



UFRR

UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS PRADO RIBEIRO

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE RECARGA DE
CARTÃO DE ÔNIBUS UTILIZANDO NFC**

Boa Vista - Roraima,
2024

LUCAS PRADO RIBEIRO

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE RECARGA DE
CARTÃO DE ÔNIBUS UTILIZANDO NFC**

Trabalho de Conclusão de Curso
apresentado ao curso de Bacharelado em
Ciência da Computação do
Departamento de Ciência da
Computação da Universidade Federal de
Roraima, em cumprimento às exigências
legais como requisito parcial à obtenção
do título Bacharel em Ciência da
Computação

Orientador: Dr. Herbert Oliveira Rocha

Boa Vista - Roraima,
2024

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me dar forças ao longo dessa jornada. Agradeço também à minha família por estar sempre ao meu lado em todos os momentos, pelos conselhos e pelo apoio incondicional, aos meus amigos e colegas pela companhia, e aos meus queridos professores pelos ensinamentos e por contribuírem para o meu desenvolvimento. Um agradecimento especial ao professor Herbert, meu orientador, por sua orientação e dedicação.

SUMÁRIO

1. Introdução	4
1.1 Objetivo geral	5
1.2. Perguntas de pesquisa	5
1.3. Objetivos	5
2. Fundamentação Teórica	5
2.1 Flutter	6
2.1.1 Dart	7
2.2 Tipos de banco de dados	8
2.2.1 MySQL	8
2.2.2 Firebase	9
2.3 NFC	10
2.4 Geolocalização	11
2.4.1 GPS	11
2.4.2 LoRaWAN	12
3. Trabalhos Relacionados	12
4. Método Proposto	14
4.1 Modelagem de Dados	15
4.2 Login e cadastro de conta	16
4.3 Acesso a conta e cadastro de cartão	17
4.4 Consulta e recarga de crédito do cartão	18
4.5 Localização GPS do ônibus	18
5 Planejamento da Avaliação Experimental	19
5.1 Testes com Usuários Reais	19
5.2 Testes Automatizados	20
6. Considerações Parciais	20
7. Cronograma	20
8. Referências	21

PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE RECARGA DE CARTÃO DE ÔNIBUS UTILIZANDO NFC

Lucas Prado Ribeiro

Curso de Bacharelado em Ciência da Computação - Departamento de Ciência da Computação -

Universidade Federal de Roraima, Brasil

lukasprado260@gmail.com

Resumo. *O transporte público em Boa Vista/RR utiliza créditos de cartões via RFID como passagem, porém, há necessidade de um sistema eficiente de recarga desses cartões. O objetivo deste estudo é modelar e avaliar um sistema com uma aplicação móvel em Flutter utilizando a tecnologia NFC para simplificar a recarga e consulta de créditos dos cartões de ônibus. O aplicativo oferece uma interface para realizar pagamento de recargas, visualizar o saldo do cartão, enviar alertas de baixo saldo e possibilitar a consulta da localização dos ônibus via GPS. Além disso, complementarmente será implementado um sistema com leitor RFID e módulo GPS para simular os ônibus. Assim, a aplicação aumenta a eficiência e conveniência no sistema de recarga de cartões de ônibus.*

Palavras-chave: *Aplicativo móvel, RFID, Flutter.*

Abstract. *Public transportation in Boa Vista/RR uses RFID card credits as tickets, however, there is a need for an efficient system to recharge these cards. The objective of this study is to model and evaluate a system with a mobile application in Flutter using NFC technology to simplify the recharge and consultation of bus card credits. The application offers an interface to make recharge payments, view the card balance, send low balance alerts and enable the consultation of the location of buses via GPS. In addition, a system with an RFID reader and GPS module will be implemented to simulate the buses. Thus, the application increases the efficiency and convenience of the bus card recharge system.*

Keywords: *Mobile application, RFID, Flutter.*

1. Introdução

Constata-se que o transporte público é o meio mais utilizado por milhões de pessoas para se deslocarem ao trabalho, estudo e lazer, especialmente em áreas metropolitanas densamente povoadas. Dentro desse contexto, os passageiros fazem uso de cartões de ônibus para pagamento de tarifas, prática que tem se destacado como eficiente e conveniente. No entanto, as filas são um problema constante para quem necessita comprar ou recarregar seu bilhete, não sendo raros os dias em que o sistema está lento ou, muitas vezes, inoperante. Assim, o usuário perde muito do seu tempo nas filas (RICARDO, FREITAS, 2017).

Esperar por um ônibus pode ser extremamente demorado e ineficiente, com muitas pessoas tendo que esperar mais de 30 minutos pela chegada do ônibus. A falta de sistemas precisos de rastreamento contribui para esse problema, tornando impossível saber a localização exata dos ônibus. Além disso, a necessidade de comprar passagens com dinheiro em espécie e lidar com troco ao entrar no ônibus gera desconforto adicional para os passageiros (SHAH, et al. 2020).

A eficiência de uma aplicação móvel para gerenciamento de recarga de cartões de ônibus e localização oferece uma solução prática e acessível para os usuários do transporte público. No trabalho de SHAH et al., 2020, é apresentada uma solução de sistema que fornece aos passageiros a localização em tempo real do ônibus, economizando assim o tempo que seria desperdiçado enquanto esperam pelo ônibus. O sistema de bilhetagem no sistema proposto é completamente sem dinheiro e sem complicações sendo a tarifa deduzida diretamente na carteira do aplicativo e ainda possui análises criadas com base na simulação do sistema para ajudar a gestão dos ônibus no planejamento adequado do cronograma dos ônibus.

O uso de tecnologias como comunicação por proximidade, aplicativos móveis, geolocalização e computação em nuvem representa um avanço significativo na modernização do sistema de transporte público. Ao adotar soluções digitais e móveis, as empresas de transporte e os órgãos responsáveis podem otimizar processos, reduzir custos operacionais e oferecer serviços mais eficientes e adaptados às demandas contemporâneas do que os existentes como a recarga em pontos físicos que só aceitam dinheiro em espécie (RICARDO e FREITAS, 2017).

A solução apresentada neste estudo consiste na criação de um aplicativo para celular que utiliza a tecnologia NFC (*Near Field Communication*) para simplificar o gerenciamento dos créditos dos cartões de ônibus, eliminando a necessidade de deslocamento para os pontos físicos de recarga. Além disso, o aplicativo permitirá que os usuários analisem a localização dos ônibus em tempo real através de GPS, o que contribuirá para uma experiência de transporte público mais eficiente, reduzindo filas e tempo de espera.

1.1 Objetivo geral

O objetivo deste estudo é projetar e desenvolver um sistema computacional que simplifique o processo de recarga de crédito e a consulta de saldo do cartão de transporte público, utilizando tecnologias de comunicação por radiofrequência e rastreamento por GPS. A aplicação permitirá a consulta em tempo real da localização dos ônibus, contribuindo para uma experiência mais conveniente e informada para os usuários do transporte público.

1.2. Perguntas de pesquisa

Como o desenvolvimento de um sistema computacional para uso e rastreamento de ônibus comerciais pode facilitar a recarga e estimativa da localização do ônibus pode contribuir para a melhoria da experiência de transporte público na região ?

1.3. Objetivos

Os objetivos específicos desta pesquisa buscaram-se:

1. Projetar um aplicativo móvel para se comunicar com o sistema computacional no ônibus;
2. Projetar um sistema computacional para pagamento e rastreamento de ônibus;
3. Estabelecer políticas de gerenciamento de alertas ao usuário do ônibus;
4. Validar o método proposto por meio de testes funcionais e casos de uso simulado.

2. Fundamentação Teórica

O principal objetivo deste capítulo é apresentar os conceitos que são necessários para o entendimento da solução deste trabalho. Os assuntos abordados neste capítulo são: Flutter, tipos de banco de dados, NFC e geolocalização.

2.1 Flutter

O Flutter é o kit de desenvolvimento de software (SDK) de interface do usuário portátil do Google para criar aplicações móveis, web e de desktop nativamente compiladas. Ele oferece um ambiente completo com framework, widgets e ferramentas. Flutter também é de código aberto e gratuito, o que significa que pode ser utilizado de maneira simples (BHAGAT, et al. 2022).

Entre as principais vantagens do Flutter está a sua capacidade de compilar diretamente para código de máquina, o que elimina problemas de performance encontrados em outros frameworks que dependem de representações intermediárias. Essa eficiência é essencial para garantir que as funcionalidades de recarga e consulta de saldo, bem como a localização dos ônibus via GPS, sejam realizadas de forma ágil e sem interrupções (BHAGAT, et al. 2022).

Outra característica importante do Flutter é o recurso de *hot reload*, que permite a visualização imediata de alterações no código, facilitando o desenvolvimento e a implementação de melhorias na aplicação, tornando-o mais próximo do código de máquina e permitindo que funcione de maneira mais ágil (BHAGAT, et al. 2022).

O conceito central do framework Flutter são os widgets. A vasta gama de widgets disponíveis e altamente personalizáveis facilita a criação de interfaces intuitivas e eficientes, o que melhora a experiência do usuário. Widgets são basicamente componentes da interface do usuário usados para criar a interface de usuário do aplicativo (BHAGAT, et al. 2022).

```
1. void main() {  
2.   runApp(const MyApp());  
3. }  
4.  
5. class MyApp extends StatelessWidget {  
6.   const MyApp({Key? key}) : super(key: key);  
7.  
8.   @override  
9.   Widget build(BuildContext context) {  
10.    return MaterialApp(  
11.      home: Scaffold(  
12.        appBar: AppBar(  
13.          title: Text('Flutter Demo'),  
14.        ),  
15.        body: Text('Hello world :)'),  
16.      ),  
17.    );  
18.  }  
19. }
```

Código 1: Exemplo de código Flutter - Hello World, fonte Própria.

Por exemplo, a hierarquia de widgets do aplicativo "Hello World" apresentado no Código 1 segue assim:

1. MyApp é o widget criado pelo usuário e é construído usando o widget nativo do Flutter, MaterialApp.
2. MaterialApp possui uma propriedade home para especificar a interface do usuário da página inicial.
3. Scaffold é o widget nativo do Flutter e tem duas propriedades — body e appBar.
4. Body é usado para especificar sua interface principal, e appBar é usado para especificar sua interface de cabeçalho.

Como exemplo de utilização do framework, temos o trabalho proposto por Weber e Cantarelli (2020), este trabalho desenvolveu uma aplicação móvel capaz de realizar agendamentos e controle de horários de atendimento para clientes e prestadores de serviços, atingindo todos objetivos traçados e entregando uma aplicação com alta qualidade e fidelidade aos requisitos funcionais traçados, onde a escolha do Flutter como framework de desenvolvimento possibilitou uma integração com mais plataformas, entregando uma solução mais com interfaces intuitivas com a utilização de widgets.

2.1.1 Dart

A linguagem de programação Dart, utilizada no desenvolvimento de aplicações Flutter, é otimizada para a criação de aplicativos rápidos em qualquer plataforma. Com uma sintaxe semelhante a outras linguagens populares, sua curva de aprendizado é relativamente baixa para desenvolvedores familiarizados com essas tecnologias. Como uma linguagem orientada a objetos, Dart suporta conceitos fundamentais como classes, herança, interfaces e tipagem opcional. Isso assegura que as aplicações desenvolvidas com Dart sejam indistinguíveis de aplicações nativas em nível de máquina (BHAGAT et al., 2022).

Entre os principais recursos do Dart, destaca-se o fato de ser uma linguagem de código aberto, com documentação detalhada e uma comunidade de desenvolvedores em crescimento. Desenvolvida pelo Google e aprovada pelo padrão ECMA, Dart é distribuída sob uma licença BSD, facilitando o aprendizado e a implementação de soluções (BHAGAT et al., 2022).

A capacidade de executar código em várias plataformas, como Windows, Linux e macOS, por meio da Dart Virtual Machine, amplia ainda mais seu uso em diferentes ambientes. Outra característica importante são os recursos de compilação JIT (Just in Time) e AOT (Ahead of Time), que aumentam o desempenho do Dart. A compilação JIT permite recargas rápidas de código, enquanto a compilação AOT proporciona um tempo de inicialização rápido e um desempenho superior — fatores cruciais para aplicações que exigem respostas ágeis (BHAGAT et al., 2022).

```
1 class Spacecraft {  
2   String name;  
3   DateTime? launchDate;  
4   int? get launchYear => launchDate?.year;  
5   Spacecraft(this.name, this.launchDate) {
```



```

6    }
7    Spacecraft.unlaunched(String name) : this(name, null);

8    void describe() {
9        print('Spacecraft: $name');
10       var launchDate = this.launchDate;
11       if (launchDate != null) {
12           int years = DateTime.now().difference(launchDate).inDays ~/ 365;
13           print('Launched: $launchYear ($years years ago)');
14       } else {
15           print('Unlaunched');
16       }
17   }
18 }

```

Código 2: Exemplo de código Dart class, fonte Dart (2024)

O código 2 apresenta um exemplo de uma classe com três propriedades, dois construtores e um método. Uma das propriedades não pode ser definida diretamente, sendo acessada por meio de um método getter (em vez de uma variável). O método utiliza interpolação de string para imprimir equivalentes de string de variáveis dentro de literais de string.

Entre as vantagens do Dart, conforme apresentado neste código, destaca-se:

Estrutura de Programação Orientada a Objetos: A estrutura da classe “Spacecraft” exemplifica a programação orientada a objetos, onde propriedades e métodos são encapsulados em um único objeto, promovendo uma melhor organização e reutilização do código. Isso também facilita a manutenção, já que as funcionalidades relacionadas à espaçonave estão agrupadas.

Null Safety: A utilização do tipo “DateTime?” para a propriedade “launchDate” e “int?” para o getter “launchYear”, nas linhas 3 e 4 do código 2, demonstra a funcionalidade de null safety do Dart. Essa característica garante que as variáveis sejam tratadas de forma segura, evitando erros em tempo de execução relacionados a valores nulos.

2.2 Tipos de banco de dados

Os bancos de dados são fundamentais para a gestão e armazenamento de informações em diversos contextos, e sua escolha pode impactar significativamente o desempenho e a escalabilidade de uma aplicação. Nesta seção, exploraremos dois tipos principais de bancos de dados: relacionais e não relacionais, com foco nas suas características e adequação a diferentes cenários de desenvolvimento.

2.2.1 MySQL

Os bancos de dados relacionais, como o MySQL, organizam dados em tabelas estruturadas, onde as informações são armazenadas em linhas e colunas. Cada linha representa um registro, e cada coluna um atributo específico do registro. A estrutura relacional permite a definição clara de

entidades e seus relacionamentos, facilitando a manipulação e consulta dos dados através da linguagem SQL (Structured Query Language). Este modelo é altamente eficiente para aplicações que necessitam de transações complexas e integridade referencial, garantindo que os dados sejam consistentes e que as relações entre eles sejam bem definidas no início.

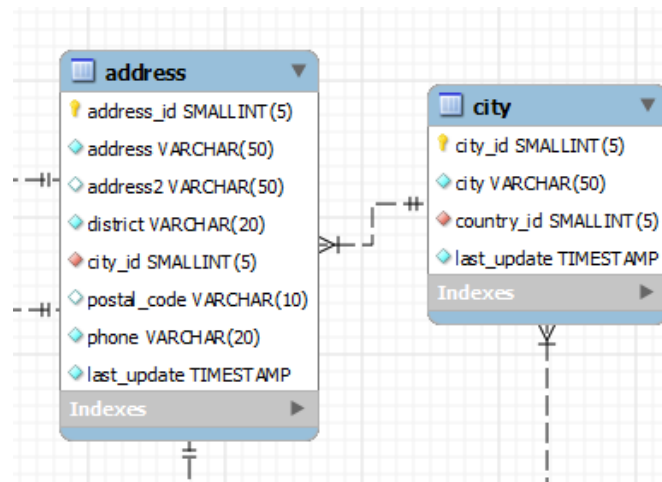


Figura 1. Exemplo de estrutura banco de dados MySQL, fonte MYSQL (2024)

Na Figura 1, refere-se à visualização e manipulação das propriedades das tabelas em um diagrama de entidade-relacionamento estendido (EER) no MySQL Workbench, onde a modelagem de objetos acontece, e que possui duas tabelas “address” e “city” relacionadas pela chave estrangeira “city_id”.

2.2.2 Firebase

Por outro lado, os bancos de dados não relacionais, como o Firebase, oferecem uma abordagem diferente para o armazenamento de dados. No Firebase, os dados são armazenados em formato JSON e sincronizados em tempo real entre todos os clientes conectados. Este modelo é ideal para aplicações que requerem atualizações instantâneas e interatividade, especialmente em ambientes móveis. (SUDIARTHA, et al. 2020)

Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web que oferece uma variedade de serviços, incluindo autenticação de usuários, banco de dados em tempo real, armazenamento em nuvem, e muito mais. Uma das principais vantagens do firebase é principalmente por possuir um sistema próprio de autenticação com segurança pré-configurado: o Firebase Authentication, ele é utilizado para gerenciar o login e cadastro dos usuários, e o Firestore, que é o banco de dados NoSQL do Firebase, utilizado para armazenar dados de usuários e cartões. O Firestore sincroniza automaticamente os clientes com os dados mais recentes, é escalável e ideal para aplicações que requerem uma resposta rápida e capacidade de lidar com grandes volumes de dados. (FIREBASE, 2024)

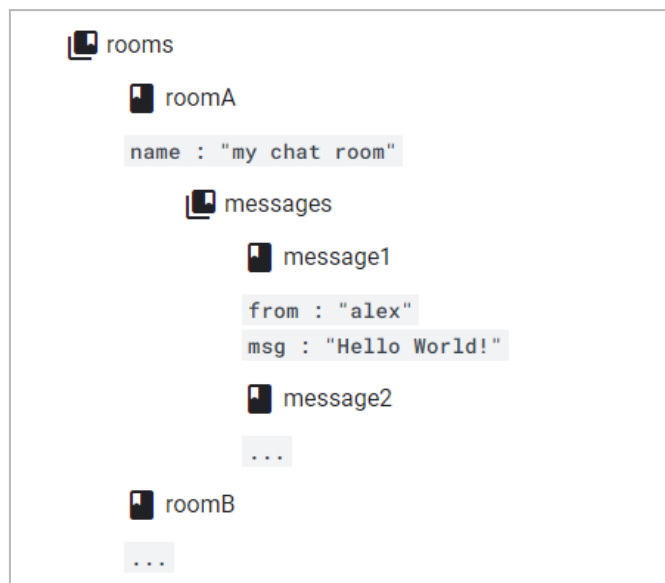


Figura 2. Exemplo de estrutura banco de dados Firebase, Fonte Firebase (2024).

O Cloud Firestore do firebase organiza os dados em documentos e coleções, permitindo uma estrutura hierárquica onde documentos podem conter subcoleções, conforme mostrado na Figura 2. A coleção "rooms" é a coleção principal que contém as salas de chat, enquanto a subcoleção "messages" contém as mensagens associadas a cada sala. Essa estrutura é especialmente útil para aplicativos que requerem relações complexas entre dados.

2.3 NFC

Near Field Communication (NFC) é uma tecnologia de comunicação sem fio de curto alcance que permite a troca de dados entre dispositivos quando estão próximos um do outro. Essa tecnologia é amplamente utilizada em pagamentos móveis, cartões de transporte público e sistemas de controle de acesso devido à sua conveniência e segurança. Os desenvolvedores de aplicativos móveis podem projetar soluções personalizadas que interagem com tags NFC e outros dispositivos, permitindo a leitura e gravação de dados por meio da conexão baseada em toque (Coskun et al., 2012).

As tags NFC são cartões de memória sem contato que armazenam dados em um formato especial chamado NDEF (NFC Data Exchange Format). Essas tags vêm em uma variedade de formatos, permitindo diversas implementações em termos de tamanhos, capacidades, custos e funcionalidades. O NFC definiu cinco tipos diferentes de tags que variam no protocolo de comunicação subjacente e na estrutura de dados para armazenar mensagens NDEF, mas o comportamento geral dessas tags é similar (Coskun et al., 2012).

No contexto do desenvolvimento de aplicativos com Flutter, o pacote "NFC_manager" se destaca por fornecer funcionalidades que permitem aos desenvolvedores incorporar operações NFC em seus projetos. Ao utilizar esse pacote, os desenvolvedores podem implementar características como a leitura e escrita de tags NFC, facilitando a interação entre o aplicativo e os cartões de transporte. Por exemplo, ao clicar em uma área específica do aplicativo, o usuário é direcionado a uma página dedicada à funcionalidade NFC. Nesta página, podem ser apresentadas opções como leitura de tags e escrita de dados, permitindo ao usuário visualizar informações relevantes, como o saldo disponível no cartão.

A implementação do NFC em aplicativos Flutter é exemplificada através da utilização de funções do pacote NFC_manager, onde ao interagir com a interface do usuário, o aplicativo pode verificar se a funcionalidade NFC está disponível no dispositivo e, em seguida, iniciar uma sessão

de leitura de tag. Este processo não apenas melhora a experiência do usuário, mas também facilita a gestão e a recarga de cartões de transporte, promovendo a inovação no setor de mobilidade urbana (WANG, 2022).

O projeto apresentado por RICARDO e FREITAS (2017), do sistema mobile para recarga de cartão para transporte público dos usuários do transporte público de São Paulo, dá um exemplo de como a tecnologia NFC se encaixa neste tipo de problema. Onde utiliza a tecnologia NFC (Near Field Communication) ou Comunicação por Campo de Proximidade para realizar a transferência de dados de um dispositivo para outro, bastando encostar o cartão na parte de trás do aparelho para fazer o pareamento das informações.

2.4 Geolocalização

A geolocalização é a técnica que permite determinar a localização de um dispositivo com base em suas coordenadas geográficas. Este recurso é cada vez mais utilizado em diversas aplicações, especialmente devido à crescente demanda por soluções que integrem dados de localização em contextos econômicos, como a maximização de vendas por meio do processamento de informações geográficas. Tecnologias modernas, como o desenvolvimento de constelações de satélites, exigem precisão elevada na medição de distâncias e separações entre pontos coordenados (Japara et al., 2023).

No contexto do Flutter, a implementação da geolocalização é facilitada pelo uso de bibliotecas como geolocator e location, que fornecem acesso fácil às informações de localização do dispositivo. Estas bibliotecas permitem que os desenvolvedores implementem funcionalidades como rastreamento de localização em tempo real e cálculo de distâncias entre coordenadas. O geolocator, por exemplo, oferece métodos para obter a posição atual do dispositivo, além de calcular a distância entre diferentes pontos geográficos (Japara et al., 2023).

Esses recursos são especialmente úteis em aplicativos que exigem localização precisa, como serviços de entrega e navegação. A integração da geolocalização em aplicativos Flutter não apenas melhora a experiência do usuário, mas também possibilita a criação de soluções inovadoras que atendem às necessidades do mercado atual, tornando a geolocalização uma competência essencial para desenvolvedores que desejam se destacar na criação de software moderno (Japara et al., 2023).

2.4.1 GPS

O GPS é uma tecnologia essencial que possibilita a localização precisa de objetos e pessoas globalmente. Utilizando uma rede de satélites em órbita terrestre, o GPS transmite sinais para receptores presentes em dispositivos como smartphones, carros e relógios. Dessa maneira, ao receber e processar esses sinais, o GPS consegue calcular com precisão a localização exata de um objeto ou pessoa em qualquer ponto do planeta (FIGUEIRA, 2024).

Um dos maiores desafios do sistema de ônibus atual é a falta de um sistema eficiente de rastreamento. Sem um monitoramento preciso, os passageiros ficam sem informações sobre a chegada do próximo ônibus (SHAH, et al. 2020). Com o GPS, é possível obter informações em tempo real sobre a localização dos ônibus, permitindo que os passageiros planejem melhor suas viagens e diminuam o tempo de espera nas paradas. Além disso, um sistema de rastreamento pode melhorar a eficiência operacional das empresas de transporte, otimizando rotas e horários dos ônibus com base nos dados coletados.

O sistema proposto por SHAH, et al. (2020) fornece aos passageiros informações em tempo real da localização do ônibus. Utilizando um módulo SIM808 GPS GSM instalado em cada ônibus, onde cada módulo irá buscar a latitude e longitude do ônibus e atualizá-lo no banco de dados. Assim, o banco de dados consistirá no número do ônibus e sua localização atual. Economizando assim seu tempo que caso contrário, será desperdiçado enquanto espera pelo ônibus.

2.4.2 LoRaWAN

LoRaWAN é uma tecnologia de comunicação sem fio projetada para conectar dispositivos IoT em grandes áreas com baixo consumo de energia. Essa rede permite que sensores, atuadores e dispositivos diversos enviem dados por longas distâncias sem precisar de grandes infraestruturas, opera baixo consumo de energia e a capacidade de conectar um grande número de dispositivos a uma única rede. A capacidade de longo alcance da camada física LoRa permite links de salto único entre dispositivos finais e gateways. O sistema suporta comunicação bidirecional e endereçamento multicast para uso eficiente do espectro durante tarefas como atualizações de Firmware Over-The-Air (FOTA). A LoRaWAN tem sido amplamente utilizada em projetos de cidades inteligentes, agricultura de precisão e monitoramento ambiental (LoRa Alliance, 2022).

Um exemplo de aplicação prática de LoRaWAN pode ser observado no estudo de Jabbar (2024), que desenvolveu um sistema IoT baseado em LoRaWAN para monitoramento da qualidade da água em áreas rurais. O sistema utiliza sensores instalados em corpos d'água para medir parâmetros como pH, temperatura e turbidez. Esses sensores transmitem os dados por meio de LoRaWAN para um servidor central, permitindo o monitoramento em tempo real da qualidade da água. A solução mostrou-se eficaz em fornecer dados confiáveis, mesmo em áreas com pouca infraestrutura de telecomunicações, além de oferecer baixo custo de manutenção e longo tempo de operação graças ao consumo reduzido de energia dos dispositivos.

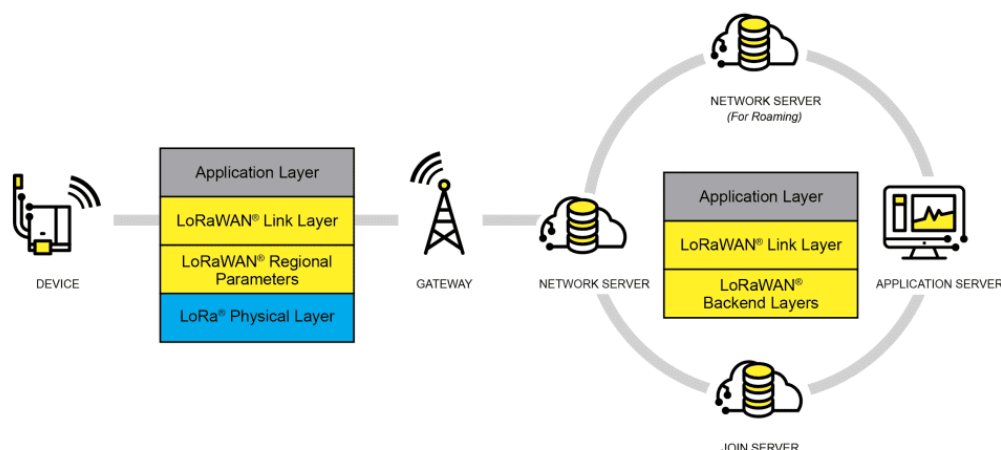


Figura 3. LoRaWAN Architecture, fonte LoRaWAN (2022).

A arquitetura de rede LoRaWAN conforme mostra a Figura 3, usa uma topologia estrela-de-estrelas, onde é composta por várias camadas que colaboram para a comunicação eficiente em redes de IoT. A camada de aplicação interage com os usuários, enquanto a camada LoRaWAN gerencia a comunicação e autenticação entre dispositivos e a rede. O servidor de rede centraliza a gestão do tráfego de dados, enquanto os parâmetros regionais definem especificações operacionais conforme regulamentações locais. Os gateways atuam como intermediários que conectam dispositivos à rede, convertendo sinais RF em pacotes IP, e a camada física utiliza a tecnologia LoRa para permitir comunicação de longo alcance.

3. Trabalhos Relacionados

Este capítulo visa apresentar os principais trabalhos relacionados com o sistema proposto, destacando as diferenças e contribuições ao método proposto. Os estudos abordados utilizam

tecnologias de radiofrequência ou referenciam sistemas que fazem uso dessa ferramenta para criar soluções que resolvam problemas nas compras de passagens de ônibus e no rastreamento dos ônibus.

O estudo de SHAH, et al. (2020) descreve o desenvolvimento de um sistema de reserva automatizada de passagens de ônibus. O sistema proposto calcula a tarifa pela distância percorrida e deduz o valor da tarifa da carteira eletrônica do passageiro, gerenciada no aplicativo móvel S-Bus. O sistema também rastreia a localização dos ônibus em tempo real.

O fluxo do sistema de ônibus inteligente baseado em IOT de SHAH, et al. (2020) envolve o uso de cartões RFID, que contêm um número exclusivo para cada passageiro. Esses cartões devem ser registrados no S-Bus, onde todos os detalhes do passageiro são associados a esse número e armazenados no banco de dados. Em cada ônibus, há duas unidades de leitores RFID, uma na porta de entrada e outra na de saída, que escaneiam os cartões dos passageiros e atualizam suas localizações no banco de dados. Esses leitores estão conectados ao NodeMCU (Módulo WiFi ESP8266) via Arduino Uno, garantindo conectividade ao banco de dados (SHAH, et al. 2020).

Os passageiros podem acompanhar a localização atual do ônibus pelo aplicativo S-Bus, que acessa informações no banco de dados e exibe a posição do ônibus em um mapa do Google. O resumo da viagem, incluindo origem, destino e tarifa, também está disponível no aplicativo (SHAH, et al. 2020). O sistema elimina o uso de dinheiro e papel, substituindo os bilhetes físicos por digitais. As análises do sistema ajudam a gerenciar os horários e rotas dos ônibus.

A pesquisa de NAILA, et al. (2020) propõe um sistema inteligente de bilhetagem de ônibus, melhorando a experiência tanto para usuários quanto para prestadores de serviço. A proposta inclui um aplicativo Android com leitor de QR Code e uma carteira digital. Passageiros podem se registrar no aplicativo, e a conta do condutor é controlada pelo administrador, que fornece suas credenciais. Uma vez logado, tanto o condutor quanto o passageiro devem vincular sua conta bancária ao aplicativo, para que possam transferir, adicionar e retirar dinheiro da carteira do aplicativo.

Ao embarcar, o passageiro vê o destino e a tarifa exibidos com um QR Code no dispositivo. O condutor escaneia o código, e o valor é debitado da carteira do passageiro e creditado à do condutor. Uma confirmação da transação é enviada ao passageiro por mensagem de texto, e todas as atividades são registradas no servidor (NAILA, et al. 2020). O sistema contribui para a sustentabilidade ao eliminar papel e resolver o problema de troco.

O trabalho de KAUSHIK, et al. (2019) apresenta um sistema de bilhetes e identificação de passageiros no transporte público, utilizando IoT para solucionar problemas como divergência de tarifas e falta de autenticação. A solução é dividida em três partes principais: emissão de cartões RFID, geração de bilhetes usando esses cartões, e verificação de bilhetes.

Os cartões RFID podem ser emitidos para passageiros em balcões de pontos de ônibus, onde o gerente do balcão tem um site no qual ele preencherá o formulário de registro com todos os detalhes do passageiro e emitirá um cartão RFID com uma etiqueta exclusiva. A geração do bilhete é automatizada, e o passageiro, após escanear o cartão, tem o valor da passagem deduzido automaticamente de sua conta. Portas de ônibus são abertas por 240 segundos após a validação do bilhete, mas, em caso de saldo insuficiente, a mensagem “Saldo insuficiente” é exibida (KAUSHIK, et al. 2019).

O protótipo do sistema utiliza tecnologias como cartões RFID, Arduino (ATmega328P), LCD para exibição de informações, módulo WiFi (ESP8266) para envio de dados, e servomotores para abertura de portas. O sistema facilita a geração de bilhetes e melhora a segurança e a eficiência do transporte público, além de reduzir o desperdício de papel associado aos bilhetes impressos manualmente. (KAUSHIK, et al. 2019).

O trabalho de RICARDO e FREITAS (2017) descreve um sistema para recarga de cartões de transporte público via celular utilizados por usuários do transporte público da cidade de São Paulo, inicialmente disponível para dispositivos com tecnologia NFC. Ao aproximar o cartão do celular irá sincronizar as informações, o saldo e opções de recarga de crédito são exibidos. O pagamento é feito no próprio celular, que funciona como um posto de recarga portátil.

Apesar das limitações, como a disponibilidade restrita de NFC, o sistema proporciona conveniência aos usuários e busca modernizar o transporte público. Contudo, a adoção do sistema foi limitada, beneficiando cerca de 2,5 milhões de pessoas em São Paulo, o que representa uma pequena parcela dos 10 milhões de passageiros transportados por dia na cidade. As limitações percebidas na elaboração desse trabalho foram as poucas informações disponíveis e também a falta de acessibilidade aos dados da empresa, ao processo de desenvolvimento e à implantação do projeto. (RICARDO e FREITAS, 2017).

As comparações das soluções propostas nos trabalhos correlatos podem ser vistas na Tabela 1.

TRABALHOS	TECNOLOGIA	GPS	RECARGA DE CRÉDITO POR APP	CARTEIRA DIGITAL
SHAH, et al. 2020	HF RFID (High Frequency)	SIM808 GPS GSM	NÃO	SIM
NAILA, et al. 2020	QR CODE	NÃO	NÃO	SIM
KAUSHIK, et al. 2019	HF RFID (High Frequency)	NÃO	NÃO	NÃO
RICARDO, FREITAS, 2017	NFC ISO/IEC 14443	NÃO	SIM	SIM

Tabela 1. Comparação dados trabalhos correlatos, fonte Própria.

A Tabela 1 apresenta uma comparação das tecnologias utilizadas nos estudos revisados, destacando as inovações e limitações nas soluções de bilhetagem e rastreamento de ônibus. Enquanto os trabalhos de SHAH et al. (2020) e NAILA et al. (2020) exploram a integração de tecnologias como RFID e QR Code, KAUSHIK et al. (2019) se concentram exclusivamente em RFID, sem incluir opções de pagamento digital ou GPS. Por outro lado, RICARDO e FREITAS (2017) introduzem a tecnologia NFC, mas limitam-se ao contexto de recargas de cartões, atingindo uma parcela pequena do público.

O diferencial do meu trabalho é a proposta de um sistema que integra todos esses elementos, oferecendo uma solução completa e multifuncional que inclui rastreamento por GPS, recarga de crédito via aplicativo, e o uso de carteiras digitais.

4. Método Proposto

Neste capítulo descreve-se o projeto e fluxo de execução da solução proposta, que utiliza tecnologias de radiofrequência, para gerenciamento dos cartões de ônibus, o banco de dados Firebase e o framework Flutter para interface e principais funcionalidades do sistema como recarga de crédito do cartão utilizando método de pagamento online e a consulta da localização dos ônibus por meio do GPS.

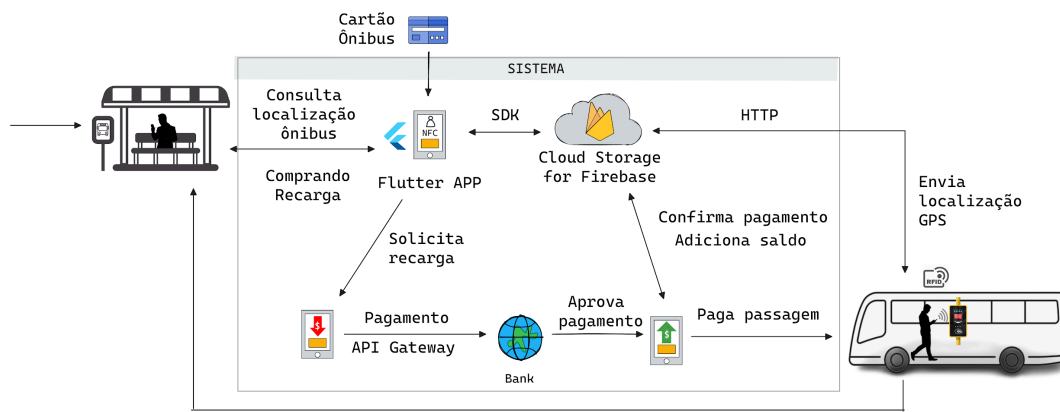


Figura 4. Ciclo de vida do sistema, fonte Própria.

Na Figura 4 é possível analisar como funciona o ciclo de vida do sistema, o usuário pode consultar os créditos do seu cartão que ele irá cadastrar na sua conta no aplicativo por meio da tecnologia NFC, aproximando o cartão no dispositivo, e verificar se possui saldo suficiente para a passagem, caso ele não tenha ele pode recarregar os créditos para o cartão no aplicativo de qualquer lugar, ele realiza o pagamento online do valor do crédito que precisa, o banco aprova o pagamento, manda a confirmação e o sistema irá atualizar em tempo real no banco de dados o crédito do cartão do passageiro, com isso, ele poderá pegar o ônibus e quando ele passar na catraca, o leitor RFID irá debitar o valor da passagem no crédito do cartão de ônibus e irá atualizar o valor no banco de dados.

4.1 Modelagem de Dados

A modelagem de dados para este sistema será baseada no Firebase Firestore, onde os dados são armazenados em documentos que pertencem a coleções. O modelo de dados se organizará em diferentes coleções, cada uma representando uma entidade importante no sistema, como usuários, cartões de ônibus, transações, e localização dos ônibus.

Key	Value	Type
▼ usuarios		Collection
▼ UfBtLK92BYNaDZXSrqf6		Document
▼ cartao		Collection
▼ dQcTJqjkHoHd9p3uYSqm		Document
▼ historico_transacoes		Collection
▼ jg2FAZWSalmdASGZi7hv		Document
data	Oct 4, 2024 12:00:00 AM	Timestamp
tipo	(empty)	String
valor	0	Integer
id_cartao	(empty)	String
saldo	0	Integer
ultima_recarga	Oct 4, 2024 12:00:00 AM	Timestamp
email	(empty)	String
nome	(empty)	String
uid	(empty)	String

Figura 5. Modelo de dados Usuários.

Na figura 5, mostra como foi modelada a coleção de usuários: Cada usuário registrado no sistema possui um documento dentro da coleção usuarios. Este documento armazena os dados pessoais do usuário, como nome, email, e um cartão vinculados à conta. A autenticação é realizada pelo Firebase Authentication, que gera o uid único de cada usuário. Já na subcoleção cartão, cada cartão de ônibus possui um documento na coleção cartão, contendo informações como o identificador único do cartão (id_card), o saldo, e o histórico de transações. Esse histórico é armazenado em uma subcoleção chamada histórico_transacoes, garantindo que as recargas e débitos sejam facilmente rastreados.

Key	Value	Type
▼ onibus		Collection
▼ qSKmapXCV0ZvhT9bUpw8		Document
id_onibus	(empty)	String
rota	(empty)	String
ultima_atualizacao	Oct 4, 2024 12:00:00 AM	Timestamp
▼ ultima_localizacao		Map
latitude	(empty)	String
longitude	(empty)	String

Figura 6. Modelo de dados Ônibus.

Na Figura 6, mostra a coleção do onibus, onde, para monitorar a localização dos ônibus, foi criada a coleção de ônibus, onde cada documento representa um ônibus. A ultima_localizacao é registrada em campos de latitude e longitude, e cada ônibus armazena as atualizações de localização GPS ao longo do tempo.

4.2 Login e cadastro de conta

Nesta seção, será detalhado o processo de login e cadastro de conta, funcionalidade essencial para o acesso ao sistema. A aplicação mobile permite que novos usuários se registrem inserindo suas credenciais, enquanto usuários já cadastrados podem realizar o login para acessar as funcionalidades do sistema, como recarga de crédito e consulta do saldo de cartões. A seguir, cada etapa será explicada com base no fluxo descrito na Figura 7.

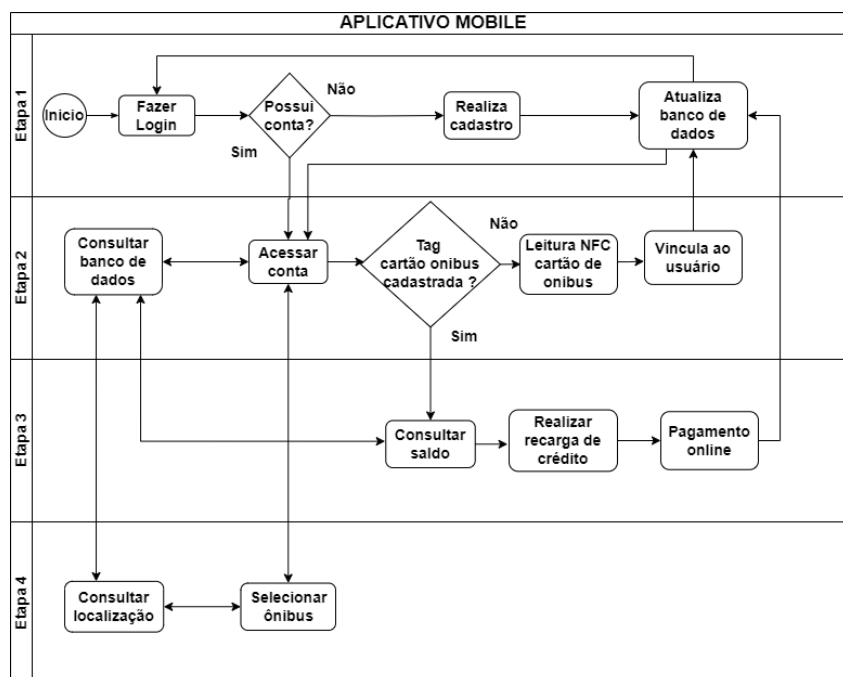


Figura 7. Diagrama de fluxo do app.

Na etapa 1 da Figura 7, será implementado o processo de autenticação utilizando o serviço Firebase Authentication, parte do banco de dados Firebase. A aplicação permite que os usuários façam login utilizando email e senha. Quando um usuário realiza o login, o Firebase gera um token de autenticação seguro que é enviado de volta ao cliente. Este token contém informações codificadas sobre a identidade do usuário e é utilizado para manter a sessão ativa.

Após a autenticação bem-sucedida, os usuários são redirecionados para a interface principal da aplicação, onde podem realizar a leitura e recarga dos cartões NFC. Caso o usuário ainda não tenha uma conta, ele poderá criá-la ao preencher um formulário com e-mail e senha. As informações fornecidas são enviadas ao Firebase, que armazena esses dados de forma segura. O Firebase utiliza criptografia SSL/TLS para proteger as informações transmitidas entre o aplicativo e seus servidores.

4.3 Acesso a conta e cadastro de cartão

Na etapa 2, conforme ilustrado na Figura 7, após o usuário realizar o login, ele deve cadastrar um cartão de ônibus e vinculá-lo à sua conta. Se o usuário já tiver um cartão vinculado, ele terá acesso às funcionalidades do sistema, podendo visualizar informações do cartão como saldo e identificador único (ID) do cartão. Caso contrário, será necessário cadastrar um novo cartão.

Para realizar o cadastro do cartão, o sistema solicita que o usuário aproxime o cartão de ônibus do dispositivo para que as informações possam ser lidas via tecnologia NFC (*Near Field Communication*). A tecnologia NFC permite a troca de dados entre o cartão e o dispositivo em proximidade imediata, garantindo uma comunicação rápida e segura. Quando o usuário aproxima o cartão, o sistema lê os dados necessários e exibe essas informações na tela, pedindo a confirmação do usuário para prosseguir com a vinculação do cartão à conta. Após a confirmação, os dados do cartão são armazenados de maneira segura no banco de dados, com criptografia em repouso para prevenir acessos não autorizados.

As informações do cartão são armazenadas em uma coleção dedicada, utilizando um documento que inclui campos como o identificador único (ID) do cartão, saldo e histórico. Cada

documento é associado ao identificador único (UID) do usuário, garantindo que os dados sejam organizados e facilmente acessíveis.

4.4 Consulta e recarga de crédito do cartão

A principal funcionalidade do sistema é a recarga de crédito do cartão. Nesta etapa, conforme ilustrado na Figura 7, o usuário escolhe o valor que deseja recarregar, seleciona o método de pagamento e realiza o pagamento online através do aplicativo. O pagamento é processado por um *gateway* de pagamento integrado ao aplicativo, que garante a segurança da transação utilizando criptografia para proteger os dados financeiros do usuário. Após a confirmação do pagamento pelo gateway, o sistema atualiza o saldo do cartão no Firestore, vinculado à conta do usuário.

Para registrar todas as transações realizadas, foi implementado um histórico de transações. Cada recarga é registrada com informações detalhadas, como o valor recarregado e a data/hora da transação. Esses dados são armazenados em subdocumentos dentro do documento do cartão no Firestore, permitindo que os usuários visualizem um histórico completo de todas as operações.

Para melhorar a usabilidade e garantir que os usuários não fiquem sem saldo, foi criada políticas de alerta de saldo baixo. Quando o saldo do cartão estiver abaixo de um valor pré-determinado, a aplicação exibe um alerta ao usuário, sugerindo que ele realize uma nova recarga. Isso foi implementado através de verificações periódicas do saldo do cartão, com notificações exibidas na interface do usuário.

4.5 Localização GPS do ônibus

A funcionalidade de rastreamento de localização do ônibus, conforme mostrado na etapa 4 da Figura 7, funciona da seguinte forma: o usuário escolhe o ônibus desejado para consulta, e cada ônibus possui um identificador único (ID) associado às suas coordenadas de latitude e longitude armazenadas no banco de dados.

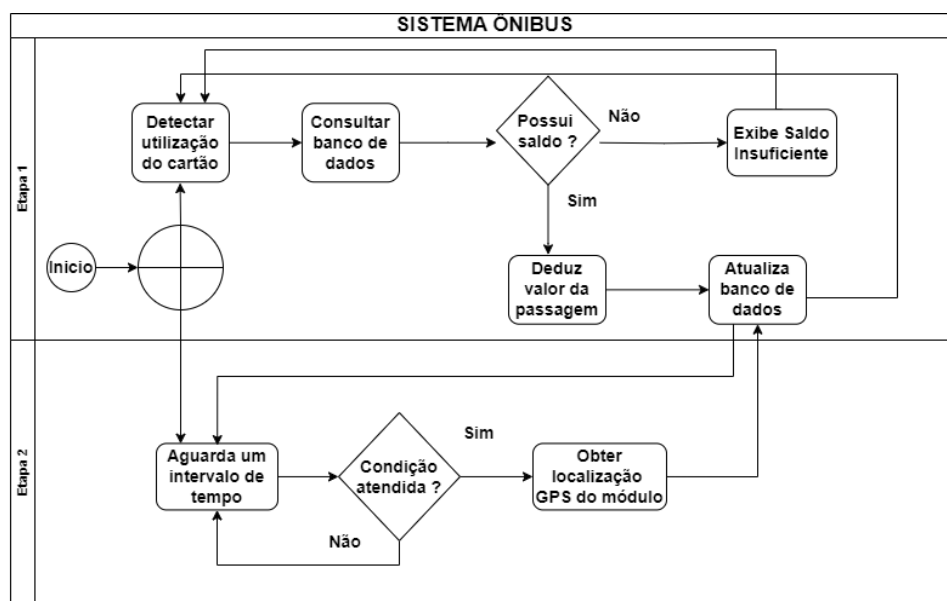


Figura 8. Diagrama de fluxo do sistema de ônibus.

O sistema monitora transações de débito no cartão e, quando detecta uma transação, coleta os dados de GPS de um módulo instalado no leitor RFID para atualizar a localização do ônibus ou caso não tenha nenhuma transação em um intervalo de tempo, conforme mostrado na figura 8.

Cada ônibus está equipado com um leitor RFID, um módulo GPS, um microcontrolador ESP (como ESP8266 ou ESP32) e um módulo de comunicação celular (como SIM800 ou SIM900). Quando um passageiro realiza uma transação com o cartão RFID, o leitor captura a informação do cartão e o módulo GPS fornece a localização atual do ônibus. O ESP centraliza a coleta desses dados e utiliza o módulo celular para enviá-los ao Firestore.

Os dados são transmitidos para o Firestore, onde são armazenados em documentos específicos associados ao ID do ônibus. As informações, como latitude, longitude, data e hora da transação, são registradas em subdocumentos para fácil acesso e atualização.

Para manter a precisão, cada ônibus tem uma rota fixa definida no sistema. Caso a localização do ônibus não seja atualizada dentro de um intervalo de tempo predeterminado, o sistema estimará a posição do ônibus com base na última atualização e no tempo decorrido, ao longo da rota definida. Esse cálculo garante a atualização da localização do ônibus mesmo na ausência de novas transações ou dados de GPS recentes.

Essa funcionalidade proporciona um serviço de rastreamento confiável e seguro, permitindo que os usuários acompanhem a localização dos ônibus em tempo real. A integração da tecnologia GPS com o Firestore e o sistema de cálculo baseado em rotas fixas assegura uma experiência aprimorada, com informações precisas e atualizadas sobre o deslocamento dos ônibus.

5 Planejamento da Avaliação Experimental

O planejamento dos testes de funcionalidade e da avaliação experimental. A avaliação experimental será composta por testes com usuários reais e testes automatizados, visando garantir que o sistema funcione corretamente e atenda aos requisitos de segurança e desempenho.

5.1 Testes com Usuários Reais

No futuro, os testes de funcionalidade serão conduzidos com usuários reais, que simularão cenários cotidianos de uso do sistema. Essa abordagem tem como objetivo avaliar a usabilidade e a eficiência das funcionalidades principais, como a leitura NFC, a recarga de créditos e o rastreamento de ônibus via GPS.

Durante os testes com usuários, será observado:

- **Leitura e Cadastro de Cartões NFC:** Avaliar a facilidade de uso do sistema ao cadastrar e consultar informações dos cartões.
- **Recarga de Créditos:** Testar o processo de recarga com métodos de pagamento reais, verificando o tempo de processamento e a experiência do usuário.
- **Rastreamento de Ônibus:** Analisar a precisão dos dados de GPS e a velocidade de atualização da localização dos ônibus no aplicativo.

Esses testes serão fundamentais para validar a experiência prática dos usuários e identificar ajustes antes da conclusão do projeto.

5.2 Testes Automatizados

Em complemento aos testes com usuários reais, também serão implementados testes automatizados no código para validar a robustez do sistema em termos de segurança e desempenho.

Os testes automatizados terão foco em:

- Validação do processo de pagamento: simulações de transações financeiras garantirão que o saldo do cartão seja corretamente atualizado e protegido.
- Persistência dos dados: verificar se os dados de transações e informações pessoais dos usuários estão devidamente protegidos e acessíveis.
- Recebimento dos avisos: garantia e precisão nos alertas e notificações enviados exatamente quando o saldo atingir ou para receber a confirmação de transação da recarga.

6. Considerações Parciais

Este trabalho propõe o projeto e desenvolvimento de um sistema de recarga de cartão de ônibus utilizando tecnologias de radiofrequência. O real objetivo consiste em oferecer uma solução eficiente para o gerenciamento de cartões de transporte público em Boa Vista.

Entretanto, desafios relacionados à precisão do rastreamento de ônibus em tempo real e à eficiência do processamento de pagamentos online foram identificados como pontos críticos a serem aperfeiçoados nas próximas fases de desenvolvimento. Além disso, o sistema de alertas de saldo baixo, embora funcional, precisa ser refinado para melhorar a experiência do usuário, especialmente em cenários de uso intensivo.

As próximas etapas do projeto envolvem a finalização das funcionalidades de consulta de saldo e recarga, além de aprimorar os mecanismos de rastreamento GPS e comunicação com os servidores do Firestore para garantir atualizações precisas e em tempo real da localização dos ônibus.

7. Cronograma

A Tabela 2 exibe o cronograma proposto para as próximas etapas deste trabalho de conclusão de curso.

Atividade	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março
Revisão de literatura sobre tecnologia NFC	X					
Desenvolvimento do protótipo aplicativo mobile	X	X	X			
Implementação da recarga de créditos		X				
Desenvolvimento do protótipo RFID do ônibus		X	X			

Implementação do rastreamento via GPS			X	X		
Testes de funcionalidade e Finalização			X	X		
Documentação e finalização do TCC					X	X
Apresentação final						X

Tabela 2 – Cronograma de atividades, fonte Própria.

8. Referências

KAUSHIK, Aman; JAIN, Nitin. RFID Based Bus Ticket Generation System. In: 2021 International Conference on Technological Advancements and Innovations (ICTAI). [s.l.]: IEEE, 2021.

Disponível em: <<http://dx.doi.org/10.1109/ictai53825.2021.9673244>>. Acesso em: 27 Jun. 2024.

Naila; Ajith; MURSHID, Muhammad. Smart Bus Ticketing System. IJSET-International Research Journal of Engineering & Technology, v. 7, n. 6, 2020. Disponível em:

<<https://www.irjet.net/archives/V7/i6/IRJET-V7I6182.pdf>>. Acesso em: 27 Jun. 2024.

RICARDO, Solange Cristina; FREITAS, Henrique Mello Rodrigues de. Análise do sistema de Mobile Payment implementado no transporte público na cidade de São Paulo. Revista Gestão & Tecnologia, [S. l.], v. 17, n. 1, p. 258–276, 2017. DOI: 10.20397/2177-6652/2017.v17i1.923.

Disponível em: <https://revistagt.fpl.emnuvens.com.br/get/article/view/923>. Acesso em: 14 set. 2024.

SHAH, Meet J; PRASAD, Rajesh P; SINGH, Aashutosh S. IOT based smart bus system. In: 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA). [s.l.]: IEEE, 2020, p. 130–134. Disponível em:

<<http://dx.doi.org/10.1109/cscita47329.2020.9137816>>. Acesso em: 27 Jun. 2024.

Coskun, V., Ozdenizci, B., & Ok, K. (2013). A Survey on Near Field Communication (NFC) Technology. Wireless personal communications, 71(3), 2259-2294. Disponível em:

<<http://dx.doi.org/10.1007/s11277-012-0935-5>>. Acesso em: 02 Out. 2024.

BHAGAT, Shreya A. et al. Review on mobile application development based on flutter platform. International journal for research in applied science and engineering technology, v. 10, n. 1, p. 803–809, 2022. Disponível em: <<http://dx.doi.org/10.22214/ijraset.2022.39920>>. Acesso em: 11 Set. 2024.

SUDIARTHA, I. K. G.; INDRAYANA, I. N. E.; SUASNAWA, I. W.; et al. Data structure comparison between MySql relational database and firebase database NoSql on mobile based tourist tracking application. Journal of physics. Conference series, v. 1569, p. 032092, 2020. Disponível em: <<http://dx.doi.org/10.1088/1742-6596/1569/3/032092>>. Acesso em: 11 Set. 2024.

WEBER, Érico ; CANTARELLI, Gustavo. Agendei: proposta de desenvolvimento de uma

aplicação móvel para realização de oferta de serviços e controle de agendamentos online. Trabalho de Conclusão de Curso Sistemas De Informação - Universidade Franciscana (UFN), p. 28. 2020. Disponível em: <https://tfgonline.lapinf.ufn.edu.br/media/midias/TFG_2_-_EricoWeber.pdf>. Acesso em: 14 Set. 2024.

LoRa Alliance. (2022). LoRaWAN: Low Power, Wide Area Network. Disponível em: <<https://lora-alliance.org/about-lorawan/>>. Acesso em: 12 Set. 2024.

Waheb A. Jabbar, Development of LoRaWAN-based IoT system for water quality monitoring in rural areas, Expert Systems with Applications, Volume 242, 2024. Disponível em: <<https://doi.org/10.1016/j.eswa.2023.122862>>. Acesso em: 12 Set. 2024.

MYSQL. Creating EER Diagrams. Disponível em: <<https://dev.mysql.com/doc/workbench/en/wb-creating-eer-diagram.html>>. Acesso em: 12 set. 2024.

FIREBASE. Firestore Data Model. Disponível em: <<https://firebase.google.com/docs/firestore/data-model?hl=pt-br>>. Acesso em: 12 set. 2024.

WANG, Yusi. Review and testing of plugins in Flutter for Android and IOS. 2022. Dissertação (Mestrado) - Politecnico di Torino, Torino, 2022. Disponível em: <<https://webthesis.biblio.polito.it/secure/22858/1/tesi.pdf>>. Acesso em: 03 Out. 2024

Japara, E. M., Arifin, S., & Irwansyah, E. (2023). Geolocation System Module Creation to Validate User Location Coordinates in an Android Application Using Flutter Framework. Journal of Computer Science, 19(8), 1050-1064. Disponível em: <<https://doi.org/10.3844/jcssp.2023.1050.1064>>. Acesso em: 03 Out. 2024