

PROTÓTIPO DE APLICATIVO MOBILE PARA MONITORAMENTO DE EMBARQUE E DESEMBARQUE DE ESTUDANTES NO TRANSPORTE ESCOLAR UNIVERSITÁRIO

MAURICIO MIGUEL DA SILVA¹ MARCEL CAMPOS INOCENCIO²

Resumo: Este trabalho apresenta um protótipo de sistema para controle de presença de alunos no transporte escolar, utilizando a leitura de QR Code, com o objetivo de melhorar a segurança no monitoramento dos alunos durante o transporte. O sistema foi desenvolvido para dispositivos Android, utilizando React Native e Expo SDK, integrado com um banco de dados MySQL e a plataforma Firebase para o envio de notificações. Testes realizados em ambiente simulado mostraram que o protótipo é funcional, proporcionando uma leitura rápida e precisa do QR Code, com um tempo médio de 2 segundos para o registro de cada aluno. O sistema também foi capaz de enviar notificações em tempo real para os responsáveis, garantindo o acompanhamento contínuo. No entanto, foram identificadas limitações, como a dependência do sistema operacional Android e a necessidade de conexão com a internet para o envio das notificações. Futuras melhorias incluem a expansão para outras plataformas e o desenvolvimento de funcionalidades para operar offline.

Palavras-chave: Controle de presença; Transporte escolar; QR Code; React Native.

This work develops a prototype system for student attendance control in school transportation, using QR Code scanning, aiming of automating and improving the safety of monitoring students during transport. The system was developed for Android devices, using React Native and Expo SDK, and integrated with a MySQL database and the Firebase platform for sending notifications. Tests conducted in a simulated environment demonstrated that the prototype is functional, providing quick and accurate QR Code scanning, with an average time of 2 seconds per student. The system was also able to send real-time notifications to guardians, ensuring continuous monitoring. However, limitations were identified, such as the dependency on the Android operating system and the requirement for an internet connection to send notifications. Future improvements include expanding to other platforms and developing offline operation capabilities.

Keywords: Attendance control; School transportation; QR Code; React Native.

¹ Aluno do Curso de Ciências da Computação da Universidade do Extremo Sul Catarinense – UNESC, mauricio_miguel@unesc.net

² Professor e Orientador do Curso de Ciências da Computação da Universidade do Extremo Sul Catarinense – UNESC, marcel.inocencio@gmail.com

1. INTRODUÇÃO

O transporte escolar exerce um papel essencial na promoção do acesso à educação, especialmente em municípios de pequeno porte que não dispõem de instituições de ensino superior. Em contextos como esses, programas governamentais, tais como o Caminho da Escola e o Programa Nacional de Apoio ao Transporte Escolar (PNATE), tornam-se fundamentais para assegurar que os estudantes possam se deslocar para cidades vizinhas e, assim, prosseguir com seus estudos. Esses programas abrangem tanto alunos da educação básica quanto do ensino superior, ampliando as oportunidades educacionais e promovendo a inclusão escolar em diferentes níveis de ensino (BRASIL, 2004; BRASIL, 2007).

Atualmente, o processo de validação do uso do transporte escolar permanece predominantemente manual, com métodos tradicionais e muitas vezes ineficazes, como carteirinhas de papel e listas físicas que exigem a conferência manual por parte dos motoristas. Além disso, os monitores, profissionais responsáveis por supervisionar os alunos durante o trajeto e assegurar que apenas usuários autorizados utilizem o transporte, relatam dificuldades na identificação dos alunos devido à quantidade expressiva de usuários. Essa dificuldade abre margem para o uso indevido do transporte por pessoas não autorizadas (CEAT App, 2021). Esses métodos comprometem tanto a segurança dos alunos quanto a organização logística do transporte escolar, uma vez que não há um controle em tempo real.

Diante desse cenário, este trabalho propõe o desenvolvimento de um protótipo de aplicativo mobile cuja principal ferramenta é a utilização de QR Code para o controle de embarque e desembarque dos estudantes no transporte escolar. O QR Code (Quick Response Code) é um código bidimensional criado pela Denso Wave em 1994, amplamente utilizado para armazenar e transmitir informações de forma rápida e eficiente. Ele permite que dados sejam lidos e processados por dispositivos móveis com alta precisão, otimizando processos que exigem validação ágil e segura (KATO; TAN, 1999). Essa tecnologia foi escolhida por sua capacidade de garantir maior segurança e praticidade no controle de presença dos alunos. A proposta é que o aplicativo permita aos alunos registrar sua presença ao escanear o QR Code fixado no interior do veículo, proporcionando uma validação digital imediata. Esse sistema visa substituir os métodos manuais atualmente empregados, oferecendo maior segurança e eficiência tanto para os alunos quanto para os motoristas.

A escolha do QR Code como tecnologia central no protótipo justifica-se por sua capacidade de armazenar uma quantidade significativa de dados em um pequeno espaço, sendo facilmente lido por diversos dispositivos móveis. Essa tecnologia foi projetada para otimizar processos que exigem alta precisão e rapidez na validação de informações, sendo amplamente utilizada em sistemas de controle e monitoramento (SOUZA, 2020). O QR Code permite que informações relevantes, como a identificação dos usuários, sejam processadas de forma ágil e eficiente, eliminando a necessidade de métodos manuais e sujeitos a erros e fraudes.

Dessa forma, a proposta deste trabalho visa melhorar o processo de validação de presença no transporte escolar, e também garantir um sistema mais seguro, prático e eficiente. A aplicação de tecnologias móveis no setor educacional, especialmente por meio do QR Code, mostra-se promissora para reduzir falhas operacionais e promover uma gestão mais efetiva dos

serviços de transporte escolar, contribuindo significativamente para a segurança e a acessibilidade educacional.

2. TRABALHOS CORRELATOS

No trabalho intitulado CEAT App (2021), foi desenvolvido um sistema para controle e monitoramento do transporte escolar, utilizando QR Code para autenticação dos alunos. Os monitores dos ônibus, responsáveis por supervisionar os alunos durante o trajeto, escaneiam o QR Code presente nas carteirinhas dos estudantes, verificando a presença e autenticidade dos mesmos. O aplicativo fornece uma interface que permite a motoristas e responsáveis acessarem, as informações sobre os alunos.

Desenvolvido em Java no ambiente Android Studio, o sistema inclui funcionalidades como cadastro de rotas e comunicação com os responsáveis, utilizando o Firebase para armazenamento de dados.

Almeida e Aguiar (2020) desenvolveram também um sistema para monitoramento do transporte escolar, o Safe-School-Bus, que permite aos responsáveis acompanhar o trajeto dos alunos, inspirado em aplicativos de transporte. O sistema possibilita que o motorista atualize o status dos estudantes durante o embarque e desembarque, enviando notificações automáticas aos responsáveis sobre a localização e o status dos alunos.

O sistema foi desenvolvido em React Native para o front-end e Node.js para o back-end, e inclui funcionalidades de geolocalização para monitoramento contínuo. Utiliza um banco de dados relacional para o armazenamento das informações de motoristas, alunos e responsáveis, embora o tipo específico de banco de dados não tenha sido mencionado no trabalho.

No Projeto de Mateus Ferreira Pinto (2017), foi desenvolvido um aplicativo voltado para o controle e organização do transporte escolar, direcionado a motoristas e responsáveis pelos alunos. O sistema permite que motoristas cadastrem estudantes e rotas, além de fornecer funcionalidades para comunicação direta com os responsáveis, possibilitando o acompanhamento do status dos alunos durante o transporte.

Desenvolvido no Android Studio e utilizando Java como linguagem de programação, o aplicativo integra o Firebase para armazenamento e sincronização de dados. A tela de login inicial direciona o motorista e o responsável para menus distintos, permitindo que motoristas acessem funcionalidades de gerenciamento de rotas, registro de presença e controle de mensalidades. Os responsáveis, por sua vez, podem visualizar o status dos alunos, sendo notificados sobre embarques e desembarques, além de informar ao motorista, antecipadamente, sobre possíveis ausências dos alunos no transporte.

No trabalho de Silva (2013), foi desenvolvido o sistema COADS, uma solução para o controle de transporte escolar que integra a validação de presença e dados administrativos dos alunos. O sistema utiliza um leitor de código de barras instalado nos veículos para verificar a presença de cada aluno ao embarcar, utilizando uma carteira de estudante com código de barras que contém informações essenciais, como os dados cadastrais e a situação de pagamento das mensalidades. A aplicação inclui uma versão desktop que permite a atualização constante da base de dados via internet, mantendo um histórico de frequência acessível aos administradores. Desenvolvido na plataforma .NET com as linguagens C#.NET, ASP.NET e HTML, o sistema usa PostgreSQL e SQL Server Compact Edition para compatibilidade com dispositivos desktop e móveis.

3. MATERIAS E METODOS

O desenvolvimento do protótipo de controle do transporte escolar foi dividido em etapas que incluem a escolha das tecnologias, a modelagem do banco de dados, a criação da API, o desenvolvimento de um site e aplicativo.

3.1 BANCO DE DADOS

Considerando a necessidade de manter a integridade das relações entre as tabelas, foi escolhido um banco de dados relacional, o MySQL na versão LTS 8.0.34. A Figura 1 a seguir ilustra o diagrama com as tabelas e seus respectivos relacionamentos:

Cada tabela desempenha um papel específico, conforme descrito a seguir:

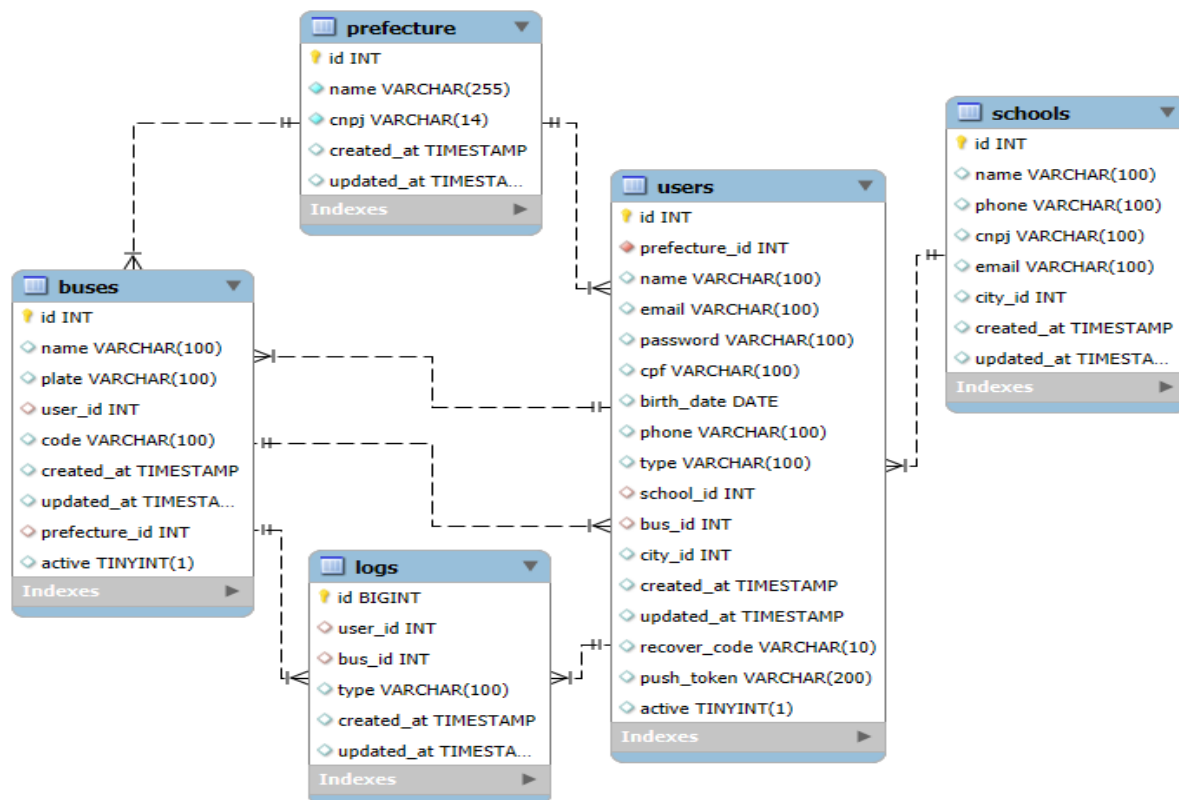
- **Users (Usuários):** Armazena as informações de todos os usuários, incluindo alunos, motoristas, administradores. Os principais campos incluem nome, e-mail, CPF, senha, tipo de usuário (admin, motorista, aluno), além de tokens para notificações push e status de atividade (ativo/inativo);
- **Buses (Ônibus):** Contém dados dos ônibus utilizados no transporte escolar, incluindo placa e id do ônibus, o ônibus está atrelado a prefeitura e ao motorista. O relacionamento é many to one (muitos para um) pois vários ônibus pertencem a uma prefeitura. E o motorista pode estar atrelado a mais de um ônibus;
- **Schools (Escolas):** Armazena informações das escolas registradas no sistema, como nome da instituição de ensino, CNPJ, dados de contato e associação com uma cidade específica;
- **Logs (Registros):** Registra todas as atividades dos usuários no sistema, como o embarque e desembarque de alunos, associando cada registro ao aluno, a um ônibus e a um horário específico. Cada vez que um aluno escaneia o QR Code ao entrar no ônibus, a presença é automaticamente registrada no banco de dados;
- **Prefecture (Prefeituras):** Armazena informações como o nome e o CNPJ da prefeitura. Essas prefeituras são responsáveis pela organização das operações de transporte.

A estrutura do banco de dados utiliza chaves únicas e chaves estrangeiras para garantir a consistência das informações e facilitar consultas, como a verificação de alunos e motoristas vinculados a cada veículo. Os relacionamentos do banco de dados foram estruturados com base no modelo many to one, one to many e many to many garantindo a organização das informações e a integridade referencial. Por exemplo, cada ônibus pode estar vinculado a apenas uma prefeitura, enquanto uma mesma prefeitura pode ser responsável por vários

ônibus. Esse tipo de modelagem assegura consultas ágeis, como a identificação dos alunos associados a um determinado ônibus ou a verificação dos motoristas vinculados às suas respectivas rotas.

Além disso, o uso de chaves estrangeiras conecta tabelas essenciais, como as de usuários, ônibus e registros de presença, permitindo que todas as operações sejam integradas de forma eficiente. Ao registrar a presença de um aluno, o sistema associa automaticamente o evento às tabelas envolvidas, como usuários, ônibus e logs de atividades. Essa estrutura facilita a análise de dados, a geração de relatórios e o acompanhamento em tempo real das operações do transporte escolar.

Figura 1: Diagrama de entidade de relacionamento



Fonte: Do autor (2024).

3.2 INTERFACE DE PROGRAMAÇÃO DE APLICAÇÃO (API)

A API (Application Programming Interface) é definida como um conjunto de definições e protocolos que permite a comunicação entre diferentes sistemas de software, funcionando como uma interface que conecta o front-end (interface do usuário) ao back-end (servidor e banco de dados). Essa tecnologia facilita a troca de informações de forma estruturada e segura, sendo amplamente utilizada no desenvolvimento de aplicações modernas devido à sua modularidade e eficiência (FIELDING; TAYLOR, 2000).

Desenvolvida utilizando Node.js (versão 20) e desempenha um papel fundamental na comunicação entre o front-end e o banco de dados. A API foi projetada com o objetivo de agilizar processos, como o registro de presença por meio da leitura de QR Code, o envio de notificações push em tempo real, a gestão de usuários e ônibus. Abaixo seguem descritas as principais funcionalidades da API:

a. **Cadastro e Autenticação de Usuários:**

- A API permite o registro de novos usuários, como motoristas e alunos, gerenciando informações como nome, CPF, e-mail e senha;
- O processo de autenticação é realizado por meio de tokens JWT para garantir o acesso seguro ao sistema;
- O sistema permite a recuperação de senha em caso de perda ou comprometimento da conta.

b. **Gerenciamento de Presenças:**

- A leitura do QR Code pelos alunos ao embarcar no ônibus faz com que a presença seja registrada pela API, que atualiza os dados de logs no banco de dados. Cada registro é vinculado a um aluno, ônibus e horário específicos, garantindo que as presenças sejam geradas de forma precisa e segura.

c. **Notificações Push:**

- A API está integrada ao Firebase para enviar notificações push em tempo real aos motoristas, informando ao motorista o momento de partida. Essa funcionalidade permite que o motorista esteja ciente que todos os alunos já estão no ônibus para seguir viagem.

d. **Gerenciamento de Ônibus:**

- A API permite que os administradores gerenciem os dados dos ônibus, incluindo a atualização de informações como placa e motorista.

e. **Consultas:**

- A API fornece rotas específicas para consulta de dados, permitindo que motoristas e alunos visualizem informações detalhadas sobre presenças e status dos usuários. Essas informações são apresentadas de forma eficiente no formato de listagem.

Quanto à estrutura e fluxo de requisições, a API foi construída no padrão REST, utilizando requisições HTTP para comunicação entre o front-end (web e mobile) e o banco de dados. As respostas da API são enviadas em formato JSON, facilitando a integração com diferentes plataformas e garantindo uma resposta rápida e eficiente.

Por fim, para garantir a segurança das operações, a API implementa JWT (JSON Web Tokens), um padrão aberto (RFC 7519) amplamente utilizado para autenticação e autorização em aplicações distribuídas. O JWT permite que informações sejam transmitidas entre o cliente e o servidor de forma compacta e segura, sendo assinadas digitalmente para garantir sua integridade (JONES; HARDT, 2015).

No sistema, o JWT é utilizado para proteger as rotas de acesso, assegurando que apenas usuários autorizados possam acessar e modificar dados. Durante o processo de autenticação, um token é gerado e enviado ao cliente, contendo informações sobre o nível de acesso do usuário. Esse token é verificado em todas as requisições, permitindo que ações

sensíveis, como o gerenciamento de usuários e veículos, sejam realizadas apenas por administradores. Essa abordagem protege as informações e limita o acesso às operações críticas.

3.3 DESENVOLVIMENTO DO SITE

O desenvolvimento do site foi realizado utilizando Vue.js (versão 12), em conjunto com o Quasar Framework (versão 1.22) para a criação da interface. A Biblioteca AXIOS foi utilizada para realizar as requisições à API, facilitando a integração entre o front-end e o banco de dados.

A escolha do Vue.js se deu devido à sua modularidade e capacidade de criar componentes reutilizáveis, o que facilitou a criação de telas dinâmicas e responsivas. O Quasar Framework foi essencial para garantir uma experiência de usuário consistente em diversos dispositivos, como desktops, tablets e smartphones.

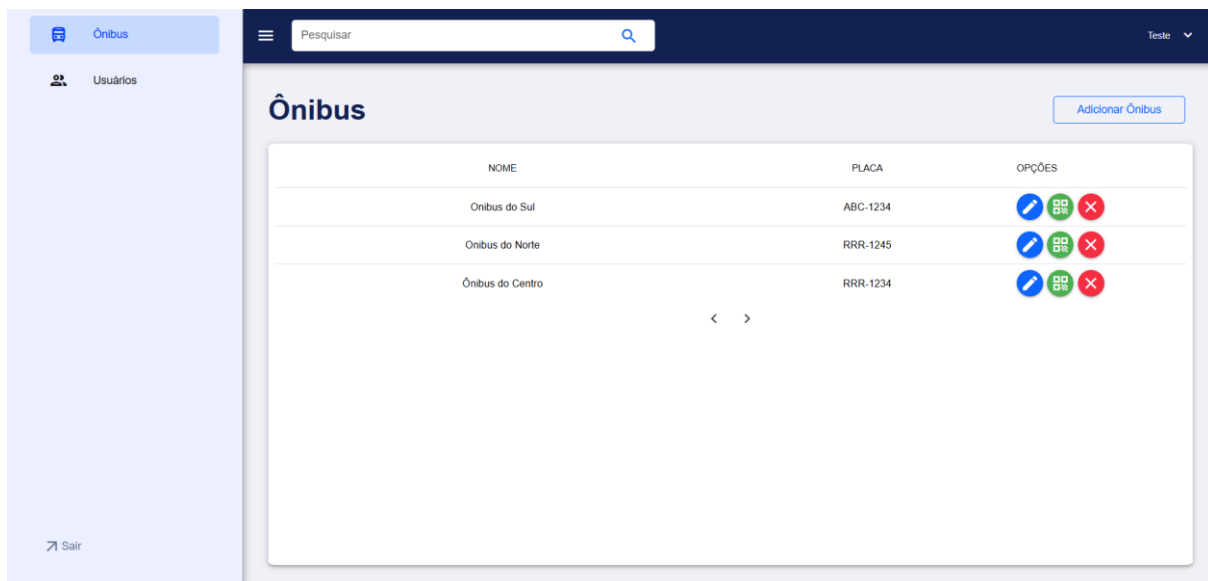
O objetivo do site é fornecer à Secretaria da Educação de cada prefeitura uma plataforma eficiente para a gestão do sistema de transporte escolar, permitindo o cadastro e a administração de alunos, motoristas e ônibus.

3.3.1 FUNCIONALIDADES DAS TELAS

A seguir, são descritas duas telas do sistema que exemplificam a interface desenvolvida:

- a. Tela de cadastro de Ônibus (Figura 2):
 - Esta tela permite ao administrador adicionar novos ônibus ao sistema, com informações como nome e placa. O administrador também pode editar ou excluir os registros de ônibus, conforme necessário.
 - Além disso, há um botão para a geração de QR Code, criado com base no ID único de cada ônibus. Esse QR Code pode ser impresso e fixado no interior do veículo, facilitando o controle de presença dos alunos.

Figura 2: Tela do site (Ônibus)

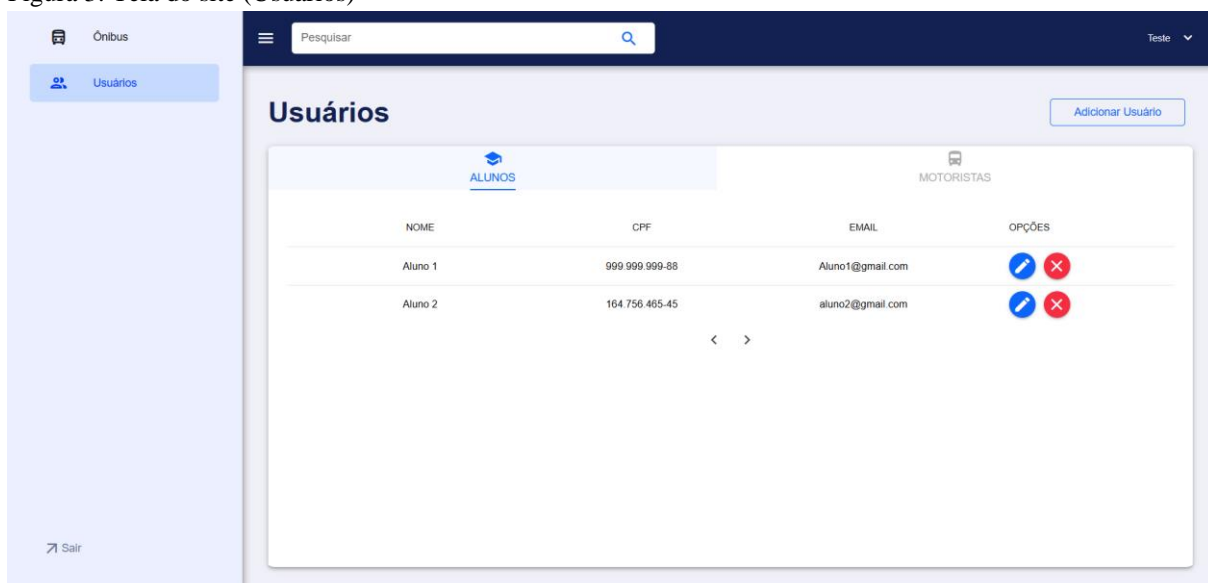


Fonte: Do autor (2024)

b. Tela de cadastro de Usuários (Figura 3):

- A tela de usuários é dividida em duas categorias: alunos e motoristas. Ao adicionar um usuário, o administrador pode especificar o tipo de usuário, incluindo dados como nome, CPF, data de nascimento, e-mail e telefone.

Figura 3: Tela do site (Usuários)



Fonte: Do autor (2024).

No caso de alunos, é possível associá-los a uma escola específica e a um ônibus, facilitando o controle e o monitoramento de seu transporte escolar. Para motoristas, o processo é semelhante, porém sem o campo de associação à escola, pois esse dado é específico dos alunos. Uma vez criado, o usuário pode ser editado ou ter seu status alterado para inativo, caso necessário, permitindo uma gestão mais dinâmica e atualizada dos registros.

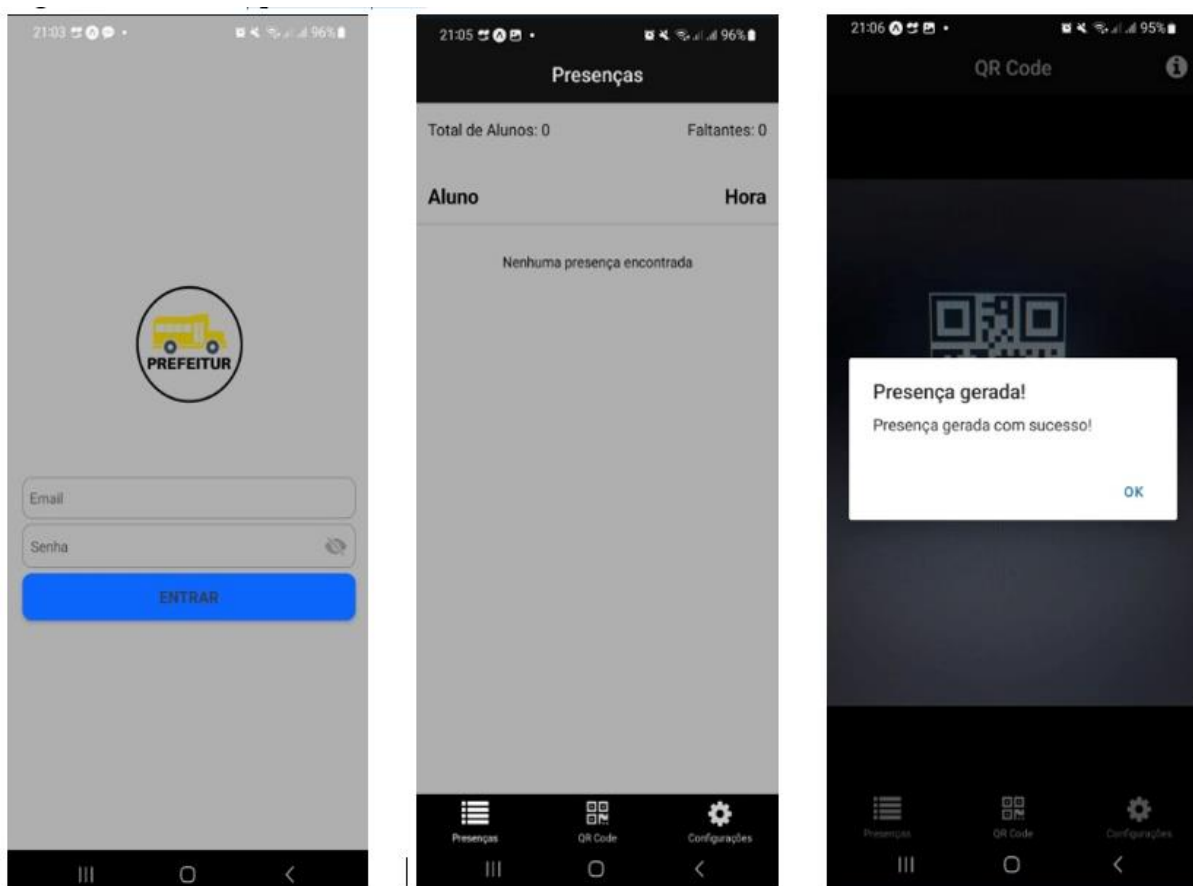
3.4 APLICATIVO MOBILE

O aplicativo mobile foi desenvolvido com o objetivo de otimizar o controle de presença dos alunos no transporte escolar. Utilizando React Native em conjunto com o Expo, o aplicativo permite que os alunos registrem sua presença via leitura de QR Code e possibilita aos motoristas monitorar o status dos alunos em tempo real. A comunicação com a API do sistema é realizada por meio da biblioteca Axios, com armazenamento de dados em MySQL.

Além disso, o Firebase é utilizado exclusivamente para o envio de uma notificação push ao motorista quando todos os alunos estiverem a bordo, sem ter a necessidade de o motorista verificar a lista. Abaixo seguem as funcionalidades das telas:

- **Tela de Login:** Na primeira tela (Figura 4, à esquerda), o aluno ou motorista realiza o login utilizando seu e-mail e senha. Essa etapa é necessária para garantir a segurança e o controle de acesso ao sistema, permitindo que apenas usuários cadastrados tenham acesso às funcionalidades do aplicativo.
- **Tela de Presenças:** Após o login, o usuário é direcionado para a tela de presenças (Figura 4, ao centro). Nesta tela, é exibida uma lista com o nome dos alunos e o horário de registro de presença, além de um contador que mostra o número total de alunos e o número de faltantes.
- **Tela de Leitura de QR Code:** Na terceira tela (Figura 4, à direita), o aluno utiliza seu smartphone para escanear o QR Code fixado no ônibus e registrar sua presença. Após a leitura do código, o aplicativo confirma o registro com a mensagem "Presença gerada com sucesso!", garantindo que o aluno foi contabilizado no sistema de forma prática e rápida.

Figura 4: Telas do aplicativo



Fonte: Do autor (2024).

O motorista possui um acesso similar ao do aluno, com telas que exibem o login e a lista de presenças. No entanto, o motorista não precisa registrar sua presença por meio do QR Code. Sua principal funcionalidade é monitorar o status de presença dos alunos cadastrados, permitindo que ele acompanhe quantos alunos já registraram presença no ônibus e, assim, possa gerenciar o transporte de forma segura e organizada.

As telas apresentadas na Figura 5 exemplificam essa funcionalidade. Na tela à esquerda, o motorista pode visualizar os alunos que entraram no ônibus e registraram presença, com atualizações em tempo real. A tela à direita exibe a contagem decrescente dos alunos faltantes quando eles começam a retornar ao ônibus. Cada aluno, ao validar novamente sua presença ao embarcar, tem seu nome destacado em verde no sistema, indicando que ele já está a bordo. Esse recurso visual facilita a identificação dos alunos presentes e melhora o controle do motorista sobre o transporte.

Quando todos os alunos estão a bordo, uma notificação push é enviada ao motorista, informando que ele pode seguir viagem, pois todos os estudantes já se encontram dentro do veículo.

Figura 5: Telas do aplicativo



Fonte: Do autor (2024).

4. RESULTADOS E DISCUSSÃO

Após a realização dos testes, verificou-se que o sistema desenvolvido foi capaz de atender de maneira eficaz às necessidades de cadastro e gerenciamento de ônibus, motoristas e alunos, proporcionando um fluxo de trabalho organizado e confiável. O site foi projetado para facilitar o processo de registro e atualização de informações, enquanto o aplicativo móvel, também desenvolvido com foco na praticidade, tornando sua principal função validar a presença dos alunos no momento do embarque do ônibus.

Entre os métodos desenvolvidos na API, o gerenciamento de informações de ônibus e escolas envolve validações complexas para garantir que dados, como a identificação de ônibus e motoristas, estejam corretos e atualizados. A atribuição de motoristas aos veículos inclui verificações adicionais para assegurar que apenas motoristas autorizados sejam vinculados, reforçando a segurança do sistema.

Para autenticação e autorização, a API utiliza JWT (JSON Web Tokens), validando o token de cada usuário antes de permitir acesso a operações sensíveis. Funções restritas a administradores têm uma camada adicional de verificação, garantindo que somente usuários com nível "ADMIN" executem essas ações. Esse processo protege os dados e limita o acesso a usuários autorizados.

Os testes foram conduzidos em diferentes cenários, dias e horários para avaliar a eficácia do sistema em condições variadas. Foram utilizados dois modelos de smartphones, Samsung S20+ e Xiaomi Redmi 9, ambos com sistema operacional Android, para validar a funcionalidade de leitura de QR Code. Os cenários de teste incluíram variações de iluminação e de conectividade com a internet. Em condições de baixa luminosidade, observou-se que a câmera de menor qualidade do Xiaomi Redmi 9 apresentou um tempo de resposta ligeiramente maior, demandando tentativas adicionais em alguns casos.

Durante os testes, avaliou-se o tempo médio de resposta do sistema para a captura e processamento do QR Code. Em média, a leitura e o registro de presença levaram aproximadamente 2 segundos para serem concluídos em cada dispositivo, o que evidencia a eficiência do algoritmo de leitura de QR Code implementado no aplicativo.

O sistema foi desenvolvido para operar com conexão à internet, essencial para o registro de presença dos alunos na tabela de dados. Em áreas onde o ônibus passa sem cobertura de internet, o funcionamento do sistema é prejudicado, pois o registro de presença não pode ser efetuado nem atualizado no servidor, comprometendo a sincronização imediata das informações. Essa dependência de conexão garante que todas as informações estejam centralizadas e atualizadas, mas limita a funcionalidade em locais com conectividade instável ou inexistente.

Como no projeto Safe-School-Bus de Almeida e Aguiar (2020), foram utilizados métodos que permitem ao responsável pelo aluno participar ativamente do monitoramento do transporte escolar. O sistema oferece funcionalidades em que o responsável pode escolher o motorista da rota e acessar a lista de alunos vinculados. Além disso, ao iniciar o trajeto, o motorista utiliza a função 'ir ao destino', que gera automaticamente uma rota no aplicativo. Durante o percurso, assim que o aluno embarca, o sistema atualiza seu status, permitindo que o responsável acompanhe a localização e presença do aluno no veículo.

No trabalho de Pinto (2017), também há uma similaridade, pois, apesar de o aluno estar previamente cadastrado no sistema, é o responsável que informa ao motorista se o filho estará presente ou ausente no transporte em determinado dia.

Analisando o trabalho CEAT App (2021), observa-se que foi utilizada a leitura de QR Code para controle e validação dos alunos no transporte escolar. O sistema adota uma abordagem em que o QR Code é impresso nas carteirinhas dos alunos, que são escaneadas pelos monitores dos ônibus para verificar a presença e autenticação de cada estudante.

No trabalho de Silva (2013), o sistema COADS apresenta um modelo eficaz para controle de presença e gestão de transporte, utilizando um leitor de código de barras para validar a presença dos alunos. Os alunos utilizam uma carteira de estudante com código de barras, que é escaneado pelo motorista ao embarcarem no veículo. Esse sistema permite verificar dados

como situação cadastral e status de pagamento de mensalidades, mantendo um histórico de frequência atualizado em uma interface desktop.

Apesar de compartilhar o objetivo de melhorar a gestão do transporte escolar, o sistema proposto neste trabalho se diferencia ao substituir o código de barras pelo QR Code, que oferece maior capacidade de armazenamento de informações e compatibilidade com dispositivos móveis. Além disso, a validação da presença dos alunos é realizada diretamente pelo aplicativo, dispensando a intervenção do motorista, o que aumenta a autonomia do processo e reduz o tempo necessário para as operações.

Os trabalhos correlatos analisados forneceram uma base essencial para o desenvolvimento deste projeto, oferecendo soluções e metodologias que inspiraram melhorias no controle de presença e na autonomia dos alunos no processo de autenticação. O estudo das diferentes tecnologias aplicadas, como a validação de presença por meio de QR Code, o acompanhamento e o uso de leitores de código de barras, permitiu identificar as melhores práticas e adaptar funcionalidades que atendem às demandas atuais do transporte escolar.

5. CONCLUSÃO

O presente trabalho teve como objetivo desenvolver um protótipo para controle de presença de alunos no transporte escolar universitário, com um aplicativo móvel que utiliza validação de presença por QR Code, para o desenvolvimento do protótipo final, foram implementados um site, uma aplicação mobile desenvolvida exclusivamente para Android, uma API para comunicação entre os componentes e uma modelagem de banco de dados. Cada um desses elementos foi projetado para garantir a integração e o funcionamento eficiente do sistema, permitindo o gerenciamento de dados, a validação de presença e a comunicação.

Embora o protótipo atual tenha sido desenvolvido apenas para dispositivos Android, futuras implementações poderão incluir compatibilidade com iOS, ampliando o alcance e flexibilidade do sistema. Além disso, como o sistema depende de conexão com a internet para o registro e sincronização de dados, sugere-se o desenvolvimento de uma funcionalidade offline via Bluetooth, permitindo o registro de presença em áreas sem conectividade e a sincronização dos dados quando a conexão for restabelecida. A adição de geolocalização e a geração de relatórios detalhados também são melhorias potenciais que aumentariam a segurança, possibilitariam o acompanhamento do trajeto e fornecendo informações detalhadas sobre a frequência e o uso do transporte escolar.

Para aumentar a eficiência e adaptabilidade do sistema, novos métodos de identificação podem ser explorados, como o uso de NFC (Near Field Communication) e reconhecimento facial, que poderiam atuar como alternativas ao QR Code. Esses métodos de autenticação complementaríamos o sistema atual, oferecendo maior flexibilidade para diferentes contextos de uso e facilitando ainda mais o monitoramento seguro dos alunos durante o transporte.

REFERÊNCIAS

ALMEIDA, João; AGUIAR, Marcos. *Safe-School-Bus: Monitoramento e segurança no transporte escolar*. São Paulo: Universidade de São Paulo, 2020.

ALMEIDA, Linykeer Renato Silva; AGUIAR, Vinícius Garcia Morgan de. *Desenvolvimento de Aplicativo Mobile para Segurança do Transporte Escolar*. Trabalho de Graduação – Faculdade de Tecnologia de Franca “Dr. Thomaz Novelino”, Centro Paula Souza, Franca/SP, 2020.

BRASIL. Lei n.º 10.880, de 9 de junho de 2004. Institui o Programa Nacional de Apoio ao Transporte do Escolar - PNATE. Diário Oficial da União, Brasília, 10 jun. 2004.

BRASIL. Programa Caminho da Escola. Ministério da Educação, 2007.

CEAT App. Revista Brasileira de Desenvolvimento, Curitiba, v.7, n.9, p. 90329-90347, sep. 2021.

COADS. Controle de empresas com planilhas eletrônicas e desafios operacionais. 2013.

FIELDING, R.; TAYLOR, R. Architectural Styles and the Design of Network-based Software Architectures. *Doctoral Dissertation*. University of California, Irvine, 2000.

JONES, M.; HARDT, D. JSON Web Token (JWT): Authentication and Authorization in Distributed Systems. Proceedings of the International Conference on Security, 2015.

PINTO, Mateus Ferreira. *Projeto de aplicativo para controle de presença e organização de transporte escolar universitário*. Goiânia: Universidade Federal de Goiás, 2017.

SOUZA, M. de F. A utilização do QR Code em processos de monitoramento e controle: uma análise aplicada. Revista de Tecnologia e Inovação, v. 12, n. 3, p. 45-62, 2020.

VILELA, J.V.D.; CAMARGO, V.H.N. Sistema Web para gerenciamento de presença utilizando QR Code via smartphone. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação), Universidade Federal de Goiás.

KATO, H.; TAN, K. Pictorial pattern for decoding and processing two-dimensional code. *Proceedings of the Asia-Pacific Conference on Information Processing*, 1999.