CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia Baixada Santista Rubens Lara

Curso Superior de Tecnologia em Sistemas para Internet

Gabriel Pimentel de Carvalho
Lucas Martins Vargas do Amaral

SETA
Sistema Especializado de Tráfego Automatizado

Santos, SP
2025

Gabriel Pimentel de Carvalho
Lucas Martins Vargas do Amaral

SETA
Sistema Especializado de Tráfego Automatizado

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Joseffe Barroso de Oliveira

Santos, SP
2025

RESUMO

O controle de acesso em instituições é fundamental para garantir a segurança e a organização do ambiente, porém os métodos tradicionais podem ser lentos e suscetíveis a falhas. A necessidade de modernizar o processo de autenticação e controlar rigorosamente quem tem permissão para acessar ambientes físicos impulsiona o desenvolvimento de soluções tecnológicas eficientes e seguras. O SETA foi criado com o objetivo de desenvolver um sistema tecnológico que utilize *QR Codes* dinâmicos para autenticar usuários, proporcionando agilidade, autonomia e confiabilidade no controle de acesso institucional. O sistema conta com uma interface intuitiva para os usuários e um *back-end* robusto para os administradores, integrando tecnologias que garantem a integridade dos dados, a segurança da autenticação e a restrição do acesso somente a pessoas autorizadas, com testes para validar a usabilidade e a segurança da aplicação. O SETA demonstra ser uma solução eficaz para modernizar o controle de acesso, facilitando a gestão por parte dos administradores e oferecendo aos usuários uma experiência segura e fluida, contribuindo para a melhoria da segurança e organização nas instituições.

Palavras-chave: Controle de acesso. *QR Code* dinâmico. Segurança. Autenticação.

ABSTRACT

Access control in institutions is essential to ensure the security and organization of the environment. However, traditional methods can be slow and prone to failures. The need to modernize the authentication process and strictly control who is authorized to access physical spaces drives the development of efficient and secure technological solutions. SETA was created with the aim of developing a technological system that uses dynamic QR codes to authenticate users, providing speed, autonomy, and reliability in institutional access control. The system features an intuitive interface for users and a robust back-end for administrators, integrating technologies that ensure data integrity, authentication security, and access restriction exclusively to authorized individuals, with tests conducted to validate the application's usability and security. SETA proves to be an effective solution for modernizing access control, facilitating management for administrators and offering users a secure and seamless experience, contributing to the improvement of security and organization within institutions.

Keywords: Access control. Dynamic QR Code. Security. Authentication.

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Atualmente, a segurança e o controle de acesso em ambientes institucionais, como universidades, empresas e órgãos públicos, tornaram-se temas de grande relevância. Segundo Stallings (2014), de um modo geral, a segurança da informação é um dos pilares fundamentais das organizações modernas, abrangendo, principalmente, a proteção de dados. Nesse contexto, os métodos tradicionais de controle de acesso — como listas manuais, crachás físicos ou simples conferências visuais — tornam-se cada vez mais obsoletos, suscetíveis a falhas e ineficientes frente às demandas atuais. Conforme Benantar (2006), a autenticação baseada em cartões magnéticos, senhas ou biometria, apesar de amplamente utilizada, apresenta limitações relacionadas à segurança física, clonagem e escalabilidade.

O avanço da transformação digital tem proporcionado a esses ambientes a possibilidade de substituir tais métodos por sistemas informatizados mais robustos, seguros e eficientes. Conforme Tanenbaum e Wetherall (2011), sistemas distribuídos e aplicações em rede oferecem flexibilidade e escalabilidade, sendo essenciais na construção de soluções modernas que envolvem autenticação e controle de acesso. Anderson (2008) complementa que sistemas distribuídos dependem de engenharia de segurança robusta e mecanismos de autenticação que preservem a integridade das permissões sem comprometer a usabilidade.

No contexto universitário, essa necessidade se torna ainda mais evidente. As instituições de ensino superior frequentemente lidam com um fluxo constante e elevado de pessoas, entre alunos, professores, colaboradores e visitantes, o que torna indispensável a adoção de soluções tecnológicas que permitam um controle de acesso eficiente, ágil e confiável. Segundo Rezende e De Abreu (2003), os ambientes acadêmicos possuem características dinâmicas que exigem sistemas de gestão e controle que garantam segurança física, sem prejudicar a mobilidade e a autonomia dos usuários.

Contudo, muitos dos sistemas disponíveis no mercado apresentam altos custos de implementação, além de exigirem integração direta com as bases de dados internas das instituições, tornando o processo burocrático e, em muitos casos, inviável, especialmente para instituições públicas que lidam com restrições orçamentárias. Neste contexto, é observável que soluções seguras não precisam

ser, necessariamente, onerosas, desde que se adote uma arquitetura bem planejada, baseada em princípios de segurança, autenticação forte e mínima dependência de infraestruturas complexas (ANDERSON, 2008; SILBERSCHATZ et al., 2019).

Diante desse cenário, este trabalho apresenta o desenvolvimento do SETA, um sistema de controle de acesso voltado para universidades, que tem como proposta oferecer uma solução tecnológica acessível, segura e escalável, utilizando *QR Codes* dinâmicos como mecanismo principal de autenticação. O uso desse recurso vem sendo amplamente adotado em soluções de autenticação, justamente pela facilidade de geração, leitura e pela possibilidade de embutir informações criptografadas ou temporárias (BENANTAR, 2006).

Um dos diferenciais do SETA está em sua independência da base de dados interna da instituição, permitindo que qualquer membro da comunidade universitária crie sua conta, desde que possua um e-mail institucional previamente autorizado. Dessa forma, elimina-se a necessidade de integrações complexas com os sistemas administrativos da universidade, mantendo a segurança e a restrição de acesso por meio de validações lógicas, garantindo acesso seguro sem sobrecarregar as instituições com processos burocráticos.

O SETA é composto por três módulos principais: uma *API back-end* desenvolvida em *Spring Boot*, responsável por toda a lógica de negócios, autenticação, geração e validação dos *QR Codes*; um painel administrativo construído com *React*, que permite aos administradores gerenciar usuários, acompanhar estatísticas, monitorar *logs* de acesso e controlar as permissões; e, por fim, um aplicativo *mobile* desenvolvido em *React Native*, no qual os usuários podem se cadastrar, gerar *QR Codes* de acesso único e interagir com o sistema de forma autônoma e segura. Toda a comunicação entre as aplicações ocorre por meio de *tokens JWT* (*JSON Web Token*), uma tecnologia descrita por Jones et al. (2015) como um formato compacto de representação de declarações destinado a ambientes com restrições de espaço, como cabeçalhos de autorização *HTTP* e parâmetros de consulta *URI*, sendo uma forma robusta, segura e eficiente para autenticação *stateless* em aplicações distribuídas.

Diante do exposto, o problema central que norteia este trabalho é: como desenvolver uma solução de controle de acesso que seja segura, eficiente, de baixo custo, independente dos dados internos da instituição e que, ao mesmo tempo,

garanta que apenas membros autorizados da comunidade universitária possam acessar os ambientes físicos da universidade? A busca pela resposta a essa questão orienta o desenvolvimento do SETA, cuja proposta visa suprir as deficiências observadas nos métodos tradicionais de controle de acesso e oferecer uma alternativa viável para instituições de ensino que demandam maior autonomia e segurança em seus processos de gestão de acesso.

1.1 OBJETIVO

Este item tem como propósito apresentar, de forma clara, tanto o objetivo geral quanto os objetivos específicos do presente trabalho. A definição dos objetivos é essencial para nortear as etapas do desenvolvimento, bem como para delimitar os resultados esperados com a implementação do sistema proposto. Dessa maneira, busca-se garantir que todas as ações realizadas estejam alinhadas à resolução do problema central, além de contribuírem para o avanço no campo de estudo relacionado à segurança e controle de acesso em ambientes universitários.

1.1.1 OBJETIVO GERAL

O objetivo geral do SETA é desenvolver uma solução tecnológica para o controle de acesso eficiente, seguro, prático e de baixo custo de implementação em universidades, utilizando QR Codes dinâmicos como forma de autenticação. O sistema busca facilitar a gestão de acessos por parte dos administradores e proporcionar uma experiência fluida e segura aos usuários, contribuindo diretamente para o aprimoramento da segurança e da organização institucional.

1.1.2 OBJETIVOS ESPECÍFICOS

- Realizar um levantamento dos principais métodos de controle de acesso utilizados atualmente em instituições de ensino.
- Analisar os requisitos de segurança, usabilidade e desempenho necessários para o desenvolvimento da solução proposta.

- Projetar e desenvolver uma *API back-end* robusta, utilizando *Spring Boot*, capaz de gerenciar autenticação, geração e validação de *QR Codes*, além do controle de usuários e *logs*.
- Implementar um painel administrativo em *React* para gestão dos usuários, monitoramento dos acessos e controle dos parâmetros do sistema.
- Desenvolver um aplicativo *mobile* com *React Native* + *Expo*, permitindo que os usuários realizem cadastro, autenticação e geração de *QR Codes* de entrada e saída
- Implementar mecanismos de segurança, como autenticação via *tokens JWT* e inutilização de *QR Codes* após o uso, visando mitigar fraudes e acessos indevidos.
- Garantir a segurança dos dados e das transações através da implementação de práticas modernas de autenticação e autorização.
- Documentar todo o processo de desenvolvimento, apresentando os desafios enfrentados e as soluções adotadas.

1.2 ESTADO DA ARTE

A segurança de acesso físico a instituições tem se tornado um dos principais focos de desenvolvimento tecnológico nos últimos anos. Com o aumento da circulação de pessoas em ambientes corporativos, acadêmicos e governamentais, a necessidade de soluções automatizadas, eficientes e seguras para controle de entrada e saída tornou-se indispensável. Dentre as tecnologias mais adotadas nesse cenário, destacam-se os sistemas baseados em cartões magnéticos, biometria, *RFID* e, mais recentemente, *QR Codes* dinâmicos (BENANTAR, 2006; KHATRI et al., 2023).

Soluções modernas como Portaria Remota 4.0, *iAccess* e TOTVS Controle de Acesso oferecem mecanismos para gerenciamento de visitantes, autenticação de usuários e registros de entrada e saída. A Portaria Remota 4.0, por exemplo, consiste no funcionamento automatizado de dispositivos para monitorar e controlar o acesso ao condomínio, com equipamentos integrados e gerenciados por um operador à distância, trazendo mais segurança (4PORT, 2023). O *iAccess* é uma aplicação *web* de controle e gestão de acessos, permitindo gerenciar usuários e dispositivos de acesso de forma eficiente (IACCESS, 2023). Já o sistema de controle

de acesso da TOTVS garante mais segurança com gerenciamento em tempo real, integrando-se aos sistemas TOTVS RH das linhas Protheus, Datasul e RM (TOTVS, 2023).

No entanto, muitos desses sistemas estão voltados para ambientes corporativos ou residenciais, com alto custo de implementação e dependência de infraestrutura robusta. Além disso, há limitação quanto à integração com plataformas institucionais e à personalização de funcionalidades conforme as rotinas específicas de uma instituição de ensino, por exemplo.

Neste contexto, o sistema SETA surge como uma solução inovadora, voltada especialmente para instituições que desejam modernizar e tornar mais seguro o controle de acesso físico de seus usuários. Utilizando autenticação via *e-mail* institucional, geração de *QR Codes* temporários e integração com banco de dados seguro, o SETA permite a validação rápida e confiável da identidade dos usuários na entrada da instituição.

Historicamente, os sistemas de controle de acesso evoluíram de métodos totalmente manuais, como listas físicas e crachás, para soluções baseadas em tecnologia de cartões magnéticos e posteriormente para *RFID* e biometria (BENANTAR, 2006), conforme Ilustração 01. Esses métodos, embora eficientes por um período, apresentam vulnerabilidades importantes, como clonagem de cartões, falsificação de credenciais e falhas em sensores biométricos em casos de desgaste físico ou condições ambientais desfavoráveis

Ilustração 01 – Evolução das formas de controle de acesso

Fonte: Autores (2025)

Nos últimos anos, o *QR Code* se consolidou como uma alternativa promissora por reunir características de baixo custo, fácil geração e leitura, além de

permitir integração com sistemas digitais. Entretanto, seu uso no controle de acesso exige cuidados específicos. *QR Codes* estáticos, por exemplo, podem ser facilmente copiados, compartilhados e reutilizados de maneira fraudulenta, tornando-os inseguros quando não acompanhados de mecanismos adicionais, como *tokens* dinâmicos, limite de validade e autenticação em tempo real (NJUGUNA; NDIA, 2025).

Nesta temática, uma grande referência no mercado nacional é a Eventim**,** uma plataforma consolidada no mercado de venda de ingressos, utilizada para eventos culturais, esportivos, *shows* e espetáculos em geral. Seu funcionamento é centrado na experiência digital, permitindo que os usuários adquiram ingressos diretamente por meio de uma aplicação para dispositivos móveis. O ingresso é disponibilizado no próprio aplicativo em formato de *QR Code*, conforme exemplificado na Ilustração 02, que deve ser apresentado na entrada do evento para validação por meio de leitores específicos. Essa abordagem visa praticidade, segurança e redução de fraudes, substituindo os ingressos físicos tradicionais.

Ilustração 02 – Visualização de ingresso na plataforma Eventim

Fonte: Eventim BR

No entanto, um ponto observado como fragilidade na plataforma é que, ao realizar a leitura do *QR Code* gerado, pode-se observar o seguinte texto: “00513205010025001001000O”. Como é possível perceber, não há uma estrutura robusta de segurança. Nesse caso apresentado, o resultado é um texto com pouca ou nenhuma criptografia significativa, que não fornece informações detalhadas ou protegidas sobre o ingresso, como o tipo (meia-entrada, idoso, estudante etc.) de maneira segura. Isso abre uma possibilidade de uso indevido, já que o controle sobre o direito ao benefício pode não ser realizado de forma eficiente no momento da validação.

2 DESENVOLVIMENTO

Este capítulo descreve as etapas do desenvolvimento do sistema proposto, abordando desde a análise dos requisitos até a definição da arquitetura de dados, camada de negócio e interface de apresentação. Assim, são apresentados os modelos conceituais, as tecnologias adotadas e as principais implementações realizadas no *back-end* e no *front-end*, de forma a proporcionar uma visão abrangente e detalhada do processo de desenvolvimento.

2.1 ANÁLISE DO SISTEMA

Para a realização da modelagem do SETA foi feita a escolha pela *Unified Modeling Language* (*UML*). A *UML*, que contém um total de 14 diagramas atualmente, é uma linguagem que permite a modelagem de vários aspectos de um *software*, seja na estrutura ou no comportamento. Com ela, podem ser realizadas, de forma visual, representações da interação entre usuário e o sistema, o relacionamento entre as classes do sistema etc. Por ser adotada desde 1997, como linguagem-padrão de modelagem pelo *Object Management Group* (*OMG*), uma organização internacional responsável por criar e manter padrões, a *UML* se tornou uma linguagem ampla e facilmente reconhecida na indústria. Guedes sobre os diagramas da *UML*:

> Cada diagrama da UML analisa o sistema, ou parte dele, sob uma determinada óptica. É como se o sistema fosse modelado em camadas. Alguns diagramas enfocam o sistema de forma mais geral, apresentando uma visão externa do sistema, como é o objetivo do Diagrama de Casos de Uso, enquanto outros oferecem uma visão de uma camada mais profunda do software, apresentando um enfoque mais técnico ou, ainda, visualizando apenas uma característica específica do sistema ou um determinado processo. A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros. (GUEDES, 2018)

2.1.1 ANÁLISE DE REQUISITOS

A análise de requisitos é uma etapa crítica no processo de desenvolvimento de *software*, pois envolve a identificação, documentação e validação das funcionalidades e restrições do sistema com base nas necessidades dos usuários e demais partes interessadas. Segundo Sommerville (2019), trata-se de um processo

interativo e de aprendizado, no qual os desenvolvedores exploram o problema a ser resolvido, considerando não apenas o que o sistema deve fazer, mas também o ambiente no qual ele irá operar. A qualidade dessa análise impacta diretamente o sucesso do sistema, uma vez que requisitos mal compreendidos levam inevitavelmente a falhas na implementação.

Nesse contexto, os requisitos funcionais (RF) são aqueles que definem as funcionalidades que o sistema deve possuir para atender às demandas dos usuários e dos administradores. Entendendo as necessidades do sistema SETA, os requisitos foram organizados considerando os diferentes perfis de usuários: administrador, super administrador e usuário do aplicativo. Desta maneira, as seguintes funcionalidades foram avaliadas como imprescindíveis:

Painel administrativo (administrador e super administrador) – tela de usuários:

[RF01] – Visualizar usuários – O sistema deve permitir que o administrador visualize a lista de usuários cadastrados;

[RF02] – Cadastrar usuário – O sistema deve permitir que o administrador cadastre novos usuários;

[RF03] – Editar usuário – O sistema deve permitir que o administrador edite os dados de usuários existentes;

[RF04] – Excluir usuário – O sistema deve permitir que o administrador exclua usuários do sistema;

Painel administrativo (administrador e super administrador) – tela de *logs*:

[RF05] – Visualizar *logs* de acesso – O sistema deve permitir que o administrador visualize os *logs* de acesso dos usuários;

Painel administrativo (administrador e super administrador) – tela de cadastros:

[RF06] – Visualizar solicitações de cadastro – O sistema deve permitir que o administrador visualize as solicitações de cadastro pendentes;

[RF07] – Reenviar *e-mail* de confirmação – O sistema deve permitir que o administrador reenvie o *e-mail* de confirmação de cadastro;

[RF08] – Excluir solicitação de cadastro – O sistema deve permitir que o administrador exclua solicitações de cadastro pendentes;

Painel administrativo (administrador e super administrador) – tela de *e-mails* permitidos:

[RF09] – Visualizar *e-mails* permitidos – O sistema deve permitir que o administrador visualize a lista de *e-mails* permitidos;

[RF10] – Cadastrar *e-mail* permitido – O sistema deve permitir que o administrador cadastre um novo *e-mail* permitido;

[RF11] – Editar *e-mail* permitido – O sistema deve permitir que o administrador edite os dados de *e-mails* permitidos;

[RF12] – Excluir *e-mail* permitido – O sistema deve permitir que o administrador exclua *e-mails* permitidos;

Painel administrativo (administrador e super administrador) – tela de suporte*:*

[RF13] – Visualizar mensagens de suporte – O sistema deve permitir que o administrador visualize as mensagens de suporte enviadas pelos usuários;

[RF14] – Excluir mensagens de suporte – O sistema deve permitir que o administrador exclua mensagens de suporte enviadas pelos usuários;

Painel administrativo (super administrador) – tela de *scanner*:

[RF15] – Acessar *scanner* de *QR Code* – O sistema deve permitir que o super administrador acesse a tela de *scanner* para leitura de *QR Code*;

[RF16] – Ler *QR Code* – O sistema deve permitir que o super administrador leia *QR Codes* e visualize o resultado da validação;

Aplicativo (usuário) – telas de cadastro:

[RF17] – Solicitar cadastro – O sistema deve permitir que o usuário realize uma solicitação de cadastro;

[RF18] – Definir senha – O sistema deve permitir que o usuário defina uma senha após a aprovação do cadastro;

Aplicativo (usuário) – tela de *QR Code*:

[RF19] – Visualizar *QR Code* – O sistema deve permitir que o usuário visualize seu *QR Code* gerado;

[RF20] – Gerar *QR Code* – O sistema deve permitir que o usuário gere um *QR Code* de acordo com seu tipo de acesso;

Aplicativo (usuário) – tela de suporte:

[RF21] – Enviar mensagem para suporte – O sistema deve permitir que o usuário envie mensagens para o suporte;

[RF22] – Visualizar mensagens de suporte – O sistema deve permitir que o usuário visualize as mensagens enviadas ao suporte;

[RF23] – Excluir mensagens de suporte – O sistema deve permitir que o usuário exclua mensagens que tenha enviado ao suporte;

Aplicativo (usuário) – tela de configuração:

[RF24] – Editar dados pessoais – O sistema deve permitir que o usuário edite seus dados pessoais da conta;

[RF25] – Solicitar exclusão da conta – O sistema deve permitir que o usuário solicite a exclusão da sua conta;

2.1.2 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso tem por objetivo representar, de maneira gráfica e semântica, as interações entre os atores (usuários do sistema) e os casos de uso correspondentes. Por meio desse diagrama, é possível visualizar de forma clara os limites do sistema, os perfis envolvidos e as funcionalidades acessíveis a cada tipo de usuário (SILVA; VIDEIRA, 2011).

O diagrama de caso de uso desse sistema pode ser visualizado no Apêndice A.

2.1.3 FLUXO DE EVENTOS

O fluxo de eventos descreve a sequência de atividades realizadas durante a execução de cada caso de uso, especificando o fluxo principal e, quando necessário, fluxos alternativos. Esta descrição tem por finalidade detalhar o comportamento esperado do sistema frente às ações dos usuários e possíveis exceções (PRESSMAN; MAXIM, 2016).

A descrição completa dos principais fluxos de eventos da aplicação está disponível no Apêndice B da presente monografia.

2.2 BANCO DE DADOS

Após a definição dos requisitos e da estrutura lógica do sistema, a etapa subsequente no desenvolvimento consiste na modelagem dos dados. Essa fase é fundamental, pois visa estruturar as informações de forma organizada, coerente e eficiente, garantindo não apenas o armazenamento, mas também a integridade, segurança e performance no acesso aos dados.

A modelagem de dados constitui uma etapa crucial no desenvolvimento de qualquer sistema, sendo responsável por traduzir os requisitos funcionais e não funcionais em uma representação formal das informações. O Modelo Entidade-Relacionamento (MER) representa um modelo de alto nível, abstrato e independente de um Sistema Gerenciador de Banco de Dados (SGBD), que permite descrever as entidades envolvidas no domínio do problema, seus atributos e os relacionamentos existentes entre elas. A partir dessa modelagem conceitual, torna-se possível evoluir para os modelos lógico e físico, que incorporam restrições específicas de um SGBD e requisitos técnicos da aplicação.

A elaboração de um banco de dados bem estruturado é um dos pilares para o bom desempenho e escalabilidade de qualquer aplicação. Conforme salienta Bazzi (2013), um banco de dados corretamente modelado possibilita a integração eficiente entre os módulos do sistema, assegura a consistência dos dados, reduz a redundância e proporciona maior facilidade na manutenção e evolução do *software*. Ademais, contribui diretamente para aspectos de segurança, controle de acesso, e performance nas operações de leitura e escrita.

O sistema proposto foi desenvolvido utilizando o *PostgreSQL*, um sistema gerenciador de banco de dados relacional de código aberto, amplamente reconhecido por sua robustez, escalabilidade e aderência aos padrões *SQL*. A interação entre a aplicação e o banco de dados ocorre por meio da tecnologia *Spring Data JPA*, que abstrai as operações de persistência, facilitando a comunicação entre os objetos Java e as tabelas do banco de dados, mediante o mapeamento objeto-relacional (*ORM*).

A definição das entidades foi baseada diretamente na análise dos requisitos do sistema. Foram identificadas, por exemplo, entidades como Usuário, Extensão de *E-mail*, *Token* Inválido, Nova Conta, Suporte e *Log* de Acesso, cada uma representando um elemento essencial para o funcionamento da aplicação. Além disso, foram definidos os relacionamentos necessários para refletir a dinâmica entre essas entidades.

Do ponto de vista físico, a modelagem foi realizada considerando as boas práticas de normalização, garantindo minimização de redundâncias e aumento na integridade dos dados. Paralelamente, algumas decisões foram tomadas no sentido de otimizar consultas específicas, como o uso de índices em colunas frequentemente filtradas ou ordenadas, e a definição de chaves estrangeiras para assegurar a integridade referencial.

Por fim, o MER (Modelo Entidade-Relacionamento) desenvolvido para este sistema está disponível no Apêndice C desta monografia. Nele, é possível observar graficamente as entidades, seus respectivos atributos, bem como os relacionamentos que refletem o funcionamento lógico da aplicação.

2.3 CAMADA DE NEGÓCIO

A camada de negócios desempenha um papel fundamental no desenvolvimento da aplicação, sendo responsável pela implementação das regras de negócio, garantindo que os processos da aplicação sejam executados de forma correta, segura e alinhada aos objetivos propostos.

No projeto desenvolvido, a camada de negócios foi construída utilizando os princípios da arquitetura *RESTful*, com uma divisão clara de responsabilidades, onde cada componente cumpre uma função específica. Esta organização favorece a escalabilidade, a manutenibilidade e a robustez do sistema.

As principais regras de negócio implementadas refletem os objetivos centrais da aplicação. Desta maneira nos parágrafos a seguir, estudos mais a fundo serão realizados sobre as características do SETA.

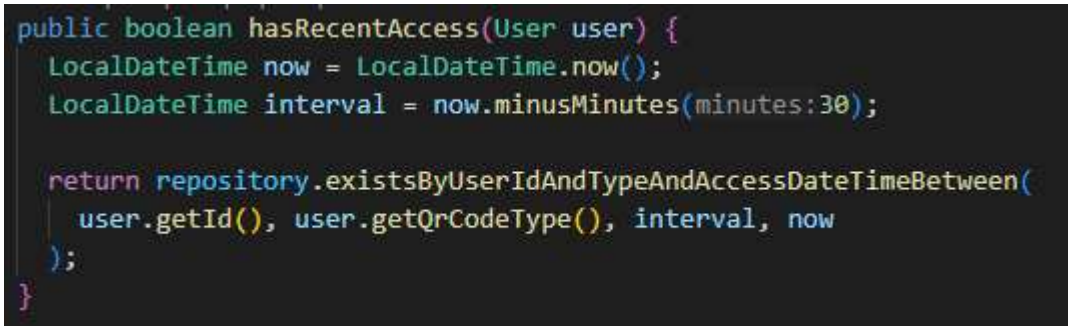
Como é possível observar nas Ilustrações 03, existe um controle de acesso por *e-mail* autorizado, dessa maneira, apenas usuários com *e-mails* previamente autorizados pelos administradores podem solicitar a criação de uma conta no sistema. Além disso, não é permitido que um mesmo *e-mail* envie múltiplas solicitações simultâneas de cadastro, evitando sobrecarga ou tentativas indevidas.

Ilustração 03 – Validação de solicitação na camada de serviço

Fonte: Autores (2025)

Para a geração de códigos de acesso, o sistema conta com uma validação rigorosa, impedindo que um usuário gere *QR Codes* do mesmo tipo (entrada ou saída) em um intervalo inferior a 30 minutos, Ilustração 04. Essa regra foi criada com o objetivo de mitigar tentativas de fraude.

Ilustração 04 – Validação de *QR Code* na camada de serviço



Fonte: Autores (2025)

Além dessa validação mencionada anteriormente, cada *token* de código de acesso gerado possui uma validade única de 3 minutos, como é possível notar no trecho de código da Ilustração 05. Caso não seja utilizado dentro do tempo limite, o token se tornará inválido, permitindo ao usuário gerar um novo código para aquele tipo de acesso. Essa medida visa mitigar a possibilidade de compartilhamento de *QR Codes* entre pessoas não autorizadas. Adicionalmente, uma vez utilizado, um *QR Code* se torna inválido, mesmo que pertença ao mesmo tipo de acesso, garantindo que não haja reutilização indevida.

Ilustração 05 – Camada de serviço de *tokens* de *QR Code*

```java
@Service
public class QRTokenService {

  private SecretKey key;
  @Value("${api.qrcode.token.secret}")
  private String SECRET;
  @Value("${api.qrcode.token.expiration}")
  private long EXPIRATION_TIME; //CONFIGURADO PARA 3 MINUTOS (180000 ms) NAS CONFIGURAÇÕES DO PROJETO

  Tabnine | Edit | Test | Explain | Document
  @PostConstruct
  public void init() {
    byte[] keyBytes = Decoders.BASE64.decode(SECRET);
    this.key = Keys.hmacShaKeyFor(keyBytes);
  }

  Tabnine | Edit | Test | Explain | Document
  public String generateToken(User user) {
    return Jwts.builder()
            .subject(user.getId().toString())
            .claim("email", user.getEmail())
            .claim("name", user.getName())
            .claim("type", user.getQrCodeType().name())
            Create Jira Issue
            .issuedAt(new Date())
            .expiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
            .signWith(key)
            .compact();
  }
```

Fonte: Autores (2025)

Como observado nos parágrafos anteriores, a camada de negócios é responsável pela geração de *tokens JWT* associados aos *QR Codes*. Esses *tokens* armazenam informações como o identificador do usuário, o tipo de acesso (entrada ou saída), o nome e o *e-mail* do usuário.

Para o processo de criação de conta, após realizar a solicitação, o usuário receberá, por *e-mail*, um código único gerado aleatoriamente vinculado ao processo, que é armazenado temporariamente no banco de dados. Para concluir o cadastro, é necessário que esse código seja enviado na requisição juntamente com a definição da senha, assegurando que apenas o proprietário do *e-mail* consiga finalizar o processo, Ilustração 06.

Ilustração 06 – *Controller* para criação de conta

```java
@PostMapping("/{token}")
public ResponseEntity<User> confirmNewAccount(
    @PathVariable String token,
    @RequestBody @Valid NewAccountRequestConfirmDTO dto
) {
  NewAccountRequest request = service.findByEmailAndToken(dto.email(), token);

  if (request == null) return ResponseEntity.badRequest().build();
  if (!dto.password().equals(dto.passwordConfirmation())) {
    throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "As senhas informadas não conferem");
  }

  UserDTO userDTO = new UserDTO(request.getName(), dto.email(), dto.password(), Role.USER);
  User savedUser = userService.store(userMapper.toEntity(userDTO));
  service.delete(request);

  return ResponseEntity.ok(savedUser);
}
```

Fonte: Autores (2025)

Como parte dos processos de negócio, a aplicação realiza envios de *e-mails*, especialmente para confirmação de cadastro. Embora atualmente o envio seja síncrono, a arquitetura foi pensada de forma que futuramente possa ser adaptado para um modelo assíncrono, melhorando o desempenho e a escalabilidade.

Além das validações descritas, a camada de negócios também contempla boas práticas no retorno de informações, adotando corretamente os *status HTTP*, implementação de filtros dinâmicos nas listagens e paginação dos dados, o que torna a experiência dos usuários mais eficiente e consistente.

Para assegurar a segurança dos dados e dos processos, a aplicação utiliza autenticação baseada em *JSON Web Tokens* (*JWT*) e controle de acesso por roles, implementados por meio do *Spring Security* em modo *stateless*. As roles definidas são *SUPER_ADMIN*, *ADMIN* e *USER*, sendo que usuários com perfil *USER* não possuem permissão para acessar rotas restritas ao painel administrativo, recebendo respostas apropriadas com *status HTTP* 401 (*unauthorized*) em caso de tentativa.

Outro aspecto relevante da camada de negócios é a adoção de *Data Transfer Objects* (*DTOs*) e mapeadores (*mappers*), que asseguram o desacoplamento entre os modelos de domínio e as representações utilizadas na comunicação com os clientes da *API*. Isso favorece tanto a segurança dos dados quanto a manutenção futura do projeto.

Por fim, toda a lógica da camada de negócios foi desenvolvida visando não apenas atender os requisitos atuais da aplicação, mas também preparando o

sistema para futuras expansões, como a integração com módulos de reconhecimento facial ou outras tecnologias que possam agregar valor ao controle de acesso oferecido pela plataforma.

2.4 CAMADA DE APRESENTAÇÃO

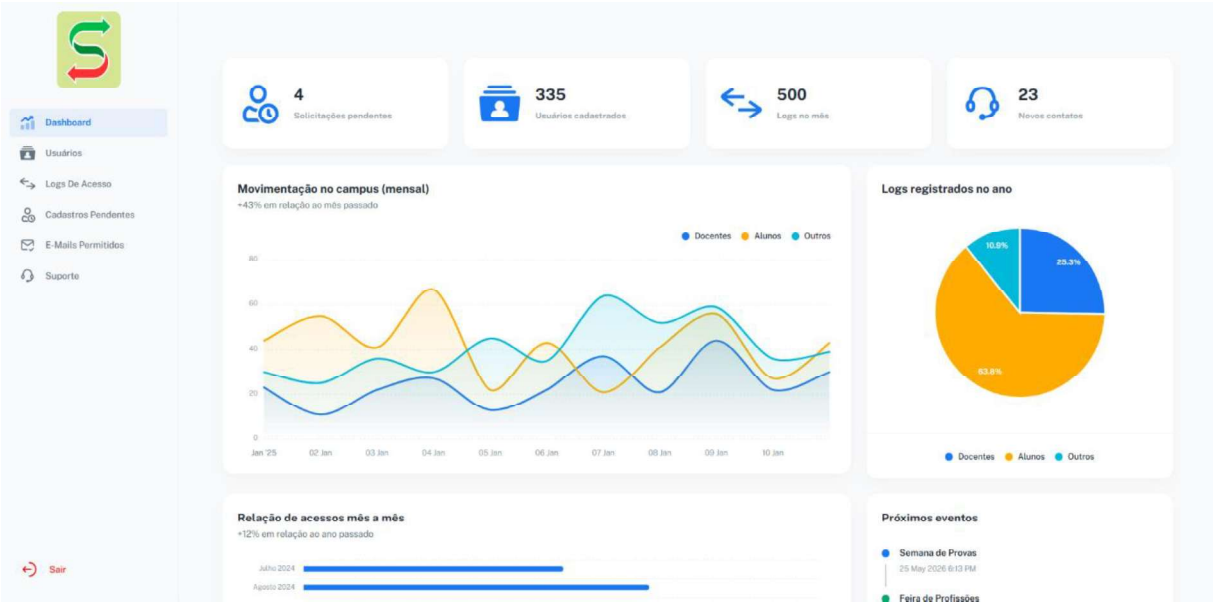
A camada de apresentação é responsável pela interação direta com os usuários, sejam eles administradores ou usuários finais. Ela se comunica com os *endpoints* da *API RESTful*, enviando dados ao *back-end* e exibindo as informações processadas. Esta camada foi desenvolvida em duas frentes principais: o painel administrativo e o aplicativo *mobile*.

2.4.1 PAINEL ADMINISTRATIVO

O painel administrativo foi desenvolvido utilizando *React*, com consumo dos dados provenientes da *API RESTful*. O projeto foi estruturado seguindo boas práticas de desenvolvimento, com a separação clara de componentes, páginas e *hooks*, cada um com responsabilidades bem definidas, o que facilita a manutenção e escalabilidade do código.

O *layout* foi baseado no *template Minimal Kit UI*, Illustração 07, porém, todas as funcionalidades foram desenvolvidas manualmente, sem o uso de geradores automáticos.

Ilustração 07 – *Dashboard* do painel administrativo

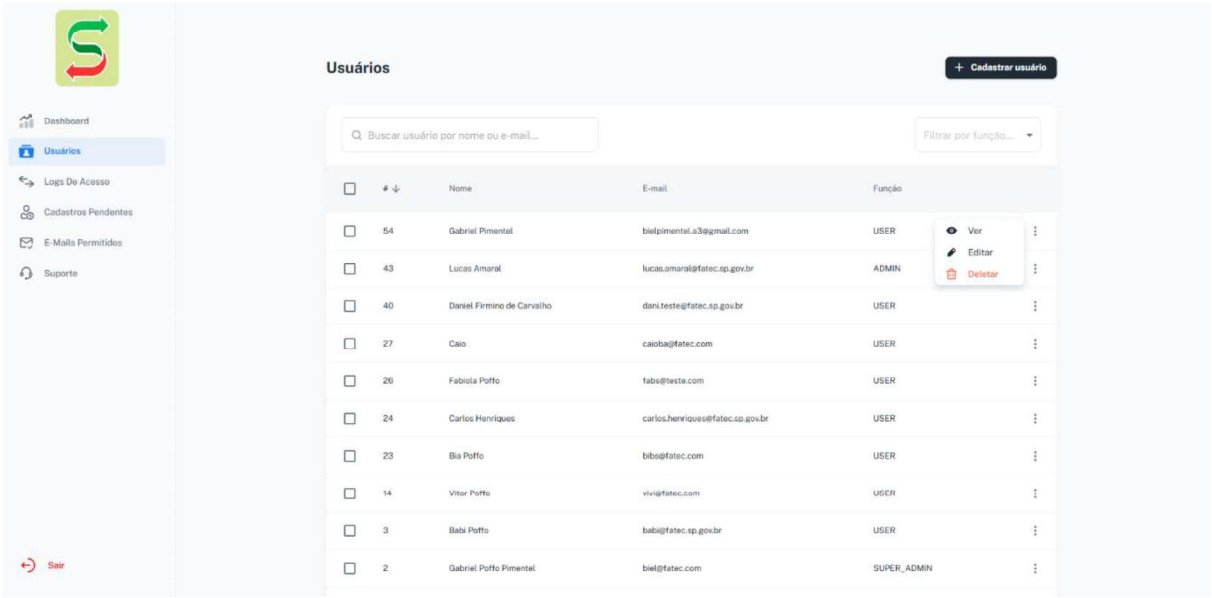


Fonte: Autores (2025)

O painel possui responsividade, podendo ser acessado via dispositivos móveis, embora seu *design* tenha sido pensado primordialmente para uso em *desktops*, garantindo uma melhor experiência de navegação em telas maiores.

A funcionalidade principal do painel administrativo é fornecer aos administradores controle completo da aplicação. Através dele, é possível acessar e gerenciar os *CRUDs* das entidades (com algumas restrições, como registros que não podem ser criados, editados ou excluídos, dependendo da regra de negócio), como o exemplo da listagem de usuários exibido na Ilustração 08. Além disso, usuários com a *role SUPER_ADMIN* possuem acesso a uma funcionalidade específica de *scanner* para leitura e validação de *QR Codes*, utilizada no controle de acesso físico ao campus.

Ilustração 08 – Listagem de usuários do painel administrativo



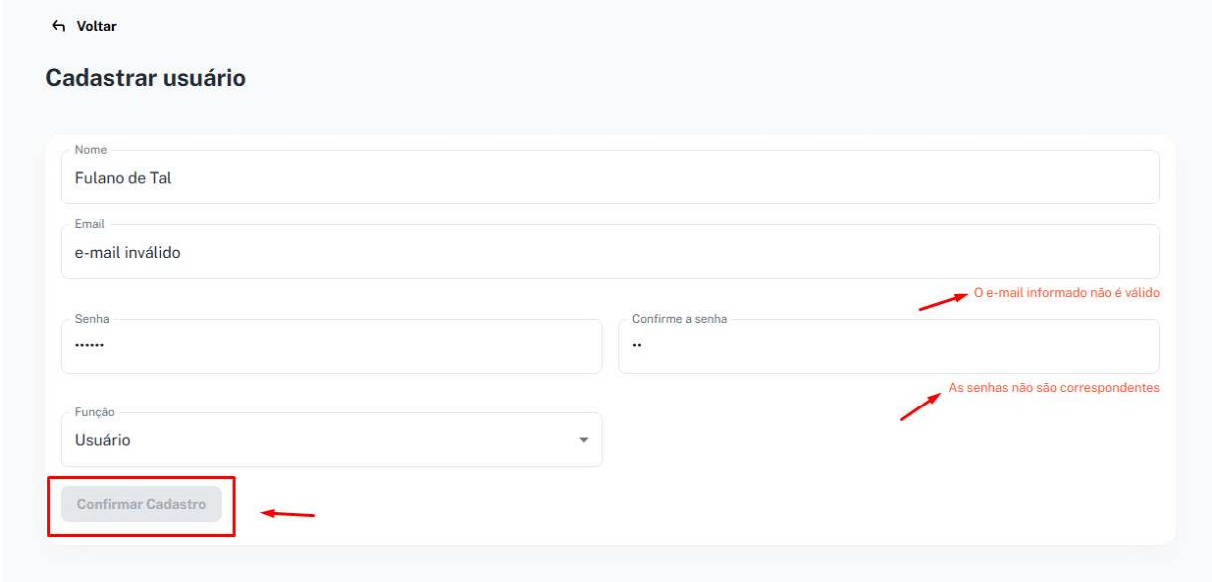
Fonte: Autores (2025)

O painel também conta com um sistema robusto de controle de acesso baseado em *JWT.* O roteamento das páginas é protegido por um *middleware* personalizado (*AuthGuard*), que verifica se o *token JWT* é válido e se o usuário possui as permissões necessárias, especialmente para áreas restritas, como a leitura de *QR Codes*.

Os gráficos presentes no *dashboard* foram desenvolvidos utilizando a biblioteca *Chart.js*, que proporciona visualizações dinâmicas e interativas.

Em relação à experiência do usuário, o painel possui validações de dados nos formulários de cadastro e edição (Ilustração 09), bem como tratamentos de erro, *feedbacks* visuais através de modais, e mensagens de confirmação ou erro, melhorando a interação e usabilidade.

Ilustração 09 – Validação de campos de formulário



Fonte: Autores (2025)

2.4.2 APLICATIVO MOBILE

O aplicativo foi desenvolvido utilizando *React Native*, com o ambiente *Expo*, uma das abordagens mais comuns e recomendadas pela comunidade para desenvolvimento híbrido. Assim como o painel administrativo, ele consome dados da mesma *API RESTful*, embora por meio de *endpoints* específicos para suas funcionalidades, com as devidas validações de autenticação e autorização.

O aplicativo serve como o principal ponto de interação dos usuários finais com o sistema SETA, permitindo que realizem seu cadastro, *login* e geração dos *QR Codes* utilizados para acesso ao campus da universidade, como observado na Ilustração 10.

Ilustração 10 – Tela de *QR Code* no aplicativo

Fonte: Autores (2025)

Para navegação, foi adotada a biblioteca *React Navigation*, que permite estruturar as telas e fluxos do *app* de maneira eficiente e intuitiva.

A gestão de autenticação no aplicativo é realizada por meio de um *Context API* próprio, que faz a persistência e verificação do *token JWT*, garantindo que somente usuários autenticados possam acessar as funcionalidades do aplicativo. O *token* é armazenado de forma segura utilizando o *AsyncStorage*.

O *design* e toda a identidade visual do aplicativo foram desenvolvidos inteiramente do zero, refletindo a proposta do projeto e proporcionando uma interface moderna e alinhada com as necessidades dos usuários.

Ilustração 11 – Tela principal do aplicativo

Fonte: Autores (2025)

3 RESULTADO

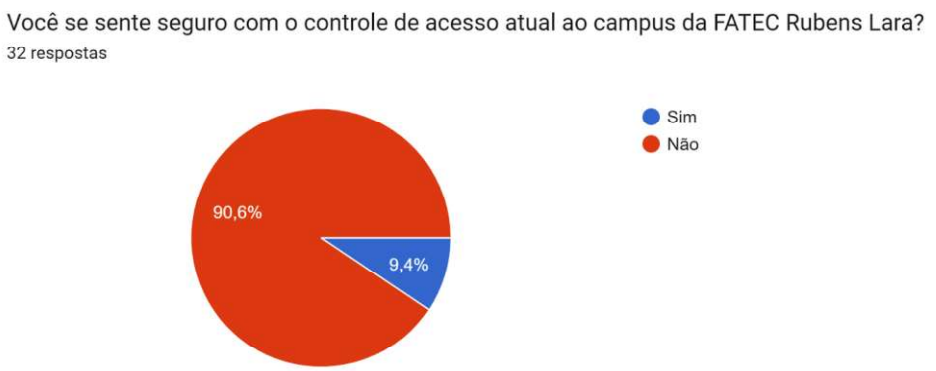
Com o desenvolvimento do sistema SETA finalizado, foi conduzida uma etapa de avaliação prática com foco na análise de sua relevância, usabilidade e eficiência do ponto de vista dos usuários finais. Para isso, entre os dias 20 e 27 de maio de 2025, foram realizados dois métodos complementares: uma pesquisa de campo, voltada à coleta de opiniões sobre o cenário atual e o interesse no uso da aplicação, e um teste de usabilidade, ambos em Santos, com foco na interação direta com o sistema.

3.1 PESQUISA DE CAMPO

A pesquisa foi aplicada a 32 alunos da Fatec Baixada Santista – Rubens Lara, abrangendo representantes dos cursos de Análise e Desenvolvimento de Sistemas, Ciência de Dados e Sistemas para *Internet* da instituição. O objetivo principal foi levantar a percepção dos alunos quanto ao sistema atual de controle de acesso, avaliando também o interesse por soluções digitais alternativas, como o sistema SETA.

Na primeira pergunta, apresentada na Ilustração 12, os resultados obtidos foram extremamente expressivos ao demonstrar um grande sentimento de insegurança com o meio de controle de acesso atualmente em vigor, com um retorno negativo de mais de 90% dos entrevistados.

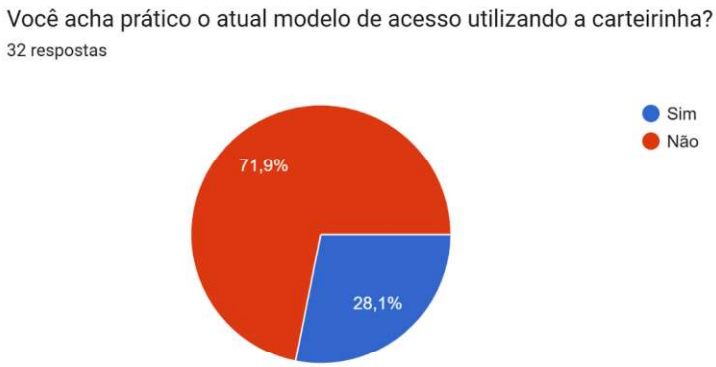
Ilustração 12 – Percepção de segurança com o controle de acesso atual



Fonte: Autores (2025)

Na segunda pergunta, apresentada na Ilustração 13, ficou clara a visão negativa dos alunos quanto à praticidade do sistema empregado atualmente.

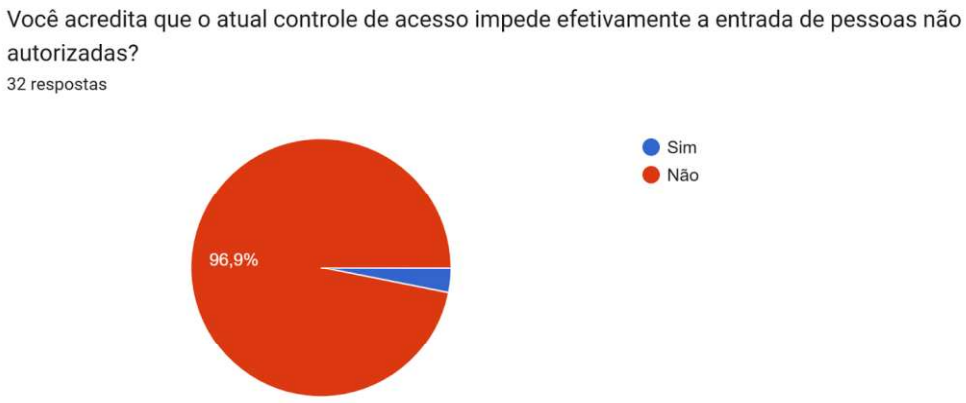
Ilustração 13 – Avaliação da praticidade do método atual



Fonte: Autores (2025)

Na terceira pergunta, conforme exibido na Ilustração 14, ao serem questionados se o método atual impede a entrada de pessoas não autorizadas, 96,9% dos participantes responderam negativamente, evidenciando falta de confiança na eficácia do sistema vigente.

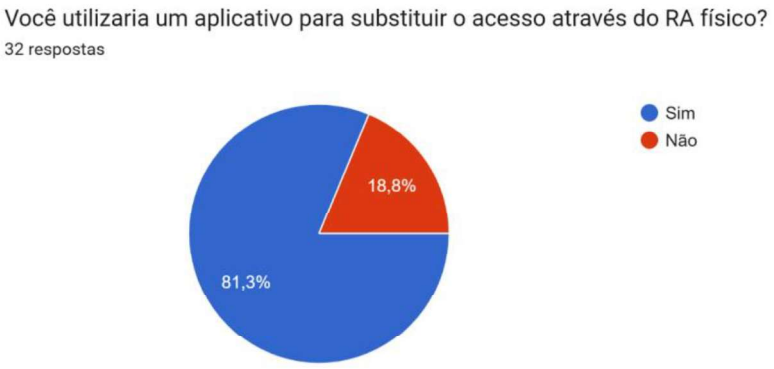
Ilustração 14 – Percepção da eficácia no bloqueio de acesso indevido



Fonte: Autores (2025)

Na quarta e última pergunta, apresentada na Ilustração 15, condizente com os resultados obtidos com as perguntas anteriores, embora não unânime, foi demonstrada uma visão inicial positiva quando a questão foi a utilização de um aplicativo para celular, em um cenário de substituição do controle de acesso atual.

Ilustração 15 – Interesse na utilização de aplicativo móvel



Fonte: Autores (2025)

3.2 TESTE DE USABILIDADE

A avaliação do público quanto a funcionalidade de um projeto é, sem dúvidas, um ponto de grande relevância ao se desenvolver um sistema. Com isso em mente e seguindo recomendações de autores como Steve Krug (2006), testes de usabilidade foram realizados com o intuito de observar a interação de usuários reais com o sistema, avaliando sua facilidade de uso, clareza na navegação, tempo de execução de tarefas e eficiência geral da aplicação.

> Um teste de usabilidade envolve observar as pessoas tentando usar o que você está criando (ou já criou), com a intenção de facilitar o uso para as pessoas.
> (KRUG, 2006)

A amostra contou com 10 alunos voluntários, com idades entre 19 e 38 anos, oriundos de diferentes cursos. Cada participante foi acompanhado por um observador da equipe de desenvolvimento, cuja função era apenas monitorar, esclarecer dúvidas pontuais e registrar as interações.

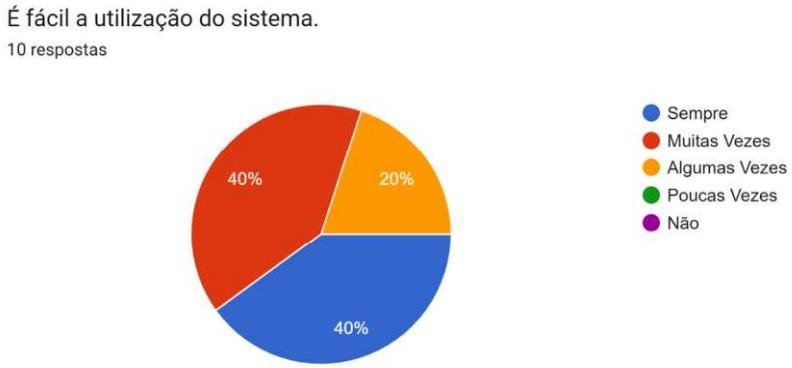
As tarefas envolveram: realizar o cadastro utilizando um *e-mail* institucional válido; efetuar *login* na plataforma; gerar um código *QR* na interface principal do aplicativo; e, por fim, validar o *QR Code* gerado por meio do *scanner*.

Durante a realização das tarefas, a experiência geral dos usuários foi positiva, destacando-se a facilidade de navegação, o tempo de resposta do sistema e a clareza das telas. A única ressalva observada foi quanto ao tamanho inicial do

QR Code, que dificultava a leitura pelo *scanner*, questão rapidamente solucionada com um ajuste no *layout*.

Ao final dos testes, foi aplicado um breve questionário com cinco perguntas fechadas, baseadas em princípios de usabilidade. A maioria dos participantes classificou o sistema como fácil de usar, destacando a clareza na navegação e a simplicidade do *layout*, com 80% dos voluntários respondendo com as opções mais altas da escala, indicando baixo nível de frustração durante a interação, conforme Ilustração 16.

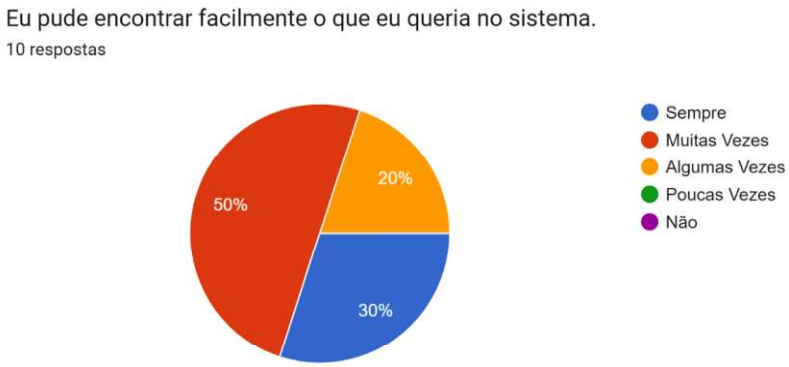
Ilustração 16 – Avaliação de facilidade de uso



Fonte: Autores (2025)

De acordo com a Ilustração 17, 80% afirmaram que as funcionalidades estavam bem localizadas e intuitivamente acessíveis, o que reforça a eficácia da organização da interface.

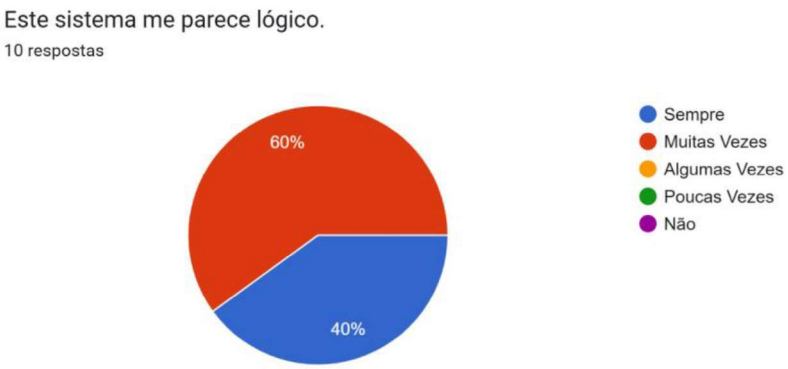
Ilustração 17 – Facilidade para encontrar recursos



Fonte: Autores (2025)

Na terceira pergunta, todos os participantes assinalaram opções positivas (Ilustração 18), o que mostra que o fluxo das ações está coerente com a expectativa do usuário.

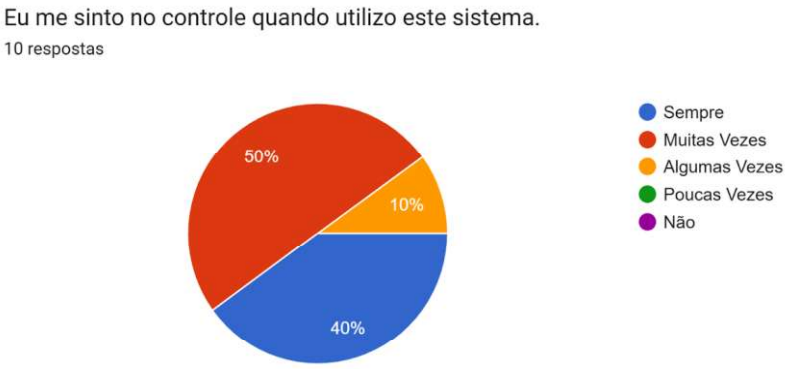
Ilustração 18 – Avaliação da lógica de uso



Fonte: Autores (2025)

Na quarta questão, referente ao sentimento de controle durante o uso, os voluntários relataram uma sensação de domínio sobre suas ações, como é possível visualizar nos resultados apresentados na Ilustração 19, o que é essencial em sistemas voltados à segurança e controle de acesso.

Ilustração 19 – Sentimento de controle durante o uso



Fonte: Autores (2025)

Por fim, o retorno obtido a partir da última pergunta, reforçou a utilidade prática da aplicação, com todos os usuários respondendo positivamente, conforme Ilustração 20.

Ilustração 20 – Avaliação da utilidade da aplicação

Fonte: Autores (2025)

3.3 CONCLUSÃO

O sistema SETA foi projetado com o objetivo de oferecer uma alternativa digital e segura ao controle de acesso físico tradicional utilizado por instituições de ensino. Os resultados obtidos nas etapas de pesquisa e testes apontam para uma aceitação positiva por parte dos usuários, bem como indicam falhas significativas no modelo atualmente em uso.

Através da arquitetura modular e escalável, apresentada no Capítulo 2, o sistema demonstrou ser capaz de atender não apenas à necessidade funcional de registrar e controlar acessos com eficiência, mas também de proporcionar usabilidade, clareza e praticidade para usuários finais e administradores.

A partir da pesquisa de campo realizada com alunos da Fatec Rubens Lara, constatou-se que há demanda concreta por soluções como o SETA, especialmente em contextos em que recursos físicos são limitados, a segurança é falha e a burocracia institucional compromete a eficiência e o desenvolvimento.

Ademais, por estar presente tanto em uma versão *mobile* — seja para usuários de sistemas *Android* ou *iOS* — como para navegadores, a aplicação se torna acessível e adaptável nos mais diferentes casos de uso.

Como passos futuros, serão implementados novos recursos visando ampliar a utilidade do sistema e atender a um escopo ainda maior de demandas institucionais. Dentre os principais aprimoramentos previstos, destacam-se:

Módulo de visitantes: inclusão de funcionalidade para permitir acesso temporário de visitantes, especialmente em eventos, palestras ou visitas à biblioteca.

Carga em massa: desenvolvimento de funcionalidade para inserção de grandes volumes de dados via arquivos .*csv* ou .*xlsx*, facilitando a importação da base de alunos ou colaboradores.

Módulo informativo: adição de funcionalidades informativas, como mapas de salas, horário de funcionamento e localização de departamentos.

Reconhecimento facial: futura integração com sistema de reconhecimento facial, aumentando ainda mais a segurança e a automação dos pontos de acesso.

Dessa forma, o SETA consolida-se como uma solução promissora, tanto do ponto de vista técnico quanto funcional, oferecendo uma alternativa eficiente, segura e moderna para o controle de acesso em ambientes educacionais. Sua estrutura robusta, combinada à boa aceitação entre os usuários-alvo, evidencia seu potencial de aplicação em instituições com diferentes níveis de infraestrutura tecnológica. Ao alinhar praticidade, baixo custo e escalabilidade, o sistema contribui para a modernização dos processos institucionais, estabelecendo bases sólidas para futuras expansões e inovações.

REFERÊNCIAS

4PORT. **Portaria Remota 4.0**. Disponível em: https://4port.com.br/portaria-remota-4-0/. Acesso em: 26 maio 2025.

ANDERSON, Ross. **Security Engineering: A Guide to Building Dependable Distributed Systems**. 2. ed. Indianapolis: Wiley, 2008.

BAZZI, Claudio. **Introdução a banco de dados**. Apostila: Universidade Tecnológica Federal do Paraná, 2013.

BENANTAR, Messaoud. **Access Control Systems: Security, Identity Management and Trust Models.** New York: Springer, 2006.

Eventim BR, 2025. Disponível em: < https://play.google.com/store/apps/details?id=br.com.eventim.mobile.app.Android&hl=pt_BR&pli=1>. Acesso em: 27 de maio de 2025.

IACCESS. **Sistema iAccess**. Disponível em: https://www.iacess.com.br/. Acesso em: 26 maio 2025.

JONES, M.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. RFC 7519. Internet Engineering Task Force (IETF), 2015. Disponível em: <https://tools.ietf.org/html/rfc7519>. Acesso em: 26 mai. 2025.

KHATRI, Sunil Kumar et al. **Smart QR code–based access control system using secure communication.** International Journal of System Assurance Engineering and Management, v. 14, p. 1–11, 2023. DOI: https://doi.org/10.1007/s13198-023-01860-7

KRUG, Steve. ***Don't Make Me Think****: A Common Sense Approach to Web Usability*. 2ª ed. Berkeley: New Riders, 2006.

NJUGUNA, David; NDIA, John. **Quick Response Code Security Attacks and Countermeasures: A Systematic Literature Review**. Journal of Cyber Security, v. 7, n. 1, p. 1–20, 2025. DOI: 10.32604/jcs.2025.059398.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: uma abordagem profissional**. 8ª ed. São Paulo: McGraw Hill, 2016.

REZENDE, Denis Alcides; DE ABREU, Aline França. **Tecnologia da informação aplicada a sistemas de informação empresariais**: o papel estratégico da informação e dos sistemas de informação nas empresas. 3ª ed. São Paulo: Atlas, 2003.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. **Sistemas Operacionais**. 9. ed. Rio de Janeiro: LTC, 2019.

SILVA, Alberto; VIDEIRA, Carlos. **UML, Metodologias e ferramentas CASE**. 1ª ed. Editora Centro Atlântico, 2001.

STALLINGS, William. **Criptografia e segurança de redes**: princípios e prática. 6ª ed. São Paulo: Pearson, 2014.

TANENBAUM, Andrew S.; WETHERALL, David J. **Redes de computadores**. 5ª ed. São Paulo: Pearson, 2011.

TOTVS. **Controle de Acesso**. Disponível em: https://www.totvs.com/segmentos/erp-educacional/controle-de-acesso/. Acesso em: 26 maio 2025.

APÊNDICE A – Diagrama de Casos de Uso

Fonte: Autores (2025)

APÊNDICE B – Principais Fluxos de Eventos

Fluxo: Cadastro de Usuário
Atores: Usuário
Pré-condições: *E-mail* institucional válido
Fluxo principal: 1 - O usuário clica na opção "Criar Conta", presente na tela de *login*. 2 - O usuário insere os dados exigidos no formulário de cadastro. 3 - O usuário clica no botão "Confirmar". 4 - O sistema verifica se o *e-mail* institucional e demais dados são válidos. 5 - O sistema salva os dados e dispara um *e-mail* com *token* e *link* para finalização do cadastro. 6 - Fim do caso de uso.
Fluxo alternativo: 4.1 - Se o usuário inserir um ou mais dados inválidos, ele permanecerá na página de cadastro e será requisitado a reinserir os dados considerados não válidos, com dados apropriados agora. 4.2 - Se o usuário deixar um ou mais campos, que são declarados como necessários, em branco, ele permanecerá na página de cadastro, onde será requisitado a inseri-los.

Fluxo: Geração de *QR Code*
Atores: Usuário
Pré-condições: O usuário deve estar logado no sistema.
Fluxo principal: 1 - O usuário acessa a tela de geração de *QR Code*. 2 - O usuário clica no botão com ícone de “atualização”. 3 - O sistema cria um *QR Code* único vinculado ao usuário e com validade definida.

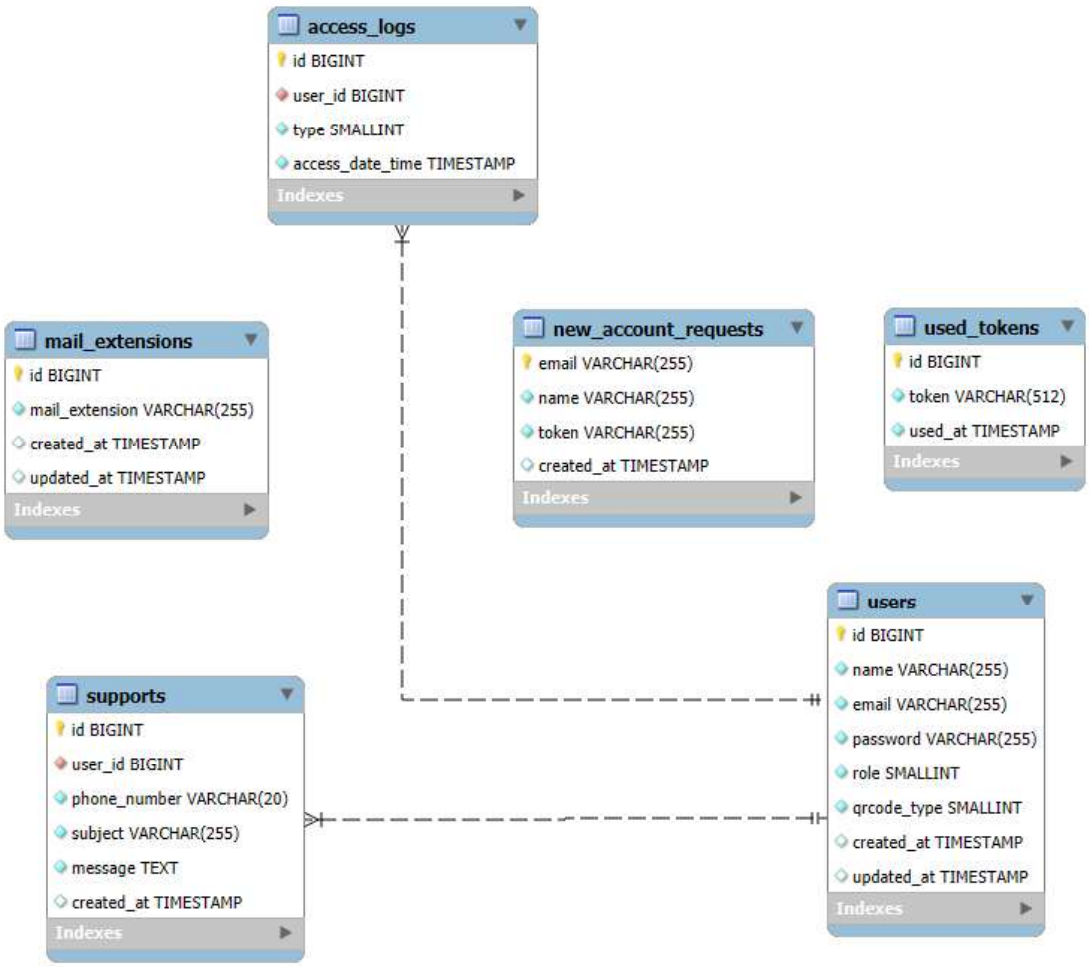
4 - O *QR Code* é exibido na tela do aplicativo. 5 - Fim do caso de uso.
Fluxo alternativo: 3.1 - Se, durante a geração do *QR Code*, ocorrer um erro, o sistema exibe uma mensagem informando o usuário, e o libera para tentar novamente.

Fluxo: Validação de *QR Code*
Atores: Usuário
Pré-condições: Um dispositivo com o *scanner* ativo e com acesso a uma câmera funcional.
Fluxo principal: 1 - O usuário apresenta o *QR Code* para a câmera do dispositivo, ajustando seu posicionamento de acordo com o que é indicado na tela do dispositivo com o *scanner*. 2 - O *scanner* processa a imagem e decodifica o *QR Code*. 3 - O sistema verifica se o *QR Code* é válido. 4 - Se for válido, exibe uma mensagem de confirmação. 5 - O acesso é autorizado e registrado no sistema. 6 - Fim do caso de uso.
Fluxo alternativo: 1.1 - Se a câmera do dispositivo apresentar algum problema, ou não estiver disponível, uma mensagem é exibida. 4.1 - Se o *QR Code* for inválido, estiver expirado, ou houver qualquer problema com ele, o sistema exibe uma mensagem de erro, nega o acesso e pede ao usuário que tente novamente.

Fluxo: Relatórios de Acesso
Atores: Administrador

Pré-condições: Estar logado no painel administrativo
Fluxo principal: 1 - O administrador acessa a seção de *logs* de acesso. 2 - O sistema exibe informações sobre acessos e horários. 3 - O administrador pode filtrar dados conforme necessário. 4 - Fim do caso de uso.
Fluxo alternativo: 2.1. Se não houver registros disponíveis, o sistema exibe uma mensagem informando que não há dados para exibição.

APÊNDICE C – Modelo Entidade-Relacionamento



Fonte: Autores (2025)