

PROTÓTIPO DE SISTEMA DE NAVEGAÇÃO INDOOR ADAPTATIVO: Um Estudo de Caso em Uma Universidade do Sul de Santa Catarina

Gabriel Correa Forcellini ¹
Matheus Leandro Ferreira ²

Resumo: Sistemas de navegação indoor são ferramentas essenciais em ambientes fechados, onde a atenuação do sinal inviabiliza o uso de GPS. Esses sistemas são particularmente úteis em espaços complexos, como universidades e centros comerciais, auxiliando na orientação e no direcionamento de rotas. Este trabalho apresenta o desenvolvimento de um protótipo adaptativo para navegação interna, projetado para o campus de uma universidade no Sul de Santa Catarina. A solução combina o algoritmo de Dijkstra, para otimização de rotas, com a conectividade de pontos de acesso Wi-Fi, que estimam a posição aproximada do usuário. A aplicação foi implementada utilizando o framework Flutter, com interface visual e um grafo que modela os prédios e conexões do campus, representados por vértices e arestas. O sistema mostrou-se eficaz na orientação de trajetos e no cálculo das rotas mais curtas, embora o uso do Wi-Fi tenha apresentado limitações devido à cobertura do sinal. A navegação intuitiva e a visualização clara proporcionaram uma experiência satisfatória para os estudantes. Como perspectivas futuras, sugere-se a integração de iBeacons para maior precisão no posicionamento, bem como a expansão do mapa para abranger todas as áreas do campus, aprimorando a usabilidade do sistema.

Palavras-chave: Navegação indoor; Algoritmo de Dijkstra; Localização;

¹gabriel.forcellini@unesc.net

²mlf@unesc.net

ABSTRACT: Indoor navigation systems are essential tools in enclosed environments where signal attenuation prevents the use of GPS. These systems are particularly useful in complex spaces such as universities and shopping centers, assisting in orientation and route guidance. This paper presents the development of an adaptive prototype for indoor navigation, designed for the campus of a university in Southern Santa Catarina. The solution combines Dijkstra's algorithm, for route optimization, with the connectivity of Wi-Fi access points, which estimate the user's approximate position. The application was implemented using the Flutter framework, with a visual interface and a graph that models the campus buildings and connections, represented by vertices and edges. The system proved effective in guiding paths and calculating the shortest routes, although the use of Wi-Fi presented limitations due to signal coverage. The intuitive navigation and clear visualization provided a satisfactory experience for students. Future prospects include the integration of iBeacons for greater positioning accuracy, as well as expanding the map to cover all areas of the campus, improving the system's usability.

Keywords: Indoor navigation; Dijkstra's algorithm; Location;

1 INTRODUÇÃO

O ato de transitar em ambientes indoor é uma atividade comum que faz parte do nosso cotidiano, porém navegar por ambientes desconhecidos é uma ação cognitiva complexa e desafiadora, a qual exige do pedestre conhecimento a respeito de sua localização, orientação correta ao se deslocar para um destino e que não se perca durante o trajeto. Os sistemas de navegação são caracterizados como ferramentas que auxiliam nas tarefas de navegação, reduzindo o esforço realizado pelo usuário (Antunes et al., 2022).

Atualmente a navegação em ambientes externos tornou-se uma prática comum, impulsionada pela ampla utilização de sistemas globais de navegação e o auxílio de Satélites (Schwertner, 2017).

A maioria dos dispositivos móveis atualmente incorpora o sistema de posicionamento global (GPS), o que tem facilitado o uso de serviços baseados em localização em todo o mundo. No entanto, quando se trata de ambientes internos, como por exemplo, o campus de uma Universidade, o uso de GPS enfrenta desafios significativos devido à atenuação do sinal causada pela estrutura das edificações. Além disso, os serviços baseados em telefonia celular, que são comuns para navegação em ambientes

externos, também não oferecem a precisão necessária para ambientes internos (Aravena; Delazari, 2021).

Um sistema de navegação ideal se caracteriza como um sistema adaptativo, que reconhece e se adequa aos ambientes que são altamente dinâmicos, adaptando-se e gerando instruções de navegação conforme o usuário transita pelo ambiente interno (Antunes et al., 2022).

Desta forma, o objetivo primordial desta pesquisa consistiu em facilitar o processo de localização e orientação dos estudantes de uma Universidade do Sul Catarinense. Para alcançar este objetivo, foi desenvolvido um protótipo de aplicativo móvel, projetado especificamente para auxiliar os estudantes na navegação pelo campus universitário. Este protótipo visou proporcionar aos usuários uma experiência eficiente e precisa ao encontrar seus destinos, oferecendo rotas claramente definidas e orientações detalhadas.

Além de facilitar o acesso por meio do sistema de navegação indoor, este estudo promete contribuir significativamente para a área da ciência da computação e para todos aqueles interessados em soluções de localização e orientação em ambientes internos. Ao abordar os desafios específicos relacionados à atenuação do sinal GPS e à imprecisão dos serviços baseados em telefonia celular em espaços fechados. Os resultados e as lições aprendidas podem servir como base para futuras pesquisas e inovações no campo da tecnologia de navegação indoor, beneficiando não apenas acadêmicos, mas também empresas, instituições e indivíduos que buscam soluções avançadas de localização e orientação.

2 TRABALHOS CORRELATOS

Para o desenvolvimento deste estudo, realizou-se uma análise de trabalhos semelhantes sobre o tema proposto, utilizando outros projetos como base para um embasamento mais sólido.

2.1 DESIGN OF A MOBILE AUGMENTED REALITY-BASED INDOOR NAVIGATION SYSTEM

O artigo desenvolvido por Xin Hui Ng e Woan Ning Lim para a Escola de Ciência e Tecnologia da Universidade Sunway na Malásia, em 2020, descreve o desenvolvimento de um aplicativo móvel destinado a facilitar a orientação das pessoas pelo campus da Universidade Sunway. Utilizando tecnologias para dispositivos Android, o aplicativo incorpora uma combinação de diferentes tipos de sensores com tecnologia de Realidade

Aumentada. O objetivo foi criar uma solução de Navegação Indoor que seja acessível e de fácil utilização, oferecendo alta precisão na localização do usuário e assistência contínua enquanto ele se desloca pelo ambiente. A avaliação do aplicativo mostrou alta precisão devido à integração dos sensores, e o feedback dos usuários confirmou sua utilidade. Contudo, a funcionalidade de navegação entre diferentes andares não foi implementada, o que limitou o acesso a certas áreas do campus.

2.2 DESENVOLVIMENTO DE APLICATIVO PARA AUXÍLIO À NAVEGAÇÃO EM AMBIENTES INTERNOS

O artigo de 2021, elaborado por Carolina Aguiar Aravena e Luciene Stamato Delazari para a Revista Brasileira de Cartografia, explora uma alternativa de posicionamento em ambientes internos, onde o sinal de GPS é insuficiente. A pesquisa discute o uso de WIFI e a representação de mapas em aplicativos móveis, além de comparar métodos de posicionamento como Proximidade, Centróide e Trilateração. O método de Proximidade atribui a posição do objeto ao transmissor mais próximo, o Centróide calcula a média das posições de vários transmissores, e a Trilateração determina a posição medindo a distância até múltiplos pontos de referência, exigindo três pontos para duas dimensões e quatro para três dimensões. Para criar o mapa base da localidade estudada, utilizou-se o Mapbox, um sistema que suporta a adição de mapas em diversas plataformas como web, dispositivos móveis, sistemas de navegação veicular e realidade virtual. A integração da base cartográfica do campus foi realizada através do Mapbox Studio, que permite a adição de camadas em diferentes formatos (shapefile, JSON e CSV) aos mapas. Os resultados indicaram que a técnica de Trilateração apresentou a maior precisão posicional, com a menor média de discrepâncias planimétricas e desvio padrão, comparada às demais técnicas. Nota-se que este estudo não incluiu testes com usuários.

2.3 FERRAMENTA PARA LOCALIZAÇÃO EM AMBIENTE CONTROLADO UTILIZANDO MAPA DE SINAIS WIRELESS

Desenvolvido por Luiz Ricardo Fiera na Universidade do Extremo Sul Catarinense (UNESC), no ano de 2013, este artigo apresenta o desenvolvimento de uma ferramenta de acessibilidade para auxiliar deficientes visuais a se localizarem em ambientes controlados. A ferramenta emprega um mapa de sinais Wireless e modelos matemáticos, incluindo teoria de grafos e inteligência computacional. O processo de levantamento de requi-

sitos foi realizado através de entrevistas com o público-alvo, os deficientes visuais, para compreender as necessidades específicas e otimizar a assistência ao usuário final. A orientação no ambiente é facilitada pelo uso do algoritmo de Dijkstra, que transforma o ambiente físico em uma estrutura de grafo, representando cada localização como um vértice. Access Points foram instalados para garantir a cobertura de Wi-Fi necessária para o funcionamento da ferramenta. O desenvolvimento da aplicação móvel foi realizado em Java, permitindo compatibilidade com a plataforma Android. Utilizou-se o banco de dados SQLite, um sintetizador de voz nativo do Android e o pacote Svox Luciana, que possibilita a leitura de texto em português. Os testes realizados incluíram uma avaliação de usabilidade da interface, que confirmou a eficácia dos recursos de acessibilidade e a facilidade de uso do protótipo. A interação da aplicação com as antenas de rede também mostrou resultados conforme o esperado. A validação final revelou que a ferramenta foi eficaz em orientar os usuários em 60% dos testes, embora a inconstância do sinal Wi-Fi tenha causado algumas falhas.

3 MATERIAIS E MÉTODOS

Esta pesquisa caracteriza-se por desenvolver uma solução tecnológica aplicada, onde a finalidade é construir um aplicativo móvel que permita que os usuários encontrem rotas otimizadas entre os diferentes blocos do câmpus de uma Universidade do Sul Catarinense.

3.1 Interface

A aplicação móvel foi desenvolvida utilizando Flutter. O Flutter foi desenvolvido e mantido pela Google, além de ser amplamente utilizada dentro do Google e diferente de outras plataformas móveis populares não usa Java, mas sim Dart como linguagem de programação. Este framework utiliza uma abordagem declarativa para construir sua interface do usuário com base em widgets (chamados de "componentes" no mundo do web) que são renderizados apenas quando necessário (Arb; Al-Majdi, 2020).

3.1.1 Estrutura do Projeto

O código foi organizado em diversas **pastas** e **arquivos**, conforme descrito a seguir:

- `main.dart`: Arquivo principal do projeto onde é definido a tela inicial e o tema e estilo global aplicado a toda a aplicação.
- **components**: Pasta do projeto que contém três arquivos fundamen-

tais, cada um deles implementando componentes reutilizáveis que desempenham papéis essenciais na aplicação.

- `card.dart`: Este componente é responsável pela exibição das informações relacionadas a um bloco específico, incluindo o avatar (representado pela sigla do bloco) e o nome. Através desse componente, os usuários podem visualizar rapidamente os dados pertinentes ao bloco em questão, facilitando a interação e compreensão das informações.
 - `location_indicatorPainter.dart`: Este arquivo contém a lógica para pintar visualmente a posição do usuário no mapa. Ele é um componente gráfico que fornece uma representação clara e precisa da localização atual do usuário, contribuindo para uma melhor experiência de navegação dentro da aplicação.
 - `routePainter.dart`: Similar ao `location_indicatorPainter.dart`, este componente é responsável por renderizar as rotas calculadas no mapa. Ele permite que os usuários visualizem o trajeto planejado, facilitando a orientação e a navegação.
- **mocks**: Abriga um único arquivo.
 - `blocos.dart`: Contém uma lista de blocos disponíveis no mapa da aplicação. Esses blocos representam todos os blocos suportados pelo protótipo. O arquivo `blocos.dart` serve como uma fonte simulada de dados, permitindo que os desenvolvedores testem e visualizem a funcionalidade da aplicação sem depender de um banco de dados real. Essa abordagem é fundamental para o desenvolvimento ágil, pois possibilita a realização de testes e a verificação do comportamento da interface do usuário em diferentes cenários.
 - **models**: Pasta do projeto que contém três arquivos principais: `bloco.dart`, `wifi.dart` e `grafo.dart`. Esses arquivos definem as classes essenciais que representam as entidades centrais da aplicação.
 - `bloco.dart`: Este arquivo contém a definição da classe `Bloco`, que encapsula as propriedades e comportamentos associados aos blocos disponíveis no mapa. A classe é responsável por armazenar informações relevantes, como a localização, o nome e

outras características dos blocos, permitindo uma gestão eficiente desses elementos na aplicação.

- `wifi.dart`: Este arquivo define a classe `Wifi`, que representa as informações relacionadas às conexões Wi-Fi disponíveis. A classe pode incluir propriedades como nome da rede, intensidade do sinal e outros atributos pertinentes, possibilitando a integração e utilização eficiente das funcionalidades de conectividade na aplicação.
 - `grafo.dart`: Neste arquivo, encontra-se a definição da classe `Grafo`, que modela as interconexões entre os blocos. A classe `Grafo` é crucial para a implementação da lógica de navegação, permitindo que a aplicação calcule rotas e relações entre os diferentes blocos disponíveis no mapa.
- **pages**: Pasta que contém três arquivos principais: `choose_where_go_to_route_page.dart`, `home_page.dart` e `wifi_info_page.dart`. Esses arquivos são responsáveis por abrigar os componentes das telas da aplicação.
 - `choose_where_go_to_route_page.dart`: Esta página fornece a interface onde o usuário pode selecionar o bloco de início e o bloco de destino para sua rota. A tela foi projetada para facilitar a navegação, permitindo que os usuários definam suas preferências de trajeto de maneira intuitiva.
 - `home_page.dart`: Esta é a tela principal que é exibida ao abrir o aplicativo. Ela serve como o ponto de partida para a interação do usuário com as funcionalidades do aplicativo, apresentando uma visão geral das opções disponíveis.
 - `wifi_info_page.dart`: Esta página exibe informações sobre a rede Wi-Fi à qual o usuário está conectado no momento. Embora a funcionalidade desta página seja útil, ela está atualmente servindo principalmente como uma ferramenta de apoio durante o desenvolvimento, permitindo que os desenvolvedores testem a conectividade e a integração com as informações da rede.
- **ui**: Pasta do projeto que contém três arquivos principais: `colors.dart`, `theme_provider.dart` e `theme.dart`. Esses arquivos são responsáveis por definir o tema e as cores da aplicação, assegurando uma experiência visual coesa e agradável para os usuários.

- **utils:** Pasta do projeto que contém funções essenciais para o funcionamento do protótipo. Essas funções desempenham papéis cruciais na operação da aplicação e incluem:
 - `monta_grafo.dart`: Esta função é responsável pela construção do grafo que representa a estrutura de interconexões entre os blocos. Ela define os pesos de cada aresta, permitindo que a aplicação calcule rotas e encontre o caminho mais eficiente entre os blocos.
 - `permissions.dart`: Esta função gerencia a solicitação de permissões de acesso à localização do usuário. O acesso à localização é fundamental para determinar em qual rede Wi-Fi o usuário está conectado, o que, por sua vez, é essencial para posicioná-lo corretamente no mapa.
 - `wifi_info.dart`: Esta função coleta dados sobre a rede Wi-Fi à qual o usuário está atualmente conectado. Essas informações são utilizadas para melhorar a experiência do usuário, permitindo uma navegação mais informada e precisa dentro da aplicação.

3.1.2 Funcionalidades

A tela principal do protótipo apresenta uma imagem de parte da planta baixa do campus, com um botão centralizado na parte inferior 1.

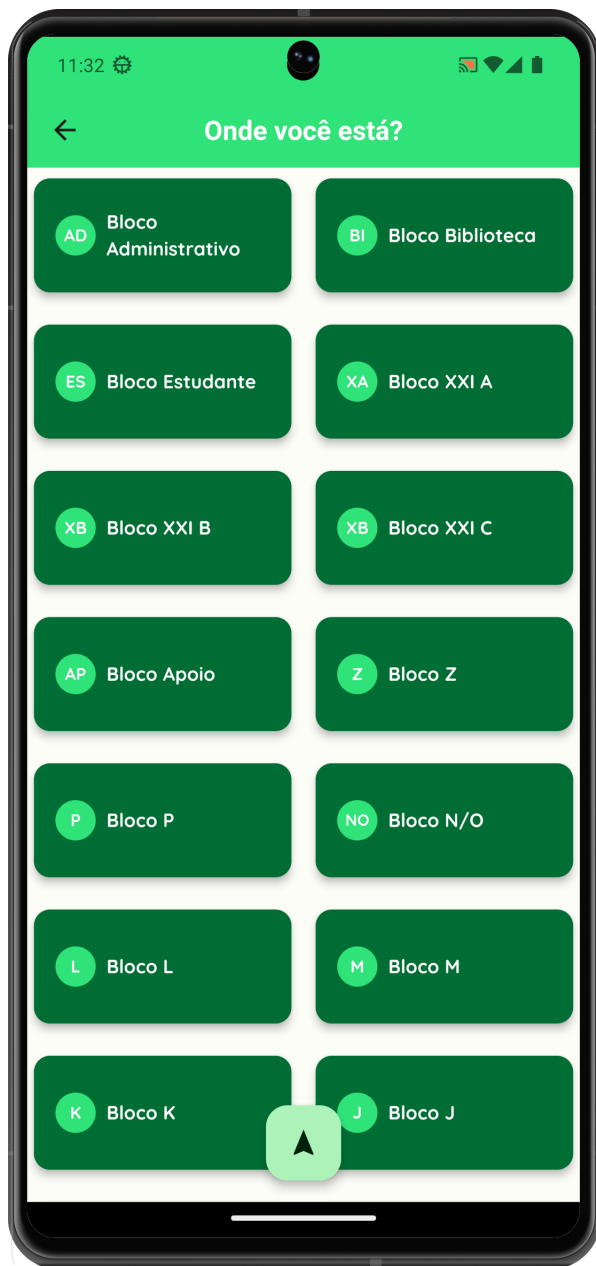
Figura 1: Tela principal do protótipo



Fonte: (Gabriel, 2024)

Para definir as rotas de origem e destino, a aplicação dispõe de uma segunda tela que lista todos os blocos suportados pelo protótipo 2.

Figura 2: Tela de escolha de rota



Fonte: (Gabriel, 2024)

3.2 Rotas

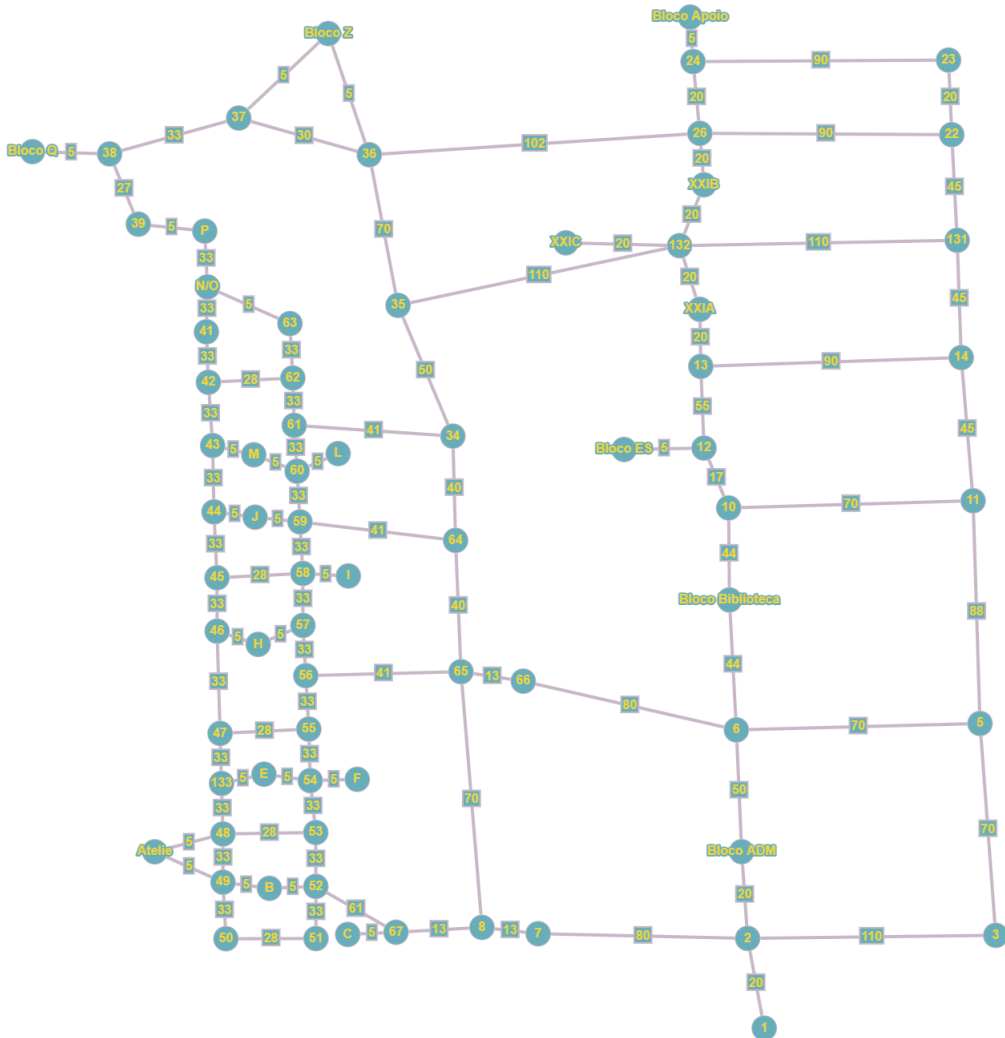
O problema de caminho mínimo ou caminho mais curto consiste em encontrar o melhor caminho entre dois nós, seja em termos de custo mínimo ou menor tempo de viagem. Este desafio é central em diversas aplicações práticas, desde roteamento de redes até sistemas de navegação em tempo real. Em uma rede, podem existir múltiplos caminhos entre um ponto de origem e um destino. O conceito de "peso" refere-se à soma

total dos valores associados aos arcos que compõem o caminho, que podem incluir o tempo total da viagem, a distância percorrida, ou qualquer outro custo associado. O caminho com o menor peso é identificado como o caminho mínimo (Wilhelm,).

Desenvolvido em 1959 por Edsger Wybe Dijkstra, o algoritmo de Dijkstra se aplica em grafos simples para determinar o caminho mínimo entre dois vértices e é desenvolvido em uma árvore de caminhos mínimos. Qualquer situação que possa ser representada por um grafo simples, que possua sua matriz de peso definida é passível de ser submetido a proposta de Dijkstra (Barros; Pamboukian; Zamboni, 2007).

Com o objetivo de determinar as rotas mais eficientes entre os diversos locais do campus, foi empregado o algoritmo acima citado. Este algoritmo permitiu modelar o campus como um grafo, onde cada bloco representa um vértice e as conexões entre os blocos correspondem às arestas. As decisões de rota possíveis foram mapeadas nos vértices, enquanto as distâncias ou custos associados a cada caminho foram atribuídos às arestas, conforme ilustrado na figura 3.

Figura 3: Grafo do mapa



Fonte: (Gabriel, 2024)

As arestas, que correspondem aos blocos ou conexões, foram definidas a partir de cada decisão de percurso. Sempre que havia alguma possibilidade distinta de escolha de caminho, uma nova conexão foi criada, e ao alcançar um bloco específico, também foi gerada uma nova aresta. Os vértices, por sua vez, representam os trajetos entre as conexões e blocos, e seus respectivos pesos foram calculados em unidades de passos.

3.3 Posicionamento

Os pontos de acesso à rede, conhecidos como APs, funcionam emitindo sinais de Wi-Fi que podem ser detectados por dispositivos móveis situados dentro de sua área de cobertura. Esses sinais são propagados com intensidade variável, o que permite que dispositivos como smartpho-

nes e tablets identifiquem a presença de um ponto de acesso dentro de uma distância limitada, desde que estejam dentro do alcance do sinal (Simões, 2015).

Neste protótipo, foram utilizados os pontos de acesso distribuídos ao longo do campus para auxiliar no mapeamento da posição dos usuários. Para isso, foi realizado um mapeamento detalhado dos pontos de Wi-Fi em cada bloco e nas áreas de transição entre eles. Assim, à medida que o usuário se desloca pelo campus e passa de um AP para outro, o sistema detecta essa mudança e atualiza a posição do usuário no mapa. Essa abordagem fornece ao usuário uma percepção aproximada de onde ele está localizado dentro do campus, tornando a navegação mais intuitiva. Pode-se observar na figura 4 um exemplo.

Figura 4: Posicionamento



Fonte: (Gabriel, 2024)

4 RESULTADO E DISCUSSÕES

Foi proposto neste trabalho a criação de um protótipo de sistema de navegação indoor, no qual compreende as áreas de estudo tecnológicos relacionados a ciência da computação.

O protótipo deste trabalho e o sistema de Aravena e Delazari (2021) diferem em complexidade e abordagem. Enquanto o protótipo utiliza Wi-Fi apenas para identificar o ponto de acesso mais próximo e calcula

rotas com o algoritmo de Dijkstra em um grafo simples, o sistema de (Ara-vena; Delazari, 2021) destaca-se pelo uso de técnicas avançadas de posicionamento e mapeamento. A trilateração determina a posição do usuário medindo a distância em relação a pelo menos três pontos de referência conhecidos, utilizando a intensidade do sinal Wi-Fi (RSSI). Essa técnica calcula a interseção das áreas de cobertura dos transmissores, obtendo maior precisão em ambientes controlados. Já o centróide ponderado utiliza a média das posições dos transmissores, aplicando pesos baseados na intensidade do sinal para priorizar transmissores mais próximos. Essa abordagem é eficiente para calcular posições em tempo real, embora dependa da densidade dos pontos de acesso para manter a precisão. O uso do Mapbox possibilitou a criação de uma base cartográfica detalhada, com suporte a camadas personalizadas e integração em dispositivos móveis. Essa ferramenta foi crucial para representar graficamente o campus e visualizar as posições calculadas, enriquecendo a experiência do usuário com mapas interativos.

Enquanto essas técnicas fornecem maior precisão e detalhamento, o protótipo deste trabalho simplifica o processo ao focar em uma solução mais prática e acessível, adequada ao seu contexto.

Ao comparar o protótipo desenvolvido neste trabalho com o sistema apresentado por (Fiera, 2013), observam-se diferenças significativas em abordagem e propósito, destacando as inovações do presente estudo.

Enquanto o sistema de Fiera foi projetado principalmente para deficientes visuais, empregando mapas de sinais Wi-Fi e o algoritmo de Dijkstra para orientação em ambientes controlados, o protótipo aqui desenvolvido destina-se a um público mais amplo, com foco em estudantes universitários. A interface visual baseada em Flutter, junto à funcionalidade de desenho de rotas dinâmicas no mapa do campus, adapta-se a usuários sem limitações visuais, proporcionando uma experiência mais intuitiva e interativa.

Além disso, o trabalho atual simplifica o processo de mapeamento e localização. Embora Fiera tenha empregado técnicas de mapeamento detalhado de RSSI (Received Signal Strength Indication) com coleta extensa de dados, este estudo utilizou uma abordagem mais direta para mapear o campus como um grafo, reduzindo a complexidade técnica e o tempo de implementação. A precisão do Wi-Fi, mesmo limitada, foi considerada suficiente para o propósito de navegação básica no campus. O presente trabalho teve como referência esta ferramenta, passando a ter ca-

racterísticas semelhantes, mas com algumas particularidades:

Interface visual do mapa do campus da Universidade do Sul Catarinense: Foi desenvolvido uma interface visual para que atenda a pessoas do campus que não são deficientes visuais, tendo um propósito de ser intuitivo e de facilitar a visualização das rotas. Para isso está inserido um mapa do campus ao centro da tela principal, onde o usuário tem noção de onde está e de onde quer chegar.

Utilização de Grafo para determinar as possibilidades de rotas entre os destinos: Neste estudo de caso, todas as possíveis rotas do campus foram mapeadas em um grafo. Dessa forma, foi possível aplicar o algoritmo de Dijkstra para encontrar o caminho mais curto entre os diversos pontos do mapa. O teste inicial foi realizado do bloco XXIA ao bloco da Administração, onde a rota calculada pelo algoritmo resultou em um tempo de aproximadamente um minuto e trinta segundos, enquanto uma rota alternativa apresentava um tempo de três minutos e vinte segundos. Esse resultado demonstrou a eficiência do algoritmo.

Para ampliar a análise, foram realizados quatro testes no total, sem um planejamento experimental definido, sendo cada teste replicado manualmente. Esses testes compararam o tempo estimado pela rota mais curta com uma rota alternativa para diferentes trajetos no campus. A tabela abaixo apresenta os resultados, evidenciando a economia de tempo obtida com o uso do algoritmo de Dijkstra:

Tabela 4 – Validações dos testes do aplicativo

Teste	Bloco de Partida	Bloco de Destino	Tempo de Rota	Tempo de Rota Alternativa	Diferença de Tempo
1	XXIA	Administração	1 min 30 seg	3 min 20 seg	1 min 50 seg
2	Biblioteca	Bloco C	2 min 14 seg	2 min 32 seg	18 seg
3	Bloco B	Bloco G	49 seg	1 min 4 seg	15 seg
4	XXIB	Bloco Estudantes	56 seg	2 min	1 min 4 seg

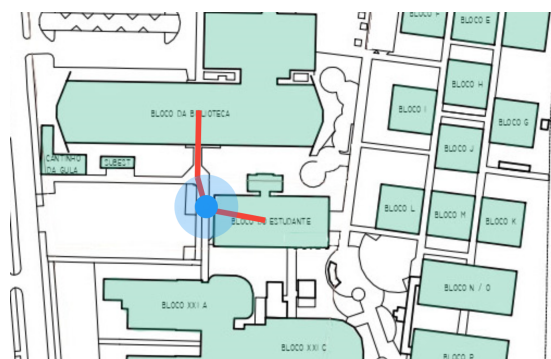
Fonte: (Gabriel, 2024)

Esses resultados confirmam a eficácia do algoritmo de Dijkstra na determinação da rota mais eficiente, proporcionando aos usuários um ganho significativo de tempo ao navegar pelo campus.

Desenhando as rotas e posicionamento: Para criar os desenhos das rotas, foi utilizada a classe CustomPainter do Flutter, onde para cada aresta do grafo foi mapeado o desenho de uma reta, conectando uma a outra, formando a rota. Já para o desenho da posicionamento foi utilizada

a mesma classe, porém agora um ponto no mapa. Esta abordagem permitiu que as rotas e o posicionamento fossem desenhados dinamicamente e em tempo real, após a escolha do usuário e se mostrou eficiente. Figura

Figura 5: Rotas e Posicionamento



Fonte: (Gabriel, 2024)

Utilização do Wi-fi para localização do usuário: A utilização do Wi-Fi como posicionamento do usuário no mapa não se mostrou eficiente, isso devido ao alcance do sinal Wi-Fi, que pode chegar a uma distância de até 100 metros conforme citado por (Simões, 2015). Embora essa tecnologia ofereça uma orientação básica, indicando a proximidade geral de um usuário em relação a um ponto de acesso, ela não consegue fornecer um posicionamento exato, ainda mais quando está passando de um ponto para outro. Em resumo, o sistema fornece uma visão ampla da localização do usuário, permitindo que ele tenha uma noção aproximada de onde se encontra para facilitar a sua orientação no campus.

5 CONCLUSÃO

Este trabalho teve como objetivo desenvolver um protótipo de sistema de navegação indoor para auxiliar os usuários a encontrarem seus destinos dentro de um campus universitário, com foco na eficiência e acessibilidade. O algoritmo de Dijkstra foi essencial para calcular rotas otimizadas entre diferentes pontos, garantindo trajetos eficientes. A interface intuitiva, desenvolvida em Flutter, proporcionou uma visualização clara das rotas, destacando a versatilidade dessa tecnologia no desenvolvimento de aplicações móveis.

Os testes, conduzidos pelo autor, compararam rotas sugeridas pelo sistema e rotas alternativas, medindo manualmente o tempo de deslocamento. Os resultados confirmaram a eficiência do protótipo em reduzir o

tempo de trajeto, validando o uso do algoritmo de Dijkstra como uma solução robusta para navegação em ambientes fechados.

Além da economia de tempo para os usuários, este protótipo representa uma contribuição relevante para o campo de navegação indoor, onde tecnologias como GPS enfrentam limitações. Para futuros desenvolvimentos, a integração de iBeacons pode aprimorar a precisão do posicionamento, enquanto a expansão do mapa e a adaptação para outros ambientes complexos, como hospitais e aeroportos, podem ampliar seu alcance e impacto.

REFERÊNCIAS

- ANTUNES, A. P. et al. **Sistemas de Navegação Indoor Não Adaptativos: Conceitos e Desafios para sua Implementação**. [s.n.], 2022. v. 74. 819–838 p. Disponível em: <<https://seer.ufu.br/index.php/revistabrasileiracartografia/article/view/60875>>.
- ARAVENA, C. A.; DELAZARI, L. S. **Desenvolvimento de aplicativo para auxílio à navegação em ambientes internos**. [s.n.], 2021. v. 73. 530–541 p. Disponível em: <<https://seer.ufu.br/index.php/revistabrasileiracartografia/article/view/50102>>.
- ARB, G. I.; AL-MAJDI, K. **A Freights Status Management System Based on Dart and Flutter Programming Language**. [S.l.]: Institute of Physics Publishing, 2020. v. 1530. ISSN 17426596.
- BARROS, E. A. R.; PAMBOUKIAN, S. V. D.; ZAMBONI, L. C. **Algoritmo de Dijkstra: Apoio didático e multidisciplinar na implementação, simulação e utilização computacional**. [S.l.: s.n.], 2007.
- FIERA, L. R. **FERRAMENTA PARA LOCALIZAÇÃO EM AMBIENTE CONTROLADO UTILIZANDO MAPA DE SINAIS WIRELES**. [S.l.: s.n.], 2013.
- GABRIEL, F. [S.l.: s.n.], 2024.
- SCHWERTNER, F. **Aplicativo para navegação indoor utilizando a câmera como sensor de posicionamento**. [S.l.: s.n.], 2017.
- SIMÕES, D. M. **Navegação indoor baseada na rede WiFi como suporte a serviços baseados na localização: estudo de caso no Campus da UL**. [S.l.: s.n.], 2015. 59 p.
- WILHELM, V. E. **Problema do Caminho Mínimo**. [S.l.: s.n.].