

NAVEGAÇÃO INDOOR PARA DEFICIENTES VISUAIS: OTIMIZAÇÃO DA DISTRIBUIÇÃO DE BEACONS E BUSCA PELO CAMINHO MÍNIMO

2022004574 - Daniel Enrique Gonzalez de Aguiar

2022004823 - Huandy Calini de Camargo Silva

2022004242 - João Lucas Moraes de Oliveira

2022002542 - Letícia Vitória dos Santos

SMAC03 - GRAFOS

Prof. Rafael Frinhani



INSTITUTO DE
MATEMÁTICA E
COMPUTAÇÃO

UNIFEI - Itajubá



Navegação Indoor para Deficientes Visuais: Otimização da Distribuição de Beacons e Busca pelo Caminho Mínimo

1 Introdução

Ao longo de suas vidas, indivíduos com deficiência visual desenvolvem táticas de locomoção e mobilidade próprias, de modo a complementar a utilização de ferramentas amplamente conhecidas, a saber, bengala branca e/ou cão guia. O desenvolvimento dessas estratégias adicionais é consequência da necessidade de conquistar maior independência e enfrentar situações cotidianas efetivamente. A título de exemplificação tem-se a elaboração de mapas mentais de ambientes frequentados com regularidade. Estes mapas envolvem a memorização espacial desses locais, sendo construídos e utilizados para que o indivíduo seja capaz de identificar obstáculos físicos e direções previamente explorados e locomover-se com confiança (Melo (1991) apud Oliveira et al. (2017)). Todavia, a eficácia desse tipo de tática requer experiências prévias relacionadas ao ambiente em questão, isto é, o indivíduo se torna impossibilitado de recorrer ao auxílio destas em caso de ambientes desconhecidos ou pouco familiares.

Diante disso, uma série de desafios se manifestam quanto a locomoção de deficientes visuais dentro de ambientes fechados. Uma das dificuldades mais significativas decorre da existência de barreiras de acessibilidade em grande parte desses espaços. Barreiras são definidas como qualquer obstáculo que limite a participação social e a autonomia, conforme a [Lei nº13.146 \(2015\)](#). Assim, percebe-se que devida a ausência de indicadores não visuais quanto a corredores, elevadores, mudança de direção e outros pontos estratégicos cerceiam a autonomia de pessoas com deficiência visual ao restringir o acesso a informações necessárias para fins de orientação e localização.

Nessa perspectiva, considerando a já mencionada ausência dos indicadores de sinalização, bem como a impossibilidade da utilização de táticas como mapas mentais de um local ao frequentá-lo pela primeira vez, é importante destacar a utilidade da navegação indoor, a qual desempenha um papel fundamental na orientação de pessoas com deficiência visual em ambientes internos e fechados. Além disso, é válido mencionar que essa tecnologia de localização, que permite determinar a posição de pessoas, objetos ou dispositivos em estruturas nas quais o GPS não funciona bem, tem crescido rapidamente. Isso ocorre porque, desde 2019, apresentou um crescimento médio anual de 38,2%, e a previsão é que o mercado alcance um valor de 28,2 bilhões de dólares até 2024 ([Goldstein Market Intelligence, 2019](#)).

Sendo assim, a navegação indoor já possui diversas aplicações práticas, como auxiliar pessoas em espaços amplos, como aeroportos, fábricas, museus, shoppings, edifícios go-

vernamentais e hospitais, fornecendo informações sobre localização e direções. Para alcançar seus objetivos, a navegação indoor faz uso de várias tecnologias, incluindo Bluetooth, WiFi, sensores inerciais, sensores de imagem, ultrassom, mapas e sistemas de informação geográfica, aplicativos móveis e softwares especializados, entre outros.

1.1 Cenário de Estudo

A presente pesquisa será conduzida e testada nas instalações do Instituto de Matemática e Computação (IMC), pertencente a Universidade Federal de Itajubá (UNIFEI). Este edifício está localizado no campus de Itajubá, Minas Gerais, sendo primordialmente designado para a condução de atividades de ensino. No que se refere à sua composição e estrutura, vale destacar que esta construção constitui-se de dois andares.

Nessa perspectiva, vale citar que o andar térreo é primordialmente dedicado a blocos de salas de aula, bem como laboratórios didáticos e de pesquisa. O andar superior, por sua vez é dividido em dois blocos, sendo um deles composto por laboratórios didáticos e o outro reservado para os docentes vinculados a instituição, oferecendo um gabinete individual designado a cada um deles. Ademais, é importante mencionar que o patamar superior pode ser alcançado por intermédio de três diferentes possibilidades, sendo elas escadas, elevador e uma rampa de fácil acesso.

Além disso, é relevante mencionar que este prédio pode ser adentrado por meio de três entradas diferentes, as quais serão consideradas ao longo desta pesquisa.

1.2 Detalhamento do Problema

Embora a navegação indoor represente uma excelente solução para a locomoção de pessoas com deficiência visual em ambientes fechados, ela enfrenta desafios significativos. Para compreender melhor os desafios que serão abordados nesta pesquisa, é importante destacar que, neste estudo específico, utilizaremos dispositivos BLE (Bluetooth Low Energy), também conhecidos como Beacons, como a principal ferramenta de auxílio. Além disso, a pesquisa será conduzida e testada nas instalações do Instituto de Matemática e Computação (IMC) da Universidade Federal de Itajubá (UNIFEI), localizado no campus de Itajubá, Minas Gerais.

Frente ao exposto, primordialmente, vale citar que o dispositivo utilizado nessa pesquisa, Beacon, possui um alcance de 70 metros e um custo médio de R\$500,00 por cada dispositivo. Dessa forma, o primeiro desafio que se apresenta está relacionado à distribuição desses dispositivos no IMC. Isso se deve à necessidade de uma modelagem estratégica para

a distribuição dos Beacons, com o objetivo de utilizar a menor quantidade possível, garantindo, ao mesmo tempo, o adequado funcionamento do sistema.

Além disso, outra questão desafiadora é a busca por rotas eficientes, o que requer o desenvolvimento de um protótipo capaz de encontrar o caminho mais curto possível, ao mesmo tempo em que considera as necessidades do usuário final. Esse tipo de consideração é fundamental para viabilizar a locomoção segura de grupos distintos, tais como idosos e indivíduos com deficiência física que fazem uso de cadeira de rodas. Nesse sentido, é crucial criar um código que execute a busca pelo destino desejado, estabelecendo conexões com os Beacons, a fim de traçar a rota mais eficaz. Portanto, após a obtenção das informações do ponto de partida e do ponto de destino desejado, o sistema deve identificar a rota mais rápida e eficiente, minimizando possíveis obstáculos e proporcionando uma experiência agradável e eficaz ao usuário final.

2 Referencial Teórico

Wohlfahrt et al. (2015) descreve o desenvolvimento de uma solução para auxiliar pessoas com deficiência visual na navegação em ambientes internos. A principal motivação para esse projeto é a dificuldade que as pessoas com deficiência visual enfrentam ao tentar se deslocar em ambientes fechados, como prédios públicos. Enquanto a navegação ao ar livre pode ser auxiliada pelo GPS, a navegação interna apresenta desafios adicionais, pois os sinais de GPS não conseguem penetrar em ambientes fechados. Ademais, a solução proposta neste trabalho utiliza etiquetas RFID (Radio-Frequency Identification) de baixo custo e dispositivos móveis para ajudar na orientação de pessoas cegas ou com deficiência visual em ambientes internos com a criação de mapas digitais que identificam a localização de etiquetas RFID em um ambiente e estabelecem uma comunicação direta com dispositivos móveis. Nesse contexto, uma aplicação servidora desempenha um papel central ao armazenar e disponibilizar informações relativas a prédios ou instalações específicas.

Ossada et al. (2016) desenvolvem um método de locomoção em ambientes fechados para deficientes visuais que utilizam smartphones. A aplicação foi desenvolvida na plataforma MIT App Inventor2 e utiliza etiquetas QR CODE para sensoriamento. Durante o desenvolvimento, foi criado um dispositivo chamado HardKey para facilitar a interação do usuário com a aplicação, na qual foi projetado para ser utilizado em conjunto com o aplicativo, permitindo que o usuário interaja com o aplicativo sem precisar olhar para a tela do smartphone. A conexão entre o dispositivo e o smartphone é estabelecida através da entrada P2, e o dispositivo dispõe de três botões que podem ser programados para executar diversas funções dentro do aplicativo. Além de atender às necessidades dos deficientes visuais, os autores sugerem que a aplicação pode ser útil para qualquer pessoa que precise explorar ambientes fechados desconhecidos.

Simões et al. (2016) descreve um sistema de navegação indoor para pessoas com deficiência visual. Ele utiliza marcadores de rádio frequência e visuais, sendo os marcadores visuais são elementos físicos que, quando combinados com algoritmos de visão computacional, permitem que o sistema de navegação reconheça e interprete a localização do usuário em

ambientes internos, auxiliando a navegação de pessoas com deficiência visual., para criar um mapa do ambiente e permite que os usuários se desloquem com segurança. O sistema opera em duas fases: uma para criar o mapa e outra para a navegação em tempo real. Resultados mostraram alta precisão na detecção de marcadores e obstáculos. Os usuários aprovaram a qualidade da orientação, independência e confiabilidade do sistema, mas ainda há espaço para melhorias na redução de erros acumulados na navegação.

Martins et al. (2021) desenvolvem de um sistema para ajudar deficientes visuais a se locomoverem em ambientes fechados, como prédios, shoppings, etc. O sistema utiliza tecnologia assistiva e substituição sensorial para auxiliar na locomoção e na identificação de obstáculos. Ele também descreve o uso de aplicativos móveis e dispositivos hápticos-sonoros para criar uma solução de baixo custo e eficaz para melhorar a qualidade de vida e a autonomia de pessoas com deficiência visual. O artigo detalha os componentes técnicos e os métodos utilizados no desenvolvimento desse sistema.

Santos (2014) trata da criação de um sistema de navegação em edifícios usando redes sem fio existentes, com foco na localização precisa dentro de edifícios. Esse sistema visa solucionar a limitação dos sistemas GPS tradicionais em ambientes fechados. A abordagem proposta calcula limites mínimos de tempo para deslocamentos entre pontos dentro de um edifício, empregando algoritmos baseados em grafos. O trabalho é parte do projeto ADINLOC, que busca desenvolver um sistema de localização para uso em hospitais e residências assistidas, com a finalidade de melhorar o cuidado e segurança de idosos. O artigo enfatiza a importância da precisão na localização, visto que erros de posicionamento podem ter consequências graves.

Silvio et al. (2016) aborda a busca de caminhos mínimos em grafos, um problema comum na ciência da computação. Ele envolve encontrar o caminho mais curto entre pontos em um grafo, com vértices representando locais e arestas representando conexões entre eles. O artigo propõe uma modificação simples em um conhecido algoritmo de busca de caminhos mínimos baseado em matrizes para melhorar seu desempenho de processamento. Experimentos mostram que essa mudança torna o algoritmo mais eficiente do que a versão original. Assim sendo de grande importância, levando em conta o poder de processamento de um celular e sua economia de bateria.

Este estudo se diferencia dos demais, pois aborda a necessidade e imperiosidade de auxiliar pessoas com deficiência visual na navegação em ambientes fechados, com destaque para a tentativa de uso estratégico de Beacons (dispositivos BLE) como ferramenta principal. A singularidade deste trabalho reside na abordagem específica dos desafios relacionados à distribuição eficiente dos Beacons no ambiente, buscando minimizar custos, e na ênfase na busca por rotas eficientes para orientar os usuários até seus destinos, levando em consideração obstáculos e proporcionando uma experiência positiva. Além disso, o contexto da pesquisa no Instituto de Matemática e Computação da Universidade Federal de Itajubá acrescenta uma relevância prática e localizada ao estudo, contribuindo para a melhoria da qualidade de vida e autonomia das pessoas com deficiência visual.

3 Desenvolvimento

Nesta seção, será apresentada a modelagem da solução proposta, juntamente com as modelagens complementares necessárias para sua elaboração, visando abordar os desafios previamente identificados. A modelagem adota como base o uso de estruturas de grafos. Além disso, os métodos de solução propostos para o mapeamento dos beacons e para a busca do menor caminho serão expostos e explicados.

3.1 Modelagem

A análise detalhada da planta do Instituto de Matemática e Computação (IMC) da Universidade Federal de Itajubá (UNIFEI) foi o ponto de partida para examinar o local e desenvolver estratégias eficazes para a distribuição dos dispositivos BLE (Bluetooth Low Energy). O instituto é composto por dois pisos, sendo que os autores deste estudo tinham acesso à planta do segundo piso, enquanto o primeiro piso foi mapeado por esses.

O mapeamento do primeiro piso baseou-se em conhecimento prévio do local e nas medições obtidas do segundo piso. Visto que ambos os pisos possuem dimensões semelhantes e a maioria das salas são correspondentes, não foi necessário realizar medições *in loco*. Assim, a modelagem do primeiro piso foi elaborada com base no conhecimento dos autores, de modo a possibilitar o mapeamento de todo o IMC.

Desse modo, partindo das medições disponíveis na Figura 1 e baseando nas localizações similares a da Figura 2 pode-se desenvolver a planta do Piso 1, a qual se encontra na Figura 3.

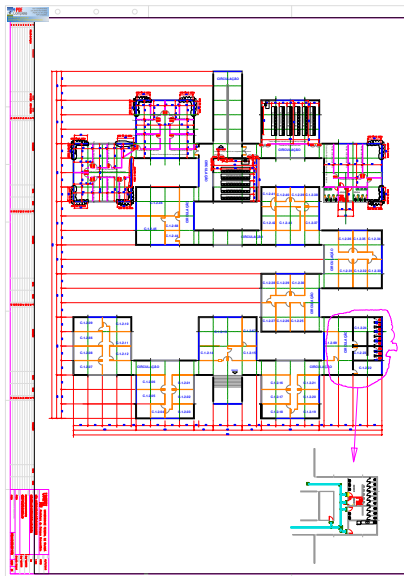


Figura 1: Planta do Segundo Piso com Dimensões.

A partir disso, essas plantas foram analisadas e debatidas entre os autores para determinar a melhor disposição dos Beacons no prédio. Como resultado dessa análise, concluiu-se que, embora o equipamento BLE utilizado tivesse um alcance de 70 metros, era necessário posicionar os Beacons mais próximos uns dos outros. Isso com o intuito de tentar criar uma rede mais robusta e confiável. Com base nessa premissa, foi decidido que os dispositivos deveriam ser distribuídos de forma a minimizar o impacto caso um deles deixasse de funcionar. No

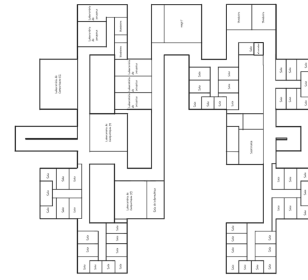


Figura 2: Planta do Segundo Piso.

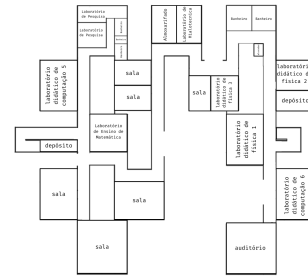


Figura 3: Planta do Primeiro Piso.

entanto, a eficácia dessa abordagem como uma vantagem precisa ser confirmada por meio de testes, a fim de assegurar com certeza que a locomoção dentro do prédio como um todo não será afetada em caso de falhas.

Assim, foi conduzido um processo de mapeamento com o objetivo de distribuir os Beacons, fazendo uso das plantas apresentadas na Figura 3 e na Figura 2. Isso resultou na criação das Figuras 4 e 5, respectivamente.

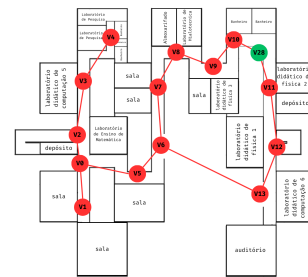


Figura 4: Mapeamento do Primeiro Piso

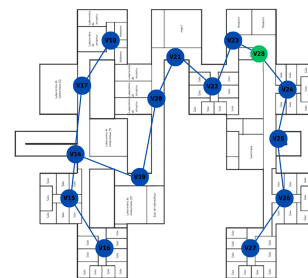


Figura 5: Mapeamento do Segundo Piso.

É relevante observar que a disposição linear das arestas

no grafo, conforme evidenciado na Figura 6, foi escolhida para facilitar a visualização. No entanto, é importante salientar que os valores de peso atribuídos a essas arestas são determinados com base na medição das distâncias em metros entre os dispositivos. Esse cálculo leva em consideração as curvas internas do prédio, resultantes de sua estrutura física.



Figura 6: Linearidade das arestas para melhor visualização.

Com base nas dimensões detalhadas na Figura 1, calculou-se as distâncias, levando em consideração as curvaturas arquitetônicas presentes no IMC, entre cada par de Beacons. Essas distâncias, em metros, foram então atribuídas às arestas dos gráficos nas Figuras 7 e 8.

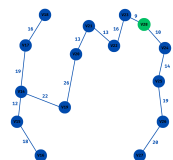
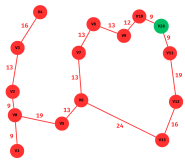


Figura 7: Grafo do Primeiro Piso Figura 8: Grafo do Segundo Piso

Por fim, com base nas plantas do instituto, é evidente que o primeiro e o segundo piso estão interligados por uma rampa, uma escada e um elevador. Portanto, tornou-se necessário mapear a aresta valorada (em metros) que conecta os vértices V2 a V14, relacionados à rampa, a aresta, também com peso em metros, que conecta os vértices V12 a V25, bem como o vértice 28 que representa o elevador, sendo um beacons (vértice) comum aos dois pisos. Assim, a modelagem final do grafo referente a todo o prédio do IMC resultou na Figura 9.

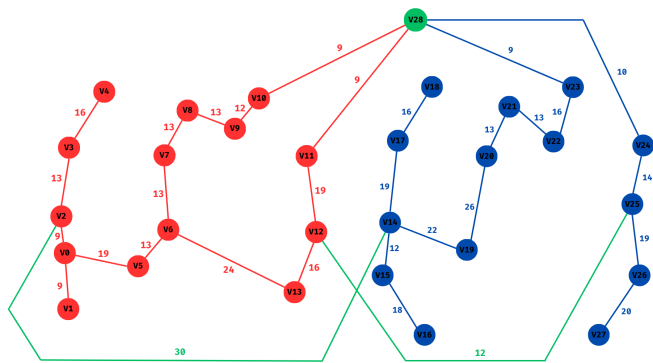


Figura 9: Grafo do Mapeamento do Instituto de Matemática e Computação (IMC).

Para estudar o comportamento de heurísticas em um grafo não direcionado com 29 vértices e 31 arestas, cujas arestas têm valores associados, é fundamental entender a representação desse grafo. Neste contexto, cada vértice representa um beacon distinto, enquanto as arestas indicam a distância em metros entre esses beacons. Cada aresta é valorada com

um número inteiro positivo, refletindo essa distância. Desse modo, tem-se um grafo $G = (V, E, f)$, que consiste em um conjunto V de vértices v_i com $|V| = m$ elementos, com $i = 0, 1, \dots, m-1$, um conjunto $E = \{v, u \mid v, u \in V, v \neq u\}$ de arestas valoradas obtidas pela associação de vértices de V , e uma função $f: E \rightarrow \mathbb{Z}^+$ que associa a cada elemento E um valor inteiro positivo.

Além disso, vale citar que o grafo é não direcionado, o que significa que a relação entre os vértices é simétrica. Isso implica que a distância entre o Beacon V0 e o Beacon V1, por exemplo, é a mesma que a distância entre o Beacon V1 e o Beacon V0, como é característico de um grafo não direcionado.

Este grafo é útil para modelar situações em que a distância entre os vértices, neste caso os beacons, desempenha um papel significativo nas decisões de busca ou otimização. Por exemplo, pode ser usado em aplicações de localização ou roteamento, onde é importante considerar as distâncias entre pontos de referência.

É importante notar que, devido às suas características específicas, como as arestas valoradas e a simetria das relações entre vértices, diferentes heurísticas e métodos de otimização podem ser aplicados a esse grafo para resolver problemas específicos. Isso permite explorar soluções eficazes para problemas que envolvem a otimização de rotas ou a tomada de decisões com base na proximidade entre beacons.

3.2 Método de Solução

Nesta seção, serão apresentadas as propostas de solução para as duas problemáticas que norteiam a pesquisa. São elas: o Mapeamento de Distribuição de Dispositivos BLE, que tem como objetivo a utilização da menor quantidade possível de beacons, e a Busca Eficiente do Menor Caminho, visando guiar de forma eficaz os indivíduos com deficiência visual pelo percurso mais curto dentro do Instituto de Matemática e Computação (IMC).

3.2.1 Proposta de Solução para Mapeamento de Distribuição de Dispositivos BLE

O método proposto para a solução da otimização de distribuição de beacons, ou seja, o grafo representado na Figura 9 contemplou um total de 29 vértices e 31 arestas. Ademais, é imperioso destacar que os vértices representam os dispositivos BLE e as arestas a distância, em metros, entre eles.

Nessa perspectiva, vale citar que, para representar a estrutura do Instituto de Matemática e Computação (IMC) da Universidade Federal de Itajubá, optou-se por empregar um grafo valorado - esse tipo de grafo é caracterizado pela atribuição de um valor, conhecido como peso, a cada uma de suas arestas (Bickle, 2020). A opção de empregar esse tipo de grafo se deu pois é amplamente reconhecida a sua capacidade intrínseca de representar de forma eficaz as distâncias entre trajetos, um atributo de suma importância para a aplicação na modelagem desenvolvida ao longo deste estudo.

Adicionalmente, deliberou-se posicionar os beacons com um alcance relativamente menor em comparação aos 70 metros suportados por tais dispositivos. Essa decisão foi tomada em virtude da preocupação com a potencial vulnerabilidade do sistema, de modo que poderia haver uma alta fragilidade do modelo se os equipamentos fossem distribuídos de modo a usar sua capacidade máxima. Ademais, foi considerada a problemática decorrente de uma possível paralisação ou mau fun-

cionamento de um dos dispositivos BLE, o que poderia acarretar inviabilidade de utilização da aplicação, caso esta fosse dependente exclusivamente de um único dispositivo. Entretanto, ao aproximar os beacons, buscou-se fortalecer a estabilidade da rede, uma vez que a aplicação pode recorrer a outros dispositivos próximos para o cálculo das rotas caso um dos equipamentos deixe de funcionar.

Outrossim, o método proposto se baseia em um grafo simples e conexo. Segundo [Chartrand \(2010\)](#), "Um grafo G é um par ordenado (V, E) , onde V é um conjunto não vazio de objetos chamados vértices e E é um conjunto de pares não ordenados de vértices chamados arestas". Já um grafo simples é definido por [West \(2001\)](#) como "um gráfico sem loops ou múltiplas arestas. Especificamos um gráfico simples por seu conjunto de vértices e conjunto de arestas, tratando o conjunto de arestas como um conjunto de pares não ordenados de vértices". Quanto à sua conectividade, pode-se afirmar que um grafo é considerado conexo, quando qualquer par de pontos é interligado por pelo menos um caminho ([Jurkiewicz, 2009](#)). Para a solução proposta, é altamente pertinente empregar um grafo com essas características específicas. Isso se justifica pela forma como os vértices se conectam através dos corredores do prédio (representados pelas arestas), pois como é possível transitar pelo corredor em ambas as direções é desnecessário que o grafo seja direcionado. Quanto à necessidade de conectividade, essa condição é crucial para assegurar que o usuário tenha a capacidade de chegar a qualquer ponto do prédio com o auxílio dos beacons, independentemente de seu ponto de partida ou destino.

Com base na análise da modelagem de proposta de solução, pode-se estabelecer que, para calcular a rota entre um beacon de origem e um beacon de destino, um requisito fundamental para a funcionalidade do aplicativo, semelhante ao funcionamento de um GPS (Sistema de Posicionamento Global), será empregado um algoritmo de caminho mínimo. Nesse contexto, um caminho mínimo é definido como uma sequência de vértices interconectados por arestas, onde cada vértice subsequente na sequência é alcançado diretamente a partir do anterior, sem repetição de vértices (Feofiloff et al., 2004). Ademais, o algoritmo de caminho mínimo em um grafo é a sequência de vértices que liga dois vértices específicos com o menor número de vértices possíveis, ou em grafos valorados com o menor custo possível, considerando os pesos das arestas. Isso permitirá ao usuário ser guiado de maneira eficaz de um ponto a outro dentro do ambiente.

Portanto, com base no método de solução proposto e na busca por um caminho mínimo, um algoritmo será desenvolvido e proposto com o fito de guiar os indivíduos de um ponto ao outro de maneira mais eficaz. Nesse algoritmo, o usuário irá informar seu vértice de origem e de destino, bem como sua preferência por passar pela rampa ou escada, em casos de necessidade de sair de um ponto no piso 1 e alcançar um ponto no piso 2, ou o contrário. De modo a garantir que seja atendido não apenas o caminho mais curto possível, mas também o caminho mais eficaz de acordo com o contexto do usuário em questão, é importante considerar que as preferências por rotas podem variar entre indivíduos em diferentes circunstância, como deficientes visuais idosos ou deficientes visuais que utilizam cadeira de rodas.

A escolha de caminhos com piso tátil e a inclusão de locais com descrição em Braille não são consideradas necessá-

rias, uma vez que os autores assumem que o prédio do IMC é acessível, já contemplando essas características. Finalmente, é importante destacar que o algoritmo proposto não inclui uma interface, uma vez que se trata de um protótipo destinado a solucionar problemas de busca por caminho mínimo. No entanto, sua utilização é altamente recomendada como um ponto de partida valioso para futuros desenvolvimentos de produtos finais mais abrangentes e funcionais.

3.2.2 Proposta de Solução para a Busca Eficiente do Menor Caminho

Para o desenvolvimento do código de busca de caminho mínimo, os autores começaram transcrevendo a modelagem do grafo para um arquivo de texto, com o objetivo de construir uma matriz de adjacências ponderada, de modo que essa matriz seria então utilizada no cálculo da rota. Como pode ser visto na Figura 10, os valores de -1 (um negativo) foram empregados para denotar a ausência de uma aresta que conecte diretamente um vértice a outro. O valor 0 (zero) foi atribuído para indicar a inexistência de um caminho do vértice para si mesmo. Por sua vez, os demais valores, inteiros positivos, representam os pesos das arestas que conectam dois vértices.

[illegible]

Figura 10: Matriz de Adjacência criada a partir do Mapeamento do Instituto de Matemática e Computação (IMC).

Após essa etapa, o processo evoluiu para o desenvolvimento do código, no qual os autores optaram pelo uso do *Python*, na versão 3.11.2. A escolha por essa linguagem de alto nível foi motivada pela sua capacidade de facilitar a compreensão do código.

Nessa perspectiva, na busca pelo menor caminho, os autores fundamentaram-se na literatura, identificando o algoritmo de Dijkstra como uma alternativa eficiente e amplamente reconhecida em pesquisas anteriores. De acordo com [Jacundá Santos et al. \(2019\)](#) "O Algoritmo de Dijkstra é capaz de determinar o caminho mínimo de um grafo ponderado partindo de um vértice de início para todos os outros vértices com pesos positivos, mostrando um ótimo desempenho em tempo computacional".

Para aprimorar a complexidade temporal do código, optou-se por empregar uma estrutura de heap. Essa escolha visa identificar o menor custo entre os pesos das arestas, sim-

plificando o processo de varredura da fila de prioridade e a localização do custo mínimo. Segundo [Gonçalves \(2015\)](#), quando o heap binário é utilizado, tem-se o custo de $O(\log_2 |V|)$ para cada operação de extração do valor mínimo da fila de prioridade, totalizando um tempo proporcional a $O(V \log_2 |V|)$ para as $|V|$ chamadas a essa função Dijkstra.

Não foi necessário a criação da função de heap durante o desenvolvimento, pois o *Python* possui uma biblioteca chamada *heapq* que realiza a organização, adição e remoção dos elementos em uma heap mínima.

A função de busca de caminho mínimo, baseada no algoritmo de Dijkstra, desenvolvida pelo autores está disponibilizada a seguir.

```

1 def dijkstra(matriz, vInicio, vTermino,
2             preferencias):
3     n = len(matriz)
4     custo = [float('inf')] * n
5     rota = [-1] * n
6     visitados = [False] * n
7
8     custo[vInicio] = 0
9
10    fila_prioridade = [(0, vInicio)]
11
12    while fila_prioridade:
13        (atual_custo, u) = heapq.heappop(
14            fila_prioridade)
15
16        if visitados[u]:
17            continue
18
19        visitados[u] = True
20
21        for v in range(n):
22            if not visitados[v] and matriz[
23                u][v] != -1:
24                custo_atualizado = custo[u]
25                + matriz[u][v] +
26                preferencias[u][v]
27            if custo_atualizado < custo
28                [v]:
29                custo[v] =
30                    custo_atualizado
31                rota[v] = u
32                heapq.heappush(
33                    fila_prioridade, (
34                        custo[v], v))
35
36    caminho = []
37    i = vTermino
38
39    while i != vInicio:
40        caminho.insert(0, i)
41        i = rota[i]
42
43    caminho.insert(0, vInicio)
44
45    return caminho, custo[vTermino]
```

Ademais, como um esforço adicional para proporcionar maior autonomia na escolha do indivíduo, foi criado um menu com quatro opções, a saber: 1. Menor Caminho, 2. Evitar Escada, 3. Evitar Rampa e 4. Evitar Escada e Rampa. A transição entre os pisos, como previamente detalhado na seção 3.1,

pode ocorrer por meio de escada, rampa ou elevador. Considerando que o elevador é uma opção livre de restrições, não sendo necessário evitá-lo, permanece sempre como uma alternativa disponível para a locomoção.

Entretanto, escadas ou rampas podem causar desconforto a determinados indivíduos. Portanto, permitindo que cada pessoa escolha a opção mais adequada às suas preferências, o caminho de menor custo, ou seja, com menor extensão, é determinado com base nessa escolha.

Após a seleção, são atribuídos pesos elevados às arestas que compõem o caminho indesejado. Esse procedimento visa garantir que, ao calcular o custo na função "dijkstra", conforme apresentado anteriormente na linha 21, a soma leve em consideração o peso da aresta na matriz e a preferência do indivíduo. A matriz de preferência possui a mesma dimensão da matriz de modelagem e é preenchida principalmente com zeros, exceto nas arestas com peso de 500 (um valor superior a qualquer caminho possível dentro do grafo, escolhido pelos autores). Esses pesos são atribuídos às arestas que precisam ser "invalidadas", atendendo assim à solicitação específica do indivíduo.

Nessa perspectiva, foi construída a função denominada "escolhaCaminho" para receber a escolha do indivíduo e atribuir os pesos às arestas, caso haja preferências. A função está apresentada a seguir:

```

1 def escolhaCaminho(preferencias):
2     situacao = 0
3
4     while situacao != "0":
5         # Limpar o terminal no sistema UNIX
6         # (Linux ou macOS)
7         # os.system('clear')
8         # Limpar o terminal no Windows
9         os.system('cls')
10
11        print("[ bold]----- Menu
12        -----[/bold]")
13
14        print("1) Menor caminho\n2) Evitar
15        escada\n3) Evitar rampa\n4)
16        Evitar rampa e escada\n0) Sair
17        ")
18
19        situacao = input("Digite o numero
20        correspondente a sua escolha:
21        ")
22
23        if situacao == "1":
24            return situacao
25        elif situacao == "2":
26            # Adiciona um valor muito alto
27            # a aresta para invalidar o
28            # caminho
29            preferencias[2][14] = 500
30            preferencias[14][2] = 500
31            return situacao
32        elif situacao == "3":
33            # Adiciona um valor muito alto
34            # a aresta para invalidar o
35            # caminho
36            preferencias[25][12] = 500
37            preferencias[12][25] = 500
38            return situacao
39        elif situacao == "4":
40            # Adiciona um valor muito alto
```

```

        as arestas para invalidar o
        caminho
28     preferencias[2][14] = 500
29     preferencias[14][2] = 500
30     preferencias[25][12] = 500
31     preferencias[12][25] = 500
32     return situacao
33 elif situacao != "0":
34     print("Entrada incorreta, por
        favor coloque uma das
        opcoes corretas")

```

Desse modo, tornou-se possível solucionar o problema de busca pelo menor caminho, baseando-se nos beacons distribuídos conforme apresentado na seção 3.1. Além disso, a abordagem contribui para facilitar a locomoção dos indivíduos dentro do Instituto de Matemática e Computação (IMC), levando em consideração suas preferências e necessidades específicas.

3.3 Aspectos Desafiadores do Estudo

Nesta seção, serão apresentados os desafios enfrentados durante o desenvolvimento das soluções propostas na seção 3.2, juntamente com a abordagem adotada pelos autores para superar essas dificuldades.

3.3.1 Desafios para a Criação da Modelagem de Distribuição dos Dispositivos BLE

Durante o processo de modelagem, os autores enfrentaram desafios significativos. Inicialmente, foi um obstáculo conceber uma solução que garantisse a não fragilidade da rede. Assim, a modelagem foi abordada de várias maneiras até que chegaram à solução proposta neste trabalho, que envolve a não utilização dos beacons com sua capacidade máxima.

Outro desafio surgiu em relação à referenciação do elevador, e inicialmente não foi encontrada uma solução imediata para essa dificuldade. Após diversas tentativas de modelagem, chegou-se à conclusão de que a colocação de um beacon no elevador seria a melhor solução, permitindo que o mesmo beacon fosse utilizado nos pisos 1 e 2.

3.3.2 Desafios para a Implementação de um Algoritmo de Caminho Mínimo Eficiente

Inicialmente, os autores desenvolveram o protótipo de cálculo de caminho mínimo baseado no algoritmo de Dijkstra, sem a utilização de heap, para encontrar o menor custo em uma lista que continha os custos dos caminhos. Dessa forma, para identificar o menor custo, era necessário percorrer toda a lista a cada iteração. Consequentemente, a função de Dijkstra, na aplicação em questão, com uma matriz 29x29, sempre executava acima de 1 milissegundo. Apesar de ser um custo temporal relativamente baixo, em aplicações com matrizes de dimensões maiores, esse tempo pode se tornar significativo para o cálculo da rota. Assim, os autores buscaram opções na literatura para implementar um algoritmo mais eficiente, chegando à solução apresentada na seção 3.2.2.

Além disso, houve extensas discussões sobre como atribuir pesos para desfavorecer certos caminhos, de acordo com as preferências dos usuários. A solução encontrada para essa

problemática foi a criação de uma matriz de mesma ordem que a matriz adjacente, 29x29, representando o mapeamento de distribuição dos beacons. Essa matriz foi preenchida em todas as posições com o valor zero e recebeu um peso elevado (500 metros) nas posições que representam os caminhos que os usuários desejam evitar.

4 Resultados

Nessa seção, estão apresentados os resultados obtidos com as soluções propostas para as problemáticas que motivaram o estudo. A referência 4.1 diz respeito à solução proposta para a distribuição dos beacons, detalhada na seção 3.2.1, enquanto 4.2 refere-se à solução de busca por caminho mínimo, detalhada na seção 3.2.2.

4.1 Resultado Mapeamento de Distribuição dos Dispositivos BLE

De acordo com o mapeamento realizado na seção 3.1, 29 beacons são utilizados. Espera-se que esse número atenda às necessidades de referenciar locais que compõem a estrutura do Instituto de Matemática e Computação (IMC). Além disso, como os beacons não estão posicionados de modo a utilizar sua capacidade máxima, 70 metros, espera-se que a rede tenha estabilidade para manter a conexão caso um dispositivo deixe de funcionar. A Figura 11 ilustra como esse processo ocorre.

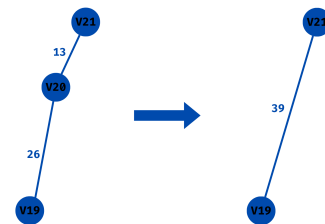


Figura 11: Conectividade dos Dispositivos BLE.

Como observado, há uma aresta entre os vértices V19 e V20, representando uma distância de 26 metros, e outra entre os vértices V20 e V21, representando 13 metros. Considerando que o pressuposto dos autores seja válido, se o vértice V20 deixar de funcionar, os vértices V19 e V21 estarão a uma distância de 39 metros. Isso, com base em sua alcançabilidade de 70 metros, manteria a conectividade da rede.

Além disso, considerando o custo médio de cada dispositivo BLE, cerca de R\$500,00 (quinhentos reais), e a quantidade desses dispositivos utilizados na modelagem proposta pelos autores é 29, resulta em um valor total aproximado de R\$14.500,00 (catorze mil e quinhentos reais). Portanto, é esperado que o modelo proposto satisfaça adequadamente as exigências de distribuição de beacons no Instituto de Matemática e Computação (IMC).

4.2 Resultado Busca por Caminho Mínimo

Para expor os resultados obtidos, primordialmente vale citar que a codificação das funções debatidas na seção 3.2.2 foi realizada em um notebook com o processador AMD Ryzen 5

5500U com Radeon Graphics 2.10 GHz, 8GB de memória principal (RAM), sistema operacional Windows 10 Pro 64 bits versão 22H2 e o ambiente de desenvolvimento integrado (IDE) utilizado foi o Visual Studio Code. Outrossim, a linguagem utilizada foi o *Python* 3.11.2, acompanhada das seguintes bibliotecas ([Python Software Foundation, 2023](https://www.python.org/)):

- *NumPy*: Adiciona suporte para arrays e matrizes multidimensionais, juntamente com funções matemáticas de alto nível.
- *Heapq*: Oferece implementações de filas de prioridade baseadas em heaps, úteis em algoritmos como Dijkstra.
- *re (Regular Expressions)*: Fornece suporte a expressões regulares, permitindo a busca e manipulação eficiente de padrões em strings.
- *os*: Fornece uma interface para funcionalidades dependentes do sistema operacional, incluindo operações de leitura/escrita de arquivos e manipulação de caminhos de diretórios.
- *Rich*: Aprimora a formatação da saída no console, oferecendo recursos para adicionar cores, estilos de texto, tabelas e gráficos.
- *timeit*: Mede o tempo de execução de trechos de código, proporcionando uma maneira simples de cronometrar operações em Python.

A seleção destas bibliotecas foi realizada, levando em consideração suas capacidades específicas para otimizar a eficiência e funcionalidade do código. Para utilização, é necessário instalar as bibliotecas *NumPy* e *Rich* através dos comandos *pip install numpy* e *pip install rich*, respectivamente. As demais bibliotecas estão incluídas na biblioteca padrão do *Python* e, portanto, não requerem instalação adicional.

Para avaliar a eficiência da busca pelo menor caminho, implementada inspirada no algoritmo de Dijkstra utilizando heap, realizou-se a medição do tempo, em milissegundos, da função denominada "dijkstra", apresentada na Seção 3.2.2. A função *timeit.default_timer()*, do módulo *timeit*, foi empregada para realizar essa medição, pois é utilizada para medir intervalos de tempo. Ela foi aplicada antes da chamada da função e depois, sendo atribuída a variáveis nas respectivas instâncias, permitindo assim, ao subtrair o tempo inicial do tempo final, obter o tempo de execução da função "dijkstra".

Dessa maneira, sem interferência de concorrência de CPU e com a execução exclusiva do código, foram armazenadas informações sobre a execução em um arquivo TXT. Para cada uma das possíveis preferências dos usuários, realizou-se a execução três vezes, com seleção aleatória de vértices, com o objetivo de obter os tempos de execução. Com base nesses experimentos, a Tabela 1 foi criada para apresentar os resultados obtidos.

Por fim, com base nos tempos de execução, observa-se uma eficiência do algoritmo de busca pelo menor caminho. A aplicação bem-sucedida em cenários variados e a capacidade de lidar com diferentes origens e destinos sugerem um resultado satisfatório. Entretanto, pode-se explorar oportunidades de otimização, especialmente em cenários onde os tempos de execução são mais longos.

Tabela 1: Tempo de Execução da Função Dijkstra

Origem	Destino	Preferência	Milissegundos
V5	V19	1	0.311699986923486
V2	V19	1	0.534399994648993
V0	V28	1	0.706099992385134
V0	V28	2	0.350299989804625
V25	V1	2	0.354100018739700
V6	V24	2	0.706999999238178
V19	V5	3	0.342199986334890
V0	V28	3	0.344100000802427
V2	V17	3	0.609000009717419
V0	V25	4	0.349099980667233
V1	V27	4	0.355499971192330
V26	V3	4	0.664699997287243

5 Conclusões

Indivíduos com deficiência visual frequentemente enfrentam desafios acerca da locomoção, principalmente no que tange a navegação em ambientes indoor. Portanto, nota-se a importância de realizar estudos que proponham soluções para essas dificuldades ou contribuam para aprimorar as abordagens existentes.

Nessa perspectiva, essa pesquisa foi realizada com o objetivo de atender às necessidades de melhoria na modelagem da distribuição dos dispositivos BLE no Instituto de Matemática e Computação (IMC) da Universidade Federal de Itajubá (UNIFEI), bem como propor um algoritmo de busca por caminho mínimo para aprimorar a experiência desses indivíduos.

No tocante ao mapeamento da distribuição dos dispositivos BLE, o estudo visou alcançar a configuração ideal, utilizando 29 beacons, com a expectativa de proporcionar uma cobertura eficiente para a estrutura do IMC. A distância entre os beacons foi cuidadosamente planejada na expectativa de garantir estabilidade na rede, mesmo em caso de falha de um dispositivo, mantendo a conectividade. No entanto, a validação dessa hipótese requer testes reais no ambiente em questão. Quanto ao custo, a preocupação foi utilizar a menor quantidade possível de beacons, tornando o investimento coerente.

No que diz respeito à busca pelo menor caminho, a implementação do algoritmo de Dijkstra, com o uso da estrutura Heap, demonstrou eficiência, conforme evidenciado pelos resultados dos testes. O tempo de execução foi medido e revelou-se razoável para diferentes preferências dos usuários. A aplicação bem-sucedida em diversos cenários e a capacidade de lidar com diferentes origens e destinos destacam a versatilidade do algoritmo. No entanto, a análise sugere a possibilidade de otimizações em casos específicos de tempos de execução mais longos.

Em resumo, os resultados indicam uma solução coerente para a distribuição de beacons e uma implementação eficiente para a busca pelo menor caminho. A abordagem adotada atende às exigências do ambiente do IMC, oferecendo uma base sólida para futuras melhorias e aplicações mais avançadas.

Referências

- Bickle, A. (2020). *Fundamentals of Graph Theory*. American Mathematical Society, 1st edição.
- Chartrand, Linda Lesniak, P. Z. (2010). *Graphs Digraphs*. CRC Press, 5st edição.
- Feofiloff, P., Kohayakawa, Y., & Wakabayashi, Y. (2004). *Uma Introdução Sucinta à Teoria dos Grafos*. São Paulo: Editora da Universidade de São Paulo.
- Goldstein Market Intelligence (2019). Global indoor positioning and indoor navigation (ipin) market outlook, 2024. Acessado em 20 de setembro de 2023.
- Gonçalves, D. M. (2015). Análise de filas de prioridades para o algoritmo de dijkstra em redes de malhas sem fio. B.S. thesis, Universidade Tecnológica Federal do Paraná.
- Jacundá Santos, J. v. A., Ribeiro de Brito, G. L., & Barbosa, G. V. (2019). Caminho mínimo de redes conectadas utilizando grafos. *REVISTA CEREUS*, 11(3), 61–68.
- Jurkiewicz, S. (2009). *Grafos—uma introdução*. São Paulo: OBMEP.
- Lei-nº13.146 (2015). Lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com deficiência).
- Martins, L. R. d. S., Silva, M. T., Moreira, B., Lima, D. A. d. S., Souza, C. F. S., Santos, L. G. F., & Junior, C. A. C. S. (2021). Guide2blind: Sistema háptico-sonoro de orientação para deficientes visuais em ambientes fechados-fase 2. *Brazilian Journal of Development*, 7(3), 23417–23435.
- Melo, H. F. R. (1991). *Deficiência Visual: Lições Práticas de Orientação e Mobilidade*. Ed. UNICAMP: Campinas.
- Oliveira, T. A. B., Santos, F. A. N., & Cinelli, M. J. (2017). Sistemas de navegação indoor e sistema de compras para pessoas com deficiência visual: potenciais no uso em supermercado. *Human Factors in Design*, 6(11), 022–042.
- Ossada, J., Aparecida, S., Ossada, R., Kawamoto, L., Cristina, S., & Martini, S. (2016). Ic4u: O desenvolvimento de um método de navegação indoor para deficientes visuais de baixo custo. *REVERTE -Revista de Estudos e Reflexões Tecnológicas da Faculdade de Indaiatuba*, 14, 1–9.
- Python Software Foundation (2023). Python.org. <https://www.python.org/>.
- Santos, C. A. A. (2014). Caminhos mais curtos em localização em espaços fechados. Master's thesis.
- Silvio, d. L. P., Akamine, L. T., & de Lira, L. N. (2016). Busca assíncrona de caminhos mínimos. *Boletim Técnico*, 41, 7.
- Simões, W. C. S., Silva, Y. M. L., & de Lucena Jr, V. F. (2016). Assistente de navegação indoor para deficientes visuais através de pedestrian dead reckoning e correção de estimativa de posição por reconhecimento de padrões.
- West, D. B. (2001). *Introduction to Graph Theory*. University of Illinois - Urbana: Pearson Education, Inc., second edição.
- Wohlfahrt, E. A., Roque, A. d. S., & Santos, C. P. (2015). Navegação indoor baseada em etiquetas rfid: Possibilidades de orientação para deficientes visuais. Acessado em 20 de setembro de 2023.

