|  |  |
| --- | --- |
| Unianchieta-01-1  **Faculdade Anchieta**  **Jundiaí - SP**  :  Guilherme Luis De Lima  2101205@escolas.anchieta.br  Leonardo de Paiva Finavaro  2104619@escolas.anchieta.br  Rafael Machiavelli Marcolino  2106575@escolas.anchieta.br  **Orientador(a):**  Prof. Me. Clayton A. Valdo  [clayton.valdo@anchieta.br](mailto:clayton.valdo@anchieta.br) | INTELIGÊNCIA ARTIFICIAL EM OTIMIZAÇÕES DE PROCESSOS.  ResuMO  Este trabalho aborda a aplicação de algoritmos genéticos, uma técnica de otimização heurística, na resolução de problemas de roteirização. Focado no uso de inteligência artificial (IA) para a otimização de processos, o estudo propõe uma solução que utiliza IA para criar rotas otimizadas com base em variáveis como distância. A solução foi implementada com um algoritmo genético, integrado a uma interface amigável que exibe as rotas geradas a partir dos dados de entrada. Além disso, foram aplicados métodos de segurança baseados na OWASP, garantindo a proteção das informações e a integridade das operações realizadas na aplicação. Essa abordagem reduz erros humanos e oferece uma alternativa automatizada, eficiente, segura e escalável para desafios relacionados à roteirização.  **Palavras-Chave**: Inteligência Artificial. Otimização de Processos. Algoritmo genético. Roteirização. OWASP. |

1. Introdução

A evolução das tecnologias de Inteligência Artificial (IA) tem sido um marco significativo na história da Computação, com impactos profundos em diversas áreas. Desde a criação da máquina de Turing em 1936 até os avanços recentes, como redes neurais profundas e aprendizado por reforço, a IA tem revolucionado a abordagem de problemas complexos, proporcionando soluções inovadoras em setores como transporte, logística e otimização ­de processos (RUSSELL; NORVIG, 2021).

A tecnologia de IA apresenta um grande potencial para resolver problemas de otimização em sistemas complexos, especialmente aqueles em que as variáveis são dinâmicas e os resultados são imprevisíveis. Apesar disso, a aplicação prática da IA em cenários reais ainda enfrenta desafios significativos, principalmente em integrar modelos computacionais de forma eficaz. Esse contexto configura o problema central deste trabalho: explorar como técnicas avançadas de IA, como os algoritmos genéticos, podem contribuir para a melhoria de processos, como a roteirização (PIGOSSO et al., 2018).

O objetivo principal deste estudo é investigar a viabilidade e os benefícios da implementação de algoritmos de IA na otimização de tarefas, com foco na melhoria da eficiência e dinâmica dos processos de roteirização. A pesquisa pretende avaliar como esses algoritmos podem ser utilizados para gerar rotas mais rápidas, econômicas e eficientes, contribuindo para a gestão otimizada de recursos.

O TCC está dividido nas seguintes seções: Fundamentação teórica, onde serão abordados as metodologias, fundamentações e os dados obtidos por meio de pesquisas. Resultados e discussão, nesta etapa em será apresentado os dados obtidos durante o desenvolvimento da prova de conceito. Conclusão, seção que examina os resultados, além de analisar os próximos passos que podem ser tomados dentro das ciências da computação junto com a inteligência artificial.

1. FUNDAMENTAÇÃO teórica

Neste capítulo serão apresentadas as definições dos principais conceitos que fornecem a fundamentação do desenvolvimento deste trabalho que contextualizam nossa prova de conceito.

**2.1 Algoritmos de Otimização Heurística (Heuristics Optimization Algorithms)**

A otimização heurística é uma técnica amplamente utilizada em problemas de otimização complexos, nos quais não se tem uma solução exata ou eficiente em tempo viável devido à grande quantidade de variáveis ou restrições. Os algoritmos de otimização embora seja algo flexível e com uma grande capacidade de resolver diferentes tipos de problemas, ele não garante uma ótima solução. Em vez de buscar uma solução perfeita, os algoritmos heurísticos procuram soluções aproximadas que podem ser suficientemente boas para o problema em questão (HOLLAND, 1992).

**2.1.1 Algoritmos de Colônia de Formigas.**

Dentre os métodos utilizados, destacam-se: Algoritmos de Colônia de Formigas, no qual é um método inspirado no comportamento coletivo das formigas na natureza, mais especificamente, no modo como elas encontram caminhos eficientes entre o ninho e fontes de alimento. Esse processo é caracterizado por uma comunicação indireta entre as formigas, chamada de feromônio, que influencia as escolhas das demais formigas e é uma das bases desse algoritmo de otimização (Dorigo, M., & Di Caro, G., 1999).

Esses algoritmos podem ser aplicados em cenários de otimização de roteirização, distribuição de tarefas e gerenciamento de estoque, onde a necessidade de soluções rápidas e próximas do ótimo é essencial para o sucesso operacional. Conforme a figura 1 no qual inspirado no comportamento coletivo das formigas, esses algoritmos são eficientes para problemas de otimização de caminhos (MUTHUKRISHNAN, 2023).

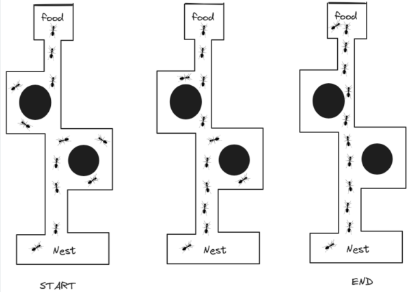


Figura 1 – Funcionamento de um algoritmo baseado em colônia de formigas (MUTHUKRISHNAN, 2023).

As formigas artificiais, em uma versão computacional, "exploram" o espaço de soluções de um problema, ou seja, elas caminham pelas possíveis soluções do problema, deixando um rastro de feromônio ao longo de seu caminho. Esse rastro é mais forte nos caminhos mais utilizados pelas formigas (Dorigo, M., & Stützle, T., 2004).

O ACO é muito utilizado em problemas de otimização combinatória, nos quais se busca encontrar a melhor solução entre várias possibilidades, como:

* Otimização de rotas (Roteirização): O algoritmo pode ser usado para encontrar o melhor caminho para veículos, como no Problema de Roteamento de Veículos (VRP), onde a tarefa é minimizar o custo de percorrer um conjunto de destinos, levando em consideração restrições como tempo, capacidade e distância (Gendreau, M., & Potvin, J. Y., 2010).
* Distribuição de Tarefas: Em sistemas multi-agentes ou em ambientes de produção, o ACO pode ser usado para distribuir tarefas de maneira eficiente entre diferentes agentes ou recursos (Moukrim, A., & Gendreau, M., 2007).

**2.2 Algoritmo Genético**

Assim como na natureza, como o exemplo da figura 2 os seres mais adaptados têm maior chance de sobreviver e deixar descendentes, no algoritmo as melhores soluções têm prioridade para gerar novas soluções (LUIGI GONTIJO, 2023).

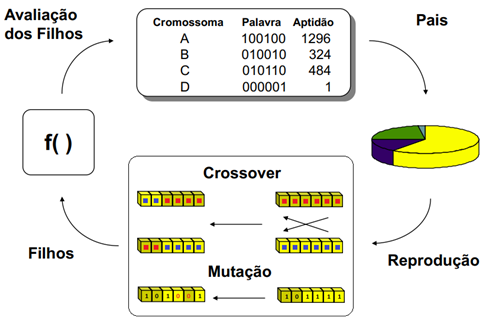


Figura 2 – Ciclo de um AG. (LUIGI GONTIJO, 2023).

É importante ressaltar que os algoritmos genéticos (AGs) não constituem um único método, mas sim uma família de algoritmos que seguem princípios gerais comuns. Cada etapa do processo pode ser implementada de maneiras distintas, o que dá origem a variações do método. Embora sejam uma simplificação dos processos biológicos naturais, os AGs têm se mostrado extremamente eficazes na busca por soluções de qualidade em uma ampla gama de problemas (HOLLAND, 1975).

Eles são capazes de explorar várias regiões do espaço de busca de forma simultânea, identificando áreas promissoras e convergindo para soluções ótimas ou próximas ao ótimo global. Essa característica torna os algoritmos genéticos ferramentas poderosas e versáteis, adequadas para a resolução de problemas complexos de otimização (GOLDBERG, 1989).

Primeiramente, cria-se um conjunto inicial de soluções aleatórias. Cada solução é avaliada e recebe uma pontuação baseada em sua qualidade, as melhores soluções são selecionadas para continuar no processo, enquanto as piores são eliminadas. As soluções escolhidas podem então ser combinadas entre si (crossover) ou sofrer pequenas alterações (mutações) para gerar novas soluções. Este ciclo se repete até encontrar uma solução satisfatória, a representação visual, da figura 2, conforme mencionada, mostra claramente como o método simula o processo de evolução natural onde as melhores soluções têm maior chance de passar suas características para as próximas gerações, enquanto novas características são introduzidas através de mutação e recombinação (MITCHELL, 1996).

O ciclo continua até que se encontre uma solução satisfatória. É importante notar que os cromossomos são representados em código binário (sequências de 0s e 1s), a aptidão é um número que quantifica a quão boa é cada solução e o processo é cíclico e contínuo, sempre gerando novas populações para que a combinação de crossover e mutação garante tanto a exploração de novas soluções quanto o aproveitamento das boas características já encontradas (GOLDBERG, 1989).

**2.3 Aprendizado por Reforço (Reinforcement Learning).**

O Aprendizado por Reforço (RL) é uma técnica de Inteligência Artificial que se baseia na interação de um agente com um ambiente dinâmico, no qual o agente toma decisões e recebe feedbacks (recompensas ou punições) que guiam o seu aprendizado. Eles aprendem com o feedback de cada ação e descobrem por si mesmos os melhores caminhos de processamento para alcançar os melhores resultados (SUTTON; BARTO, 2018).

No Aprendizado por Reforço, o agente aprende a tomar as melhores decisões em um determinado contexto, a fim de maximizar uma recompensa acumulada ao longo do tempo. Essa estratégia é especialmente útil em problemas complexos, nos quais não é possível prever todas as possíveis situações e ações, e o agente precisa adaptar-se e aprender com suas próprias experiências (SUTTON; BARTO, 2018).

**2.3.1 O processo de decisão**

Baseado no processo de decisão de Markov (MDP), que é uma modelagem matemática da tomada de decisão em um ambiente de tempo discreto, com o objetivo de aprender uma política que maximize a soma das recompensas acumuladas ao longo do tempo, considerando as probabilidades de transição entre estados. Em cada etapa do processo, o agente escolhe uma ação, o que resulta em uma transição para um novo estado do ambiente, e recebe uma recompensa associada a essa transição, como demonstrado na figura 3 (SUTTON, R. S., & BARTO, A. G. 2018).

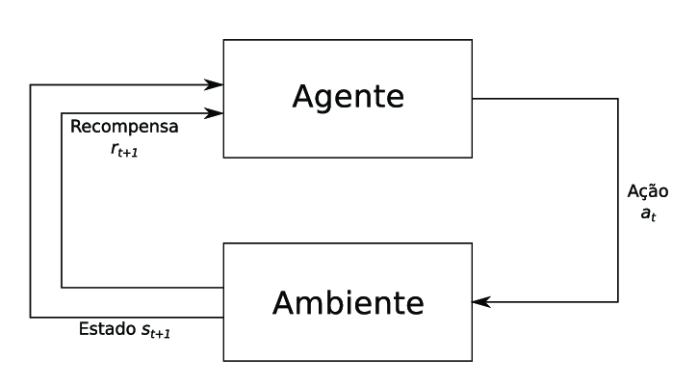


Figura 3 – Modelo de aprendizado por reforço. (EINAR SANTOS, 2019).

O MDP pode ser formalizado por uma sequência de decisões tomadas ao longo do tempo, onde:

* Estado (s): Representa a configuração atual do ambiente (SUTTON, R. S., & BARTO, A. G. 2018).
* Ação (a): A escolha que o agente faz em um estado (SUTTON, R. S., & BARTO, A. G. 2018).
* Função de Recompensa (R): A recompensa recebida após a execução de uma ação em um determinado estado (SUTTON, R. S., & BARTO, A. G. 2018).
* Política (π): A estratégia ou regra que o agente segue para decidir qual ação tomar em cada estado (SUTTON, R. S., & BARTO, A. G. 2018).
* Função de Valor (V): A medida da qualidade de um estado, que depende da recompensa que pode ser obtida ao seguir a política a partir desse estado (SUTTON, R. S., & BARTO, A. G. 2018).

**2.4 Sistemas de Gerenciamento de Armazém (WMS)**

Os Sistemas de Gerenciamento de Armazém (WMS) são softwares desenvolvidos para otimizar a gestão das operações de armazenamento e distribuição de mercadorias. Eles controlam e coordenam atividades como entrada de produtos, estocagem, picking (separação de itens) e expedição, com o objetivo de melhorar a eficiência, precisão e rastreabilidade dentro de um armazém (HARRIS, 2020).

Esses sistemas são essenciais no contexto da logística moderna, especialmente com o aumento da demanda por soluções rápidas e precisas, impulsionadas pelo comércio eletrônico e pela globalização (GAGLIARDI, 2017).

Sua implementação, aliada a tecnologias emergentes como IA e automação, pode transformar a gestão de armazéns, oferecendo uma significativa melhora na eficiência operacional, redução de custos e precisão no controle de inventário. Ao integrar essas tecnologias com algoritmos de otimização heurística e aprendizado por reforço, os WMS se tornam ainda mais poderosos, permitindo uma gestão dinâmica e adaptativa, capaz de lidar com ambientes de alta complexidade e demanda variável (CHOI, 2022).

1. Resultados e Discussão

A estratégia principal foi a implementação de uma solução baseada em inteligência artificial, especificamente um algoritmo genético. Uma aplicação que auxiliasse o usuário na roteirização, foi desenvolvido desde a interface do usuário, processamento dos dados e com toda a segurança de informação para garantir integridade dos dados, conforme a arquitetura desenhada na figura 4.

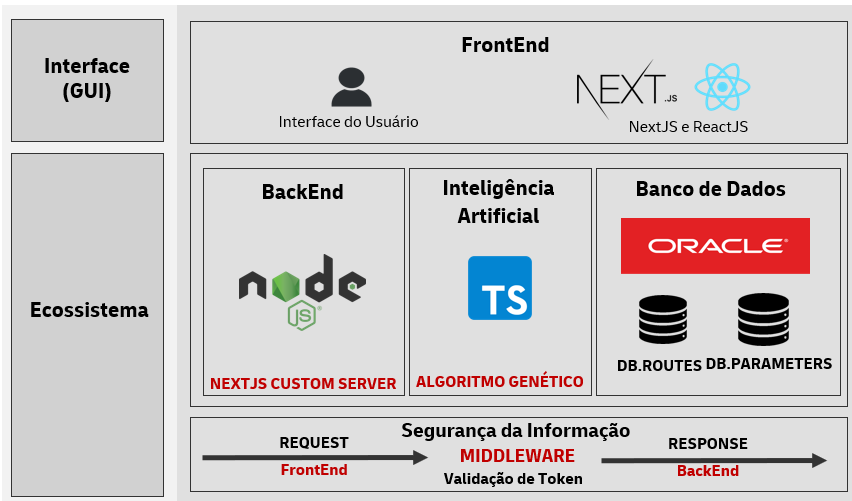


Figura 4 – Diagrama de arquitetura da aplicação.

**3.1 Interface Gráfica do Usuário (GUI)**

Com o objetivo de proporcionar uma coleta de dados eficiente e uma experiência de uso que atenda às necessidades dos usuários, foi desenvolvido um sistema web com uma interface intuitiva e amigável. A aplicação foi construída utilizando o framework Next.js, facilitando a integração e a conexão com os componentes do ecossistema da solução.

**3.1.1 Front-End**

A interface foi projetada para ser dinâmica e de fácil compreensão, permitindo que, com poucas interações, o usuário seja capaz de gerar a melhor rota. A partir dessa geração, o sistema exibe as sequências de locais a serem visitados e as distâncias entre os destinos, oferecendo uma visão clara e prática das roteirizações. A figura 5 ilustra a página inicial da aplicação, destacando suas opções de consulta e interação.

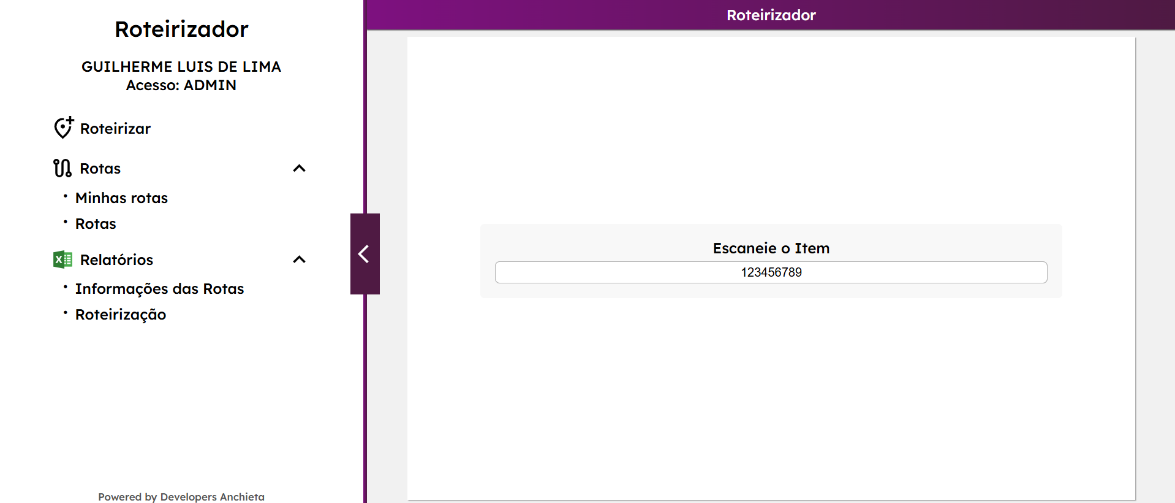


Figura 5 – Interface projetada para interação no sistema

Com todas as interfaces funcionando corretamente, a aplicação permite visualizar os dados gerados, como a exibição das rotas criadas, ilustrada na Figura 6. Nessa imagem, podemos observar as rotas geradas e os metros percorridos.

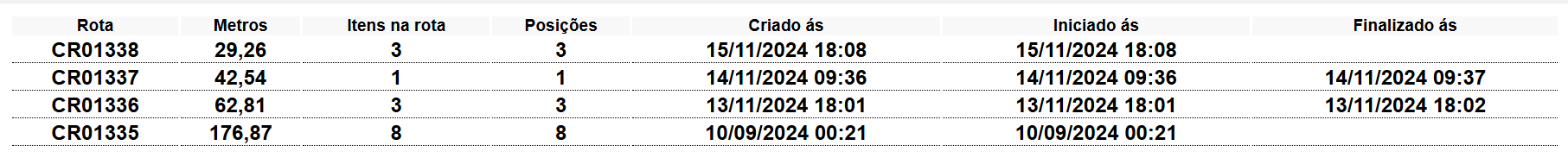


Figura 6 - Lista de rotas geradas, com distâncias e datas de criação.

Na Figura 7, a rota detalhada e ordenada conforme a sequência definida pelo algoritmo genético. Essa ordenação reflete a melhor sequência de visitas, levando em consideração as distâncias otimizadas entre os pontos.



Figura 7 – Detalhamento da rota, ordenada pela sequência de visitas definida pela IA.

**3.2 Ecossistema**

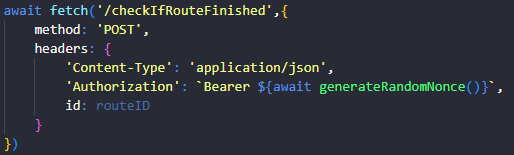
Neste tópico, é composto toda a estrutura que compõe o ecossistema da solução, destacando os elementos implementados para garantir sua estabilidade e eficiência. Serão abordados os métodos e validações aplicados para assegurar a segurança do sistema, bem como as funcionalidades projetadas para oferecer suporte confiável e garantir uma experiência de uso fluida, com renderizações otimizadas e responsivas para o usuário final.

**3.2.1 Back-End**

Com o objetivo de proporcionar a melhor experiência ao usuário, foi implementado um Custom Server utilizando o framework Next.js. Essa abordagem permite a renderização do lado do servidor (SSR) e a geração de sites estáticos (SSG), otimizando o desempenho da aplicação e reduzindo a carga de processamento nos dispositivos dos usuários.

A arquitetura também inclui um Middleware que facilita a comunicação entre o servidor e as requisições de renderização. Esse Middleware realiza verificações importantes, como a validação de cookies necessários para o acesso à aplicação, garantindo segurança e personalização no uso do sistema.

Além disso, foi utilizado a biblioteca HTTPServer para gerenciar as requisições enviadas pelo front-end, coletando informações das APIs e garantindo a integridade e segurança das interações, assim como evidenciado o exemplo de chamada de API na figura 8.

  
Figura 8 - Exemplo de chamada de API para a rota no back-end.

**3.2.2 Segurança da informação**

Para garantir a integridade das informações no sistema, foi implementado medidas robustas de segurança no aplicativo, evitando ataques e prevenindo o acesso não autorizado a dados sensíveis, sendo assim, é executado a função de validação de sessão, como a função ilustrada na figura 9 onde é executada através do back-end.

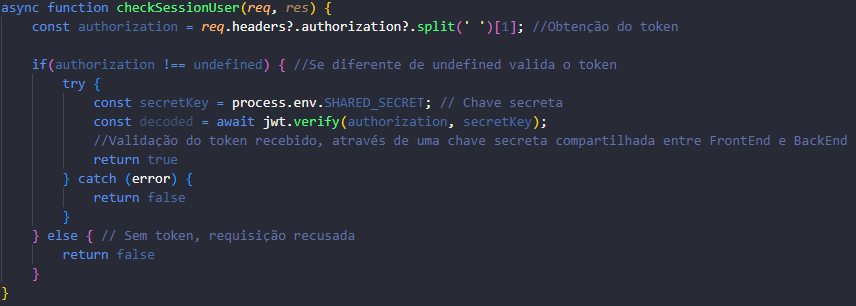


Figura 9 – Função de validação do token recebido.

A aplicação é baseada nas diretrizes da OWASP, com foco no Top 10, para mitigar vulnerabilidades críticas. Todas as requisições utilizam um Bearer Token (figura 8), validado pelo back-end, que compara o token recebido com a chave compartilhada com o front-end. Essa abordagem garante conformidade com o item A01 (Broken Access Control).

Adicionalmente, a proteção dos cabeçalhos da aplicação é assegurada por meio de um arquivo de configuração nativo do framework. Esse arquivo padroniza os headers de segurança, atendendo às exigências do item A03 (Injection) da OWASP, reforçando a proteção contra-ataques comuns (OWASP, 2024).

A validação dos cabeçalhos foi realizada por meio de ferramentas especializadas, como o SecurityHeaders, que permite verificar a implementação adequada das configurações de segurança. A Figura 10 ilustra o resultado da validação, evidenciando os headers configurados e sua conformidade com os padrões recomendados (SecurityHeaders, 2024).

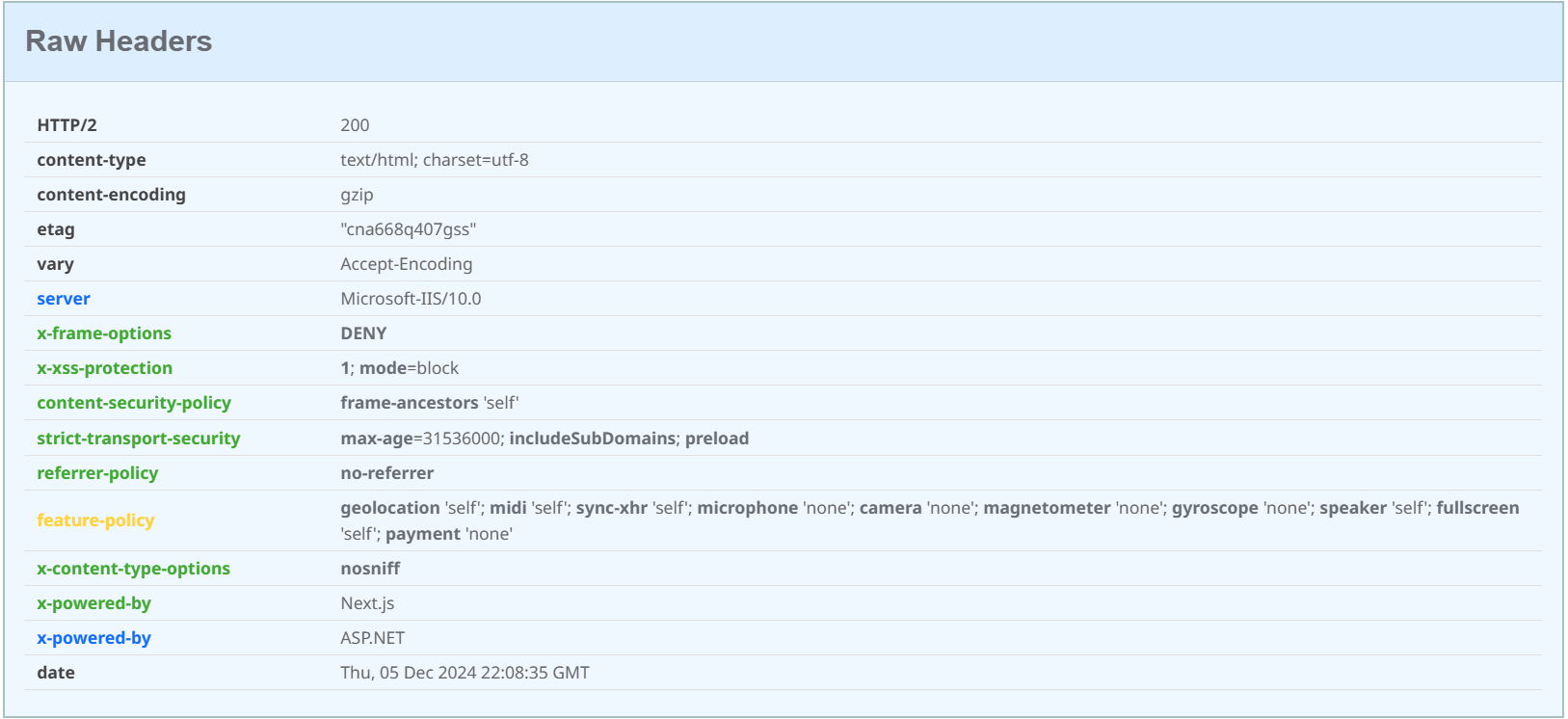


Figura 10 – Resultado extraído da ferramenta de validação de headers.

**3.2.3 Banco de Dados**

A solução foi estruturada através do OracleDB como banco de dados principal, devido à sua robustez, confiabilidade e capacidades avançadas de segurança. Essa escolha possibilitou uma integração eficiente com o back-end e o tratamento adequado de dados nas rotas de API.

Nas queries enviadas ao banco, foi utilizado parâmetros para envio de informações, aproveitando as funcionalidades nativas do OracleDB para detectar e prevenir tentativas de SQL Injection. Essa abordagem fortalece a segurança, garantindo conformidade com o item A03 (OWASP, 2024).

As tabelas do banco foram projetadas para armazenar rotas criadas e dados de usuários, priorizando uma estrutura segura e eficiente para consultas e operações.

**3.2.4 Inteligência Artificial**

Como mecanismo de inteligência artificial foi utilizado o algoritmo genético desenvolvido em TypeScript, com essa linguagem é garantido maior velocidade e eficiência no processamento das informações. O principal objetivo do algoritmo é otimizar a roteirização, avaliando rapidamente diferentes combinações de soluções possíveis e determinando a rota mais eficiente.

A lógica de funcionamento da inteligência artificial baseia-se no mapeamento espacial de cada ponto de entrega ou coleta, utilizando um plano cartesiano. Cada posição é definida por coordenadas XXX e YYY, permitindo que a fórmula de cálculo de distância euclidiana seja aplicada para medir com precisão as distâncias entre os pontos. É necessário destacar que a fórmula de distância foi definida como fitness para o AG.

Com base nas informações recebidas, a inteligência artificial inicia o processo de geração da população, essencial para o funcionamento do algoritmo genético. Nesse processo dois métodos fundamentais do AG: mutação e crossover. Esses métodos garantem a criação de um grande número de amostras e combinações possíveis, aumentando a diversidade e a qualidade das soluções.

O objetivo principal dessa etapa é realizar todas as verificações necessárias para identificar a melhor rota possível. A quantidade de cromossomos gerados é proporcional ao número de locais que o usuário precisa visitar. Cada cromossomo representa uma solução potencial, codificada de forma a refletir a sequência de pontos a serem percorridos.

A Figura 11 apresenta um exemplo da geração inicial de populações e cromossomos, demonstrando a forma como as combinações são organizadas e avaliadas durante o processo.

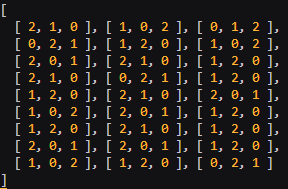


Figura 11 - População com os cromossomos gerados com base no algoritmo.

**3.3 Processo de Otimização**

O funcionamento baseia-se na coleta das informações das posições iniciais e, considerando as posições definidas para cada posição, o AG realizará a geração dos cromossomos e utilizando a fórmula de distância como fitness, para a identificação da rota mais eficiente para que conecta o locais de coletas aos locais de destino.

Essa abordagem elimina a necessidade de decisões manuais, muitas vezes suscetíveis a erros, durante as operações. Além disso, promove uma locomoção mais ágil, organizada e economicamente vantajosa, contribuindo diretamente para a eficiência operacional.

Após gerar as diferentes possibilidades de combinações, o algoritmo inicia um processo iterativo (loop) para avaliar cada cromossomo em busca da solução mais eficiente. Será a função fundamental para realizar o cálculo da distância total percorrida para cada sequência de posições selecionadas.

**3.4 Resultados**

Durante a análise, o algoritmo compara todas as distâncias obtidas e seleciona a rota mais curta como a melhor opção. A Figura 12 apresenta um exemplo prático, destacando as distâncias calculadas para diferentes cromossomos e, ao final, a melhor rota encontrada.

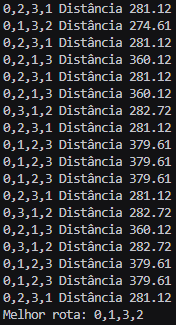


Figura 12 – Registro da distância de cada rota e a melhor rota encontrada.

A partir da análise das rotas geradas pelos cromossomos foi utilizado a fórmula para calcular o tempo aproximado para cada rota. Com isso, foi identificado a média de redução de tempo proporcionada pelo algoritmo genético ao encontrar a rota mais eficiente. Para os cálculos, foi definido uma velocidade média de 4,5 km/h, convertido para 1,25 m/s. Aplicando a fórmula, obtivemos os tempos, conforme evidenciado na tabela abaixo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rota | Metros (m) | Velocidade Média (m/s) | Tempo Médio (s) | Tempo Aproximado |
| 0,2,3,1 | 281,12 | 1,25 | 224,896 | 3 minutos e 45 segundos |
| 0,1,3,2 | 274,61 | 1,25 | 219,688 | 3 minutos e 40 segundos |
| 0,2,1,3 | 360,12 | 1,25 | 288,096 | 4 minutos e 49 segundos |
| 0,1,2,3 | 379,61 | 1,25 | 303,688 | 5 minutos e 4 segundos |
| 0,3,1,2 | 282,72 | 1,25 | 226,176 | 3 minutos e 47 segundos |
|  |  | **Redução Estimada** | **41,026 (s)** | **0 minutos e 42 segundos** |

Fonte: Elaborada pelos autores com base nos resultados do AG.

Como evidenciado, o algoritmo genético foi capaz de entregar dados relevantes e reduzir o tempo médio de roteirização em 42 segundos por rota, considerando a análise de 20 rotas geradas. Com base em pesquisas, o salário de um auxiliar de processos, que é de aproximadamente R$ 2.177,00 por mês, o valor recebido por segundo trabalhado é de R$ 0,003. Aplicando esse cálculo ao tempo reduzido, obtemos uma economia de R$ 0,13 por rota otimizada (Glassdoor, 2024).

Considerando 20 auxiliares executando pelo menos 10 rotas diárias, a economia diária alcança R$ 23,09, o que representa um potencial de R$ 60.956,00 por ano e 6,1 horas. Esses resultados demonstram a eficiência do algoritmo em gerar otimizações que impactam diretamente nos custos da empresa.

1. Conclusão

Com base nas análises realizadas e nos resultados alcançados, ficou evidente que os objetivos propostos foram atendidos. A utilização da Inteligência Artificial (IA) na otimização de processos demonstrou ser uma estratégia eficaz, com impactos significativos na eficiência e reduções de custo.

A implementação de algoritmos genéticos para a roteirização e sequenciamento de tarefas foi um diferencial importante, contribuindo para a redução de deslocamentos e tempos de execução. Esses avanços resultaram em benefícios, como a diminuição de custos, aumento da velocidade dos processos e uma maior confiabilidade na execução das rotas.

Além disso, a solução proposta foi além das expectativas técnicas, promovendo maior alinhamento estratégico. O Algoritmo Genético não apenas otimizou fluxos, mas também capacitou os usuários a realizar tarefas de forma mais eficaz, consolidando a confiança na tecnologia como uma ferramenta transformadora.  
Como extensão deste estudo, sugere-se explorar novas aplicações de IA com a implementação de novos algoritmos de otimização heurística como algoritmos de colônia de formigas e algoritmo genético em outros processos, como na separação de mercadorias e na gestão de estoques. Essas iniciativas podem desbloquear ganhos adicionais em produtividade e qualidade, reforçando ainda mais o papel estratégico da tecnologia na transformação digital dos fluxos operacionais.

Essa abordagem reafirma o potencial da IA como uma aliada essencial para alcançar maiores níveis de eficiência e competitividade das organizações.

Referências

BARTHOLDI, J. J. Warehouse & Distribution Science. 2. ed. Florida: Warehouse & Distribution Science, 2020. Disponível em: https://www.warehouse-science.com/book/editions/wh-sci-0.98.1.pdf. Acesso em: 22 out. 2024.

BOWERSOX, D. J.; CLOSS, D. J.; COOPER, M. B. Supply Chain Logistics Management. 4. ed. McGraw-Hill, 2013. Disponível em: https://archive.org/details/supplychainlogis00dona/page/n13/mode/2up. Acesso em: 22 out. 2024.

COSTA, L. H. et al. Inteligência Artificial e Otimização de Processos Computacionais: Aplicação em Sistemas de Gestão de Tarefas. In: Simpósio Brasileiro de Pesquisa Operacional, 2022. Disponível em: https://www.sbpog.org.br. Acesso em: 28 nov. 2024.

DORIGO, M.; DI CARO, G. The Ant Colony Optimization Metaheuristic. Proceedings of the 1999 Congress on Evolutionary Computation. 1999.

GAGLIARDI, G. R. Warehouse management: a complete guide to improving efficiency and minimizing costs in the modern warehouse. 3. ed. London: Kogan Page, 2017. 448 p. Disponível em: https://search.worldcat.org/pt/title/warehouse-management-a-complete-guide-to-improving-efficiency-and-minimizing-costs-in-the-modern-warehouse/oclc/1009176768. Acesso em: 29 nov. 2024.

GAMBARDELLA, L. M.; DORIGO, M. Solving asymmetric and Euclidean TSPs by Ant Colony System. In Proceedings of the European Conference on Artificial Intelligence. 2000.

GLASSDOOR. Salários de Auxiliar De Processos. Disponível em: https://www.glassdoor.com.br/Sal%C3%A1rios/auxiliar-de-processos-sal%C3%A1rio-SRCH\_KO0,21.htm. Acesso em: 04 dez, 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. Cambridge, MA: MIT Press, 2016. Disponível em: https://www.deeplearningbook.org. Acesso em: 28 nov. 2024.

HOLLAND, J. H. Adaptation in Natural and Artificial Systems. MIT Press, 1992. Disponível em: https://archive.org/details/adaptation-in-natural-and-artificial-systems. Acesso em: 28 nov. 2024.

HOLLAND, J. H. Genetic Algorithms. Scientific American, v. 267, n. 1, p. 66-72, 1992. Disponível em: https://www.scientificamerican.com. Acesso em: 22 out. 2024.

OWASP. HTTP Headers Cheat Sheet. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/HTTP\_Headers\_Cheat\_Sheet.html. Acesso em: 29 nov. 2024.

PIGOSSO, D. C. A. et al. Ecodesign Maturity Model: A Management Framework to Support Ecodesign Implementation into Manufacturing Companies. Journal of Cleaner Production, v. 59, p. 160-173, 2018. Disponível em: https://doi.org/10.1016/j.jclepro.2013.07.049. Acesso em: 28 nov. 2024.

PIGOSSO, D. C. A.; PINA, A. C. S.; CAVALIERI, S. A.; PIROLI, R. P. An Approach for the Integration of Sustainability in Manufacturing Systems: The Role of Artificial Intelligence. Springer, 2018. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-63416-0\_12. Acesso em: 28 nov. 2024.

RUSSELL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2021. Disponível em: https://www.amazon.com.br/Artificial-Intelligence-Modern-Approach-4th/dp/0134610997. Acesso em: 28 nov. 2024.

RUSSELL, S.; NORVIG, P. Inteligência Artificial: Estruturas e Estratégias para Solução Complexa de Problemas. 3. ed. Pearson, 2021. Disponível em: Acesso em: 28 nov. 2024.

SCHWAB, K. The Fourth Industrial Revolution. Penguin UK, 2016. Disponível em: https://www.penguin.co.uk/books/314/314779/the-fourth-industrial-revolution/9780241283916. Acesso em: 28 nov. 2024.

SECURITYHEADERS. Probely Snyk Business. Disponível em: https://securityheaders.com/. Acesso em: 03 nov. 2024.

SILVA, M. R. L.; REIS, L. P.; MARTINS, A. P. Inteligência Artificial e Computação Evolutiva: Fundamentos e Aplicações. Ed. Ciência Moderna, 2020. Disponível em: https://www.cienciasmoderna.com.br. Acesso em: 28 nov. 2024.

SOUZA, João. Tipos de aprendizado de máquina e algumas aplicações. 2024. Disponível em: https://www2.decom.ufop.br/terralab/tipos-de-aprendizado-de-maquina-e-algumas-aplicacoes/. Acesso em: 29 nov. 2024.

STÜTZLE, T.; DORIGO, M. A Short Note on the Asymptotic Behavior of Ant Colony Optimization Algorithms. Theoretical Computer Science, v. 200, n. 1, p. 263-283, 2002.

SUTTON, R. S.; BARTO, A. G. Reinforcement Learning: An Introduction. 2. ed. MIT Press, 2018. Disponível em: https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf. Acesso em: 28 nov. 2024.