



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

Grupo 1

2022004574 - Daniel Enrique Gonzalez de Aguiar

2022004823 - Huandy Calini de Camargo Silva

2022004242 - João Lucas Moraes de Oliveira

2022002542 - Letícia Vitória dos Santos

Contextualização



O mundo no qual vivemos é majoritariamente construído visando atender as necessidades de pessoas sem deficiência.


Isso implica que pessoas com deficiência precisam encontrar formas alternativas para executar tarefas cotidianas, incluindo locomover-se em determinados ambientes.

Técnicas de locomoção e mobilidade utilizadas por pessoas com deficiência visual

- Bengala branca;
- Cão guia;
- Mapas mentais → Memorização de características físicas de um ambiente, isto é, disposição de objetos e trajetórias acessíveis entre dois ou mais pontos.



Desafios enfrentados por pessoas com deficiência visual ao locomover-se em ambientes pouco familiares



Ambientes não familiares ou pouco frequentados trazem consigo uma série de desafios, entre os quais podem ser citados:

- Inexistência de mapas mentais referentes a aquele ambiente;
- Ausência de sinalizações táteis e/ou sonoras;
- Orientações por meio de sistemas gps podem não estar disponíveis.

Navegação indoor

Capacidade de orientação e locomoção em ambientes internos.

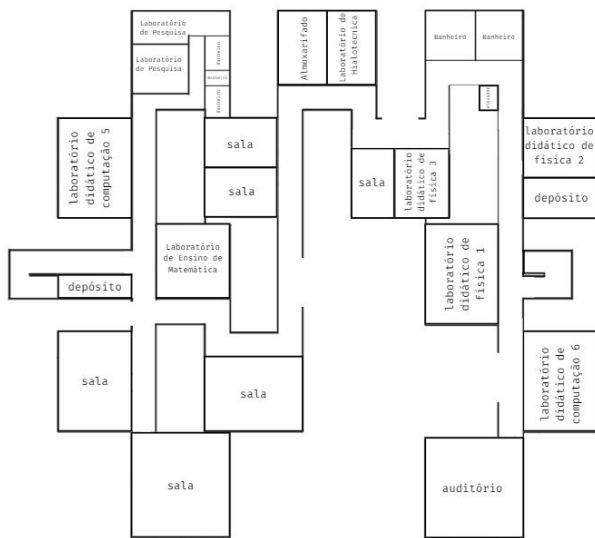
- Pode ser obtida por meio de tecnologias como dispositivos beacon e wifi;
- Auxilia pessoas com deficiência visual ao fornecer orientações não visuais acerca do ambiente em que se encontram, isto é, contribui para a minimização dos desafios supracitados.

Problemática

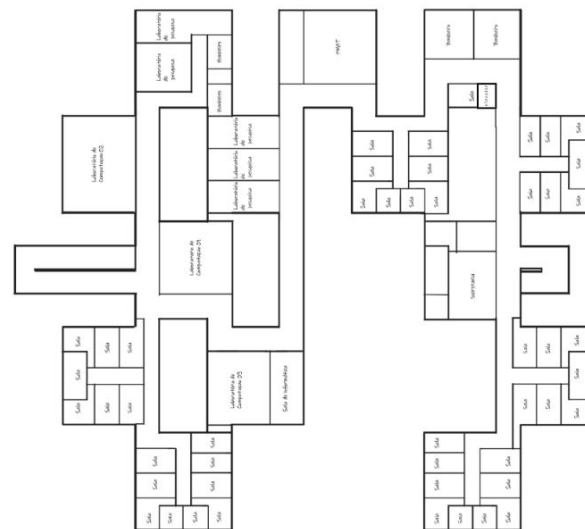
- O estudo considera como cenário de estudo o Instituto de Matemática e Computação (IMC) da Universidade Federal de Itajubá (UNIFEI);
- Poucas soluções no instituto para torná-lo mais acessível.
- Desafios para implementação de uma aplicação válida:
 - Mapeamento de distribuição de beacons com a menor quantidade possível;
 - Encontrar caminho mínimo a partir desse mapeamento.

Mapeamento

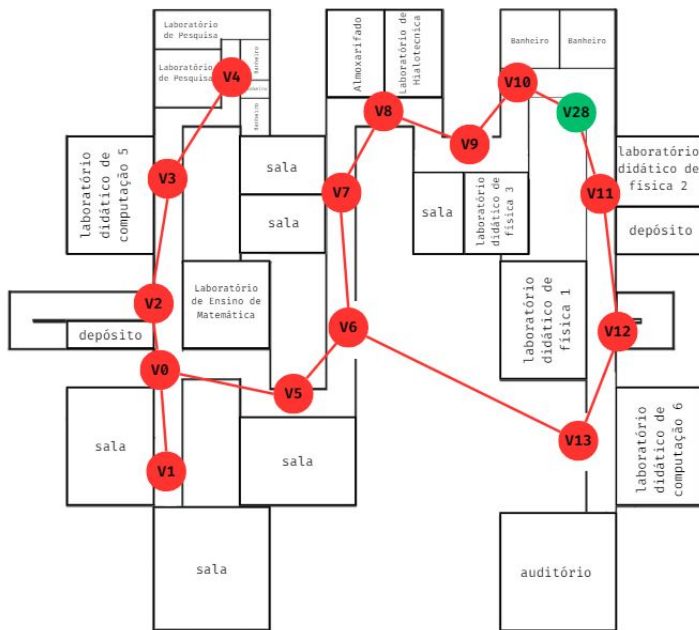
Piso 1



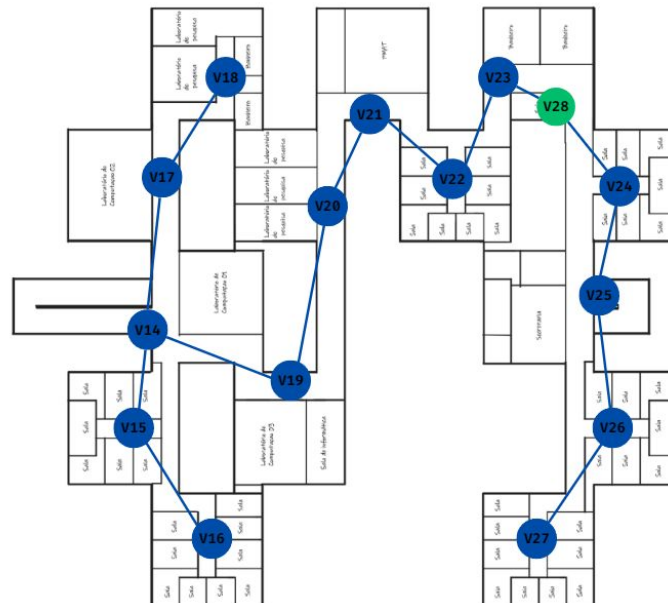
Piso 2



Piso 1



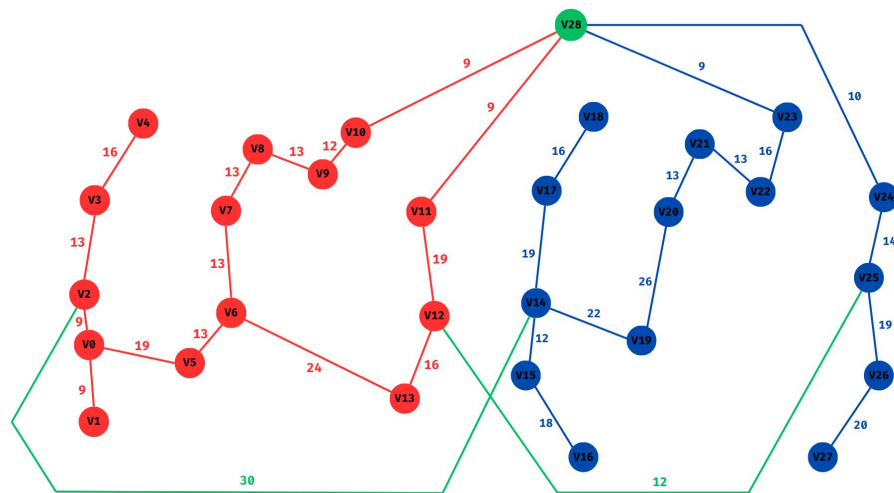
Piso 2



Grafo resultante

Características:

- Valorado;
- Não dirigido;
- Peso das arestas indica distância em metros entre pontos conectados;
- Estrutura composta por 29 vértices e 31 arestas;



O uso da matriz tem como objetivo fazer uma tradução do que se encontra nos mapas elaborados durante o trabalho, para uma forma que permita ser usada pelo algoritmo de caminho mínimo desenvolvido.

[illegible]

Algoritmo de Dijkstra

```
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     fila_prioridade = [(0, vInicio)]
10
11     while fila_prioridade:
12         (atual_custo, u) = heapq.heappop(
13             fila_prioridade)
14
15         if visitados[u]:
16             continue
17
18         visitados[u] = True
19
20         for v in range(n):
21             if not visitados[v] and matriz[
22                 u][v] != -1:
23                 custo_atualizado = custo[u]
24                 + matriz[u][v] +
25                 preferencias[u][v]
26                 if custo_atualizado < custo
27                     [v]:
28                     custo[v] =
29                         custo_atualizado
30                     rota[v] = u
31                     heapq.heappush(
32                         fila_prioridade, (
33                             custo[v], v))
```

```
27     caminho = []
28     i = vTermino
29
30     while i != vInicio:
31         caminho.insert(0, i)
32         i = rota[i]
33
34     caminho.insert(0, vInicio)
35
36     return caminho, custo[vTermino]
```

Foi escolhido este algoritmo pela capacidade que proporciona para encontrar o caminho mínimo desde um vértice de início para todos os outros vértices com peso positivo, demonstrando, ao mesmo tempo, um bom rendimento computacional.

Função escolhaCaminho

```
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         print("1) Menor caminho\n2) Evitar
14             escada\n3) Evitar rampa\n4)
15             Evitar rampa e escada\n0) Sair
16             ")
17         situacao = input("Digite o numero
18             correspondente a sua escolha:
19             ")
```

```
13
14     if situacao == "1":
15         return situacao
16     elif situacao == "2":
17         # Adiciona um valor muito alto
18         # a aresta para invalidar o
19         # caminho
20         preferencias[2][14] = 500
21         preferencias[14][2] = 500
22         return situacao
23     elif situacao == "3":
24         # Adiciona um valor muito alto
25         # a aresta para invalidar o
26         # caminho
27         preferencias[25][12] = 500
28         preferencias[12][25] = 500
29         return situacao
30     elif situacao == "4":
31         # Adiciona um valor muito alto
32         # as arestas para invalidar o
33         # caminho
34         preferencias[2][14] = 500
35         preferencias[14][2] = 500
36         preferencias[25][12] = 500
37         preferencias[12][25] = 500
38         return situacao
39     elif situacao != "0":
40         print("Entrada incorreta , por
41             favor coloque uma das
42             opcoes corretas")
```

Permite que a pessoa que utilize o algoritmo possa ter a possibilidade de descartar uma rota se esta inclui uma estrutura que representa uma dificuldade para seu deslocamento.



Execução do código

Desafios



Criação da Modelagem de Distribuição dos Dispositivos BLE

- Fragilidade do mapeamento usando beacons na capacidade máxima;
- Referenciação do Elevador.



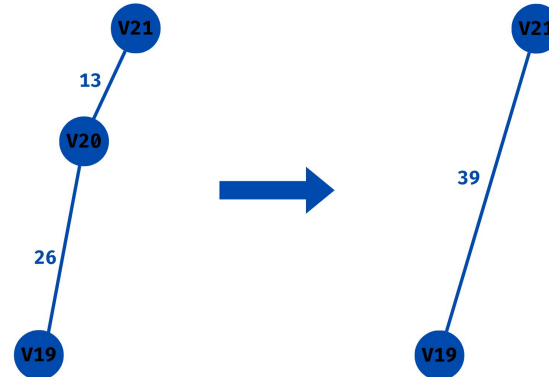
Implementação de um Algoritmo de Caminho Mínimo Eficiente

- Algoritmo de Dijkstra;
- Invalidar caminho de acordo com a escolha do usuário.

Resultados

Criação da Modelagem de Distribuição dos Dispositivos BLE

- Mapeamento sem utilizar os dispositivos BLE na capacidade máxima, totalizando 29 beacons;
- Valor médio para instalação: R\$14.500,00.



Resultados

Implementação de um Algoritmo de Caminho Mínimo Eficiente

- Implementação que resulta no caminho a ser seguido, instruções a cada etapa e metragem total a ser percorrida;
- Diminuição de consumo de tempo do algoritmo de Dijkstra realizando a varredura da lista com Heap.

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

Referências

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.

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.



Dúvidas?