

# Aplicativo de Eventos utilizando Flutter e Firebase

Alexandre Pereira Vega

Universidade Federal de Santa Catarina

Departamento das Engenharias

Blumenau, Brasil

Email: alexandre.xeno@gmail.com

Aléxei Felipe Paim

Universidade Federal de Santa Catarina

Departamento das Engenharias

Blumenau, Brasil

Email: alexei.felipe@grad.ufsc.br

Beatriz Picinin Pinheiro

Universidade Federal de Santa Catarina

Departamento das Engenharias

Blumenau, Brasil

Email: biappinheiro@gmail.com

Vinícius Dias Costa

Universidade Federal de Santa Catarina

Departamento das Engenharias

Blumenau, Brasil

Email: vinicius.d.c@grad.ufsc.br

**Resumo**—Os dispositivos móveis se estabeleceram como ferramentas essenciais tanto na vida profissional quanto no lazer, representando uma alternativa para diversas atividades, como gestão de carteira, agenda e comunicação. O desenvolvimento de aplicativos visa ampliar suas funcionalidades, abordando problemas recorrentes do cotidiano e se tornando modelos de negócios significativos. Diante da necessidade de gerenciar informações de eventos e coletar feedbacks de participantes, foi concebido o aplicativo Debrave. Este projeto inclui a implementação de práticas de usabilidade para assegurar uma navegação satisfatória e intuitiva na interface do aplicativo. Utilizando o Framework Flutter em conjunto com o serviço de Banco de Dados Firebase, a aplicação surge como uma solução que incorpora os conhecimentos adquiridos nas disciplinas de especialização do curso de Engenharia de Controle e Automação.

**Palavras chave**—Android, Aplicativo Móvel, Firebase, Flutter

**Abstract**—Mobile devices have become essential tools in both professional and leisure aspects, serving as alternatives for various activities such as wallet management, scheduling, and communication. The development of applications aims to enhance their functionalities, addressing recurring everyday problems and evolving into significant business models. In response to the need for managing information from large-scale events and collecting participant feedback, the Debrave application was developed. This project includes the implementation of usability practices to ensure a satisfactory and intuitive navigation experience within the app's interface. Using the Flutter framework in conjunction with the Firebase Database service, the application emerges as a solution that incorporates the knowledge gained from specialized courses in the field of Control and Automation Engineering.

**Index Terms**—Android, Firebase, Flutter, Mobile Application

## I. INTRODUÇÃO

A disciplina de projeto especializado tem como finalidade que os alunos do curso de Engenharia de Controle e Automação da UFSC realizem, em grupos de 3 a 4 pessoas, um projeto buscando integrar as disciplinas de especialização da graduação, sejam elas em mecatrônica, controle ou programação.

Para desenvolvimento da disciplina foi escolhido um desafio pelo grupo, o desenvolvimento de um aplicativo que resolvesse algumas problemáticas constantes no dia a dia das pessoas, a dificuldade de obter informações e avaliar eventos e também a grande quantidades de links e aplicativos que estão ligados a

um evento e poderiam estar em um só lugar facilitando a vida das pessoas.

De acordo com a Abrape,[1] o setor de eventos representa cerca de 4'%' do PIB do Brasil e gera 6,6'%' dos empregos. Para as empresas do setor é necessário saber o que leva um consumidor a ser atraído pelo evento. Isso pode determinar questões como preços de ingressos, atrações e local, logo, um local destinado a coleta de feedbacks de maneira segura e precisa seria fundamental. Do ponto de vista do consumidor, muitas vezes ele chega sem informações aos eventos, tendo uma discrepância da expectativa criada, ou então não tendo a possibilidade de desbravar o evento de uma maneira que favoreça sua experiência, logo, seria de extrema importância fornecer informações de maneira antecipada e de fácil acesso.

## II. REVISÃO BIBLIOGRÁFICAS

Eventos são frequentemente considerados a melhor opção para ajudar recém-chegados a encontrar oportunidades sociais, eles proporcionam um ambiente propício para conhecer novas pessoas e interagir em um contexto específico [2]. Enquanto isso, a adoção de *smartphones* é crescente, e em 2015, os aplicativos móveis atingiram uma receita de 41,1 bilhões de dólares [3]. O uso de smartphones e tablets ocupa 60% do tempo gasto em mídias digitais [4].

Há muitos estudos sobre segurança e privacidade em apps, bem como sobre o usuário e suas preferências [3]. Dispositivos móveis são as ferramentas mais convenientes e eficientes para obter dados .

Atualmente *Android* e *iOS* são os dois maiores sistemas operacionais móveis do mundo. O *Android* possui melhor velocidade de download, enquanto o *iOS* possui melhor desempenho em termos de latência. Em relação às medidas de *jitter* e perda de pacotes, a performance varia de acordo com o operador móvel [5].

Se tratando de aplicativos móveis, eles podem se tratar de aplicativos nativos ou *cross-plataform*. Um aplicativo móvel nativo é desenvolvido especificamente para um único sistema operacional, como *Android* ou *iOS*. Ele é projetado para aproveitar os recursos e funcionalidades específicas de cada plataforma [6], e tende a proporcionar melhor desempenho, uma boa experiência do usuário e total liberdade no uso

do aplicativo [7]. Além disso, a facilidade de *debug* e a facilidade de publicação são benefícios significativos para os desenvolvedores [8].

Por outro lado, uma aplicação *mobile cross-plataform* é desenvolvida para funcionar em várias plataformas, como *Android* e *iOS*, com um único conjunto de código, o que reduz significativamente o tempo necessário para desenvolver e manter o aplicativo [6]. No entanto, tendem a ser mais lentos do que os aplicativos nativos devido às camadas de abstração adicionais. Além disso, pode haver ausência de certas ferramentas específicas para cada plataforma [8].

Com sua segurança aprimorada, apoio do *Google* e recursos modernos, o *Kotlin* se tornou uma escolha popular para o desenvolvimento de aplicativos *Android*. Enquanto o *Java* permite que ocorram ponteiros nulos, o *Kotlin* elimina essa possibilidade, reduzindo a ocorrência de erros relacionados a *nulls*. Além disso, oferece recursos nativos que não são fornecidos nativamente no *Java*, que permitem que os desenvolvedores escrevam códigos mais modernos, concisos e expressivos. O *Kotlin* também oferece suporte a recursos como lambdas e funções de extensão, proporcionando uma programação mais eficiente [9].

O *Kotlin* demonstra ser superior em termos de linhas concisas de código e uso reduzido de dados, enquanto o *Java* se destaca em relação ao tempo de compilação. Com base nisso, é recomendado usar *Java* em projetos em que o tamanho do APK e o tempo de compilação e construção são prioridades, enquanto o *Kotlin* é mais adequado para projetos que valorizam a redução de *bugs*, código conciso e desenvolvimento rápido. Vale ressaltar que uma desvantagem do *Java* é que ele é uma linguagem mais verbosa, o que significa que mesmo em tarefas simples, é necessária uma quantidade significativa de código [10]. Quanto à comparação entre *Flutter*, *React* e *Kotlin*, é importante mencionar que o tamanho do aplicativo desenvolvido com *Flutter* é geralmente maior em comparação com o desenvolvido nativamente. Além disso, foi observado que o uso da CPU no *React* foi significativamente maior, enquanto o *Kotlin* teve um desempenho um pouco melhor nesse aspecto [8].

Quando se trata de *React* e *Flutter*, existem diferenças em termos de espaço e tempo de carregamento. Em relação ao espaço, o *React* tende a exigir mais espaço de armazenamento do que o *Flutter* devido às suas dependências e bibliotecas extras. Quanto ao tempo de carregamento, o *Flutter* possui uma vantagem com o recurso chamado "*Hot Reload*", que permite visualizar as alterações em tempo real sem reiniciar o aplicativo, tornando o processo mais rápido. No entanto, o *React* pode levar mais tempo para carregar as alterações usando o "*Hot Load*" [6].

A popularidade do *Flutter* se deve à sua capacidade de resolver o problema de desempenho no desenvolvimento *cross-plataform*. Além disso, as linhas de código necessárias para desenvolver em *Flutter* são menores em comparação com o desenvolvimento em *Java* [11]. *Flutter* é grátis e *open-source User Interface Software* da *Google*. Utilizado para desenvolvimento multiplataforma *Android*, *iOS*, *Linux*, *Mac*, *Windows*, *Google Fuchsia* e *web* [12].

O *Flutter* possui algumas características que o tornam uma escolha popular para o desenvolvimento de aplicativos. Ele oferece um avanço simples e rápido, utilizando um dialeto de aperfeiçoamento simples. Além disso, o *Flutter* é uma plataforma *cross-plataform* eficiente, permitindo que os desenvolvedores criem aplicativos que funcionem em várias plataformas. Ele também é conhecido por lidar de forma eficiente com usuários finais e suportar novas *features*. O *Flutter* possui a funcionalidade *Gesturu Detector* e é eficiente na correção de *bugs* relacionados a ele [13].

### III. PROBLEMA

Cada dia mais surgem eventos e festas voltadas para diversos grupos de pessoas, trazendo uma variedade muito grande dos mesmos. Dentro desses eventos, cada vez mais contamos com atrações, marcas e patrocinadores buscando se promover, devido a toda globalização que temos no mundo as empresas sabem da importância de um bom marketing envolvido em seu negócio. Porém, justamente por esse crescimento exponencial, relacionado a variedade de marcas que encontramos expostas em um evento, as pessoas muitas vezes deixam de explorar melhor as marcas de seu interesse e acabam não aproveitando muito o local que visitaram, diminuindo consideravelmente a qualidade de sua experiência. Outro fator impactante na vida dos clientes é a impossibilidade de ver ou realizar avaliações, o que impacta consecutivamente as marcas que perdem a oportunidade de coletar feedbacks, algo considerado extremamente importante atualmente.

Outra problemática presente na vida das pessoas que participam destes eventos é que elas já estão saturadas de baixar diversos aplicativos em seu celular para o dia a dia, porém hoje existem aplicativos separados para comprar ingressos, links variados para conhecer as empresas e atrações, logo dificulta a experiência do cliente. Pensando nessas problemáticas, foi desenvolvido um aplicativo para centralizar todas as aplicações em um único local, agregando funcionalidades, que serão abordadas ao decorrer do presente trabalho, para coleta de feedback dos usuários e que tenha uma interface que respeite as heurísticas de usabilidade. O nome escolhido para o app é Desbrave.

### IV. FUNCIONALIDADE

O potencial de funcionalidades presentes no aplicativo são inúmeras, porém o foco inicial foi focar em uma abordagem que ainda não é muito popular em eventos, embora existam. O foco principal do Desbrave em sua etapa inicial foi atrelar a experiência do cliente a feedbacks, informação e divulgação. A sua funcionalidade até o presente momento se baseia em possibilitar que as pessoas pesquisem e explorem o evento antes mesmo de estar no local.

O primeiro passo é o usuário abrir o aplicativo e ver quais serão os eventos, festas, shows, entre outros entretenimentos disponíveis no app. Após isso, ele consegue embarcar no perfil do evento e desbravar tudo que tem para oferecer descrição do evento, curtidas de outros usuários e acessar a página de comentários do evento. A seção de descrição reúne as principais informações para gerar o interesse do usuário, a aba

de curtidas e comentários expõe observações e satisfações dos demais membros do app.

## V. TRABALHOS FUTUROS

Explorando tudo que é desejado e foi idealizado, que só não é possível até o presente momento por conta da curva de aprendizado que demanda o desenvolvimento de um aplicativo desenvolvido em flutter, o Desbrave tem potencial e intenção de suprir todas as problemáticas citadas anteriormente, sendo robustecido, aprimorado e possibilitando novas ações e interações como: compra de ingressos, salvar e compartilhar eventos como em uma rede social, filtrar os eventos de acordo com o perfil das pessoas, através de um algoritmo, comercialização das divulgações e propagandas, possibilidade do usuário programar todos os locais dentro do próprio app e compra de itens dentro do próprio aplicativo, diminuindo filas nos locais.

Todas essas funcionalidades têm um único intuito, agregar valor no evento, ser um aliado das empresas na promoção das mesmas e proporcionar uma experiência única para o cliente.

## VI. OBJETIVOS

De início foram estipulados 2 objetivos gerais e 3 objetivos específicos a serem cumpridos

- Objetivos gerais:
  - Protótipo Funcional com funcionalidades básicas:
    - \* Capacidade de atender especificações como possibilidade de cadastro de usuários, uma página inicial do aplicativo e a possibilidade de realizar comentários e realizar avaliações.
  - Entrega da Interface do usuário:
    - \* Um espaço prático para os usuários interagirem e explorar o Desbrave.
- Objetivos específicos:
  - Um aplicativo com Interface apropriada às heurísticas de Usabilidade:
    - \* Totalmente voltado à importância de proporcionar uma boa experiência ao usuário através de uma interface amigável, seguindo as recomendações de usabilidade criadas por pesquisadores da área de design, buscando características positivas em interfaces, a fim de garantir a boa experiência de uso.
  - Funcionalidade de cadastrar perfil do usuário:
    - \* Cada pessoa que ingresse no aplicativo deve fazer um cadastro se identificando e tendo sua conta registrada e protegida por senha para navegar em seu perfil.
  - Capacidade de armazenar avaliações e feedbacks dos usuários:
    - \* Através de um banco de dados possibilitar que as informações fiquem registradas para todos os usuários visualizarem.

## VII. DESENVOLVIMENTO

### A. Design

Tanto o nome, design e cores do aplicativo foram pensados buscando proporcionar uma experiência agradável ao usuário,

uma vez que a partir do momento em que um aplicativo está apropriado as heurísticas de usabilidade, em relação a estética e detalhes minimalistas. De acordo com Jakob Nielsen, responsável por desenvolver as heurísticas de habilidade, a chance dos usuários continuarem a utilizar o produto desenvolvido sem se cansarem após um determinado período de tela, após certo tempo consumindo, diminui consideravelmente. Partindo um pouco para a escolha das cores, o azul foi escolhido como cor predominante, pois segundo a psicologia das cores passa ao usuário confiança, força, credibilidade e profissionalismo, além de carregar um senso de inovação e segurança e ser a cor mais utilizada em projetos UI (criação de uma interface amigável) e UX (voltado para a maneira como o usuário vive o uso de um produto, entendendo os comportamentos e emoções dessas pessoas no uso do produto). A UI do projeto foi desenvolvida 100

Para a escolha do nome foi realizado um Brainstorming, organizando ideias, palavras chave e aperfeiçoando-as. Após isso foram feitas pesquisas de nomes em comum, pois como foi descrito anteriormente, existem inúmeros aplicativos no mundo e começar do zero, com um nome muito comum não é o mais indicado a se fazer. Outro ponto importante na escolha do nome foi a facilidade de memorização e facilidade de pronunciar, favorecendo sua popularização, além de claro um bom slogan, que seja simples e faça sentido com as funcionalidades e finalidades do app, no caso: Desbrave, o guia para seu evento.

### B. Desenvolvimento em Flutter

Flutter tem ganhado popularidade entre os desenvolvedores por causa da sua capacidade de criar aplicativos nativos com rapidez, mantendo a performance e a aparência consistente em diversas plataformas.[14] Por se tratar de um aplicativo destinado a funcionar em múltiplas plataformas, optou-se por utilizar o Framework Flutter, pois ele garante essa funcionalidade cross-plataform entre Android e iOS [15]. O Flutter é um framework de UI (Interface de Usuário) que utiliza a linguagem Dart desenvolvida pela Google.

Como o Flutter é um poderoso framework voltado para o desenvolvimento mobile, ele nos permite utilizar os chamados widgets, facilitando a criação e customização de várias interfaces interativas. A utilização de widgets neste trabalho está diretamente ligada ao desenvolvimento visual, desde o botão mais básico até listas, layouts e elementos interativos, como ilustra a Figura 1 que evidencia a forma de utilização de widgets neste projeto.

A utilização de widget possibilitou a combinação e customização de tal forma que a interface gráfica ficou o mais perto possível do que havia sido almejado inicialmente.

Durante o desenvolvimento desse projeto todos os códigos foram diretamente compilados dentro de um ambiente Android o que permitiu que problemas fossem detectados de forma rápida. Figura 2. Assim proporcionou um desenvolvimento mais dinâmico e eficaz na implementação deste projeto.

### C. Desenvolvimento em Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos para dispositivos móveis e web. Ele oferece uma variedade de serviços e ferramentas para ajudar os desenvolvedores a

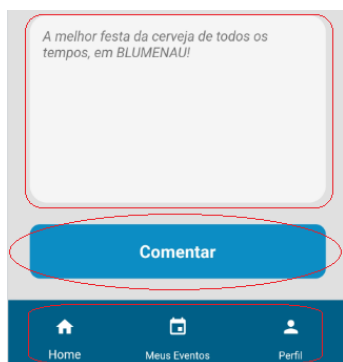


Fig. 1. Exemplo de como widgets são aplicados

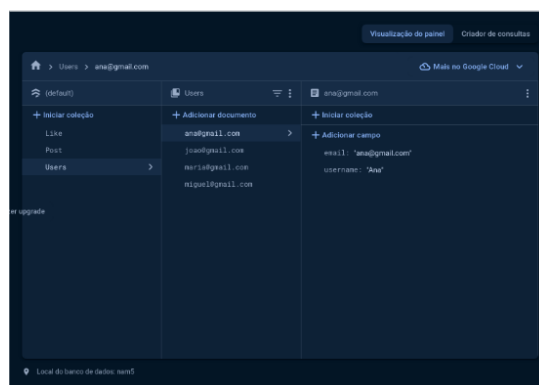


Fig. 3. Banco de dados Firebase

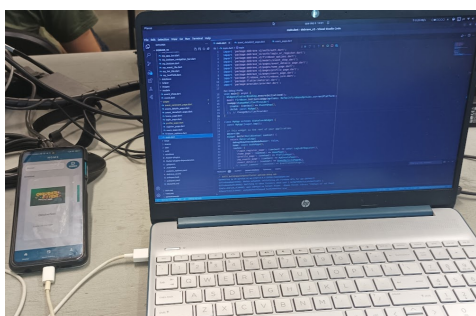


Fig. 2. Exemplificação do método de compilação

criar aplicativos de alta qualidade, escaláveis e com recursos avançados, sem precisar se preocupar com a complexidade da infraestrutura de back-end [16].

Neste projeto, a escolha do Firebase foi fundamental para a integração entre o front-end e o back-end desenvolvidos em Flutter. A compatibilidade entre essas duas plataformas, ambas pertencentes ao Google, proporcionou uma integração mais fluida.

Um dos pontos relevantes que influenciaram a escolha desta plataforma foi a necessidade de um banco de dados capaz de armazenar informações de cadastro e avaliações de eventos. O Firebase oferece um banco de dados NoSQL em tempo real baseado na nuvem, permitindo o armazenamento e a sincronização de dados entre os clientes em tempo real[16].

No desenvolvimento do banco de dados, utilizou-se a ferramenta disponibilizada pelo Firebase, chamada Firestore, que implementa a comunicação da aplicação móvel com o banco de dados. A imagem na Figura3 ilustra o funcionamento desse processo, demonstrando as informações inseridas ou requeridas, as quais foram explicadas na seção de design.

#### D. Usabilidade

A usabilidade foi tratada com significativa importância, pois é essencial para a efetiva propagação entre os futuros usuários. Uma forma de caracterizar a usabilidade dos sistemas é avaliar o modelo de interface e interação desses sistemas, podendo ser conduzida através da aplicação de métodos de avaliação. Dentre os métodos mais difundidos, destaca-se a Avaliação Heurística (AH), um método empírico que julga a interface de acordo com um conjunto de heurísticas (por exemplo, Consistência e Padronização e Design Estético e Minimalista), que buscam aprimorar a usabilidade da interface e interação[17].

Embora originalmente tenham sido propostas para usabilidade em interfaces de websites, muitas dessas heurísticas também são aplicáveis ao desenvolvimento de aplicativos móveis[17]. Desde o projeto do aplicativo até o desenvolvimento, a preocupação em atender a essas heurísticas foi levada a sério. Durante o desenvolvimento do projeto, foi trabalhado para que a Visibilidade do Estado do Sistema, a Correspondência entre o Sistema e o Mundo Real, e o Design Estético e Minimalista fossem notáveis na experiência do usuário, o que nos levou aos designs apresentados nas Figura de resultados que atendem a esses requisitos.

Outros pontos, como a Prevenção de Erros e a Consistência e Padrões, não são tão visíveis, mas estão diretamente incorporados no código que sustenta a aplicação.

## VIII. RESULTADOS

Dos objetivos listados, todos foram alcançados, cada um em sua determinada etapa do projeto, a começar pelo cadastro do perfil do usuário, como é possível observar na imagem 4.

Após isso, o próximo passo foi a criação da interface do usuário. A plataforma mantém as heurísticas de usabilidade, como é possível visualizar na imagem 5. A home do aplicativo, principal página para o usuário, está em pleno funcionamento e com as devidas heurísticas de usabilidade aplicadas.



Fig. 4. imagem de cadastro de usuários

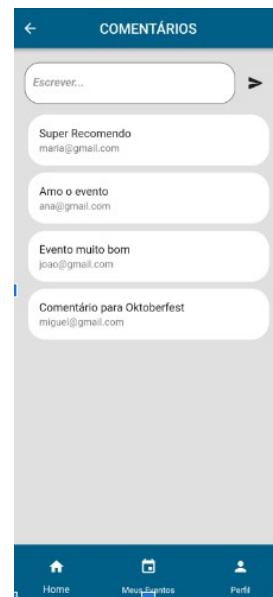


Fig. 6. banco de dados dos comentários

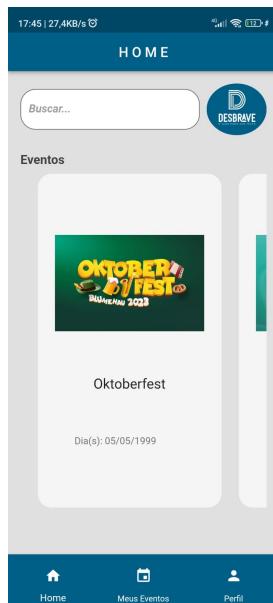


Fig. 5. página principal de navegação

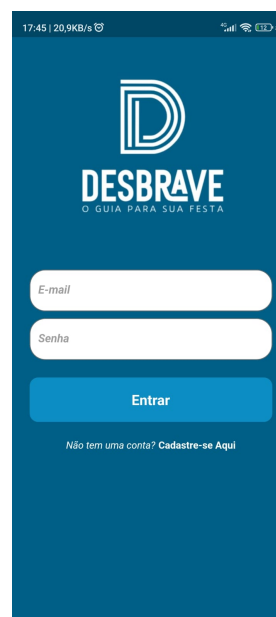


Fig. 7. Página de login para usuários cadastrados

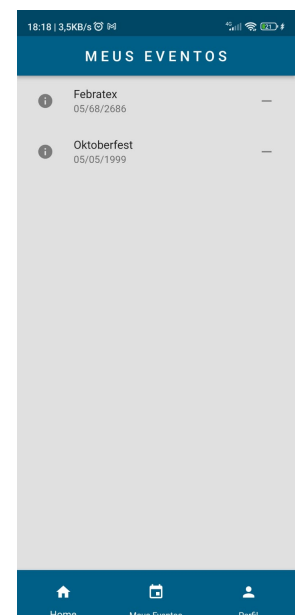


Fig. 8. Página "Meus Eventos"

Por fim, o último objetivo era que fosse possível utilizar um banco de dados para armazenar as avaliações e feedbacks dos usuários, algo que é possível observar na imagem6

Descrevendo como funciona o aplicativo, durante o tempo de desenvolvimento foi possível criar toda logo e design do projeto. Após isso iniciar o frontend e backend do aplicativo, comunicando o mesmo com o banco de dados no Firebase. Abaixo é possível algumas imagens dos marcos alcançados no projeto, com a tela de login para usuários já cadastrados, os eventos salvos e a página do evento em si.



Fig. 9. Página do Evento

## IX. DESAFIOS

Os autores enfrentaram desafios significativos devido à falta de experiência no desenvolvimento de aplicativos para Android e à não utilização do *Framework* Flutter e do serviço Firebase, ambos fornecidos pela Google. Para dominar essas ferramentas, foi crucial realizar um planejamento de tempo adequado, assegurando a conclusão oportuna do aplicativo Debrave. Apesar de o Flutter não apresentar uma curva de aprendizado acentuada, a familiarização com o *Widget*, a principal funcionalidade do *Framework*, ainda se fez necessária. A linguagem Dart (utilizada pelo Flutter) representou o primeiro contato da equipe com uma Linguagem Orientada por Objetos, uma habilidade possuída apenas por um dos autores, o que exigiu um investimento adicional de tempo em estudo.

Os dispositivos pessoais dos autores não atendiam aos requisitos para suportar o desenvolvimento, com exceção de um dos membros. Essa limitação foi contornada ao utilizar os dispositivos fornecidos pela faculdade.

A lógica de desenvolvimento da interface do usuário revelou-se distinta, gerando estranheza e desconforto aos autores até que se habituassem ao processo. Devido à falta de experiência, a construção da UI não atendeu a todas as características de *User Experience*, tornando necessário um retrabalho futuro para criar uma interface mais acessível. A interface inicialmente planejada não pôde ser produzida, ocorrendo modificações ao longo do projeto conforme as habilidades eram desenvolvidas.

## X. CONCLUSÃO

Era esperado na disciplina que fosse colocado em prática aprendizados obtidos em matérias especializadas e no fim esse foi o real acontecimento, o grupo conseguiu colocar em prática disciplinas como Sistemas Computacionais para Controle, Programação Orientada a Objetos e Sistemas para Automação.

Os objetivos foram cumpridos com êxito, conseguindo executar o que foi permitido. O desenvolvimento do aplicativo, juntamente com as pesquisas, serviu para ressaltar ainda mais a importância do uso de apps no nosso dia a dia.

O flutter se mostrou uma excelente ferramenta para o desenvolvimento e sua curva de aprendizado atrelada a quantidade e facilidade de informações usadas para o desenvolvimento foi fundamental para cumprir com o esperado. Também é importante ressaltar a importância da comunicação com o banco de dados, garantindo segurança e praticidade. Além da relevância da usabilidade, tornando o aplicativo em uma ferramenta útil e prática para o consumidor.

## REFERÊNCIAS

- [1] ABRAPE - Associação Brasileira de Pesquisa de Energia.  
url<https://abrape.com.br>. Acesso em: 17 de outubro de 2021.
- [2] Rogerio A. Bordini e Oliver Korn. "A Mobile App to Combat Loneliness and Social Isolation Among University Students." Em: *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '20. Oldenburg, Germany: Association for Computing Machinery, 2021. ISBN: 9781450380522. DOI: 10.1145/3406324.3409260. URL: <https://doi.org/10.1145/3406324.3409260>.
- [3] Haoyu Wang, Zhe Liu, Yao Guo, Xiangqun Chen, Miao Zhang, Guoai Xu e Jason Hong. "An Explorative Study of the Mobile App Ecosystem from App Developers' Perspective". Em: *Proceedings of the 26th International Conference on World Wide Web* (2017).
- [4] Ivano Malavolta. "Web-Based Hybrid Mobile Apps: State of the Practice and Research Opportunities". Em: *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. 2016, pp. 241–242. DOI: 10.1145/2897073.2897133.
- [5] Engin Zeydan. "Android vs. IOS: a comparative analysis over mobile operator infrastructures based on crowd-sourced mobile dataset". Em: *Telecommunication Systems* 78.3 (ago. de 2021), pp. 405–419. ISSN: 1572-9451. DOI: 10.1007/s11235-021-00820-y. URL: <http://dx.doi.org/10.1007/s11235-021-00820-y>.
- [6] Kamal Kishore, Shanu Khare, Vaibhav Uniyal e Sahil Verma. "Performance and stability Comparison of React and Flutter: Cross-platform Application Development". Em: *2022 International Conference on Cyber Resilience (ICCR)*. 2022, pp. 1–4. DOI: 10.1109/ICCR56254.2022.9996039.
- [7] Anil Patidar e Ugrasen Suman. "Towards Analyzing Mobile App Characteristics for Mobile Software Development". Em: *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2021, pp. 786–790.
- [8] Bhawna Suri, Shweta Taneja, Isha Bhanot, Himanshi Sharma e Aanchal Raj. "Cross-Platform Empirical Analysis of Mobile Application Development Frameworks: Kotlin, React Native and Flutter". Em: *Proceedings of the 4th International Conference on Informa-*

tion Management & Machine Intelligence. ICIMMI '22. Jaipur, India: Association for Computing Machinery, 2023. ISBN: 9781450399937. DOI: 10.1145/3590837.3590897. URL: <https://doi.org/10.1145/3590837.3590897>.

- [9] Matias Martinez e Bruno Gois Mateus. “Why Did Developers Migrate Android Applications From Java to Kotlin?” Em: *IEEE Transactions on Software Engineering* 48.11 (2022), pp. 4521–4534. DOI: 10.1109/TSE.2021.3120367.
- [10] Bambang Purnomosidi Dwi Putranto, Robertus Saptoto, Ovandry Chandra Jakaria e Widyastuti Andriyani. “A Comparative Study of Java and Kotlin for Android Mobile Application Development”. Em: *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. 2020, pp. 383–388. DOI: 10.1109/ISRITI51436.2020.9315483.
- [11] Yoonsik Cheon e Carlos Chavez. “Converting Android Native Apps to Flutter Cross-Platform Apps”. Em: *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2021, pp. 1898–1904. DOI: 10.1109/CSCI54926.2021.00355.
- [12] Baivab Maulik, Aditi P Nayak, Sanjana U, Simmi Alok e Divyaprabha K N. “Design and Implementation of Virtual Tour Guide App”. Em: *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*. 2022, pp. 1–6. DOI: 10.1109/ICACTA54488.2022.9752804.
- [13] K Nagaraj, B Prabakaran e M.O Ramkumar. “Application Development for a Project using Flutter”. Em: *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*. 2022, pp. 947–951. DOI: 10.1109/ICOSEC54921.2022.9951938.
- [14] Google. *Flutter*. <https://flutter.dev/>. 18/11/2023.
- [15] “Convertendo aplicativos nativos do Android em aplicativos multiplataforma Flutter”. Em: *2021 Conferência Internacional sobre Ciência Computacional e Inteligência Computacional (CSCI)*. DOI: 10.1109/CSCI54926.2021.00355.
- [16] Google. *Firebase*. <https://firebase.google.com/?hl=pt>. acesso em 5 dezembro, 2023.
- [17] Glívia Barbosa, Erica Oliveira e Deborah D’Carlo. “Usabilidade em aplicativos móveis educacionais: Um conjunto de heurísticas para avaliação”. Em: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. Vol. 27. 1. 2016, p. 777.