

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS – *CAMPUS* SÃO JOÃO EVANGELISTA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

José Guilherme Verissimo Andrade

Vanderson Tiago de Paulo

**DESENVOLVIMENTO DE UM APLICATIVO PARA ORIENTAÇÃO E
APRESENTAÇÃO DAS FUNCIONALIDADES DA UAN AOS DISCENTES DO IFMG
- *CAMPUS* SÃO JOÃO EVANGELISTA**

São João Evangelista

2023

JOSÉ GUILHERME VERISSIMO ANDRADE

VANDERSON TIAGO DE PAULO

**DESENVOLVIMENTO DE UM APLICATIVO PARA ORIENTAÇÃO E
APRESENTAÇÃO DAS FUNCIONALIDADES DA UAN AOS DISCENTES DO IFMG
- *CAMPUS SÃO JOÃO EVANGELISTA***

Trabalho de Conclusão de Curso apresentado ao Curso Bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais – *Campus São João Evangelista* para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Me. Rosinei Soares de Figueiredo

São João Evangelista

2023

A553d Andrade, José Guilherme Verissimo.

Desenvolvimento de um aplicativo para orientação e apresentação das funcionalidades da UAN aos discentes do IFMG *campus* São João Evangelista / José Guilherme Verissimo Andrade, Vanderson Tiago – 2023.

66f.: il.

Orientador: Prof. Me. Rosinei Soares de Figueiredo.

Trabalho de Conclusão de Curso (bacharelado em Sistemas de Informação) – Instituto Federal Minas Gerais. *Campus* São João Evangelista, 2023.

1. React Native. 2. Aplicativo móvel. 3. UAN. 4. Tickets de refeição. I. Andrade, José Guilherme Verissimo. II. Vanderson Tiago III. Instituto Federal de Minas Gerais *Campus* SJE. IV. Título.

CDD 005.1

Catálogo: Esther Soares Cunha - CRB-6/MG-003372/P

JOSÉ GUILHERME VERISSIMO ANDRADE

VANDERSON TIAGO DE PAULO

**DESENVOLVIMENTO DE UM APLICATIVO PARA ORIENTAÇÃO E
APRESENTAÇÃO DAS FUNCIONALIDADES DA UAN AOS DISCENTES DO IFMG
- CAMPUS SÃO JOÃO EVANGELISTA**

Trabalho de Conclusão de Curso apresentado ao Curso Bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais – *Campus* São João Evangelista para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Me. Rosinei Soares de Figueiredo

Aprovado em 24/11/2023 pela banca examinadora:

Prof. Me. Rosinei Soares de Figueiredo (Orientador) – IFMG - SJE

Prof. Me. Eduardo Augusto Costa Trindade

Analista de TI – João Antônio Gomes de Almeida Júnior – IFMG - SJE

Dedicamos este trabalho a todas as pessoas que, de alguma forma contribuíram para a sua realização, e aos nossos familiares, que sempre nos apoiaram.

AGRADECIMENTOS

Eu, José Guilherme Veríssimo Andrade, primeiramente, expresso minha gratidão a Deus por ter me dado a força e coragem necessárias para alcançar este momento, que representa a realização de um sonho.

Agradeço de todo o coração aos meus pais, José Geraldo e Maria Aparecida, que sempre acreditaram no meu potencial e me deram um apoio incondicional. Sou profundamente grato por todo o suporte e encorajamento que recebi deles ao longo dessa caminhada rumo à realização deste sonho.

Agradeço imensamente a toda a minha família: irmãos (Mariana e José Augusto), tios, primos, amigos e padrinhos. Também quero expressar minha eterna gratidão àquelas pessoas que tanto iluminaram minha vida e já partiram desta terra, como meus amados avós, Luzia e Vicente, minha madrinha, Ana, e meu fiel amigo, Paulo. Enfim, agradeço do fundo do meu coração a todos vocês, pois vocês são peças fundamentais na minha vida.

Agradeço com uma enorme gratidão ao Instituto Federal de Minas Gerais por me proporcionar essa valiosa oportunidade de aprendizado. Além disso, quero agradecer a todos os membros do corpo discente e docente que estiveram ao meu lado, apoiando-me e ajudando-me ao longo desta jornada acadêmica. Em especial, sou imensamente grato aos meus professores Eduardo Trindade e Rosinei Soares. Suas contribuições significaram muito para o meu desenvolvimento pessoal e profissional sou verdadeiramente grato pela presença inspiradora em minha vida acadêmica.

Gostaria de expressar minha gratidão ao meu amigo Vanderson Tiago por aceitar esse desafio ao meu lado e por todo o esforço e conhecimento dedicados ao desenvolvimento deste trabalho.

Gostaria de expressar meu sincero agradecimento a todos os membros da turma SI201. Vivenciar essa etapa da minha vida ao lado de vocês foi uma experiência incrível. Em especial Luan, Mateus Henrique, Mateus Dias, Arthur e Daniel que se tornaram companheiros inseparáveis nessa jornada. Suas contribuições, amizade e apoio foram inestimáveis e fizeram dessa experiência acadêmica uma lembrança ainda mais significativa. Sou grato por ter compartilhado momentos de aprendizado, desafios e conquistas com cada um de vocês.

Gratidão, a mim mesmo pela coragem e persistência de correr atrás dos meus sonhos.

Eu, Vanderson Tiago de Paulo, quero começar expressando minha profunda gratidão a Deus. Sua presença constante em minha vida e Sua orientação divina foram

fundamentais para a conclusão deste trabalho. Sou infinitamente grato por Sua sabedoria e bênçãos que me acompanharam ao longo dessa jornada.

Aos meus amados pais, Geraldo Izidorio e Rosângela Rocha, pelo apoio incondicional, amor e encorajamento proporcionados ao longo desta jornada. Vocês têm sido pilares de força e fontes constantes de inspiração em todos os momentos da minha vida, incluindo a realização deste trabalho acadêmico.

À minha irmã Valquiria e meu cunhado Gleison, pela presença constante e pelo apoio inigualável. Sua presença significou muito para mim, e sou imensamente grato por estarem sempre presentes, prontos para ajudar e oferecer seu apoio valioso.

Agradeço imensamente a todos os professores que estiveram me acompanhando nessa jornada, principalmente aos professores Eduardo Trindade e Rosinei Soares. Gostaria de expressar minha gratidão por todo o apoio que me deram. A ajuda que vocês proporcionaram, compartilhando conhecimento e dedicando seu tempo e energia para me ajudar a alcançar meus objetivos, foi inestimável. Vocês deixaram um legado duradouro e inspirador em minha jornada acadêmica, sendo verdadeiras fontes de inspiração para mim.

À equipe de TI, pela incrível disposição em ajudar e responder às dúvidas que surgiram ao longo deste trabalho. Sou extremamente grato por toda a assistência valiosa que vocês forneceram.

Ao meu amigo José Guilherme, por sua colaboração e trabalho em equipe ao longo deste projeto. Sua contribuição foi inestimável, e nosso trabalho conjunto foi fundamental para alcançarmos os resultados. Sua presença e apoio foram essenciais para o sucesso deste trabalho. Obrigado pela parceria e pela amizade durante todo o processo.

Aos meus colegas das turmas 201 e 211. O apoio de vocês ao longo dessa jornada acadêmica foram excepcionais. Compartilhar experiências, conhecimentos e desafios com vocês enriqueceu meu aprendizado e contribuiu para o crescimento mútuo.

Gostaria de estender meus sinceros agradecimentos também ao IFMG, por todas as oportunidades e por abrir tantas portas. Sou grato ao corpo docente, à direção e a cada um dos funcionários que trabalharam nos bastidores para garantir o funcionamento eficiente da instituição.

E a todos que contribuíram de alguma maneira para a minha formação e a execução deste trabalho, meu muito obrigado.

“Só vive o propósito quem suporta o processo”

André Mansur

RESUMO

Este trabalho teve como objetivo principal a concepção e implementação de um aplicativo móvel multiplataforma destinado ao sistema operacional Android, para disponibilização de informações referentes Unidade de Alimentação e Nutrição (UAN) do Instituto Federal de Minas Gerais, Campus São João Evangelista. O aplicativo desenvolvido visa aprimorar a comunicação entre a UAN e seus usuários, fornecendo funcionalidades essenciais para uma experiência mais conveniente e eficiente. A aplicação oferece aos usuários acesso imediato aos avisos e comunicados da UAN, garantindo que informações cruciais sejam amplamente divulgadas e facilmente acessíveis. Além disso, o aplicativo disponibiliza o cardápio semanal da UAN, permitindo que os usuários planejem suas refeições com antecedência, de acordo com suas preferências alimentares e necessidades dietéticas. O desenvolvimento deste aplicativo envolveu a utilização de linguagens de programação modernas, frameworks e ferramentas apropriadas para criar uma solução robusta e eficaz. O processo de desenvolvimento abrangeu a análise de requisitos, prototipagem, codificação. A fim de obter ao final do processo, um aplicativo funcional, simples, seguro e rápido, alcançando então os objetivos estipulados.

Palavras-chave: React Native. Aplicativo Móvel. UAN. Tickets de Refeição.

ABSTRACT

This work aimed at the conception and implementation of a cross-platform mobile application intended for the Android operating system, with a focus on providing information for the Dining Services (UAN) at the Federal Institute of Minas Gerais, São João Evangelista Campus. The developed application seeks to enhance communication between the UAN and its users by providing essential features for a more convenient and efficient experience. The application grants users immediate access to UAN notices and announcements, ensuring that critical information is widely disseminated and easily accessible. Additionally, the app presents the UAN's weekly menu, allowing users to plan their meals in advance according to their dietary preferences and requirements. The development of this application involved the utilization of modern programming languages, frameworks, and appropriate tools to create a robust and effective solution. The development process followed a well-defined lifecycle, encompassing requirements analysis, prototyping, coding, rigorous testing, and validation. With the aim of achieving a functional, user-friendly, secure, and fast application, the project's ultimate goal was to meet the established objectives

Keywords: React Native. UAN. Mobile Application. Meal Tickets.

LISTA DE ILUSTRAÇÕES

Figura 1 - Editores de códigos mais usados	24
Figura 2 - Desenvolvimento nativo	28
Figura 3 - Desenvolvimento multiplataformas	34
Figura 4 - Arquitetura React Native	38
Figura 5 - Arquitetura MVC	42
Figura 6 - Diagrama de Comunicação HTTP entre Navegador e Servidor	43
Figura 7 - VsCode em conjunto com Android Studio	48
Figura 8 - Tela de home do aplicativo	50
Figura 9 - Design do aplicativo na ferramenta Figma	53
Figura 10 - Diagrama de Caso de uso do aplicativo SIAR	55
Figura 11 - Diagrama de Atividade - Cardápio	56
Figura 12 - Diagrama de Atividade - Tickets	57
Figura 13 - Diagrama de Atividades - Avisos	58
Figura 14 - Tela de login	59
Figura 15 - Tela de home do aplicativo	60
Figura 16 - Tela de avisos do aplicativo	61
Figura 17 - Tela de tickets do aplicativo	62
Figura 18 - Tela de agendamento de final de semana	63
Figura 19 - Tela de cardápio semanal	64

LISTA DE TABELAS

Tabela 1 - Vantagens e Desvantagens do desenvolvimento nativo.....	30
Tabela 2 - Comparação entre objective-C e Swift.....	32
Tabela 3 - Popularidade das Linguagens de Programação	33
Tabela 4 - Avaliação do Aplicativo.....	65
Tabela 5 - Avaliação do aplicativo com usuários.....	66

LISTA DE ABREVIATURAS E SIGLAS

APP - *Application*

CSS - *Cascading Style Sheets*

GPS - *Global Positioning System*

HTML - *Hypertext Markup Language*

HTTP - *Hypertext Transfer Protocol*

IDE - *Integrated Development Environment*

IFMG - Instituto Federal de Minas Gerais

IFPB - Instituto Federal da Paraíba

IOS - *iPhone Operating System*

IPA - *iOS App Store Package*

JS - *JavaScript*

MVC - *Model-View-Controller*

PIB - Produto Interno Bruto

PHP - *Hypertext Preprocessor*

RA - Registro Acadêmico

REST - *Representational State Transfer*

RF - Requisitos Funcionais

RNF - Requisitos Não Funcionais

SIAR - Sistema de Auxílio ao Refeitório

TI - Tecnologia da Informação

UAN - Unidade de Alimentação e Nutrição

Ui - *User Interface*

URL - *Uniform Resource Locator*

USB - *Universal Serial Bus*

UX - *User Experienc*

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo Geral	17
1.2	Objetivos Específicos	17
1.3	Justificativa	17
2	REFERENCIAL TEÓRICO	19
2.1	Unidade de Alimentação e nutrição	19
2.1.1	<i>Fluxo de trabalho da UAN</i>	19
2.2	Computação Móvel	20
2.2.1	<i>Evolução da computação móvel</i>	20
2.2.2	<i>Surgimento dos sistemas operacionais Android e iOS</i>	21
2.3	Desenvolvimento de Softwares para Dispositivos Móveis	22
2.3.1	<i>Ambiente de Desenvolvimento integrado</i>	23
2.3.2	<i>Emuladores de dispositivos móveis</i>	24
2.3.3	<i>Tipos de aplicativos móveis</i>	25
2.4	Desenvolvimento nativo	27
2.4.1	<i>Vantagens e desvantagens do desenvolvimento nativo</i>	28
2.4.2	<i>Linguagens de programação para desenvolvimento nativo</i>	30
2.5	Desenvolvimento multiplataforma	33
2.5.1	<i>Vantagens e desvantagens do desenvolvimento multiplataforma</i>	35
2.5.2	<i>JavaScript para desenvolvimento multiplataforma (ES6)</i>	35
2.5.3	<i>Frameworks multiplataforma</i>	36
2.5.3.1	<i>React Native</i>	37
2.6	Integração de APIs no Desenvolvimento de Aplicativos	39
2.6.1	<i>Utilização de PHP no backend de aplicativos</i>	40
2.7	Trabalho Relacionados	43
3	METODOLOGIA	45
3.1	Procedimentos metodológicos	45
3.2	Ferramentas e tecnologias utilizadas	47
3.3	Prototipagem	49
4	RESULTADOS E DISCUSSÃO	51

4.1	Requisitos do sistema	51
4.1.1	Requisitos Funcionais	51
4.1.2	Requisitos Não Funcionais	52
4.2	Sessão de modelagem	54
4.3	Aplicação desenvolvida	58
4.4	Avaliação do sistema	64
5	CONSIDERAÇÕES FINAIS.....	67
	REFERÊNCIAS.....	68

1 INTRODUÇÃO

Os aplicativos móveis têm se tornado cada vez mais relevantes no cotidiano das pessoas, sendo utilizados para entretenimento, comunicação, negócios e outras funcionalidades que facilitam a realização de tarefas e a interação com outras pessoas e empresas. Aplicativos como WhatsApp, Facebook e Instagram são exemplos de ferramentas de comunicação e colaboração que permitem conexão com amigos, familiares e colegas de trabalho em qualquer lugar e a qualquer hora, tornando a comunicação mais rápida e eficiente.

De acordo com Wu (2018), "As aplicações móveis estão desempenhando um papel cada vez mais importante em nossas vidas diárias". Wu também observa que desde novembro de 2016, mais tráfego de rede é gerado por dispositivos móveis em comparação com desktops/laptops. Isso sugere que os dispositivos móveis estão se tornando cada vez mais populares para navegar na Internet, usar aplicativos e consumir conteúdo online.

Os dispositivos móveis têm transformado a maneira como as pessoas interagem com as empresas e realizam suas tarefas cotidianas. Como afirmado por Danielsson (2016), desde o lançamento do primeiro smartphone, os aplicativos móveis têm experimentado um rápido crescimento em termos de uso e demanda. Muitas empresas têm reconhecido essa tendência e estabelecido serviços por meio de aplicativos móveis para melhorar a experiência do usuário e aumentar a eficiência dos negócios.

À medida que os smartphones e tablets se popularizam, as empresas e instituições também estão aproveitando as tecnologias móveis como uma ferramenta eficaz de comunicação e engajamento com o público-alvo. Como destacado por um artigo do TechTarget (2020), os aplicativos móveis oferecem uma comunicação conveniente de uma ou duas vias entre pequenas e médias empresas e seus clientes, permitindo que elas se comuniquem de maneira mais eficiente e melhorando a experiência do usuário.

A utilização de aplicativos móveis tem se mostrado uma estratégia relevante para melhorar o relacionamento e satisfação de clientes e usuários. De acordo com a MadeinWeb (2017), através do aplicativo, as empresas podem estreitar a relação com seus clientes ao enviar mensagens instantâneas e notificações diretamente aos usuários. É destacado ainda, que ao realizar o download do aplicativo em seu smartphone, o cliente pode acessar a empresa de forma mais ágil e simplificada, o que pode resultar em uma relação mais próxima e satisfatória entre as partes.

Uma tecnologia que vem se destacando atualmente no mercado para a criação de aplicativos móveis é o React Native¹, framework para desenvolvimento nativo de aplicativos, tem se sobressaído como uma tecnologia viável e eficaz para a construção de aplicativos móveis multiplataforma, em especial para as principais plataformas disponíveis no mercado, como iOS e Android.

Como mencionado por Wu (2018), o React Native possui uma das comunidades mais fortes no mundo de código aberto, sendo o terceiro projeto mais popular no GitHub em março de 2018. Isso sugere que o React Native é uma tecnologia bem estabelecida e popular, com uma ampla gama de contribuidores, incluindo grandes empresas como Microsoft e Samsung, o que reforça sua viabilidade como solução para a construção de aplicativos multiplataforma.

Este destaque do React Native se deve às suas características, que incluem a possibilidade de reutilização de código, o que traz ganhos de produtividade e agilidade no desenvolvimento, bem como a capacidade de proporcionar uma experiência de usuário mais fluida e consistente em diferentes dispositivos móveis. Além disso, o React Native é uma tecnologia de código aberto, o que significa que é continuamente aprimorado e atualizado pela comunidade de desenvolvedores, contribuindo para a sua robustez e confiabilidade.

Gikas e Grant (2013) afirmam que os dispositivos móveis, como os smartphones, podem ser uma ferramenta valiosa, pois permitem que estes acessem informações e recursos a qualquer momento e em qualquer lugar. Além disso, os autores ressaltam que a adoção de aplicativos móveis pelas instituições de ensino superior pode melhorar a comunicação com os alunos, como, por exemplo, nas unidades de alimentação e nutrição (UANs). Isso possibilita a entrega ágil e eficiente de informações importantes e relevantes.

O setor de alimentação e nutrição (UAN) é uma parte importante da gestão de refeições para diversas instituições, incluindo escolas, hospitais e empresas. Neste trabalho, a UAN é uma unidade que trabalha prestando serviços relacionados à alimentação, nutrição e saúde. Seu objetivo principal é fornecer serviços como a elaboração de cardápios balanceados e nutritivos, que atendam às necessidades dos usuários, o preparo de refeições de acordo com as orientações nutricionais e as normas sanitárias, a distribuição de alimentos após o preparo, o controle de qualidade dos alimentos.

Assim, ela desempenha um papel fundamental na promoção da saúde e da qualidade de vida, garantindo que os alunos e servidores das instituições de ensino recebam alimentos

¹ <https://reactnative.dev/>

saudáveis e de qualidade, a fim de manter, melhorar ou recuperar a saúde dos usuários atendidos. Além disso, adequada à cultura alimentar e à vocação agrícola da região, segura em relação aos aspectos higiênico-sanitários e ajustada aos limites financeiros da instituição. O desenvolvimento de um aplicativo móvel para a UAN pode ajudar na divulgação de informações importantes para os usuários, além de estreitar o relacionamento da UAN com os mesmos.

1.1 Objetivo Geral

O objetivo deste trabalho foi desenvolver um aplicativo móvel, com foco na otimização da comunicação entre a Unidade de Alimentação e Nutrição (UAN) e os estudantes. Com isso, buscou-se criar um canal de comunicação que facilitasse o acesso às informações pertinentes, contribuindo para uma gestão mais eficiente da UAN.

1.2 Objetivos Específicos

Com o intuito de alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- a) Levantar requisitos do aplicativo;
- b) Preparar o ambiente de desenvolvimento;
- c) Modelar e implementar o aplicativo;
- d) Realizar testes com usuários e corrigir possíveis erros;

1.3 Justificativa

A Unidade de Alimentação e Nutrição (UAN) é responsável por prover a alimentação diária de um grande número de estudantes. Nesse sentido, a comunicação eficiente entre a UAN e os estudantes é de extrema importância para assegurar que os alunos tenham acesso às informações pertinentes sobre o cardápio, bem como sobre quaisquer avisos ou comunicados relevantes relacionados à alimentação na instituição.

Atualmente, apesar de dispor de meios de comunicação com os estudantes do campus, o esforço para algumas atividades ou obtenção de informações entre as partes pode ser

algo a ser melhorado. Por exemplo, a consulta de tickets exige que o estudante se dirija ao local de venda dos tickets para verificar seu saldo disponível, o que pode levar a longas filas e atrasos desnecessários. Uma solução mais eficaz seria disponibilizar essa informação de maneira mais acessível e rápida. Outro exemplo diz respeito ao quadro de avisos da UAN, que pode ser acessado atualmente apenas por meio do site do IFMG, o que cria uma barreira para o usuário, que precisa navegar até o site para encontrar os avisos. Isso pode resultar na perda de informações importantes e gerar dúvidas e conflitos desnecessários.

Nesse contexto, a criação de um aplicativo móvel específico para a UAN do IFMG pode contribuir significativamente para aprimorar a comunicação entre a UAN e os estudantes. Tal aplicativo permitiria aos alunos acesso rápido e fácil às informações sobre seus tickets, cardápios e comunicados relacionados à UAN, simplificando o processo de comunicação e melhorando a experiência dos estudantes em relação aos serviços prestados pela instituição. Dessa forma, o aplicativo seria uma solução abrangente para os problemas abordados anteriormente. Além disso, o desenvolvimento do aplicativo permitiria a implementação de tecnologias inovadoras, proporcionando uma experiência mais moderna e interativa aos usuários.

2 REFERENCIAL TEÓRICO

Este capítulo contempla a fundamentação teórica para a implementação do trabalho, discorrendo sobre conceitos, ferramentas e tecnologias utilizadas.

2.1 Unidade de Alimentação e nutrição

As Unidades de Alimentação e Nutrição (UAN), são responsáveis pela distribuição de alimentação em larga escala para coletividades, tais como hospitais, escolas, empresas e presídios. Segundo Colares e Freitas (2007), o objetivo principal das UANs é administrar a produção de refeições nutricionalmente equilibradas, com bom padrão higiênico, que possam contribuir para manter ou recuperar a saúde de coletividades, além de auxiliar no desenvolvimento de hábitos alimentares saudáveis.

2.1.1 Fluxo de trabalho da UAN

As UANs passam por diversas etapas no fluxo de trabalho, que vão desde o planejamento das refeições até a distribuição aos consumidores.

A seguir, algumas etapas do fluxo de trabalho em uma UAN:

- a) **Planejamento:** Nessa etapa é definido o cardápio, levando em conta os aspectos nutricionais, orçamentários e a disponibilidade de alimentos.
- b) **Compra de alimentos:** Após a definição do cardápio, é realizada a compra dos alimentos necessários para a produção das refeições.
- c) **Recebimento e armazenamento:** Quando os alimentos chegam à UAN é necessário verificar se estão em conformidade com as especificações de qualidade e segurança alimentar.
- d) **Pré-preparo:** Antes da produção das refeições é necessário realizar algumas etapas de pré-preparo, tais como; lavagem, descascamento e corte dos alimentos.
- e) **Preparo:** O preparo das refeições envolve o uso de técnicas culinárias para transformar os alimentos em pratos saborosos e nutritivos.
- f) **Distribuição:** Após a produção das refeições, elas são distribuídas aos consumidores, respeitando as condições de temperatura e higiene.

- g) **Higienização:** Por fim, é necessário realizar a higienização dos equipamentos, utensílios e instalações da UAN, garantindo a segurança alimentar e a qualidade das refeições produzidas.

É importante ressaltar que o fluxo de trabalho das UANs pode variar de acordo com as especificidades de cada instituição. No entanto as etapas acima são fundamentais para garantir a produção de refeições saudáveis e seguras.

O uso das tecnologias nas unidades de alimentação e nutrição (UANs) é de extrema importância, apresentando qualidade e eficiência dos serviços prestados pelas UANs, pois pode ajudar a simplificar tarefas, reduzir custos, melhorar a gestão, facilitar comunicação entre o usuário e as UANs.

2.2 Computação Móvel

A computação móvel² é um termo popular e pode ser definida como o acesso à informação em qualquer lugar, a qualquer momento e com qualquer equipamento. Porém, temos também a computação pervasiva, que é o acesso à informação distribuída no ambiente de forma imperceptível e perceptível ao usuário. Explicando: é algo invisível a olho nu, mas sabe-se que a computação está presente no local.

A seguir serão discutidos a evolução da computação móvel, as principais características dos dispositivos móveis, como os sistemas operacionais Android, e como a computação móvel tem transformado a forma como as pessoas interagem com a tecnologia. Além disso, será abordado como a computação móvel está impactando a indústria e a sociedade como um todo.

2.2.1 *Evolução da computação móvel*

A evolução da computação móvel³ tem sido marcada por diversos avanços ao longo dos anos. No entanto, ela teve início no final dos anos 60 e início dos anos 70 com o surgimento das primeiras calculadoras eletrônicas de mão. A partir da década de 90, presenciamos uma inovação tecnológica significativa com o lançamento de novos dispositivos móveis, como

² <https://www.logicalminds.com.br/o-que-e-computacao-movel-e-como-voce-pode-fazer-uso-dessa-tecnologia/>

³ <https://nuvemandroid.wordpress.com/2013/10/15/historia-da-computacao-movel/>

telefones celulares e videogames portáteis aprimorados, os quais ganharam enorme popularidade pela população.

A partir de 2005, a computação móvel continuou impressionando seus usuários com o avanço dos dispositivos móveis cada vez mais avançados e poderosos apresentando novas funcionalidades como câmeras, acesso à internet, jogos e aplicativos. Hecke (2013) afirma que, para um aparelho ser definido como computador móvel, ele deve ter capacidade de processamento, conseguir se comunicar via rede sem fio, e ser transportado pelo usuário.

Vale ressaltar que a importância da computação móvel⁴ na atualidade é inegável, uma vez que os dispositivos móveis se tornaram essenciais na vida cotidiana das pessoas. Eles permitem a facilidade de acesso à internet, envio de e-mails, leitura de notícias entre outros. Isso torna a tecnologia móvel uma ferramenta crucial para a produtividade e conectividade das pessoas em diversas áreas da vida, desde a pessoal até a profissional. A tecnologia móvel também tem impactado significativamente na economia global⁵.

Segundo estudo da GSMA⁶, órgão que representa os interesses das operadoras de redes móveis em todo o mundo, em 2018 as tecnologias móveis geraram 3.9 trilhões de dólares em valor econômico agregado, o que representa cerca de 4,6% do PIB (Produto Interno Bruto) global. Além disso, há previsões de que a evolução da tecnologia 5G possa trazer mais 2.2 trilhões de dólares para a economia global nos próximos 15 anos. Tal impacto econômico tem impulsionado o desenvolvimento de novos mercados e criado oportunidades para empresas e empreendedores. Portanto, a evolução da computação móvel não apenas transformou a vida das pessoas, mas também tem sido um fator chave para o crescimento econômico global.

2.2.2 Surgimento dos sistemas operacionais Android e iOS

O Android⁷ é um sistema operacional desenvolvido para dispositivos móveis que surgiu em 2003, criado pelos empresários Andy Rubin, Rich Miner, Nick Sears e Chris White na empresa Android Inc. Inicialmente, o objetivo geral do Android era ser um sistema operacional para câmeras digitais, no entanto, logo foi observado que esse nicho não tinha potencial, o que influenciou a empresa a mudar o seu foco para celulares.

⁴ <https://consultech.srv.br/a-computacao-movel-no-dia-a-dia/>

⁵ <https://celere-ce.com.br/tecnologia/tecnologia-movel-engenharia-e-construcao/>

⁶ <https://www.gsma.com/>

⁷ <https://tecnoblog.net/responde/o-que-e-o-android-entenda-a-diferenca-para-o-ios-do-iphone/>

Em 2005, a Google comprou a Android Inc com muitos planos de mudanças, começando a trabalhar no desenvolvimento do sistema operacional para smartphones. Em 2008, a empresa lançou o primeiro dispositivo com o sistema operacional Android. Desde então, o Android tem atraído a atenção de milhares de pessoas e dispositivos por conseguir atender as empresas que buscam plataformas ágeis e modernas para o desenvolvimento de aplicações para auxiliar seus negócios e lucros, bem como os usuários que procuram dispositivos com visual elegante e moderno, de fácil navegação e uma infinidade de recursos (LECHETA, 2013).

O IOS é um sistema operacional móvel desenvolvido pela Apple, que foi lançado em 2007. Naquele ano, Steve Jobs apresentou ao mundo o primeiro dispositivo móvel com o sistema iOS, que foi desenvolvido especificamente para o iPhone, mas também é utilizado no iPod, iPad e Apple TV. Uma de suas principais características é não permitir que o sistema seja executado em hardware de terceiros, ou seja, ele é encontrado somente em aparelhos da própria marca (SARTORELI; KUCHAUSKI, 2013).

O iOS, como sistema operacional dos dispositivos móveis da Apple, desempenha uma função essencial no gerenciamento dos comandos do usuário e no funcionamento dos diversos aplicativos e processos do sistema. É importante destacar que outros sistemas operacionais, como o Android, Windows e Linux, também possuem essa mesma função de atuar como intermediário entre o usuário e o sistema. Portanto, compreender a importância do iOS é fundamental para usuários que desejam desfrutar de todas as funcionalidades de seus dispositivos móveis, bem como para desenvolvedores que buscam aprimorar suas aplicações para essa plataforma. Dessa forma, a utilização correta e eficiente do iOS contribui para a melhoria da experiência do usuário e para a evolução tecnológica como um todo (LARROSSA, 2019).

2.3 Desenvolvimento de Softwares para Dispositivos Móveis

O desenvolvimento de softwares para dispositivos móveis consiste na criação de aplicativos para smartphones, tablets e outros dispositivos móveis, visando atender necessidades específicas dos usuários e prover funcionalidades adequadas a esses dispositivos. Esse processo segue padrões definidos, que abrangem diversas etapas, tais como a concepção, design, desenvolvimento, teste e implantação. Outro aspecto importante é a escolha das ferramentas e tecnologias adequadas para o desenvolvimento do aplicativo.

Os aplicativos móveis podem ser criados para uma variedade de finalidades, desde jogos e entretenimento até aplicativos empresariais e de produtividade. Alguns aplicativos móveis são gratuitos, enquanto outros são pagos ou oferecem compras dentro do aplicativo para desbloquear recursos adicionais. Os desenvolvedores de aplicativos móveis também precisam estar cientes de questões de segurança, como proteção de dados do usuário e vulnerabilidades de segurança. Eles também precisam estar atualizados sobre as últimas tendências e tecnologias em desenvolvimento de aplicativos móveis para garantir que seus aplicativos sejam inovadores e relevantes para os usuários.

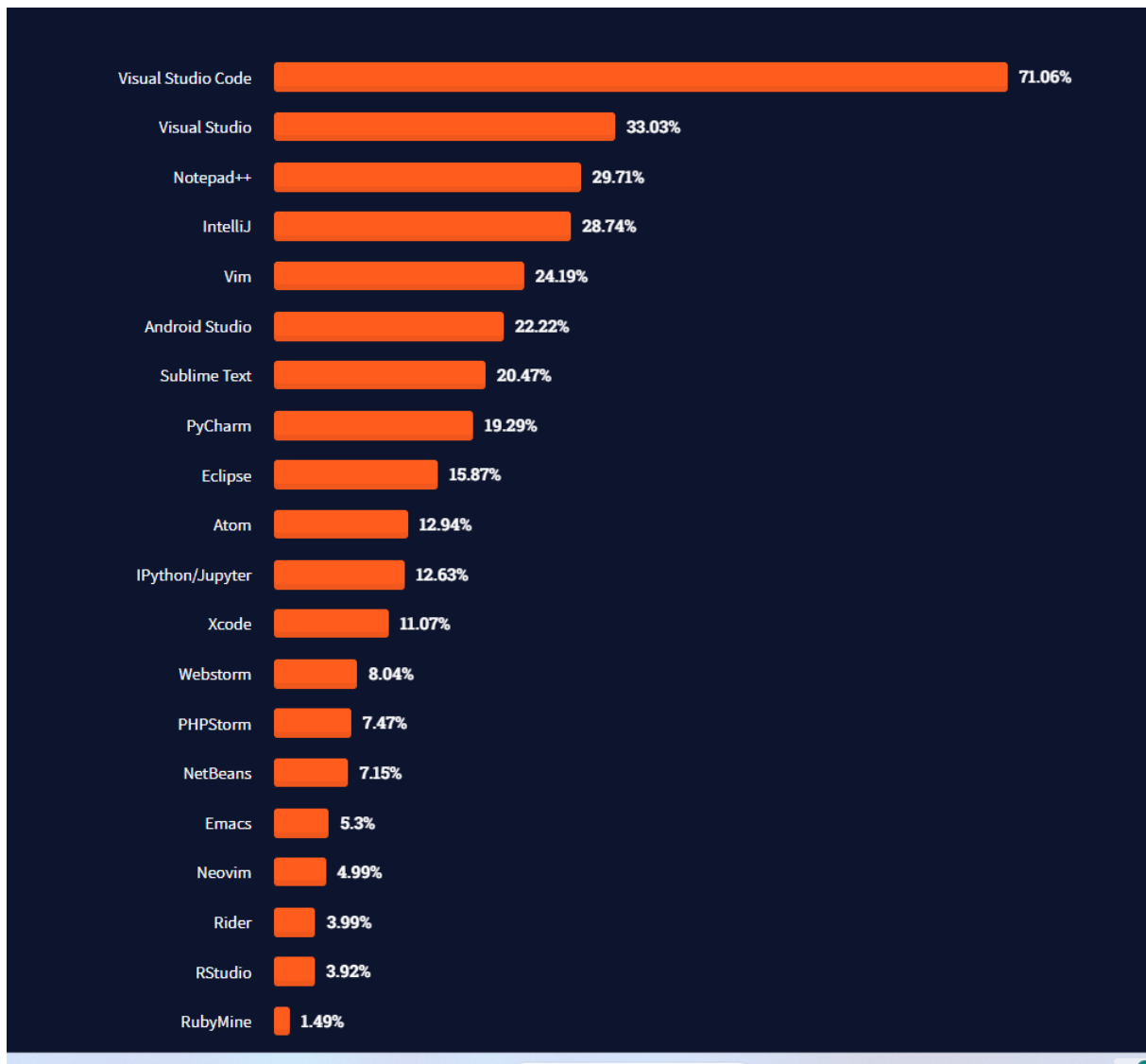
2.3.1 *Ambiente de desenvolvimento integrado*

O ambiente de desenvolvimento integrado (IDE) é um software com o objetivo em ajudar os desenvolvedores a escrever, testar e depurar programas de computadores. Ele geralmente inclui um editor de texto com recursos avançados de edição como realce de sintaxe, preenchimento automático de código, refatoração de código e integração com ferramentas de depuração. Os IDEs são usados para desenvolver uma ampla variedade de aplicativos, desde de sistemas desktop até sistemas móveis e web. Alguns geralmente são projetados para serem usados com uma linguagem de programação específica, mas a maioria dos IDEs suportam várias linguagens de programação. (DEITEL; DEITEL; DEITEL, 2015)

Existem diversos tipos de IDEs disponíveis, cada um com suas próprias vantagens e desvantagens, vai depender muito do projeto ou a linguagem que vai ser desenvolvido para a escolha do editor que melhor atende. O Stack Overflow⁸ criou uma pesquisa para saber quais editores de código são mais utilizados. A Figura 1 apresenta um resultado da pesquisa acerca dos editores de códigos mais usados.

⁸ <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-new-collab-tools>

Figura 1 - Editores de códigos mais usados



Fonte: (StackOverflow, 2021)

2.3.2 Emuladores de dispositivos móveis

Os emuladores⁹ de dispositivos móveis são softwares que permitem simular o funcionamento de um sistema ou dispositivo em um ambiente de computador. Eles têm uma ampla variedade de aplicações, mas são particularmente úteis para desenvolvedores de software que precisa testar e depurar aplicativos em diferentes dispositivos e sistemas operacionais. Neste contexto, os emuladores de dispositivos móveis são uma ferramenta importante para os

⁹ <https://finalverse.com.br/o-que-sao-emuladores-definicao-funcionamento-tipos-e-exemplos/>

desenvolvedores de aplicativos móveis, pois permitem simular a operação de dispositivos móveis em um ambiente de computador.

O principal objetivo dos emuladores de dispositivos móveis é permitir que os desenvolvedores de aplicativos móveis testem e depurem seus aplicativos em diferentes dispositivos móveis e sistemas operacionais sem a necessidade de possuir fisicamente cada um desses dispositivos, como por exemplo prover os componentes típicos de um aparelho celular, como botões de navegação, um teclado, e até mesmo uma tela que pode ser clicada com o mouse. No entanto, há algumas limitações, como não ter suporte a conexões USB, à câmera, a fones de ouvido, ao estado de carga da bateria, a Bluetooth e, é claro, a realização de chamadas telefônicas (ACKER; WEBER; CECHIN, 2010).

Em resumo os emuladores são uma ferramenta importante para desenvolvedores de aplicativos móveis que desejam testar e depurar seus aplicativos em diferentes dispositivos móveis e sistemas operacionais. Eles permitem simular a operação de dispositivos móveis em um ambiente de computador, tornando mais fácil e acessível a depuração e teste de aplicativos. Existem vários emuladores de dispositivos móveis disponíveis no mercado, cada um com suas próprias vantagens e desvantagens, e a escolha do melhor emulador dependerá das necessidades específicas do projeto e da plataforma de desenvolvimento. Em resumo, os emuladores de dispositivos móveis são uma ferramenta essencial para o desenvolvimento de aplicativos móveis e devem ser considerados como parte do processo de desenvolvimento.

2.3.3 Tipos de aplicativos móveis

Aplicativos móveis, também conhecidos como apps, são softwares desenvolvidos especificamente para serem executados e acessados por dispositivos móveis, como celulares, smartphones e tablets. Esses aplicativos desempenham um papel significativo em nossa vida cotidiana, fornecendo uma ampla gama de funcionalidades e serviços, com o objetivo principal de facilitar a realização de atividades práticas pelos usuários. Essas atividades podem se manifestar em diferentes formas, abrangendo aplicativos de entretenimento, educação, música, saúde, entre outros exemplos (ENGELMANN; SÁPIRAS, 2016).

A seguir veja alguns tipos de aplicativos moveis:

- a) **Aplicativos de redes sociais:** São aplicativos como Facebook, Instagram, Twitter, LinkedIn, entre outros, que permitem aos usuários se conectar e

interagir com outras pessoas, compartilhar conteúdo, postar fotos e vídeos, e acompanhar as últimas novidades e tendências.

- b) **Aplicativos de mensagens instantâneas:** Exemplos famosos incluem WhatsApp, Messenger, Telegram e Snapchat. Esses aplicativos permitem que os usuários enviem mensagens de texto, áudio, imagens e vídeos instantaneamente, facilitando a comunicação em tempo real com amigos, familiares e colegas de trabalho.
- c) **Aplicativos de entretenimento:** Incluem aplicativos de streaming de música, como Spotify e Apple Music, aplicativos de streaming de vídeo, como Netflix e YouTube, e jogos móveis populares. Esses aplicativos oferecem uma ampla variedade de conteúdo para entretenimento pessoal.
- d) **Aplicativos de produtividade:** São aplicativos projetados para ajudar as pessoas a gerenciar tarefas, criar e editar documentos, fazer anotações, planejar agendas e melhorar a eficiência no trabalho. Exemplos incluem o Microsoft Office, Evernote e Trello.
- e) **Aplicativos de comércio eletrônico:** Aplicativos como Amazon, eBay e AliExpress permitem que os usuários comprem produtos e serviços diretamente de seus dispositivos móveis. Esses aplicativos oferecem uma experiência de compra conveniente, com opções de pesquisa, pagamento seguro e rastreamento de pedidos.

Em conclusão, fica evidente que os aplicativos móveis desempenharam um papel revolucionário ao transformar a maneira como vivemos e interagimos com a tecnologia. Eles nos permitem acessar informações instantaneamente, comunicar-nos com facilidade e realizar tarefas cotidianas de maneira eficiente. Os diferentes tipos de aplicativos móveis atendem a uma ampla gama de necessidades e interesses, proporcionando entretenimento, conectividade social, produtividade e conveniência comercial. Com o avanço contínuo da tecnologia móvel, podemos esperar uma crescente diversidade de aplicativos e recursos inovadores que aprimorarão ainda mais nossa experiência móvel. Os aplicativos móveis se tornaram parte essencial da nossa vida digital, transformando a maneira como nos relacionamos com o mundo ao nosso redor (LIMA, 2017).

2.4 Desenvolvimento nativo

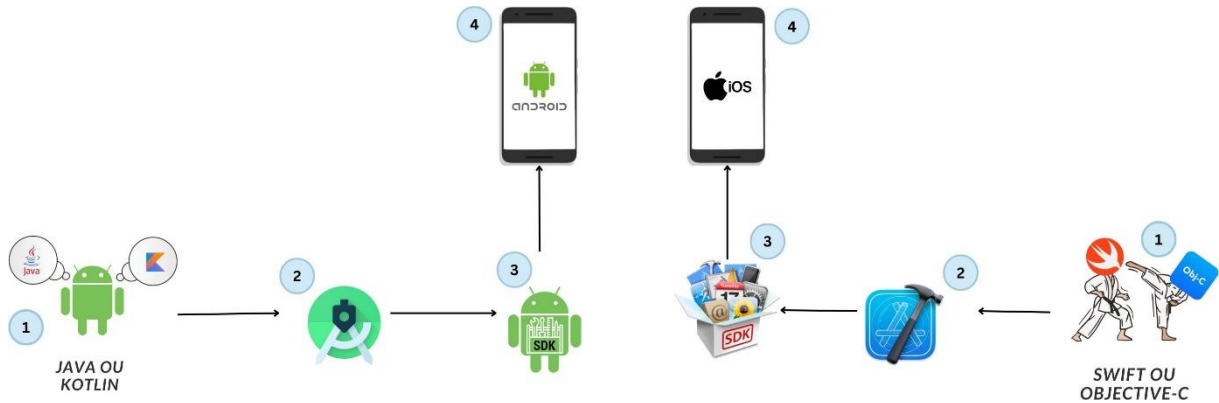
O desenvolvimento nativo se baseia principalmente na criação de aplicativos específicos para cada plataforma móvel, como por exemplo, iOS e Android, utilizando linguagens de programação e ferramentas de desenvolvimento nativas de cada plataforma. Isso permite que o aplicativo tenha acesso completo a funcionalidades e recursos do dispositivo. De acordo com Silva e Santos (2014), "o desenvolvimento de um aplicativo nativo para aparelhos celulares exige conhecimentos específicos a respeito das tecnologias utilizadas pela plataforma na qual se deseja executá-lo".

A abordagem de desenvolvimento nativo de aplicativos móveis possui vantagens e desvantagens que precisam ser cuidadosamente avaliadas antes de serem aplicadas em um projeto específico. A tomada de decisão envolve, entre outros fatores, a identificação das necessidades específicas do projeto e a escolha da linguagem de programação adequada. Não existe uma receita sobre quando desenvolver um aplicativo (app) nativo ou um app multiplataformas. Deve-se fazer uma boa avaliação do cenário, como por exemplo qual público deseja-se atingir, ou o quanto se tem para investir (GUEDES, 2017).

Algumas das vantagens são: melhor desempenho e integração com os recursos do dispositivo, além de permitir uma maior personalização da interface do usuário. No entanto, também apresenta algumas desvantagens, como a necessidade de desenvolver e manter diferentes versões do aplicativo para cada plataforma, o que aumenta o tempo e o custo do desenvolvimento.

No que se refere às linguagens de programação, cada plataforma possui sua própria linguagem nativa. Para o desenvolvimento de aplicativos iOS, a linguagem utilizada é o Swift ou o Objective-C, enquanto que para o desenvolvimento de aplicativos Android, a linguagem nativa é o Java ou o Kotlin. Cada uma dessas linguagens possui suas vantagens e desvantagens em termos de desempenho, facilidade de uso e comunidade de desenvolvedores. Por exemplo, o Swift é uma linguagem mais moderna e intuitiva, enquanto o Java possui uma grande base de usuários e é amplamente suportado por ferramentas e bibliotecas. Já o Kotlin é considerado uma evolução do Java, com recursos mais avançados e menos verbosidade. A figura 2 representa os fluxos de desenvolvimento nativo para Android e iOS.

Figura 2 - Desenvolvimento nativo



Fonte: Elaborado pelos autores, 2023

Em resumo, o desenvolvimento nativo de aplicativos móveis oferece vantagens e desvantagens, e a escolha da linguagem de programação depende das necessidades e objetivos do projeto. É importante avaliar cuidadosamente as opções disponíveis e escolher a abordagem que melhor atenda às demandas do aplicativo e do público-alvo.

2.4.1 Vantagens e desvantagens do desenvolvimento nativo

A elaboração de aplicativos nativos apresenta tanto vantagens quanto desvantagens. Uma vantagem é que, como esses aplicativos são desenvolvidos usando as soluções padrão da plataforma, os desenvolvedores têm acesso total e mais fácil aos recursos do dispositivo; como todos os sensores do dispositivo, o catálogo de endereços do usuário e qualquer que seja a última e melhor tecnologia que o telefone oferece (GRIFFITH, 2017).

Outra vantagem do desenvolvimento nativo de aplicativos móveis é o seu desempenho superior em comparação a outras abordagens, como aquelas baseadas em tecnologias web ou híbridas. A literatura especializada aponta que os aplicativos nativos apresentam melhor desempenho devido ao fato de que seu código é desenvolvido especificamente para cada plataforma e está mais próximo do "metal". Além disso, eles permitem acesso completo aos controles e layouts nativos da interface do usuário, o que proporciona uma melhor experiência para o usuário final (GRIFFITH, 2017). É importante destacar que o desempenho dos aplicativos nativos pode influenciar diretamente na satisfação dos usuários, tornando-se um fator crucial no sucesso de um aplicativo móvel.

De acordo com um artigo publicado pela empresa UDS (2022), o desenvolvimento de aplicativos nativos também pode apresentar vantagens em relação à interface do usuário (UI) e à experiência do usuário (UX). Isso se deve, em grande parte à necessidade de seguir as

diretrizes específicas de cada plataforma, como a de Design Humano da Apple. Essas diretrizes são criadas para garantir que os aplicativos atendam a determinados padrões de qualidade, usabilidade e acessibilidade, o que pode resultar em uma interface mais intuitiva e uma experiência mais agradável para o usuário final.

Os aplicativos nativos são frequentemente considerados uma opção mais segura do que outros tipos de aplicativos móveis devido à sua capacidade de aproveitar ao máximo as medidas de segurança integradas das plataformas móveis. Por exemplo, tanto o iOS quanto o Android oferecem recursos de segurança robustos, como autenticação de usuário, gerenciamento de permissões e criptografia de dados, que podem ser facilmente incorporados em aplicativos nativos. Isso ajuda a proteger os dados do usuário e impede que invasores acessem informações confidenciais armazenadas no aplicativo. Segundo a empresa ImagineDone (2022) "o app nativo possui um desenvolvimento mais robusto e não tem a necessidade de intermediários, já que se comunica de forma direta com o aparelho. Isso torna a aplicação mais segura e estável".

Embora o desenvolvimento nativo tenha muitas vantagens em relação a outras abordagens, não podemos ignorar suas desvantagens. Um exemplo disso é que, um aplicativo nativo consome bem mais recursos para ser desenvolvido. Como os aplicativos nativos só funcionam na plataforma para a qual foram projetados, portar um aplicativo nativo para diferentes plataformas consome mais recursos para o desenvolvimento de código e tempo de teste. Em alguns casos, pode ser necessário um redesign completo do aplicativo, uma vez que diferentes sistemas hospedeiros fornecem diferentes restrições ao acesso de funções de hardware e software (SELVARAJAH et al., 2013).

Além disso, de acordo com Sousa, Brito e Reis (2014), os aplicativos nativos têm uma curva de aprendizagem muito maior em comparação com outras abordagens. Isso ocorre não apenas devido à quantidade de conhecimento necessário, mas também pela extensão do SDK (conjunto de ferramentas de criação específicas da plataforma para desenvolvedores) e da API (Interface de Programação de Aplicações) do Android, que exigem que os desenvolvedores utilizem-se dos conhecimentos de Java.

Outra desvantagem comum no desenvolvimento de aplicativos nativos é o alto custo que o mesmo apresenta para ser desenvolvido e mantido. De acordo com o artigo publicado pela empresa UDS (2022), os custos para desenvolver aplicativos nativos são significativamente maiores do que outras opções, o que pode limitar o acesso a essa tecnologia para empresas menores ou com orçamentos mais limitados. Isso se deve pela necessidade de

suportar duas plataformas simultaneamente (iOS e Android) o que pode exigir a contratação de vários profissionais, aumentando ainda mais os custos envolvidos.

Por fim, é importante mencionar mais uma desvantagem do desenvolvimento nativo, que está relacionada à demora e complexidade em realizar manutenções. Regularmente faz-se necessário atualizar o aplicativo para mantê-lo compatível com as últimas versões das plataformas móveis, que podem mudar significativamente ao longo do tempo. Esse trabalho de atualização pode ser árduo. Segundo Dharmwan (2021) no desenvolvimento nativo, muito tempo é consumido em esforços de desenvolvimento separados para cada plataforma, tornando um processo lento e por vezes complicado durante cada atualização de recurso.

A Tabela 1 apresenta as principais vantagens e desvantagens do desenvolvimento nativo de maneira sintetizada.

Tabela 1 - vantagens e Desvantagens do desenvolvimento nativo

Vantagens	Desvantagens
Acesso total e mais fácil aos recursos do dispositivo.	Consome mais recursos para o desenvolvimento de código e tempo de teste.
Desempenho superior aos aplicativos que utilizam outras tecnologias.	Curva de aprendizado maior se comparado com outras abordagens.
Melhor interface do usuário(UI) experiência do usuário(UX).	Custos para desenvolvimento significativamente maiores.
Melhor segurança graças às medidas integradas das plataformas.	Manutenções mais demoradas e complexas.

Fonte: Elaborado pelos autores, 2023

2.4.2 Linguagens de programação para desenvolvimento nativo

Conforme abordado na seção anterior, o desenvolvimento nativo demanda a criação de um aplicativo específico para cada plataforma (iOS ou Android), o que levanta a questão sobre qual linguagem de programação é utilizada para o desenvolvimento em cada uma dessas plataformas. Nos próximos parágrafos deste trabalho serão discutidas as linguagens de programação mais utilizadas para o desenvolvimento nativo em cada plataforma, assim como as suas vantagens e desvantagens.

Segundo Coursera (2022) "é preciso entender que ao escrever um aplicativo de software, você deve convertê-lo em instruções que um computador possa entender". Para fazer isso, é preciso escrever códigos em uma linguagem que a máquina consiga interpretar. Essas linguagens são conhecidas como linguagens de programação. Existem diversas linguagens de programação, cada uma com sua própria sintaxe e finalidade.

No contexto do desenvolvimento nativo de aplicativos para a plataforma Android, as linguagens de programação mais amplamente utilizadas são o Java e Kotlin. Embora existam outras linguagens disponíveis, estas duas têm se destacado devido às suas características e suporte fornecido pela Google, empresa responsável pelo sistema operacional Android. De acordo com Manchanda (2022), "embora a indústria continue a mudar à medida que novas tecnologias surgem e as tendências mudam, o desenvolvimento de aplicativos Android envolve principalmente o conhecimento de duas linguagens principais (Java e Kotlin)".

Java é uma das duas linguagens de programação oficiais para o desenvolvimento de aplicativos Android, tendo sido a principal até o surgimento do Kotlin. Apesar da existência de outras opções, o Java continua a ser uma das linguagens mais amplamente utilizadas para esse propósito, devido à sua estabilidade, segurança, suporte a múltiplo encadeamento e portabilidade, bem como à sua capacidade de lidar com arquiteturas complexas, o que é altamente valorizado por muitos desenvolvedores. Entretanto, vale ressaltar que o Java também apresenta algumas desvantagens relacionadas ao desempenho (MANCHANDA, 2022).

O Kotlin, por sua vez, surgiu como uma alternativa moderna ao Java para o desenvolvimento de aplicativos Android. Ele foi oficialmente endossado pela Google como uma linguagem de programação preferencial para o desenvolvimento Android, e muitos desenvolvedores têm migrado para o Kotlin devido às suas vantagens em relação ao Java. O Kotlin oferece uma sintaxe mais concisa, segurança nula (null safety) integrada, suporte a programação funcional e uma integração perfeita com o código Java existente. Essas características ajudam a melhorar a produtividade dos desenvolvedores e reduzir a ocorrência de erros.

Em relação à plataforma iOS, a linguagem de programação principal para o desenvolvimento nativo de aplicativos é o Swift, uma linguagem de programação compilada para os sistemas operacionais iOS, macOS, watchOS, tvOS e Linux. Introduzido pela Apple em 2014, o Swift substituiu a linguagem Objective-C como a linguagem preferencial para desenvolvimento iOS. O Swift foi projetado para ser mais moderno, seguro e fácil de usar do que o Objective-C, oferecendo uma sintaxe clara e concisa, inferência de tipos e recursos avançados de segurança, como segurança de memória e gerenciamento de opções. Esses

recursos tornam o Swift mais eficiente e menos propenso a erros em comparação com o Objective-C.

A linguagem Objective-C era a linguagem de programação principal usada para desenvolver aplicativos para sistemas operacionais iOS e macOS. Ela era a linguagem de programação padrão para o desenvolvimento de aplicativos para dispositivos da Apple e era amplamente utilizada na comunidade de desenvolvimento de aplicativos móveis. De acordo com o ItForum (2021), o Objective-C era a estrela em ascensão no índice Tiobe, sendo vencedora do prêmio de linguagem do ano em 2011 e 2012. Sua fama, no entanto, acabou em 2014, quando a Apple anunciou que havia uma nova linguagem chamada Swift.

Embora o Swift seja a linguagem principal para desenvolvimento iOS, ainda é possível usar o Objective-C, especialmente ao trabalhar com código legado ou bibliotecas existentes. O Objective-C é uma linguagem de programação mais antiga, amplamente utilizada antes da introdução do Swift, e ainda é suportada pela Apple para garantir a compatibilidade com projetos existentes.

A Tabela 2 apresenta uma breve comparação entre Objective-C e Swift. A comunidade de desenvolvedores do Swift também contribui ativamente para o seu crescimento, criando bibliotecas e frameworks que estendem as funcionalidades da linguagem e fornecem soluções para diversos domínios de aplicação. A disponibilidade dessas bibliotecas adicionais facilita o desenvolvimento de aplicativos iOS mais complexos e ricos em recursos.

Tabela 2 - Comparação entre objective-C e Swift

	Swift	Objective-C
Primeira aparição	2014	1984
Paradigma	Multiparadigma	Imperativo, Orientado a Objetos
SO	iOS, iPadOS, macOS, watchOS, tvOS, Ubuntu, CentOS, Amazon Linux, Windows	Plataforma cruzada (mas raramente vista em plataformas que não sejam da Apple)
Influenciado por	Objective-C, Python, Go, JavaScript e inúmeros outros	Smalltalk, C
Tipo de sistema	Estático	Dinâmico
Código fonte	Código aberto	
	Proprietário	É ativamente
desenvolvido?	Sim	Não

Fonte: (Itmagination, 2023)

A Tabela 3 apresenta o ranking das dez linguagens de programação mais populares, segundo pesquisa do Stack Overflow, feita em 2021.

Tabela 3 - Popularidade das Linguagens de Programação

Posição	Linguagem de Programação	Popularidade
1 ^o	Python	19.04%
2 ^o	Typescript	15.29%
3 ^o	Javascript	14.59%
4 ^o	Go	14.54%
5 ^o	Rust	14.09%
6 ^o	Node.js	11.9%
7 ^o	C++	8.8%
8 ^o	Kotlin	8.57%
9 ^o	Java	6.79%
10 ^o	C#	6.45%

Fonte: (StackOverflow, 2021)(adaptado)

Em resumo, as linguagens de programação mais comuns para o desenvolvimento nativo em cada plataforma são:

- a) **Android:** Java e Kotlin (sendo Kotlin a opção mais moderna e recomendada).
- b) **iOS:** Swift (com suporte adicional para Objective-C).

É importante notar que essas são as linguagens de programação mais amplamente utilizadas, mas existem outras opções disponíveis para o desenvolvimento nativo em ambas as plataformas. A escolha da linguagem depende das preferências do desenvolvedor, requisitos do projeto e suporte da plataforma.

No desenvolvimento mobile, o TypeScript e o JavaScript desempenham papéis importantes, especialmente em aplicativos desenvolvidos usando tecnologias híbridas ou multiplataforma. Clark (2016) diz que "existem muitas linguagens diferentes que são usadas atualmente para o desenvolvimento Android e iOS. Enquanto alguns tornam as coisas muito fáceis para os desenvolvedores, outros trazem a maioria dos recursos e habilidades para os aplicativos". A partir da próxima seção iremos explorar essa conexão.

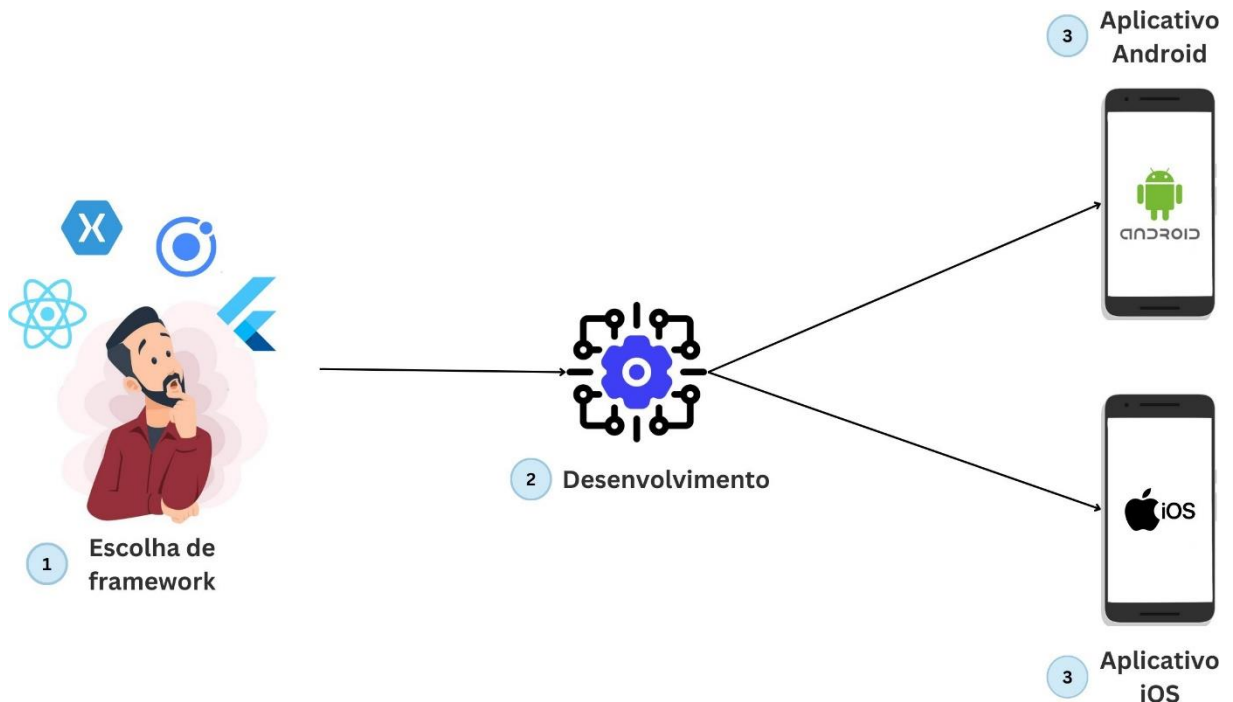
2.5 Desenvolvimento multiplataforma

Desenvolver um aplicativo que atenda as principais plataformas do mercado pode se tornar uma dor de cabeça devido ao alto custo de mão de obra especializada (programadores) com competência técnica em cada uma das linguagens de programação necessárias. Outro aspecto relevante é o tempo necessário para deixar o aplicativo funcionando sem nenhum problema (bug). A fim de solucionar essa questão, foram implementados os aplicativos multiplataformas. (FORTE et al., 2017).

O desenvolvimento móvel multiplataforma é uma estratégia de desenvolvimento de software que permite criar aplicativos que possam ser usados em mais de uma plataforma, como, Android e iOS. Essa abordagem oferece aos desenvolvedores a capacidade de usar um único código-fonte para criar aplicativos que sejam compatíveis com várias plataformas móveis, em vez de desenvolver aplicativos separados para cada sistema operacional. É nesse ponto que se evidencia a distinção entre o desenvolvimento móvel multiplataforma e o desenvolvimento de aplicativos nativos, visto que este último envolve a criação de um aplicativo para uma plataforma específica, ao passo que a primeira abordagem permite que um único conjunto de código seja usado para diversas plataformas móveis.

A Figura 3 representa os fluxos de desenvolvimento multiplataformas para Android e iOS.

Figura 3 - Desenvolvimento multiplataformas



Fonte: Elaborado pelos autores, 2023

2.5.1 *Vantagens e desvantagens do desenvolvimento multiplataforma*

Com a crescente variedade de dispositivos e sistemas operacionais disponíveis no mercado, desenvolver aplicativos que possam ser executados em diferentes plataformas pode oferecer diversas vantagens. Entretanto, assim como em qualquer outra tecnologia, o desenvolvimento multiplataforma também apresenta suas desvantagens. Neste tópico, será explorado tanto as vantagens quanto as desvantagens do desenvolvimento multiplataforma, a fim de entender melhor essa abordagem e ajudar na tomada de decisões em projetos de desenvolvimento de software.

O paradigma de desenvolvimento multiplataforma tem diversas vantagens potenciais. Por exemplo, ela permite que os usuários acessem o mesmo aplicativo em diferentes plataformas, possibilitando que eles possam comparar e escolher a melhor opção para suas necessidades. Além disso, essa abordagem de desenvolvimento também reduz os custos de desenvolvimento e manutenção, uma vez que permite que os desenvolvedores criem um único aplicativo que possa ser executado em várias plataformas. Outra coisa é que com essa abordagem, os desenvolvedores podem reaproveitar o código existente e economizar tempo e recursos no desenvolvimento de aplicativos para diferentes plataformas. Como os aplicativos podem ser lançados em várias plataformas diferentes, os desenvolvedores podem promovê-los e vendê-los em diferentes mercados de distribuição, o que pode aumentar sua exposição e alcance, o que é outra vantagem dessa abordagem (CORRAL; JANES; REMENCIUS, 2012).

Além das vantagens, o desenvolvimento multiplataformas também apresenta potenciais desvantagens. Dentre essas desvantagens é que estes aplicativos não oferecem uma experiência de usuário totalmente nativa ou não exploram todas as capacidades do dispositivo. Uma das desvantagens também do desenvolvimento multiplataforma é que as ferramentas de desenvolvimento podem ter acesso limitado a algumas características do dispositivo móvel, o que pode limitar a funcionalidade dos aplicativos. Além disso, a implantação em diferentes plataformas pode exigir solução de problemas e personalização específicas da plataforma, o que pode ser mais demorado e difícil do que a implantação em uma única plataforma (CORRAL; JANES; REMENCIUS, 2012).

2.5.2 *JavaScript para desenvolvimento multiplataforma (ES6)*

O JavaScript ES6, como dito por Paltoglou et al. (2021), é uma versão atualizada e aprimorada da linguagem de programação JavaScript, desempenhando um papel fundamental no desenvolvimento multiplataforma. Com sua sintaxe moderna e recursos avançados, o ES6 oferece aos desenvolvedores uma ampla gama de funcionalidades para a criação de aplicativos multiplataforma de alta qualidade. Uma das principais vantagens do JavaScript ES6 é o suporte ao desenvolvimento de aplicativos complexos por meio do encapsulamento de variáveis de nível superior e declarações de função em um arquivo JavaScript. Essas características permitem a declaração de variáveis de escopo local e constante, respectivamente, melhorando a legibilidade e a segurança do código.

Além disso, o JavaScript ES6 representa um avanço significativo em relação à versão anterior da linguagem de programação. Com recursos aprimorados e uma sintaxe mais moderna, o ES6 oferece aos desenvolvedores as ferramentas necessárias para criar aplicativos multiplataforma sofisticados e de alta qualidade. Com novas funcionalidades avançadas, contribui para a organização do código, melhorando a legibilidade e a segurança. Ao adotar o JavaScript ES6, os desenvolvedores têm a oportunidade de aproveitar ao máximo as funcionalidades avançadas da linguagem, facilitando o desenvolvimento de aplicativos complexos. Portanto, o JavaScript ES6 desempenha um papel essencial no cenário atual de desenvolvimento de software, oferecendo uma base sólida para a criação de soluções multiplataforma eficientes e robustas (PALTOGLOU et al., 2021).

2.5.3 Frameworks multiplataforma

Os frameworks são tecnologias desenvolvidas com o objetivo de tornar mais fácil a criação de aplicativos para dispositivos móveis e, ao mesmo tempo, buscar uniformizar as experiências em diferentes dispositivos. Ao utilizar esses frameworks, é possível portar um mesmo código para diversas plataformas, o que resulta em uma maneira mais equilibrada e acelerada de desenvolver aplicativos para dispositivos móveis. Isso porque, ao invés de se ter que criar diferentes versões do mesmo aplicativo para cada sistema operacional, por exemplo, com o uso dos frameworks mobile, é possível construir uma única base de código que possa ser utilizada em diversos ambiente, o que gera uma economia de tempo e esforço considerável (CERQUEIRA; BITTENCOURT, 2016).

Existem diversos frameworks multiplataformas disponíveis no mercado atualmente, como por exemplo o Ionic, que é baseado em tecnologias web como HTML, CSS

e JavaScript, o Flutter, que utiliza a linguagem Dart e é conhecido por sua rapidez e performance, o Xamarin, que permite o desenvolvimento de aplicativos em C#, e o React Native, que é baseado no framework React e é conhecido por sua facilidade de uso e performance. Além desses, existem muitos outros frameworks disponíveis, cada um com suas próprias características e vantagens, o que oferece uma variedade de opções para desenvolvedores que desejam criar aplicativos móveis multiplataforma.

2.5.3.1 React Native

O React Native¹⁰ é um framework multiplataforma para desenvolvimento de aplicativos móveis que foi criado pelo Facebook em 2015. Ele permite que os desenvolvedores criem aplicativos para iOS e Android usando a linguagem JavaScript e a biblioteca React, que é muito popular no desenvolvimento web. O React Native permite que os desenvolvedores reutilizem grande parte do código entre as duas plataformas, economizando tempo e esforço no desenvolvimento.

A utilização do React Native no desenvolvimento de aplicações móveis não somente possibilita que essas sejam compatíveis com múltiplas plataformas, mas também permite que os recursos nativos dos dispositivos sejam acessados diretamente. Isso é possível graças à adoção de tecnologias Web, como o Javascript, no processo de desenvolvimento. Com essa abordagem, não é necessário contar com um intermediário para acessar os componentes nativos dos dispositivos, especialmente aqueles relacionados à interface com o usuário, que geralmente consomem uma parcela significativa dos recursos utilizados pelas aplicações. De maneira resumida, o uso do React Native proporciona uma experiência de desenvolvimento mais simplificada e uma maior capacidade de integração com os dispositivos móveis em que as aplicações são executadas (TOSTES; COSTA, 2020).

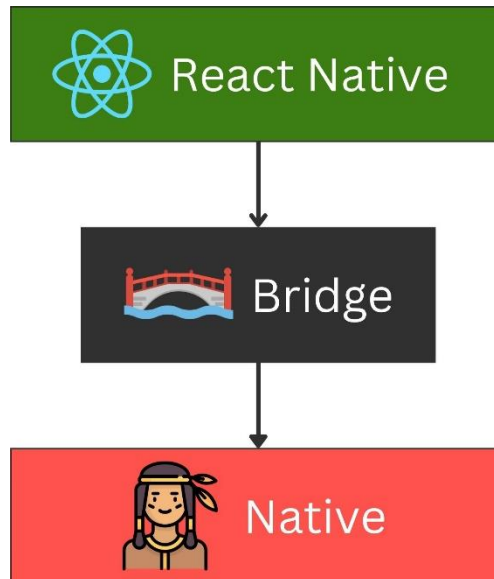
A estrutura da aplicação React Native se baseia em três elementos essenciais, isto é, a máquina virtual JavaScript, a ponte React Native e os módulos nativos, formando uma arquitetura. O código da aplicação é executado na máquina virtual JS juntamente com as bibliotecas de terceiros aplicadas. Os chamados dos módulos nativos são direcionados através da ponte do React Native, interligando as APIs nativas e as bibliotecas de terceiros, e os resultados são repassados pela ponte, conforme necessário. Esse arranjo possibilita a utilização

¹⁰ <https://reactnative.dev/>

do JavaScript na criação do aplicativo, mantendo-se a capacidade de empregar os componentes e recursos nativos da interface do usuário das plataformas. (KUITUNEN, 2019).

A Figura 4 representa a arquitetura do React Native.

Figura 4 - Arquitetura React Native



Fonte: Elaborado pelos autores, 2023

A imagem acima apresenta as três principais camadas de um aplicativo móvel desenvolvido com React Native. A primeira camada é o React Native em si, que é uma biblioteca JavaScript que permite o desenvolvimento de aplicativos móveis para iOS e Android. A segunda camada é a bridge, que atua como uma ponte entre o código JavaScript do React Native e as APIs nativas do sistema operacional. Por fim, temos a camada nativa, que representa o sistema operacional do dispositivo móvel e é responsável por executar os comandos recebidos da bridge e interagir com o hardware do dispositivo. Juntas, essas três camadas permitem que os desenvolvedores criem aplicativos móveis poderosos e eficientes com o React Native.

2.5.3.2 O que é o Expo e como funciona?

O Expo é uma plataforma de desenvolvimento de aplicativos móveis para iOS e Android baseada em React Native. Ele fornece uma série de ferramentas e serviços para facilitar o desenvolvimento de aplicativos, como um ambiente de desenvolvimento integrado (IDE), um conjunto de bibliotecas JavaScript, APIs nativas e uma infraestrutura de hospedagem em nuvem. O Expo é uma opção popular para desenvolvedores que desejam criar aplicativos

móveis rapidamente, sem precisar se preocupar com detalhes de baixo nível do sistema operacional ou com a configuração de ambientes de desenvolvimento.

Com o objetivo de simplificar a criação e distribuição de software multiplataforma, o Expo é uma ferramenta construída em torno do React Native. Além de um ambiente de construção gerenciado e ferramentas para testes e depuração, ele oferece uma variedade de recursos que podem ser usados para desenvolver, construir e publicar aplicativos React Native. Entre eles, destaca-se a plataforma-neutral API que libera os desenvolvedores de terem que escrever código específico da plataforma para acessar funções nativas do dispositivo, como a câmera, GPS e notificações push. Além disso, o Expo também oferece uma série de recursos adicionais, como gerenciamento de ativos, sistema de notificações push, atualizações de aplicativos sem a necessidade de baixar novas versões da loja de aplicativos, gerenciamento de compras no aplicativo, análise de uso e monitoramento do aplicativo. Segundo a própria documentação do Expo, esses recursos adicionais não estão disponíveis no React Native puro (HUTRI, 2023).

O Expo é uma ótima escolha para projetos que exigem um desenvolvimento mais rápido e simples, sem a necessidade de conhecimento profundo em codificação nativa. É ideal para projetos menores e para desenvolvedores que desejam criar aplicativos rapidamente. No entanto, se o projeto requer recursos nativos complexos, ou se você precisa de um controle mais avançado sobre a compilação do aplicativo, é melhor usar o React Native puro ou outras ferramentas nativas. Portanto, é importante avaliar cuidadosamente as necessidades do projeto e escolher a estrutura que melhor atenda a essas necessidades.

Hutri (2023) conclui no final de sua pesquisa que o Expo é uma ferramenta de desenvolvimento de aplicativos móveis que oferece uma abordagem mais simples e rápida do que o React Native puro. Embora o React Native possa fornecer um conjunto mais flexível e poderoso de ferramentas para o desenvolvimento de aplicativos móveis, o Expo é mais adequado para projetos que exigem uma abordagem mais simples e direta, sem a necessidade de conhecimento aprofundado em codificação nativa. No entanto, aplicações que exigem implementações nativas altamente específicas ou complexas podem ser melhor desenvolvidas com o React Native puro ou com ferramentas nativas, como o Android Studio e o Xcode. Portanto, é importante avaliar cuidadosamente as necessidades de cada projeto para escolher a estrutura que melhor atenda a essas necessidades.

2.6 Integração de APIs no Desenvolvimento de Aplicativos

As Application Programming Interface (APIs) são ferramentas que facilitam a comunicação entre dois elementos de software por meio de um conjunto de especificações e protocolos. Elas permitem que desenvolvedores e empresas aproveitem as funcionalidades e os dados de um software por meio de uma interface bem definida. De acordo com o site TechTudo (2020), por meio de APIs, desenvolvedores podem criar novos softwares e aplicativos capazes de se comunicar com outras plataformas. Por exemplo: caso um desenvolvedor queira criar um aplicativo de fotos para Android, ele poderá ter acesso à câmera do celular através da API do sistema operacional, sem ter a necessidade de criar uma nova interface de câmera do zero.

A utilização de APIs facilita a integração entre sistemas, permitindo a troca de dados de forma padronizada e eficiente. Além disso, ao seguir os princípios da arquitetura REST, as APIs tornam-se mais flexíveis e escaláveis, possibilitando o desenvolvimento de aplicações mais robustas e interoperáveis. É importante ressaltar que o uso de APIs tem se tornado cada vez mais comum no desenvolvimento de software, tanto em aplicações web quanto em dispositivos móveis. Através das APIs, é possível realizar a comunicação entre diferentes sistemas e plataformas, facilitando a criação de soluções integradas e proporcionando uma melhor experiência para os usuários.

Vale ressaltar a importância da distinção entre APIs públicas e privadas. Conforme Iugu (2021) relata, uma API é considerada pública quando pode ser utilizada por qualquer pessoa ou empresa interessada, caracterizando-se pela sua acessibilidade irrestrita devido à sua disponibilização pública. Geralmente, empresas cujo modelo de negócio envolve o compartilhamento de dados ou a oferta de serviços computacionais aos consumidores ou público em geral disponibilizam APIs públicas. Por outro lado, as APIs privadas possuem acesso restrito, sendo desenvolvidas para uso interno em empresas ou organizações. Um exemplo seria uma API privada que permite ao sistema de pagamentos da empresa acessar dados do sistema de gestão de recursos humanos. Devido à sua natureza voltada para os interesses da organização, essas APIs geralmente só são acessíveis a partir da rede interna da empresa.

2.6.1 Utilização de PHP no backend de aplicativos

O PHP é uma linguagem de programação de código aberto. Com ele, é possível criar aplicações dinâmicas e interativas, desde sites simples até sistemas complexos. O PHP

funciona no lado do servidor, processando os scripts e códigos antes de enviar o resultado para o usuário. Além disso, o PHP oferece recursos avançados para interagir com bancos de dados, facilitando a criação de sites que armazenam e recuperam informações. Através da linguagem PHP, é possível construir um backend robusto e flexível, capaz de processar dados, realizar consultas a bancos de dados, autenticação de usuários e gerenciar a lógica de negócios de um aplicativo (PHP, 2023).

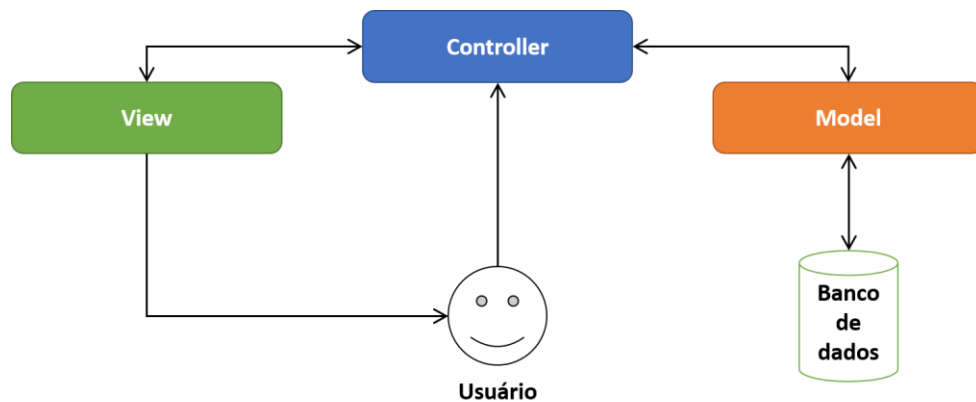
O PHP opera nos bastidores, sem ser diretamente visível ao usuário. Ele trabalha de forma discreta e eficiente nos bastidores, processando as solicitações e executando tarefas complexas, sem interferir na experiência do usuário na interface. Sua atuação silenciosa e eficaz permite que as páginas sejam dinâmicas e interativas, sem revelar o seu funcionamento interno ao usuário final. Em resumo, o PHP tem pouco a ver com layout, eventos ou qualquer coisa relacionada à aparência, pois a maior parte do que o PHP realiza é invisível para o usuário final (CONVERSE; PARK, 2003).

O PHP em conjunto com a arquitetura MVC pode ser uma escolha excelente. O MVC é um padrão de arquitetura de software. Ele sugere uma maneira para você pensar na divisão de responsabilidades. O princípio básico dele é a divisão da aplicação em três camadas: a camada de interação do usuário (view), a camada de manipulação dos dados (model) e a camada de controle (controller). Com o MVC, é possível separar o código relativo à interface do usuário das regras de negócio, o que sem dúvida traz muitas vantagens (TREINAWEB, 2021).

O site da TreinaWeb (2021) ainda conclui dizendo que no MVC, cada uma das camadas apresenta suas próprias responsabilidades, por exemplo, o Model tem a responsabilidade de representar o negócio. Sendo também responsável pelo acesso e manipulação dos dados na aplicação. A view é responsável pela interface que será apresentada, mostrando as informações do model para o usuário. Já o Controller é a camada de controle, responsável por ligar o model e a view, fazendo com que os models possam ser repassados para as views e vice-versa.

A Figura 5 ilustra um exemplo da estrutura da arquitetura MVC.

Figura 5 - Arquitetura MVC



Fonte: (TreinaWeb, 2021)

No desenvolvimento backend de aplicações utilizando PHP, o gerenciamento de dados desempenha um papel fundamental na manipulação e armazenamento eficiente das informações. De acordo com a Oracle (2019), "Gerenciamento de dados é a prática de coletar, manter e usar dados de forma segura, eficiente e econômica. O objetivo do gerenciamento de dados é ajudar pessoas, organizações e itens conectados a otimizar o uso de dados dentro dos limites da política e da regulamentação, para que possam tomar decisões e ações que maximizem o benefício para a organização. Uma estratégia robusta de gerenciamento de dados está se tornando mais importante do que nunca, já que as organizações dependem cada vez mais de ativos intangíveis para criar valor."

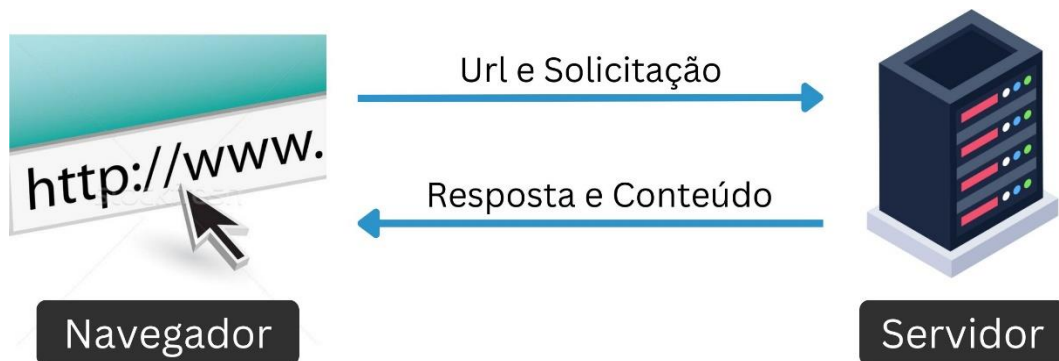
Sabendo como funciona o PHP no backend, o gerenciamento de dados, é importante compreender o HTTP. O protocolo HTTP (Hypertext Transfer Protocol) desempenha um papel fundamental na comunicação entre clientes e servidores. Ao utilizar o PHP no desenvolvimento backend de aplicações, é comum interagir com o HTTP para receber solicitações dos clientes, processá-las e enviar respostas adequadas. O PHP oferece recursos poderosos para lidar com as várias partes do protocolo HTTP, como cabeçalhos, métodos de requisição (como GET e POST), cookies e redirecionamentos. Essas funcionalidades permitem que os desenvolvedores implementem a lógica de negócios necessária para responder às solicitações dos clientes de forma eficiente e segura.

O HTTP, ou HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto), representa a pedra angular em que se baseia toda a estrutura da web. Cada interação conduzida por meio do HTTP se desenrola em duas etapas fundamentais: uma solicitação seguida de uma resposta. O protocolo HTTP, em si, é composto por múltiplas partes que se entrelaçam de formas inérgicas. Dentre essas partes, destacam-se a URL, que indica o destino da solicitação, o verbo utilizado para especificar a ação desejada, bem como uma variedade de cabeçalhos e códigos de status que auxiliam na compreensão e tratamento da transação. Por fim, mas não menos importante, encontramos o corpo das respostas, que

constitui o elemento visível e tangível que frequentemente capturamos ao explorar a web por meio de um navegador Mitchell (2013).

A Figura 6 ilustra o Diagrama de Comunicação HTTP entre navegadores e servidores.

Figura 6 - Diagrama de Comunicação HTTP entre Navegador e Servidor



Fonte: Elaborado pelos autores, 2023

Nesta seção foi explorada a relevância do PHP no desenvolvimento backend de aplicações, bem como a sua integração com o gerenciamento de dados e o protocolo HTTP. Ao utilizar o PHP, é possível criar sistemas eficientes e seguros, capazes de manipular e armazenar dados de forma adequada. O gerenciamento de dados desempenha um papel essencial nesse processo, garantindo a integridade e a consistência das informações. Além disso, o PHP oferece recursos para interagir com o protocolo HTTP, possibilitando a comunicação eficiente entre clientes e servidores, assim como a integração com APIs externas. Ao compreender esses conceitos, os desenvolvedores têm as ferramentas necessárias para criar aplicativos backend robustos e eficazes.

2.7 Trabalho Relacionados

Em uma revisão da literatura, (FREITAS et al., 2021) desenvolveram um aplicativo cujo objetivo principal é avaliar o consumo alimentar global de crianças escolares brasileiras entre 4 e 9 anos. O presente trabalho trata-se de um estudo metodológico que envolve a elaboração de um aplicativo online para smartphones e tablets, que permite a avaliação dietética das crianças.

Existem inúmeros trabalhos na literatura voltados para facilitar a gestão de refeitórios universitários. Conforme o trabalho apresentado por Nascimento (2021), para auxiliar o IFPB Campus Cajazeiras no gerenciamento do restaurante universitário, foi criado o

aplicativo MANAGER RU. O foco principal do aplicativo é a redução do tempo de espera na fila, além de conscientizar os alunos sobre o desperdício de alimentos, notificando-os sobre suas faltas.

Seguindo outra perspectiva, os trabalhos citados apresentam abordagens específicas relacionadas à área de alimentação e nutrição. No entanto, possuem objetivos distintos, enquanto esse trabalho tem como objetivo a criação de um aplicativo móvel que facilite a comunicação entre a UAN e os estudantes, os demais objetivam avaliar o consumo alimentar de crianças escolares brasileiras e auxiliar na gestão do restaurante universitário.

3 METODOLOGIA

A pesquisa é caracterizada como uma pesquisa aplicada. Esse tipo de pesquisa é orientada para a resolução de desafios práticos e específicos identificados nas operações diárias da Unidade de Alimentação e Nutrição (UAN), buscando aplicar conhecimentos teóricos preexistentes de maneira efetiva.

Nesse contexto, o objetivo foi desenvolver um aplicativo para otimizar a comunicação entre a Unidade de Alimentação e Nutrição (UAN) e os estudantes, proporcionando uma experiência mais ágil e conveniente para os usuários. Assim, a pesquisa envolveu a análise e a utilização de informações relevantes disponíveis na literatura, bem como a aplicação de métodos e técnicas adequadas para o desenvolvimento do aplicativo, visando a obtenção de resultados práticos e aplicáveis no contexto da UAN.

Ao mesmo tempo, a pesquisa também envolveu um processo de desenvolvimento do aplicativo, utilizando uma abordagem metodológica ágil. Esse enfoque permitiu um desenvolvimento iterativo e incremental, no qual o aplicativo é construído em etapas, com a possibilidade de ajustes e melhorias contínuas ao longo do processo. A metodologia ágil é adequada para o desenvolvimento de aplicativos, pois permite uma maior flexibilidade para lidar com mudanças e adaptações necessárias durante o desenvolvimento, considerando as necessidades e feedback dos usuários.

Assim, o processo de desenvolvimento do aplicativo incluiu etapas de planejamento, implementação e testes, sendo guiado pela abordagem ágil para garantir a entrega de um produto final eficiente de qualidade. A seguir, será apresentado uma visão geral do sistema, enfatizando os procedimentos metodológicos adotados e os requisitos levantados.

3.1 Procedimentos metodológicos

Os procedimentos metodológicos adotados durante o desenvolvimento da aplicação seguiram uma abordagem estruturada e orientada em etapas. Dentre os principais procedimentos utilizados, destacam-se a fase de definição dos requisitos, o design da interface, a implementação do aplicativo, a criação de diagramas, dentre outras. Abaixo será abordado um pouco mais a fundo sobre cada um dos procedimentos.

A definição dos requisitos do aplicativo foi uma etapa fundamental no processo, visando documentar as necessidades e expectativas do aplicativo. Para isso, adotou-se uma

abordagem estruturada, começando pelo levantamento dos requisitos funcionais e não funcionais. Os requisitos funcionais foram identificados com base nas funcionalidades essenciais que o aplicativo deve oferecer, como a listagem de tickets dos alunos, o cardápio diário e os avisos da UAN. Além disso, foram considerados os requisitos não funcionais, que abrangem aspectos de desempenho, segurança, usabilidade e compatibilidade com diferentes dispositivos móveis. Essa análise dos requisitos permitiu estabelecer uma base sólida para o desenvolvimento do aplicativo, garantindo que as necessidades dos usuários sejam devidamente contempladas.

O design de interface do usuário desempenhou um papel crucial no desenvolvimento do aplicativo, para que o mesmo fosse eficiente e mais agradável de usar. Os protótipos permitiram visualizar e testar a estrutura e funcionalidade do aplicativo, evitando que se tivesse retrabalhos. Já os fluxos de navegação claros e intuitivos serviram para facilitar a interação do usuário, garantindo uma experiência mais agradável. Além disso, o design pensando na UI/UX promoveu a usabilidade dos usuários, contribuindo para uma melhor aceitação do aplicativo.

Logo após a parte da prototipagem do aplicativo ter sido feita, se deu início a fase de implementação do mesmo, utilizando a linguagem JavaScript juntamente com o Framework React Native para a criação da parte visual do aplicativo. Já para a parte de obtenção dos dados, como tickets, avisos e o cardápio, foi utilizada uma API que foi criada utilizando a linguagem de programação PHP. Durante esse processo, ocorreu a integração dos componentes do aplicativo e a codificação das regras de negócio.

Durante a etapa de implementação do aplicativo, foi criado um diagrama de caso de uso para o aplicativo que foi de suma importância. O diagrama serviu como um mapa direcionador, delineando as interações principais entre os usuários e o sistema. Essa abordagem estratégica proporcionou uma compreensão clara das funcionalidades essenciais, orientando o desenvolvimento para atender às expectativas dos usuários da UAN. A presença e análise desse caso de uso foi fundamental para estabelecer uma base sólida durante a implementação.

Após a conclusão da implementação, iniciaram-se as fases de testes e validações. Essas etapas desempenharam um papel fundamental para assegurar a qualidade do aplicativo. Inicialmente, foram conduzidos testes em colaboração com a nutricionista da UAN. Posteriormente, prosseguiram-se testes de usabilidade envolvendo os usuários da UAN, com o objetivo de avaliar a experiência geral do usuário, identificando eventuais problemas de interface ou obstáculos na navegação. Durante essa fase, a coleta de feedbacks e sugestões dos usuários foi de suma importância, pois contribuiu significativamente para o aprimoramento do

aplicativo. Algumas dessas informações foram incorporadas ao processo, visando proporcionar uma experiência ainda mais satisfatória.

Esses procedimentos metodológicos foram adotados para garantir a eficácia, qualidade e usabilidade da aplicação. O acompanhamento de cada etapa permitiu uma abordagem estruturada e organizada, resultando em um produto final alinhado com as necessidades dos usuários e com os objetivos propostos.

Através da definição clara dos requisitos, do design da interface, da implementação, dos testes e validações, e da implantação monitorada foi possível alcançar um aplicativo que atendesse às necessidades e expectativas impostas, garantindo usabilidade e funcionalidade.

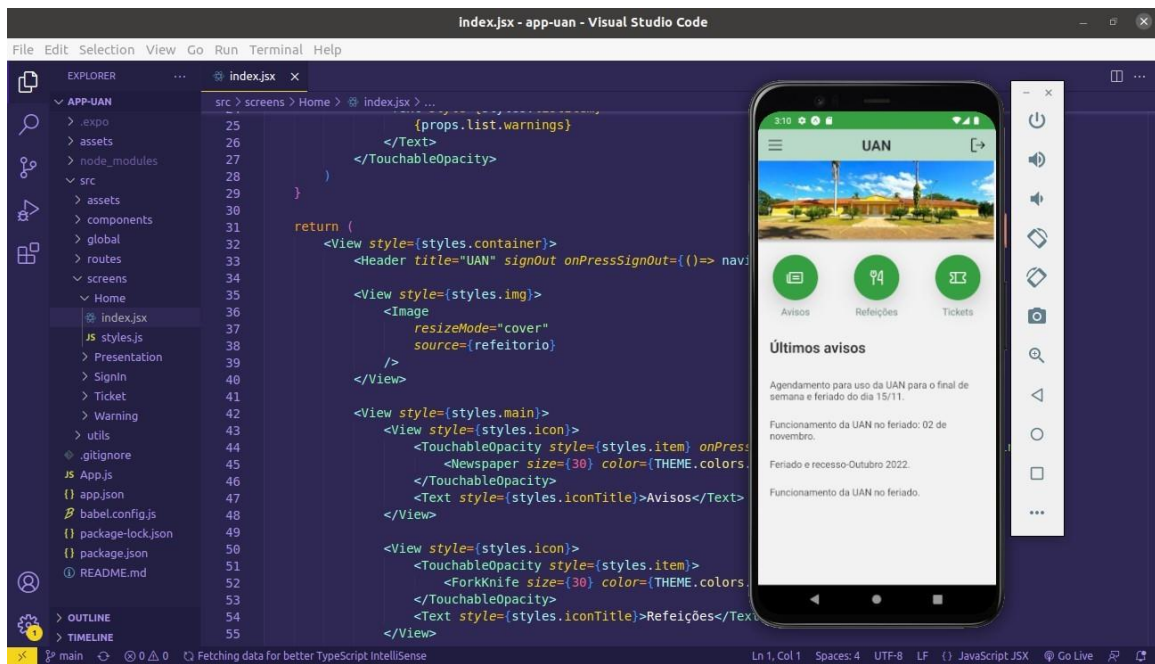
Visando atender as etapas de planejamento, o levantamento de requisitos foi fundamental para garantir que as funcionalidades fossem implementadas e as necessidades atendidas.

3.2 Ferramentas e tecnologias utilizadas

Para o desenvolvimento do aplicativo SIAR, foi utilizada uma combinação de IDEs (Seção 2.3.1), o visual Studio Code e o Android Studio. O Visual Studio Code foi empregado como editor de código principal, fornecendo recursos avançados de edição, personalização e extensibilidade. Já o Android Studio desempenhou um papel fundamental na criação e emulação do aplicativo Android, oferecendo recursos específicos para o desenvolvimento nessa plataforma. Essas IDEs proporcionaram um ambiente integrado e eficiente para escrever, testar e depurar o código, garantindo a qualidade e a funcionalidade adequadas do aplicativo.

A Figura 7 ilustra a IDE Vscod e Android Studio rodando em conjunto para criação de um app.

Figura 7 - VsCode em conjunto com Android Studio



Fonte: Elaborado pelos autores, 2023

No desenvolvimento da API do aplicativo foi utilizado a linguagem de programação PHP em conjunto com a arquitetura MVC (Model-View-Controller). O PHP (Seção 2.6.1) foi utilizado do lado do servidor para criar uma API que conectará o aplicativo aos dados armazenados nos bancos de dados do Campus. Com isso foi possível autenticar o usuário e obter seus tickets, o cardápio da semana e os avisos da UAN.

No desenvolvimento do aplicativo, foi utilizado o framework React Native (Seção 2.5.3.1). Ele é uma poderosa ferramenta que permitiu a criação de um aplicativo nativo para dispositivos móveis usando JavaScript. Com o React Native, foi possível desenvolver uma interface de usuário rica e responsiva, com componentes nativos para iOS e Android. Através do uso de JavaScript, foi possível implementar a lógica do aplicativo, lidar com a interação do usuário e acessar recursos do dispositivo.

Ao longo da realização do projeto foi utilizado o sistema de controle de versão Git e a plataforma de hospedagem GitHub. O Git é uma ferramenta amplamente reconhecida para gerenciar o controle de versões do código-fonte. Com o Git, foi possível acompanhar e registrar todas as alterações feitas no código ao longo do tempo, facilitando o rastreamento e a reversão de alterações, quando necessário.

Além disso, o GitHub serviu como um repositório remoto para hospedar o projeto, permitindo o compartilhamento do código-fonte com todos os membros envolvidos na criação.

Através do GitHub, se tornou mais fácil controlar as versões, gerenciar problemas e solicitações de pull, e trabalhar de forma colaborativa no desenvolvimento do aplicativo. A utilização do Git e do GitHub contribuiu para um fluxo de trabalho eficiente, garantindo a integridade do código e facilitando a colaboração entre os membros.

Durante o desenvolvimento, foi utilizado a ferramenta de prototipagem (Seção 3.3) e design Figma¹¹. Trata-se de uma plataforma colaborativa baseada na nuvem, que permite a criação de wireframes (esqueleto de um site, sistema ou aplicativo), protótipos interativos e designs visuais de alta qualidade para o aplicativo. Com o Figma, ficou mais fácil criar protótipos interativos para validar e testar as funcionalidades do aplicativo antes da implementação final. Essa etapa de prototipagem foi fundamental para identificar melhorias, ajustes de usabilidade e obter feedbacks. Além disso, o Figma ofereceu recursos avançados, como a criação de componentes reutilizáveis, o compartilhamento fácil de projetos e a colaboração em tempo real com outros membros. Com a ajuda do Figma, foi possível uma melhor visualização e comunicação das ideias de design e funcionalidades, garantindo um resultado final esteticamente agradável e funcional.

3.3 Prototipagem

A prototipagem¹² é um processo essencial no desenvolvimento de aplicativos, permitindo que os designers e desenvolvedores criem representações visuais interativas de suas ideias. Um protótipo é uma versão preliminar de um aplicativo, geralmente não funcional, que permite testar e validar conceitos, fluxos de interação, layouts e funcionalidades antes da implementação completa. Ao criar o protótipo, é possível experimentar diferentes abordagens de design, testar a usabilidade e obter feedback dos usuários, permitindo aprimorar e refinar o aplicativo antes de investir tempo e recursos significativos na sua construção. Existem várias ferramentas disponíveis para a prototipação, desde ferramentas de desenho vetorial até ferramentas específicas de prototipagem interativa, que permitem simular interações e fluxos de usuário.

Na Figura 8 é apresentada a prototipação da tela inicial do aplicativo, na qual o usuário terá acesso às principais funcionalidades de navegação. Essas funcionalidades incluem

¹¹ <https://www.figma.com/best-practices/guide-to-developer-handoff/components-styles-and-documentation/>

¹² <https://www.take.net/blog/tecnologia/o-que-e-prototipacao/>

o cardápio semanal, a listagem de tickets e os avisos diários. A figura permite visualizar a disposição e organização desses elementos na interface do aplicativo.

Figura 8 - Tela de *home* do aplicativo



Fonte: Elaborado pelos autores, 2023

Em resumo, a prototipagem é uma etapa crucial no desenvolvimento de aplicativos, permitindo testar e aprimorar as ideias antes da implementação completa. Para a tela inicial do aplicativo, o objetivo da prototipagem é ajudar a criar uma experiência de usuário eficaz e atraente, garantindo que os elementos visuais e interações estejam bem projetados e alinhados com os objetivos do aplicativo.

4 RESULTADOS E DISCUSSÃO

Esta seção expõe os resultados da aplicação móvel desenvolvida neste trabalho, que se destina a aprimorar a gestão de informações da Unidade de Alimentação e Nutrição (UAN) do Instituto Federal de Minas Gerais - Campus São João Evangelista. Além disso, detalha as fases cruciais realizadas durante o processo de desenvolvimento do aplicativo e inclui os resultados obtidos a partir das avaliações realizadas. Esta seção está estruturada da seguinte sequência: Requisitos do Sistema; Sessão de Modelagem e Aplicação Desenvolvida.

4.1 Requisitos do sistema

Os requisitos do sistema são especificações detalhadas que descrevem as capacidades técnicas necessárias para que um software ou sistema funcione corretamente em um computador ou dispositivo específico. Eles são essenciais para garantir que um software ou sistema seja executado de forma eficaz e sem problemas.

A documentação completa dos requisitos do sistema é fundamental para o desenvolvimento, implementação e uso bem-sucedido de um software ou sistema. Eles servem como um guia para desenvolvedores, administradores de sistemas e usuários finais, garantindo que todos tenham uma compreensão clara das necessidades e limitações do sistema.

Os requisitos podem ser classificados em dois tipos principais: funcionais e não funcionais.

4.1.1 Requisitos Funcionais

Os requisitos funcionais definem as funcionalidades específicas que o sistema ou aplicativo deve oferecer. Eles descrevem as ações que o sistema deve ser capaz de realizar. Neste trabalho foram identificados os principais requisitos funcionais:

1. Autenticação de Usuário (Login):
 - a) O sistema deve permitir que os usuários se autentiquem usando seu Registro Acadêmico (RA) e senha.
 - b) Os dados referentes a esta autenticação são os mesmos utilizados pelo discente nos sistemas do IFMG, não cabendo ao aplicativo ações como: alteração, recuperação ou desativação de senha.

2. Consulta de Tickets:
 - a) Os usuários autenticados devem ter a capacidade de visualizar e consultar os tickets relacionados a si próprio.
 - b) A tela de tickets deve exibir informações relevantes, como data atual e quantidade de tickets disponíveis.
3. Consulta de Cardápio do Dia:
 - a) Os usuários devem poder acessar o cardápio do dia, que deve incluir informações sobre as refeições oferecidas.
4. Consulta de Avisos da UAN:
 - a) Os usuários devem ter acesso aos avisos publicados pela Unidade de Alimentação e Nutrição (UAN).
 - b) Os avisos devem ser organizados por data e serem facilmente navegáveis.

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais especificam os critérios que não estão diretamente relacionados às funcionalidades, mas são cruciais para a eficácia do sistema. Eles se concentram em aspectos como desempenho, segurança, usabilidade e qualidade. Neste trabalho, considera-se os principais requisitos não funcionais:

1. Compatibilidade com Android:
 - a) O aplicativo deve ser compatível com dispositivos Android com uma versão mínima do sistema operacional (Android 7.0 ou posterior).
 - b) A interface do aplicativo deve se adaptar a diferentes tamanhos de tela e resoluções.
2. Desempenho:
 - a) O tempo de resposta do aplicativo não deve exceder 3 segundos em ações comuns, como fazer login ou consultar os tickets.
 - b) O aplicativo deve ser otimizado para uso eficiente de recursos, como CPU e memória.
3. Segurança:
 - a) Senhas dos usuários não devem ser armazenadas no aplicativo, nem mesmo com segurança usando técnicas de criptografia. Toda a parte de

segurança de senha está implementada nos sistemas do IFMG, cabendo ao aplicativo apenas autenticar-se por método HTTP em endpoint específico.

- b) Os dados pessoais dos usuários, como Nome e RA, devem ser protegidos de acesso não autorizado, estando disponíveis apenas ao usuário que autenticou.

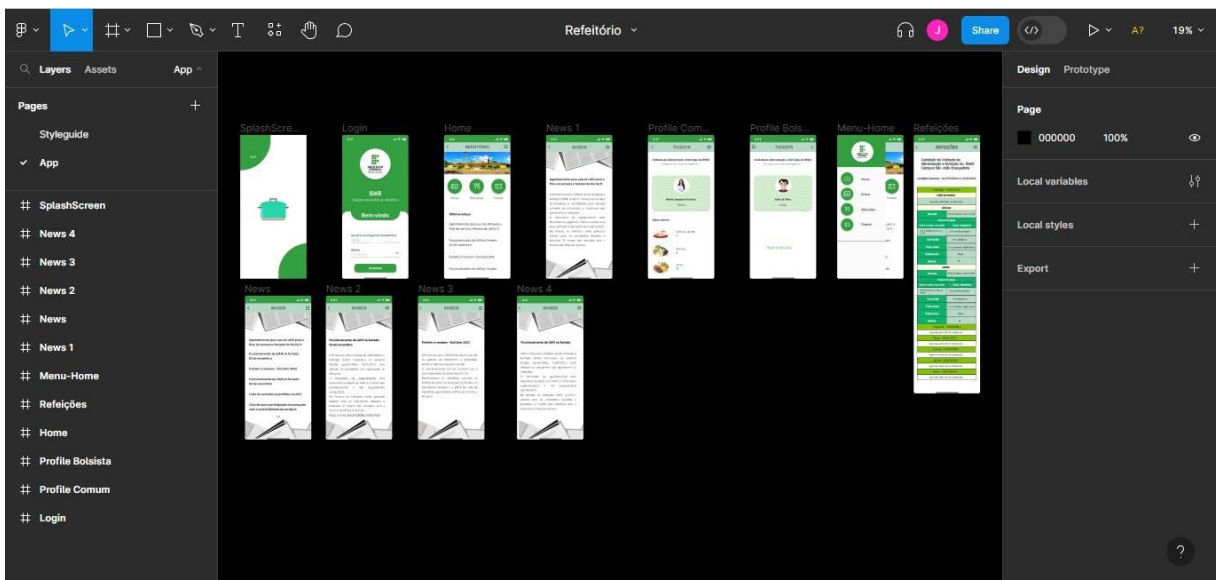
4. Usabilidade:

- a) O aplicativo deve seguir as diretrizes de design de interface do usuário (UI) para garantir que seja intuitivo e fácil de usar.
- b) A linguagem e os ícones devem ser compreensíveis para o público-alvo.

Para cumprir os requisitos, adotou-se boas práticas de programação, como otimização de código, refatoração constante, gerenciamento eficiente de memória e minimização de atividades em segundo plano quando o aplicativo não estiver ativamente em uso. Isso é importante para garantir que o aplicativo seja responsivo e eficiente, independentemente do dispositivo em que for executado.

Após a etapa de levantamento de requisitos, realizou-se a criação da interface do aplicativo utilizando a ferramenta Figma. A etapa de prototipagem teve como objetivo simplificar o desenvolvimento da interface e prevenir questionamentos por parte dos usuários em relação ao layout. A seguir, pode ser observado na Figura 9 o resultado final do protótipo das telas do aplicativo no Figma.

Figura 9 - Design do aplicativo na ferramenta Figma



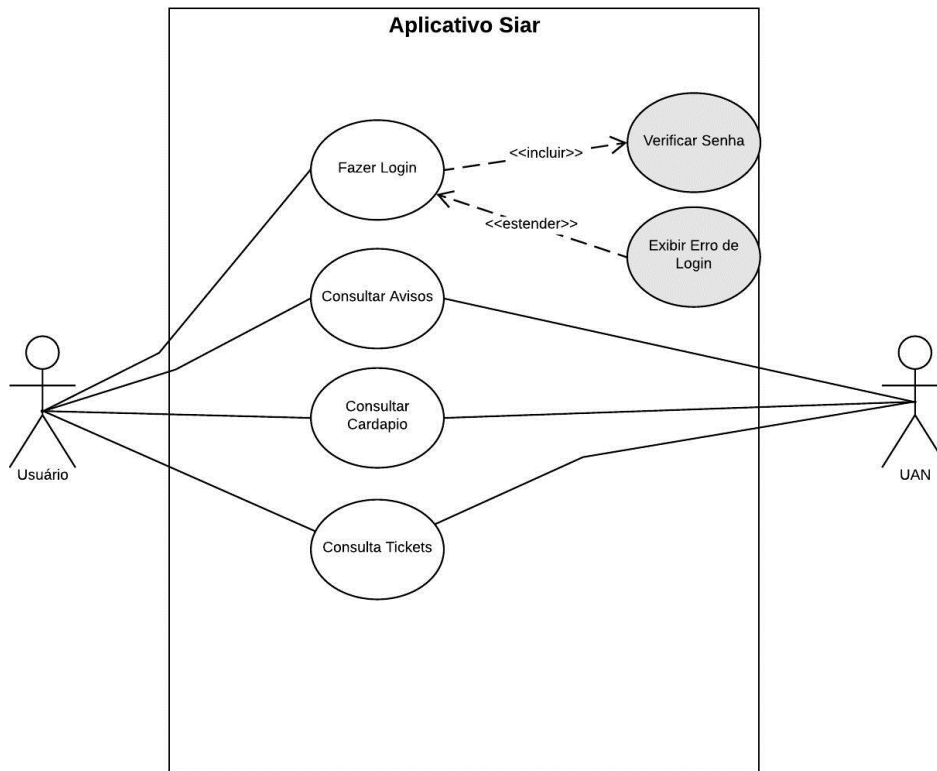
Fonte: Elaborado pelos autores, 2023

4.2 Sessão de modelagem

Nesta seção abordaremos a estrutura e a funcionalidade do aplicativo projetado para UAN. Por meio da criação e análise de diagramas, esta seção fornecerá uma visão abrangente sobre como o aplicativo opera e interage com os usuários. Os diagramas, serão apresentados de forma a ilustrar os principais recursos e as interações-chave que os usuários terão ao utilizar o aplicativo. Além disso, será examinado detalhadamente a arquitetura geral do sistema, garantindo uma compreensão sólida de como as diferentes partes do aplicativo se conectam para oferecer aos usuários a capacidade de consultar seus tickets de alimentação, acessar o cardápio da semana e receber informações importantes da Unidade de Alimentação e Nutrição (UAN) da instituição.

A Figura 10 mostra o Diagrama de Caso de Uso do aplicativo SIAR, que ilustra suas funcionalidades e a interação dos usuários e da Unidade de Alimentação e Nutrição (UAN) com o aplicativo. Os casos de uso representam as ações específicas que os usuários podem realizar, enquanto os atores são agentes externos, como estudantes e a equipe da UAN, envolvidos na interação. Esse diagrama oferece uma visão clara de como o SIAR atende às necessidades dos usuários, incluindo consulta de tickets de alimentação, acesso ao cardápio semanal e recebimento de informações cruciais da UAN.

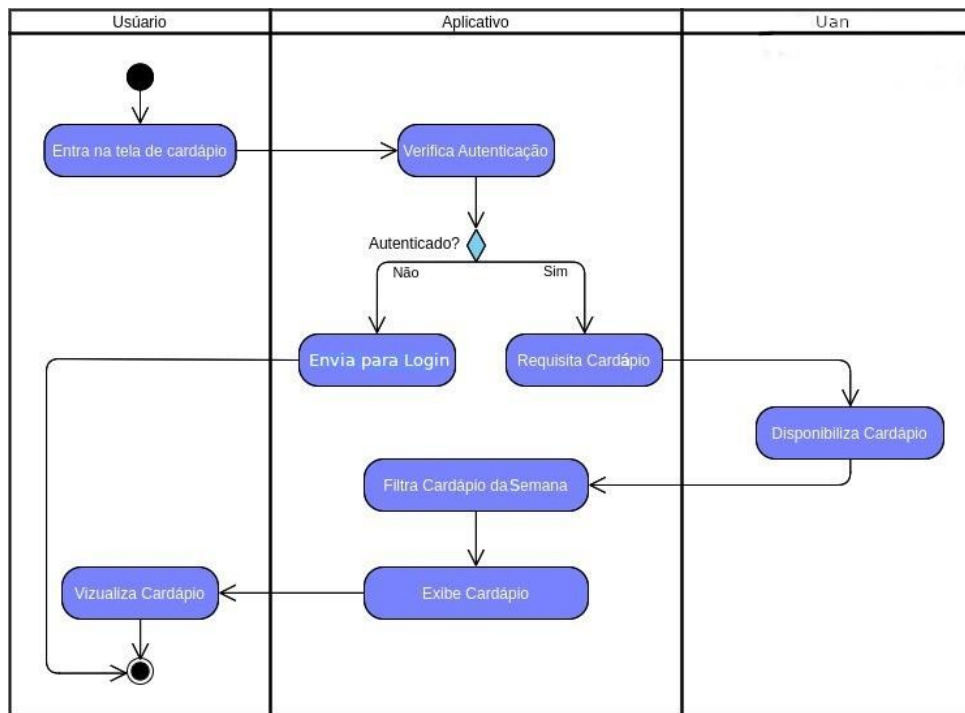
Figura 10 - Diagrama de Caso de uso do aplicativo SIAR



Fonte: Elaborado pelos autores, 2023

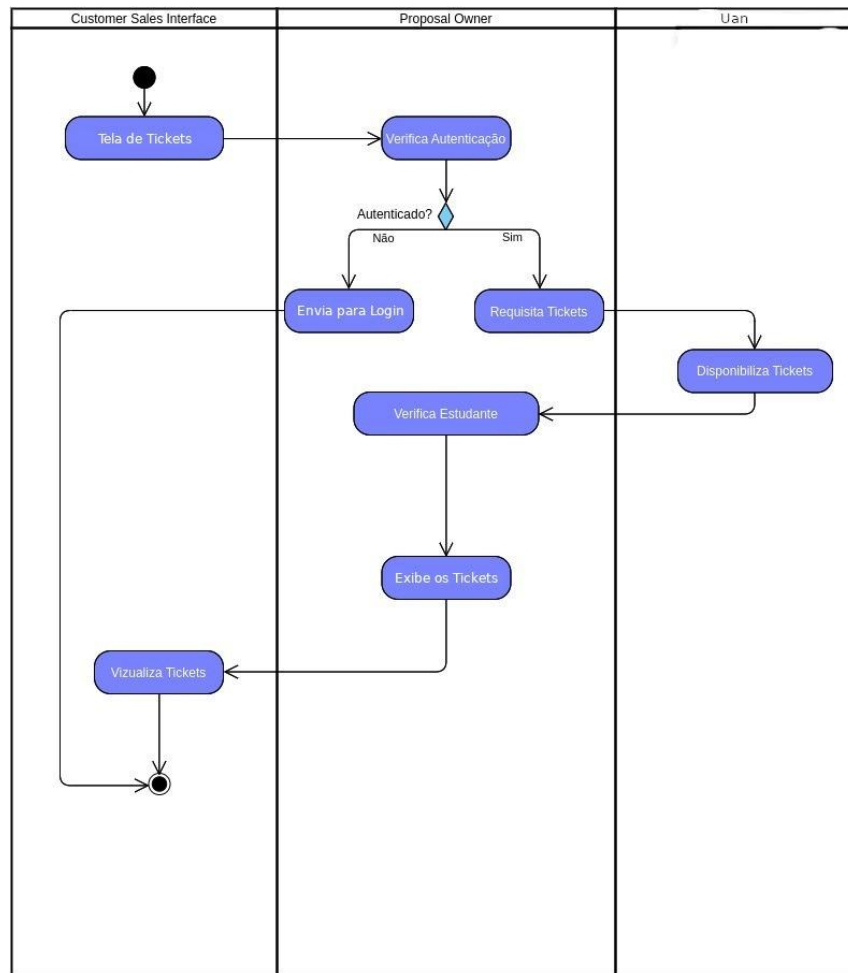
A seguir serão apresentados três Diagramas de Atividade distintos, cada um detalhando um aspecto fundamental do funcionamento do nosso aplicativo. Esses diagramas abordarão as atividades relacionadas à consulta de Tickets de Alimentação, ao acesso ao Cardápio Semanal e à visualização dos Avisos da Unidade de Alimentação e Nutrição (UAN). Através desses diagramas, buscamos proporcionar uma compreensão abrangente e visual das interações entre os usuários e o sistema, destacando como os diversos recursos do aplicativo são acessados e utilizados.

Figura 11 - Diagrama de Atividade - Cardápio



Fonte: Elaborado pelos autores, 2023

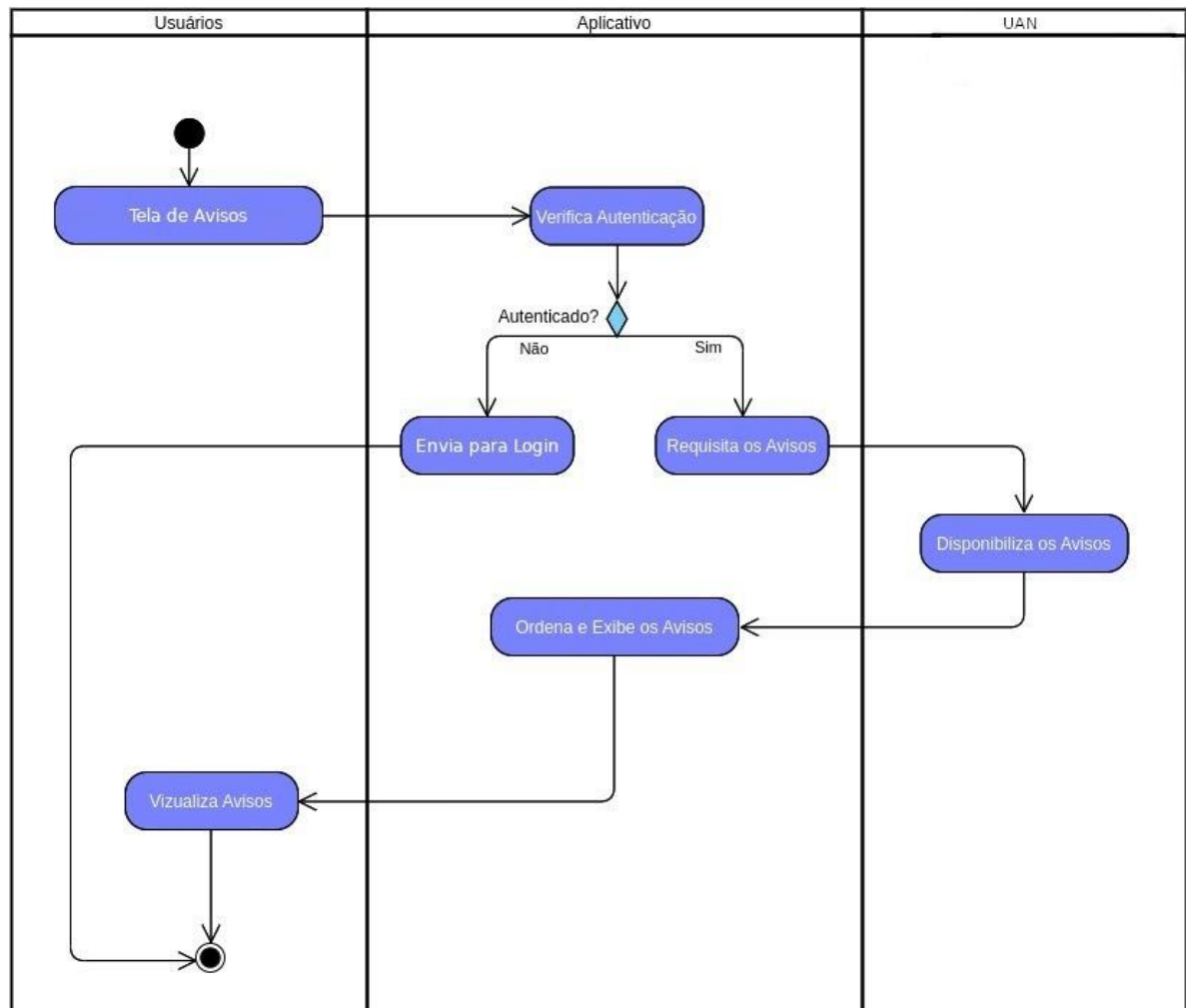
A Figura 11 ilustra o Diagrama de Atividade do Cardápio. Nela é possível constatar que quando o usuário logado clicar para entrar na tela de cardápio, o sistema requisita os dados do cardápio e logo em seguida faz a filtragem desses dados para filtrar apenas os dados da semana atual e em seguida exibe os dados de forma organizada na tela para o usuário.

Figura 12 - Diagrama de Atividade - *Tickets*

Fonte: Elaborado pelos autores, 2023

A Figura 12 representa o Diagrama de Atividade dos Tickets. Semelhante à Figura 11, neste diagrama é possível constatar que quando o usuário logado clicar para entrar na tela de tickets, o sistema requisita os dados do mesmo e logo em seguida faz a filtragem desses dados para recuperar os tickets de café, almoço e jantar associados ao usuário que está logado. Em seguida o sistema exibe os dados na tela separando-os em suas respectivas seções e mostrando aos usuários seus tickets.

Figura 13 - Diagrama de Atividades - Avisos



Fonte: Elaborado pelos autores, 2023

Já na Figura 13, é possível vislumbrar o Diagrama de Atividades dos Avisos. Assim como os diagramas anteriores, este diagrama possibilita aos usuários logados acessar a tela de avisos e visualizar todos os avisos disponibilizados pela UAN.

4.3 Aplicação desenvolvida

Quando o usuário iniciar o aplicativo, deparar-se com a tela de login, onde poderá ser feito o login por meio do seu Registro Acadêmico (RA), e senha já cadastrados no próprio IFMG.

Figura 14 - Tela de login



Fonte: Elaborado pelos autores, 2023

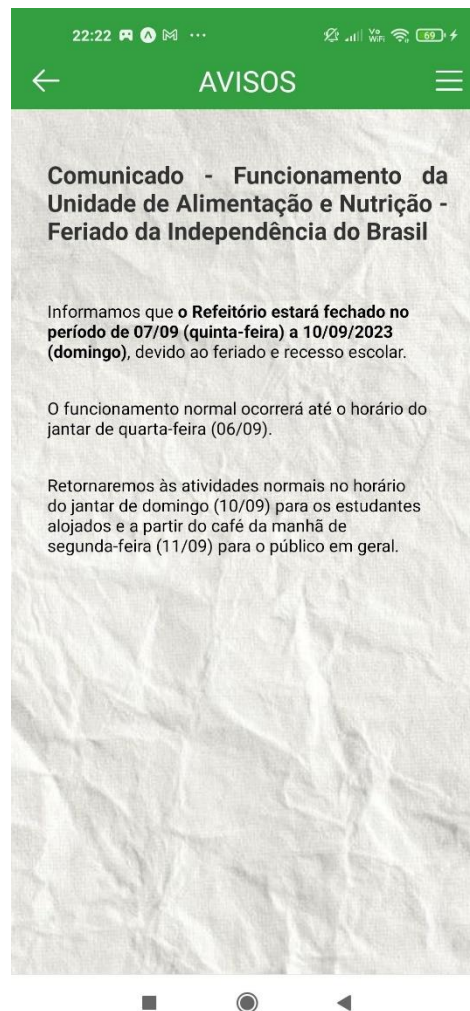
Após efetuar o login, é apresentada ao usuário a tela inicial. Nesta tela, o usuário tem acesso a todas as funcionalidades de navegação disponíveis no aplicativo, incluindo avisos, tickets, informações sobre refeições e agendamento para o final de semana.

Figura 15 - Tela de *home* do aplicativo

Fonte: Elaborado pelos autores, 2023

A Figura 16 representa a tela de aviso que será exibida sempre que um usuário clicar em um título de aviso listado na página inicial. Esta tela de aviso é projetada para fornecer informações importantes da UAN ao usuário e é acionada quando o usuário interage com um dos títulos de aviso listados.

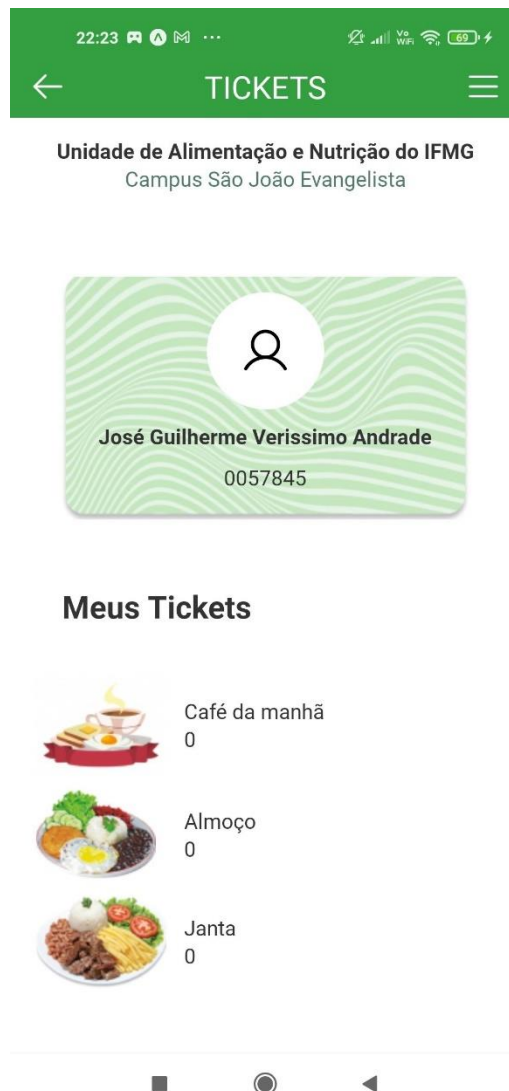
Figura 16 - Tela de avisos do aplicativo



Fonte: Elaborado pelos autores, 2023

A Figura 17 representa a tela de listagem de tickets que exibe se o usuário possui créditos para almoço, jantar ou café da manhã.

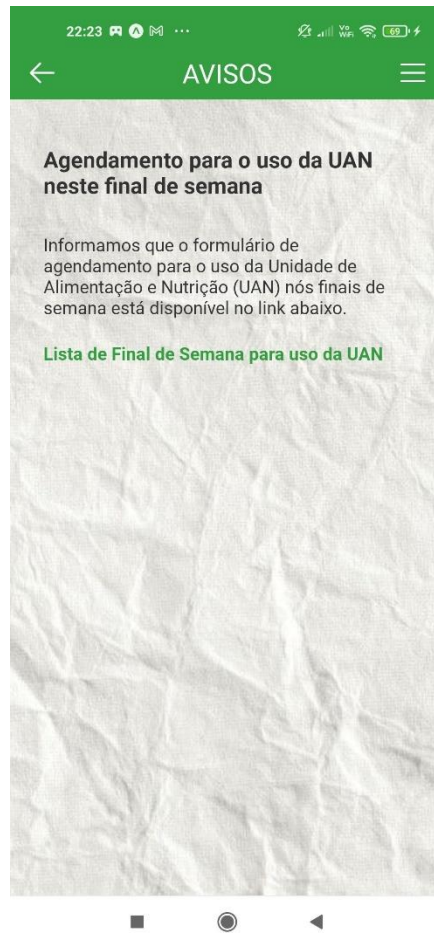
Figura 17 - Tela de tickets do aplicativo



Fonte: Elaborado pelos autores, 2023

A Figura 18 representa a tela de agendamento para o final de semana destinada aos usuários que desejam realizar alguma refeição durante esse período. Os interessados devem acessar o formulário e efetuar o cadastro.

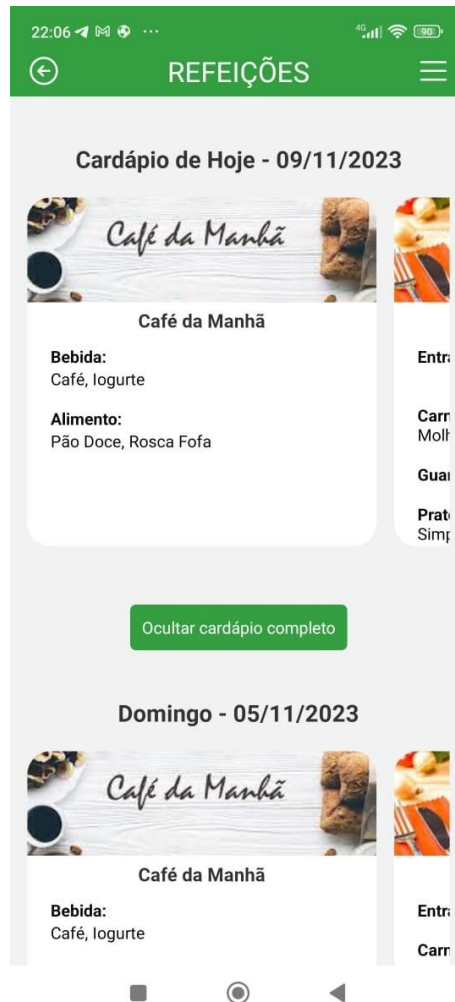
Figura 18 - Tela de agendamento de final de semana



Fonte: Elaborado pelos autores, 2023

A Figura 19 apresenta a tela de visualização de refeições semanais. Ao acessar essa tela inicialmente, o usuário se depara com as opções de refeição do dia, incluindo almoço, café da manhã e jantar. Caso o usuário tenha interesse em conhecer as refeições dos demais dias da semana, é necessário clicar no botão "Ver cardápio completo". Ao acionar esse botão, serão listadas todas as refeições semanais disponibilizadas, pela UAN (Unidade de Alimentação e Nutrição).

Figura 19 - Tela de cardápio semanal



Fonte: Elaborado pelos autores, 2023

4.4 Avaliação do sistema

Após a conclusão da fase de desenvolvimento, foi agendada uma reunião com a Nutricionista, vinculada à Unidade de Alimentação e Nutrição (UAN) do Instituto Federal de Minas Gerais (IFMG) - Campus São João Evangelista. Nessa reunião, o aplicativo desenvolvido foi apresentado com o propósito de permitir que a profissional realizasse os testes necessários relacionados ao software. Com o intuito de avaliar de maneira abrangente, foram elaboradas cinco perguntas específicas, nas quais a Nutricionista pôde atribuir uma pontuação de 1 a 5 para

cada questão, além de fornecer observações detalhadas sobre sua experiência e percepções durante a avaliação do aplicativo.

A Tabela 6 apresenta as respostas da nutricionista durante a etapa de teste. Neste contexto, 1 representa a menor nota possível para a respectiva pergunta, enquanto 5 representa a nota máxima atribuível a essa pergunta.

Tabela 4 - Avaliação do Aplicativo

Pergunta	Avaliação 1-5
O aplicativo é intuitivo e fácil de usar para os usuários da UAN?	5/5
Os recursos do aplicativo atendem às necessidades específicas da UAN e de seus usuários?	5/5
O aplicativo fornece informações precisas e atualizadas sobre cardápio, horários e ingredientes dos pratos?	5/5
O aplicativo atente o objetivo que foi proposto inicialmente?	5/5

Fonte: Elaborado pelos autores, 2023

Considerando que este aplicativo foi desenvolvido para atender às necessidades de estudantes e servidores que frequentam a UAN, torna-se relevante disponibilizá-lo para um grupo seletivo de usuários, a fim de realizarem testes e fornecerem feedbacks. Com esse propósito, elaborou-se um formulário contendo cinco perguntas estruturadas, para permitir que os usuários atribuíssem uma pontuação de 1 a 5 para cada questão, refletindo suas avaliações. O formulário foi disponibilizado de forma online para uma amostra representativa de 9 usuários da UAN. Destes, 8 pertencem ao curso de Sistemas de Informação, enquanto 1 está matriculado no curso de Agronomia. A escolha por disponibilizá-lo de maneira virtual facilitou a participação efetiva dos usuários.

A amostra conta apenas com 9 participante, pois, uma amostra modesta é muitas vezes suficiente para identificar tendências e avaliar a usabilidade do aplicativo. Além disso, a limitação da amostra a 9 usuários se justifica pelo perfil específico dos participantes, todos vinculados à UAN. De acordo com Cristina (2023) em sua pesquisa sobre tamanho amostral, embora amostras maiores proporcionem maior precisão, amostras pequenas são viáveis, especialmente em contextos desafiadores. Nesse contexto específico, a amostra reduzida

permitiu uma análise mais focada da usabilidade do aplicativo, demonstrando que, em certas situações, amostras pequenas podem fornecer insights valiosos.

As respostas dos usuários foram compiladas na Tabela 7, refletindo suas percepções e experiências. Cada pontuação atribuída contribui para uma análise abrangente da eficácia e usabilidade do aplicativo.

Tabela 5 - Avaliação do aplicativo com usuários

Pergunta	Avaliação 1-5
Os recursos do aplicativo atendem ao que foi proposto?	5/5
O aplicativo fornece informações precisas e atualizadas sobre os dados?	5/5
Como você avalia a interface visual do aplicativo?	4.9/5
A navegação do aplicativo é suave e sem complicações?	4.8/5
O aplicativo é eficiente em fornecer informações como avisos e cardápio?	4.7/5
Você considera que o aplicativo é fácil de usar?	4.8/5
Como você avalia o aplicativo como um todo?	4.9/5

Fonte: Elaborado pelos autores, 2023

Esse processo de avaliação é crucial para garantir que o aplicativo atenda às expectativas e requisitos dos usuários finais, resultando em melhorias contínuas e refinamentos. A participação ativa dos usuários na fase de teste e *feedback* é valorizada, pois contribui significativamente para a qualidade e aprimoramento do projeto. Ademais, os bons resultados demonstrados na Tabela 6 enfatizam que o objetivo inicial proposto e discutido em conjunto com a UAN foram atendidos em sua totalidade. Durante a interação com os usuários, foram identificadas algumas dificuldades que apontam para áreas específicas com potencial de aprimoramento, incluindo desafios relacionados à adaptação a diferentes tipos de dispositivos que precisou ser melhorado. Por exemplo, algumas configurações específicas de alguns dispositivos poderiam impactar a apresentação visual do aplicativo. Além disso, foram observadas dificuldades na assimilação de funcionalidades específicas e na familiarização com a interface do aplicativo. Essas identificações críticas proporcionaram insights valiosos, destacando a necessidade de uma atenção mais detalhada e um esforço aprimorado nas áreas mencionadas.

5 CONSIDERAÇÕES FINAIS

O principal objetivo deste projeto foi criar uma aplicação que tornasse mais fácil o acesso a informações importantes da Unidade de Alimentação e Nutrição (UAN). Ao percorrer todas as etapas deste estudo, alcançamos uma compreensão mais profunda sobre o processo de desenvolvimento de aplicativos.

Ao analisarmos tanto o sistema quanto as respostas obtidas por meio dos questionários aplicados, fica evidente que a aplicação foi não apenas bem recebida, mas também prontamente aceita pelos potenciais usuários. Essa aceitação representa não apenas a conquista dos objetivos estabelecidos, desde a compreensão e levantamento de requisitos até a implementação, teste e disponibilização do aplicativo, mas também simboliza uma jornada de aprendizado contínuo e superação de desafios.

Apesar de termos atingido os objetivos nesta versão inicial, identificamos oportunidades para a implementação de novas funcionalidades. Como perspectiva para futuros desenvolvimentos, consideramos a viabilidade de incorporar notificações, permitindo que o aplicativo alerte os usuários sobre avisos ou cardápios adicionados pela UAN. Além disso, a inclusão de um modo escuro, proporcionando aos usuários a flexibilidade de escolher a configuração conforme sua preferência. Por fim, mas não menos relevante, a criação de uma seção dedicada às avaliações dos usuários sobre o desempenho da UAN são sugestões valiosas para trabalhos futuros. Seria também extremamente crucial incluir uma versão do aplicativo para dispositivos iOS, o que faria o aplicativo estar disponível para boa parte dos estudantes.

No encerramento, ressaltamos que este estudo proporcionou diversas experiências positivas, permitindo a aplicação prática dos conhecimentos adquiridos em ambiente acadêmico. O desenvolvimento desta aplicação não apenas representa uma contribuição significativa para o instituto, mas também proporcionou uma compreensão mais aprofundada das fases envolvidas no processo de criação de um aplicativo móvel e sua interação com bancos de dados.

Esse percurso não apenas consolidou o aprendizado teórico, mas também destacou a importância de adaptação e melhoria contínua no desenvolvimento de soluções tecnológicas. Concluímos este projeto com a certeza de que o aplicativo não é apenas uma ferramenta funcional, mas um reflexo do comprometimento com a qualidade e a busca constante por melhorias na experiência do usuário e na eficiência da UAN.

REFERÊNCIAS

- ACKER, E. V.; WEBER, T. S.; CECHIN, S. L. **Injeção de falhas para validar aplicações em ambientes móveis**. In: SBC. Anais do XI Workshop de Testes e Tolerância a Falhas. [S.l.], 2010. p. 61–74.
- CERQUEIRA, D.; BITTENCOURT, R. **Comparação e Avaliação de Frameworks Mobile Multiplataforma, 2016**.
- CLARK, M. **Qual linguagem de programação é usada para desenvolver aplicativos iOS? 2016**. Disponível em: <<https://blog.back4app.com/pt/qual-linguagem-de-programacao-e-usada-para-desenvolver-aplicativos-ios/>>.
- COLARES, L. G. T.; FREITAS, C. M. d. **Processo de trabalho e saúde de trabalhadores de uma unidade de alimentação e nutrição: entre a prescrição e o real do trabalho**. Cadernos de Saúde Pública, SciELO Brasil, v. 23, p. 3011–3020, 2007.
- CONVERSE, T.; PARK, J. **PHP: a bíblia**. [S.l.]: Gulf Professional Publishing, 2003.
- CORRAL, L.; JANES, A.; REMENCIUS, T. **Potential advantages and disadvantages of multiplatform development frameworks a vision on mobile environments**. Procedia Computer Science, v. 10, p. 1202–1207, 2012. ISSN 1877-0509. ANT 2012 and MobiWIS 2012.
- Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050912005303>>.
- COURSERA. **Quais são as melhores linguagens para desenvolvimento de aplicativos? 2022**. Disponível em: <<https://www.coursera.org/articles/best-language-for-app-development>>.
- CRISTINA, A. **Tamanho Amostral: 4 Coisas Que Você Precisa Saber**. 2023. Disponível em: <<https://estatmg.com.br/2023/02/11/tamanho-amostal-4-coisas-que-voce-precisa-saber/>>.
- DANIELSSON, W. **React native application development**. Linköpings universitet, Swedia, v. 10, n. 4, p. 10, 2016.
- DEITEL, H.; DEITEL, P.; DEITEL, A. **Android: Como Programar**. [S.l.]: Bookman Editora, 2015.
- DHARMWAN, S. **Aplicativos híbridos versus aplicativos nativos - a lista de verificação para descobrir o que é certo para você**, cynoteck, 2021.
- ENGELMANN, L. K.; SÁPIRAS, L. A. **MovIES: Aplicativo Integrado de Serviços para Instituições de Ensino Superior**, 2016.

- FORTE, G.; FRANÇA, J. A.; SOUZA, D. A. de; NABARRO, C. B. M.; SABA, W. B. **Estratégias inteligentes para desenvolvimento de aplicativos mobile multiplataforma. Gestão e Tecnologia: Reflexões e Práticas**, p. 43, 2017.
- FREITAS, J. V. et al. **Desenvolvimento de um aplicativo móvel para avaliar a dieta de crianças escolares brasileiras entre 4 e 9 anos de idade**. Tese (Doutorado), 2021.
- GIKAS, J.; GRANT, M. M. **Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones social media. Internet and Higher Education**, v. 19, p. 18–26, 2013.
- GRIFFITH, C. **O que é o desenvolvimento de aplicativos móveis híbridos?** 2017. Disponível em: <<https://ionic.io/resources/articles/what-is-hybrid-app-development>>.
- GUEDES, M. **App Nativo x App Híbrido: existe o melhor?** 2017. Disponível em: <<https://www.treinaweb.com.br/blog/app-nativo-x-app-hibrido-existe-o-melhor>>.
- HECKE, C. **Como dispositivos móveis estão mudando forças de trabalho ao redor do mundo**. 2013. Disponível em: <<https://www.tecmundo.com.br/tendencias/45649-como-dispositivos-moveis-estaomudando-forcas-de-trabalho-ao-redor-do-mundo.htm>>.
- HUTRI, H. **Comparison of React Native and Expo**, 2023. Citado na página 37.
- IMAGINEDONE. **Aplicativo nativo: entenda o que é e conheça as principais vantagens para empresas**, Imaginedone, 2022.
- ITFORUM. **Objective-C deixa de fazer parte das 20 linguagens mais populares do índice Tiobe**. 2021. Disponível em: <<https://itforum.com.br/noticias/objective-c-deixa-de-fazer-parte-das-20-linguagens-mais-populares-do-indice-tiobe/>>.
- ITMAGINATION. **O Objective-C ainda é relevante em 2023 ou o Swift é a única escolha real? [ATUALIZADA]**. 2023. Disponível em: <<https://www.itmagination.com/blog/is-objective-c-still-relevant-in-2022-or-is-swift-the-only-real-choice>>.
- IUGU. **APIs: O que são? Quais os principais tipos? 2021**. Disponível em: <<https://www.iugu.com/blog/tipos-de-api>>.
- KUITUNEN, M. **Cross-Platform Mobile Application Development with React Native. Dissertação (B.S. thesis)**, 2019.
- LARROSSA, L. **O que é iOS: tudo que precisa saber sobre o sistema da Apple**. 2019. Disponível em: <<https://www.apptuts.net/tutorial/ipad/o-que-e-ios/>>.
- LECHETA, R. R. **Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. [S.l.]: Novatec Editora, 2013.

- LIMA, C. C. B. d. **Aplicativos móveis de interesse público: limites e possibilidades para a cidadania no brasil.** 2017.
- MADEINWEB. **Quais são as facilidades que um aplicativo traz aos seus clientes?** 2017. Disponível em: <<https://madeinweb.com.br/facilidades-aplicativo-clientes/>>.
- MANCHANDA, A. **The 7 Best Programming Languages to Write Develop Native Android Apps, Net Solutions,** 2022.
- MITCHELL, L. J. **Web services em php.** São Paulo: Novatec Editora, 2013.
- NASCIMENTO, C. R. d. L. **Manager RU.** Dissertação (B.S. thesis), 2021.
- ORACLE. **O que é Gerenciamento de Dados?** 2019. Disponível em: <<https://www.oracle.com/br/database/what-is-data-management/>>.
- PALTOGLOU, K.; ZAFEIRIS, V. E.; DIAMANTIDIS, N.; GIAKOUMAKIS, E. A. **Automated refactoring of legacy javascript code to es6 modules.** Journal of Systems and Software, Elsevier, v. 181, p. 111049, 2021.
- PHP, G. de Documentação do. **PHP: Documentação.** 2023. Disponível em: <<https://www.php.net/manual/en/intro-what-is.php>>.
- SARTORELI, C. E.; KUCHAUSKI, N. A. U. **Comparativo entre ios, android e windows phone.** ETIC-ENCONTRO DE INICIAÇÃO CIENTÍFICA-ISSN 21-76-8498, v. 9, n. 9, 2013.
- SELVARAJAH, K. et al. **Native apps versus web apps: Which is best for healthcare applications?** In: SPRINGER. Human-Computer Interaction. Applications and Services: 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II 15. [S.l.], 2013. p. 189–196.
- SILVA, M. M.; SANTOS, M. T. P. **Os paradigmas de desenvolvimento de aplicativos para aparelhos celulares.** Tecnologias, Infraestrutura e Software, Universidade de São Paulo, v. 3, n. 2, p. 162–170, 2014.
- SOUSA, L. A.; BRITO, L.; REIS, U. **Estudo comparativo entre as modalidades de desenvolvimento mobile: web app, híbrida e nativa,** SENAI/CIMATEC, 2014.
- STACKOVERFLOW. **Em maio de 2021, mais de 80.000 desenvolvedores nos contaram como aprendem e sobem de nível, quais ferramentas estão usando e o que desejam.** 2021. Disponível em: <<https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-language-want>>.
- TECHTARGET. **7 reasons why businesses need mobile apps.** 2020. Disponível em: <<https://www.techtarget.com/searchcustomerexperience/tip/7-reasons-why-businesses-need-mobile-apps>>.

TOSTES, R. N.; COSTA, R. M. de R. **Aplicações multiplataforma com react desenvolvimento de uma aplicação educacional**. ANALECTA-Centro Universitário Academia, v. 5, n. 5, 2020.

TREINAWEB. **TreinaWeb**. 2021. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-mvc>>.

UDS. **Aplicativos nativos, híbridos e PWAs: qual o melhor para sua empresa?**, uds, 2022.

WU, W. **React Native vs Flutter, Cross-platforms mobile application frameworks**, Metropolia Ammattikorkeakoulu, 2018.