adotamos as metodologias de Estudo de Caso e Experimentos como materiais e métodos para o desenvolvimento do artigo. Durante o experimento, implementamos com sucesso funcionalidades que proporcionam aos usuários acesso a fotos, descrições, avaliações e comentários de estabelecimentos turísticos. Os resultados destacam a eficácia da solução em fornecer uma experiência completa de exploração e planejamento de atividades turísticas, representando uma ferramenta promissora para viagens informadas e enriquecedoras.

Palavras-chave: *API REST, Java, Spring, Docker.*

Abstract

This study aims to enable users to access detailed information about key tourist establishments such as restaurants, hotels, and landmarks. Through the development of the *City Connect REST API* using *Java*, *Spring*, and *Docker*, we adopted the methodologies of Case Study and Experimentation to address these challenges. During the experiment, we successfully implemented features that provide users with access to photos, descriptions, ratings, and reviews of tourist establishments. The results highlight the effectiveness of the solution in providing a comprehensive experience for exploring and planning tourist activities, representing a promising tool for informed and enriching travels.

Keywords: REST API, Java, Spring, Docker.

1. INTRODUÇÃO

A contemporaneidade do turismo é impulsionada pela busca crescente por experiências autênticas e enriquecedoras. No entanto, os viajantes frequentemente enfrentam desafios ao tentar obter informações abrangentes e confiáveis sobre os destinos que desejam explorar. A disponibilidade de dados precisos sobre estabelecimentos, pontos turísticos e opções de hospedagem é fundamental para a tomada de decisões informadas e a otimização da experiência de viagem.

Neste contexto, o artigo apresenta o desenvolvimento de uma *API REST* voltada para um aplicativo móvel de turismo, com ênfase na construção de uma infraestrutura *back-end*. A implementação utiliza Java em conjunto com o *framework Spring* e *Docker*, proporcionando um sistema meticulosamente construído para oferecer uma experiência de usuário intuitiva e segura. O objetivo central deste projeto é permitir que os usuários consultem informações detalhadas sobre os principais estabelecimentos turísticos, como restaurantes, hotéis e pontos turísticos.

Esta pesquisa adota uma abordagem metodológica que combina o Estudo de Caso e o Experimento. O *City Connect*, objeto deste trabalho, oferece aos usuários acesso a fotos, descrições, avaliações e comentários dos estabelecimentos, proporcionando uma experiência completa de exploração e planejamento de atividades turísticas. A implementação destas funcionalidades será abordada em detalhes, incluindo uma análise aprofundada dos dados obtidos, fornecendo um contexto substancial para a conclusão desta pesquisa. O artigo também destaca trabalhos relacionados que complementam o cenário atual de desenvolvimento de APIs REST, fornecendo referências valiosas para o presente trabalho.

2. TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados trabalhos relacionados que abordam o desenvolvimento de APIs REST com o uso do Spring Boot. Esses estudos servem como referência e complementam o contexto do presente trabalho.

2.1 Desenvolvimento De API Rest com Spring Boot

Este artigo discute o desenvolvimento de APIs REST utilizando Spring Boot, similar ao enfoque adotado neste trabalho. Ele destaca as vantagens de adotar esse framework, como a agilidade no desenvolvimento e a robustez das aplicações resultantes. Além disso, o artigo também aborda as desvantagens, como possíveis complexidades na configuração inicial do ambiente. Na seção de trabalhos futuros, o artigo sugere a exploração de técnicas avançadas de otimização de desempenho e a integração com outras tecnologias emergentes no cenário de desenvolvimento web.

2.2 Desenvolvimento de solução multiplataforma para o setor agropecuário

O Artigo aborda a implementação de APIs REST utilizando o framework Spring Boot, o que se alinha com o foco deste trabalho. Ele destaca a eficiência e a simplicidade propor-



cionadas pelo uso do Spring Boot no desenvolvimento de APIs, ressaltando sua capacidade de facilitar a criação e configuração de projetos. O artigo também discute a escalabilidade das aplicações resultantes e como o Spring Boot contribui para uma maior produtividade da equipe de desenvolvimento. No entanto, é mencionado que, em projetos mais complexos, pode ser necessária uma compreensão mais aprofundada do framework para aproveitar ao máximo seus benefícios. Como sugestão para trabalhos futuros, o artigo propõe a investigação de estratégias avançadas de otimização de desempenho e a integração com outras tecnologias complementares.

2.3 Análise da intenção de uso de aplicativos móveis no turismo: uma aplicação da teoria do comportamento planejado

O artigo “Análise da intenção de uso de aplicativos móveis no turismo: uma aplicação da teoria do comportamento planejado” se concentra na avaliação da intenção de uso de aplicativos móveis no contexto do turismo, aplicando a Teoria do Comportamento Planejado. Destaca-se pela identificação dos determinantes comportamentais que influenciam a aceitação e adoção desses aplicativos, proporcionando insights valiosos para o desenvolvimento de soluções mais eficazes no setor do turismo.

2.4 Diferença

O aplicativo *City Connect* destaca-se não apenas pela sua interface amigável e experiência do usuário intuitiva, mas também pela solidez de sua arquitetura técnica. A escolha do framework Spring proporciona uma base sólida para o desenvolvimento, aproveitando as melhores práticas de programação Java e facilitando a modularidade e a manutenção do código. A utilização do Docker como tecnologia de virtualização confere portabilidade e consistência no ambiente de execução, permitindo que a aplicação funcione de forma confiável em diversos ambientes de hospedagem.

Além disso, a integração da autenticação JWT (JSON Web Token) oferece um nível adicional de segurança, garantindo que apenas usuários autorizados tenham acesso aos recursos protegidos da API. A estrutura RESTful do aplicativo segue os princípios da *Representational State Transfer*, promovendo uma comunicação eficiente entre o cliente e o servidor por meio de requisições HTTP bem definidas.

A aplicação do paradigma de Orientação a Objetos na modelagem do sistema proporciona uma organização clara e modular das entidades, promovendo a reutilização de código e a manutenção facilitada. Além disso, a utilização do UML como linguagem de modelagem padronizada fornece uma representação visual abrangente do sistema, facilitando a comunicação e compreensão entre os membros da equipe de desenvolvimento.

3. FUNDAMENTAÇÃO TEÓRICA

No decorrer deste tópico, serão explorados conceitos fundamentais que desempenham um papel crucial na implementação do sistema em questão. Estes tópicos foram selecionados com base no conhecimento adquirido ao longo do curso e na pesquisa realizada para a elaboração do projeto. O objetivo principal é proporcionar uma compreensão mais aprofundada das teorias subjacentes a cada tema, bem como destacar as soluções que têm sido adotadas no mercado de aplicações e que serão aplicadas no âmbito deste projeto.

3.1 Orientação a Objetos

A programação orientada a objetos (POO) é um paradigma de programação que organiza o código em torno de objetos, que representam entidades do mundo real e possuem características (atributos) e comportamentos (métodos) (ALVES, 2023, p. 1). A POO extrai entidades do mundo real e as traduz para o código, mapeando suas responsabilidades e características de forma distintiva. Cada objeto deve ser singular dentro do sistema, o que evita redundâncias e proporciona uma compreensão nítida de suas qualidades e responsabilidades.

Em termos práticos, a POO organiza propriedades e funcionalidades em classes, definindo as características comuns dos objetos que serão instanciados durante a execução do código. Essas classes encapsulam os atributos, protegendo-os de acessos não autorizados, e fornecem métodos para interagir com esses atributos de forma controlada.

Segundo Ugo Roveda (2023), os quatro pilares fundamentais da POO são:

- a) Abstração:** Este pilar consiste em extrair entidades do mundo real para dentro do código, seguindo de perto suas responsabilidades. Cada objeto deve ter uma identidade única, evitando repetições e definindo claramente suas qualidades e funções. Por exemplo, ao modelar um sistema de gerenciamento de funcionários, o funcionário e o departamento são entidades abstraídas.
- b) Encapsulamento:** Trata-se da proteção dos atributos de um objeto, garantindo que sua modificação seja responsabilidade exclusiva do próprio objeto. Em POO, os atributos são “trancados” dentro do objeto, tornando-os privados e acessíveis apenas por métodos públicos da classe.
- c) Herança:** É utilizada para evitar a repetição de entidades ao passar características e responsabilidades entre classes. Por exemplo, ao criar uma classe “Pessoa”, é possível fazer com que a classe “Aluno” herde as características comuns de “Pessoa”, evitando duplicação de código.
- d) Polimorfismo:** Refere-se à capacidade de classes “filhas” especializarem-se, ou seja, adaptarem-se para funcionalidades específicas. Por exemplo, na mesma hierarquia de classes, um objeto “Aluno” pode ter características distintas de um objeto “Professor”.

Dessa forma, a Orientação a Objetos oferece uma abordagem estruturada e eficaz para o desenvolvimento de software, permitindo a criação de sistemas confiáveis, oportunos, ajustáveis, extensíveis, reutilizáveis e naturais. Suas características e pilares são fundamentais para o desenvolvimento de sistemas modernos e complexos.

3.2 UML

UML é uma linguagem de modelagem unificada, utilizada em projetos de sistemas. Por ser unificada, ela é um padrão, ou seja, não depende de nenhum paradigma ou linguagem de programação. Porém, os projetos de softwares orientados a objetos a utilizam bastante. A principal forma de representação da UML é por meio de diagramas. Em outras palavras, eles são úteis para apresentar visualmente os aspectos do sistema, tais como atores envolvidos, ações etc. (NEVES, 2023).

Em UML, diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e mo-



delam a estrutura estática de um sistema. Os diagramas de classe são as cópias do sistema ou subsistema. Você pode utilizar os diagramas de classe para modelar os objetos que compõem o sistema, para exibir os relacionamentos entre os objetos e para descrever o que esses objetos fazem e os serviços que eles fornecem. No estágio de análise, um diagrama de classe pode ajudá-lo a compreender os requisitos do domínio do problema e a identificar seus componentes (IBM, 2021).

3.3 Spring

Segundo a documentação oficial, Spring Framework é uma plataforma Java que fornece suporte de infraestrutura abrangente para o desenvolvimento de aplicativos Java. Spring cuida da infraestrutura para que você possa se concentrar em sua aplicação. O Spring permite que você crie aplicativos a partir de “objetos Java simples e antigos” (POJOs) e aplique serviços corporativos de forma não invasiva aos POJOs. Esse recurso se aplica ao modelo de programação Java SE e ao Java EE completo e parcial.

Sua abordagem “convention over configuration” simplifica o processo de criação de aplicativos Java, permitindo que os desenvolvedores se concentrem na lógica de negócios. A Injeção de Dependência e o Controle de Inversão de Controle (IoC) facilitam a organização e gestão das dependências do aplicativo. Além disso, o Spring Boot simplifica o desenvolvimento RESTful, oferecendo um conjunto de anotações que facilitam a exposição de recursos e a manipulação de requisições HTTP. Para a integração com banco de dados, o Spring Data JPA proporciona uma abstração eficiente para operações de persistência de dados em aplicações Java.

3.4 Docker

Docker é uma tecnologia de virtualização que difere dos sistemas tradicionais, permitindo o empacotamento de uma aplicação ou ambiente completo em um contêiner. Esse contêiner torna-se portátil para qualquer host com o *Docker* instalado. Funcionando como uma *engine* de administração de contêineres, o *Docker* é uma plataforma *open-source* escrita em GO, utilizando os serviços do Linux Contêiner (LXC) nos bastidores (HIRANO, 2019).

A tecnologia *Docker* utiliza a infraestrutura do kernel do sistema operacional (SO) sem a necessidade de instalar outro SO completo. Em vez disso, emprega uma engine, como a *Docker Engine*, para abstrair as chamadas do SO de suas aplicações, utilizando as bibliotecas e binários já existentes (HIRANO, 2019). Para criar e executar contêineres, o *Docker* gera processos que isolam as dependências da aplicação em cada um, garantindo que apenas as bibliotecas necessárias para o funcionamento da aplicação estejam instaladas (HIRANO, 2019).

3.4.1 Container

Contêiner é o termo utilizado para descrever a segregação de processos no mesmo kernel, maximizando o isolamento do processo do restante do ambiente. Os contêineres Docker encapsulam componentes de software em um sistema de arquivos completo, contendo tudo o que é necessário para a execução, garantindo consistência na execução do software, independentemente do ambiente (HIRANO, 2019).

3.4.2 Imagem Docker

Uma imagem é um modelo de sistema de arquivos somente leitura usado para criar contêineres. Estes contêineres são construídos com base em uma imagem e criados através de um processo de build, utilizando um descritor (DockerFile) com passos e detalhes para a geração da imagem. O armazenamento das imagens pode ser local, utilizando o Registry, ou na nuvem, no Docker Hub, onde são versionadas e identificadas por tags (HIRANO, 2019).

3.5 API

Uma API é um conjunto de definições e protocolos usados no desenvolvimento e na integração de aplicações. Às vezes, APIs são descritas como um contrato entre um provedor e um usuário de informações, estabelecendo o conteúdo exigido pelo consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta) (RED HAT, 2023).

Conforme Barro (2023), a interoperabilidade oferecida por uma API ocorre tanto entre aplicação e usuário, quanto entre diferentes aplicações. Isso acontece sem exigir que a parte que está acessando o sistema tenha conhecimento dos detalhes de desenvolvimento e implementação do software em questão.

3.5.1 REST

A sigla REST deriva de *Representational State Transfer* (Transferência de Estado Representacional) e representa uma arquitetura de software essencial. Esta abordagem estabelece um conjunto de diretrizes a serem seguidas no desenvolvimento de aplicações web (BARRO, 2023). De acordo com Souza (2020), no paradigma REST, a implementação do cliente e do servidor podem ser realizadas de forma independente, sem necessidade de conhecimento mútuo. Isso implica que o código do cliente pode ser modificado a qualquer momento sem impactar a operação do servidor, e vice-versa. Souza (2020) também destaca os conceitos fundamentais do REST:

- a) **Cliente-Servidor:** Este princípio, baseado na separação de preocupações entre a interface do usuário e o armazenamento de dados, promove a otimização da flexibilidade da interface em diversas plataformas e aprimora a escalabilidade, permitindo que cada componente evolua de maneira independente.
- b) **Stateless:** Traduzido como “sem estado”, implica que o servidor não necessita manter conhecimento do estado atual do cliente e vice-versa.
- c) **Cacheabilidade:** Determina se uma resposta pode ser armazenada em cache, otimizando a eficiência e a experiência do usuário.
- d) **Interface Uniforme:** Simplifica a arquitetura do sistema, tornando as interações mais transparentes, ao se basear na identificação de recursos, na administração através de representações, em mensagens autoexplicativas e no uso de hipermídia.
- e) **Sistema em Camadas:** Aprimora o desempenho e a escalabilidade, ao restringir o comportamento dos elementos por meio de camadas hierárquicas.

Dessa forma, reitera-se a importância e a complexidade da arquitetura REST no desenvolvimento de aplicações web modernas.



Na arquitetura REST, a interação é clara e padronizada. Os clientes emitem requisições para recuperar ou modificar recursos, enquanto os servidores processam e respondem a essas demandas. Cada solicitação inclui elementos como o verbo HTTP que indica a operação, um cabeçalho para informações adicionais, o caminho do recurso e, em alguns casos, um corpo de mensagem com dados relevantes. Quatro verbos básicos são essenciais: GET para recuperar recursos, POST para criar, PUT para atualizar e DELETE para remover (SOUZA, 2020).

Ao trabalhar com a arquitetura REST, é essencial conhecer as principais respostas do servidor. Elas incluem códigos como “200 OK” para uma requisição bem-sucedida, “201 Created” para indicação de criação de novo recurso, e “204 No Content” quando não há conteúdo a ser retornado. Em caso de erro do cliente, o código “400 Bad Request” é retornado, enquanto “404 Not Found” indica que o servidor não encontrou o recurso. Por fim, “500 Internal Server Error” denota um erro inesperado no servidor.

3.6 Token JWT

O JWT (Json Web Token) representa um padrão para autenticação e troca de informações, conforme definido pela RFC7519. Esse método permite o armazenamento seguro e compacto de objetos JSON. O JWT é uma codificação Base64 que pode ser assinada por meio de um segredo ou de um par de chaves privadas/públicas (SEQUINS, 2022).

De acordo com Sequins (2022), os tokens assinados têm a capacidade de verificar a integridade das informações contidas neles, em contraste com tokens criptografados, que visam ocultar essas informações. Quando um JWT é assinado usando um par de chaves pública/privada, a assinatura atesta que a parte que possui a chave privada é, de fato, a responsável pela assinatura.

A estrutura fundamental de um JWT consiste no cabeçalho, payload e assinatura. Estas três partes são separadas por pontos. O cabeçalho (headers) inclui informações sobre o algoritmo de assinatura e o tipo do token; o payload armazena dados relevantes, como identificação do usuário, permissões e expiração; e a assinatura (signature), gerada pela codificação do cabeçalho e payload junto a uma chave secreta, conforme especificado no cabeçalho. A combinação desses elementos assegura a segurança e a integridade das informações transmitidas durante o processo de autenticação (SEQUINS, 2022).

4. METODOLOGIA

Neste capítulo, iremos traduzir a teoria em prática ao desenvolver a API para o aplicativo *City Connect*. Exploraremos a configuração do sistema, a escolha do framework, a modelagem da API e a implementação segura do JWT para autenticação. Em seguida, detalharemos as camadas essenciais: *Repository*, *Controller* e *Service*, abordando métodos para criação, leitura, atualização e exclusão de dados. Por fim, discutiremos o *ExceptionHandler* para gerenciamento de exceções.

4.1 Descrição do Sistema

O aplicativo *City Connect* é uma plataforma projetada para simplificar a experiência de viajantes, fornecendo informações abrangentes sobre estabelecimentos e pontos turís-

ticos. Os usuários podem explorar avaliações e fotos, contribuindo com suas próprias opiniões. O sistema é desenvolvido utilizando o framework Java (SPRING), garantindo eficiência e escalabilidade na criação e manutenção da API. Essa escolha tecnológica foi crucial para a confiabilidade da aplicação.

4.2 Escolha do Framework

A seleção do *framework* Java (*Spring*) para o desenvolvimento do aplicativo *City Connect* se baseia em critérios técnicos fundamentais. O *Spring* oferece um ambiente de desenvolvimento flexível e modular, com destaque para a inversão de controle e injeção de dependência, facilitando a organização e testabilidade do código. A integração nativa com o *Hibernate* e a implementação do padrão MVC garantem uma arquitetura escalável e de fácil manutenção. A utilização do *Spring Boot* simplifica a configuração e acelera o desenvolvimento. O *Spring Security* assegura a implementação de políticas de autenticação e autorização de forma personalizável e segura. Com uma comunidade ativa e documentação abrangente, o *Spring* proporciona suporte consistente e soluções eficazes para desafios técnicos

4.3 Análise do Aplicativo

A fase inicial de planejamento do aplicativo *City Connect* envolveu uma análise minuciosa das classes, seus atributos e os tipos de interações esperadas com o sistema. Essa análise foi fundamental para a criação dos diagramas UML, que fornecem uma representação visual clara e detalhada do sistema.

4.3.1 Diagrama de Classes

Na figura 1, apresenta as relações entre as diferentes classes e seus atributos. Neste contexto, foram identificadas as classes principais do sistema, como Estabelecimento, Avaliação, Usuário e Endereço.

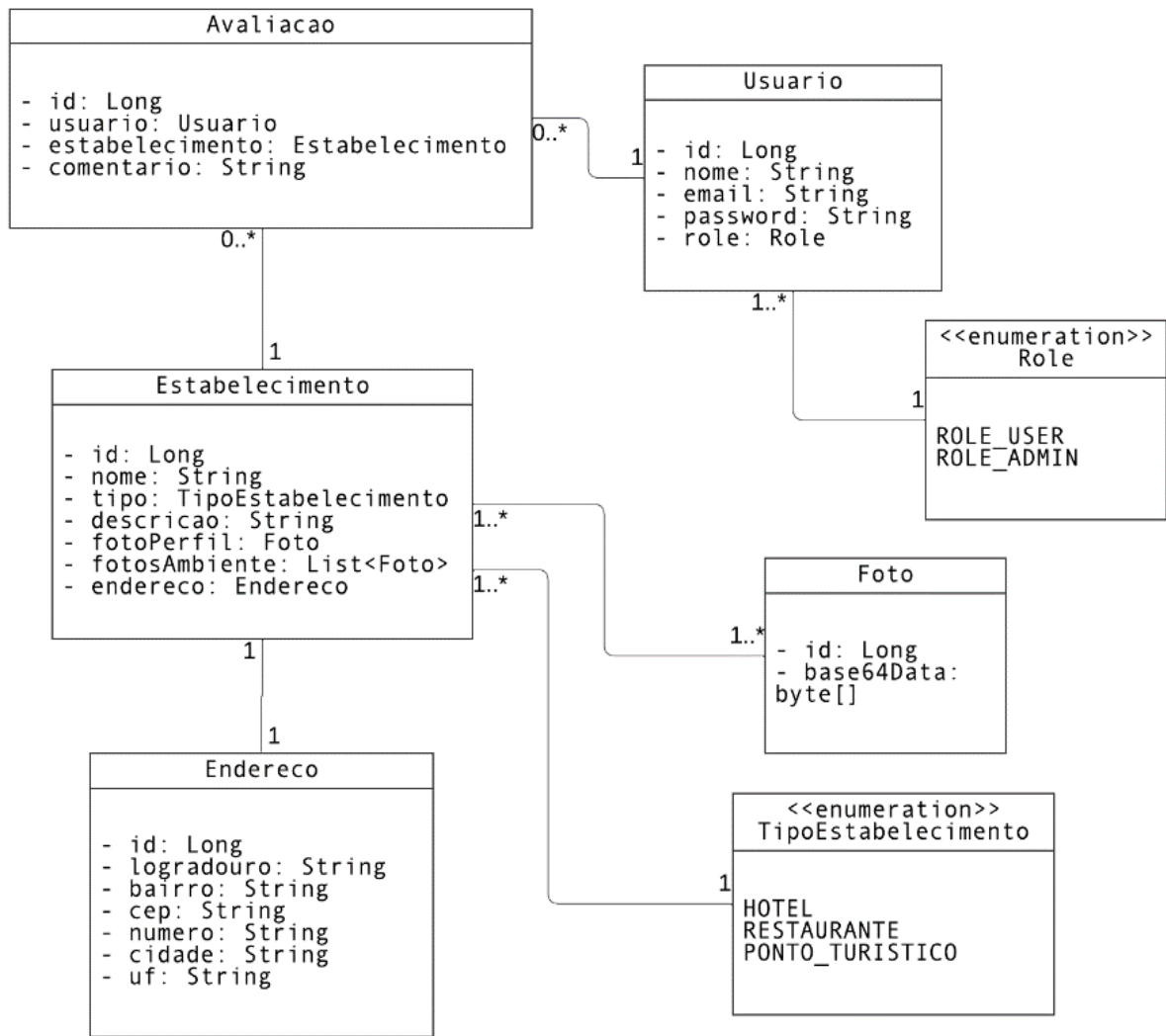


Figura 1. Diagrama de classe do aplicativo

Fonte: Autoral, 2023.

Neste diagrama de classes (Figura 1), destacam-se as relações e interações fundamentais entre as classes principais do sistema. A representação visual oferece uma visão abrangente da estrutura do sistema. As associações claramente delineadas proporcionam uma compreensão aprofundada das interações que constituem a espinha dorsal do sistema.

4.4 Implementação do JWT (JSON Web Token)

A implementação do JSON Web Token (JWT) no projeto City Connect desempenha um papel fundamental na segurança da API. Na classe *JwtTokenService*, o processo de geração e validação dos tokens é centralizado. Ao criar um token, informações cruciais como emissor, função do usuário, instante de emissão, assunto (email do usuário) e tempo de expiração são configuradas, resultando em um tempo de expiração de 1 hora. A aplicação do algoritmo de assinatura HMAC256 garante a autenticidade e integridade do token, adicionando uma camada extra de segurança.

A classe *JwtAuthenticationFilter* atua como um filtro de autenticação, inspecionando os cabeçalhos de autorização das requisições em busca de um token JWT válido. Uma vez

identificado um token válido, o filtro extrai o email do usuário do token e realiza a autenticação no contexto de segurança da aplicação. Isso garante que apenas usuários autenticados e autorizados tenham acesso aos recursos protegidos da API, proporcionando um ambiente seguro e confiável para a interação com o *City Connect*.

4.5 Docker e Containerização

No *Dockerfile* apresentado na Figura 2, delineamos cuidadosamente o ambiente de execução da aplicação no *Docker*. Inicialmente, optamos por uma imagem base incorporando JDK 17 e *Maven* para a compilação da aplicação. O processo continua com o download das dependências e a compilação do código, resultando na criação de uma imagem de produção.

```

1  # Base image com JDK 17 e Maven
2  FROM maven:3.8-openjdk-17-slim as build
3  WORKDIR /app
4  COPY pom.xml .
5  # Baixa as dependências do Maven
6  RUN mvn dependency:go-offline
7  COPY src src
8  # Compila o aplicativo
9  RUN mvn package -DskipTests
10
11 # Imagem de produção que contém apenas o JAR compilado e o Java Runtime Environment
12 FROM openjdk:17-slim
13 WORKDIR /app
14 COPY --from=build /app/target/*.jar cityconnect.jar
15
16 ENV TZ America/Fortaleza
17
18 EXPOSE 80
19 CMD ["java", "-jar", "cityconnect.jar"]

```

Figura 2. Dockerfile do aplicativo

Fonte: Autoral, 2023.

Essa imagem é essencialmente composta pelo JAR e pelo *Java Runtime Environment*, otimizada para garantir eficiência e consistência no ambiente de produção. O ajuste do fuso horário e a exposição da porta 80 são configurados, evidenciando a adaptabilidade e portabilidade da aplicação em diferentes contextos. Convido à análise atenta do *Dockerfile* subsequente para uma compreensão mais detalhada da configuração proposta.

4.6 Modelagem da API

A API é estruturada em quatro controladores principais: *AuthenticationController*, *AvaliacaoController*, *EstabelecimentoController* e *UsuarioController*.

a) *AuthenticationController*: Gerencia a autenticação de usuários através do *endpoint* POST `/api/auth`. Ele permite o login fornecendo e-mail e senha, retornando um token JWT para autenticação subsequente.

b) *AvaliacaoController*: Responsável por operações relacionadas às avaliações. Oferece a criação de novas avaliações (POST `/api/avaliacoes`) e a consulta de avaliações por estabelecimento (GET `/api/avaliacoes/estabelecimentos/{id}`) e GET `/api/avaliacoes/{idAvaliacao}/estabelecimentos/{idEstabelecimento}`).

c) *EstabelecimentoController*: Lida com operações relacionadas aos estabelecimentos. Permite a criação de novos estabelecimentos (POST /api/estabelecimentos), a associação de fotos de ambientes (POST /api/estabelecimentos/{idEstabelecimento}/adicionarFotoAmbiente), a busca por ID (GET /api/estabelecimentos/{id}) e a listagem de todos os estabelecimentos (GET /api/estabelecimentos).

d) *UsuarioController*: Trata das operações com usuários, possibilitando a criação de novos usuários (POST /api/usuarios) e a consulta por ID (GET /api/usuarios/{id}). Também oferece a listagem de todos os usuários (GET /api/usuarios).

Cada controlador interage com os respectivos serviços (*AvaliacaoService*, *EstabelecimentoService* e *UsuarioService*) para executar as operações necessárias. Esses serviços estendem a classe genérica *GenericService*, proporcionando métodos como listar todos, buscar por ID e salvar, facilitando a manipulação dos dados.

4.7 Repository

A camada de *Repository* desempenha um papel crucial na interação entre a aplicação e o banco de dados. Em nosso projeto, cada entidade principal (Avaliação, Estabelecimento, Foto e Usuário) possui seu próprio repositório correspondente. Esses repositórios fornecem métodos para persistir e recuperar dados no banco de dados de forma eficiente. Além disso, utilizamos o *Spring Data JPA* para simplificar operações comuns, como salvar e buscar por *ID*. Essa camada facilita a organização e a eficiência na manipulação de dados, mantendo uma clara separação de responsabilidades.

4.8 Service

A camada de serviço no nosso projeto desempenha um papel essencial na implementação da lógica de negócios da aplicação. Cada serviço corresponde a uma entidade ou a um conjunto de operações relacionadas. Por exemplo, o *EstabelecimentoService* gerencia operações relacionadas a estabelecimentos, como a criação, edição e consulta desses registros. Além disso, a camada de serviço também é responsável por interagir com a camada de repositório para persistir e recuperar dados no banco de dados. Ela encapsula a complexidade da interação com o banco, permitindo que o controlador foque na manipulação das requisições HTTP.

Outro aspecto importante é que a camada de serviço é onde ocorre a validação de dados e a aplicação das regras de negócio. Por exemplo, ao criar um estabelecimento, o *EstabelecimentoService* verifica se todos os campos obrigatórios foram preenchidos corretamente, antes de persistir os dados no banco. Essa abstração da lógica de negócios em serviços facilita a manutenção e a escalabilidade da aplicação, pois permite a separação clara das responsabilidades entre as diferentes camadas do sistema.

4.9 Controller

A camada de *Controller* atua como a interface entre o cliente e a aplicação, gerenciando as requisições HTTP e coordenando a execução das operações necessárias. No nosso projeto, cada entidade principal (Autenticação, Avaliação, Estabelecimento e Usuário) possui seu próprio controlador dedicado. Utilizamos anotações do Spring, como `@Rest`

Controller e *@RequestMapping*, para mapear *endpoints* específicos e definir os métodos associados a eles. Além disso, empregamos a anotação *@PreAuthorize* para controlar o acesso aos *endpoints* com base em papéis de usuário. Essa camada desempenha um papel vital na garantia da segurança e na integração eficiente das operações fornecidas pela aplicação.

4.9.1 POST

O método POST desempenha uma função essencial em nossa aplicação, facilitando a criação eficiente e segura de novos recursos. Ilustrativamente, na Figura 3, o método “criar” do controlador *EstabelecimentoController* é um exemplo prático dessa funcionalidade. Ao receber uma requisição com os dados do novo estabelecimento, o controlador processa essas informações, garantindo sua validade e consistência por meio da anotação *@Valid*. Posteriormente, utiliza o serviço correspondente para persistir o novo estabelecimento no banco de dados, conforme figura 3.

```
@PreAuthorize("hasAuthority('ROLE_ADMIN')")
@PostMapping(consumes = {MediaType.MULTIPART_FORM_DATA_VALUE})
public EstabelecimentoResponseDto criar(@ParameterObject @Valid @ModelAttribute EstabelecimentoRequestDto entity)
    throws IOException {
    return estabelecimentoService.salvar(entity);
}
```

Figura 3. Print do código do método criar

Fonte: Autoral, 2023.

Essa abordagem robusta oferece uma maneira segura de inserir novos dados na aplicação, certificando-se de que estejam disponíveis para utilização futura. Convido à análise mais aprofundada da figura 3 para compreender melhor a implementação prática desse processo.

4.9.2 GET

Os métodos de requisição GET permitem a recuperação de informações específicas do banco de dados. No método *buscarPorId*, figura 4, do controlador *EstabelecimentoController*, utilizamos um método GET para buscar um estabelecimento pelo seu ID. Ao receber uma requisição com o ID do estabelecimento desejado, o controlador verifica se o estabelecimento existe e, em caso afirmativo, retorna os dados correspondentes. Em caso contrário, retorna uma resposta indicando que o estabelecimento não foi encontrado. Esse processo é otimizado pelo uso de consultas JPA, que permitem a busca eficiente de entidades por ID.

```
48
49 @PreAuthorize("hasAnyAuthority('ROLE_USER', 'ROLE_ADMIN')")
50 @GetMapping("/{id}")
51 public ResponseEntity<EstabelecimentoResponseDto> buscarPorId(@PathVariable Long id) {
52     Optional<Estabelecimento> entity = estabelecimentoService.buscarPorId(id);
53     if (entity.isPresent()) {
54         EstabelecimentoResponseDto estabelecimentoResponseDto = estabelecimentoMapper.estabelecimentoToestabelecimentoResponseDto(entity.get());
55         return ResponseEntity.ok(estabelecimentoResponseDto);
56     } else {
57         return ResponseEntity.notFound().build();
58     }
59 }
60
```

Figura 4. Print do código do método buscarPorId

Fonte: Autoral, 2023.

Além disso, temos outra aplicação deste tipo de requisição, como mostra a figura 5, no controlador *AvaliacaoController* possui um método GET para listar todas as avaliações de um determinado estabelecimento. Ao receber uma requisição com o ID do estabelecimento, o controlador utiliza o serviço correspondente para recuperar a lista de avaliações associadas a esse estabelecimento.

```

28
29     @PreAuthorize("hasAnyAuthority('ROLE_USER', 'ROLE_ADMIN')")
30     @GetMapping("/estabelecimentos/{id}")
31     public List<AvaliacaoResponseDto> consultarTodas(@PathVariable(value = "id") Long estabelecimentoId) {
32         return avaliacaoService.consultarTodas(estabelecimentoId);
33     }
34

```

Figura 5. Print do código do método consultarTodas

Fonte: Autoral, 2023.

A exploração mais detalhada da figura 5 proporcionará insights adicionais sobre a implementação prática desse processo de obtenção de avaliações de estabelecimentos.

5. RESULTADOS

No âmbito do nosso projeto, a fase de desenvolvimento do *back-end* está concluída. Agora, o próximo passo é integrá-lo com um front-end, que pode ser desenvolvido utilizando diferentes tecnologias como JavaScript puro, Flutter, Angular, entre outras.

Para realizar os testes, utilizamos a ferramenta *Postman*, que nos permite fazer requisições HTTP para os *endpoints* da API. Um exemplo é a requisição do tipo POST feita para a URI `http://localhost:8080/api/usuarios`, que representa o endereço do endpoint de criação de usuários. No corpo da requisição, utilizamos o formato JSON. Após o preenchimento, enviamos a solicitação para a API. Em seguida, recebemos a resposta com o código 201 *CREATED*, indicando que o usuário foi criado e armazenado no banco de dados. O corpo da resposta apresenta os objetos que foram salvos, conforme ilustrado na figura 6.

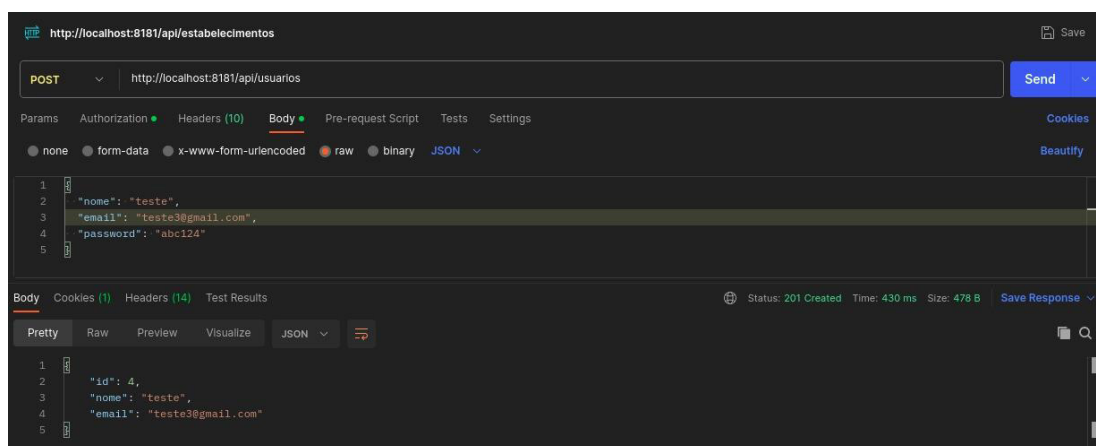


Figura 6. Post e response usuário feito no *Postman*

Fonte: Autoral, 2023.

Anteriormente, foi apresentada a operação de cadastro utilizando a aplicação *Postman*. Agora, na figura 7, é possível observar a interface de registro do aplicativo, a qual se conecta à API. Nessa tela, são exibidos os campos correspondentes aos dados necessários, permitindo que o usuário insira suas informações e conclua o cadastro. Posteriormente, esses dados são registrados no banco de dados do sistema, assegurando sua disponibilidade para uso futuro. Essa funcionalidade oferece uma forma eficiente de adicionar novas



Figura 9. Tela de Login no aplicativo

Fonte: Autoral, 2023.

Após a inserção correta, o sistema irá validar as informações, concedendo o acesso caso estejam corretas. A análise mais detalhada da Figura 9 proporciona uma visão mais aprofundada desse processo de autenticação na interface da aplicação.

6. CONSIDERAÇÕES FINAIS

Ao longo do desenvolvimento deste projeto, a API REST City Connect, fundamentada em tecnologias como Java, Spring e Docker, desempenhou um papel vital na facilitação do acesso à informação turística. Adotando metodologias de Estudo de Caso e Experimento, buscamos superar os desafios enfrentados pelos viajantes na obtenção de dados confiáveis sobre destinos.

Durante o experimento, implementamos com sucesso funcionalidades que proporcionam aos usuários acesso a informações detalhadas sobre estabelecimentos turísticos, incluindo restaurantes, hotéis e pontos turísticos. Os resultados destacam a eficácia da solução em fornecer uma experiência completa de exploração e planejamento de atividades turísticas.

Para trabalhos futuros, consideramos a expansão da API para incorporar recursos adicionais, como a introdução de um sistema de recomendação personalizado com base nas preferências do usuário e a implementação de recursos sociais para compartilhamento de experiências entre viajantes. A constante evolução e adaptação da API são cruciais para acompanhar as demandas dinâmicas do setor de turismo, garantindo que o City Connect permaneça uma ferramenta inovadora e relevante.

REFERÊNCIAS

BARRO, B. B. **O Que é uma API RESTful e Porque Isso Importa**. Disponível em: <<https://www.hostinger.com.br/tutoriais/api-restful>>. Acesso em: 4 nov. 2023.

DE PINHO ALVES, E. M. **Resumo de programação orientada a objetos**. Disponível em: <<https://www.estrategiaconcursos.com.br/blog/programacao-orientada/>>. Acesso em: 3 nov. 2023.

DE SOUZA, I. **Saiba o que é REST (Representational State Transfer) e como usá-lo neste tutorial**. Disponível em: <<https://rockcontent.com/br/blog/rest/>>. Acesso em: 4 nov. 2023.

DOS SANTOS, G. P. O.; SILVA, E. H. Desenvolvimento de solução multiplataforma para o setor agropecuário. **Perquirere**, v. 1, n. 13, p. 122–134, 2016.

FILHO, L. M. et al. **Análise da intenção de uso de aplicativos móveis no turismo: uma aplicação da teoria do comportamento planejado**. Disponível em: <<https://www.anptur.org.br/anais/anais/files/12/146.pdf>>. Acesso em: 7 nov. 2023.

HIRANO, M. **Docker e containers**. Disponível em: <<https://medium.com/tecnologia-e-afins/o-que-%C3%A9-docker-188e283088dd>>. Acesso em: 4 nov. 2023.

HORN, M. Diagrama de casos de uso UML: o que é e como fazer? Disponível em: <<https://blog.betrybe.com/tecnologia/diagrama-de-casos-de-uso-uml-o-que-e-e-como-fazer/>>. Acesso em: 5 nov. 2023.

IBM Documentation. Disponível em: <<https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>>. Acesso em: 5 nov. 2023.

JUNIOR, E. A. G.; ROCHA, R. D.; DE SOUZA MACIEL, R. **Desenvolvimento De Api Rest Com Spring Boot**. v. 15, n. 29, p. 499–525, 2021.

NEVES, C. S. F. **Engenharia de Software em Destaque: Diagramas Comportamentais da UML**. Disponível em: <<https://www.estrategiaconcursos.com.br/blog/engenharia-software-diagramas-comportamentais-uml/>>. Acesso em: 5 nov. 2023.

ROVEDA, U. **O que é programação orientada a objetos e quais são seus 4 pilares?** Disponível em: <<https://kenzie.com.br/blog/programacao-orientada-a-objetos/>>. Acesso em: 3 nov. 2023.

SEGUINS, N. **O que é JSON Web Tokens?** Disponível em: <<https://www.alura.com.br/artigos/o-que-e-json-web-tokens>>. Acesso em: 4 nov. 2023.

Spring Framework Reference Documentation. Disponível em: <<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>>. Acesso em: 4 nov. 2023.



DESENVOLVIMENTO DA PLATAFORMA GUARANI USANDO *VUE.JS* E *SPRING BOOT* PARA DIVULGAÇÃO DE PESQUISAS CIENTÍFICAS NA WEB

11

*Development of the GUARANI platform using VUE.JS and Spring
Boot for dissemination of scientific research on the web*

Bruno Emanuel Santos Lima
Edilson Carlos Silva Lima
Yonara Costa Magalhães

D.O.I.: 10.29327/5391075.1-11



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

No contexto atual, é evidente a presença crescente de tecnologias e aplicativos em variados setores, emergindo com funcionalidades diversas que visam simplificar o cotidiano das pessoas. Mesmo diante de diversas tecnologias, com a expansão da internet, o compartilhamento de informações tornou-se mais acessível. Entretanto, a profusão de conteúdo dificulta a localização de trabalhos acadêmicos específicos e respostas técnicas. Este trabalho tem como objetivo discutir e desenvolver uma plataforma web com um serviço que oferece um repositório de pesquisas científicas utilizando *vue.js* e *spring boot*. Apresenta-se aqui o desenvolvimento de uma *API REST* voltada à divulgação de trabalhos acadêmicos. Sob essa perspectiva, as tecnologias empregadas possibilitam a criação de uma aplicação web que aborda as questões propostas.

Palavras-chave: *Vue.js*, *Spring Boot*, *API REST*.

Abstract

In the current context, the growing presence of technologies and applications in various sectors is evident, emerging with diverse functionalities that aim to simplify people's daily lives. Even in the face of different technologies, with the expansion of the internet, sharing information has become more accessible. However, the profusion of content makes it difficult to find specific academic works and technical answers. This work aims to discuss and develop a web platform with a service that offers a repository of scientific research using *vue.js* and *spring boot*. Here we present the development of a *REST API* aimed at disseminating academic works. From this perspective, the technologies used enable the creation of a web application that addresses the proposed questions.

Keywords: *Vue.js*, *Spring Boot*, *API REST*



1. INTRODUÇÃO

Atualmente, a internet proporciona uma vasta fonte de pesquisa, mas a carência de ferramentas acessíveis para acompanhamento próximo cria desafios na curva de aprendizado, destacando a predominância de soluções pagas. Com a disseminação da internet, o compartilhamento de informações tornou-se mais fácil, no entanto, a abundância de conteúdo dificulta a busca por trabalhos acadêmicos específicos e respostas técnicas, levantando a questão de como superar essa dificuldade. Esses desafios inspiraram o início de um projeto voltado para oferecer soluções e estratégias que facilitem a busca por conteúdo específico. No contexto do ensino superior, onde a busca contínua por conhecimento é essencial, o acesso a materiais acadêmicos enfrenta barreiras frequentes. Diante desse cenário, continuamos a implementação da plataforma online “Guarani”, que visa facilitar a pesquisa e o armazenamento de documentos acadêmicos, proporcionando uma solução eficaz para o acesso a recursos educacionais.

Para abordar o problema identificado, foi criada uma aplicação que atua como intermediária, oferecendo aos usuários um ambiente propício para encontrar auxílio. A implementação desse projeto ocorreu na plataforma Java, utilizando o *framework Spring*. O objetivo central é fornecer uma solução eficaz para as necessidades identificadas, combinando o *front-end* em *JavaScript*, *CSS* e *Vue.js* com o *back-end* em Java, utilizando o *framework Spring*, e integrando uma *API REST* para a completa funcionalidade do sistema.

O artigo está organizado em cinco capítulos. O objetivo deste artigo é detalhar os processos e estágios envolvidos na atualização e aprimoramento da plataforma guarani. Capítulo 1 – Introdução, 2 -Trabalhos Relacionados, 3 – Fundamentação Teórica, 4 – Metodologia, 5 – Resultados e Discussão e 6 – Conclusão.

2. TRABALHOS RELACIONADOS

Neste capítulo a revisão dos trabalhos relacionados revela uma diversidade de iniciativas inovadoras no campo do desenvolvimento de plataformas e sistemas voltados para a divulgação e acesso a trabalhos científicos e acadêmicos.

2.1 O uso do *Vue.js* no desenvolvimento de uma plataforma web para a disponibilização de trabalhos científicos

Este projeto elabora uma aplicação web destinada a simplificar a consulta a trabalhos científicos, impulsionando, assim, a investigação acadêmica. Ao empregar tecnologias contemporâneas, como *Vue.js* e *Spring Boot*, a aplicação proporciona eficiência e a capacidade de reutilização de componentes. No entanto, os desafios inerentes à complexidade do desenvolvimento, aos custos associados e à necessidade de conectividade com a internet não podem ser subestimados.

Essa iniciativa almeja aprimorar a acessibilidade aos recursos acadêmicos, demandando, contudo, um esforço considerável para superar as complexidades tecnológicas e financeiras associadas, além de assegurar a utilidade da aplicação em ambientes com acesso limitado à internet.

2.2 Implementação de uma API REST usando o Spring para um sistema de divulgação acadêmica e tutoria

Este texto descreve um sistema inovador projetado para simplificar a divulgação de trabalhos acadêmicos e fomentar a colaboração entre pesquisadores por meio de funcionalidades de tutoria. As vantagens abrangem a centralização e organização de trabalhos acadêmicos, a formação de uma comunidade para compartilhamento de conhecimento e a eficácia na busca e filtragem de conteúdo acadêmico. Adicionalmente, a escolha do *framework Spring* para desenvolver uma arquitetura *REST* robusta contribui para a escalabilidade do sistema.

Entretanto, algumas desvantagens podem incluir a necessidade de promover a adoção da plataforma entre os pesquisadores, garantir a segurança dos dados e a qualidade do conteúdo compartilhado. Além disso, a manutenção e atualização contínua do sistema representam desafios que precisam ser superados para assegurar sua relevância ao longo do tempo.

2.3 Criação de aplicações web front-end com Vue.js e consumindo API com o Axios

Atualmente, com o vasto acesso à internet, existe uma extensa fonte de informações disponíveis para pesquisa. Apesar dessa abundância de conteúdo online, surgem dúvidas e a necessidade de suporte mais próximo para alguns usuários, o que cria um desafio na jornada de aprendizado. Encontrar ferramentas que ofereçam esse suporte e conectar-se com pessoas disponíveis torna-se uma tarefa difícil, especialmente considerando que a maioria das opções disponíveis são soluções pagas.

Uma aplicação focada em oferecer uma forma eficaz de adquirir novos conhecimentos ou aprimorar os existentes. Ele destaca as tecnologias utilizadas, como o *framework Vue.js* na sua segunda versão e o uso do *SASS* para estilização, além da aplicação da metodologia do *Atomic Design* para a organização e estruturação do projeto. O objetivo central desse projeto é criar uma plataforma que atue como um intermediário, conectando pessoas que buscam auxílio ou acompanhamento mais próximo em seus estudos ou pesquisas acadêmicas.

2.4 Diferencial do trabalho desenvolvido

A plataforma acadêmica “Guarani” se sobressai graças à sua abordagem ampla e à integração de tecnologias avançadas. A aplicação de linguagens modernas como *JavaScript*, *CSS*, *Vue.js* e *Java* com *Spring* proporciona uma experiência de usuário dinâmica e intuitiva no acesso aos recursos acadêmicos, enriquecendo a qualidade da experiência educacional.

Houve implementações significativas, especialmente na seção de segurança do *Spring Boot*, introduzindo mudanças relevantes e aprimoramentos notáveis. Essas atualizações podem abranger aprimoramentos na autenticação, autorização, gestão de sessões, controle de acesso, entre outros aspectos relacionados à segurança do aplicativo. A plataforma “Guarani” se destaca no cenário acadêmico de nível superior pela sinergia entre tecnologias modernas, acessibilidade e consideração dos desafios educacionais, estabelecendo-se como uma opção excepcional.



3. FUNDAMENTAÇÃO TEÓRICA

Esta seção concentra-se em uma revisão referencial, explorando áreas cruciais como ferramentas de desenvolvimento e os conceitos fundamentais associados a cada uma delas.

3.1 Desenvolvimento Web

O desenvolvimento *web* abrange um universo dinâmico de atividades, desde a concepção inicial de um projeto até sua execução e manutenção contínua. Em suma, o desenvolvimento *web* é um processo contínuo e criativo, onde a inovação e a adaptabilidade são essenciais para acompanhar o ritmo em constante evolução da internet e das demandas dos usuários. Segundo Eric Freeman e Elisabeth Robson (2013). Programar para a *web* é como criar um quebra-cabeça: você precisa colocar cada peça no lugar certo para que tudo funcione corretamente de acordo.

A maioria dos desenvolvedores *web* usa linguagem de marcação de hipertexto (*HTML*), *Cascading Style Sheets (CSS)* e *JavaScript* para desenvolver sites. Os desenvolvedores da *Web* geralmente são divididos em desenvolvedores *front-end*, desenvolvedores *back-end* e desenvolvedores *full-stack*.

3.2 Front- end

O desenvolvimento *front-end* transcende simplesmente a construção visual das interfaces; é a arte de traduzir conceitos e funcionalidades complexas em elementos intuitivos e atrativos para os usuários. Utilizando *HTML* como a espinha dorsal da estruturação do conteúdo, os desenvolvedores *front-end* moldam a base do que será visualizado. Segundo Deitel (2015), o *front-end*, também reconhecido como “interface gráfica do usuário” (GUI), corresponde à camada de um sistema de software responsável pela interação direta entre o usuário e o aplicativo, oferecendo uma interface visual que incorpora componentes como botões, caixas de texto e menus para compor a tela, facilitando interações intuitivas.

Segundo Lindley, (2012) o *front-end* é a fachada digital de uma aplicação, onde a estética se encontra com a funcionalidade para criar uma experiência envolvente que transcende o visual para se tornar uma narrativa interativa.

3.3 Vue.js

De acordo com a ideia expressa, Vue.js não busca reinventar a roda, mas sim, prioriza tornar o uso da roda mais acessível e simples (HANCHETT *et al.*, 2018). Em vez de redesenhar completamente os conceitos existentes, ele se concentra em simplificar o processo de criação de interfaces de usuário.

Vue.js é um framework JavaScript progressivo, focado na construção de interfaces de usuário interativas e reativas. Sua abordagem é pragmática, não buscando reinventar conceitos existentes, mas sim tornar o desenvolvimento mais acessível. Com uma curva de aprendizado suave, oferece uma sintaxe clara e flexível, permitindo aos desenvolvedores criar desde pequenos componentes até aplicações complexas de forma eficiente. Sua natureza incremental permite a integração fácil em projetos existentes, e sua versatilidade o torna uma escolha popular entre aqueles que buscam uma maneira ágil de construir interfaces dinâmicas e responsivas para aplicações *web*. Tem como principal objetivo o reaproveitamento de código (VILARINHO, 2021).

3.4 Back-end

O desenvolvimento de *back-end* é o núcleo lógico e operacional por trás de aplicativos e sites. Ele processa dados, gerencia solicitações do cliente e interage com bancos de dados para fornecer informações dinâmicas. O *back-end* é crucial em qualquer aplicativo, demandando a criação de algoritmos e conexões complexas.

Conforme a ideia expressa por Copeland (2016), o *backend* pode ser comparado aos bastidores de um teatro, onde o que é visível para o público representa apenas uma pequena fração do que realmente está ocorrendo. Ele utiliza linguagens de programação, *frameworks* e bancos de dados para criar a lógica por trás das funcionalidades visíveis aos usuários. As tecnologias no *back-end* permitem a comunicação entre o servidor e o cliente, garantindo a segurança e confiabilidade do sistema por meio de métodos de autenticação, autorização e criptografia. Em suma, o *back-end* é responsável por tudo o que acontece “nos bastidores” para que a aplicação funcione de maneira eficiente e segura.

3.5 Spring Boot

O *Spring Boot* simplifica a configuração e a inicialização de projetos Java ao automatizar grande parte do processo de configuração do *Spring Framework* e fornecer starters pré-configurados. Sua integração com servidores de aplicativos embutidos torna a execução de aplicativos autônomos mais simples. Além disso, oferece recursos integrados, suporte para testes automatizados e mantém a modularidade para personalizar aplicativos, sendo amplamente utilizado para construir aplicativos Java eficientes e escaláveis.

O renomado autor e especialista Martin Fowler enfatiza a importância de escolher e utilizar *frameworks* com sabedoria. O *Spring Boot*, além de impulsionar o desenvolvimento para microsserviços, também ajuda na configuração importando e configurando automaticamente todas as dependências (BOAGLIO, 2017).

3.6 REST

De acordo com Saudate (2021), uma das maneiras mais ágeis de começar um projeto de uma *API REST* em Java atualmente é por meio da utilização de algum *framework*, sendo o *Spring* o mais popular entre eles.

A Figura 1 esboça a função crucial da *API REST* como uma ponte estratégica entre o cliente e o banco de dados. Sua responsabilidade principal reside na interpretação e execução de requisições provenientes do cliente, viabilizando operações como leitura, criação, atualização e exclusão de dados em uma fonte remota, o banco de dados.

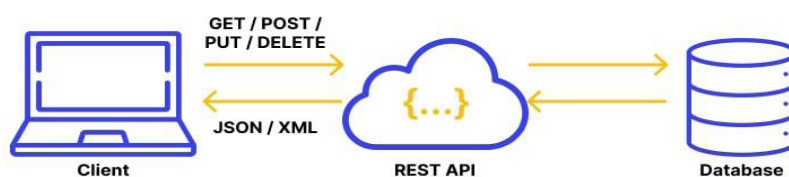


Figura 1. Cliente, API REST, Database.

Fonte: Wallarm, 2023



Esta interface simplifica a comunicação e interação entre o cliente e a fonte de dados, assegurando o processamento correto das solicitações e a execução precisa e segura das operações. Ao empregar métodos convencionais, tais como GET, POST, PUT e DELETE, a API viabiliza a manipulação dos dados conforme exigido, aderindo ao princípio do CRUD.

3.7 MySQL

O MySQL é um renomado sistema de gerenciamento de banco de dados relacional de código aberto, com ampla utilização global. Originado pela MySQL AB, uma empresa sueca, e mais tarde adquirido pela Oracle Corporation, tornou-se uma das principais escolhas para bancos de dados relacionais, especialmente em ambientes voltados para aplicações web. Segundo Vinícius Carvalho (2015), ao optar pelo MySQL como Sistema Gerenciador de Banco de Dados (SGBD), não apenas se escolhe uma ferramenta gratuita baseada em licença de software livre, mas também se opta por qualidade, robustez e segurança.

A visualização na Figura 2 oferece uma perspectiva mais abrangente e nítida sobre as características do MySQL.

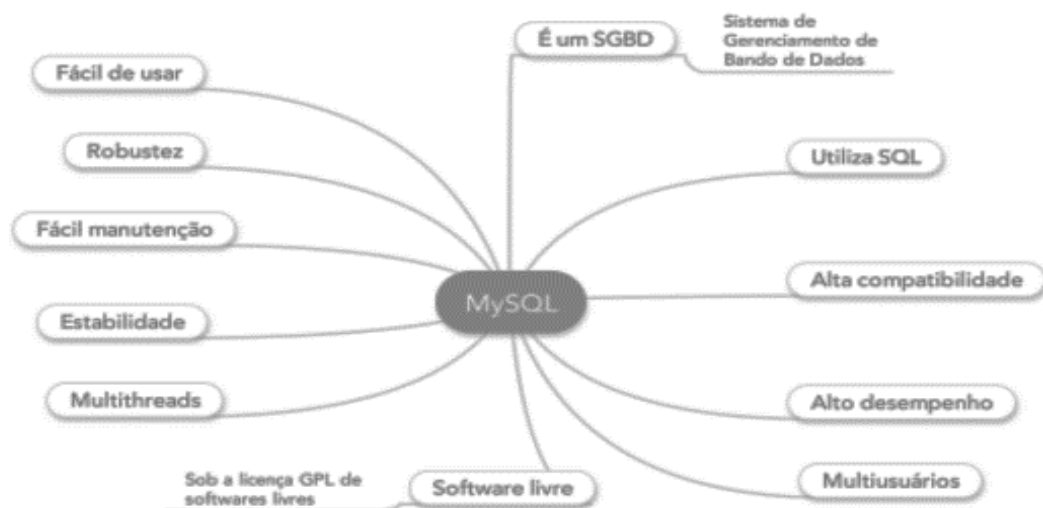


Figura 2. Características do MySQL.

Fonte. Casa do Código, 2015.

Além disso, sua comunidade ativa garante suporte consistente e integração flexível com uma ampla gama de plataformas, consolidando o MySQL como uma escolha confiável e versátil para gerenciamento de dados.

4. METODOLOGIA

A metodologia adotada neste artigo é a pesquisa de campo, a qual inclui questionamentos diretos às pessoas relevantes para os objetivos da investigação. Essa pesquisa foi conduzida por meio de um questionário abordando aspectos relacionados à plataforma desenvolvida. Em seguida, as respostas foram analisadas e apresentadas por meio de gráficos.

Buscando compreender as dificuldades e necessidades dos usuários na busca por trabalhos acadêmicos, com o objetivo de aprimorar e tornar essa experiência mais fácil e eficiente.

5. RESULTADOS E DISCUSSÃO

A base metodológica de um estudo é o alicerce sobre o qual os resultados são construídos, particularmente quando se trata do armazenamento de pesquisas acadêmicas. Nesta pesquisa, buscamos estabelecer um arcabouço sólido para compreender a influência da escolha do sistema de armazenamento no acesso, organização e preservação de dados acadêmicos.

5.1 Desenvolvimento do Front-end

Na Figura 3, é apresentada a estrutura da página 'ArtigosPage' do nosso projeto, a qual é desenvolvida utilizando o framework Vue.js.

```

1 <template>
2 <div class="artigos-container">
3   
4   <div class="artigos-formulario">
5     <v-row>
6       <v-col>
7         <v-text-field
8           height="10"
9           outlined
10          flat
11          filled
12          v-model="tituloSelecioneado"
13          label="Título / Palavra-chave"
14          color="#277049"
15        ></v-text-field>
16      </v-col>
17      <v-col>
18        <Autocomplete
19          @getItemSelecioneado="receberInstituicao"
20          itemText="nomeInstituicao"
21          :items="instituicoes"
22          label="Instituição"
23          placeholder="Selecione uma instituição"
24        />
25      </v-col>
26    </v-row>
27    <v-row>
28      <v-col>
29        <v-text-field
30          height="10"
31          outlined
32          flat
33          filled
34          v-model="pesquisadorSelecioneado"
35          label="Pesquisador proponente"

```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO **TERMINAL** PORTAS

DONE Compiled successfully in 24482ms

App running at:
 - Local: <http://localhost:8080/>
 - Network: <http://192.168.100.83:8080/>

Note that the development build is not optimized.
 To create a production build, run `yarn build`.

Figura 3. Artigos.Vue.

Fonte. Autoral, 2023.

Os componentes utilizados incluem campos de texto para inserir título, pesquisador proponente e orientador, um componente de autocomplete para selecionar instituições e áreas de conhecimento, e um campo para escolher o ano. Os métodos definidos são responsáveis por buscar informações de instituições e áreas de conhecimento a partir de serviços externos, além de filtrar e limpar os campos conforme as ações do usuário. As estilizações definidas visam estruturar e formatar os elementos na página de busca de artigos.

Essa página permite filtrar artigos com base em diversos critérios e realizar a busca correspondente. Esse código possui funções para:

- Buscar instituições e áreas de conhecimento.
- Receber seleções do usuário nos campos de filtro.
- Realizar a pesquisa de artigos baseada nos filtros aplicados.
- Limpar os campos de filtro para nova busca.
- Estruturar e organizar visualmente os elementos na página.

Na figura 3 observamos o *log* gerado após a compilação bem-sucedida no ambiente de desenvolvimento, mostrando onde ele está sendo executado localmente e na rede.

Na Figura 4 é exibida a interface de filtragem de artigos.

Figura 4. Filtragem de Artigos.

Fonte: Autoral, 2023

A tela de filtragem de artigos foi desenvolvida para receber e processar informações inseridas nos campos de busca, aprimorando a precisão na localização de conteúdos específicos. Com o propósito de simplificar a acessibilidade a um vasto acervo de conhecimento, a plataforma visa oferecer facilidade na busca, na leitura e no compartilhamento desses artigos. Ao proporcionar uma experiência intuitiva, busca-se atender às necessidades da comunidade acadêmica e dos pesquisadores, permitindo a localização rápida e eficaz de informações relevantes para seus estudos e investigações. Por meio da facilidade de acesso, a plataforma busca promover a disseminação do conhecimento, atuando como um facilitador para a pesquisa e o estudo de variadas áreas científicas.

5.2 Banco de dados

Com a utilização do software *MySQL Workbench* foi criado e configurado o banco de dados com 13 tabelas.

Na figura 3 está representada a visualização do banco de dados, sendo o *software Workbench* empregado para a completa criação e configuração deste banco, essencial para o desenvolvimento do projeto.

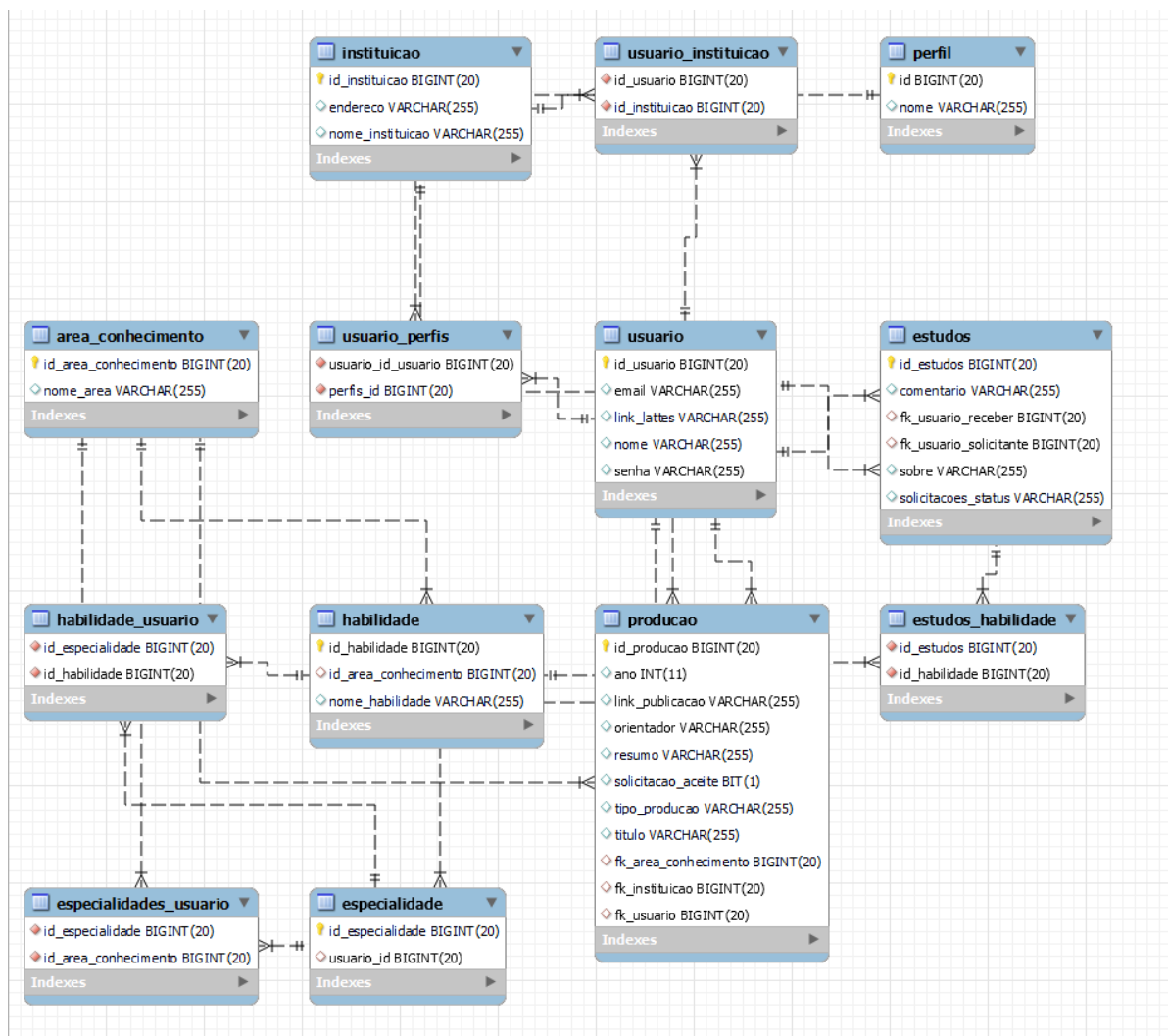


Figura 4. Banco de dados relacional do projeto

Fonte: Autoral, 2023.

Este banco vai armazenar as informações essenciais para o funcionamento e garantindo a fluidez da aplicação.

5.3 API

Foi desenvolvida a *API* para abranger as funcionalidades, estruturando-a de acordo com nosso banco de dados criado. As classes foram elaboradas utilizando a linguagem Java. A *API* desenvolvida proporciona uma gama de funcionalidades, incluindo a capacidade de executar operações essenciais, como a inserção e consulta de dados em nosso banco de dados.

```

1 package br.com.guarani.api.controller;
2
3 > import ...
30
31
32
33
34 no usages
35 @RestController
36 @RequestMapping("/usuarios")
37 @Api(value = "Usuários", tags = "Usuários")
38 public class UsuarioController {
39
40     4 usages
41     @Autowired
42     private UsuarioService usuarioService;
43     no usages
44     @ApiOperation(value = "Cadastrar Usuário")
45     @PostMapping("/cadastrarUsuario")
46     @
47     public ResponseEntity<?> cadastrar(@RequestBody @Valid UsuarioCadastro usuarioDto){
48         try{
49             Usuario usuarioSalvo = usuarioService.salvar(usuarioDto.converterCadastro());
50             return new ResponseEntity(usuarioSalvo, HttpStatus.CREATED);
51         }catch (EntidadeJaExisteException e){
52             return ResponseEntity.status(HttpStatus.CONFLICT).body(e.getMessage());
53         }
54     }
55     no usages
56     @ApiOperation(value = "Listar Usuários")
57     @GetMapping("/listarUsuario")
58     public Page<UsuarioListagemDto> listar(
59         @PageableDefault(page = 0, size = 15) Pageable paginacao){
60         Page<Usuario> pesquisadores = usuarioService.listaUsuario(paginacao);
61         return UsuarioListagemDto.converter(pesquisadores);
62     }
63
64

```

Figura 5. Classe UsuarioController

Fonte: Autoral, 2023.

A classe *controller* atua como um intermediário entre a interface do usuário e a lógica de negócios, garantindo que as solicitações sejam processadas corretamente, os dados sejam manipulados de acordo com as operações desejadas e os resultados sejam apresentados de volta ao usuário de maneira apropriada. Por meio desta classe, estabelecemos o controle para listar, buscar, adicionar e remover informações na tabela 'usuario'. Este procedimento foi replicado para as demais tabelas presentes no banco de dados relacional, assegurando consistência e padronização nas operações realizadas em todas elas.

5.4 Avaliação

Neste tópico, serão enfatizados os resultados da pesquisa conduzida, O questionário foi elaborado para coletar informações detalhadas sobre a percepção dos usuários em relação à plataforma, permitindo uma compreensão mais abrangente dos aspectos positivos e oportunidades de aprimoramento da aplicação.

Realizamos uma pesquisa de campo por meio de questionários para avaliar a experiência de uso da plataforma, com alunos da Universidade Ceuma. Apresentando o aplicativo e incentivando seu uso. Esses dados serão utilizados para extrair informações que serão discutidas por meio de gráficos.

Analisaremos os resultados obtidos na pesquisa conduzida com 10 alunos.

Recomendação da Plataforma

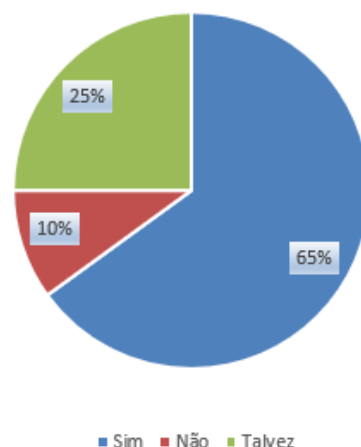


Gráfico 1. Resultado da pesquisa sobre recomendação.

Fonte: Autoral, 2023.

Esses resultados fornecem insights valiosos sobre a aceitação da plataforma. A pesquisa revelou que a maioria expressiva, com 65% dos entrevistados, estaria disposta a recomendar a plataforma, sugerindo uma aceitação considerável.

Buscando compreender as melhorias que os usuários desejariam na plataforma vamos fazer uma análise do gráfico 2, abaixo.

Sugestão de Melhorias

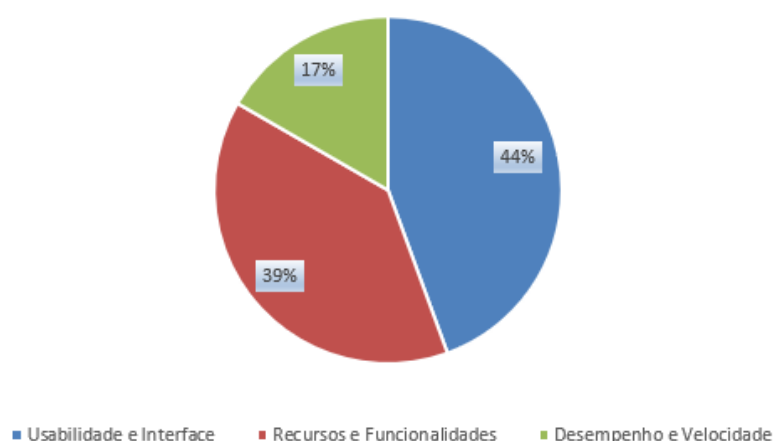


Gráfico 2. Melhorias.

Fonte: Autoral, 2023.

Esses resultados sugerem que a maioria dos participantes está focada em melhorias na interface e usabilidade da plataforma, seguida por um interesse moderado na ampliação de recursos e funcionalidades. Esta análise pode ser útil para direcionar esforços de de-

envolvimento ou aprimoramento da plataforma, priorizando as áreas de maior interesse para os usuários.

O Gráfico 3 ilustra os resultados do questionário aplicado aos 10 alunos da Universidade Ceuma que participaram da pesquisa sobre recomendação da plataforma e sugestões para melhorias. Foi solicitado que respondesse o questionário para ser analisado e discutido.

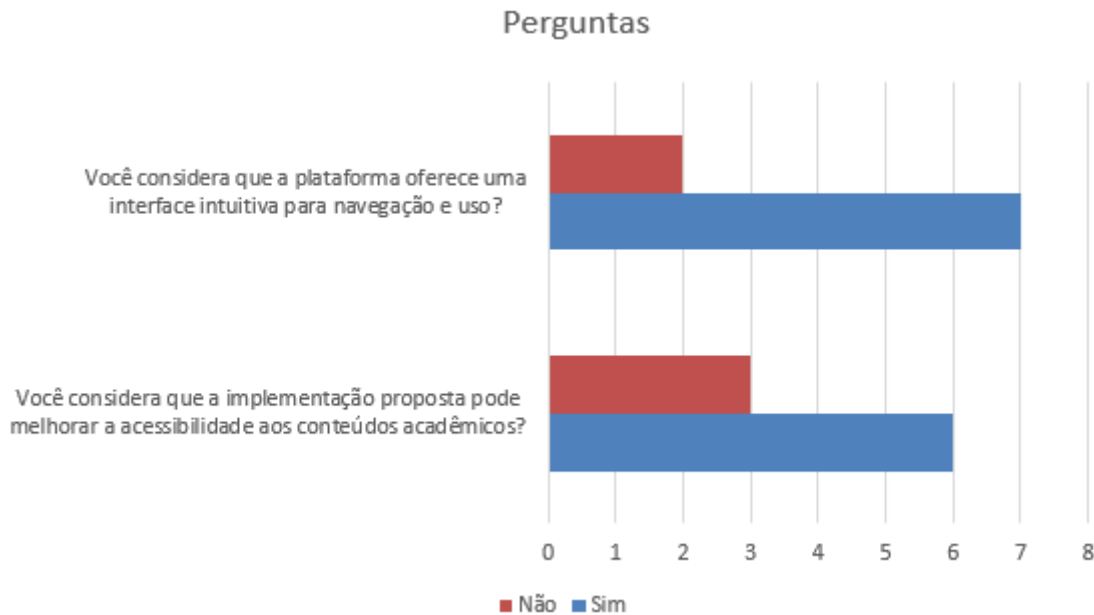


Gráfico 3. Resultado questionário.

Fonte: Autoral, 2023.

No contexto do projeto Guarani, a interface intuitiva se destaca pela sua usabilidade eficaz, oferecendo uma experiência amigável aos usuários. A partir do retorno dos usuários, destaca-se a priorização da facilidade de compreensão e de um fluxo de trabalho coerente, sempre buscando melhorar continuamente a experiência do usuário por meio de aprimoramentos na interface.

A integração de tecnologias modernas como *JavaScript*, *Vue.js* e *Java com Spring* no desenvolvimento da plataforma “Guarani” resultou em melhorias substanciais na experiência do usuário e na acessibilidade ao vasto repositório de trabalhos acadêmicos. A aplicação dessas tecnologias permitiu a criação de uma interface dinâmica e intuitiva, simplificando a busca por materiais específicos e promovendo interações entre os usuários.

No entanto, apesar dos avanços notáveis, alguns desafios foram enfrentados durante o processo de implementação. A integração harmoniosa de diferentes tecnologias e a garantia de compatibilidade entre elas foram questões complexas que demandaram tempo e esforço significativos. Além disso, a complexidade da gestão de segurança, embora tenha sido aprimorada, ainda exige constante vigilância e atualizações para acompanhar as demandas crescentes de proteção de dados e informações sensíveis.

Olhando para o futuro, a plataforma “Guarani” tem potencial para expandir suas funcionalidades, considerando a adição de recursos como recomendações personalizadas com base no histórico de pesquisa do usuário, a incorporação de ferramentas de análise de dados para aprimorar a busca e a categorização de materiais acadêmicos, bem como a exploração de tecnologias emergentes para promover uma experiência ainda mais enriquecedora para os usuários.

6. CONCLUSÃO

O artigo oferece *insights* sobre como as instituições educacionais lidam com trabalhos acadêmicos, identificando as principais dificuldades enfrentadas por aqueles que buscam acesso a esses arquivos de pesquisa. Além disso, destaca-se o papel da aplicação em facilitar a busca por auxílio próximo durante os estudos, proporcionando um ambiente de fácil utilização, recursos de apoio e promovendo a interação entre indivíduos com domínio de conhecimento. O desenvolvimento de uma *API REST* para divulgação de trabalhos acadêmicos é apresentado, fornecendo funcionalidades que simplificam a publicação e pesquisa de conteúdo acadêmico.

Além disso, oferece um ambiente propício para a troca de saberes, facilitando não apenas a absorção de informações, mas também o compartilhamento de experiências entre os usuários. Esse público-alvo, inserido no contexto do ensino superior, encontra na plataforma uma ferramenta valiosa para se aprofundar em áreas específicas, explorar novos temas e fortalecer habilidades acadêmicas e profissionais. Ao criar um espaço interativo e de colaboração, essa iniciativa não apenas promove a ampliação do conhecimento, mas também estimula a construção de uma comunidade acadêmica engajada e colaborativa.

O projeto pode evoluir na possibilidade de integração com redes sociais oferecendo a possibilidade de compartilhamento de conteúdo em redes sociais populares podendo ampliar sua visibilidade e atrair mais resultados, outra ideia seria introduzir fóruns ou grupos de estudos virtuais, onde os usuários possam discutir tópicos específicos, tirar dúvidas e colaborar, promovendo a interação entre os participantes.

REFERÊNCIAS

- BLOCH, Joshua. **Effective Java**. Addison-Wesley Professional, v. 3, f. 188, 2008. 375 p.
- BOAGLIO, Fernando. **Spring Boot: Acelere o desenvolvimento de microsserviços**. Casa do Código, v. 3, f. 94, 2017. 187 p.
- CARVALHO, Vinícius. **MySQL: Comece com o principal banco de dados open source do mercado**. Editora Casa do Código, v. 3, f. 108, 2015. 215 p.
- COPELAND, David Bryant. **Rails, Angular, Postgres, and Bootstrap: Powerful, Effective, Efficient Full-stack Web Development**. 2016.
- DEITEL, Harvey; DEITEL, Paul; DEITEL, Abbey. **Android: Como programar - 2ed**. Bookman Editora, f. 364, 2015. 728 p.
- FREEMAN, Eric; ROBSON, Elisabeth. **Use A Cabeça! Programação Em HTML5**. 2013.
- HANCHETT, Eric; LISTWON, Benjamin. **Vue.js in Action**. Pearson Professional, f. 188, 2018. 375 p.
- LINDLEY, Cody. **JavaScript Enlightenment**. “O’Reilly Media, Inc.”
- REDHAT. **O que é API?**. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em: 22 nov.2023
- RICHARDSON, Leonard; AMUNDSEN, Mike; RUBY, Sam. **RESTful Web APIs: Services for a Changing World**. “O’Reilly Media, Inc.”, v. 1, f. 202, 2013. 403 p.
- SAUDATE, Alexandre. **APIs REST: Seus serviços prontos para o mundo real**. Casa do Código, v. 3, f. 177, 2021. 353 p.
- UGO ROVEDA. **DESENVOLVIMENTO WEB: O QUE É E COMO SER UM DESENVOLVEDOR WEB**. 2020. Disponível em: <https://kenzie.com.br/blog/desenvolvimento-web/>. Acesso em: 22 nov.2023
- VILARINHO, Leonardo. **Front-end com Vue.js: Da teoria à prática sem complicações**. Editora Casa do Código, v. 3, f. 114, 2021. 228 p.





12

DESENVOLVIMENTO DE APLICATIVO PARA GERENCIAMENTO DE FINANÇAS PESSOAIS

*Development of Financial Applications: Innovation and Usability
in Personal Finance Management*

Thiago Vinicius Araujo de Souza
Edilson Carlos Silva Lima
Jonathan Araújo Queiroz

D.O.I.: 10.29327/5391075.1-12



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Em um cenário econômico global desafiador, a gestão financeira eficaz tornou-se crucial. Nesse contexto, foi desenvolvido o aplicativo “Poupe”, uma solução mobile intuitiva e altamente eficiente, projetada para otimizar o controle de entradas e saídas financeiras. Este aplicativo, construído em Flutter, destaca-se por sua facilidade de uso, sendo uma ferramenta valiosa tanto para fins comerciais quanto pessoais. O objetivo principal do projeto vai além da simples gestão financeira; ele visa educar e conscientizar os usuários sobre seus gastos, sejam eles necessários ou supérfluos. A aplicação não apenas registra transações, mas também fornece insights perspicazes por meio de gráficos e relatórios detalhados. Ao oferecer uma visão clara dos padrões de gastos, o aplicativo capacita os usuários a tomar decisões financeiras mais informadas.

Palavras-chave: *Flutter*, aplicativo, mobile.

Abstract

In a challenging global economic scenario, effective financial management has become crucial. In this context, the “Poupe” application was developed, an intuitive and highly efficient mobile solution, designed to optimize the control of financial inflows and outflows. This application, built in Flutter, stands out for its ease of use, being a valuable tool for both commercial and personal purposes. The main objective of the project goes beyond simple financial management; it aims to educate and make users aware of their expenses, whether necessary or unnecessary. The application not only records transactions but also provides insightful insights through detailed charts and reports. By offering a clear view of spending patterns, the app empowers users to make more informed financial decisions.

Keywords: Flutter, application, mobile.



1. INTRODUÇÃO

A evolução do desenvolvimento de aplicativos financeiros reflete um cenário onde a tecnologia e a necessidade de gerenciamento eficiente de finanças pessoais convergem. A contextualização deste tema se faz cada vez mais presente na sociedade contemporânea, onde a praticidade e a segurança são requisitos fundamentais. A integração de funcionalidades inovadoras, como categorização automática de despesas e projeções financeiras, exemplifica a transformação digital em curso (Makalew, 2022). A usabilidade desses aplicativos torna-se um diferencial competitivo, influenciando diretamente na experiência do usuário e na sua capacidade de gerir suas finanças com autonomia e precisão.

O problema de pesquisa emerge da necessidade de entender como a inovação em aplicativos financeiros pode melhorar o gerenciamento de finanças pessoais. A questão central é determinar quais características de design e funcionalidade contribuem para uma experiência de usuário otimizada, promovendo uma gestão financeira mais eficaz e intuitiva. Identificar esses elementos é crucial para o desenvolvimento de soluções que não apenas atendam às necessidades dos usuários, mas também os eduquem financeiramente.

As hipóteses levantadas sugerem que aplicativos com interfaces intuitivas e recursos de personalização promovem uma melhor gestão financeira pessoal. Além disso, presume-se que a integração de tecnologias como inteligência artificial e aprendizado de máquina possa prever tendências de gastos e oferecer recomendações proativas. Essas hipóteses buscam estabelecer uma relação direta entre a inovação tecnológica e a eficiência no gerenciamento de finanças pessoais.

O objetivo geral deste estudo é analisar como a inovação e a usabilidade em aplicativos financeiros impactam o gerenciamento de finanças pessoais. Os objetivos específicos incluem avaliar a eficácia de diferentes interfaces de usuário, entender o papel da personalização na experiência do usuário e investigar o impacto da tecnologia na previsão de comportamentos financeiros. Estes objetivos visam fornecer insights valiosos para o desenvolvimento de aplicativos mais eficientes.

A metodologia adotada é a revisão bibliográfica, que permite uma análise abrangente e sistemática da literatura existente sobre o desenvolvimento de aplicativos financeiros. Este método facilita a identificação de tendências, lacunas de conhecimento e oportunidades para inovação no campo. Através desta abordagem, é possível construir uma base sólida para recomendações práticas e teóricas. Em seguida, será feita uma análise exploratória sobre um aplicativo

A justificativa para esta pesquisa reside na crescente dependência de ferramentas digitais para o gerenciamento de finanças pessoais. Com o aumento do acesso à internet e a popularização de dispositivos móveis, os aplicativos financeiros tornaram-se ferramentas essenciais no dia a dia dos usuários (Giudici; Paleari, 2000). Entender como a inovação e a usabilidade podem ser aprimoradas é fundamental para atender às expectativas e necessidades dos usuários.

A relevância profissional deste estudo é evidente, pois fornece insights valiosos para desenvolvedores, designers e profissionais de finanças que buscam criar ou aprimorar aplicativos financeiros. As descobertas podem orientar a criação de soluções mais eficazes, melhorando a saúde financeira dos usuários e promovendo a inclusão financeira (Birch; Chiappetta; Artyushina, 2020). Além disso, a pesquisa contribui para o desenvolvimento de

práticas recomendadas no campo da tecnologia financeira.

Do ponto de vista acadêmico, o estudo enriquece a literatura sobre tecnologia da informação e comportamento do consumidor. Ao explorar a interseção entre inovação, usabilidade e finanças pessoais, a pesquisa oferece uma contribuição significativa para o entendimento de como as pessoas interagem com a tecnologia financeira (King, 1993). Isso pode levar a avanços teóricos e metodológicos no estudo do design de interação e da psicologia do consumidor.

Sendo assim, o desenvolvimento de aplicativos financeiros é um campo dinâmico e em constante evolução, onde a inovação e a usabilidade desempenham papéis cruciais. Este estudo busca desvendar como esses fatores influenciam o gerenciamento de finanças pessoais, fornecendo uma base para futuras inovações que possam beneficiar usuários em todo o mundo. Através de uma metodologia rigorosa e uma análise detalhada, espera-se contribuir tanto para a prática profissional quanto para o conhecimento acadêmico no campo da tecnologia financeira.

2. TRABALHOS RELACIONADOS

Para criar o artigo, foi feito um estudo bibliográfico, os quais serão citados nos próximos tópicos. Para a problemática abordada, artigos de controle de finanças têm sido amplamente explorados por acadêmicos e profissionais, refletindo a importância da gestão financeira pessoal de empresas e pessoas. Entre esses trabalhos, destacam-se a pesquisa de um “bolso virtual” (2.1), que aborda o problema da organização financeira e dificuldades nas declarações de imposto. Também há um artigo que trata sobre controle financeiro e orientação de investimento (2.2) e, por último, um artigo mais atual, produzido pós-pandemia, que discute finanças pessoais e os desafios atuais para estruturar um aplicativo financeiro e pedagógico (2.3). A principal ideia foi trazer um aplicativo móvel que promova o controle financeiro, solucionando problemas abordados nos trabalhos relacionados.

2.1 Bolso Virtual: Aplicação Web Móvel

O trabalho apresentado por Max Brayner, foi feito um levantamento na qual a problemática era o controle de dinheiro e com a evolução da tecnologia os sistemas tem ajudado as pessoas na falta de ORGANIZAÇÃO FINANCEIRA e dificuldades na DECLARAÇÃO DE IMPOSTOS, na qual exige tempo e ele encontrou uma forma de reduzir esse tempo através do seu aplicativo web e um aplicativo Android.

No qual mesmo cumprindo com os objetivos propostos nas problemáticas precisa de melhorias para aprimorar a utilização do sistemas, tais como: Novas pesquisas para atualização do sistema, integração com cartões de crédito e débito, inclusão de lançamento na a partir da entrada e saída e exportações de dados para DIRF que é onde seriam importados para a receita federal.

2.2 Controle financeiro e orientação de investimentos: proposta de modelo de aplicativo para celulares

O trabalho apresentado por De Paula *et al.* (2021), tratar-se à falta de ética de administração de movimentos financeiros: Controle de receita, nível de conhecimento em finan-

ças e educação financeira básica. Na qual a importância da educação financeira é essencial para tomadas de decisões e planejamentos e assim criando a solução do problema foi buscar ajuda em aplicativos de controle de finanças.

Por ser um artigo pré-pandemia do Covid-19, todas as pesquisas foram feitas antes do acontecimento, então surgiram novos desafios, especialmente para os leitores e futuros projetos e desenvolvimentos de tecnologias; No entanto precisam fazer uma nova pesquisa de campo pois necessitam ser mais ágeis e mais atuais para que a orientação sobre investimento e compreensão da educação financeira seja suprida através de aplicativos baseadas no artigo produzido.

2.3 Finanças Pessoais e Desafios Atuais: uma Pesquisa com Estudantes Universitários para Estruturação de APP Financeiro e Pedagógico

O trabalho apresentado por Dos Santos *et al.* (2022) possui um conteúdo mais atual e mais atualizado. Nesse artigo foram em busca de desafios atuais em uma pesquisa com universitários para fazer uma estruturação de um app financeiro e pedagógico, na qual fizeram uma análise na gestão financeira pessoal dos estudantes de graduação, no qual sofreram com a renda antes e depois da pandemia do covid-19 é tendo como resposta para seus problemas a aceitação de um app tecnológico para controlar as finanças pessoais.

É as metodologias usadas foram: Etapas de classificação de pesquisa de campo e estratégias e análise de dados. Nos quais os dados obtidos foram elaborados um projeto de um app no qual baixava seus endividamentos e controlavam e lhes deixavam financeiramente mais éticos e controlados.

2.4 Diferencial do projeto desenvolvido

Nos trabalhos abordados acima, todos eles têm como objetivo controlar apenas uma conta de cada vez. No “Poupe”, podem ser controladas várias contas simultaneamente e também são mostrados dados em forma de gráficos ao longo do ano ou do mês, conforme o usuário preferir. Além de ter uma compreensão mais realista e também mais atualizada dos acontecimentos atuais sobre as complexidades financeiras modernas, onde é possível controlar contas bancárias, investimentos, entradas e saídas, o aplicativo consolida a visão do usuário, auxiliando em seu panorama financeiro global.

Além disso, a apresentação dos dados em forma de gráficos para diferentes períodos, seja no ano ou em escala mensal, destaca a ênfase no protótipo do “POUPE”. Isso pode oferecer aos usuários “insights” mais claras sobre seus padrões de gastos, economias e investimentos ao longo do tempo, facilitando as tomadas de decisões informadas.

Essa abordagem torna-se mais flexível e abrangente no “POUPE”, não apenas atendendo às necessidades práticas dos usuários, mas também alinhando-se às tendências emergentes em aplicativos de controle financeiro, que buscam cada vez mais oferecer soluções personalizadas e intuitivas. Além de ser um protótipo, os autores estão sempre dispostos a ouvir novas ideias e implementá-las para melhorar ainda mais o aplicativo, proporcionando uma experiência de gestão financeira mais completa e adaptável aos problemas e desafios do dia a dia dos usuários.

3. FUNDAMENTAÇÃO TEÓRICA

Nesse tópico foi feito um estudo sobre aplicativos com funções financeiras(3.1) para melhor entendimento do aplicativo e também para implementação do código foi feito um estudo sobre o *flutter* e sua linguagem *dart*(3.2), e para o back-end e o banco de dados foi utilizado o *firebase* (3.3) e para complementarmos mais ainda foi preciso ter uma noção de inovação tecnológica (3.4) e de gerenciamento de finanças pessoais(3.5).

3.1 Aplicativos com funções financeiras

Os aplicativos financeiros, conforme descrito por Makalew (2022), são ferramentas projetadas para facilitar a gestão de finanças pessoais. Eles oferecem uma variedade de funcionalidades, desde o rastreamento de despesas até a projeção de orçamentos futuros. A definição desses aplicativos não se limita apenas à sua funcionalidade, mas também à sua capacidade de fornecer uma experiência de usuário intuitiva e personalizada. A característica distintiva é a sua acessibilidade, permitindo que os usuários gerenciem suas finanças em tempo real, o que reflete uma mudança significativa na maneira como as pessoas interagem com suas finanças.

No mercado atual, existem diversos tipos de aplicativos financeiros, cada um atendendo a diferentes necessidades dos usuários. Xie (2016) destaca que, desde aplicativos de orçamento até plataformas de investimento, a variedade disponível permite que indivíduos de todos os níveis de habilidade financeira encontrem ferramentas adequadas às suas necessidades. Essa diversidade é essencial para promover a inclusão financeira e a autonomia, permitindo que cada usuário escolha soluções que se alinhem com seus objetivos financeiros pessoais.

Os componentes essenciais de aplicativos financeiros, como identificado por Grelita, Suzianti e Ardi (2017), incluem uma interface de usuário limpa e funcionalidades que promovem a usabilidade. A simplicidade no design e a facilidade de navegação são fundamentais para que os usuários possam realizar suas tarefas financeiras com eficiência. Além disso, a segurança dos dados é um aspecto crítico, pois os usuários precisam confiar que suas informações financeiras estão protegidas.

A inovação em aplicativos financeiros não se restringe apenas à sua funcionalidade, mas também à forma como eles são desenvolvidos e atualizados. King (1993) argumenta que o crescimento financeiro está intrinsecamente ligado à inovação. Os aplicativos financeiros modernos devem, portanto, incorporar as mais recentes tecnologias para oferecer soluções avançadas e adaptativas que atendam às demandas em constante mudança dos usuários.

A usabilidade dos aplicativos financeiros é um fator que Kishkovich (2021) considera crucial para a adoção e eficácia dessas ferramentas. Um aplicativo que é difícil de usar ou entender pode desencorajar os usuários de gerenciar suas finanças, enquanto uma experiência de usuário positiva pode promover hábitos financeiros saudáveis e uma melhor compreensão da saúde financeira pessoal.

Giudici e Paleari (2000) enfatizam a importância do financiamento à inovação, especialmente em pequenas empresas tecnológicas. Isso é igualmente aplicável ao desenvolvimento de aplicativos financeiros, onde o investimento em novas tecnologias pode levar a avanços significativos na forma como os aplicativos são projetados e funcionam.

A inovação tecnológica, segundo Birch, Chiappetta e Artyushina (2020), também le-

vanta questões sobre a propriedade e o uso de dados pessoais. No contexto dos aplicativos financeiros, isso se traduz na necessidade de equilibrar a inovação com a privacidade do usuário, garantindo que os dados sejam usados de maneira ética e responsável.

Ayyagari, Demirguc-Kunt e Maksimovic (2011) discutem o papel da inovação empresarial em mercados emergentes, destacando que o acesso ao financiamento e um ambiente de governança robusto são fundamentais para o sucesso. Isso se aplica ao desenvolvimento de aplicativos financeiros, onde a capacidade de inovar é muitas vezes determinada pela disponibilidade de recursos e pela estrutura regulatória em que as empresas operam.

A usabilidade e a inovação em aplicativos financeiros são, portanto, interdependentes. A inovação impulsiona a usabilidade, e a usabilidade pode, por sua vez, incentivar a inovação contínua. À medida que os aplicativos financeiros se tornam mais sofisticados, eles têm o potencial de transformar ainda mais o gerenciamento de finanças pessoais, tornando-o mais acessível, eficiente e seguro para os usuários em todo o mundo.

3.2 Flutter

Flutter é um framework UI incrivelmente poderoso que permite criar experiências nativas belas em todas as plataformas móveis. Seidel (2018), o Flutter também é um framework de código aberto desenvolvido pelo Google para a criação de aplicativos nativos em dispositivos móveis, web e desktop, tudo a partir de uma única base de código. Ao usar a linguagem de programação Dart, o *Flutter* oferece uma maneira rápida e eficiente de criar interfaces bonitas e altamente responsivas, e foi usado para o desenvolvimento do “Poupe”. Pois assim sua arquitetura permite atualizações em tempo real, o que significa que as mudanças feitas no código são instantaneamente refletidas na interface do aplicativo. Essa característica torna o Flutter uma escolha popular entre os desenvolvedores que procuram uma forma eficiente de criar aplicativos multiplataforma com alta qualidade.

No flutter com grande parte do aplicativo feito em flutter, tudo que o usuário interage na aplicação é com widgets é os widgets são os blocos de construção fundamentais para criar interfaces de usuário. Eles são componentes visuais, como botões, textos, caixas de texto e até mesmo layouts inteiros. Widgets são facilmente combináveis para criar a aparência e o comportamento desejados em seu aplicativo.

3.3 Firebase

Para todo controle do banco de dados do protótipo foi feito um estudo sobre o *firebase*, que é uma ferramenta do google na qual auxilia no gerenciamento e na proteção do banco de dados. O mesmo fornece aos desenvolvedores uma forma para conectar suas aplicações mobile e web a serviços na nuvem a partir de APIs e SDKs Batschinski (2016). O *firebase* também é responsável por conservar e sincronizar informações em tempo real. No qual é um conjunto de serviços de computação em nuvem de back-end e plataformas de desenvolvimento de aplicativo ele também hospeda bancos de dados, serviços, autenticação e integração para uma variedade de aplicativos, incluindo Android, iOS, JavaScript, Node.js, Java, Unity, PHP e C++.

- **Facilidade de Integração:** O Firebase oferece SDKs para várias linguagens de programação, incluindo Objective-C, Java, JavaScript, Swift, Node.js e Java. Independentemente da plataforma que você está utilizando, integrar o Firebase em seu aplicativo é simples devido à sua variedade de SDKs. A documentação completa

e amigável do Firebase é uma ferramenta valiosa que torna mais fácil utilizar seus serviços, oferecendo instruções precisas.

- **Escalabilidade Automática:** Através da utilização da infraestrutura do Google Cloud, o Firebase garante escalabilidade automática com base nas demandas do seu aplicativo. Com esse método, as preocupações com a gestão de servidores e a adição de mais recursos à medida que o aplicativo cresce são eliminadas. A escalabilidade é gerenciada suavemente pelo Firebase, permitindo que você se concentre no desenvolvimento de aplicativos.
- **Ampla Variedade de Serviços:** O Firebase oferece uma infinidade de serviços cruciais para o desenvolvimento de software, como push, banco de dados e autenticação.

3.3 Inovação Tecnológica

A inovação tecnológica no desenvolvimento de aplicativos financeiros é um vetor crucial para a transformação digital no gerenciamento de finanças pessoais. Conceitualmente, a inovação em TI abrange a criação e a aplicação de ideias novas e melhoradas em produtos, processos e serviços (Birch; Chiappetta; Artyushina, 2020). No contexto dos aplicativos financeiros, isso se traduz na implementação de interfaces intuitivas, algoritmos de segurança robustos e funcionalidades que simplificam e enriquecem a experiência do usuário. A inovação não é apenas uma questão de novidade, mas também de aplicabilidade prática e melhoria contínua.

Recentemente, os aplicativos financeiros têm incorporado inovações como inteligência artificial e aprendizado de máquina para fornecer análises preditivas e personalizadas (Ayyagari; Demirguc-Kunt; Maksimovic, 2011). Essas tecnologias permitem que os aplicativos aprendam com os hábitos dos usuários, oferecendo recomendações personalizadas e alertas proativos sobre oportunidades de investimento ou riscos potenciais. A integração dessas inovações está redefinindo a maneira como os usuários interagem com suas finanças, tornando a gestão financeira mais acessível e menos intimidadora.

O impacto da inovação na adoção de novas tecnologias é significativo. Quando os aplicativos financeiros incorporam inovações tecnológicas, eles não apenas melhoram a eficiência, mas também aumentam a confiança dos usuários nas plataformas digitais (Giudici; Paleari, 2000). A adoção de novas tecnologias em aplicativos financeiros pode levar a um aumento na inclusão financeira, permitindo que mais pessoas tenham acesso a ferramentas de gestão financeira que anteriormente eram inacessíveis.

A inovação em aplicativos financeiros também está intimamente ligada à usabilidade. Um design inovador deve sempre considerar a facilidade de uso para garantir que os avanços tecnológicos sejam efetivamente adotados pelos usuários (Gretalita; Suzianti; Ardi, 2017). Isso significa que os desenvolvedores devem equilibrar a introdução de novas funcionalidades com uma interface clara e um fluxo de navegação que os usuários possam seguir intuitivamente.

A inovação tecnológica também desempenha um papel fundamental na personalização da experiência do usuário em aplicativos financeiros. Makalew (2022) salienta que a personalização não se limita à customização visual, mas também à adaptação do comportamento do aplicativo às necessidades individuais do usuário. Isso pode incluir a configuração de orçamentos personalizados, o rastreamento de gastos em categorias específicas e a recepção de conselhos financeiros baseados em padrões de gastos pessoais.



A segurança é outra área onde a inovação tecnológica é vital. Com a crescente preocupação com a privacidade e a segurança dos dados, os aplicativos financeiros estão inovando com tecnologias como a autenticação biométrica e a criptografia de ponta a ponta para proteger as informações dos usuários (Kishkovich, 2021). Essas inovações não só protegem contra o acesso não autorizado, mas também reforçam a confiança do usuário na plataforma.

A inovação tecnológica também tem um impacto direto na acessibilidade dos aplicativos financeiros. O desenvolvimento de interfaces que suportam vários idiomas e que são adaptáveis a diferentes dispositivos aumenta o alcance desses aplicativos, tornando-os acessíveis a uma base de usuários mais ampla e diversificada (Xie, 2016). Isso é particularmente importante em mercados emergentes, onde o acesso a serviços financeiros tradicionais pode ser limitado.

Além disso, a inovação em aplicativos financeiros está impulsionando a mudança em direção a uma maior transparência financeira. Com recursos que permitem aos usuários rastrear suas despesas em tempo real e receber insights sobre seu comportamento financeiro, os aplicativos estão promovendo uma maior consciência financeira (King, 1993). Isso não apenas ajuda os usuários a gerenciar melhor suas finanças, mas também pode levar a decisões financeiras mais informadas e responsáveis.

A colaboração entre instituições financeiras e desenvolvedores de tecnologia é essencial para impulsionar a inovação em aplicativos financeiros. Parcerias estratégicas podem levar ao compartilhamento de conhecimentos e à criação de soluções que atendam às necessidades emergentes dos consumidores (Giudici; Paleari, 2000). Essa colaboração é fundamental para garantir que os aplicativos financeiros continuem a evoluir e a atender às expectativas dos usuários.

Por fim, a inovação tecnológica em aplicativos financeiros é um campo dinâmico que requer uma abordagem holística. Não se trata apenas de implementar a tecnologia mais recente, mas de entender como essa tecnologia pode melhorar a vida financeira dos usuários (Birch; Chiappetta; Artyushina, 2020). À medida que a tecnologia avança, os aplicativos financeiros devem continuar a inovar para oferecer soluções que sejam não apenas tecnicamente avançadas, mas também significativas e valiosas para os usuários.

3.4 Gerenciamento de Finanças Pessoais

O gerenciamento de finanças pessoais é uma área que tem se beneficiado enormemente da inovação tecnológica, especialmente através do desenvolvimento de aplicativos. Estes aplicativos oferecem uma variedade de ferramentas e técnicas que auxiliam os indivíduos a planejar e gerenciar suas finanças de maneira mais eficaz. Segundo Birch, Chiappetta e Artyushina (2020), a inovação tecnológica tem permitido que dados pessoais sejam transformados em ativos privados, o que pode ser aplicado ao gerenciamento financeiro pessoal, permitindo uma personalização e uma gestão de dados sem precedentes. As teorias de gerenciamento financeiro pessoal enfatizam a importância da educação financeira e do comportamento econômico prudente, aspectos que podem ser reforçados pelo uso de aplicativos.

As ferramentas e técnicas de planejamento financeiro disponíveis em aplicativos modernos são diversas e vão desde simples orçamentos até complexos modelos de previsão financeira. Makalew (2022) destaca que o design e o desenvolvimento de aplicativos baseados em Android para a gestão de finanças pessoais têm se concentrado em melhorar a

experiência do usuário, tornando o planejamento financeiro mais acessível e menos intimidador. Essas ferramentas digitais permitem que os usuários acompanhem suas despesas, estabeleçam metas de economia e visualizem seu progresso financeiro ao longo do tempo.

Os benefícios dos aplicativos no gerenciamento financeiro são notáveis. Eles oferecem conveniência e acessibilidade, permitindo que os usuários gerenciem suas finanças a qualquer hora e em qualquer lugar. Conforme observado por Ayyagari, Demirguc-Kunt e Maksimovic (2011), a inovação em mercados emergentes, incluindo o desenvolvimento de aplicativos financeiros, desempenha um papel significativo na promoção da governança e na competição no mercado financeiro. Isso, por sua vez, tem um impacto positivo no gerenciamento financeiro pessoal, pois oferece aos usuários uma variedade de opções e ferramentas para melhorar sua saúde financeira.

A usabilidade e a inovação dos aplicativos financeiros têm um impacto direto na eficácia do gerenciamento de finanças pessoais. Gretalita, Suzianti e Ardi (2017) ressaltam a importância do design visual na usabilidade dos aplicativos financeiros. Um design bem pensado não só torna o aplicativo mais atraente, mas também mais funcional, o que é essencial para manter os usuários engajados e comprometidos com o gerenciamento de suas finanças.

A implementação de sistemas de gerenciamento financeiro pessoal baseados em Android, como discutido por Xie (2016), demonstra a capacidade da tecnologia de simplificar processos financeiros complexos. Esses sistemas permitem que os usuários integrem todas as suas informações financeiras em um único local, facilitando a visualização e o gerenciamento de suas finanças. A integração de várias contas e a capacidade de acompanhar transações em tempo real são apenas alguns exemplos de como os aplicativos podem simplificar o gerenciamento financeiro.

A inovação em aplicativos financeiros também tem um impacto significativo na inclusão financeira. King (1993) argumenta que a financeirização da economia pode contribuir para o crescimento econômico, e os aplicativos financeiros desempenham um papel importante nesse processo ao tornar as ferramentas de gerenciamento financeiro acessíveis a uma base de usuários mais ampla. Isso é especialmente importante para aqueles que podem ter sido excluídos dos serviços financeiros tradicionais.

A segurança dos dados é uma consideração primordial no desenvolvimento de aplicativos financeiros. Kishkovich (2021) enfatiza a importância de modelos conceituais de sistemas de gerenciamento financeiro que incorporam tecnologias de informação modernas para garantir a segurança dos dados dos usuários. A confiança é um componente essencial no uso de aplicativos financeiros, e a segurança robusta é fundamental para manter essa confiança.

A colaboração entre instituições financeiras e desenvolvedores de tecnologia é crucial para o avanço dos aplicativos financeiros. Giudici e Paleari (2000) observam que o fornecimento de financiamento para inovação é vital para pequenas empresas baseadas em tecnologia, e isso se estende ao desenvolvimento de aplicativos financeiros. Parcerias estratégicas podem levar a inovações que beneficiem tanto as instituições financeiras quanto os usuários finais.

Sendo assim, o desenvolvimento de aplicativos financeiros é um campo dinâmico que exige uma compreensão profunda das necessidades dos usuários e da tecnologia disponível. Birch, Chiappetta e Artyushina (2020) argumentam que a inovação no capitalismo tecnocientífico apresenta desafios, mas também oportunidades significativas para o gerenciamento de finanças pessoais. Os aplicativos financeiros devem continuar a evoluir

para atender às demandas dos usuários e aproveitar as novas tecnologias para melhorar a saúde financeira dos indivíduos.

Portanto, o gerenciamento de finanças pessoais através de aplicativos financeiros representa uma fusão de inovação tecnológica e usabilidade prática. Ayyagari, Demircug-Kunt e Maksimovic (2011) destacam que a inovação em mercados emergentes é impulsionada pelo financiamento, governança e competição, o que é diretamente aplicável ao desenvolvimento de aplicativos financeiros. À medida que esses aplicativos se tornam mais sofisticados e centrados no usuário, eles têm o potencial de transformar o gerenciamento de finanças pessoais, tornando-o mais eficiente, seguro e inclusivo.

4. METODOLOGIA DE DESENVOLVIMENTO DO APLICATIVO

Este conjunto diversificado de tópicos abrange aspectos cruciais do desenvolvimento do aplicativo, proporciona a metodologia usada para o desenvolvimento do aplicativo (4.1). O diagrama de caso de uso (4.2) oferece uma representação visual das interações entre usuários e sistema, fornecendo uma base sólida para o design funcional.

O padrão Repository (4.3) é introduzido como uma estratégia arquitetônica essencial para simplificar a manipulação de dados, melhorando a eficiência e a manutenibilidade do código. Esse padrão oferece uma abordagem organizada para lidar com operações de leitura e gravação de dados.

A exploração das telas do aplicativo (4.5) traz uma compreensão tangível da interface do usuário, destacando a importância do design centrado no usuário e da experiência intuitiva para o sucesso do aplicativo.

A abordagem de segurança na conexão do banco de dados com a API (4.4) destaca a seriedade dada à proteção dos dados.

4.1 Processo de Desenvolvimento do Aplicativo

O processo de desenvolvimento do aplicativo “Poupe” começou com a criação do projeto usando o comando flutter create, estabelecendo as bases e configurando o ambiente necessário. A arquitetura inicial foi deliberadamente planejada com uma ênfase proeminente em widgets, proporcionando uma abordagem visual consistente e simplificando a manutenção e expansão do código ao longo do desenvolvimento. A implementação cuidadosa da Splash Page e Onboarding Page refletiu a atenção aos detalhes visuais, enquanto a prática contínua de refatoração e reuso de código assegurou eficiência, manutenibilidade e escalabilidade ao longo do processo de construção do aplicativo.

Durante a criação da Onboarding Page, a ênfase em um layout intuitivo e botões interativos destacou-se como fundamental para orientar os usuários nas funcionalidades do aplicativo. A prática consistente de refatoração e reuso de código foi incorporada, garantindo eficiência na implementação e preparando o terreno para a manutenção sustentável do projeto. A subsequente implementação da Sign Up Page, com o uso de TextFormField e TextField, evidenciou a atenção detalhada à personalização de campos e temas, otimizando a gestão visual e reforçando o controle e a segurança dos formulários, promovendo uma experiência de usuário aprimorada e confiável.

4.2 Diagrama de Caso de uso

O diagrama de caso de uso representa as interações entre os usuários e o sistema em nosso aplicativo de controle financeiro, desenvolvido para simplificar o gerenciamento de entradas e saídas financeiras. Abaixo estão os principais atores e casos de uso delineados no diagrama (figura 1):

Ator Principal:

Usuário autenticado:

Este ator representa usuários que realizaram o login no aplicativo. Eles têm acesso a funcionalidades avançadas, incluindo o registro de despesas e receitas, a visualização do resumo financeiro e a consulta do histórico de transações.

Principais Casos de Uso:

Realizar Login:

Permite que usuários não autenticados acessem suas contas, fornecendo as credenciais necessárias para autenticação.

Cadastrar Conta:

Possibilita que novos usuários criem contas no aplicativo, fornecendo as informações necessárias para registro.

Visualizar Resumo Financeiro:

Permite que usuários autenticados visualizem um resumo consolidado de suas entradas e saídas financeiras, oferecendo uma visão geral das finanças.

Registrar Despesa:

Permite que usuários autenticados registrem novas despesas, inserindo detalhes como valor, categoria e data.

Registrar Receita:

Oferece aos usuários autenticados a capacidade de registrar novas receitas, incluindo informações como valor, origem e data.

Visualizar Histórico de Transações:

Permite que usuários autenticados acessem o histórico completo de suas transações, oferecendo uma visão detalhada das atividades financeiras passadas.

Logout:

Possibilita que usuários autenticados encerrem a sessão, garantindo a segurança e privacidade de sua conta.

Este diagrama é uma representação visual intuitiva das principais interações entre os usuários e o aplicativo, proporcionando uma compreensão clara das funcionalidades oferecidas pelo sistema de controle financeiro.

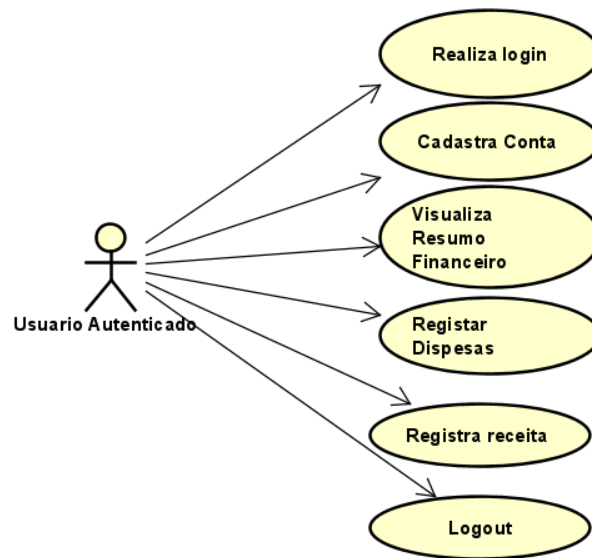


Figura 1. Diagrama de caso uso.

Fonte: Autoral, 2023.

Este fluxo de interação foi projetado com o objetivo de oferecer uma experiência intuitiva, garantindo que o usuário tenha conhecimento sobre o processo e possa explorar as funcionalidades do aplicativo de forma eficiente. Ao incorporar essas etapas em nosso design, estamos comprometidos em criar uma jornada de usuário fluida e envolvente.

4.3 Mapa Visual de Integração

O GraphQL Flutter e o GraphQL Service possibilitaram uma conectividade eficaz com a API. Ao remover consistentemente widgets e componentes, a modularidade e a reutilização de código foram fortalecidos, enquanto componentes como BalanceCard, ConstrainedBox e BoxConstraints foram adicionados para aprimorar a experiência do usuário e a estética. A implementação da funcionalidade Transactions demonstrou a eficácia na agregação de dados no aplicativo. A funcionalidade Wallet e a paginação foram implementadas para garantir flexibilidade e escalabilidade, e o uso inteligente das Dart Extensions facilitou processos rotineiros. A evolução constante do código foi assegurada pela aplicação de princípios de design, revisões contínuas e pequenos aprimoramentos. A robustez foi garantida mesmo em circunstâncias imprevistas por meio de um gerenciamento proativo de exceções. Um mapa mostrando como a conexão do aplicativo com o servidor está integrada é apresentado na Figura 2.

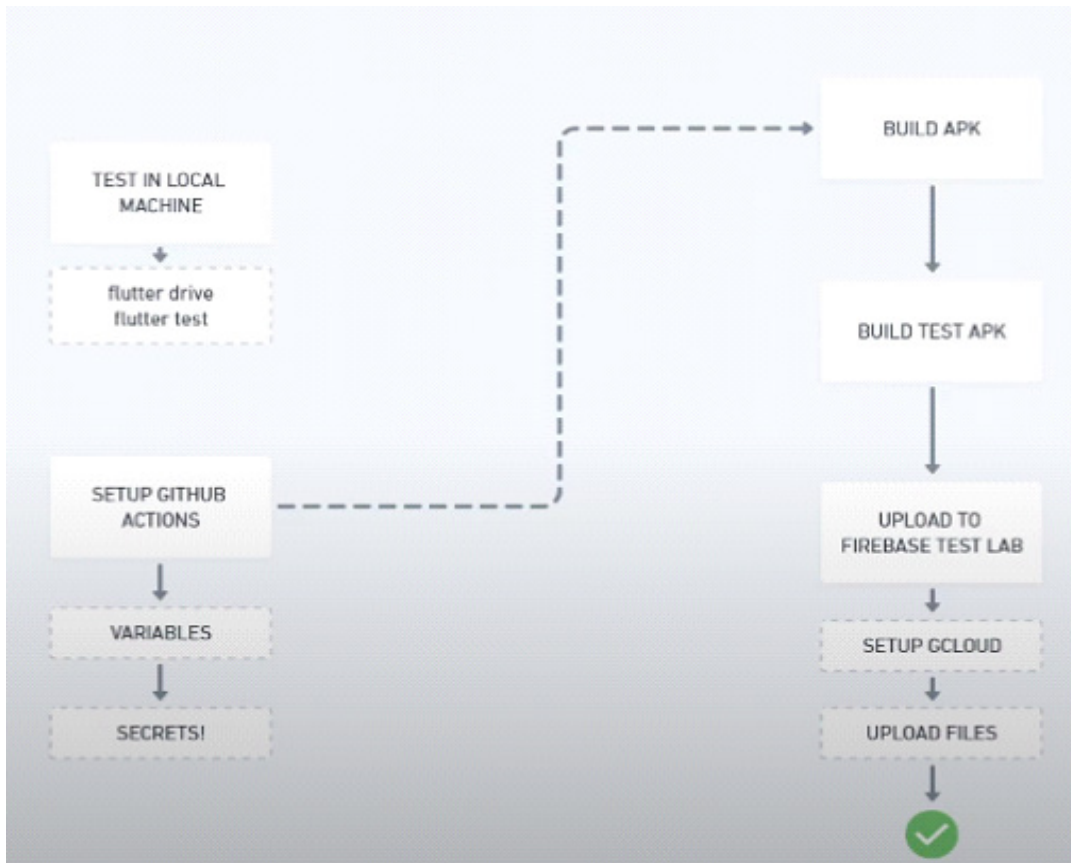


Figura 2. Mapa Visual de Integração.

Fonte: DevKaio (2023)

A implementação da abordagem Offline First utilizando Sqflite e Sync Service proporcionou uma experiência fluida mesmo em condições de conectividade intermitente. A aplicação foi avaliada em diversas situações com testes de integração realizados no Firebase Test Lab. A abordagem metodológica do Desenvolvimento Orientado a Testes (Test-Driven Development - TDD) foi reforçada pela utilização do monitoramento de cobertura de testes do Codecov, em conjunto com o Github Actions para integração contínua. O TDD foi seguido na construção do Controlador de Perfil, e a eficiência do código foi enfatizada na criação da Página de Perfil. Pull Requests foram utilizados para examinar as funcionalidades de perfil. A implementação da Funcionalidade de Estatísticas com Gráficos demonstrou a capacidade da aplicação para visualização de dados.

Essa abordagem estratégica permitiu que o aplicativo mantivesse uma experiência consistente, independentemente das condições de conexão. A avaliação abrangente, realizada por meio de testes de integração, validou a robustez e confiabilidade da aplicação em diversos cenários. O comprometimento com o TDD, evidenciado pelo uso conjunto de Codecov e Github Actions, reforça a qualidade do código e a prática consistente de testes. A construção cuidadosa do Controlador de Perfil e a ênfase na eficiência do código ao criar a Página de Perfil contribuíram para um desenvolvimento sólido. A utilização de Pull Requests para revisão de funcionalidades garantiu uma abordagem colaborativa e transparente. Além disso, a implementação da Funcionalidade de Estatísticas com Gráficos enriqueceu a aplicação, proporcionando uma representação visual intuitiva dos dados financeiros.

4.4 Programação de segurança da Tela de login

Em resumo a programação à cima mostra um *endpoint* irá responder ao um evento de criação de um novo usuário e assim se comunicando com o *firebase*, no qual ele vai capturar as informações do usuário e inseri-las no banco de dados, proporcionando uma maneira de manter os dados de usuários além de informações fornecidas pelo mesmo. Ou seja, toda a segurança é feita pelo *firebase* ao fazer o cadastro as informações vão ser passadas para o banco de dados, como observado na figura 3:

```
//função que será chamada pelo cliente flutter
exports.registerUser = functions.https.onCall(async (data) => {
  const email = data.email;
  const password = data.password;
  const displayName = data.displayName;

  if (email == null || password == null || displayName == null) {
    throw new functions.https.HttpsError('register-failed', 'missing information');
  }

  try {
    var userRecord = await admin.auth().createUser({
      email: email,
      password: password,
      displayName: displayName
    });

    const customClaims = {
      "https://hasura.io/jwt/claims": {
        "x-hasura-default-role": "user",
        "x-hasura-allowed-roles": ["user"],
        "x-hasura-user-id": userRecord.uid
      }
    };

    await admin.auth().setCustomUserClaims(userRecord.uid, customClaims);

    return userRecord.toJSON();
  } catch (error) {
    console.error('Error processing register:', error);
    throw new functions.https.HttpsError('internal', 'Error processing register.');
```

Figura 3: Código-Fonte: trecho de registros.

Fonte: Autoral, 2023.

O aplicativo financeiro em desenvolvimento é uma ferramenta projetada para auxiliar os usuários no controle de suas finanças pessoais, monitorando entradas e saídas de receitas. Utilizando um projeto da comunidade do Figma como base, o aplicativo promete oferecer uma interface intuitiva e funcionalidades práticas para o gerenciamento financeiro diário, a Figura 3 acima mostra a função de registros dos dados de usuário a partir de seu código-fonte.

4.5 Resultado da prototipação

Na figura 4, a tela inicial, os usuários são recebidos por uma *splash screen* seguida de um *onboarding* que orienta sobre as funcionalidades do aplicativo. A home page é projetada para fornecer uma visão geral das finanças do usuário, com todas as informações relevantes já cadastradas. A partir daí, pode-se acessar diferentes seções do aplicativo, como a página de estatísticas, que apresenta dados detalhados sobre os gastos, permitindo filtrar por dia, semana, mês ou ano e diferenciar entre receitas e despesas.



Figuras 4. Telas de uso do aplicativo.

Fonte: Autoral, 2023.

O aplicativo também inclui uma funcionalidade para adicionar despesas de forma rápida e simples (figura 5). Um gráfico de linhas (ou outro tipo apropriado) exibirá a evolução dos gastos ao longo do tempo, exigindo um certo nível de cálculo matemático que provavelmente será facilitado por recursos disponíveis na comunidade *Flutter*.

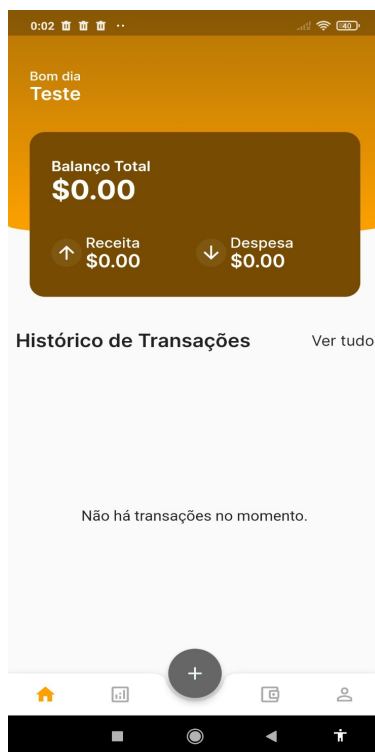


Figura 5: Tela principal do aplicativo.

Fonte: Autoral, 2023.

Além disso, haverá uma seção dedicada à carteira do usuário (figura 6), que mostrará um resumo do saldo geral, combinando receitas e despesas. Esta seção também permitirá realizar novas transações e visualizar itens pendentes, como contas a pagar, fornecendo uma visão clara da situação financeira atual do usuário.

O aplicativo está sendo projetado para incluir funcionalidades futuras, como a integração de pagamentos e a gestão de contas e cartões. No entanto, o foco inicial está em criar um Produto Mínimo Viável (MVP) que inclua as funcionalidades essenciais: criação de conta, login, home, registro de transações e perfil do usuário.

Detalhes adicionais de transações específicas, como a visualização de comprovantes e a exportação de dados para PDF, também estão previstos para serem incluídos, enriquecendo a experiência do usuário e fornecendo um controle financeiro mais robusto.

O desenvolvimento do aplicativo será acompanhado por uma série de publicações, guiando os usuários através do processo de criação e explicando cada etapa do desenvolvimento. O ambiente de desenvolvimento escolhido é o Flutter, uma escolha popular para a criação de aplicativos devido à sua flexibilidade e eficiência.



Figura 6. Tela de adição de receita e despesa do aplicativo.

Fonte: Autoral, 2023.

Sendo assim, o aplicativo promete ser uma solução abrangente para o gerenciamento de finanças pessoais, com uma interface amigável e funcionalidades que atendem às necessidades básicas e avançadas de controle financeiro.

5. DISCUSSÃO

Nos tempos atuais, a necessidade de gerenciar finanças pessoais de forma eficiente converge com o desenvolvimento contínuo de aplicativos financeiros. Essa convergência destaca a importância crescente da tecnologia para atender à demanda por soluções financeiras práticas e seguras. Os usuários buscam uma experiência otimizada ao administrar suas finanças diárias, considerando a praticidade como um requisito fundamental.

No entanto, ao examinar o panorama, identificamos desafios significativos enfrentados pelos usuários na gestão de suas finanças por meio de aplicativos convencionais. Um dos principais desafios reside na complexidade associada à categorização manual de despesas. Muitos usuários enfrentam dificuldades em classificar de maneira clara e eficiente seus gastos, resultando em uma compreensão inadequada de seus padrões financeiros. Outro obstáculo observado é a limitação na capacidade de prever comportamentos financeiros futuros e receber recomendações proativas.

A falta de uma abordagem preditiva de tendências de gastos compromete a habilidade dos usuários de planejar suas finanças de maneira eficaz, levando a decisões menos informadas e, por vezes, a situações financeiras desafiadoras.

Resolução Através do App POUPE surge como uma resposta inovadora a esses desafios. Para abordar a complexidade na categorização de despesas, o projeto incorpora um sistema inteligente de categorização automática. Essa funcionalidade simplifica significativamente o processo para os usuários, proporcionando uma experiência mais intuitiva e eficiente na gestão de suas despesas diárias. Além disso, para superar a limitação na previsão de comportamentos financeiros, o aplicativo integra recursos avançados de inteligência artificial e aprendizado de máquina. Essas tecnologias analisam padrões de gastos anteriores, oferecendo recomendações proativas aos usuários. Essa abordagem preditiva capacita os usuários a antecipar e planejar suas finanças de maneira mais informada, contribuindo para uma gestão financeira mais eficaz e adaptável às suas necessidades individuais.

6. CONCLUSÃO

A conclusão deste estudo recapitula os objetivos inicialmente propostos: desenvolver um aplicativo financeiro que facilite o gerenciamento de finanças pessoais, utilizando a tecnologia Flutter e baseando-se em designs da comunidade do Figma. Os objetivos foram alcançados com a criação de um protótipo funcional que oferece uma interface intuitiva e ferramentas práticas para o controle de entradas e saídas financeiras. A implementação bem-sucedida do protótipo demonstra a viabilidade do projeto e estabelece uma base sólida para futuras expansões e melhorias.

Os resultados obtidos indicam que o aplicativo possui potencial significativo para ajudar os usuários a gerenciar suas finanças de maneira eficiente. As funcionalidades implementadas, como o acompanhamento de despesas, visualização de estatísticas financeiras e gestão de transações, fornecem uma visão clara da saúde financeira do usuário. A capacidade de filtrar dados por período e tipo de transação permite uma análise detalhada dos hábitos de consumo e oportunidades de economia.

Do ponto de vista técnico, o uso do Flutter como ambiente de desenvolvimento provou ser uma escolha acertada, oferecendo flexibilidade e uma comunidade ativa para suporte. A integração com o Figma facilitou a prototipagem e o design, permitindo uma iteração rápida e eficaz. As implicações técnicas do estudo incluem a confirmação de que

ferramentas de código aberto e colaborativo podem resultar em produtos robustos e de alta qualidade.

Gerencialmente, o aplicativo oferece implicações significativas. Ele serve como uma ferramenta de empoderamento financeiro, proporcionando aos usuários maior controle e visibilidade sobre suas finanças. A gestão de finanças pessoais é simplificada, e a tomada de decisão é informada por dados precisos e atualizados. Isso pode levar a uma melhor gestão de recursos e a uma saúde financeira mais estável para os usuários.

As implicações futuras do estudo são amplas. A plataforma estabelecida pode ser expandida para incluir funcionalidades como integração de pagamentos, gestão de investimentos e conselhos personalizados baseados em inteligência artificial. O aplicativo tem o potencial de se tornar uma solução financeira abrangente, atendendo a uma gama mais ampla de necessidades financeiras e oferecendo serviços mais sofisticados.

A principal contribuição deste estudo para o campo de pesquisa é a demonstração de que aplicativos financeiros podem ser desenvolvidos de forma eficiente e eficaz usando ferramentas modernas de desenvolvimento de software. O projeto aborda a necessidade de ferramentas de gerenciamento financeiro pessoal que sejam acessíveis, fáceis de usar e tecnicamente avançadas. Ele responde à pergunta de pesquisa sobre como a tecnologia pode facilitar o gerenciamento financeiro pessoal, fornecendo um exemplo concreto de sucesso nessa área.

Sendo assim, o estudo alcançou seus objetivos de criar um aplicativo financeiro útil e intuitivo. As implicações técnicas e gerenciais destacam o impacto positivo que tais ferramentas podem ter na vida financeira dos usuários. As perspectivas futuras apontam para um caminho de inovação contínua e melhoria da saúde financeira através da tecnologia. A contribuição para o campo de pesquisa é clara, fornecendo insights valiosos sobre o desenvolvimento de aplicativos financeiros e estabelecendo um precedente para futuras investigações e desenvolvimentos.

REFERÊNCIAS

- Ayyagari, M., Demircuc-Kunt, A., & Maksimovic, V. (2011). Firm Innovation in Emerging Markets: The Role of Finance, Governance, and Competition. *Journal of Financial and Quantitative Analysis*, 46, 1545-1580. <https://doi.org/10.1017/S0022109011000378>
- Birch, K., Chiappetta, M., & Artyushina, A. (2020). The problem of innovation in technoscientific capitalism: data rentiership and the policy implications of turning personal digital data into a private asset. *Policy Studies*, 41, 468-487. <https://doi.org/10.1080/01442872.2020.1748264>
- Bolso Virtual: Aplicação Web Móvel para controle de finanças pessoais ALVES, Max Braynner Menezes (2017) (UFS) https://ri.ufs.br/bitstream/riufs/11104/2/Max_Braynner_Menezes_Alves.pdf
- Eric Seidel, diretor sênior de engenharia do Google(2018) .<https://www.dio.me/articles/descubra-flutter-o-poderoso-framework-da-google>
- Giudici, G., & Paleari, S. (2000). The Provision of Finance to Innovation: A Survey Conducted among Italian Technology-based Small Firms. *Small Business Economics*, 14, 37-53. <https://doi.org/10.1023/A:1008187416389>
- Gretalita, N., Suzianti, A., & Ardi, R. (2017). Visual Usability Design of Financial Personal Assistant Application. <https://doi.org/10.1145/3176653.3176700>
- King, R. (1993). Finance and Growth: Schumpeter Might Be Right. *Quarterly Journal of Economics*, 108, 717-737. <https://doi.org/10.2307/2118406>
- Kishkovich, Y. (2021). Conceptual Model of Enterprise Financial Management System with the

Application of Modern Information Technologies. The World of New Economy. <https://doi.org/10.26794/2220-6469-2020-14-4-47-55>

Makalew, B. (2022). Android Based Personal Finance Management Application: Design and Development. Engineering, Mathematics and Computer Science (EMACS) Journal, 4(1). <https://doi.org/10.21512/emacsjournal.v4i1.8085>

Oliveira, Kaio C. de. Financy_app. Aplicativo Financeiro, [s. l.], 2023. Disponível em: https://github.com/devkaio/financy_app e https://youtube.com/playlist?list=PLtlg0Apoubs_xVS8QxIX51zl0iucEcKyK&si=4sAzUQ7DzBRKel4k. Acesso em: 8 ago. 2023.

De Paula, Carolina Cristiane et al. Controle financeiro e orientação de investimentos. LIBERTAS: Revista de Ciências Sociais Aplicadas, v. 11, n. 1, p. 1-18, 2021.

Dos Santos, João Renan Almeida; VERONESE, Arthur Longo; LONGO, Luci. Finanças Pessoais e Desafios Atuais: Uma Pesquisa com Estudantes Universitários para Estruturação de um App Financeiro e Pedagógico. Administração: Ensino e Pesquisa, v. 23, n. 3, 2022.

Xie, Y. (2016). The Design and Implementation of Personal Finance Management System Based on Android, 633-636. <https://doi.org/10.2991/ICCSAE-15.2016.118>.

BATSCHINSKI, G. Backend as a Service: Prós e Contras. 2016. Disponível em: <https://www.infoq.com/br/news/2016/07/backend-pros-e-contras/>





CONECTANDO ATLÉTICAS ACADÊMICAS COM FLUTTER

13

Connecting Academic Associations with Flutter

Carlos Magno Souza dos Santos

Edilson Carlos Silva Lima

Leonardo de Castro Mesquita

D.O.I.: 10.29327/5391075.1-13



Criando Soluções Tecnológicas com a Engenharia de Computação

Resumo

Este artigo descreve um aplicativo inovador que tem o potencial de transformar a interação entre as atléticas acadêmicas de diferentes instituições de ensino. Atualmente, a comunicação entre atléticas é muitas vezes limitada à rede pessoal de contatos de cada atlética. Foram feitas pesquisas com membros atuais e antigos de algumas atléticas para entender melhor o problema e assim propor uma solução viável. Este aplicativo visa romper essas barreiras de comunicação e criar uma rede unificada, promovendo a colaboração e o compartilhamento de experiências entre as atléticas.

Palavras-chave: Flutter, Firebase, Aplicativo Móvel.

Abstract

This article describes an innovative application with the potential to transform the interaction among academic athletic organizations from different educational institutions. Currently, communication among such organizations is often limited to personal networks within each athletic group. Research involving current and former members of various athletic organizations was conducted to better understand the problem and propose a viable solution. This application aims to break down these communication barriers and create a unified network, fostering collaboration and the sharing of experiences among these academic athletic groups.

Keywords: Flutter, Firebase, Mobile APP.



1. INTRODUÇÃO

No contexto da vida universitária, as atléticas acadêmicas desempenham um papel central, promovendo o espírito esportivo e a coesão entre estudantes. Essas organizações, geralmente formadas por estudantes de uma mesma instituição de ensino, são conhecidas por suas atividades esportivas e sociais, tornando-se uma parte fundamental da experiência universitária. No entanto, até recentemente, a interação entre atléticas estava limitada a conexões pessoais, muitas vezes baseadas no acaso e nas redes de amizade. Isso gerava obstáculos à comunicação, à organização de eventos conjuntos e à partilha de experiências entre diferentes instituições de ensino.

Nesse contexto, este artigo descreve um aplicativo inovador que visa revolucionar a forma como as atléticas acadêmicas interagem e se conectam. Atualmente, a comunicação entre essas atléticas é frequentemente restrita a contatos pessoais, o que torna desafiador o estabelecimento de parcerias e o compartilhamento de experiências entre estudantes de diferentes instituições. Essa limitação cria uma barreira à organização de eventos conjuntos, à troca de informações relevantes e à criação de uma rede unificada de estudantes universitários.

Neste artigo, exploraremos em detalhes como o aplicativo proposto facilita a conexão entre as atléticas acadêmicas, promovendo uma colaboração eficaz e uma comunicação fluente. Além disso, discutiremos os benefícios significativos que essa inovação traz não apenas para as atléticas, mas também para os estudantes universitários em geral. O objetivo fundamental deste artigo é destacar o imenso potencial transformador desse aplicativo e incentivar sua adoção por um número cada vez maior de atléticas acadêmicas. À medida que mais instituições abraçam essa plataforma, o futuro das interações universitárias se configura como mais inclusivo e repleto de emocionantes oportunidades.

2. TRABALHOS RELACIONADOS

Realizando um estudo sobre a problemática apresentada no resumo, foram investigados diversos trabalhos relacionados que abordam temas semelhantes. No item 2.1, foram explorados artigos que tratam de projetos voltados para o desenvolvimento de aplicativos de comunicação. No item 2.2, a pesquisa se concentrou em investigações que abordam a interação entre grupos acadêmicos, especificamente atléticas universitárias, e as dificuldades enfrentadas nesse contexto. No item 2.3, foram analisados trabalhos que destacam a importância da inovação tecnológica no aprimoramento das interações entre grupos sociais. Essa revisão de trabalhos relacionados é essencial para situar o presente estudo no contexto acadêmico e tecnológico, fornecendo uma base sólida para a análise e compreensão dos resultados apresentados a seguir, o item 2.4, fala sobre o diferencial desse projeto em relação aos outros citados.

2.1 Aplicação com framework flutter para o terceiro setor que realizam ações sociais

Este estudo explora o uso do framework Flutter para desenvolver uma aplicação móvel direcionada ao terceiro setor, visando aproximar projetos sociais e possíveis voluntários. O desafio é criar uma plataforma que permita a interação entre doadores e instituições do

terceiro setor de forma econômica e eficaz. A aplicação oferece a criação de campanhas de arrecadação e compartilhamento de informações sobre missões e objetivos das campanhas.

A fome e a precariedade de vida afetam uma parcela significativa da população mundial. As organizações do terceiro setor desempenham um papel fundamental na busca por soluções, dependendo, muitas vezes, de doações e colaborações. No entanto, a criação de aplicativos para auxiliar essas instituições pode ser dispendiosa. A necessidade de criar aplicativos nativos para diferentes plataformas e a complexidade do desenvolvimento podem tornar a iniciativa inacessível para muitas organizações.

O uso do framework Flutter mostrou-se eficaz na criação de uma aplicação que facilita a interação entre doadores e projetos sociais. A plataforma permite a criação de campanhas de arrecadação, compartilhamento de informações e aproximação de pessoas e organizações que compartilham objetivos comuns. No entanto, desafios futuros incluem melhorias na interface gráfica, integração com bases de dados, relatórios e validação da usabilidade. A aplicação representa uma tentativa de contribuir para uma sociedade mais igualitária e solidária, aproveitando as possibilidades oferecidas pela tecnologia.

2.2 Gerenciamento de Projetos na Realização de Jogos Universitários

O projeto de Joseane Borges aborda a necessidade de melhorar a organização dos Jogos Universitários (jogos entre atléticas), com foco no evento anual JOIA Fronteira. O principal problema identificado foi a falta de um sistema de gerenciamento eficaz, levando a desafios na organização e execução do evento. Para resolver essa questão, o projeto propôs a aplicação de metodologias de gerenciamento de projetos e ferramentas.

Essas abordagens ajudaram a simplificar o planejamento e execução dos jogos, melhorando a comunicação, definindo objetivos claros e otimizando os custos de arbitragem e premiação. No entanto, o projeto observou que a falta de uma literatura específica para jogos universitários levou as comissões organizadoras a adotarem diferentes metodologias e ferramentas de acordo com a complexidade de cada edição.

As vantagens desse projeto incluem uma organização mais eficiente dos eventos esportivos universitários, melhorias na comunicação e a capacidade de adaptar metodologias de gerenciamento de projetos de acordo com as necessidades de cada edição. No entanto, as desvantagens podem surgir se as equipes de organização não possuírem o conhecimento e a maturidade necessários para aplicar essas abordagens de forma eficaz.

2.3 Desenvolvimento de aplicativo de rede social com foco em jogos: Games Connect

Nasceu da necessidade de fornecer à comunidade de jogadores um espaço de interação e compartilhamento de experiências. Baseado em metodologias de gerenciamento de projetos, o aplicativo foi criado com o objetivo de unir jogadores de todo o mundo em uma plataforma exclusiva para gamers. Ele oferece interatividade, sistemas de pontuação e estatísticas detalhadas para melhorar a experiência dos usuários.

No entanto, desafios podem surgir no processo de atração de usuários e anunciantes para a plataforma. É importante ressaltar que a qualidade do serviço, com recursos como chat, reações às postagens e um sistema de pontuação, é uma vantagem significativa.



Para garantir o sucesso contínuo do projeto, será necessário enfrentar questões relacionadas à expansão da base de usuários e ao envolvimento de anunciantes, o que requer estratégias de marketing e desenvolvimento contínuo.

Em resumo, o “Games Connect” é uma plataforma que visa atender às necessidades da comunidade gamer, oferecendo interação e recursos atraentes. No entanto, para alcançar seu pleno potencial, é fundamental superar desafios relacionados à atração de usuários e anunciantes, garantindo assim o crescimento contínuo e o sucesso do aplicativo.

2.4 Diferencial dos trabalhos relacionados com o projeto desenvolvido

No contexto acadêmico, a gestão de atléticas representa um elemento fundamental da vida universitária, promovendo a interação entre estudantes e a realização de eventos esportivos. No entanto, até recentemente, as atléticas enfrentavam desafios significativos em termos de comunicação, coordenação de eventos e compartilhamento de informações. Este artigo explora um projeto inovador que se concentra nas necessidades específicas das atléticas acadêmicas, oferecendo uma solução personalizada para lidar com esses desafios.

O projeto em questão visa a criação de um aplicativo móvel dedicado a conectar as atléticas acadêmicas, fornecendo uma plataforma especializada para os estudantes universitários. O diferencial deste projeto é a sua capacidade de atender às necessidades exclusivas desses grupos, oferecendo funcionalidades projetadas para melhorar a comunicação, a organização e o gerenciamento de eventos esportivos. Através da aplicação, as atléticas podem agendar jogos, compartilhar informações, coletar opiniões e aprimorar sua coordenação interna.

Este projeto se destaca por seu foco específico nas atléticas acadêmicas, o que o diferencia dos aplicativos de gerenciamento de projetos em geral. Ele promove a conectividade entre os estudantes, fortalece a comunidade universitária e oferece potencial para melhorar a experiência universitária, proporcionando uma plataforma especializada para a coordenação de eventos esportivos e atividades sociais. Além disso, ele se concentra nas necessidades únicas desses grupos, abordando diretamente os desafios de comunicação e organização que as atléticas acadêmicas enfrentam, contribuindo assim para uma experiência mais enriquecedora no âmbito acadêmico.

3. REVISÃO BIBLIOGRÁFICA

Nesta seção, será realizada uma revisão bibliográfica abrangente abordando aspectos relevantes para o entendimento do projeto. A revisão é dividida em diferentes subseções para melhor compreensão.

3.1 Scrum

Nesse projeto, a metodologia ágil escolhida para orientar o desenvolvimento da aplicação foi o Scrum, uma estrutura popular conhecida por sua ênfase em entregas incrementais, adaptação contínua e colaboração interdisciplinar. Adaptado para ser aplicado por um único desenvolvedor, o Scrum manteve sua estrutura essencial, proporcionando uma abordagem sistemática e eficaz, mesmo em um contexto de trabalho individual.

O Scrum baseia-se em princípios fundamentais, onde o Product Owner representa as necessidades dos *stakeholders*, garantindo que o produto final atenda às expectativas. O Scrum Master facilita o processo, removendo obstáculos que possam surgir durante o desenvolvimento. Por sua vez, o Time de Desenvolvimento é responsável por entregar incrementos do produto, priorizando as funcionalidades essenciais definidas no Product Backlog.

O ciclo de desenvolvimento no Scrum é caracterizado por diversas reuniões, como o *Sprint Planning*, que estabelece metas específicas para cada iteração, e o *Daily Scrum*, uma reunião diária para atualizações, identificação de obstáculos e planejamento futuro. Ao final de cada *sprint*, o *Sprint Review* avalia o progresso, ajusta o *backlog* conforme necessário, e a *Sprint Retrospective* oferece uma oportunidade para reflexão sobre o processo, identificando melhorias contínuas. Essa metodologia provou ser eficaz ao promover a transparência, colaboração e entrega contínua de valor, mesmo em um ambiente de desenvolvimento individual.

3.2 UML e Astah UML

A *Unified Modeling Language* (UML) é uma linguagem de modelagem padronizada amplamente empregada na engenharia de software para visualizar, especificar, construir e documentar sistemas complexos. Ela oferece um conjunto de notações gráficas que permitem aos desenvolvedores criar representações visuais abstratas e compreensíveis de sistemas, facilitando a comunicação entre os membros da equipe e partes interessadas.

A UML fornece uma linguagem visual unificada para descrever aspectos diversos de um sistema, incluindo suas estruturas, comportamentos, interações e funcionalidades. Ela engloba vários tipos de diagramas, como diagramas de classes, diagramas de caso de uso, diagramas de sequência e muitos outros, cada um enfatizando aspectos específicos do sistema. Essa padronização na representação visual contribui para uma compreensão mais clara e consistente dos sistemas, promovendo a eficiência no desenvolvimento de software e a comunicação efetiva entre os envolvidos no processo. O Astah, como uma ferramenta de modelagem UML, desempenha um papel crucial na aplicação desses conceitos, permitindo a criação de diagramas claros e estruturados que auxiliam no desenvolvimento eficaz do sistema.

O diagrama de caso de uso é uma representação gráfica fundamental na modelagem de sistemas, destacando as interações entre atores externos e o sistema em questão. Ele fornece uma visão de alto nível das funcionalidades oferecidas pelo sistema, identificando os atores envolvidos e os cenários de uso. No contexto do desenvolvimento de software, o diagrama de caso de uso desempenha um papel crucial ao mapear os requisitos do usuário para as funcionalidades do sistema, facilitando a compreensão das relações entre diferentes partes do sistema.

O diagrama de classes é uma representação visual crucial na modelagem de sistemas orientados a objetos, oferecendo uma visão estruturada das classes presentes no sistema e suas relações. No contexto do desenvolvimento de software, esse diagrama ajuda a identificar as principais entidades e seus atributos, bem como a compreender as interações entre elas.



3.3 Aplicativos Móveis e sua importância

Os aplicativos móveis se tornaram uma parte integral da vida cotidiana, abrangendo diversos aspectos, desde entretenimento até produtividade. A facilidade de acesso a informações e serviços através de dispositivos móveis tornou os aplicativos um canal fundamental de comunicação. No contexto desse projeto, desenvolver um aplicativo é uma abordagem sensata, dada a ubiquidade dos dispositivos móveis entre estudantes universitários e a necessidade de melhorar a comunicação entre atléticas acadêmicas.

3.4 Framework Flutter

O Flutter, desenvolvido pela Google, é um framework de código aberto para o desenvolvimento de aplicativos móveis multiplataforma. Ele se destaca por sua capacidade de criar aplicativos para Android e iOS com um único conjunto de código-fonte, o que o torna econômico e eficiente.

3.5 Dart

A linguagem de programação Dart é a base do Flutter e oferece vantagens como o Ahead-Of-Time (AOT) de compilação, melhorando o desempenho do aplicativo, a linguagem Dart, desenvolvida pela Google, é projetada para ser uma linguagem eficiente e intuitiva, especialmente otimizada para o desenvolvimento de aplicativos móveis. Com uma sintaxe clara e concisa, o Dart facilita o processo de codificação, permitindo aos desenvolvedores expressar ideias de forma eficaz. Além disso, sua capacidade de compilação AOT contribui para a eficiência do Flutter, garantindo uma execução rápida e uma experiência de usuário mais fluida nos aplicativos desenvolvidos com essa tecnologia (Google, 2023). Essa combinação de Flutter e Dart oferece uma abordagem robusta para o desenvolvimento ágil de aplicativos móveis, proporcionando benefícios tanto em termos de desempenho quanto de produtividade.

3.6 Firebase

O Firebase, da Google, é uma plataforma de desenvolvimento de aplicativos que oferece recursos para melhorar a autenticação, armazenamento de dados em tempo real, hospedagem, análise e notificações de aplicativos. Sua integração é benéfica em aplicativos móveis, pois simplifica a criação de aplicativos que necessitam de recursos em tempo real, como chats e atualizações de conteúdo em tempo real. Nesse projeto, a integração do Firebase é crucial para fornecer comunicação em tempo real e melhorar a colaboração entre as atléticas acadêmicas.

4. PROJETO

Neste projeto, foi desenvolvido um aplicativo chamado 'theta', com o objetivo de resolver o problema de comunicação entre as atléticas acadêmicas. Este processo envolveu a adaptação da metodologia Scrum para um único indivíduo, demonstrando sua eficácia mesmo em uma configuração de equipe reduzida. Inicialmente, foi conduzido um *Sprint Planning*, estabelecendo metas específicas para cada iteração, definindo tarefas e priori-

dades. Dado o curto prazo disponível, essa metodologia se revelou ideal. Frequentemente, realizaram-se *Daily Scrums*, revisões diárias de progresso, para identificar obstáculos e planejar o trabalho futuro. Incrementos funcionais foram desenvolvidos em ciclos curtos, priorizando funcionalidades essenciais. Além disso, foram realizados *Sprint Reviews* regulares para avaliar o progresso e ajustar o plano conforme necessário. *Sprint Retrospectives* periódicas foram conduzidas, proporcionando uma reflexão sobre o que funcionou bem e identificando áreas passíveis de aprimoramento na abordagem de desenvolvimento. Seguem abaixo as figuras explicativas sobre o aplicativo ‘theta’, com diagramas UML e telas essenciais do aplicativo.

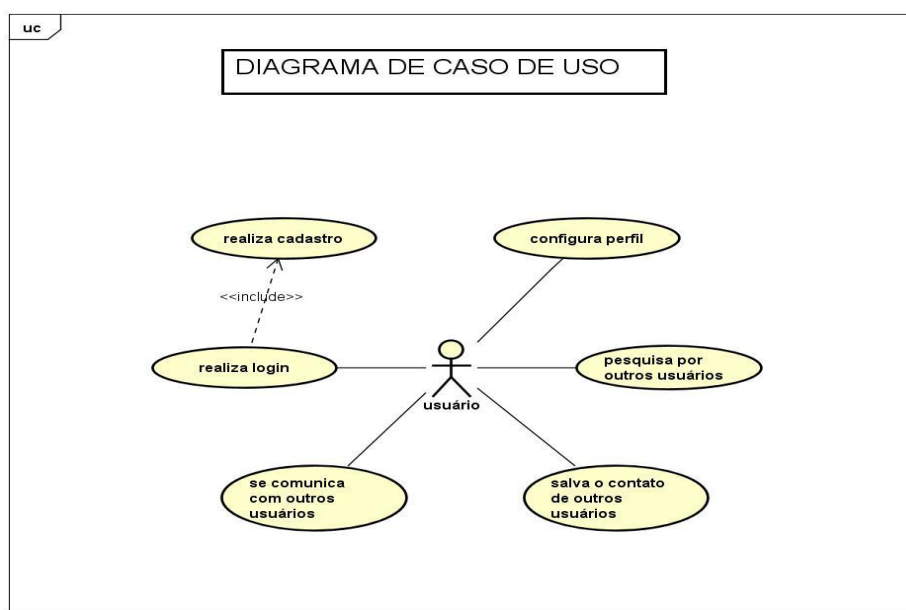


Figura 1. Diagrama de caso de uso

Fonte: autoral, 2023.

No diagrama de caso de uso proposto, o ator principal é o “usuário”, representando o usuário final da aplicação. O caso de uso “realizar login” indica a capacidade do usuário de acessar o sistema, enquanto o caso de uso “realizar cadastro” está incluído no “realizar login”, indicando que o cadastro é uma etapa essencial durante o processo de login.

Além disso, o ator “usuário” está associado a uma série de outros casos de uso, demonstrando as funcionalidades disponíveis no sistema após o cadastro e login.

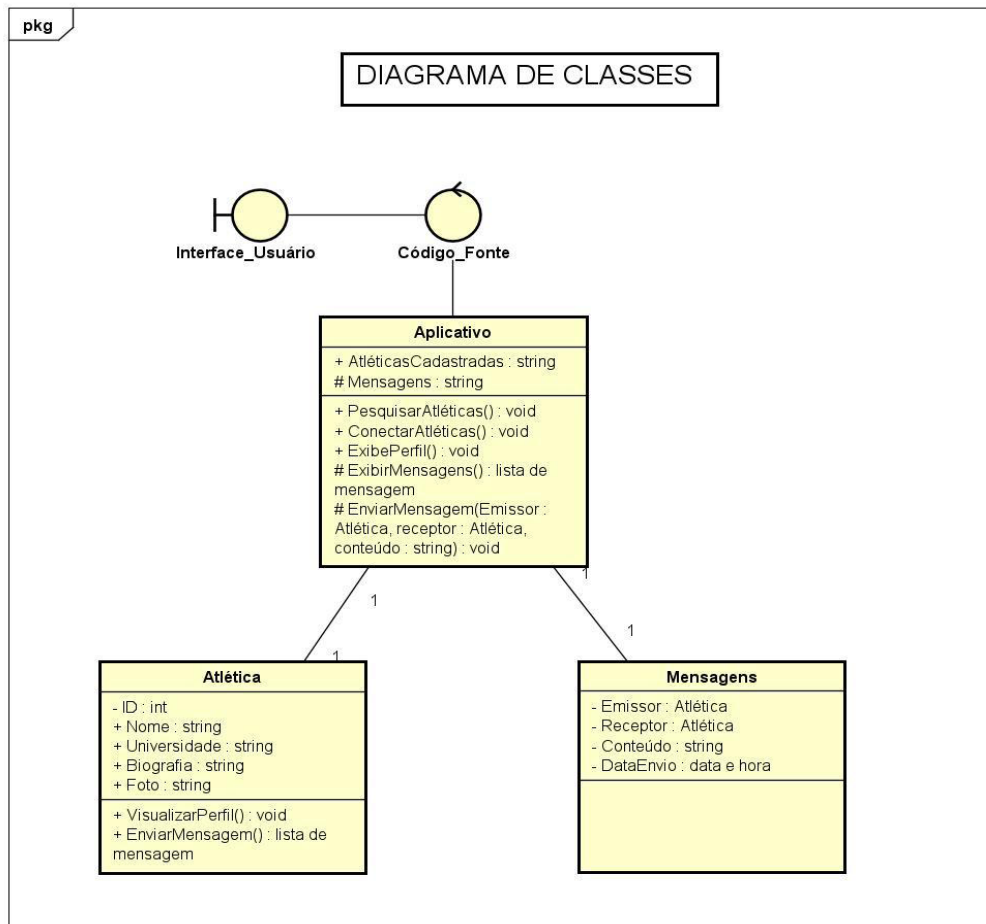


Figura 2. Diagrama de classes

Fonte: autoral, 2023.

O diagrama de classes proposto para o projeto apresenta três classes principais: “Aplicativo”, “Atléticas” e “Mensagem”. Cada uma dessas classes desempenha um papel crucial no funcionamento da aplicação.

A classe “Aplicativo” representa a camada de interface do usuário, indicando a interação direta do usuário com a aplicação. Essa classe serve como a ponte entre a lógica do código fonte e a experiência do usuário, facilitando a navegação e a utilização eficiente da aplicação.

A classe “Atlética” representa as informações relacionadas às atléticas acadêmicas, refletindo a estrutura de dados necessária para armazenar e gerenciar dados sobre diferentes atléticas. Essa classe é essencial para a funcionalidade principal da aplicação, conectando as atléticas de diferentes instituições de ensino.

A classe “Mensagens” refere-se à troca de mensagens entre os usuários da aplicação. Ela representa a capacidade de comunicação eficaz entre as atléticas, promovendo a colaboração e o compartilhamento de informações.

A presença dessas três classes no diagrama reflete a estrutura essencial do sistema, mostrando como a interface do usuário, as informações sobre as atléticas e a comunicação por meio de mensagens estão interconectadas para fornecer uma experiência completa e funcional aos usuários da aplicação.

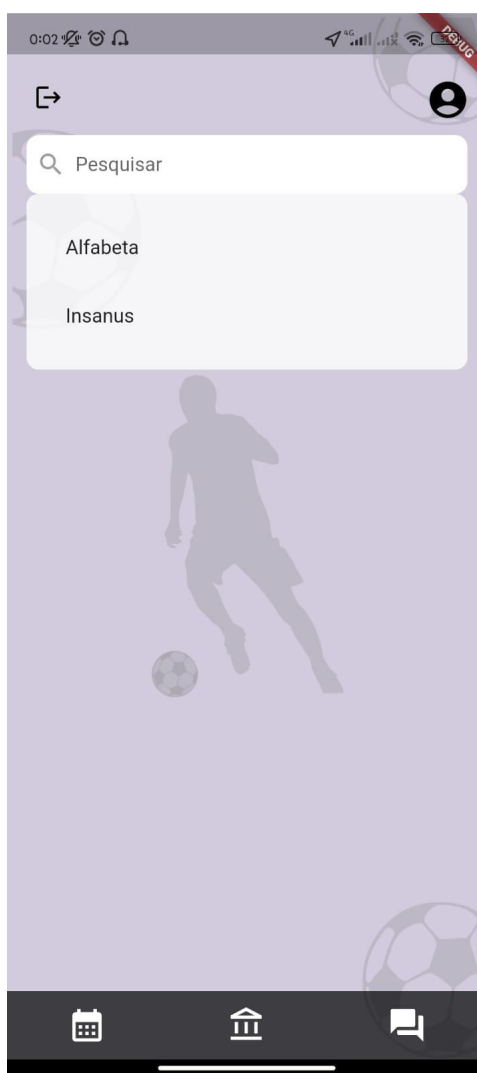


Figura 3. Campo de pesquisa

Fonte: Autoral, 2023.

A figura 3 (campo de pesquisa) é uma ferramenta fundamental na tela de início do aplicativo. Sua principal função é permitir que os usuários pesquisem e localizem atléticas acadêmicas de interesse com facilidade e rapidez. Ao tocar no campo de pesquisa, uma lista com todas as atléticas cadastradas é exibida, proporcionando aos usuários uma visão abrangente das opções disponíveis.

Além de fornecer uma lista completa de atléticas, o campo de pesquisa permite que os usuários insiram o nome da atlética que desejam encontrar. Ao começar a digitar, o aplicativo gera resultados em tempo real, exibindo as atléticas que correspondem à pesquisa, tornando o processo de busca ainda mais eficiente, e ao clicar no nome da atlética, o usuário é levado a tela de perfil da mesma (figura 4).

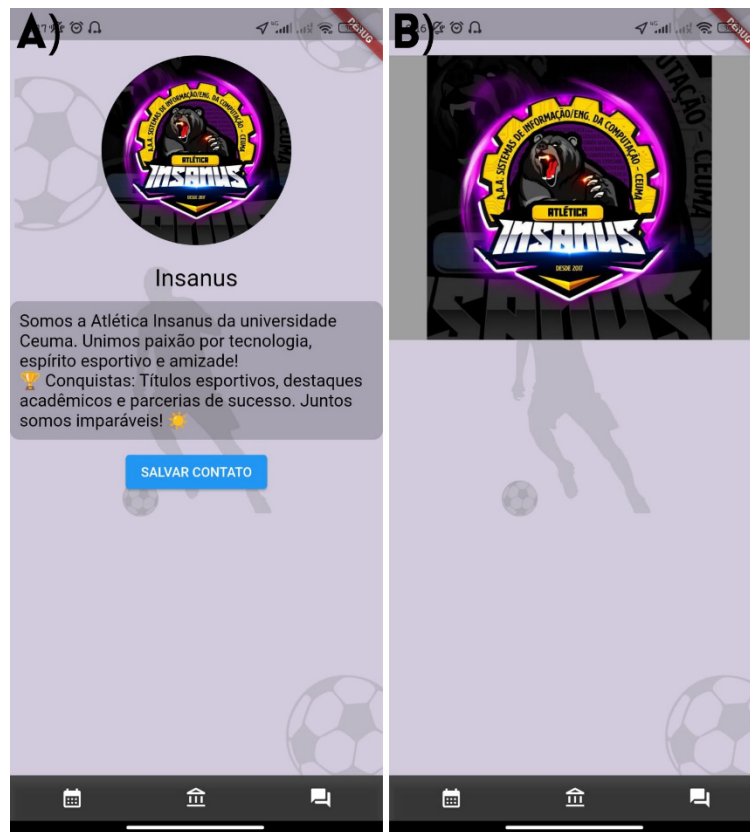


Figura 4. Visualização de perfil

Fonte: Autoral, 2023.

A figura 4 (Visualização de perfil) é um componente essencial do aplicativo, permitindo que os usuários acessem informações detalhadas sobre uma atlética acadêmica específica e estabeleçam conexões com ela. Esta tela oferece uma experiência envolvente e informativa, incentivando os usuários a explorar, interagir e se conectar com as atléticas acadêmicas de seu interesse. Abaixo estão os principais elementos desta tela:

A tela de perfil exibe uma foto de perfil da atlética em formato arredondado. Ao tocar na foto, ela se expande para mostrar a imagem em um tamanho maior, permitindo aos usuários visualizar com mais detalhes e clareza a imagem da atlética.

Abaixo da foto de perfil, os usuários encontram informações relevantes sobre a atlética. Isso pode incluir o nome da atlética, uma breve “biografia” ou descrição que destaca suas atividades, objetivos e realizações. Essas informações proporcionam aos usuários uma compreensão imediata do propósito e da missão da atlética.

No perfil da atlética, os usuários têm a opção de salvar o contato dessa atlética em sua lista de contatos pessoais. Esse botão permite que os usuários estabeleçam um vínculo direto com a atlética, facilitando futuras interações e comunicações, ao clicar, o usuário é levado a tela de contatos (figura 5, item A).

A tela de perfil de atlética é projetada para dar aos usuários uma visão abrangente de uma atlética acadêmica específica, permitindo-lhes entender sua identidade, atividades e propósito. Ao permitir a expansão da foto de perfil e o salvamento de contatos, essa tela promove a interação e a conexão significativa entre os usuários e as atléticas acadêmicas, contribuindo para o fortalecimento da comunidade do aplicativo.

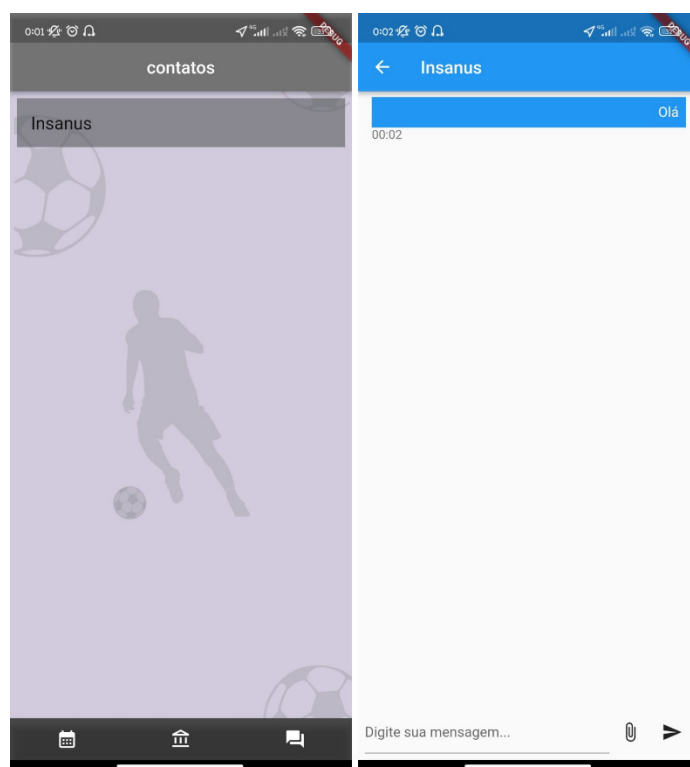


Figura 5. Tela de contatos (A) e tela de Chat(B)

Fonte: Autoral, 2023.

A figura 5 (tela de contatos) é uma parte fundamental do aplicativo, fornecendo aos usuários uma maneira rápida e eficiente de acessar suas conexões com as atléticas acadêmicas. Esta tela é um ponto central para gerenciar e interagir com as atléticas que o usuário escolheu se conectar.

A tela de chat é onde ocorre a comunicação direta entre o usuário e uma atlética acadêmica específica. Esta tela facilita a troca de mensagens e o compartilhamento de informações.

5. RESULTADOS E DISCUSSÃO

Nesta seção, apresentaremos os resultados do desenvolvimento do aplicativo que conecta atléticas acadêmicas e sua relação com a metodologia científica adotada durante o projeto.

5.1 Pesquisa

Foi conduzida uma pesquisa junto aos presidentes das atléticas e membros da diretoria, e a seguir estão representadas em figuras as perguntas e respostas mais relevantes, oferecendo uma visão abrangente sobre as percepções e práticas desses líderes no contexto da comunicação entre atléticas acadêmicas.

Como você classificaria a eficácia da comunicação atual entre atléticas acadêmicas?
10 respostas

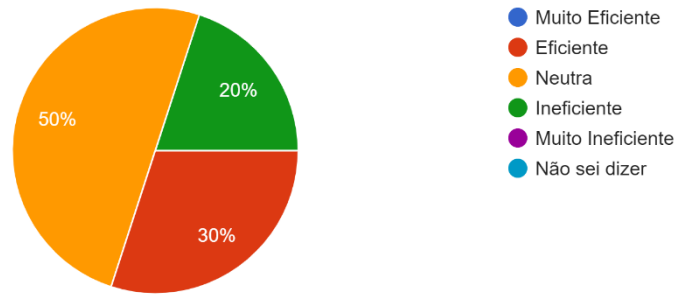


Gráfico 1. Pesquisa

Fonte: Autoral, 2023.

O gráfico 1, apresenta a distribuição percentual das respostas ao questionário sobre a eficiência da comunicação entre as atléticas acadêmicas. Observa-se que 50% dos participantes têm uma visão neutra, 30% consideram a comunicação eficiente, enquanto 20% a percebem como ineficiente.

Com que frequência você utiliza aplicativos ou plataformas online para se comunicar com membros de outras atléticas acadêmicas?
10 respostas

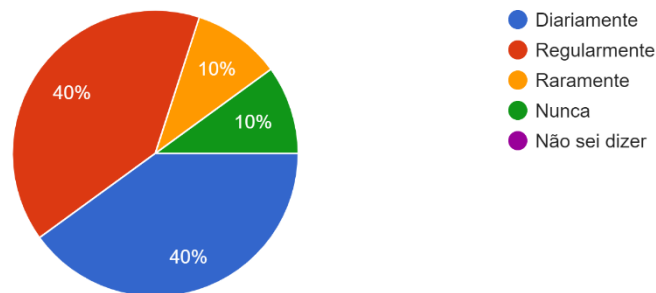


Gráfico 2. Pesquisa

Fonte: Autoral, 2023.

O gráfico 2, ilustra a frequência com que os participantes utilizam aplicativos ou plataformas online para se comunicar com membros de outras atléticas acadêmicas. Nota-se que 40% relataram utilizar diariamente, outros 40% o fazem regularmente, enquanto 10% optam por raramente e 10% indicam nunca utilizar tais meios de comunicação.

Quais dos seguintes métodos de comunicação sua atlética mais utiliza atualmente? (Marque todas as que se aplicam)

10 respostas

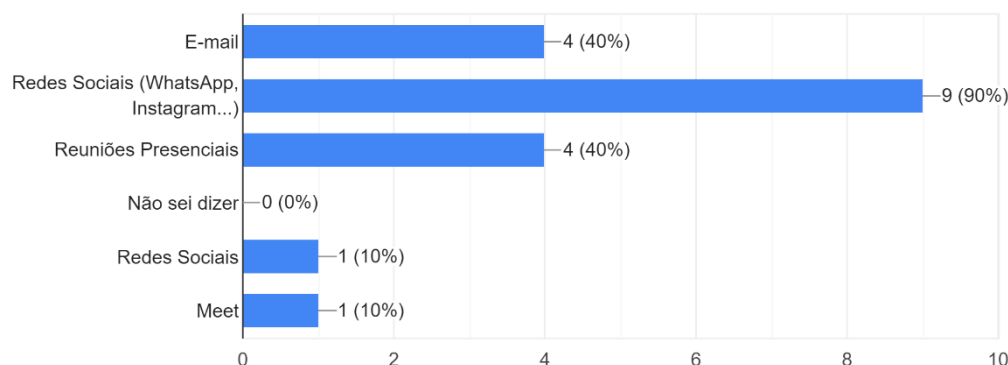


Gráfico 3. Pesquisa

Fonte: Autoral, 2023.

O gráfico 3, reflete os métodos de comunicação preferenciais das atléticas acadêmicas, conforme indicado pelos participantes. Os resultados revelam que 40% das atléticas ainda utilizam o e-mail como método de comunicação, enquanto expressivos 90% optam por plataformas de redes sociais, como WhatsApp e Instagram. Além disso, 40% mencionaram a realização de reuniões presenciais, 10% destacaram exclusivamente o uso de redes sociais, e 10% indicaram preferência por plataformas de videoconferência, como o Meet.

A pesquisa realizada revela insights significativos sobre a interação entre as atléticas acadêmicas e as preferências dos usuários em relação às ferramentas de comunicação. Os resultados indicam uma necessidade clara de soluções inovadoras que promovam a colaboração e a troca eficiente de informações entre as atléticas. A disposição universal dos participantes em adotar um aplicativo centralizado para contatos das atléticas destaca a importância de uma plataforma dedicada, capaz de superar as limitações das formas tradicionais de comunicação. Esses achados não apenas validam a pertinência do projeto em desenvolvimento, mas também ressaltam a oportunidade de contribuir de maneira significativa para aprimorar as interações entre as atléticas acadêmicas em ambientes universitários.

5.2 Resultados

O aplicativo desenvolvido se mostrou eficaz na resolução do problema de comunicação limitada entre as atléticas acadêmicas. Ele oferece uma solução abrangente que une essas instituições em uma plataforma única, permitindo uma comunicação direta e eficaz. Isso promove uma colaboração mais sólida entre as atléticas, fortalecendo suas atividades e relacionamentos. Os resultados indicam que o aplicativo tem potencial para melhorar a qualidade das interações e promover uma comunidade mais coesa.

As vantagens do aplicativo incluem a melhoria da comunicação, a promoção da colaboração e a possibilidade de eventos mais bem-sucedidos. No entanto, também existem desafios, como a necessidade de aprendizado para utilizar o aplicativo e a dependência de conectividade com a internet. É importante que as atléticas considerem esses aspectos ao adotar a plataforma.

A metodologia adotada para o desenvolvimento do aplicativo envolveu várias etapas. Primeiramente, foi realizada uma revisão da literatura para compreender os desafios enfrentados pelas atléticas acadêmicas e as necessidades de comunicação. Em seguida, foram conduzidas pesquisas com membros de atléticas para coletar dados sobre as dificuldades enfrentadas.

Com base nos dados coletados, o desenvolvimento do aplicativo ocorreu utilizando o framework Flutter e o Firebase para construir a infraestrutura necessária. O uso dessas tecnologias permitiu criar uma aplicação multiplataforma com recursos de alta escalabilidade.

A metodologia científica empregada no projeto incluiu a definição clara do problema, a coleta de dados, a escolha das ferramentas tecnológicas adequadas, o desenvolvimento, os testes e a validação. A pesquisa e a coleta de dados foram fundamentais para orientar o desenvolvimento do aplicativo e garantir que ele atendesse às necessidades identificadas.

Em resumo, a metodologia científica adotada permitiu o desenvolvimento de um aplicativo que aborda eficazmente o problema da comunicação limitada entre as atléticas acadêmicas. Os resultados obtidos indicam o potencial de melhorar a qualidade das interações e promover uma colaboração mais sólida entre essas instituições. A combinação de uma metodologia científica sólida com as tecnologias apropriadas resultou em uma solução inovadora para o contexto das atléticas acadêmicas.

5.3 Funcionalidades Adicionais do Chat

Além das funcionalidades anteriormente discutidas, o aplicativo também apresenta uma característica adicional no chat que permite aos usuários criar e agendar eventos esportivos ou sociais diretamente dentro da conversa. Ao clicar em um ícone específico, os usuários podem preencher um formulário com informações como modalidade esportiva, data, hora e local para propor um evento.

Essa funcionalidade promove a organização de jogos e eventos de maneira mais prática e simplificada. Os outros membros do chat podem aceitar ou recusar o convite para o evento. Quando um evento é aceito, ele é automaticamente adicionado a uma tela de calendário no aplicativo, proporcionando uma visão geral de todos os eventos agendados pelas atléticas.

Um aspecto importante a ser observado é que, devido a restrições de tempo durante o desenvolvimento do projeto, a funcionalidade adicional do chat que permite aos usuários criar e agendar eventos esportivos ou sociais não pôde ser completamente finalizada. Embora tenha sido concebida como parte do escopo do aplicativo, a implementação completa desse recurso exigiria mais tempo e recursos.

Essa funcionalidade não finalizada representa uma oportunidade para futuras melhorias no aplicativo. Quando completamente implementada, ela poderá oferecer uma maneira prática e conveniente para as atléticas e seus membros organizarem eventos esportivos e sociais de maneira mais eficaz.

5.4 Futuras Atualizações e Expansões

O aplicativo desenvolvido demonstra um grande potencial para futuras atualizações e expansões. Uma das possibilidades é a inclusão de funcionalidades que envolvam não apenas as atléticas, mas também os membros individuais. Isso pode incluir recursos que permitem que os membros participem diretamente de eventos, inscrevam-se em compe-

tições esportivas ou contribuam com ideias e feedback para as atléticas.

Além disso, considerando a rápida evolução das tecnologias móveis e a necessidade constante de aprimorar a experiência do usuário, as futuras atualizações podem focar na otimização da interface do aplicativo, melhorias de desempenho e a incorporação de recursos de realidade virtual ou aumentada para uma experiência mais envolvente.

Também é possível que o aplicativo venha a incluir integrações com redes sociais ou plataformas de streaming, permitindo que as atléticas alcancem um público mais amplo e compartilhem suas atividades de maneira mais eficaz. Com o constante desenvolvimento e aprimoramento, o aplicativo pode se tornar uma ferramenta central para a interação e colaboração entre atléticas acadêmicas e seus membros.

Dessa forma, o aplicativo tem o potencial de evoluir e crescer, adaptando-se às necessidades em constante mudança das atléticas e dos usuários, tornando-se uma parte fundamental da comunidade atlética acadêmica.

À medida que o aplicativo continua a evoluir e receber atualizações, essas funcionalidades adicionais poderão ser concluídas e refinadas, ampliando as capacidades do aplicativo para a comunidade de atléticas acadêmicas. Essas funcionalidades não finalizadas podem ser consideradas uma das possíveis áreas de desenvolvimento futuro do aplicativo, uma vez que se torna uma ferramenta central na interação e colaboração entre as atléticas e seus membros.

6. CONCLUSÃO

Este projeto de desenvolvimento do aplicativo “theta” oferece uma solução inovadora para o desafio da comunicação e colaboração eficaz entre as atléticas acadêmicas de diferentes instituições de ensino. Atualmente, a comunicação entre atléticas muitas vezes se limita à rede de contatos pessoais de cada atlética, tornando-a ineficiente e fragmentada. O “ConectAtletica” aborda essa lacuna, proporcionando uma plataforma unificada que promove a colaboração e o compartilhamento de experiências entre as atléticas.

Por meio da implementação do aplicativo, os resultados obtidos foram significativos. O aplicativo permitiu que as atléticas se conectassem de maneira mais eficaz, compartilhando informações sobre eventos esportivos, sociais e acadêmicos. Além disso, a capacidade de criar perfis e manter listas de contatos facilitou a interação entre os membros das atléticas, promovendo um ambiente mais colaborativo e unificado.

No entanto, alguns desafios e limitações surgiram durante o desenvolvimento do projeto. Devido a restrições de tempo, algumas funcionalidades planejadas, como a capacidade de agendar e organizar eventos diretamente no aplicativo, não foram totalmente implementadas. Essas funcionalidades representam oportunidades para futuras melhorias no aplicativo, à medida que ele continua a evoluir e receber atualizações.

Em resumo, o aplicativo “theta” demonstrou ser uma solução viável para melhorar a comunicação e colaboração entre as atléticas acadêmicas. Ele resolveu o problema de comunicação fragmentada, proporcionando uma plataforma que unifica as atléticas e promove a colaboração. Embora algumas funcionalidades não tenham sido finalizadas devido a restrições de tempo, o aplicativo representa um passo importante na direção certa, e futuras atualizações podem aprimorar ainda mais sua utilidade e eficácia.

REFERÊNCIAS

ASTAH. **Powerful and Fast UML Diagramming Software**, 2022. Disponível em: <https://astah.net/products/astah-uml/> Acesso em: 8 ago 2022.

BORGES, Joseane da Silva. **Gerenciamento de projetos na realização de jogos universitários**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) - Universidade Tecnológica Federal do Paraná, Medianeira, 2018. <http://repositorio.utfpr.edu.br/jspui/handle/1/12732>

FIREBASE. **Firestore**, 2022. Disponível em: <https://firebase.google.com/?hl=pt-br> Acesso em: 28 ago 2022.

FLUTTER. **Flutter - Beautiful native apps in record time**, 2022. Disponível em: <https://flutter.dev/> Acesso em: 14 set 2022. , 2018.

Google LLC. (Ano). **Google Forms**. <https://docs.google.com/forms>

OLIVEIRA, Joao Paulo de; PAZ, Kleiber Augusto de Araujo. **Desenvolvimento de aplicativo de rede social com foco em jogos: games connect**. Trabalho de conclusão de curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia Ribeirão Preto, Ribeirão Preto, 2022. <https://ric.cps.sp.gov.br/handle/123456789/13971>

Rovaris, Fernando Henrique. **Criar uma aplicação própria com o objetivo de alcançar a maior população possível torna-se inviável economicamente, principalmente quando se trata de instituições do terceiro setor que realizam ações sociais**. Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, 2023. <http://repositorio.unesc.net/handle/1/10342>

SCHWABER, K. **SCRUM Development process**. 1995. Disponível em: <http://jeffsutherland.com/oopsla/schwapub.pdf> Acesso: jul. 2003.



◀ CRIAÇÃO DE UM APLICATIVO PARA CONDOMÍNIOS UTILIZANDO *FLUTTER* E *REST API* VOLTADOS PARA A UTILIZAÇÃO DO MORADOR E PORTEIRO

14

*Development of an application for condominiums using Flutter
and a Rest API designed for use by residents and gatekeepers*

Pedro Henrique Silva de Sousa Carneiro

Edilson Carlos Silva Lima

Leonardo de Castro Mesquita

D.O.I.: 10.29327/5391075.1-14



Resumo

A segurança é o principal fator que leva as pessoas a escolherem morar dentro de condomínios fechados; no entanto, no Brasil, há inúmeros fatores de risco que fragilizam até mesmo esse sistema. Um desses fatores é a falta de praticidade e agilidade na entrada de convidados e prestadores de serviço. Portanto, foi planejado criar um aplicativo utilizando Flutter no front-end e desenvolver uma Rest API utilizando o Framework Spring para o back-end. Para atender a esse propósito, serão adotadas as metodologias de entrevista e estudo de caso. Essa aplicação será voltada para proporcionar ao porteiro a capacidade de analisar todos os convites feitos pelos moradores, proporcionando maior praticidade na autorização de pessoas para a entrada no condomínio e reservas dos espaços.

Palavras-chaves: Segurança, Rest API, flutter, back-end, front-end.

Abstract

The main factor driving people to choose living in gated communities is security; however, in Brazil, there are numerous risk factors that even undermine this system. One of these factors is the lack of convenience and speed in admitting guests and service providers. Therefore, it was planned to create an application using Flutter on the front-end and develop a Rest API using the Spring Framework for the back-end. To achieve this goal, the methodologies of interviews and case studies will be employed. This application will be focused on providing the gatekeeper with the ability to analyze all invitations made by residents, offering greater convenience in authorizing people to enter the condominium and to reserve spaces.

Keywords: Security, Rest API, flutter, back-end, front-end.

1. INTRODUÇÃO

A entrada segura e eficiente em condomínios é um aspecto crucial para a tranquilidade dos moradores e a proteção de suas propriedades. Condomínios estão constantemente buscando maneiras de aprimorar o controle de acesso, garantindo que apenas indivíduos autorizados entrem nas instalações. A necessidade de agilidade nesse processo é evidente, uma vez que moradores, visitantes e prestadores de serviços frequentemente enfrentam longas filas nos portões de entrada. Nesse contexto, a tecnologia desempenha um papel fundamental na melhoria da segurança e na otimização da entrada, tornando-a mais rápida e eficaz.

O desenvolvimento de um aplicativo para condomínios voltado para o uso do porteiro requer uma interface de usuário (UI) eficiente e amigável, conhecida como “front end”. O Flutter, uma estrutura de código aberto da Google, tem se destacado como uma escolha excepcional para o desenvolvimento de front end de aplicativos móveis. Sua capacidade de criar interfaces de usuário atraentes e responsivas para dispositivos Android e iOS em um único código-base tornou-o uma opção popular. Ao utilizar o Flutter, podemos garantir que o Tranca seja fácil de usar e proporcione uma experiência de entrada suave para os porteiros.

Por trás da interface do aplicativo do porteiro, a lógica de funcionamento é alimentada pela programação orientada a objetos (POO). A POO é uma metodologia de programação que se concentra na organização de dados e funcionalidades em objetos interconectados, facilitando a manutenção e a expansão do código. O Spring Java é um popular framework de desenvolvimento que oferece suporte à criação de aplicativos baseados em Java, tornando mais eficiente o desenvolvimento de sistemas robustos e escaláveis, permitindo o desenvolvimento de uma REST API (Interface de Programação de Aplicativos) permite que o aplicativo do porteiro se comunique de forma eficaz com outros sistemas e serviços, melhorando a interoperabilidade e a eficiência geral. Tendo em vista as informações anteriores, as metodologias empregadas foram de estudo de caso e entrevista.

Em resumo, o desenvolvimento de um aplicativo para condomínios voltado para o uso do porteiro é uma resposta direta à necessidade de melhorar a segurança e a agilidade na entrada de condomínios. Utilizando tecnologias como o Flutter para o front end e o Spring Java com uma REST API para o back end, é possível criar um sistema que atenda às demandas dos moradores e facilite o trabalho dos porteiros. Essa abordagem inovadora não apenas agiliza a entrada, mas também aumenta a segurança e a eficiência operacional, proporcionando uma solução moderna para os desafios enfrentados pelos condomínios nos dias de hoje.

2. TRABALHOS RELACIONADOS

Neste tópico serão abordados trabalhos que por meio do método de análise bibliográfica foram utilizados como base para o desenvolvimento deste artigo, visando inserir novas soluções que sejam compatíveis com os seguintes trabalhos.

2.1 Análise da utilização do framework Spring em sistema web para o aplicativo de gerenciamento de condomínio.

Este projeto teve como objetivo o desenvolvimento de uma API REST, garantindo relacionamentos de tabelas no banco de dados que são bem estruturadas para o controle de condomínios, fazendo com que sejam cadastrados usuários, condomínios e convites, na conclusão do trabalho foram apresentadas aberturas para trabalhos futuros. Por exemplo: criação de contatos para visitantes frequentes, e reserva de espaços do condomínio.

2.2 O uso de *Clean Architecture* com *Monorepo* para o desenvolvimento de um aplicativo no Flutter consumindo serviços de uma API REST

Este artigo foi o desenvolvimento do front-end de um aplicativo de controle de condomínio desenvolvido em flutter que foi criado telas de criação de convites, convites criados e uma home que mostra várias funcionalidades futuras, sendo elas convites e também a reserva de espaços, este front preparado para o consumo de REST API, que deixou abertura para criação de contatos frequentes e reserva de espaços no condomínio.

2.3 Segurança em condomínio: a importância do controle de acesso

Este artigo foi desenvolvido na intenção de alertar a relevância no controle de acesso de pessoas em condomínio, destacando a importância da fiscalização pessoal, ou seja, sendo feita por alguma pessoa capacitada e treinada na área de segurança para criar uma tranquilidade para os moradores dos condomínios, e frisou que somente o uso de tecnologias, não é suficiente para garantia dessa tranquilidade.

2.4 Diferencial do projeto desenvolvido

Analisando os trabalhos apresentados acima, foram incrementadas ao código da API a funcionalidade de reservar espaços do condomínio, assim como a criação de contatos frequentes para o morador, e para controlar a entrada de convidados e prestadores de serviços, foi desenvolvido o app para manuseio do porteiro, que somente ele possa confirmar a entrada ao condomínio.

3. FUNDAMENTAÇÃO TEÓRICA

Neste tópico serão abordados assuntos que foram tomados como base para a produção do projeto, estes foram adquiridos a partir de pesquisas, além disso também durante o curso de graduação. O objetivo é identificar e aprofundar sobre o que cada assunto trata, e mostrar o porquê de serem aplicáveis nas soluções do projeto.

3.1 Metodologia ágil

Segundo Giulia Losnak (2023), A metodologia ágil é uma forma de gerir projetos, em que se busca a otimização dos processos. Em vez de seguir um plano rígido — como na gestão tradicional —, na metodologia ágil, podemos fazer ajustes e melhorias ao longo

do desenvolvimento de projetos. Isso significa identificar e corrigir falhas rapidamente, garantindo resultados mais efetivos. Portanto, essa forma de gestão permite às pessoas gerentes uma abordagem moderna e eficiente para enfrentar desafios do mercado.

3.1.1 Scrum

O Scrum é um framework ágil que visa facilitar a colaboração, a adaptação e a entrega de valor de forma iterativa e incremental. Ele se baseia em três pilares fundamentais: transparência, inspeção e adaptação. O Scrum divide o trabalho em iterações chamadas de Sprints, que geralmente têm duração de duas a quatro semanas e utiliza artefatos como o Product Backlog e o Sprint Backlog, além de cerimônias como as reuniões de planejamento, revisão e retrospectiva (Losnak, 2023).

3.2 Programação orientada ao objeto

Segundo Gillis (2021), a programação orientada a objetos (POO) é um modelo de programação de computador que organiza o design de software em torno de dados ou objetos, em vez de funções e lógica. Um objeto pode ser definido como um campo de dados que possui atributos e comportamento exclusivos.

OOP se concentra nos objetos que os desenvolvedores desejam manipular, e não na lógica necessária para manipulá-los. Esta abordagem de programação é adequada para programas grandes, complexos e ativamente atualizados ou mantidos. Isto inclui programas de fabricação e design, bem como aplicativos móveis; por exemplo, OOP pode ser usado para software de simulação de sistema de fabricação.

A organização de um programa orientado a objetos também torna o método benéfico para o desenvolvimento colaborativo, onde os projetos são divididos em grupos. Os benefícios adicionais da OOP incluem reutilização de código, escalabilidade e eficiência.

- **Encapsulamento:** Encapsulamento refere-se à construção de um objeto de maneira a resguardar o acesso direto aos seus dados internos. Ao encapsular um objeto, agrupamos propriedades e métodos intimamente relacionados dentro do mesmo objeto, possibilitando o acesso a essas propriedades exclusivamente por meio de métodos públicos. Dessa maneira, abordamos questões cruciais, tais como a segurança e a confiabilidade do estado do objeto (Toledo, 2019).
- **Abstração:** Abstração implica em “ocultar” partes da implementação do objeto, revelando apenas uma interface simplificada para sua utilização. Considere, por exemplo, um forno de micro-ondas. Você não precisa compreender toda a complexidade do funcionamento interno, responsável por gerar ondas e produzir calor. Seu desejo é simplesmente pressionar um ou dois botões para obter uma refeição quente. Toda essa complexidade representa detalhes de implementação, os quais não é necessário conhecer (Toledo, 2019).
- **Herança:** A Herança é um método para evitar a repetição de código no qual, conforme o próprio nome indica, um objeto pode herdar características (ou seja, propriedades e métodos) de outra classe, sem a necessidade de reescrever essas mesmas características (Toledo, 2019).
- **Polimorfismo:** Poli significa muitas e Morphos significa forma, então Polimorfismo significa muitas formas. No paradigma de Programação Orientada a Objetos

(POO), o Polimorfismo ocorre quando duas ou mais classes possuem métodos com o mesmo nome, mas podem ter implementações diferentes. Dessa forma, torna-se viável utilizar qualquer objeto que implemente o mesmo método sem a necessidade de preocupação com o tipo do objeto (Toledo, 2019).

3.3 UML

Segundo Grady Booch, James Rumbaugh e Ivar Jacobson, A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.

3.3.1 Diagrama de Caso de Uso:

Na UML, os diagramas de caso de uso modelam o comportamento de um sistema e ajudam a capturar os requisitos do sistema. Os diagramas de caso de uso descrevem funções de alto nível e escopo de um sistema. Esses diagramas também identificam as interações entre o sistema e seus agentes. Os casos de uso e os agentes nos diagramas de caso de uso descrevem o que o sistema faz e como os agentes o usam, mas não como o sistema opera internamente (IBM, 2021).

3.3.2 Diagrama de Classes

Em UML, diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema. Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classe para modelar um sistema inteiro ou vários diagramas de classe para modelar os componentes de um sistema (IBM, 2021).

3.4 Back end

Segundo Ugo Roveda 2021 back-end é tudo aquilo que está por trás da interface de uma aplicação: seus sistemas, banco de dados, toda parte de segurança de dados, envio e recebimento de informações, armazenamento etc.

O back-end serve para que os sites, aplicativos, softwares ou outros tipos de sistemas de informação tenham todas as suas funcionalidades operando de maneira efetiva e cumprindo seus objetivos.

O back-end também serve para dar suporte aos projetos de tecnologia, unindo uma equipe multidisciplinar com uma visão mais abrangente do que está por trás de todos os processos.

3.4.1 API (*Application Programming Interface*)

Segundo Vitor de Andrade (2016) uma API - *Application Programming Interface*, ou interface de programação de aplicativos - é o que expõe as funcionalidades de um sistema. Ela é uma interface que permite a troca de informações entre os sistemas, independentemente de suas tecnologias ou detalhes das implementações.

Uma aplicação envia uma requisição à API, informando o que ela deseja fazer. Pode ser uma consulta aos dados do sistema, um cadastro de uma nova informação, uma atualização de dados etc. Por exemplo, em uma API de cadastramento de usuários, poderão ser solicitadas a criação de um novo usuário, uma lista de todos os usuários do sistema, as informações de um usuário etc. A partir daí, a API irá retornar uma resposta à aplicação, indicando se o processamento da requisição foi feito com sucesso ou não, e retornando quaisquer dados relevantes a ela.

3.5 Spring Framework

Segundo Cavalcante (2020), O Spring Framework é um framework desenvolvido para a plataforma Java, fundamentado nos padrões de projeto (Design Patterns), inversão de controle e injeção de dependência. Ele é composto por vários módulos abrangentes, capazes de impulsionar significativamente a aplicação Java.

3.5.1 Injeção de dependências

A injeção de dependência visa evitar o acoplamento de código em uma aplicação. Em termos simples, consiste na provisão de instâncias de classes que um objeto necessita, sem que esse objeto as instancie por si mesmo. Podemos afirmar que a injeção de dependência representa uma maneira de implementar a inversão de controle (Cavalcante, 2020).

3.5.2 Rest architecture

Segundo Lima (2020), REST (Representational State Transfer) é um modelo de arquitetura, não uma linguagem de programação ou tecnologia, que fornece orientação para sistemas distribuídos se comunicarem diretamente usando princípios e protocolos da Web existentes sem a necessidade de SOAP ou outro protocolo complexo

REST requer que o cliente faça uma solicitação ao servidor para enviar ou modificar dados. A solicitação inclui: um verbo ou método HTTP que define o que o servidor fará, um header (cabeçalho) com cabeçalhos de solicitação que transmitem informações sobre a solicitação, Informações no corpo da solicitação, essas informações são opcionais (Lima, 2020).

Na arquitetura REST tem os métodos que são cinco: O método GET é o método mais comumente usados e geralmente é usado para solicitar ao servidor o envio de recursos, o método POST foi projetado para enviar dados de entrada para o servidor. Na prática, é frequentemente usado para dar suporte a formulários HTML, o método PUT edita e atualiza o documento no servidor e método DELETE, como o nome sugere, exclui um dado ou uma coleção do servidor (Lima, 2020).

Cada resposta retornada por um aplicativo REST envia um código que define o status

da solicitação. Por exemplo: 200 (OK), a solicitação foi concluída com sucesso, 201 (CREATED), o objeto ou recurso foi criado com sucesso, 204 (SEM CONTEÚDO), o objeto ou recurso foi excluído com sucesso, 400 (BAD REQUEST), ocorreu um erro na solicitação (podem ser vários os motivos), 404 (Não Encontrado), rota ou coleção não encontrada, 500 (Erro de servidor interno), ocorreu algum erro de servidor (Lima, 2020).

3.6 ORM

Segundo Brunno Kriger, (2023), ORM (Object Relational Mapping, ou Mapeamento Objeto Relacional em português) é uma técnica de programação que visa facilitar a comunicação entre bancos de dados relacionais e linguagens de programação orientadas a objetos, como Java, Python e JavaScript.

Ela consiste em um conjunto de ferramentas que automatizam a persistência de dados em um banco de dados relacional, fazendo a conversão de dados de objetos para tabelas.

3.6.1 Hibernate

Segundo Kriger, (2023), o Hibernate destaca-se como uma das bibliotecas de Mapeamento Objeto-Relacional (ORM) mais amplamente utilizadas para Java. Possuindo suporte para uma vasta gama de bancos de dados relacionais, ele capacita os desenvolvedores a manipular dados de forma mais intuitiva, utilizando objetos Java em vez de recorrer diretamente à escrita de consultas SQL.

Além dessa funcionalidade fundamental, o Hibernate oferece recursos avançados que enriquecem o desenvolvimento de aplicações. Isso inclui a capacidade de realizar mapeamento de herança, suporte eficiente a cache de segundo nível e a habilidade de criar consultas dinâmicas, proporcionando flexibilidade e eficiência no acesso e manipulação de dados.

3.7 LOMBOCK

Segundo Silva (2021) o Lombok é um framework para Java que permite escrever código eliminando a verbosidade, o que permite ganhar tempo de desenvolvimento para o que realmente é importante. Seu uso permite gerar em tempo de compilação os métodos getters e setters, métodos construtores, padrão builder e muito mais.

3.7.1 Anotações importantes do Lombok

@Data: A @Data anotação é provavelmente a anotação usada com mais frequência no conjunto de ferramentas do Project Lombok. Ele combina a funcionalidade de @ToString, @EqualsAndHashCode, @Getter e @Setter.

Essencialmente, usar @Data em uma classe é o mesmo que anotar a classe com um padrão @ToString @EqualsAndHashCode, bem como anotar cada campo com ambos @Getter e @Setter. Anotar uma classe com @Data também aciona a geração do construtor do Lombok. Isso adiciona um construtor público que usa qualquer campo @NonNull ou

final como parâmetro. Isso fornece tudo o que é necessário para um Plain Old Java Object (POJO). (Michael Kimberlin, 2010).

@EqualsAndHashCode: Esta anotação em nível de classe fará com que o Lombok gere métodos equals e hashCode, já que os dois estão ligados intrinsecamente pelo hashCode contrato.

Por padrão, qualquer campo da classe que não seja estático ou transitório será considerado por ambos os métodos. Assim como @ToString, o exclude parâmetro é fornecido para evitar que um campo seja incluído na lógica gerada. Também pode-se usar o of parâmetro para listar apenas os campos que devem ser considerados. (Michael Kimberlin, 2010).

3.8 Front end

Segundo Kriger (2023), o front-end é uma área da programação que se dedica a criar a parte visual e interativa de um site, aplicativo ou software. É o que o usuário vê e usa quando acessa uma plataforma digital

O front-end serve para que os usuários possam interagir com uma aplicação; em outras palavras, para construir a interface de utilização de um site, aplicativo ou software.

3.9 Dart

Segundo Barro (2023), Dart é uma linguagem de programação multi-paradigma utilizada para desenvolvimento de aplicações web, aplicações móveis e de desktop. A linguagem Dart permite que o código criado rode tanto no lado do cliente quanto no servidor.

3.10 Flutter

Segundo Matheus Alberto (2023), Flutter é um framework (ferramenta) de desenvolvimento com ênfase em multiplataforma para dispositivos móveis. Desenvolvido pela Google, tem ampla aceitação no mercado e, mais recentemente, incorporou a capacidade de criar aplicações para desktop (Linux, Windows e macOS).

Na prática, o Flutter é uma ferramenta que viabiliza o desenvolvimento simultâneo de aplicativos para diferentes plataformas - Android e iOS - com um único conjunto de códigos. Assim, seus principais benefícios incluem versatilidade, uma curva de aprendizado reduzida e maior agilidade.

Este framework apresenta vantagens significativas para as empresas, permitindo que um único desenvolvedor crie aplicativos para diferentes sistemas operacionais, ao contrário do modelo nativo, que demanda dois desenvolvedores para cada plataforma.

4. METODOLOGIAS

As metodologias adotadas para a realização desse projeto foram estudo de caso no qual, estudou-se quais as melhores soluções possíveis para a problematização apresentada e também utilizamos entrevistas para poder analisar os resultados do projeto como um todo, que no qual foram propostas perguntas para pessoas que utilizaram os apps que



5. RESULTADOS E DICUSSÃO

Neste tópico serão demonstrados na prática os tópicos abordados anteriormente na fundamentação teórica, como por exemplo o desenvolvimento da API, utilizando o *Lombok* e suas anotações.

5.1 Descrição do sistema

O sistema desenvolvido é uma continuação do aplicativo para gerência de condomínios o Tranca, que tem como objetivo de facilitar a criação de convites para entrada no condomínio (para moradores e prestadores de serviço), assim como também reserva de espaços (quadra, piscina, churrasqueira). Desenvolvido em Scrum ele foi montado tanto no front-end em flutter, quanto a API feita com o spring framework, sendo feito 2 apps diferentes para morador e para o porteiro/administração.

5.2 Scrum

No desenvolvimento do aplicativo Tranca, adotou-se a metodologia Scrum, uma abordagem ágil e iterativa. O Product Backlog, colaborativamente estruturado com o Product Owner, direcionou as prioridades de funcionalidades. Durante as Sprints, as equipes de front-end e back-end, responsáveis pelo desenvolvimento em Flutter e Spring Framework, respectivamente, colaboraram para atender às metas estabelecidas. A realização de reuniões regulares, como o Daily Scrum e Sprint Review, promoveu a sincronização eficaz entre as equipes, permitindo feedback contínuo e aprimoramentos iterativos ao longo do ciclo de desenvolvimento.

5.3 LPOO

A linguagem de programação orientada a objetos, foi utilizada dentro do projeto, no momento de desenvolver a API dentro das modelagens das classes de JAVA.

5.3.1 Encapsulamento e abstração

O encapsulamento pode ser visto nos modificadores de acesso a classes e objetos, por exemplo temos os modificadores `public` e `private` como pode ser observado na Figura 1.

```

18 public class Usuario {
19
20     @EqualsAndHashCode.Include
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     @Column(name = "tr01idusuario")
24     private Long id;
25
26     @Column(name = "tr01nome", nullable = false, length = 45)
27     private String nome;

```

Figura 1. Java Class “Usuario”.

Fonte: Autoral, 2023.

A abstração pode ser encontrada dentro da classe Java que abstrai informações para a modelagem do objeto.

5.4 UML

A UML foi utilizada no desenvolvimento lógico do projeto, no momento de desenvolvimento de diagrama de casos de uso e o diagrama de classes.

5.4.1 Diagrama de casos de uso:

O diagrama de casos de usos é utilizado para diagramar que tipos de ações cada tipo de usuário pode tomar em relação ao sistema proposto, para este projeto foi feita a diagramação conforme a Figura 3.

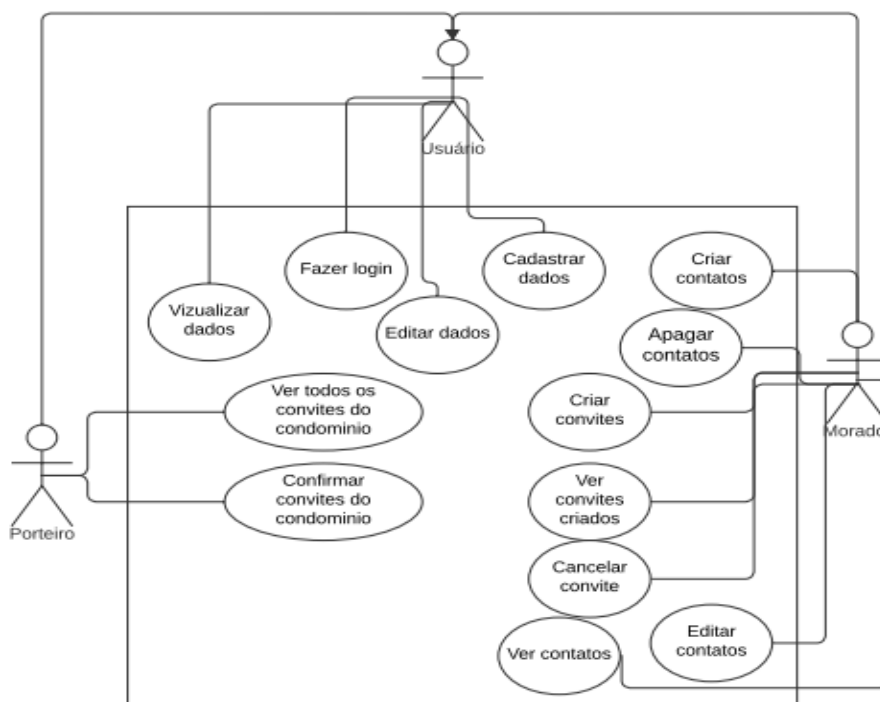


Figura 2. Diagrama de casos de uso do Tranca

Fonte: Autoral, 2023.

De acordo com a imagem acima pode ser visualizado cada ação que cada tipo de usuário é capaz de fazer dentro do condomínio, por exemplo somente o morador consegue adicionar contatos e criar convites e qualquer usuário consegue fazer login.

5.4.2 Diagrama de caso de usos:

O diagrama de classes é feito para determinar os relacionamentos entre as classes presentes dentro da estrutura do código da api, para assim determinar a estrutura das tabelas do banco de dados do programa, este modelo pode ser visto exemplificado pela Figura 2.

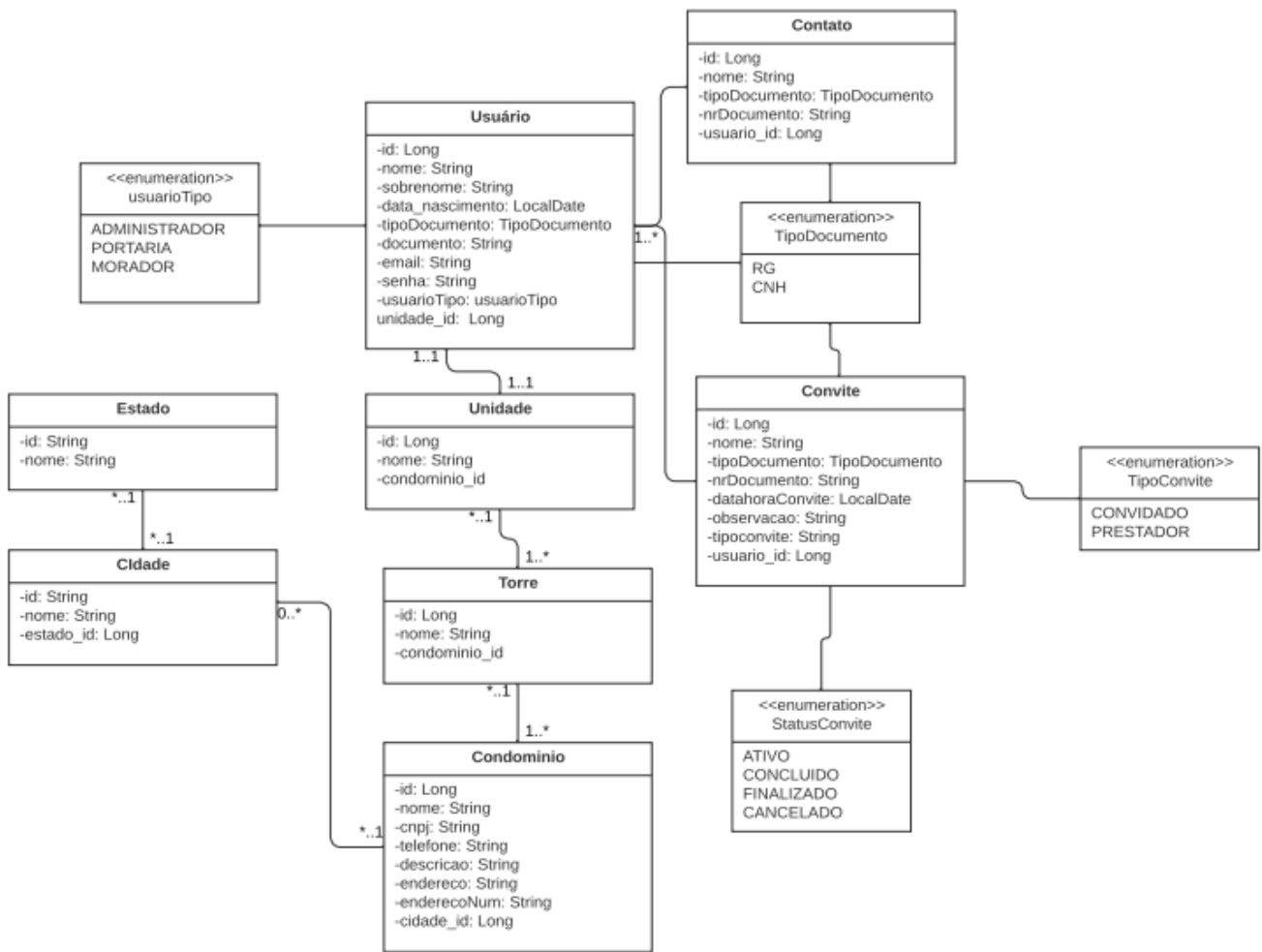


Figura 3. Diagrama de classe do Tranca

Fonte. Autoral, 2023

O diagrama acima mostra cada relacionamento entre essas classes, que são posteriormente, a base fundamental para a construção do banco de dados do aplicativo.

5.5 Implementação do Lombok

O Lombok entra como inovação nesse código com a anotação `@Data` que permite o desenvolvedor, com essa anotação um autoincremento dos *getters* e *setters* das classes e objetos da API, gerando assim muito mais facilidade e reduzindo a complexidade do desenvolvimento da API, como pode ser observado na Figura 4.

```

@JsonRootName("contato")
@Data
@Entity
@Table(name = "tr08contato")
public class Contato {

    @EqualsAndHashCode.Include
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "tr08idcontato")
    private Long id;

    @Column(name = "tr08nome", nullable = false, length = 60)
    private String nome;
}

```

Figura 4. Uso do @Data annotation

Fonte. Autoral, 2023

Dessa maneira o @Data cria todos os métodos *getters* e *setters* de todos os atributos dentro dos objetos no código.

5.6 Controllers

Os *controllers* são responsáveis pelos *endpoints* do projeto, que são os métodos de adicionar, deletar, listar e atualizar, estes *endpoints* foram testados no POSTMAN, como pode ser visto na Figura 5.

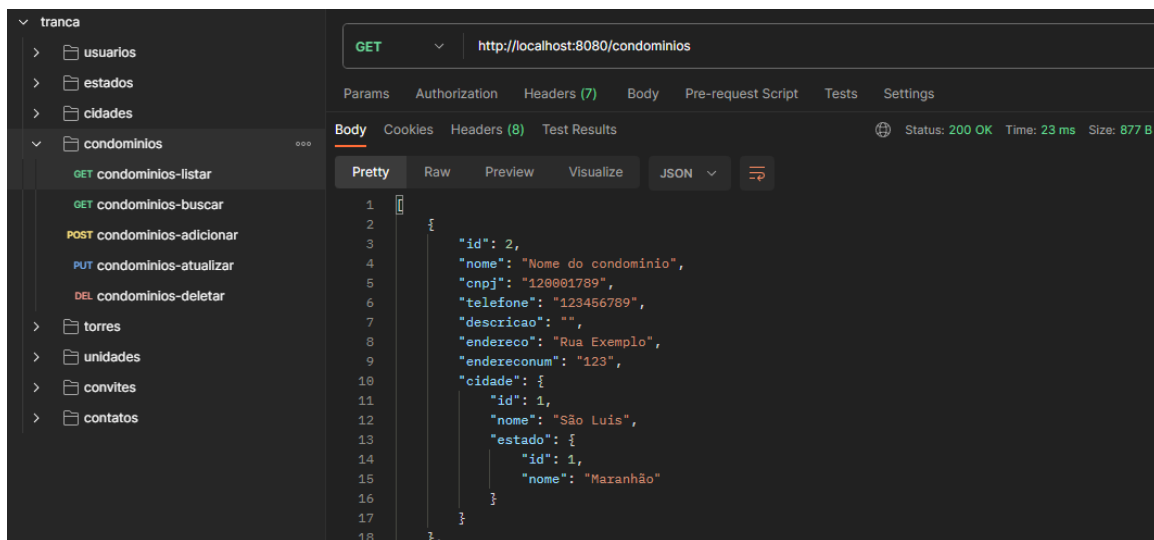


Figura 5. Listando condomínios no POSTMAN

Fonte. Autoral, 2023

Na imagem acima é utilizado o método *get* para poder listar todos os condomínios inseridos no banco de dados do Tranca.

5.7 Front-end

O desenvolvimento do front-end desse aplicativo foi feito de dois aplicativos, que são eles voltados, um para o morador e o outro para funções administrativas e para o porteiro. As figuras 6.a e 6.b mostram as interfaces de login e do porteiro respectivamente.



Figura 6. a. Tela de login; **b.** Tela inicial porteiro

Fonte. Autoral, 2023

A tela do porteiro serve para apresentar para ele, quais tipos de convites existem, são eles: ativos (foram feitos pelo morador e ainda não foram confirmados pelo porteiro), finalizados (São os que já foram confirmadas as entradas pelo porteiro), expirados (Os quais o porteiro não confirmou dentro do prazo de validade) e por fim os cancelados (que foram cancelados pelo morador que gerou o convite).

5.8 Entrevistas

Para avaliar a eficiência e praticidade do aplicativo foram feitas cinco perguntas sobre a usabilidade e efetividade do Tranca em uma amostra de 50 pessoas, dentre elas moradores e porteiros de condomínio.

5.8.1 Perguntas e respostas

1-Você considera que o processo de criação de convites no Tranca é intuitivo e fácil de usar?

R: 90% (45 pessoas) responderam Sim, 6% (3 pessoas) disseram Não e 4% (2 pessoas) não souberam responder.

2-O Tranca proporcionou maior praticidade e rapidez no momento de realizar reservas de espaços comuns no condomínio?

R: 84% (42 pessoas) responderam Sim, 8% (4 pessoas) disseram Não e 8% (4 pessoas) não souberam responder.

3-Você percebeu benefícios na utilização do recurso de contatos frequentes do Tranca para o controle de acesso?

R: 96% (48 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 2% (1 pessoa) não soube responder.

4-O Tranca agilizou a autorização de entrada de convidados e prestadores de serviço no seu condomínio?

R: 90% (45 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 8% (4 pessoas) não souberam responder.

5-A experiência geral com o Tranca contribuiu para uma interação mais eficiente entre moradores, porteiros e prestadores de serviço?

R: 94% (47 pessoas) responderam Sim, 2% (1 pessoa) disse Não e 4% (2 pessoas) não souberam responder.

6. CONCLUSÃO

O aplicativo Tranca, desenvolvido com Flutter para o front-end e uma REST API baseada no framework Spring para o back-end, emerge como uma solução inovadora para aprimorar a segurança e praticidade em condomínios. Ao incorporar melhorias como a criação de contatos frequentes e a reserva de espaços, o Tranca vai além de simples agilidade na entrada, priorizando também a segurança e eficiência operacional, alinhando-se às demandas contemporâneas. A escolha estratégica de tecnologias como Flutter e Spring, respaldada por conceitos fundamentais como programação orientada a objetos e metodologias ágeis, sustenta a robustez e eficácia do projeto. Com interfaces intuitivas tanto para moradores quanto para o porteiro, o Tranca representa uma resposta moderna e eficiente para os desafios enfrentados pelos condomínios na gestão de acesso.

A fundamentação teórica, que abrange temas como UML, back-end, API, Spring Framework, ORM (Hibernate), e Lombok, fornece alicerces sólidos para a implementação prática do Tranca. Os resultados e discussões destacam a aplicação efetiva desses conceitos, desde a modelagem de classes até a implementação do Lombok para simplificar o código. Os diagramas de classe e caso de uso evidenciam a estrutura lógica do sistema, reforçando a abordagem ágil e eficiente adotada no projeto. Em resumo, o Tranca não apenas atingiu, mas ultrapassou suas metas, emergindo como uma solução tecnológica abrangente e contemporânea para os requisitos dinâmicos dos condomínios, amalgamando eficiência, segurança e praticidade.

REFERÊNCIAS

ALBERTO, Matheus, **Flutter: o que é e tudo sobre o framework**, 2023. Disponível em: < <https://www.alura.com.br/artigos/flutter> > Acessado em 12/11/2023

ANDRADE, V., I: **UM ESTUDO SOBRE PADRÕES E TECNOLOGIAS**

BARRO, Bruna B., **Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem**, 2023. Disponível em : <<https://www.hostinger.com.br/tutoriais/linguagem-dart> > Acessado em 12/11/23

BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML, Guia do Usuário: tradução**; Fábio Freitas da Silva, Rio de Janeiro, Campus ,2012.

CAVALCANTE, Paulo; **Spring Framework: o que é, seus módulos e exemplos!**; 2020; Disponível



em: < <https://blog.geekhunter.com.br/spring-framework/> > Acessado em: 12/11/2023.

CORREIA, Marcelo da Conceição, **O uso de Clean Architecture com Monorepo para o desenvolvimento de um aplicativo no Flutter consumindo serviços de uma API REST**, 2022.

DIAS, E. M., CARDOSO, F. T., GONSALVES, J. D. S., FERREIRA, A. L. P. ; **SEGURANÇA EM CONDOMÍNIO: A IMPORTÂNCIA DO CONTROLE DE ACESSO**, 2016. Disponível em: < <https://www.linkedin.com/pulse/seguran%C3%A7a-em-condom%C3%ADnio-import%C3%A2ncia-do-controle-de-padi-lha-ferreira/> > Acessado em 28/10/2023

FERREIRA, C.E.J., **Análise da utilização do framework Spring em sistema web para o aplicativo de gerenciamento de condomínio**, 2022.

GILLIS, Alexander S. **object-oriented programming (OOP)**, 2021; Disponível em: <<https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP> > acessado 12/11/2023.

IBM, Documentação. **Diagramas de Caso de Uso**, IBM, 2021. Disponível em: < <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case> > acessado em: 12/11/2023.

IBM, Documentação. **Diagramas de Caso de Uso**, IBM, 2021. Disponível em: < <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams> > Acessado em 12/11/2023

KIMBERLIN, M., **REDUCING BOILERPLATE CODE WITH PROJECT LOMBOK**, 2010. Disponível em < <https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok> > Acessado em 12/11/2023

KRIGER, Bruno; **FRONT END: ENTENDA O QUE É, PARA QUE SERVE, COMO APRENDER ESSA ESPECIALIDADE!**, 2023. Disponível em: < <https://kenzie.com.br/blog/front-end/> > Acessado em 12/11/2023

KRIGER, Bruno; **ORM (OBJECT RELATIONAL MAPPER) – SAIBA O QUE É E IMPORTÂNCIA NA PROGRAMAÇÃO**, 2023. Disponível em: < <https://kenzie.com.br/blog/orm/> > Acessado em 12/11/2023

LIMA, Guilherme, **REST: Conceito e fundamentos**, 2020. Disponível em: < <https://www.alura.com.br/artigos/rest-conceito-e-fundamentos> > Acessado em 12/11/2023

LOSNAK, Giulia, **Metodologia Ágil - O que é?**, 2023. Disponível em: < <https://www.alura.com.br/artigos/o-que-e-metodologia-agil> > Acessado em 12/11/2023

PARA O DESENVOLVIMENTO WEB – BACK-END, Rio de Janeiro – RJ, Brasil, Escola politécnica, UFRJ, 2016.

ROVEDA, Ugo, **O QUE É BACK END, PARA QUE SERVE E COMO APRENDER EM 2021**, 2021. Disponível em: < <https://kenzie.com.br/blog/back-end/> > acessado em 12/11/2023

SILVA, Alex, **Como usar o Lombok em projetos Java**, 2021. Disponível em: < <https://www.dio.me/articles/como-usar-o-lombok-em-projetos-java> > Acessado em 12/11/2023

TOLETO, Sergio. **Os 4 pilares da Programação Orientada a Objetos**, Sergiotoleto, 2019. Disponível em: <https://www.sergiotoledo.com.br/artigos/os-4-pilares-da-programacao-orientada-a-objetos>. Acessado em 10/11/2023



AUTORES



Criando Soluções Tecnológicas com a Engenharia de Computação

CARNEIRO, Pedro Henrique Silva de Sousa

Graduado em Engenharia da Computação, é um profissional apaixonado pelo seu segmento, seu trajeto acadêmico na Universidade Ceuma o permitiu construir uma sólida base de conhecimentos sobre Engenharia de Computação. Com uma abordagem prática e orientada para resultados, possui experiência destacada em desenvolvimento de aplicações full-stack e automação. Ao longo de sua carreira acadêmica ajudou a desenvolver o projeto do aplicativo Tranca ao que sugere uma abordagem mais tecnológica e segura para o controle de entradas em condomínios. Sua habilidade em saber lidar com as adversidades no processo de criação de aplicativos e o compromisso com a inovação o classifica a ser uma ótima adição ao corpo de autores deste livro. O capítulo de Pedro Carneiro oferece uma perspectiva única sobre a “Criação de um aplicativo para condomínios utilizando Flutter e Rest API voltados para a utilização do morador e porteiro”, apresentando insights fundamentais e abordagens práticas que enriquecem o conhecimento dos leitores. Seu comprometimento com a excelência acadêmica e profissional reflete-se claramente nesta contribuição.

CARVALHO, André Kleysson Gois de

No capítulo elaborado pelo autor, ele apresenta uma perspectiva singular sobre a utilização da gamificação para o ensino da linguagem nordestina, oferecendo informações fundamentais e estratégias concretas que aprimoram a compreensão dos leitores. A sua dedicação à excelência, tanto no âmbito acadêmico quanto profissional, é nitidamente destacada por meio desta contribuição valiosa, ressaltando-o como um autor comprometido no contexto desta obra. A abordagem prática e os resultados das metodologias ágeis utilizadas foram fundamentais durante o desenvolvimento do aplicativo, juntamente com as habilidades distintas em programação e dedicação. A aplicação efetiva de metodologias ágeis, como Scrum, proporcionou uma gestão flexível e adaptável do projeto, permitindo ajustes rápidos conforme as necessidades surgiam. Essa agilidade foi crucial para superar desafios e garantir que o aplicativo atendesse às expectativas, mantendo um ciclo de desenvolvimento eficiente. Portanto, a sinergia entre uma abordagem prática, a aplicação de metodologias ágeis, proficiência em programação e comprometimento foi crucial para o êxito no desenvolvimento do aplicativo, culminando em um produto final sólido, eficaz e perfeitamente alinhado com as demandas e expectativas dos usuários.

COELHO, Marcos Levy de Freitas

Como autor do Capítulo 9. OcupApp: Controle Inteligente de Espaços, Marcos Levy destaca-se como um entusiasta apaixonado pela tecnologia. Sua experiência abrange o desenvolvimento de sites, aplicativos e robótica, com contribuições notáveis em projetos como o aplicativo OcupApp e um braço robótico inovador. No livro, seu capítulo aborda o Desenvolvimento do Aplicativo OcupApp usando Flutter, oferecendo insights valiosos e abordagens práticas. Marcos é reconhecido não apenas por suas habilidades técnicas, mas também por seu compromisso com a excelência acadêmica e profissional. Sua presença no corpo de autores do livro enriquece a obra, proporcionando uma perspectiva única sobre as complexidades do mundo tecnológico. Marcos Levy destaca-se como uma figura proeminente no campo, combinando paixão pela inovação com uma contribuição valiosa para o avanço da tecnologia.

CRUZ, Kaio Eduardo Pereira

Como autor do Capítulo 7 neste livro, trago uma paixão inabalável pela tecnologia, fundamentada em minha sólida formação em Engenharia de Computação pela Universidade Ceuma. Ao longo de minha trajetória profissional, tenho adotado uma abordagem prática e voltada para resultados, com experiência destacada em integração de front-end e back-end, além de ampla familiaridade com metodologias ágeis e frameworks de desenvolvimento. Durante minha carreira acadêmica, contribuí de forma significativa para uma variedade de projetos e realizações relevantes. Destaco meu envolvimento em projetos de pesquisa focados na otimização de sistemas web e no desenvolvimento de interfaces de usuário eficientes. Minha capacidade de analisar e resolver problemas complexos, combinada com meu compromisso com a inovação, me permitiu alcançar resultados excepcionais mesmo em ambientes desafiadores.

LIMA, Bruno Emanuel Santos

Como autor do capítulo 11 “Desenvolvimento da plataforma GUARANI usando VUE.JS e Spring Boot para divulgação de pesquisas científicas na web” um profissional dedicado e apaixonado por tecnologia, cujo percurso acadêmico na Universidade Ceuma o capacitou com sólidos conhecimentos em Engenharia de Computação. Com uma abordagem prática e orientada para resultados, possui experiência destacada em programação e robótica, desenvolvidas em diversos contextos profissionais. Ao longo de sua carreira acadêmica, contribuiu significativamente para projetos notáveis como a plataforma Guarani, um braço robótico inovador, entre outros projetos. Sua habilidade em desenvolvimento de sistemas, softwares e aplicativos é uma expressão tangível desse compromisso com a inovação o tornam uma adição valiosa ao corpo de autores deste livro. Apresentando insights fundamentais e abordagens práticas que enriquecem o conhecimento dos leitores. Seu comprometimento com a excelência acadêmica e profissional reflete-se claramente nesta contribuição.

LIMA, Kelve Antonio Oliveira

O “Ludo Educacional” correspondente da parta a qual fui incumbido representa a plataforma de acesso concebida para professores, capacitando-os a gerar perguntas e integrá-las em grupos específicos, nos quais se conectam diretamente com seus alunos. Essa solução visa otimizar a prática de ensino, oferecendo uma abordagem recreativa e dinâmica. Por meio de um ambiente interativo, tanto educadores quanto alunos podem participar ativamente, favorecendo um processo de aprendizado notável e mais envolvente. A implementação técnica envolve aspectos de engenharia de software, design de interface e integração de sistemas para assegurar a eficácia e a eficiência na plataforma em todo o contexto de aprendizado educacional.

MONTEIRO, Alanna Rocha

Emerge como a mente criativa por trás do envolvente capítulo 2 deste livro. Sua contribuição singular é refletida no conteúdo empático, perspicaz e autêntico apresentado no segmento intitulado ‘Desenvolvimento de app para gestão médica em Neurologia com Front End em Flutter e API REST com Spring Boot’. Ao explorar a rotina de uma unidade de terapia intensiva (**UTI**), percebeu-se a necessidade da praticidade na questão do preenchimento de formulários para avaliação do paciente neurológico, principalmente por conta da demanda, que, com o esforço, juntamente com a tecnologia, concretizou-se nos ótimos resultados. Graduada pela Universidade Ceuma, utilizou da sua formação acadêmi-

ca em Engenharia da Computação para efetivar, na prática, os conhecimentos obtidos ao longo desses anos, com exercícios em desenvolvimento Web e Mobile, especialmente com a criação do “NeuroApp”, apresentado neste livro, aliando inovação e tecnologia com interdisciplinaridade à área da saúde, de modo a também se destacar pela participação em projetos de robótica, como o desenvolvimento de prensa automatizada com microcontroladores e sensores de distâncias. Neste contexto, oferece uma perspectiva única e valiosa, enriquecendo, inclusive, a narrativa através do empenho em soluções inovadoras, pensamento analítico, conhecimento técnico e adaptabilidade para solução de problemas, além de comunicação afetiva com o nicho a ser explorado. Sendo assim, é inegável que Alanna Rocha detém a autoria e efetividade acerca do seu trabalho como coautora deste livro, destacando-se como um componente essencial à obra completa, pelo comprometimento acadêmico, excelência, além da organização e gestão de projetos, incluindo o entusiasmo por outras novas criações.

PEREIRA, Nikaelson Passos

Graduado em Engenharia de Computação, possui conhecimentos sólidos em análise e desenvolvimento de software. Atualmente trabalha com automação de processos administrativos e desenvolvimento de software para logística. Neste livro é autor do capítulo “Plataforma web desenvolvida em Node.JS e React para auxílio na aprendizagem de ciências exatas” onde com uma abordagem prática contribui para enriquecer o conhecimento dos leitores.

RODRIGUES, Ada Kezia Freitas Ribeiro

Como autora do Capítulo 6 “Aplicativo Mobile que fornece Apoio Pedagógico e psicossocial ao corpo discente e docente”, sobre o aplicativo “NADD” neste livro, compartilho um feito de reflexão e de pesquisa, apresento-me como uma profissional dedicada e fascinada por tecnologia, cujo percurso acadêmico na Universidade Ceuma, capacitou com sólidos e ricos conhecimentos em Engenharia de Computação. Com uma abordagem voltada para a prática e focada em resultados, possui experiência destacada em desenvolvimento de sites, Aplicativos, Robótica. Ao longo da carreira acadêmica contribui significativamente para o desenvolvimento do aplicativo mobile usando o framework Flutter e interação com o IntelliJ IDEA e MySQL Workbench, para criar uma resposta revolucionária atendendo a problemática abordada nesta pesquisa. Sua Habilidade em Desenvolvimento Mobile e o comprometimento em alto grau, transparece neste capítulo do livro, O capítulo neste livro oferece uma perspectiva única sobre o aplicativo NADD que fornece apoio ao corpo discente e docente. Apresentando insights fundamentais e abordagem que enriquecem o conhecimento dos leitores.

RODRIGUES, Mateus da Rocha

Como autor do Capítulo 1 deste livro, sou um engenheiro de computação com habilidades sólidas em desenvolvimento front-end, especialmente em Javascript, Reactjs e Nextjs. Neste capítulo, ofereço uma análise abrangente sobre “Estratégias de Otimização de Desempenho em Aplicações Web com o Google Lighthouse”. Meu objetivo é fornecer insights valiosos e práticas fundamentais para os leitores interessados na área. A dedicação à excelência acadêmica e profissional é evidente nesta contribuição, refletindo meu compromisso com o avanço da tecnologia e minha busca contínua por inovação.

SANTOS, Carlos Magno Souza dos

Como autor do capítulo 13 “Conectando Atléticas Acadêmicas com flutter” sobre o aplicativo “THETA” no livro, apresento-me como profissional comprometido com a inovação e tecnologia estou comprometido em transformar a interação entre atléticas acadêmicas. formado em Engenharia de Computação pela Universidade Ceuma, adquiri muitos conhecimentos essenciais para enfrentar desafios tecnológicos e complexos. Minha abordagem prática e orientada para resultados com metodologia ágil foi fundamental durante o desenvolvimento do aplicativo, utilizando o framework Flutter e a integração do Firebase para criar uma solução inovadora, consegui unir o útil ao agradável e o resultado foi esse projeto. Minha habilidade em desenvolvimento mobile e meu compromisso com a excelência refletem-se no capítulo deste livro, oferecendo uma visão única sobre como o aplicativo “theta” revoluciona a comunicação e colaboração entre atléticas acadêmicas. A experiência adquirida ao abordar desafios reais, como a limitação na comunicação entre atléticas, moldou minha contribuição para este livro. O capítulo que apresento oferece insights fundamentais e abordagens práticas, enriquecendo o conhecimento dos leitores sobre a importância da inovação tecnológica no contexto das atléticas acadêmicas.

SANTOS, Mayron Sergio

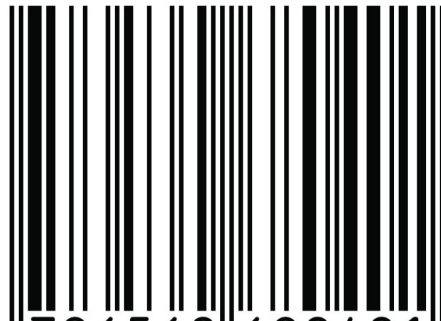
É um engenheiro de computação com sólidos conhecimentos em desenvolvimento backend, especialmente em Java e Spring Framework, adquiridos durante sua jornada acadêmica na Universidade Ceuma. Sua participação ativa no desenvolvimento do aplicativo City Connect demonstra sua capacidade de liderança e habilidade em resolver desafios complexos. Como autor do Capítulo 10 deste livro, ele oferece uma análise detalhada sobre ‘O Papel do Spring Framework no Desenvolvimento do Aplicativo City Connect’, apresentando insights valiosos e práticas fundamentais para os leitores interessados na área. Sua dedicação à excelência acadêmica e profissional é evidente em sua contribuição, refletindo seu compromisso com o avanço da tecnologia e sua busca contínua por inovação.

SOUSA, Thiago Vinicius Araújo de

Como autor do capítulo “Desenvolvimento de Aplicativo Para Gerenciamento de Finanças Pessoais”, apresento-me como um profissional dedicado e entusiasta da tecnologia, com uma formação em Engenharia de Computação pela Universidade Ceuma. Ao longo de minha jornada acadêmica, acumulei conhecimentos e experiência destacada em áreas específicas, contribuindo significativamente para projetos notáveis que moldaram minha trajetória. O capítulo que apresento oferece uma perspectiva única sobre o tema do desenvolvimento de aplicativos para gerenciamento de finanças pessoais. Com abordagens práticas, no qual busco enriquecer o conhecimento dos leitores, proporcionando uma visão abrangente e relevante. Minha dedicação à excelência acadêmica e profissional se reflete de maneira clara nesta contribuição, e estou entusiasmado para compartilhar meu conhecimento com os leitores deste livro. Meus agradecimentos serão à minha família que me apoiou todo tempo durante minha vida acadêmica e a Deus por me dar saúde e proteção em toda minha caminhada.

ISBN: 978-65-6068-040-1

BR



9 786560 680401

