UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RAFAEL MAESTRO DOS SANTOS

SISTEMA DE REGISTRO DE PRESENÇA POR GEOLOCALIZAÇÃO

BAURU
12/2024

RAFAEL MAESTRO DOS SANTOS

SISTEMA DE REGISTRO DE PRESENÇA POR GEOLOCALIZAÇÃO

Trabalho de Conclusão de Curso do Curso de Sistemas de Informação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Higor Amario de Souza

BAURU
12/2024

D724s

dos Santos, Rafael Maestro
Sistema de registro de presença por geolocalização / Rafael Maestro dos Santos. -- Bauru, 2024
73 f. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Higor A. de Souza

1. Software. 2. Mobile. 3. Geolocalização. 4. Presença. 5. Eventos. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Resumo

Este trabalho apresenta o desenvolvimento de um aplicativo móvel, intitulado Sistema de Registro de Presença por Geolocalização, que visa facilitar o controle de presença em eventos, inicialmente focando no ambiente de sala de aula, mas com potencial de expansão para diversos outros domínios e espaços de encontro. A proposta central do aplicativo é utilizar a tecnologia de geolocalização para simplificar o processo de registro de presença dos participantes, proporcionando uma experiência rápida, precisa e segura, através de dispositivos móveis. Ao aliar a praticidade da geolocalização com a flexibilidade de um aplicativo móvel, espera se que o Sistema de Registro de Presença por Geolocalização proporcione uma experiência aprimorada tanto para os organizadores de eventos (professores) quanto para os participantes (alunos), contribuindo para uma gestão mais eficiente e transparente da presença em eventos de todos os tipos.

Palavras-chave: Software, Mobile, Geolocalização, Sala de Aula, Eventos, Presença.

Abstract

This work presents the development of a mobile application, titled "Geolocation Attendance Registration System", aimed at facilitating attendance control in events, initially focusing on the classroom environment, but with potential for expansion to various other domains and meeting spaces. The central proposal of the application is to use geolocation technology to simplify the process of registering the presence of participants, providing a fast, accurate, and secure experience through mobile devices. By combining the practicality of geolocation with the flexibility of a mobile application, it is expected that the Geolocation Attendance Registration System will provide an enhanced experience for both event organizers (teachers) and participants (students), contributing to a more efficient and transparent management of attendance at events of all kinds.

Keywords: Software, Mobile, Geolocation, Classroom, Events, Attendance.

Lista de figuras

Figura 1 – Diagrama de componentes 16
Figura 2 – Diagrama de Atividades: Acessar o Aplicativo 20
Figura 3 – Diagrama de Atividades: Criar Novo Evento 21
Figura 4 – Diagrama de Atividades: Realizar Evento (Sem Intervalo) 22
Figura 5 – Diagrama de Atividades: Realizar Evento (Com Intervalo) 23
Figura 6 – Diagrama de Casos de Uso 24
Figura 7 – Diagrama de Classes 32
Figura 8 – Modelo Entidade Relacionamento (MER) 33
Figura 9 – Tela de Login 37
Figura 10 – Tela de Login - Validação Token 38
Figura 11 – Tela de Cadastro (1/2) 39
Figura 12 – Tela de Cadastro (2/2) 40
Figura 13 – Tela Inicial 41
Figura 14 – Tela de “Meus Eventos” (Estado 1) 43
Figura 15 – Tela de “Meus Eventos” (Estado 2) 44
Figura 16 – Tela de “Meus Eventos” (Estado 3) 45
Figura 17 – Tela de Check-in (Caso 1) 46
Figura 18 – Tela de Evento (Convidado) 47
Figura 19 – Tela de Evento (Convidado) - Mensagem Check-out 48
Figura 20 – Tela de Evento (Organizador 1/2) 49
Figura 21 – Tela de Evento (Organizador 2/2) 50
Figura 22 – Tela Histórico de Eventos 51
Figura 23 – Tela Histórico de Eventos (Cartão Selecionado) 52
Figura 24 – Tela Histórico de Eventos (Visualização de Convidados Presentes) 53
Figura 25 – Tela de Cadastro de Eventos (Etapa 1 - (1/2)) 54
Figura 26 – Tela de Cadastro de Eventos (Etapa 1 - (2/2)) 55
Figura 27 – Tela de Cadastro de Eventos (Etapa 2) 56
Figura 28 – Tela de Cadastro de Eventos (Etapa 3 - Mensagem) 57
Figura 29 – Tela de Cadastro de Eventos (Etapa 3 - Convidados) 58
Figura 30 – Tela de Cadastro de Eventos (Etapa 3 - Mensagem) 59
Figura 31 – Tela de Perfil 60
Figura 32 – Pirâmide de Testes 61
Figura 33 – Testes Caixa-Branca x Testes Caixa-Preta 62
Figura 34 – Unidade de Software 63
Figura 35 – Caso de teste utilizando Jest 64
Figura 36 – Testes Exploratórios 66

Lista de tabelas

Tabela 1 – Comparativo entre as soluções 15

Lista de abreviaturas e siglas

API	Interface de Programação de Aplicação
BD	Banco de Dados
ORM	Object-Relational Mapping (Mapeamento Objeto-Relacional)
TCC	Trabalho de Conclusão de Curso
UML	Unified Modeling Language (Linguagem Unificada de Modelagem)

Sumário

1 INTRODUÇÃO 10

2 SOLUÇÕES EXISTENTES 12
2.1 Softwares de Controle de Ponto 12
2.2 Registro de Presença em Eventos 12
2.3 Softwares de Apoio no Ambiente Escolar 13
2.4 Marketing de Geolocalização 13

3 DESCRIÇÃO DO PROJETO 16
3.1 Aplicativo Mobile 16
3.2 API Back-End 17
3.3 Banco de Dados 18

4 ESPECIFICAÇÃO 19
4.1 Atividades 19
4.2 Casos de Uso 23
4.2.1 Diagrama de Casos de Uso 24
4.2.2 Descrição dos Casos de Uso 24
4.3 Classes 31
4.4 Modelo Entidade-Relacionamento 32

5 TECNOLOGIAS 34
5.1 Aplicativo Mobile 34
5.2 API Back-End e Banco de Dados 34

6 INTERFACE GRÁFICA 36
6.1 Login 36
6.2 Cadastro 38
6.3 Inicial 40
6.4 Meus Eventos 42
6.5 Check-in 45
6.6 Evento (Convidado) 46
6.7 Evento (Organizador) 48
6.8 Histórico de Eventos 50
6.9 Cadastro de Eventos 53
6.10 Perfil 59

7 TESTES 61
7.1 Testes de Unidade 62
7.2 Testes de Integração 64
7.3 Testes Exploratórios 65

8 CONCLUSÃO 67
8.1 Desafios 67
8.2 Resultados 67
8.3 Trabalhos Futuros 68

REFERÊNCIAS 71

1 Introdução

O contínuo e marcante progresso da tecnologia da informação tem impulsionado transformações significativas em diversos contextos, incluindo o ambiente educacional. A expansão da internet tem viabilizado o desenvolvimento de novos sistemas, promovendo maior eficácia nas atividades cotidianas. Entre essas transformações, destaca-se a gestão de presença, que evoluiu dos métodos tradicionais de "papel e caneta"para soluções baseadas em software (TOTVS, 2020). Essa mudança permitiu às instituições educacionais um controle mais eficiente dos dados, possibilitando tomadas de decisão mais precisas e fundamentadas.

Entretanto, a maioria desses softwares é projetada para uso em computadores desktop, que são, na maioria dos casos, menos acessíveis do que os dispositivos móveis, tanto em termos de custo quanto de usabilidade. Dados da CETIC.BR (2023) indicam que, em 2023, 84% da população brasileira utilizou a internet, sendo que 58% acessaram exclusivamente por celulares. Esses números evidenciam a importância de adaptar ferramentas tecnológicas para plataformas móveis, garantindo acessibilidade e facilidade de uso.

No contexto educacional, o registro de presença eficiente vai além de uma mera formalidade, impactando diretamente na gestão acadêmica e na alocação de recursos. A ausência de um sistema adequado pode gerar problemas para diversos *stakeholders*, como:

- Estudantes: Registros imprecisos ou ausentes podem prejudicar a avaliação da participação e o desempenho acadêmico.
- Professores: O monitoramento manual consome tempo que poderia ser dedicado à preparação de aulas ou à interação com os alunos.
- Administradores de instituições de ensino: A falta de precisão no controle de presença dificulta o planejamento, a conformidade com regulamentações e a alocação eficiente de recursos.

A solução apresentada por este trabalho é um aplicativo móvel que utiliza tecnologia de geolocalização para simplificar e agilizar o registro de presença, com foco inicial no ambiente escolar. O aplicativo permitirá que os alunos registrem suas presenças de forma rápida e precisa utilizando seus dispositivos móveis, enquanto os professores terão uma plataforma para criar e gerenciar aulas de maneira eficaz. A geolocalização proporciona segurança e confiabilidade no registro, reduzindo erros e fraudes. Essa abordagem beneficia economicamente as instituições, automatizando processos, otimizando tempo e melhorando a alocação de recursos.

Além de ser portátil e acessível, a tecnologia móvel oferece conectividade constante via redes de dados móveis (3G/4G/5G), ampliando ainda mais sua aplicabilidade em ambientes

diversos. No contexto educacional, essa solução garante registros imediatos e integrados aos sistemas acadêmicos, melhorando tanto a experiência dos alunos quanto a eficiência dos professores.

Por fim, este trabalho está estruturado em capítulos que abrangem desde o detalhamento do problema e das soluções existentes no mercado, até as especificações do sistema, tecnologias utilizadas, apresentação da interface gráfica desenvolvida e as estratégias de teste de validacão. O Capítulo 2 apresenta as soluções já existentes no mercado para problemas similares. No Capítulo 3, estão descritas todas as unidades que compõem o projeto. No Capítulo 4 estão descritas as especificações do sistema desenvolvido além de diagramas UML. No Capítulo 5 estão descritas as tecnologias utilizadas diretamente e indiretamente para o desenvolvimento do projeto. No Capítulo 6 são apresentadas as telas que compõem a interface gráfica do protótipo. No Capítulo 7 são apresentadas as estratégias de validação e testes realizados, além dos resultados obtidos. No Capítulo 8 estão relacionadas as conclusões do projeto quanto ao alcance dos objetivos e, por último, as referências utilizadas.

2 Soluções Existentes

Neste capítulo, será realizada uma exploração de algumas soluções e técnicas existentes que, embora não correspondam exatamente à proposta em questão, compartilham conceitos ou tecnologias semelhantes. Serão discutidos especificamente softwares de controle de ponto que utilizam geolocalização, software de registro de presença em eventos, softwares de apoio no ambiente escolar e o conceito de marketing de geolocalização. A análise dessas soluções proporcionará uma compreensão mais aprofundada do cenário atual e identificará oportunidades para a proposta de um aplicativo para controle de presença através de geolocalização.

2.1 Softwares de Controle de Ponto

Os softwares de controle de ponto que utilizam geolocalização são uma inovação tecnológica que tem transformado a maneira como as empresas gerenciam a jornada de trabalho de seus colaboradores. Esses sistemas permitem que os funcionários registrem o início e o fim de sua jornada de trabalho diretamente de seus dispositivos móveis, e a localização geográfica do registro é automaticamente capturada pelo sistema.

Essa tecnologia é particularmente útil para empresas que possuem equipes remotas ou que realizam trabalho externo. Com o controle de ponto por geolocalização, é possível garantir que o colaborador está no local de trabalho no momento do registro do ponto.

Alguns exemplos de softwares que oferecem essa funcionalidade são:

- Factorial : Este software oferece uma solução completa de gestão de RH, incluindo controle de ponto com geolocalização. Ele permite que os colaboradores registrem o ponto diretamente do aplicativo móvel, e a localização é automaticamente registrada. (FACTORIAL, 2024)
- Simpax.Geoloc.Multi: A Simpax oferece um sistema de controle de ponto que também utiliza a geolocalização. Além disso, o sistema oferece outras funcionalidades, como gestão de folha de pagamento e controle de benefícios. (SIMPAX.GEOLOC.MULTI, 2024)

2.2 Registro de Presença em Eventos

Atualmente existem algumas opções de softwares que realizam registro de presença em evento no mercado, desde softwares de empresas conhecidas nesse segmento até empresas governamentais, como a Sympla e Serpro, respectivamente.

A plataforma Sympla oferece uma solução de controle de entrada e registro de presença em eventos. Seu aplicativo permite o registro dos participantes por meio da leitura automática de QR Codes. Mas não possui funcionalidades relacionadas à geolocalização (SYMPLA, 2024).

Já a Serpro, possui uma solução chamada "Check-in Serpro", onde o usuário pode utilizar de forma gratuita semelhante ao Sympla, ou seja, realizando a leitura de QR Codes (SERPRO, 2024).

Em resumo, as duas soluções apresentadas, sendo a primeira via assinatura e a segunda gratuita, tratam de questões semelhantes às abordadas na presente proposta, diferenciando-se pela utilização da tecnologia de geolocalização, e gratuidade, visando praticidade para todos os perfis de usuários e segurança.

2.3 Softwares de Apoio no Ambiente Escolar

Já no contexto de ambientes de ensino, existem alguns softwares que trabalham com controle de frequência. A TOTVS, uma empresa brasileira especializada em tecnologia e desenvolvimento de soluções empresariais (TOTVS, 2024b) possui uma solução chamada TOTVS Educacional , que trabalha de várias formas, desde leitura biométrica para registro de presença, até soluções mais simples como aplicativos móveis integrados aos sistemas utilizados pela instituição (TOTVS, 2024a).

Apesar de possuir várias funcionalidades e maneiras de se trabalhar com controle de presença, os softwares da TOTVS Educacional não possuem funcionalidades ligadas a geolocalização.

2.4 Marketing de Geolocalização

O marketing de geolocalização é uma técnica que utiliza a localização geográfica de um indivíduo para entregar conteúdo relevante e personalizado em dispositivos móveis. Ele permite que as empresas alcancem potenciais clientes de maneira mais pessoal e criem um vínculo mais próximo com eles. A geolocalização é ativada por meio de aplicativos de mapeamento e redes sociais, permitindo que as empresas entreguem anúncios e ofertas personalizadas em tempo real, baseados na localização do usuário. Além disso, o marketing de geolocalização também pode ser usado para rastrear o comportamento do consumidor e entender melhor os hábitos de compra e locais frequentados pelo público-alvo(HELLOMIDIA, 2023).

A partir da análise das soluções e técnicas mencionadas neste capítulo, é possível observar alguns pontos relevantes:

Em relação aos softwares de controle de ponto que utilizam geolocalização, percebe-se que eles compartilham um conceito semelhante com a proposta deste trabalho. Esses softwares buscam facilitar o controle de presença e a gestão da jornada de trabalho, oferecendo praticidade tanto para os gestores quanto para os colaboradores. Embora esses softwares sejam geralmente voltados para o ambiente corporativo, o conceito pode ser facilmente adaptado para o ambiente de salas de aula, contribuindo para a eficiência do controle de presença dos alunos.

Quanto ao marketing de geolocalização, este conceito se torna relevante à medida que a solução proposta neste trabalho começa a abranger eventos em geral. O marketing de geolocalização pode ser utilizado como uma plataforma para a publicidade de empresas interessadas em promover seus eventos com base na localização do público-alvo. Isso pode aumentar a visibilidade dos eventos e atrair um público mais amplo.

Portanto, embora as soluções existentes não correspondam exatamente à deste trabalho, a análise dessas soluções e técnicas fornece percepções valiosas que podem contribuir para o desenvolvimento do aplicativo. Além disso, ajuda a posicionar a proposta dentro do contexto tecnológico atual, destacando sua relevância e inovação. Para sintetizar essas informações, a Tabela 1 compara as funcionalidades e diferenças entre as soluções analisadas e a desenvolvida neste projeto.

Tabela 1 – Comparativo entre as soluções

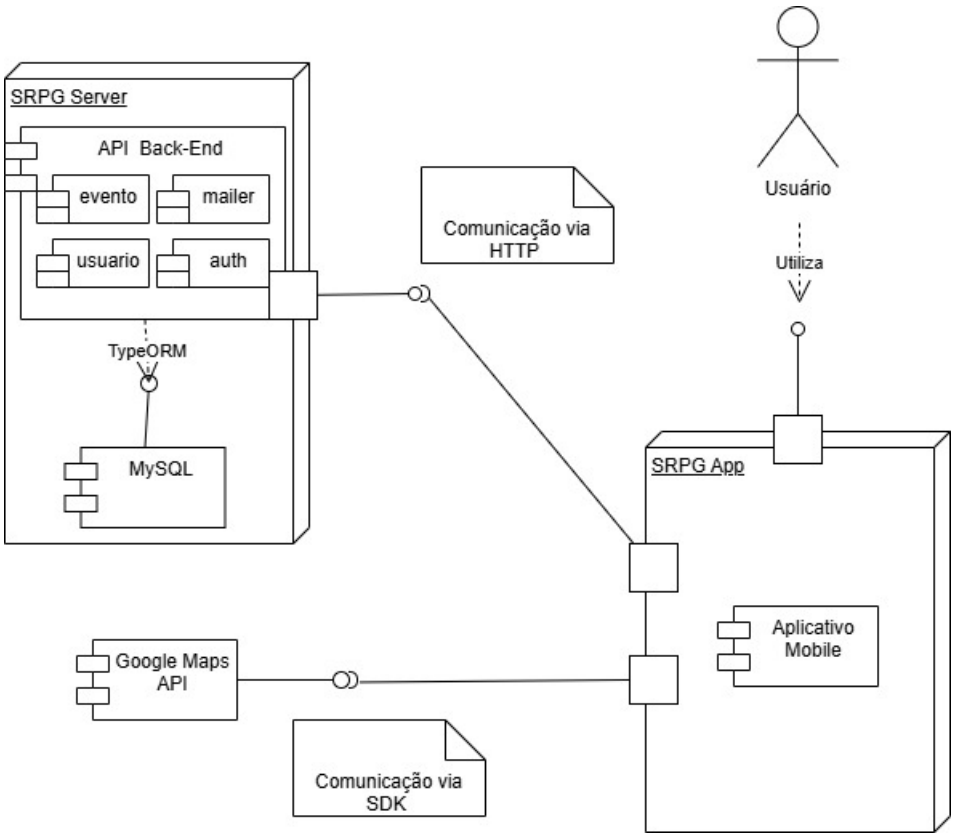
Solução/Conceito	**Funcionalidades Principais**	**Diferenças para a solução apresentada**
Factorial (Controle de Ponto)	Controle de ponto com registro de geolocalização; acessível via aplicativo móvel.	Voltado exclusivamente para o ambiente corporativo e controle de jornada de trabalho, não aplicável a salas de aula.
Simpax.Geoloc.Multi	Controle de ponto com geolocalização, gestão de folha de pagamento e controle de benefícios.	Similar ao Factorial, foca no ambiente corporativo, sem funcionalidades específicas para contextos educacionais.
Sympla (Registro em Eventos)	Registro de presença em eventos via leitura de QR Codes; gerenciamento de participantes.	Não utiliza geolocalização; projetado para eventos e não para ambientes educacionais.
Check-in Serpro	Registro de presença gratuito via leitura de QR Codes.	Sem geolocalização e funcionalidades educativas; também projetado para eventos.
TOTVS Educacional	Soluções variadas para controle de frequência, incluindo biometria e integração com sistemas institucionais.	Não utiliza geolocalização; funcionalidades robustas, mas voltadas para instituições que utilizam TOTVS.
Marketing de Geolocalização	Envio de anúncios personalizados em tempo real com base na localização do usuário; rastreamento de hábitos do público.	Conceito complementar, mas não diretamente aplicável ao controle de presença; relevante para expandir funcionalidades.
Proposta do TCC	Controle de presença em ambientes educacionais via geolocalização; registro simplificado e seguro usando smartphones.	Foco em geolocalização para controle de presença; voltada para o contexto educacional e eventos, com maior acessibilidade e praticidade.

Fonte: Elaborada pelo autor.

3 Descrição do Projeto

A Figura 1 apresenta uma visão geral do sistema, destacando os principais blocos (módulos).

Figura 1 – Diagrama de componentes



Fonte: Elaborada pelo autor.

3.1 Aplicativo Mobile

O aplicativo mobile é a interface principal para os usuários interagirem com o sistema.

As funcionalidades incluem: Registro de presença, histórico de presenças, histórico de aulas/eventos realizados pelo professor/organizador, geração de relatórios específicos para o perfil de usuário, entre outros.

A arquitetura e componentes desse bloco se resumem em três principais:

- Telas (Widgets) : O aplicativo é composto por várias telas, como a tela de registro de presença, histórico e configurações.

- Integração com Back-End: O aplicativo se comunica com o Back-End para enviar e receber dados. Isso inclui autenticação, envio de localização e solicitações de histórico.
- Geolocalização: O aplicativo utiliza recursos de geolocalização do dispositivo para obter a posição atual do usuário, inclusive quando estiver em "segundo plano".

Os dados são transmitidos de forma que:

- Entradas: Localização do usuário, solicitações de registro de presença, criação de salas.
- Saídas: Notificações, informações de presença, relatórios.
- Processamento: Comunicação com o Back-End, exibição de dados no aplicativo.

3.2 API Back-End

Responsável por processar requisições, gerenciar dados e garantir a comunicação eficiente entre o aplicativo mobile e o banco de dados esse bloco do sistema é vital para seu funcionamento pleno. Fornecendo APIs para registro de presença, consulta de histórico, notificações, validação de dados e a autenticação de usuários.

A arquitetura e componentes desse bloco se resumem em três principais:

- Controladores (Controllers): Controlam as rotas da API, recebendo requisições e enviando respostas.
- Serviços (Services): Lidam com a lógica de negócios, como processamento de presenças, criação de aulas/eventos e outros.
- Módulos (Modules): Agrupam controladores, serviços e outros componentes relacionados.

Os dados são transmitidos de forma que:

- Entradas: Requisições do aplicativo, dados do banco de dados.
- Saídas: Respostas para o aplicativo, atualizações no banco de dados.
- Processamento: Autenticação, validação, acesso ao banco de dados, processamento de regras negócio.

3.3 Banco de Dados

O banco de dados é fundamental para armazenar e gerenciar os dados do sistema. Neste projeto, será utilizado para persistir as informações relacionadas a presenças, eventos e quaisquer outros dados relevantes.

Os dados são transmitidos de forma que:

- Entradas: Dados do Back-End.
- Saídas: Dados para o Back-End.
- Processamento: Armazenamento e recuperação de informações.

4 Especificação

As especificações são uma parte crucial do desenvolvimento de software. Elas fornecem uma descrição detalhada do comportamento, definindo o que deve fazer e como deve responder a entradas específicas. As especificações ajudam a estabelecer um entendimento comum entre todas as partes interessadas (desenvolvedores, clientes, usuários, etc.) sobre o que o sistema deve alcançar (ALFF, 2018).

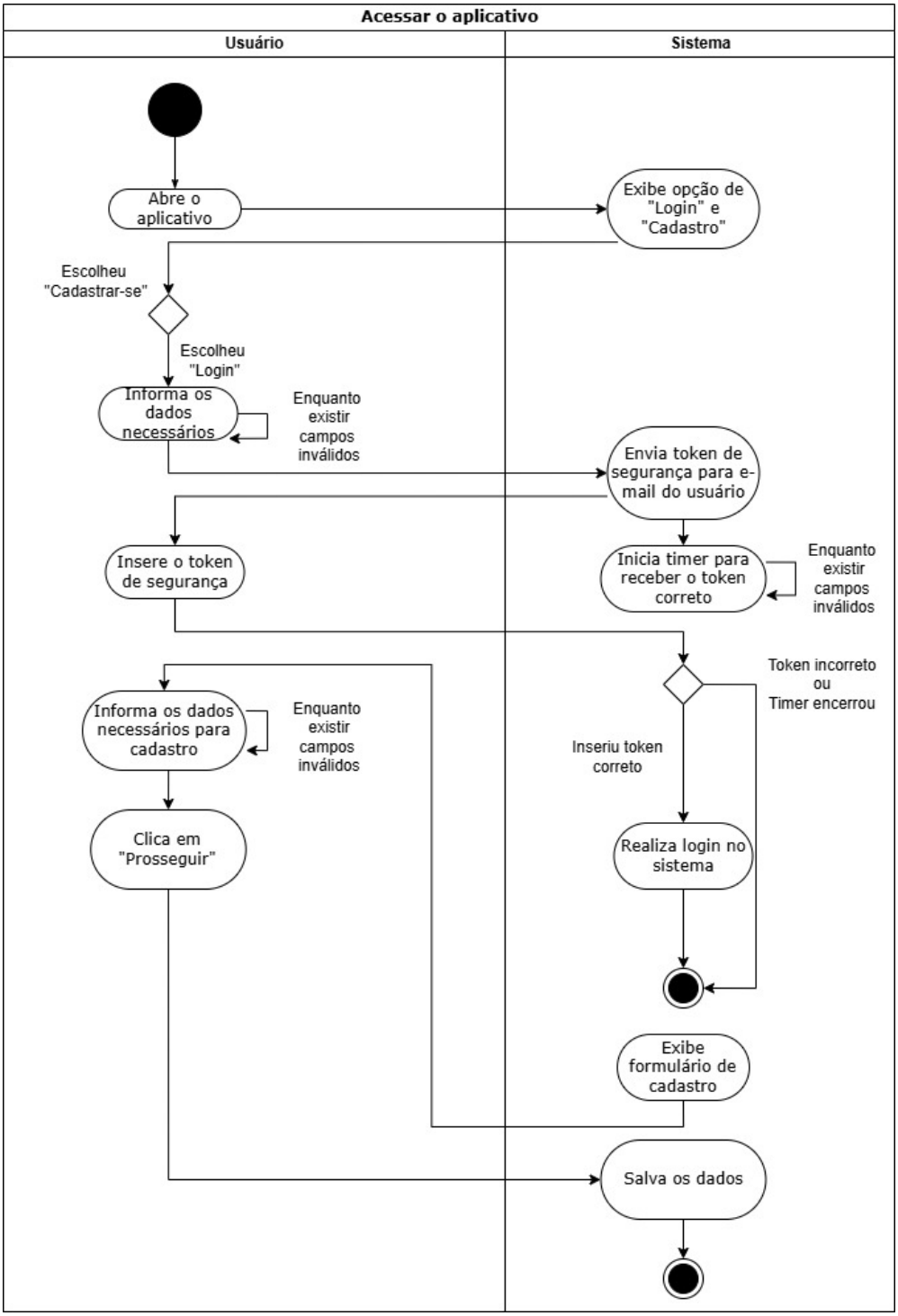
Este capítulo aborda as especificações do sistema desenvolvido, incluindo a descrição detalhada dos casos de uso e a representação visual dos mesmos através de diagramas UML (Unified Modeling Language), diagrama de classes, e outros (ALFF, 2022).

4.1 Atividades

O diagrama de atividades possibilita a modelagem de atividades, que compreendem um método ou algoritmo a partir de blocos de comandos correspondentes às ações de um usuário ou objeto. (DEVMEDIA, 2024d)

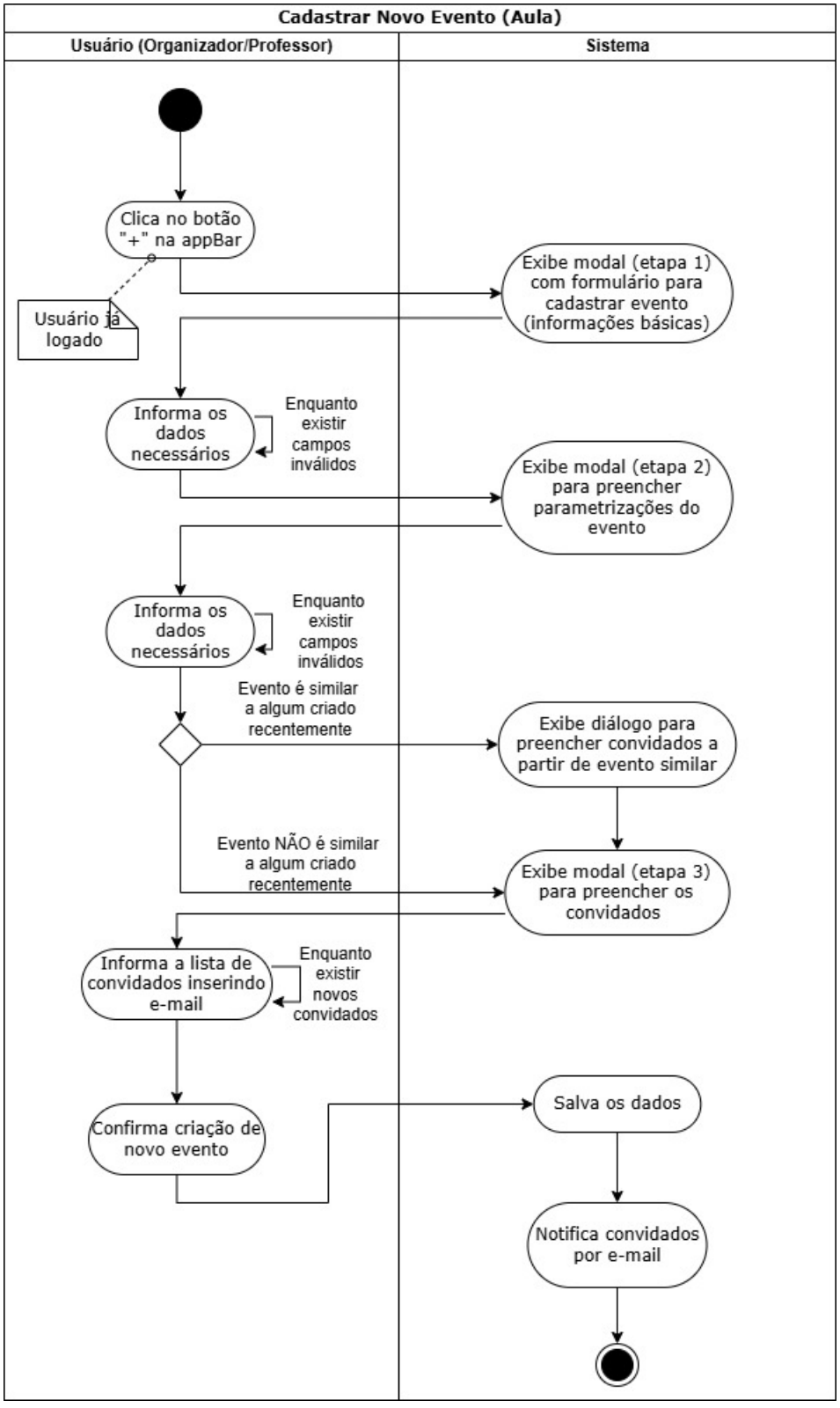
Os desenvolvedores optaram por representar as principais atividades do Sistema de Gerenciamento por Geolocalização: Acessar o Aplicativo (Figura 2), Criar Novo Evento (Figura 3), e Realizar Evento, com intervalo (Figura 4) e sem intervalo (Figura 5). São estas as responsáveis pelo funcionamento básico do sistema: primeiramente o usuário deve se cadastrar (ou realizar o login), posteriormente cadastrar um evento ou participar de um, sendo que este evento deve ter sido iniciado pelo organizador.

Figura 2 – Diagrama de Atividades: Acessar o Aplicativo



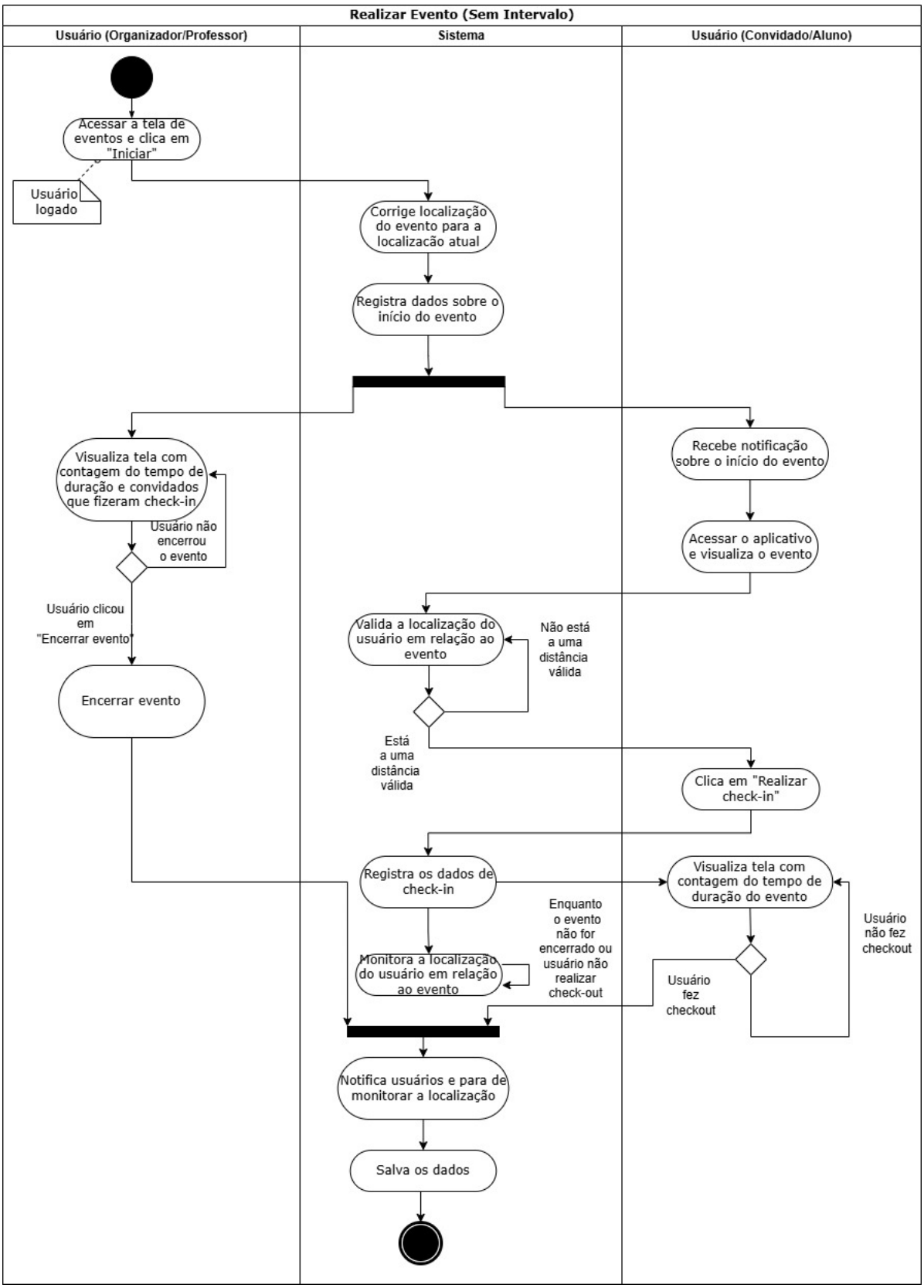
Fonte: Elaborada pelo autor.

Figura 3 – Diagrama de Atividades: Criar Novo Evento



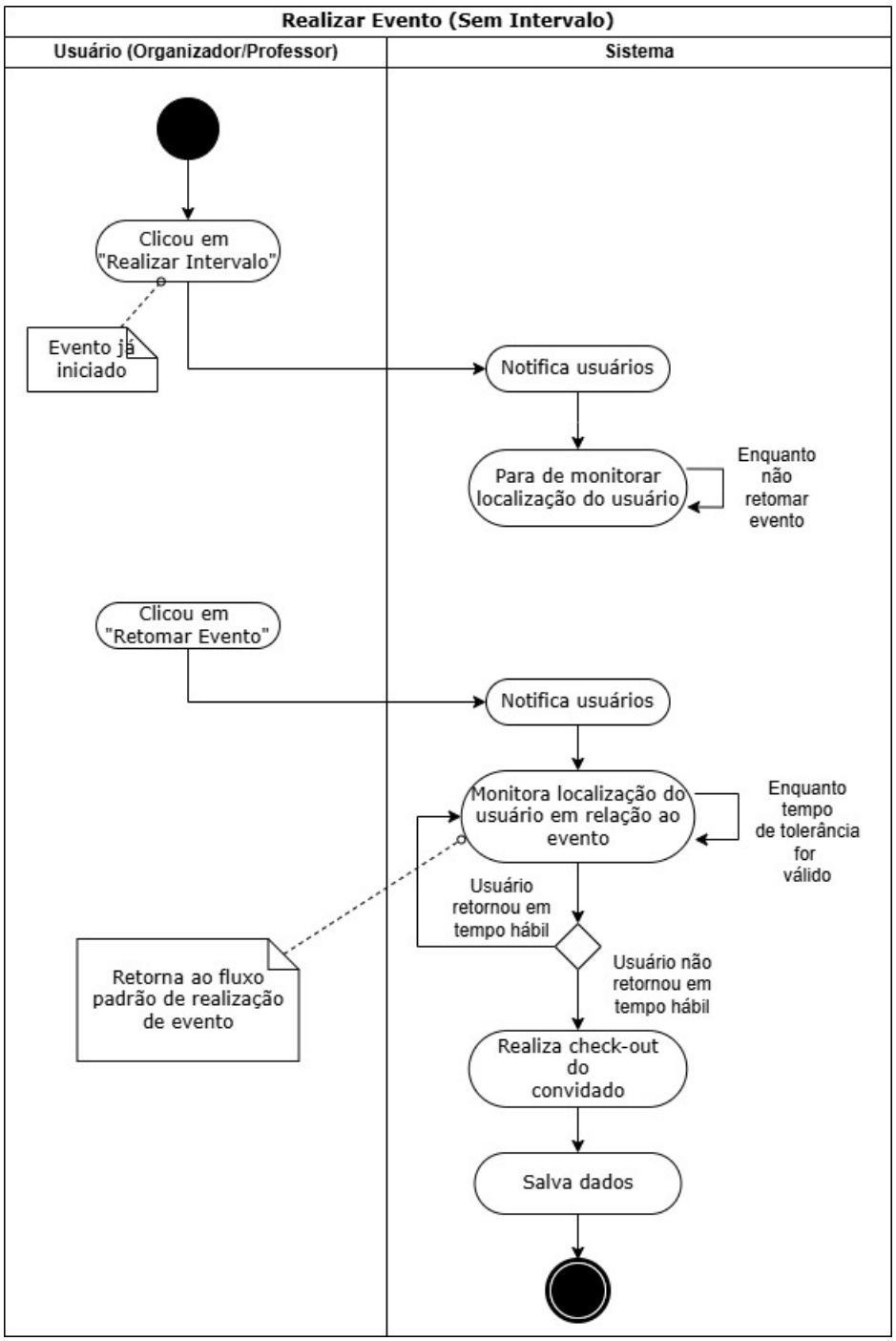
Fonte: Elaborada pelo autor.

Figura 4 – Diagrama de Atividades: Realizar Evento (Sem Intervalo)



Fonte: Elaborada pelo autor.

Figura 5 – Diagrama de Atividades: Realizar Evento (Com Intervalo)



Fonte: Elaborada pelo autor.

4.2 Casos de Uso

Os casos de uso são uma técnica de modelagem que descreve a interação entre os usuários (ou "atores") e o sistema, em termos das atividades ou tarefas que os usuários realizam no sistema. Cada caso de uso representa um fluxo de trabalho específico que o sistema deve suportar e inclui informações como pré-condições, pós-condições, cenário principal, exceções e

assim por diante. Os casos de uso são úteis para capturar requisitos funcionais e podem ser facilmente compreendidos tanto pelos usuários finais quanto pelos desenvolvedores (SOUZA, 2023). Nesta seção, descreveremos os casos de uso para o sistema de registro de presença em salas de aulas/eventos através de geolocalização.

4.2.1 Diagrama de Casos de Uso

O diagrama de casos de uso (Figura 6) é utilizado para documentar o que o sistema faz do ponto de vista do usuário, ou seja, descreve as principais funcionalidades do sistema e sua interação com os usuários (atores do sistema). (DEVMEDIA, 2024c)

Figura 6 – Diagrama de Casos de Uso

Fonte: Elaborada pelo autor.

4.2.2 Descrição dos Casos de Uso

1. CDU1: Realizar login

- Descrição: Este caso de uso permite que um usuário realize login e pode acessar o aplicativo.
- Pré-condição: O usuário deve ter o aplicativo instalado e deve estar na tela inicial do mesmo.
- Disparador: O usuário seleciona a opção "Entrar".
- Cenário Principal:
  a) O sistema apresenta o formulário de login.
  b) O usuário preenche as informações solicitadas (email ou CPF e senha). O sistema valida e registra as informações fornecidas.
  c) O sistema confirma o se as informações enviadas estão corretas, caso estejam, exibe uma janela para que o usuário insira o token enviado para o e-mail cadastrado e inicia um contador para que o campo seja preenchido de forma correta.
  d) O usuário preenche o campo com o token.
  e) O sistema valida o token e permite o acesso do usuário, o encaminhando para a página inicial do sistema.
- Pós-condições: Usuário realiza login, permitindo-lhe acessar funcionalidades do sistema.
- Exceções:
  a) Se o usuário não preencher algum campo ou preencher de forma incorreta, o sistema deve exibir uma mensagem.
  b) Se o usuário não inserir o token enviado para o e-mail durante o tempo do contador ou inserir um valor incorreto, o sistema deve exibir uma mensagem e não permitir o login.

2. CDU2: Realizar cadastro

- Descrição: Este caso de uso permite que um novo usuário crie uma conta no aplicativo.
- Pré-condição: O usuário deve ter o aplicativo instalado e deve estar na tela inicial do mesmo.
- Disparador: O usuário seleciona a opção "cadastrar-se".
- Cenário Principal:
  a) O sistema apresenta o formulário de cadastro.
  b) O usuário preenche as informações básicas (email, senha e outros). O sistema valida e registra as informações fornecidas.
  c) O sistema confirma o cadastro do usuário.

- Pós-condições: A conta do usuário é criada, permitindo-lhe acessar funcionalidades restritas a usuários cadastrados.
- Exceções:
  a) Se o email já estiver em uso, informar ao usuário e solicitar outro email.

3. CDU3: Cadastrar um evento/aula
   - Descrição: Este caso de uso permite que um usuário logado crie um novo evento ou aula.
   - Pré-condição: O usuário deve estar logado no aplicativo.
   - Disparador: O usuário seleciona a opção "cadastrar evento/aula".
   - Cenário Principal:
     a) O sistema apresenta o formulário de cadastro de evento/aula.
     b) O usuário insere as informações relativas ao evento, incluindo a localização onde será realizado, parâmetros, entre outros.
     c) O usuário convida, por email, outros usuários para participarem do evento/aula.
     d) O sistema valida e registra as informações fornecidas.
     e) O sistema confirma o cadastro do evento/aula.
   - Pós-condições: O evento/aula é criado e os usuários convidados são notificados.
   - Exceções
     a) Se o usuário não preencher todas as informações necessárias, informar ao usuário e solicitar as informações faltantes.

4. CDU4: Iniciar um evento/aula
   - Descrição: Este caso de uso permite que um usuário logado inicie um evento ou aula que ele criou.
   - Pré-condição: O usuário deve estar logado no aplicativo e deve estar no local do evento.
   - Disparador: O usuário seleciona a opção "iniciar evento/aula".
   - Cenário Principal:
     a) O sistema verifica a localização do usuário e a atribui ao evento.
     b) O usuário clica em "iniciar".
     c) O sistema inicia o evento e notifica os usuários convidados.
   - Pós-condições: O evento/aula é iniciado e os usuários convidados são notificados.

5. CDU5: Realizar check-in em evento

- Descrição: Este caso de uso permite que um usuário convidado faça check-in em um evento ou aula, seja manualmente pelo organizador/professor ou automaticamente pelo sistema.
- Pré-condição: O usuário deve estar logado no aplicativo, deve ter sido convidado para o evento e deve estar no local do evento (para check-in automático) ou o organizador/professor deve estar logado no aplicativo e o evento deve ter sido iniciado (para check-in manual).
- Disparador: O usuário seleciona a opção "fazer check-in" ou o organizador/professor seleciona a opção "inserir registro de presença manual".
- Cenário Principal (Check-in pelo Convidado/Aluno):
  a) O sistema verifica se o usuário está no local do evento.
  b) O usuário clica em "fazer check-in".
  c) O sistema registra o check-in do usuário e começa a monitorar sua localização durante o evento.
- Cenário Secundário (Check-in manual pelo professor/organizador)
  a) O sistema apresenta um formulário para inserir o e-mail do convidado e o tempo de permanência
  b) O organizador/professor insere as informações necessárias
  c) O sistema valida e registra as informações fornecidas
  d) O sistema confirma que a presença do usuário foi registrada
- Pós-condições: O usuário está registrado como presente no evento.
- Exceções:
  a) Se o usuário não estiver no local do evento, bloquear o botão de check-in e mostrar uma mensagem indicativa.
  b) Se o e-mail fornecido não corresponder a um usuário convidado, informar ao organizador/professor e solicitar um e-mail válido.

6. CDU6: Realizar check-out de evento

- Descrição: Este caso de uso permite que um usuário convidado faça check-out de um evento ou aula manualmente ou seja automaticamente registrado como ausente ao se distanciar do evento.
- Pré-condição: O usuário deve estar logado no aplicativo e deve ter feito check-in no evento.
- Disparador: O usuário seleciona a opção "fazer check-out" ou o sistema detecta que o usuário se distanciou do evento.
- Cenário Principal (Check-out Manual):

a) O usuário clica em "fazer check-out"
b) O sistema registra o check-out do usuário
c) Tempo de presença será enviado ao organizador/professor ao fim do evento

- Cenário Secundário (Check-out Automático):
  a) O sistema detecta que o usuário se distanciou do evento
  b) O sistema conta o tempo de tolerância fora do evento, que deve ser cadastrado pelo organizador/professor
  c) Após o tempo de tolerância, o sistema faz o check-out do usuário de forma automática
- Pós-condições: O usuário está registrado como ausente do evento e o tempo de presença é registrado.
- Exceções:
  a) Se o usuário retornar ao local do evento dentro do tempo de tolerância, o sistema não faz o check-out automático.

7. CDU7: Encerrar evento
   - Descrição: Este caso de uso permite que o organizador/professor encerre um evento ou aula.
   - Pré-condição: O usuário deve estar logado no aplicativo e deve ter iniciado o evento.
   - Disparador: O usuário seleciona a opção "encerrar evento".
   - Cenário Principal:
     a) O usuário clica em "encerrar evento".
     b) O sistema encerra o evento e faz check-out automático de todos os usuários.
     c) O sistema possibilita a visualização de informações do evento encerrado na tela de "Histórico", além de possibilitar o envio de um relatório por e-mail ao organizador/professor em até 15 minutos do encerramento.
     d) O sistema envia uma notificação por e-mail para todos os convidados, com informações básicas sobre o evento e o tempo de permanência capturado.
   - Pós-condições: O evento é encerrado.
8. CDU8: Visualizar relatórios de eventos
   - Descrição: Este caso de uso permite que um usuário visualize relatórios de eventos que ele organizou.
   - Pré-condição: O usuário deve estar logado no aplicativo.
   - Disparador: O usuário seleciona a opção "visualizar relatórios".

- Cenário Principal:
  a) O sistema apresenta uma lista de todos os eventos que o usuário organizou..
  b) O usuário seleciona o evento que deseja visualizar o relatório
  c) O sistema exibe o relatório ou exporta para o e-mail do usuário.
- Pós-condições: O usuário visualiza ou exporta os relatórios desejados.
- Exceções:
  a) Se o usuário não organizou nenhum evento, não será possível visualizar nenhum dado.

9. CDU9: Pausar um evento
   - Descrição: Este caso de uso permite que o organizador/professor pause um evento ou aula em andamento.
   - Pré-condição: O organizador/professor deve estar logado no aplicativo e o evento deve ter sido iniciado.
   - Disparador: O organizador/professor seleciona a opção "pausar evento".
   - Cenário Principal:
     a) O organizador/professor clica em "pausar evento".
     b) O sistema pausa o evento e impede que o check-out automático seja feito se os usuários se distanciarem durante a pausa.
     c) O sistema confirma que o evento foi pausado.
   - Pós-condições: O evento é pausado e os usuários podem se distanciar sem ter o check-out feito automaticamente.
   - Exceções:
     a) Se o evento não foi iniciado, informar ao organizador/professor e solicitar que ele inicie o evento.
10. CDU10: Alterar senha
    - Descrição: Este caso de uso permite que um usuário altere sua senha.
    - Pré-condição: O usuário deve estar logado no aplicativo ou deve ter selecionado a opção "esqueci minha senha".
    - Disparador: O usuário seleciona a opção "alterar minha senha" ou "esqueci minha senha".
    - Cenário Principal (Usuário logado):
      a) O sistema apresenta um modal para alterar a senha
      b) O usuário infroma a senha atual e a nova

c) O sistema confirma a alteração de senha

- Cenário Secundário (Usuário não logado):
  a) O usuário seleciona a opção "Esqueci minha senha"
  b) O sistema pede que o usuário preencha o e-mail para enviar as informações de redefinifição de senha
  c) O usuário insere o e-mail
  d) O sistema realiza o envio das instruções
  e) O usuário acessa o e-mail e segue as instruções para alteração
  f) O sistema realiza a alteração de senha
- Pós-condições: O usuário tem sua senha redefinada e é deslogado do sistema.
- Exceções:
  a) Se a nova senha não atender aos critérios de segurança (por exemplo, comprimento mínimo, presença de números e letras, etc.), o sistema deve informar ao usuário e solicitar que ele insira uma nova senha válida.
  b) Se a senha e a confirmação da senha não corresponderem, o sistema deve informar ao usuário e solicitar que ele insira novamente as senhas.
  c) Se o usuário estiver logado e a senha atual fornecida for incorreta, o sistema deve informar ao usuário e solicitar que ele insira a senha correta.
  d) Se o usuário não estiver logado e o e-mail fornecido para redefinição de senha não estiver associado a nenhuma conta, o sistema deve informar ao usuário e solicitar um e-mail válido.

11. CDU11: Monitorar localização

- Descrição: Este caso de uso permite que o sistema monitore a localização do usuário durante um evento ou aula.
- Pré-condição: O usuário deve estar logado no aplicativo, deve ter sido convidado para o evento, deve ter feito check-in no evento e deve ter permitido que o aplicativo acesse sua localização.
- Disparador: O usuário faz check-in no evento.
- Cenário Principal:
  a) O sistema começa a monitorar a localização do usuário.
  b) O sistema verifica continuamente se o usuário está dentro do raio do local do evento.
- Pós-condições: A localização do usuário é monitorada durante o evento.
- Exceções:

a) Se o usuário desativar o acesso à localização, o sistema deve informar ao usuário e solicitar que ele ative o acesso à localização.

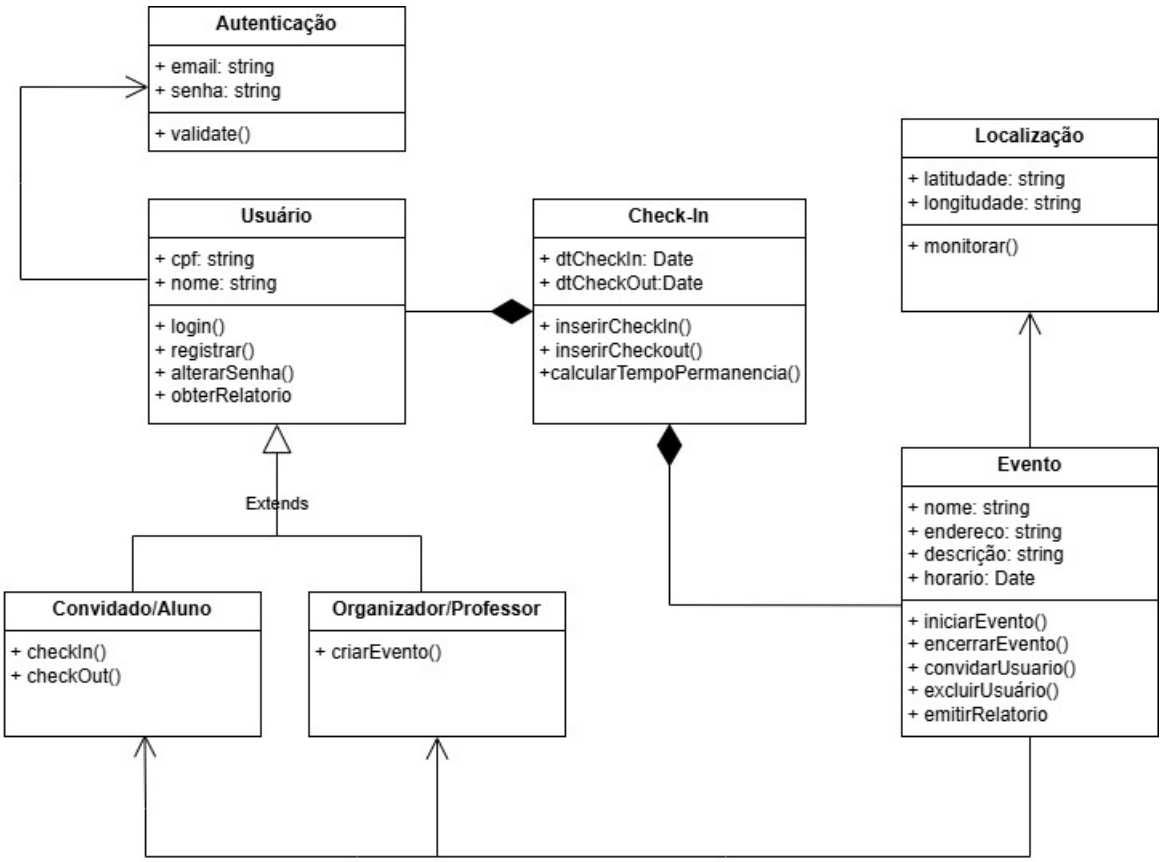
12. CDU12: Notificar usuários

- Descrição: Este caso de uso permite que o sistema envie notificações para os usuários.
- Pré-condição: N/A
- Disparador: Dependendo do contexto, diferentes disparadores podem se aplicar.
- Cenário Principal:
  a) O sistema cria uma notificação com a mensagem apropriada.
  b) O sistema envia a notificação para os usuários relevantes.
- Pós-condições: Os usuários recebem a notificação.
- Exceções:
  a) Se o usuário desativar as notificações, o sistema não poderá enviar notificações para esse usuário.

4.3 Classes

O diagrama de classes (Figura 7) permite a visualização das classes que compõem o sistema, além de uma visão superficial de seus respectivos atributos e métodos. O relacionamento entre classes também é ilustrado neste diagrama. (DEVMEDIA, 2024a)

Figura 7 – Diagrama de Classes

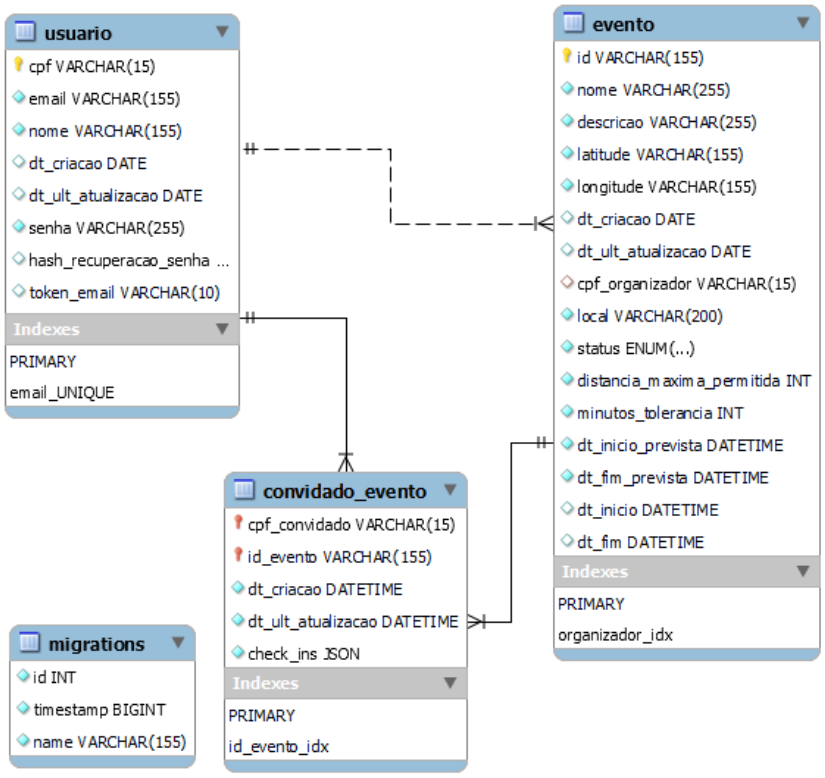


Fonte: Elaborada pelo autor.

4.4 Modelo Entidade-Relacionamento

O Modelo Entidade Relacionamento (MER), ilustrado na Figura 8, representa de forma abstrata a estrutura de um BD, descrevendo as entidades, seus atributos e relacionamentos. (DEVMEDIA, 2024b)

Figura 8 – Modelo Entidade Relacionamento (MER)



Fonte: Elaborada pelo autor.

5 Tecnologias

Neste capítulo, estão apresentadas as tecnologias que foram utilizadas em cada módulo do sistema.

5.1 Aplicativo Mobile

Para desenvolvimento da interface de contato com o usuário foi utilizado Flutter[1], que é uma estrutura de aplicativo móvel de código aberto criada pelo Google que utiliza linguagem Dart[2]. O Flutter foi escolhido para o desenvolvimento do aplicativo móvel devido à sua capacidade de permitir o desenvolvimento de aplicativos para iOS e Android a partir de uma única base de código. Isso significa que os desenvolvedores podem escrever o código uma vez e ele funcionará em ambas as plataformas, economizando tempo e esforço.

Além disso, foi utilizado como recurso no aplicativo, a Google Maps API[3], que é uma coleção de APIs fornecidas pelo Google para uso com o Google Maps. Ela permite aos desenvolvedores incorporar funcionalidades do Google Maps em seus próprios aplicativos, incluindo a visualização de mapas, a busca de lugares e a obtenção de direções.

A Google Maps API é composta por várias APIs individuais, cada uma projetada para um propósito específico. Por exemplo, a Maps JavaScript API permite personalizar mapas com seu próprio conteúdo e imagens, a Places API fornece informações sobre milhões de locais, e a Geocoding API permite converter entre endereços e coordenadas geográficas.

No contexto deste projeto, a Google Maps API foi utilizada para fornecer funcionalidades de mapeamento e localização no aplicativo móvel. Isso permitirá aos usuários visualizar sua localização atual, buscar locais e obter direções. Além disso, a API foi usada para implementar o controle de presença baseado em geolocalização, permitindo verificar se os alunos estão realmente presentes na sala de aula ou no local do evento.

5.2 API Back-End e Banco de Dados

Para desenvolvimento da API Back-End foi utilizado Node.js que é um ambiente de tempo de execução para JavaScript[4] que possibilita execução de código fora de um navegador. A escolha da tecnologia foi feita devido à sua eficiência e capacidade de lidar com operações

[1] <https://docs.flutter.dev/ui>
[2] <https://dart.dev/guides>
[3] <https://developers.google.com/maps/apis-by-platform?hl=pt-br>
[4] <https://developer.mozilla.org/pt-BR/docs/web/javascript/guide/introduction>

simultâneas, o que é crucial para um aplicativo que precisa lidar com múltiplas solicitações de usuários ao mesmo tempo.

Além disso, também foi utilizado um framework para o desenvolvimento. Um framework de software é como uma estrutura que facilita o desenvolvimento de aplicativos. Imagine que você precisa cortar 1000 folhas de papel com as mesmas dimensões de 5 metros por 5 metros. Em vez de medir e cortar cada folha individualmente, você criaria uma estrutura de 5 metros por 5 metros e, com base nela, cortaria todas as folhas de papel de forma mais eficiente. Da mesma forma, um framework de Back-End oferece facilidades e abstração para o desenvolvimento de aplicativos, economizando tempo e esforço. Para esse desenvolvimento foi escolhido o NestJS[5] que é construído com TypeScript[6] (superconjunto de JavaScript, ou seja, um conjunto de ferramentas e formas mais eficientes de escrever código JavaScript, adicionando recursos que não estão presentes de maneira nativa na linguagem) e combina elementos de programação orientada a objetos (OOP), programação funcional (FP) e programação reativa funcional (FRP) que facilita diversas operações relevantes com o Node.js.

O Banco de Dados escolhido foi o MySQL[7]. Além dele, foram cogitados o PostgreSQL e o Oracle. Foi observado pelos desenvolvedores que o MySQL tem tudo que os Sistemas de Gerenciamento de Banco Dados (SGBD) maiores oferecem, mas com vantagens importantes, como dispor de código fonte aberto, ser gratuito e ter alto desempenho, sem deixar de lado a segurança.

Além disso, para apoio às operações no Banco de Dados será utilizado um ORM, ou Mapeamento Objeto-Relacional, que é uma técnica que permite consultar e manipular dados de um banco de dados usando um paradigma orientado a objetos. Nesse caso, será utilizado o TypeORM[8] que pode ser usado com TypeScript e JavaScript (ES2021). Seu objetivo é sempre suportar os recursos mais recentes do JavaScript e fornecer recursos adicionais que ajudam a desenvolver qualquer tipo de aplicação que usa bancos de dados. A utilização de um ORM, e especificamente o TypeORM, permite uma abstração da camada de acesso ao banco de dados, facilitando a manipulação dos dados e garantindo a integridade dos mesmos. Além disso, o TypeORM tem um bom suporte para TypeScript, o que melhora a qualidade do código.

Além disso, para execução da aplicação, no lado do servidor, foi utilizado Docker[9], que é uma tecnologia que permite construir, executar, testar e implantar aplicações distribuídas. Ele usa virtualização de nível de sistema operacional para entregar software em pacotes chamados contêineres. A escolha da tecnologia se deve à sua capacidade de isolar a aplicação e suas dependências em um contêiner, garantindo que ela funcionará da mesma maneira em qualquer ambiente. Isso facilita o processo de implantação e escalação da aplicação.

---

5 <https://docs.nestjs.com/>
6 <https://www.typescriptlang.org/docs/>
7 <https://dev.mysql.com/doc/>
8 <https://typeorm.io/>
9 <https://docs.docker.com/>

6 Interface Gráfica

Neste capítulo, estão apresentadas as principais telas que a interface gráfica do aplicativo de controle de presença que são a ponte de contato entre os usuários e as funcionalidades oferecidas.

6.1 Login

A tela de login é um componente crucial do aplicativo, pois é o primeiro ponto de contato entre o usuário e o aplicativo, ela permite que os usuários autentiquem-se e acessem as funcionalidades. Os principais componentes dessa tela incluem:

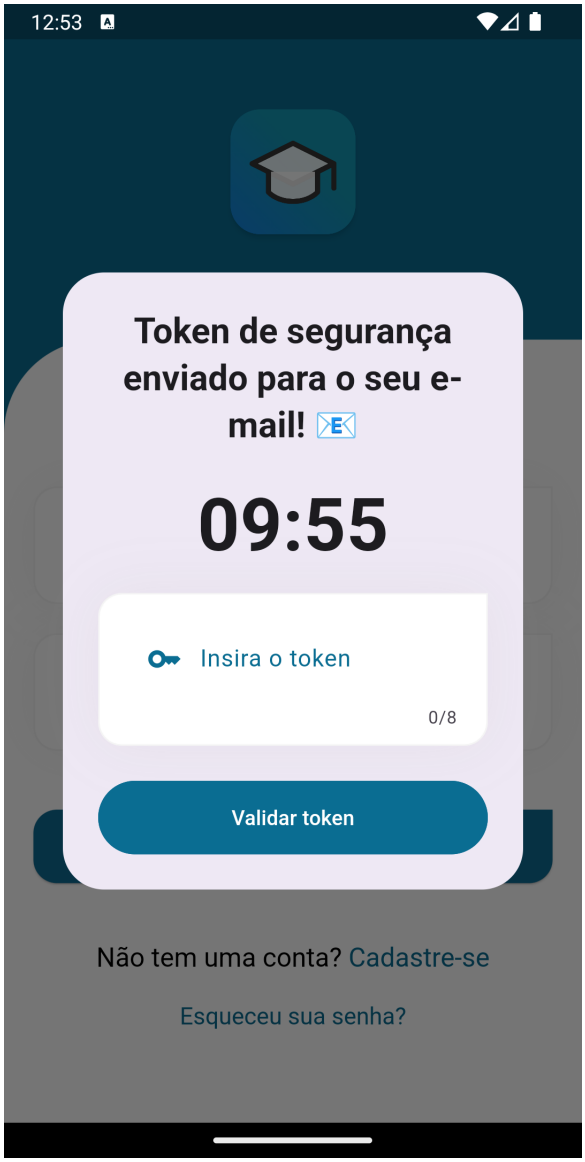
1. Campos de Entrada:
    - Os usuários devem preencher seu e-mail ou CPF no campo apropriado.
    - A senha também é solicitada para autenticação.
2. Botões Relevantes:
    - O botão "Cadastre-se" permite que novos usuários criem uma conta.
    - O botão "Esqueceu sua senha?" oferece uma opção para recuperar a senha por e-mail ou CPF.
3. Validação:
    - Ao realizar a solicitação de login, caso as informações estejam corretas, será enviado um token de segurança para o e-mail do usuário e exibida uma janela com contador para inserção e validação deste token, conforme mencionado na Seção 4.2.2.

As Figuras 9 e 10 mostram a disposição desses elementos na tela de login.

Figura 9 – Tela de Login

Fonte: Elaborada pelo autor.

Figura 10 – Tela de Login - Validação Token



Fonte: Elaborada pelo autor.

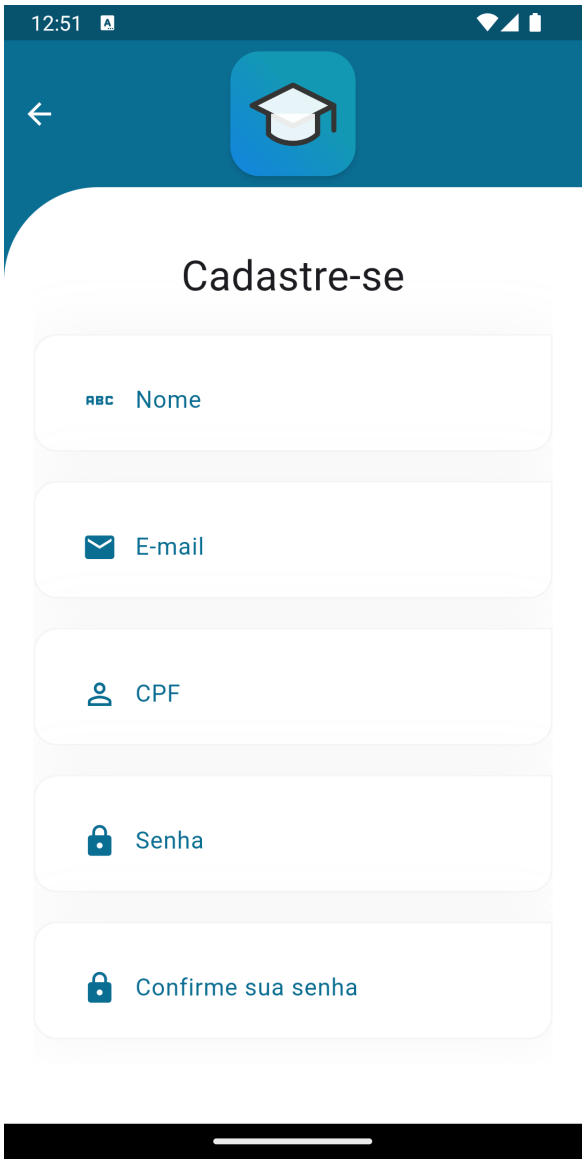
6.2 Cadastro

A tela de cadastro permite que os usuários criem novas contas fornecendo as informações necessárias. Os principais componentes dessas telas incluem:

1. Campos de Entrada:
    - Os usuários devem preencher os campos de e-mail, nome, senha, CPF.
2. Botões Relevantes:
    - O botão "Prosseguir" ao final página realiza a solicitação de criação do usuário caso as informações fornecidas cumpram os requisitos.

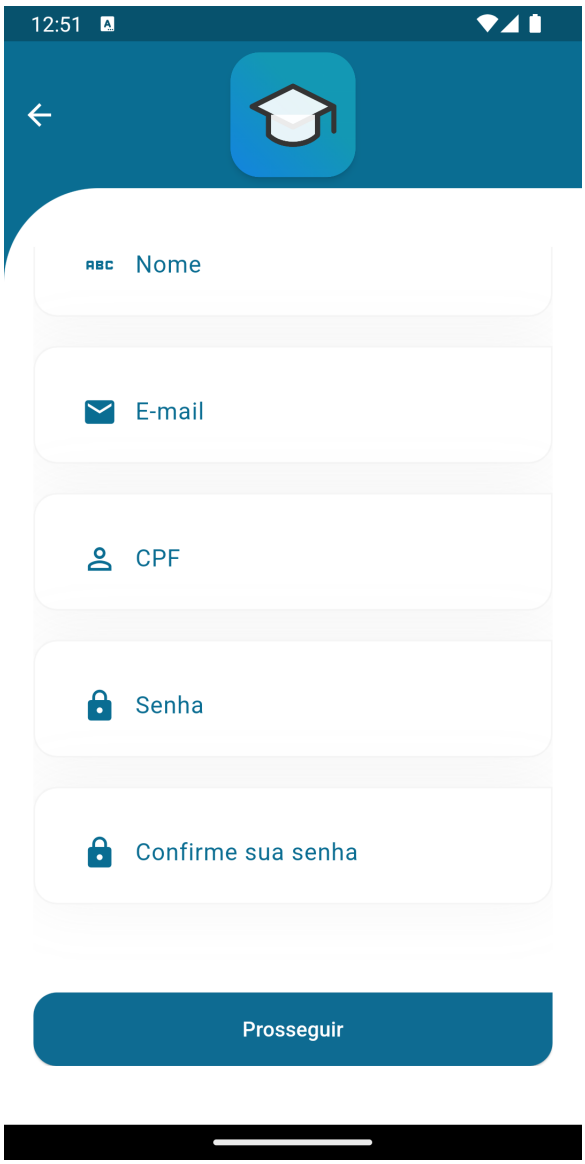
As Figuras 11 e 12 mostram a disposição desses elementos na tela de cadastro.

Figura 11 – Tela de Cadastro (1/2)



Fonte: Elaborada pelo autor.

Figura 12 – Tela de Cadastro (2/2)



Fonte: Elaborada pelo autor.

6.3 Inicial

A tela inicial é onde o usuário terá a primeira visualização do aplicativo após se autenticar. Os principais componentes dessa tela incluem:

1. Mapa: O usuário visualizará e poderá interagir com um mapa no modelo do Google Maps, além de interagir com os marcadores, o usuário poderá ainda realizar interações como obter a rota até o determinado evento.

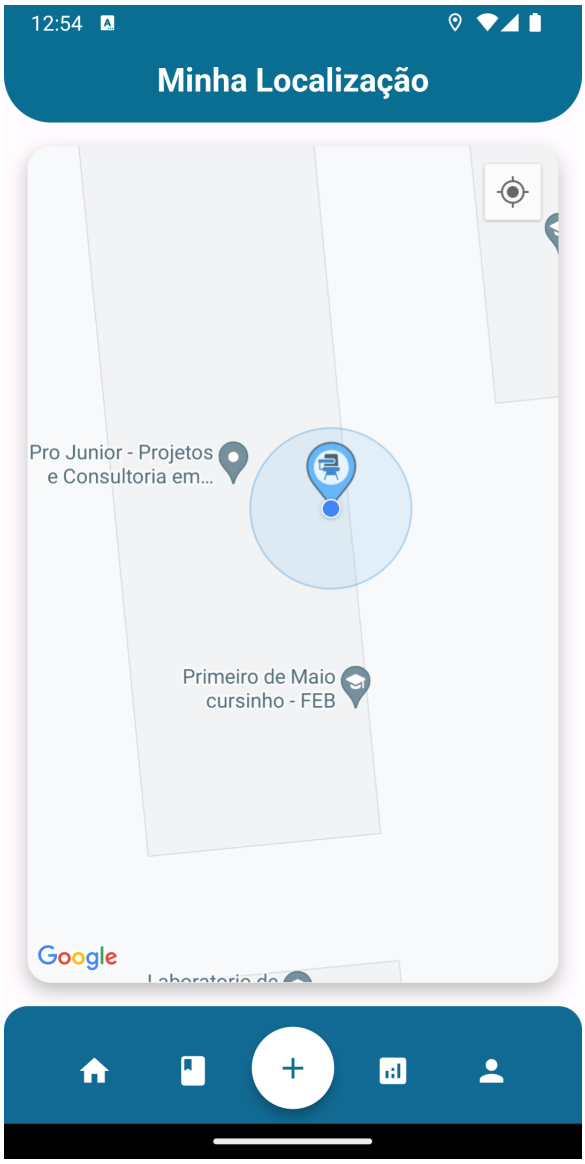
2. Marcadores: Os marcadores, representados na Figura 13 com ícone azul com uma cadeira no centro, representam as aulas em andamento, eles marcam o local onde está ocorrendo o

evento e possuem interações com o usuário, ao clicar em um marcador, será exibido outro componente na tela, onde será possível realizar o check-in no evento, esse comportamento está melhor descrito na seção 6.5.

3. *Bottom App Bar*: *Bottom App Bar* ou ainda Barra de Aplicativos Infererior, traduzido do inglês, se refere à um componente comumente utlizado no desenvolvimento de aplicativos móveis e serve como um menu, onde o usuário pode acessar páginas ou recursos específicos de forma facilitada (DESIGN, 2024)

A Figura 13 mostra a disposição desses elementos tela.

Figura 13 – Tela Inicial



Fonte: Elaborada pelo autor.

6.4 Meus Eventos

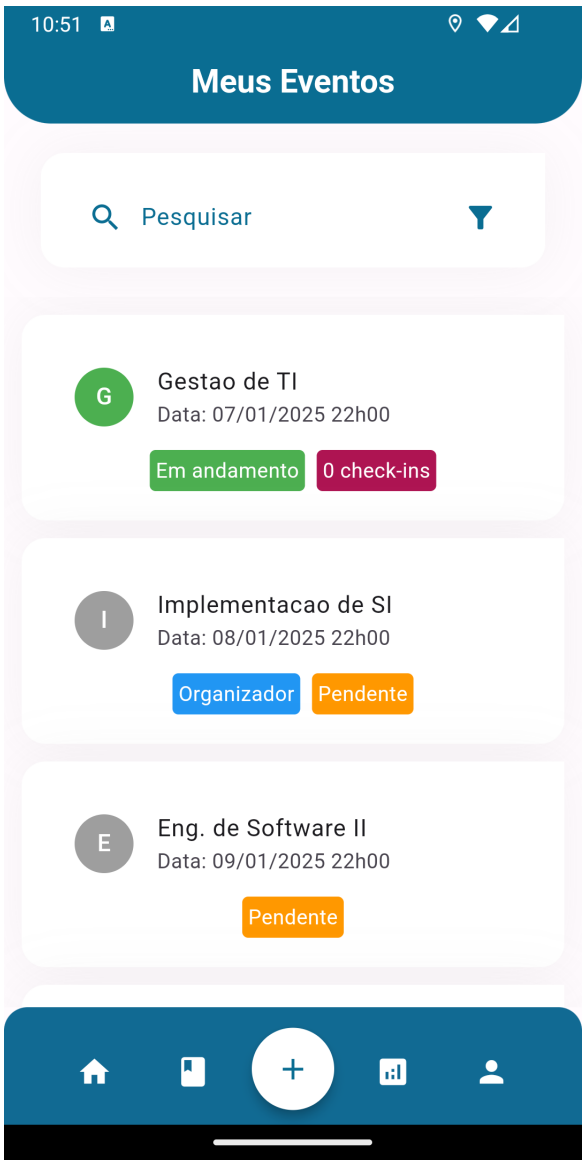
Ao clicar no segundo ícone, da esquerda para direita, da *Bottom App Bar* mencionada na seção 6.3, será exibida a tela de "Meus Eventos".

A tela de "Meus Eventos" apresenta uma visualização dos eventos do usuário que estão prestes à acontecer ou que já estão acontecendo, tanto dos que é convidado quanto dos que é organizador, por meio dessa tela o usuário pode realizar diferentes ações, dependendo do perfil em que se encaixa para o evento em questão. Os principais componentes dessa tela incluem:

1. Barra de Pesquisa e Filtros: O usuário poderá realizar a pesquisa pelo nome do através através desse componente, além de realizar filtros simples de eventos onde é organizador ou convidado, conforme a figura 15.
2. Cartões: Os cartões são o principal elemento dessa tela, eles apresentam algumas informações básicas sobre o evento em questão, como nome, data e local e interagem com o usuário de maneira diferente, à depender do perfil. Caso o usuário seja convidado no evento em questão, ao clicar no cartão será exibida uma tela para realizar o fluxo de check-in, descrito de forma mais abrangente na seção 6.5. E caso seja organizador do evento, ao clicar no cartão, será exibida uma tela para que o usuário inicie o evento em questão, conforme a figura 16.

A Figura 14 mostra a disposição desses elementos tela.

Figura 14 – Tela de “Meus Eventos” (Estado 1)



Fonte: Elaborada pelo autor.

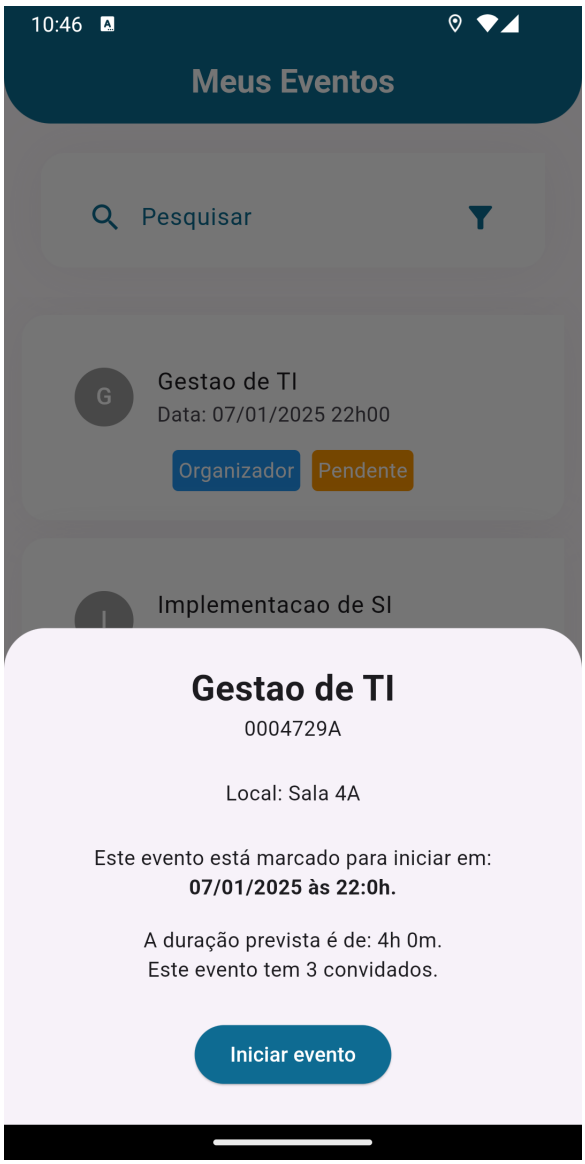
A Figura 15 mostra um diferente estado dessa mesma tela, onde o usuário pode interagir com filtros de busca para os eventos.

Figura 15 – Tela de “Meus Eventos” (Estado 2)

Fonte: Elaborada pelo autor.

A Figura 16 mostra outro diferente estado dessa mesma tela, onde o usuário sendo organizador, selecionou o cartão do evento, nesse caso, é apresentada uma tela para que o evento seja iniciado. Muito similar ao apresentado na seção 6.5.

Figura 16 – Tela de “Meus Eventos” (Estado 3)



Fonte: Elaborada pelo autor.

6.5 Check-in

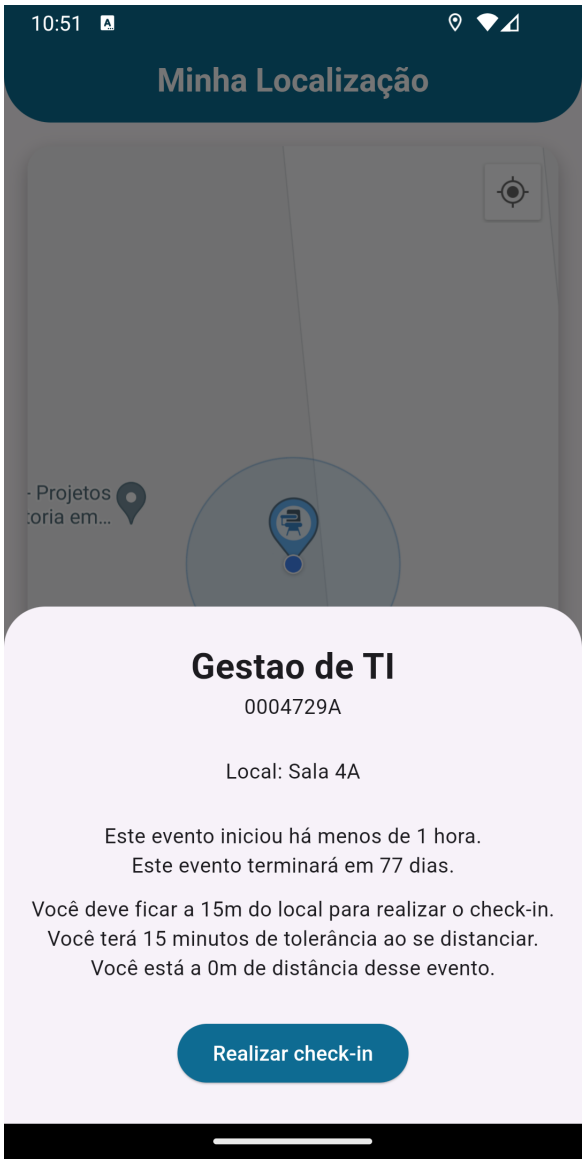
O componente onde se exibe a opção de realizar check-in trata-se, na verdade de um *Bottom Sheet*, ou “Folha Inferior”, que é um componente muito utilizado em desenvolvimento de aplicativos móveis onde se exibe uma segunda tela ancorado ao inferior da tela “pai” (DEV, 2019).

Neste caso, esse componente é exibido em duas diferentes situações, a primeira por interação do usuário através do marcador no mapa, mencionado na seção 6.3. E a outra é, também por interação do usuário, na tela de “Meus Eventos” mencionado na seção 6.4.

A Figura 17 demonstra a exibição desse componente através da interação com o

marcador, no mapa da tela inicial.

Figura 17 – Tela de Check-in (Caso 1)



Fonte: Elaborada pelo autor.

Ambas as interações exibem o mesmo componente, caso seja o mesmo evento, e ao clicar em "Realizar check-in", o usuário é encaminhado para a tela de andamento do evento do convidado, mencionado na seção 6.6

6.6 Evento (Convidado)

Ao realizar check-in em um evento, o usuário visualizará as telas dessa seção. Onde ficará até que realize o check-out de forma manual ou se distancie do evento, o que causa

um check-out automático, conforme os casos de uso descritos no Capítulo 4. Os principais componentes dessa tela incluem:

1. Botão "Realizar check-out": Através desse botão, o usuário poderá realizar o check-out de forma manual do evento, e registrar a sua presença apenas pelo tempo em que permanceu presente. Ao clicar no botão será exibida uma mensagem para confirmação dessa ação, conforme ilustrado na Figura 19.
2. Contador: É realizado uma contagem do tempo de permanência do usuário no evento, levando em consideração horas, minutos e segundos.

A Figura 18 demonstra a disposição desses elementos na tela.

Figura 18 – Tela de Evento (Convidado)

Fonte: Elaborada pelo autor.

A Figura 19 demonstra a mensagem de aviso exibida ao usuário que se distanciar do evento após realizar o check-in.

Figura 19 – Tela de Evento (Convidado) - Mensagem Check-out

Fonte: Elaborada pelo autor.

6.7 Evento (Organizador)

Ao iniciar um evento o usuário visualizará as telas dessa seção. Onde ficará até que o encerre de forma manual. Os principais componentes dessa tela incluem:

1. Botão "Pausar Evento": Através desse botão, o usuário organizador poderá pausar o andamento do evento. O que permitirá que os usuários se distanciem do evento sem que seja feito check-out de forma automática.

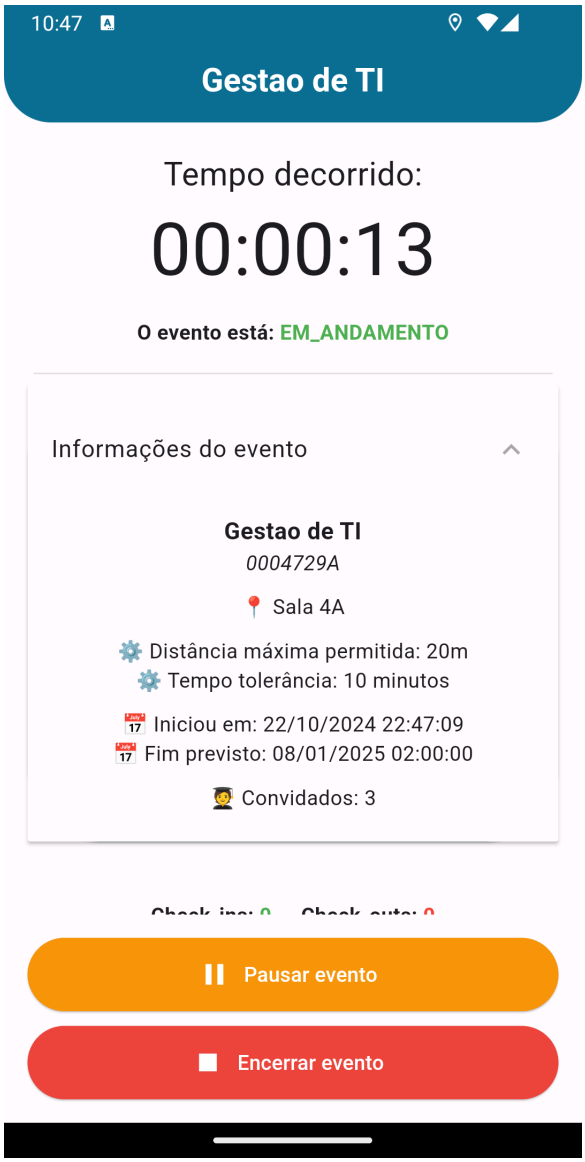
2. Botão “Encerrar Evento”: Através desse botão, o usuário organizador poderá encerrar o evento, o que irá realizar o check-out automático de todos os convidados que estiverem presentes.

3. Menus “Check-ins” e “Check-outs”: Através desses menus o usuário organizador pode visualizar os usuários presentes no evento, os que realizaram check-out e ainda, realizar o check-out manual de qualquer usuário que esteja no evento.

4. Contador: É realizado uma contagem do tempo de permanência do usuário no evento, levando em consideração horas, minutos e segundos.

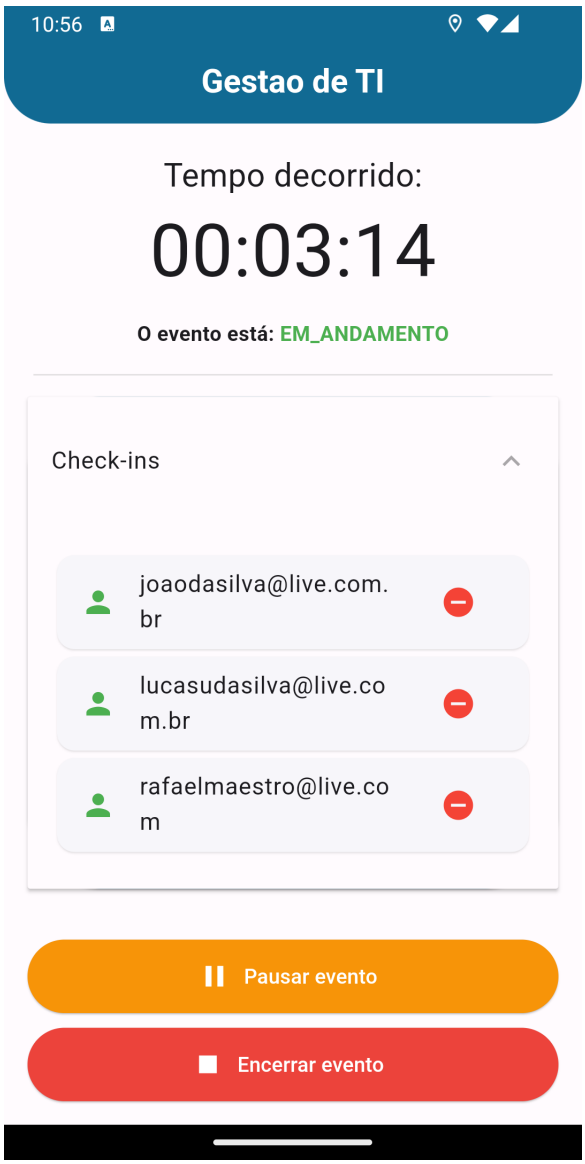
As Figuras 20 e 21 demonstram a disposição desses elementos na tela.

Figura 20 – Tela de Evento (Organizador 1/2)



Fonte: Elaborada pelo autor.

Figura 21 – Tela de Evento (Organizador 2/2)



Fonte: Elaborada pelo autor.

6.8 Histórico de Eventos

Ao clicar no segundo ícone, da direita para esquerda, da *Bottom App Bar* mencionada na Seção 6.3, será exibida a tela de "Histórico".

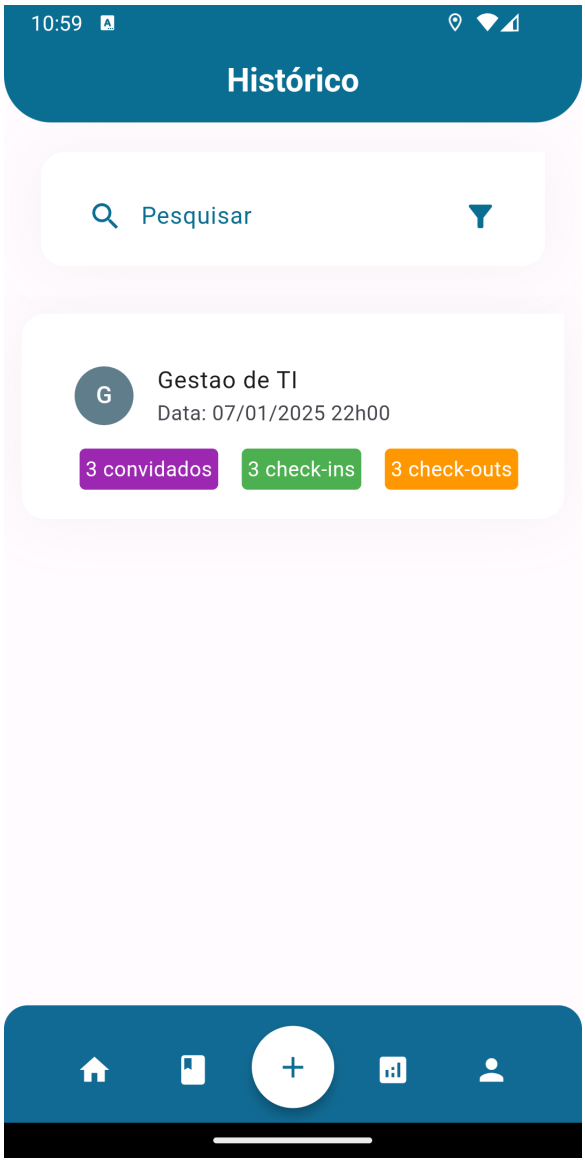
A tela de "Histórico" apresenta uma visualização dos eventos que o usuário já finalizou, por meio dessa tela o usuário pode visualizar informações sobre o evento, tais como os convidados que estiveram presentes e o tempo de permanência de cada um, além de poder solicitiar um relatório em planilha que será enviado para o e-mail cadastrado. Os principais componentes dessa tela incluem:

1. Barra de Pesquisa: O usuário poderá realizar a pesquisa pelo nome do evento através

desse componente, conforme a Figura 22.

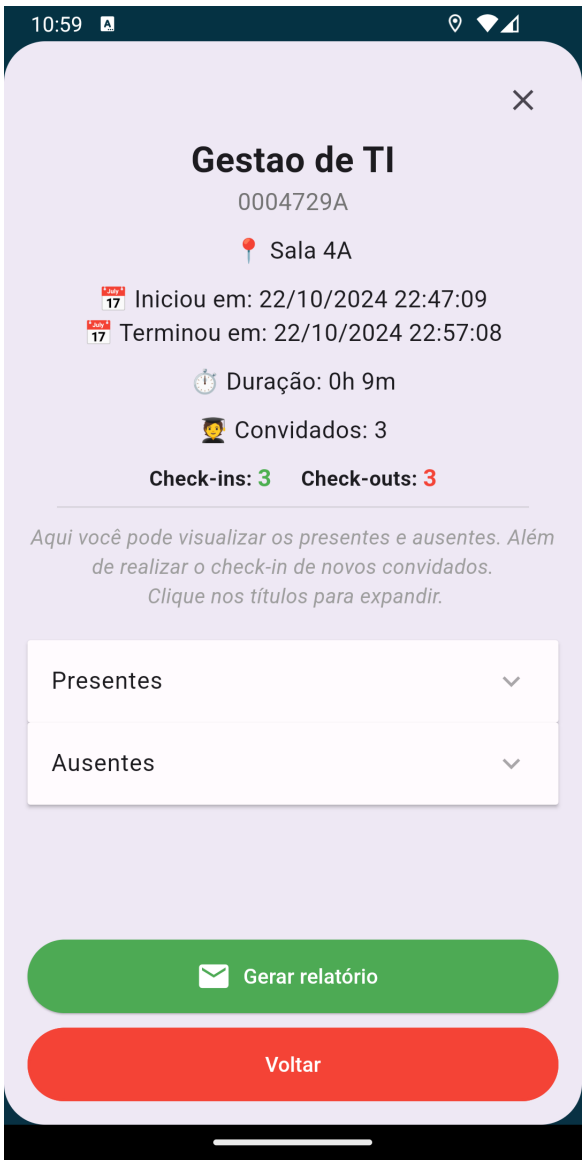
2. Cartões: Os cartões são o principal elemento dessa tela, eles apresentam algumas informações básicas sobre o evento em questão, como nome, data e local. Ao clicar no cartão será exibida uma tela de informações mais detalhadas sobre o evento, conforme as Figuras 23 e 24.

Figura 22 – Tela Histórico de Eventos



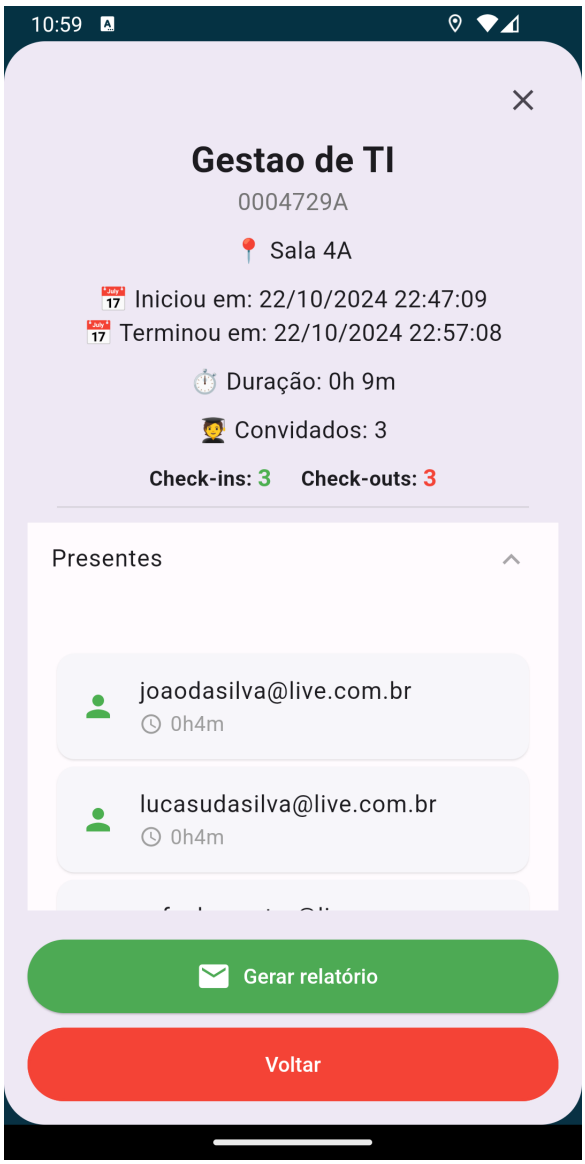
Fonte: Elaborada pelo autor.

Figura 23 – Tela Histórico de Eventos (Cartão Selecionado)



Fonte: Elaborada pelo autor.

Figura 24 – Tela Histórico de Eventos (Visualização de Convidados Presentes)



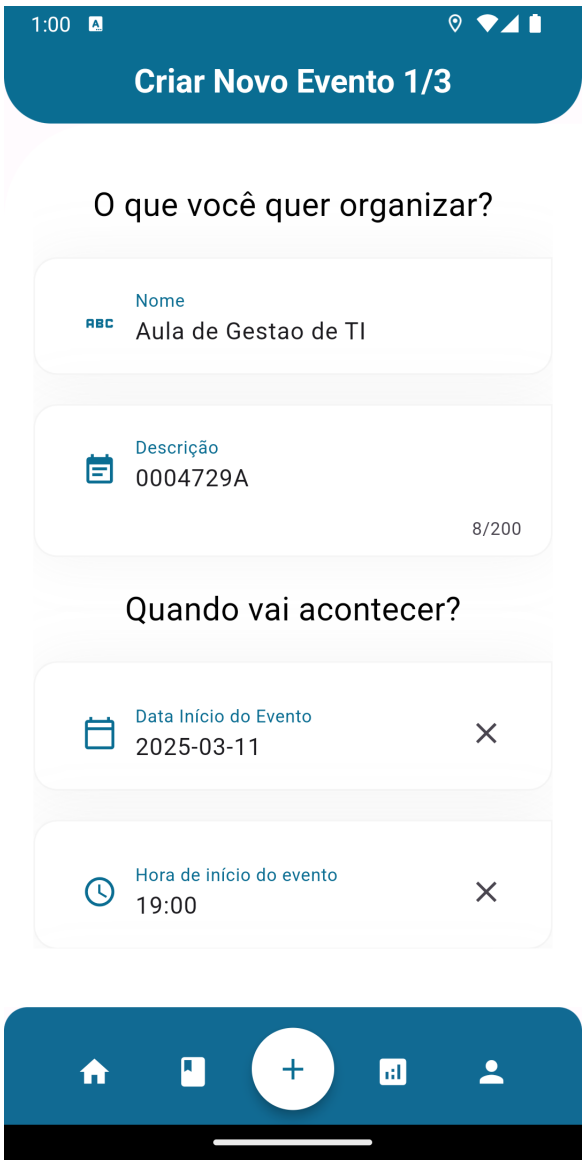
Fonte: Elaborada pelo autor.

6.9 Cadastro de Eventos

Ao clicar no ícone de "+", da *Bottom App Bar* mencionada na Seção 6.3, será exibida a tela de cadastro de novos eventos. Esse fluxo é dividido em três etapas, sendo a primeira pra preenchimento de informações básicas de evento, como nome, local, data. E a segunda, para parametrizações, como a distância máxima permitida de um convidado em relação ao evento, e por último, o preenchimento dos usuários convidados, à partir do e-mail.

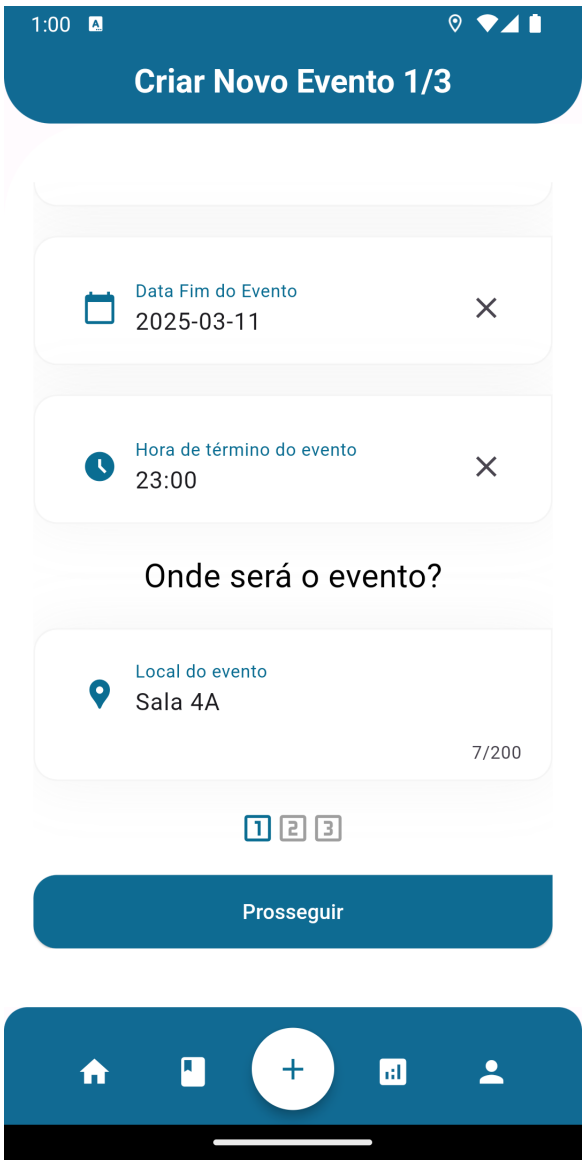
As Figuras 25 e 26 demonstram a disposição desses elementos na etapa 1 de cadastro de novos eventos.

Figura 25 – Tela de Cadastro de Eventos (Etapa 1 - (1/2))



Fonte: Elaborada pelo autor.

Figura 26 – Tela de Cadastro de Eventos (Etapa 1 - (2/2))

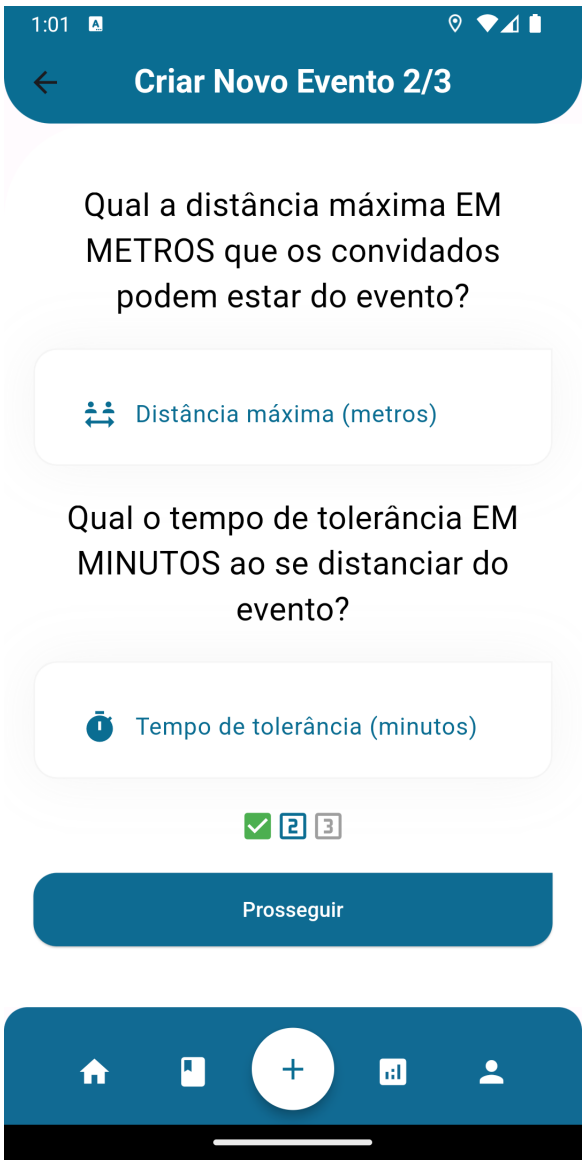


Fonte: Elaborada pelo autor.

Ao preencher todas as informações obrigatórias, o usuário deve clicar no botão "Prosseguir", o que o leverá para a tela da segunda etapa, onde devem ser preenchidos os campos de parametrização.

A Figura 27 demonstra a disposição dos elementos na etapa 2 de cadastro de novos eventos.

Figura 27 – Tela de Cadastro de Eventos (Etapa 2)

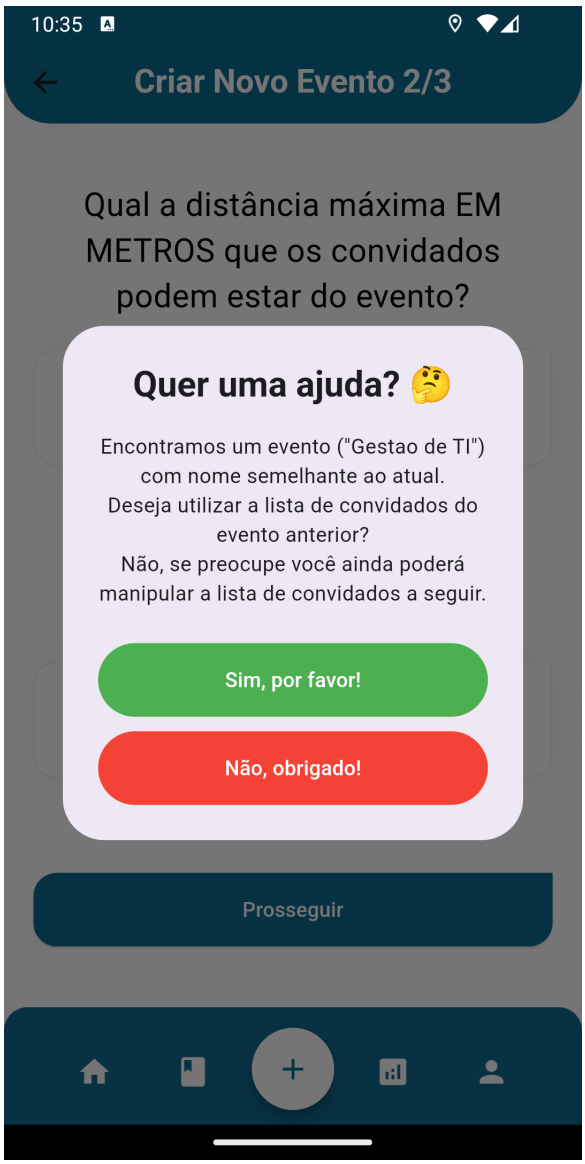


Fonte: Elaborada pelo autor.

Após preencher todas as informações obrigatórias e clicar no botão "Prosseguir", caso não haja nenhum erro ou mensagem do sistema, o usuário será encaminhado para a tela da terceira etapa, onde devem ser preenchidos os convidados do evento, através do e-mail.

Caso o usuário já tenha cadastrado algum evento com nome similar, o sistema exibirá uma mensagem questionando se deseja utilizar a mesma lista de convidados. A Figura 28 demonstra a mensagem em questão.

Figura 28 – Tela de Cadastro de Eventos (Etapa 3 - Mensagem)



Fonte: Elaborada pelo autor.

Caso o usuário opte por usar a mesma lista de convidados, ele ainda poderá manipular a lista de convidados, incluindo ou excluindo algum convidado. Se optar por não utilizar a mesma lista, seguirá para a tela de próxima etapa para preenchimento dos convidados.

Ao preencher usuários, serão criados cartões abaixo do botão “Adicionar” onde é possível remover um convidado da lista. A Figura 29 demonstra a disposição da tela com a adição dos cartões de convidados.

Figura 29 – Tela de Cadastro de Eventos (Etapa 3 - Convidados)

Fonte: Elaborada pelo autor.

O preenchimento dos convidados é obrigatório, e caso o usuário tente prosseguir sem convidados, será exibida uma mensagem informativa, conforme evidenciado pela Figura 30.

Figura 30 – Tela de Cadastro de Eventos (Etapa 3 - Mensagem)

Fonte: Elaborada pelo autor.

Aoós o preenchimento dos usuário, o usuário poderá clicar em "Prosseguir", realizando uma requisição de criação do evento. Dessa forma, o evento será criado e todos os usuários convidados serão notificados através do seu e-mail, incluindo usuários que ainda não possuam cadastro no sistema.

6.10 Perfil

Ao clicar no ícone de "usuário" (primeiro, da direita para esquerda), da *Bottom App Bar* mencionada na Seção 6.3, será exibida a tela de perfil do usuário.

A tela de perfil é por onde o usuário terá acesso à informações sobre o seu cadastro,

como nome, e-mail, e outros. Além de poder realizar ações como alterar sua senha, ainda alterar seu e-mail, ou ainda, deslogar sua conta do aplicativo.

A Figura 31 demonstra a disposição desses elementos na tela de Perfil.

Figura 31 – Tela de Perfil

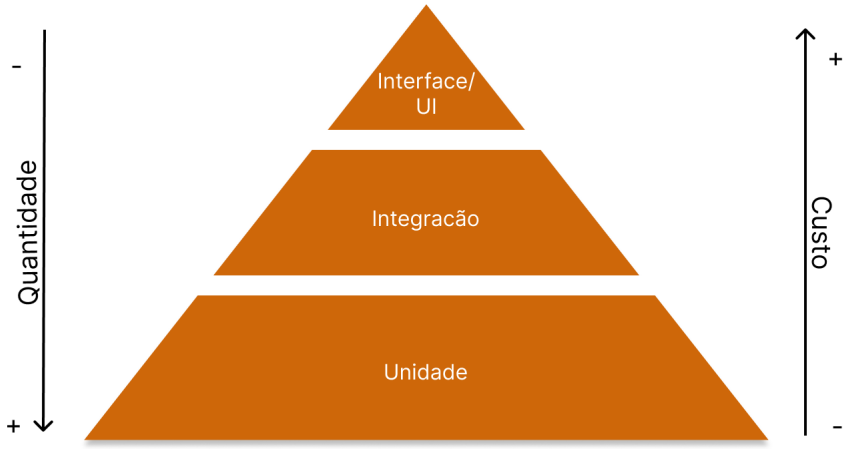
Fonte: Elaborada pelo autor.

7 Testes

Para garantir a qualidade e a funcionalidade do sistema desenvolvido, a estratégia de teste adotada é composta por várias técnicas, incluindo testes unitários, testes de integração e testes exploratórios.

A Figura 32 trás uma representação visual da estrutura dos testes, além da relação entre quantidade e custo entre os tipos de teste adotados nesse projeto.

Figura 32 – Pirâmide de Testes

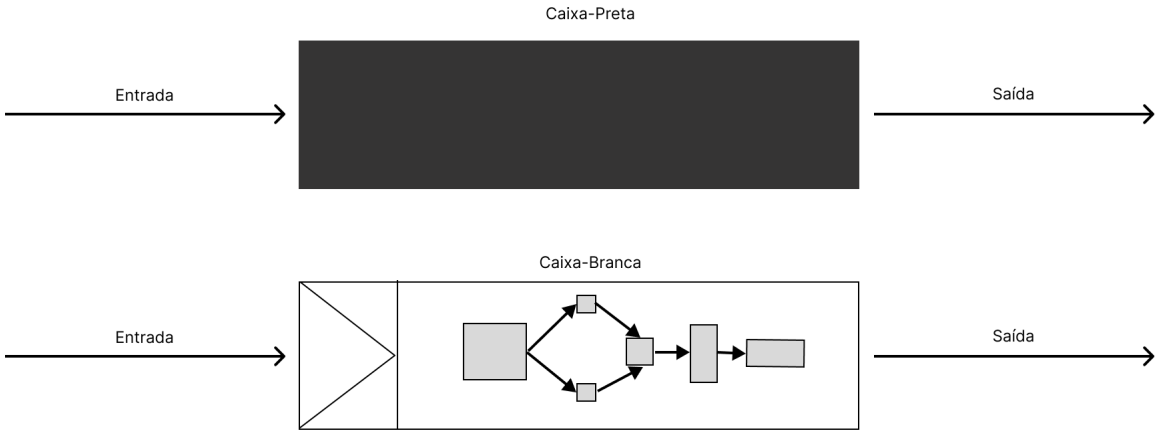


Fonte: Elaborada pelo autor.

Com exceção do testes exploratórios, as demais estratégias aplicam o modelo de caixa-branca, que são uma técnica de teste que se concentra em verificar a funcionalidade interna de um módulo específico. Ao contrário dos testes de caixa preta, que se concentram apenas nas entradas e saídas de um módulo, os testes de caixa branca levam em consideração a estrutura interna e o funcionamento do módulo.

A Figura 33 trás uma comparação visual entre os testes caixa-branca e caixa-preta.

Figura 33 – Testes Caixa-Branca x Testes Caixa-Preta



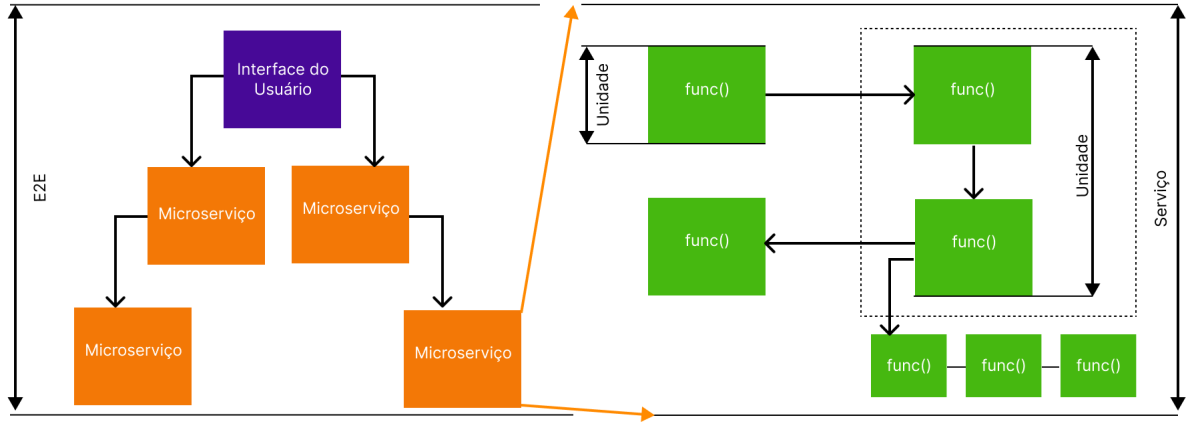
Fonte: Elaborada pelo autor.

7.1 Testes de Unidade

Os testes de unidade são projetados para verificar se um bloco de código específico é executado conforme o esperado, de acordo com a lógica do desenvolvedor por trás dele. Eles interagem com o bloco de código por meio de entradas e saídas declaradas, capturando resultados verdadeiros ou falsos. Cada bloco de código pode ter um conjunto de testes unitários, conhecidos como casos de teste. Um conjunto completo de casos de teste abrange todo o comportamento esperado do bloco de código, embora nem sempre seja necessário definir esse conjunto completo.

A Figura 34 ilustra a composição de uma unidade de software, que são os componentes testados nos testes de unidade.

Figura 34 – Unidade de Software



Fonte: Elaborada pelo autor.

No contexto deste projeto, os testes unitários foram realizados utilizando o Jest[1], um framework de testes JavaScript amplamente utilizado. O objetivo desses testes é garantir que cada unidade de software funcione conforme esperado em uma variedade de cenários.

Como um exemplo de parte prática, o código de um caso de teste utilizando Jest, onde é verificado a existência de objetos com chave duplicada à partir de uma lista de objetos fornecida. A Figura 35 demonstra um exemplo de um caso de teste desenvolvido utilizando o framework Jest.

[1] <https://jestjs.io/pt-BR/docs/getting-started>

Figura 35 – Caso de teste utilizando Jest

```
// Caso de teste usando Jest
describe('isUniqueByKey', () => {
    it('deve retornar objetos unicos com base nas chaves especificadas', => {
        // Array de objetos de teste
        const objects = [
            { id: 1, name: 'John', age: 30 },
            { id: 2, name: 'Alice', age: 25 },
            { id: 3, name: 'John', age: 35 },
            { id: 4, name: 'Bob', age: 30 },
            { id: 5, name: 'Alice', age: 25 }
        ];

        // Chaves para verificar a unicidade
        const keysToCheck = ['name', 'age'];

        // Resultado esperado
        const expectedUniqueObjects = [
            { id: 1, name: 'John', age: 30 },
            { id: 2, name: 'Alice', age: 25 },
            { id: 3, name: 'John', age: 35 }
        ];

        // Chamar a função que queremos testar
        const uniqueObjects = isUniqueByKey(objects, keysToCheck);

        // Verificar se o resultado da função é o esperado
        expect(uniqueObjects).toEqual(expectedUniqueObjects);
    });
});
```

Fonte: Elaborada pelo autor.

Após a implementação de uma bateria de testes unitários, o projeto atingiu um índice de cobertura de 99,73% das linhas de código 96.87% de cobertura das ramificações. Esse nível elevado de cobertura permite um alto grau de confiança na estabilidade do código, minimizando o risco de falhas inesperadas em produção e facilitando a refatoração de componentes no futuro.

7.2 Testes de Integração

Os testes de integração são uma fase crucial do processo de teste, onde módulos individuais são combinados e testados como um grupo. O objetivo dos testes de integração

é identificar quaisquer problemas que possam ocorrer quando diferentes módulos interagem entre si (CHERNYAK, 2022).

Os testes de integração podem ser realizados de várias maneiras, dependendo da complexidade do sistema e das necessidades específicas do projeto. Para o projeto proposto nesse documento, a estratégia utilizada será a de integração incremental, no modelo Top-down (de cima para baixo). Nesta abordagem, os módulos são adicionados um de cada vez e testados em conjunto, começa-se com os módulos de nível superior e, em seguida, os módulos de nível inferior são adicionados um de cada vez.

No contexto do Back-End, os testes de integração foram realizados de duas maneiras. Em algumas partes, o framework Jest foi utilizado, semelhante aos testes de unidade, mencionados anteriormente, permitindo a execução automatizada de cenários de integração específicos entre módulos. Isso assegurou que as funcionalidades principais estivessem integradas corretamente e que as respostas do sistema atendessem às expectativas. Em outras áreas, foram realizados testes manuais seguindo a abordagem incremental top-down, na qual a integração de módulos acontecia gradualmente, começando pelos componentes de nível superior e descendo aos níveis inferiores.

Para a interface e o aplicativo, os testes de integração também foram realizados de forma manual, garantindo que os componentes de front-end estivessem em conformidade com os requisitos de interação entre si e com o backend.

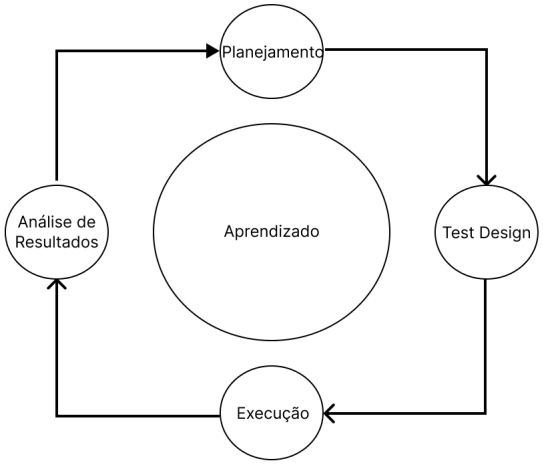
O resultado desse processo foi altamente positivo. A abordagem permitiu identificar uma série de problemas de compatibilidade e comunicação entre os módulos. Esses problemas, que poderiam gerar inconsistências e falhas de comunicação em produção, foram detectados e corrigidos a tempo, aumentando a confiabilidade e a robustez do sistema final.

7.3 Testes Exploratórios

Os testes exploratórios são uma abordagem de teste não estruturada e empírica, onde os testadores exploram o sistema para descobrir erros ou problemas. Neste tipo de teste, os testadores exploram o sistema de forma livre, sem um roteiro predefinido, e adaptam seus testes com base no que descobrem durante o processo de teste (SANTOS, 2018).

A Figura 36 ilustra a lógica por trás da execução de testes exploratórios.

Figura 36 – Testes Exploratórios



Fonte: Elaborada pelo autor.

Os testes exploratórios, realizados de forma manual, foram essenciais para encontrar erros em fluxos de baixa frequência, ou seja, em trajetórias da aplicação que não representam o caminho principal ou “caminho feliz”. O termo "caminho feliz"refere-se ao fluxo ideal em que os usuários percorrem a aplicação sem encontrar obstáculos ou comportamentos inesperados, passando apenas pelas etapas esperadas e com as entradas corretas.

Por meio destes, foi possível simular cenários onde o usuário segue rotas alternativas ou executa ações fora do esperado, revelando bugs e inconsistências que, de outra forma, poderiam passar despercebidos em um teste convencional.

8 Conclusão

8.1 Desafios

Durante o desenvolvimento deste projeto, enfrentei diversos desafios que influenciaram diretamente tanto o escopo quanto as funcionalidades finais entregues. Como ponto de partida, destaco que a escolha de Flutter como framework de desenvolvimento foi estratégica pela sua versatilidade em permitir o desenvolvimento de aplicativos multiplataforma. No entanto, por ser uma tecnologia nova para mim, sem experiência anterior com Flutter, além da minha falta de prática com layouts e desenvolvimento Front-End, o processo de aprendizado e adaptação exigiu mais tempo e dedicação do que o inicialmente planejado.

Um dos problemas mais marcantes envolveu o uso do emulador, que me forçou a modificar parte do escopo original. Uma funcionalidade prevista inicialmente, como a biometria facial para a confirmação da presença, foi deixada de lado devido a dificuldades técnicas com a integração da câmera do Flutter no emulador Android. Essa questão limitou a implementação dessa tecnologia e a mesma foi redirecionada para trabalhos futuros, pois não consegui resolver os problemas dentro do prazo estipulado.

Outro desafio significativo foi lidar com os múltiplos casos alternativos no gerenciamento da presença dos convidados. Situações como check-outs automáticos exigiram maior complexidade na lógica de controle, já que precisei prever e tratar fluxos não triviais, como a geolocalização em áreas limítrofes da sala de aula e a necessidade de tratar inconsistências de conexão durante o monitoramento em tempo real.

Além disso, tive dificuldades para implementar notificações push, o que novamente foi relacionado aos problemas com o emulador. Contudo, o maior empecilho foi a limitação de tempo e escopo para esta entrega específica. Devido à necessidade de apresentar uma versão funcional e dentro do prazo à banca, precisei optar por uma solução mais simples e de rápida implementação, como o envio de notificações por e-mail, deixando as notificações push para versões futuras do aplicativo.

Esses desafios, embora complexos, trouxeram aprendizados valiosos sobre o processo de desenvolvimento de software, especialmente no que diz respeito a novas tecnologias e às restrições práticas impostas por ambientes de desenvolvimento e testes.

8.2 Resultados

Os resultados obtidos com este projeto refletem os objetivos iniciais, bem como os desafios enfrentados ao longo do desenvolvimento. Em termos de funcionalidade, o sistema

alcançou uma taxa de cobertura de código alta pelos testes automatizados incluindo unitários e de integração, além dos testes manuais que foram essenciais para detectar e corrigir possíveis falhas, tanto em fluxos principais quanto em trajetórias alternativas. Com isso, o sistema demonstrou alta confiabilidade, especialmente nas funcionalidades de controle de presença e monitoramento em tempo real, que operaram com sucesso em condições variadas de uso, levando em consideração a margem de erro para precisão da geolocalização baseada em GPS, que costuma ser de 5 até 20 metros, dependendo da configuração feita. Para essa solução, foi utilizado a configuração *priority high accuracy*[1], que garante uma maior precisão.

Na prática, os resultados mostraram que o sistema é capaz de monitorar, controlar e registrar a presença de usuários de forma eficiente, atendendo aos requisitos estabelecidos para o ambiente de sala de aula. Em cenários com conexão de rede instável, foi mitigar possíveis inconsistências de maneira eficaz, confirmando a viabilidade de sua aplicação em ambientes onde a conectividade pode variar.

No tocante à segurança e à conformidade com a Lei Geral de Proteção de Dados (LGPD), foram adotadas medidas para resguardar os dados sensíveis dos usuários, tais como a criptografia das informações e um controle de acesso rigoroso. Além disso, assegurou-se a transparência no uso dos dados de geolocalização, obtendo o consentimento informado dos usuários no primeiro acesso ao aplicativo.

Adicionalmente, a estrutura desenvolvida oferece uma base sólida para expansão a outros domínios. A arquitetura modular e as boas práticas de design adotadas permitirão integrar futuras funcionalidades, como notificações push e reconhecimento facial de forma ágil e escalável. Estes resultados destacam o potencial de evolução do sistema para se adaptar a diferentes contextos e necessidades, desde ambientes educacionais até eventos de grande porte, com possibilidade de monetização em aplicações comerciais.

Esses resultados confirmam a validade do sistema proposto e destacam as melhorias realizadas em relação ao planejamento inicial, consolidando o projeto como uma solução tecnológica viável e promissora para monitoramento e gestão de presença.

8.3 Trabalhos Futuros

Com base nos desafios enfrentados e nas escolhas feitas durante o desenvolvimento deste projeto, diversas oportunidades de melhoria e expansão foram identificadas, as quais podem ser exploradas em futuras versões do sistema. Abaixo, estão descritos os principais pontos que podem guiar o desenvolvimento futuro:

- Integração de Biometria Facial: Como mencionado anteriormente, um dos maiores desafios enfrentados no desenvolvimento foi a implementação da biometria facial para o controle

[1] <https://developers.google.com/android/reference/com/google/android/gms/location/Priority>

de presença. Devido a problemas técnicos relacionados à funcionalidade de câmera no emulador, a implementação foi adiada para trabalhos futuros. Com a solução adequada desses problemas, será possível adicionar essa funcionalidade para melhorar a segurança e a precisão na identificação dos usuários.

- Notificações Push: Embora o sistema atual utilize notificações via e-mail para alertar os usuários sobre atualizações e mudanças, a adição de notificações push está prevista para futuras versões. A implementação dessa funcionalidade será especialmente útil para garantir uma comunicação mais ágil e direta com os usuários, sem depender de plataformas externas de e-mail.

- Expansão para Outros Domínios: Embora o foco inicial do sistema tenha sido o ambiente de sala de aula, a arquitetura e os conceitos desenvolvidos podem ser facilmente aplicados a outros domínios e contextos. Por exemplo, o sistema pode ser adaptado para controlar a presença em eventos como conferências, feiras, shows, workshops, ou qualquer outro tipo de encontro em que o monitoramento da presença seja relevante. Esta expansão representaria um grande avanço em termos de versatilidade e impacto do sistema, com diversas formas de monetização.

- Integração com Outras Tecnologias: Outra oportunidade de melhoria significativa está na integração do sistema com plataformas e softwares já utilizados por instituições de ensino, como sistemas de gestão acadêmica das universidades. A automatização do envio de relatórios de presença para esses sistemas reduziria a necessidade de intervenção manual, tornando o processo mais eficiente e menos suscetível a erros humanos. Além disso, o sistema pode se integrar a soluções de automação de eventos em outros setores, ampliando sua aplicabilidade.

- Melhoria na Interface e Experiência do Usuário (UX): Durante o desenvolvimento, um dos grandes desafios foi a criação de uma interface agradável e intuitiva, dado que a experiência com desenvolvimento Front-End era limitada. Trabalhos futuros podem focar em refinar o design responsivo para garantir que o aplicativo funcione perfeitamente em diferentes tamanhos de tela e orientações, além de otimizações de performance e acessibilidade.

- Mais Opções de Configuração para Eventos: Para aumentar a flexibilidade do sistema, futuras iterações podem incluir opções adicionais para a configuração dos eventos. Isso inclui a definição de um tempo mínimo de permanência para que o participante seja considerado presente ou elegível para receber certificados, além da criação de eventos recorrentes, que simplificariam o planejamento e a gestão de séries de encontros regulares.

Em suma, as oportunidades para o aprimoramento do sistema são vastas. As melhorias propostas nesta seção visam não apenas ampliar o escopo de aplicação, mas também tornar o

sistema mais eficiente, fácil de usar, e capaz de atender a novas demandas tecnológicas e de usabilidade, garantindo sua relevância no longo prazo.

Referências

ALFF, F. R. O que são requisitos de software. *O que são requisitos de software*, 2018. Acesso em: 20 abr. 2024. Disponível em: <https://analisederequisitos.com.br/requisitos-de-software/#:~:text=Tecnicamente%2C%20conceitua-se%20que%20um%20requisito%20de%20software%20%C3%A9,de%20aceite%2C%20para%20atender%20%C3%A0s%20necessidades%20dos%20stakeholders.>

ALFF, F. R. Diagramas uml: O que são? conheça os 14 tipos. *Diagramas UML*, 2022. Acesso em: 20 abr. 2024. Disponível em: <https://analisederequisitos.com.br/o-que-sao-diagramas-uml/>.

CETIC.BR. Tic domicílios. *TIC Domicílios 2023*, 2023. Acesso em: 25 mar. 2024. Disponível em: <https://cetic.br/media/analises/tic_domicilios_2023_coletiva_imprensa.pdf>.

CHERNYAK, A. Z. O que são testes de integração? mergulho profundo nos tipos, processo e implementação. *Medium*, 2022. Acesso em: 04 abr. 2024. Disponível em: <https://www.zaptest.com/pt-br/o-que-sao-testes-de-integracao-mergulho-profundo-nos-tipos-processo-e-implementacao>.

DESIGN, M. App bars: bottom. *App bars: bottom*, 2024. Acesso em: 29 jun. 2024. Disponível em: <https://m2.material.io/components/app-bars-bottom>.

DEV, A. B. Android ui: Bottom sheet. *Android UI: Bottom Sheet*, 2019. Acesso em: 29 jun. 2024. Disponível em: <https://medium.com/android-dev-br/android-ui-bottom-sheet-4709cad826d2>.

DEVMEDIA. Diagrama de classes uml. *Diagrama de Classes UML*, 2024. Acesso em: 20 abr. 2024. Disponível em: <https://www.devmedia.com.br/diagrama-de-classes-uml/12251>.

DEVMEDIA. Mer e der: Modelagem de bancos de dados. *MER*, 2024. Acesso em: 20 abr. 2024. Disponível em: <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>.

DEVMEDIA. O que é uml e diagramas de caso de uso: Introdução prática à uml. *O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML*, 2024. Acesso em: 20 abr. 2024. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>.

DEVMEDIA. Utilizando uml: Diagrama de atividade. *Artigo SQL Magazine 66*, 2024. Acesso em: 20 abr. 2024. Disponível em: <https://www.devmedia.com.br/diagrama-de-classes-uml/12251>.

FACTORIAL. *Factorial RH*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://factorialhr.com.br/>.

HELLOMIDIA. Marketing de geolocalização: o que é, vantagens e exemplos. *Hello Mídia*, 2023. Acesso em: 02 abr. 2024. Disponível em: <https://hellomidia.com.br/marketing-de-geolocalizacao-o-que-e-vantagens-e-exemplos/>.

SANTOS, L. Testes exploratórios. *Medium*, 2018. Acesso em: 02 abr. 2024. Disponível em: <https://medium.com/@lazarofsjunior/teste-explorat%C3%B3rios-a0a0f4265154>.

SERPRO, C. in. *Check-in Serpro*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://checkin.serpro.gov.br/>.

SIMPAX.GEOLOC.MULTI. *Simpax*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://simpax.com.br/geoloc-multi-aplicativo-de-ponto-e-presenca/>.

SOUZA, P. Caso de uso - o que é, propósito, no desenvolvimento e requisitos. *Caso de uso - O que é, propósito, no desenvolvimento e requisitos*, 2023. Acesso em: 20 abr. 2024. Disponível em: <https://conceito.de/caso-de-uso>.

SYMPLA. *Sympla*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://www.sympla.com.br/>.

TOTVS. Controle de frequência escolar: Como um sistema ajuda? *Controle de frequência escolar: Como um sistema ajuda?*, 2020. Acesso em: 20 abr. 2024. Disponível em: <https://www.totvs.com/blog/instituicao-de-ensino/controle-de-frequencia-escolar/>.

TOTVS. *Educacional TOTVS*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://www.totvs.com/educacional/>.

TOTVS. *TOTVS*. 2024. Acesso em: 02 abr. 2024. Disponível em: <https://www.totvs.com/>.