

**UNIVERSIDADE NOVE DE JULHO
DIRETORIA DOS CURSOS DE INFORMÁTICA**

**ANDRÉ CARDOSO ARAGÃO - RA 918119401
MARIANA RODRIGUES DE SOUZA - RA 918114559
JEAN LUCA BERNE DE OLIVEIRA - RA 918100772**

PROJETO EM SISTEMAS INTELIGENTES

**SÃO PAULO
2021**

ANDRÉ CARDOSO ARAGÃO - RA 918119401
MARIANA RODRIGUES DE SOUZA - RA 918114559
JEAN LUCA BERNE DE OLIVEIRA - RA 918100772

PROJETO EM SISTEMAS INTELIGENTES

Trabalho apresentado à Universidade Nove de Julho,
UNINOVE, em cumprimento parcial às exigências da disciplina
de Projeto em sistemas inteligentes, sob orientação dos Prof.
Edson Melo de Souza e Sergio Joao Guimaraes da Silva

SÃO PAULO
2021

SUMÁRIO

1.	INTRODUÇÃO E OBJETIVO	4
2.	DESCRIÇÃO DO SOFTWARE	5
2.1.	DESCRIÇÃO DAS PARTES QUE COMPÕE O SOFTWARE	7
2.2.	UTILIZAÇÃO DO SOFTWARE	7
3.	VANTAGENS E DESVANTAGENS DO ALGORITMO	9
3.1.	VANTAGENS	9
3.2.	DESVANTAGENS	9
4.	LINKS PARA ACESSO E EVIDÊNCIAS	11
5.	CONCLUSÃO	12
6.	REFERÊNCIAS	13

1. INTRODUÇÃO E OBJETIVO

Nos últimos anos, nota-se um grande crescimento do comércio online, especialmente nos últimos 2 anos devido à pandemia da Covid-19, que impulsionou o ecommerce no Brasil e no mundo. Só em 2020, foram 13 milhões de novos consumidores, um aumento de 29% em relação ao ano anterior, segundo pesquisa Webshoppers 43, feita pela consultoria Ebit/Nielsen em parceria com o Bexs Banco. Consequentemente, para suprir essas demandas, cresceram também o número de empresas de entrega.

O objetivo do presente trabalho é criar um sistema de logística de entrega de produtos. Com o mercado online estando mais competitivo, uma das vantagens é a velocidade na entrega, uma vez que os clientes estão cada vez mais imediatista. Para suprir essa necessidade de entregas mais rápidas, foi utilizado o algoritmo de busca A Estrela, que propõe a estratégia mais conhecida de busca pela melhor escolha, procurando o melhor caminho baseado no custo e distância, possibilitando assim oferecer prazos de entrega mais rápidos.

2. DESCRIÇÃO DO SOFTWARE

Foi desenvolvido um software na linguagem Python com um algoritmo de Busca com base em inteligência artificial, a Busca A* (A estrela). Para desenvolvermos esse software era necessário a idealização de um mapa para resolver nosso problema, fizemos um mapa usando como base a cidade de Osasco, foi também definidos alguns bairros que seriam as “sedes” da transportadora e também definida uma distribuidora principal, que seria a origem dos pacotes de entrega.

Foi escolhido o bairro “Bela Vista” como bairro de origem, o destino é escolhido pelo usuário do software, no mapa colocamos a unidade “Remédios” como destino para demonstração. O mapa é baseado no sistema de grafos, onde cada nó é um bairro.

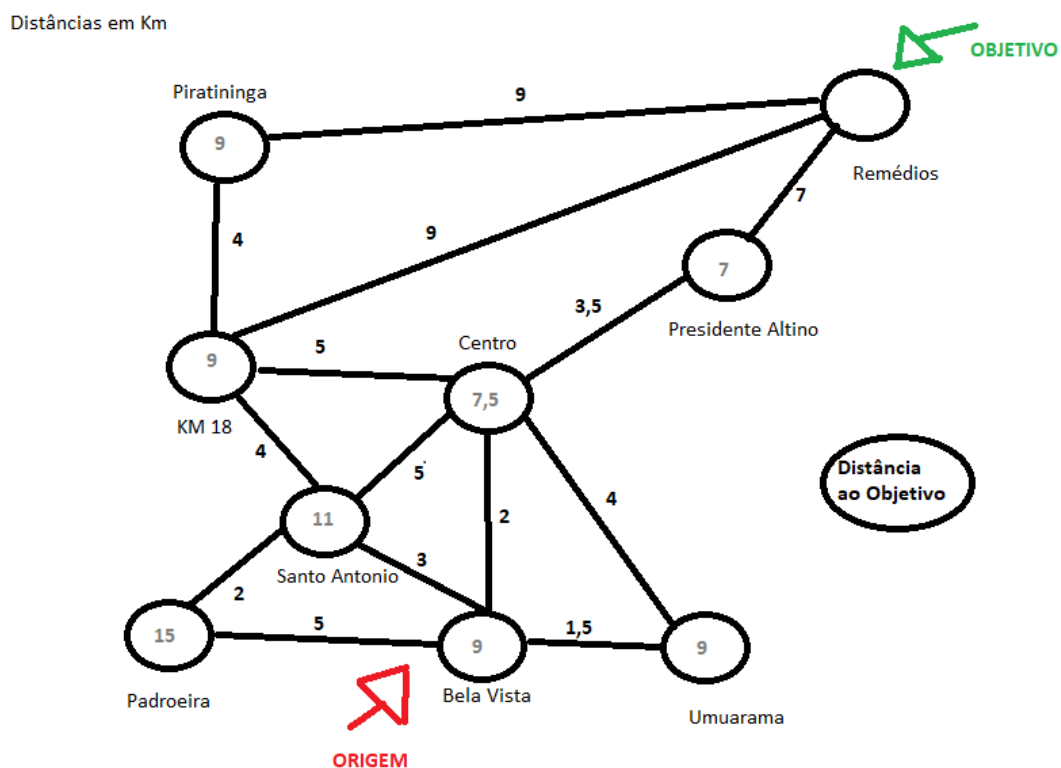


Figura 1 - Mapa

As distâncias foram baseadas numa aproximação da distância real entre os bairros da cidade de Osasco, para seguirem um padrão.

Para realizar o cálculo da busca A* é feita usada a seguinte fórmula:

$$f(n) = g(n) + h(n)$$

onde:

$g(n)$ = custo para alcançar cada nó

$h(n)$ = custo para ir do nó até o objetivo

Para cada nó, os cálculos são feitos para cada bairro adjacente, é comparado os valores e selecionado o nó com o menor resultado, assim selecionando o nó que haverá a menor distância, esse processo se repete até chegar no destino final. Abaixo há o exemplo do mapa, mas feito no software, com início em Bela Vista, com destino Remédios.

```
Digite o destino:
1- Remédios
2- Piratininga
3- Presidente Altino
4- Km18
5- Centro
6- Santo Antonio
7- Umuarama
8- Padroeira
1

Atual: belaVista
centro - 9.5
umuarama - 10.5
santoAntonio - 14.0
padroeira - 20.0

Atual: centro
presAltino - 10.5
km18 - 14.0

Atual: presAltino
remedios - 7.0

Atual: remedios
In [2]:
```

Figura 2 – Software executado

Dá para notar que em cada processo, é escolhido o com menor valor de distância.

2.1. DESCRIÇÃO DAS PARTES QUE COMPÕE O SOFTWARE

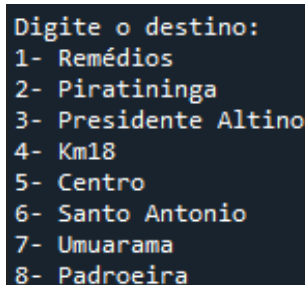
O software é composto por cinco arquivos em Python, dois arquivos foram feitos com a classe “bairro” e a classe “adjacente” contendo as informações dos nós, com seu nome e o nome dos nós adjacentes.

O arquivo “Mapa.py” contém as informações necessárias para definir o melhor caminho, contém a distância entre os nós, e a distância de cada nó até o destino final. O destino final é escolhido pelo próprio usuário. O arquivo “VetorOrdenadoAdjacente.py” é onde é feita a ordenação dos nós, baseada na fórmula da busca A*, o com menor resultado é o escolhido.

O arquivo “AEstrela.py” é onde tem o método buscar, que chama todos os outros métodos e classes para executar a tarefa.

2.2. UTILIZAÇÃO DO SOFTWARE

Para utilizar o software, é necessário executar o arquivo “AEstrela.py”, ele irá acionar o método buscar, e irá solicitar o destino.



```
Digite o destino:  
1- Remédios  
2- Piratininga  
3- Presidente Altino  
4- Km18  
5- Centro  
6- Santo Antonio  
7- Umuarama  
8- Padroeira
```

Figura 3 – Selecionar destino

Ao selecionar o destino, o software irá selecionar o caminho mais viável utilizando o cálculo da busca, e irá mostrar na saída todo o caminho feito, com as informações da distância calculada em cada nó e etapa, até chegar no destino final, como mostrado abaixo.

```
Atual: belaVista
centro - 7.0
umuarama - 9.0
santoAntonio - 10.0
padroeira - 15.0

Atual: centro
presAltino - 8.5
km18 - 10.0

Atual: presAltino
remedios - 12.5

Atual: remedios
piratininga - 9.0

Atual: piratininga
```

Figura 4 – Software executado, com as informações de distância

3. VANTAGENS E DESVANTAGENS DO ALGORITMO

3.1. VANTAGENS

Como o software que desenvolvemos é algo voltado para entrega de produtos a utilização do algoritmo de Busca A* possibilita a criação de um mapa com rotas mais viáveis e rápidas para que nossos clientes recebam seus produtos sem atrasos. E o algoritmo de busca A* faz exatamente isso, e apenas precisamos mostrar a ele um local de início e um destino, assim através de IA ele irá analisar e escolher a melhor rota possível para que não tenham atrasos ou outros tipos de problemas em nossas entregas. Mas de uma forma geral o algoritmo A* é melhor que os outros que fazem algo parecido pois enquanto outros algoritmos expandem seus nós em grandes escalas o algoritmo A* é extremamente mais direto analisando apenas a rota mais eficiente caso exista claro, se não houver rotas ele não dará um caminho. Uma das maiores vantagens do algoritmo de busca A* é sua eficiência e velocidade na hora de analisar uma rota do ponto inicial ao destino final, ele sem dúvida nem se compara com os algoritmos mais simples de buscas, ele chega a ser 8x mais rápido que eles, sendo assim um dos algoritmos mais utilizados nesses tipos de softwares, e não importa o tamanho do caminho que ele terá que analisar, o algoritmo A* irá ganhar disparado sem dúvida.

3.2. DESVANTAGENS

O algoritmo de A* possui poucas desvantagens, algumas desvantagens que esse algoritmo possui é temos que sempre estar o atualizando pois se alguma rua que está no mapa for alterada ou estiver congestionada a IA terá que escolher outra rota melhor, para não causar problemas ao entregador e não deixar o cliente insatisfeito com o atraso da entrega, outra desvantagem pode ser o desempenho em si do software mas também depende do tamanho da rota que ele irá analisar, se for pequena ele talvez não apresente problemas de desempenho, mas se for muito grande existe uma grande possibilidade de apresentar problemas de desempenho. Algo que deve ser levado muito a sério também é sua implementação, pois se caso for feita de qualquer jeito o software final irá apenas causar problemas e não facilitará a vida de quem

estiver o utilizando. Caso saia tudo bem com seu desenvolvimento ele sempre irá procurar a rota que ele designa como ótima para o cliente contanto com sua posição atual.

4. LINKS PARA ACESSO E EVIDÊNCIAS

Os arquivos de documentação, do software e das evidências podem ser acessados pelo github através do link abaixo. Há evidências tanto em vídeo quanto em imagem do software sendo utilizado, abaixo há uma evidência visual do software após ser executado.

```
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/AA_8SEM/Projeto/Software/Teste Busca A Estrela(escolher destino)/
AEstrela.py', wdir='D:/AA_8SEM/Projeto/Software/Teste Busca A Estrela(escolher destino)')

Digite o destino:
1- Remédios
2- Piratininga
3- Presidente Altino
4- Km18
5- Centro
6- Santo Antonio
7- Umuarama
8- Padroeira
1

Atual: belaVista
centro - 9.5
umuarama - 10.5
santoAntonio - 14.0
padroeira - 20.0

Atual: centro
presAltino - 10.5
km18 - 14.0

Atual: presAltino
remedios - 7.0

Atual: remedios

In [2]:
```

Figura 5 – Evidência

Link para acesso no GitHub:

https://github.com/andrecardoso19/projetos_ciencia_da_computacao/tree/main/Projeto%20em%20Sistemas%20Inteligentes

5. CONCLUSÃO

O presente trabalho apresenta o desenvolvimento do sistema de logística de entrega, utilizando o algoritmo de busca heurística A Estrela. Desenvolvido na linguagem Python, o algoritmo é vantajoso pois utiliza-se de heurística, ou seja, conhecimentos sobre o problema para propor a melhor solução, diferente dos algoritmos de busca cega.

O sistema cumpre o objetivo, uma vez que a ideia é aumentar a agilidade da entrega de produtos, considerando o crescimento do mercado online e a necessidade de entregas cada vez mais rápidas. Foi utilizado como base a cidade de Osasco, e inserido bairros e informações sobre as distâncias entre os pontos, para que fosse calculado o menor caminho a partir da sede até o ponto desejado. Comparando o algoritmo A Estrela com o de Busca Gulosa, que também é um algoritmo de busca pela melhor escolha, notamos que o utilizado possui a vantagem de considerar o custo do deslocamento e a função heurística para decidir o nó a ser expandido, ou seja, o bairro mais próximo do objetivo, e consequentemente o caminho mais vantajoso a ser seguido.

6. REFERÊNCIAS

A* Algorithm Concepts and Implementation. **SIMPLILEARN**, 2021. Disponível em: < <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm> >. Acesso em: 18 de novembro de 2021.

FIGUEIREDO, João Neiva de; GONZAGA, Clóvis C. Aplicação de métodos de busca em grafos com nós parcialmente ordenados à locação de torres de transmissão. **SCIELO**, 2003. Disponível em: <<https://www.scielo.br/j/pope/a/RXRmqqbzxdQY4RQ9f5drYZg/?lang=pt/>>. Acesso em: 15 de novembro de 2021.

ARAUJO, Ricardo. Algoritmos de Busca para Inteligência Artificial. **RICARDO ARAUJO**, 2020. Disponível em: < <https://ricardomatsumura.medium.com/algoritmos-de-busca-para-intelig%C3%A2ncia-artificial-7cb81172396c> >. Acesso em: 15 de novembro de 2021.