

# ORIENTAÇÕES

## Estudo de caso



### 1. TÍTULO:

Estudo de Caso em Rust: Implementação de um Sistema de Recomendação de Produtos Utilizando Grafos

### 2. CENÁRIO:

#### Cenário

A "MegaStore", uma gigante do varejo com uma vasta operação de e-commerce, oferece milhões de produtos em seu catálogo online, desde eletrônicos e vestuário até itens de decoração e alimentos. Com um catálogo tão extenso, a empresa enfrenta um desafio crucial: garantir que os clientes encontrem os produtos que desejam de forma rápida e eficiente.

#### O Problema

A "MegaStore" percebeu que seu sistema de busca atual, baseado em métodos tradicionais, não está atendendo às expectativas dos clientes. As buscas estão lentas, imprecisas e muitas vezes retornam resultados irrelevantes. Isso leva à frustração dos clientes, que abandonam o site sem encontrar o que procuram, resultando em perda de vendas e prejudicando a reputação da empresa.

#### A Solução

# ORIENTAÇÕES

## Estudo de caso



Para solucionar esse problema, a "MegaStore" decidiu investir na otimização do sistema de busca de seu catálogo de produtos. A empresa busca uma solução que seja capaz de:

- Indexar o catálogo de produtos de forma eficiente: Utilizar uma estrutura de dados que permita o acesso rápido aos produtos, com base em diferentes critérios de busca, como nome, marca, categoria, etc.
- Realizar buscas rápidas e precisas: Implementar algoritmos de busca que encontrem os produtos relevantes em tempo hábil, mesmo em um catálogo com milhões de itens.
- Lidar com grandes volumes de dados: Garantir que o sistema de busca seja escalável e capaz de lidar com o crescimento contínuo do catálogo de produtos da "MegaStore".

### A Oportunidade

Com a implementação de um sistema de busca otimizado, a "MegaStore" poderá oferecer uma experiência de compra superior aos seus clientes, o que resultará em:

- Aumento da taxa de conversão de vendas: Clientes satisfeitos encontram os produtos que desejam com mais facilidade, o que os leva a finalizar a compra.
- Melhora na fidelização dos clientes: Uma experiência de compra positiva aumenta a probabilidade de os clientes voltarem a comprar na "MegaStore".
- Fortalecimento da marca: Um sistema de busca eficiente contribui para a imagem de uma empresa moderna e orientada para o cliente.

# ORIENTAÇÕES

## Estudo de caso



### 3. DESAFIO:

#### O Desafio

A "MegaStore" precisa implementar o sistema de busca de forma eficiente e escalável, garantindo que ele seja capaz de lidar com o grande volume de dados e o crescente número de clientes da empresa. Além disso, a empresa busca uma solução que seja segura e confiável, protegendo os dados dos clientes e garantindo a privacidade das informações.

#### O Papel dos Alunos

Os alunos da disciplina de Data Structures Strategy and Implementation são desafiados a desenvolver e implementar o sistema de busca otimizado para o catálogo de produtos da "MegaStore". Para isso, eles deverão utilizar seus conhecimentos sobre estruturas de dados, algoritmos e programação em Rust para criar uma solução eficiente, escalável e segura.

### 4. TEORIA NA PRÁTICA:

#### Entregas

- **Implementação do sistema de busca:** Os alunos deverão implementar o sistema de busca utilizando a linguagem de programação Rust, explorando os recursos da linguagem para garantir a segurança de memória e o desempenho do sistema.

# ORIENTAÇÕES

## Estudo de caso



- **Relatório técnico:** Os alunos deverão elaborar um relatório técnico detalhado, descrevendo a solução implementada, os desafios enfrentados e os resultados obtidos.
- **Apresentação:** Os alunos deverão apresentar o sistema de busca para a equipe da "MegaStore", demonstrando seu funcionamento e explicando as decisões de projeto.

### Critérios de Avaliação

- **Eficiência:** O sistema de busca deverá ser eficiente em termos de tempo de resposta e utilização de recursos computacionais.
- **Escalabilidade:** O sistema de busca deverá ser escalável para lidar com o crescimento do número de produtos e clientes da "MegaStore".
- **Segurança:** O sistema de busca deverá garantir a segurança dos dados dos clientes e a privacidade das informações.
- **Qualidade do código:** O código implementado deverá seguir as boas práticas de programação em Rust e ser bem documentado.
- **Qualidade do relatório técnico:** O relatório técnico deverá ser claro, conciso e completo, descrevendo todos os aspectos relevantes do projeto.
- **Qualidade da apresentação:** A apresentação deverá ser clara, objetiva e convincente, demonstrando o domínio dos alunos sobre o tema.

Este é o cenário do estudo de caso. Com base nessas informações, os alunos podem começar a trabalhar no desenvolvimento do sistema de busca otimizado para o catálogo de produtos da "MegaStore".

# ORIENTAÇÕES

## Estudo de caso



### 5. LEVANTAMENTO DE SOLUÇÕES:

Com o objetivo de auxiliar os alunos no levantamento de soluções para o estudo de caso da "MegaStore", propomos as seguintes questões norteadoras, que visam estimular a reflexão e o aprofundamento no problema:

#### Questões para Explorar o Problema

1. **Quais são as principais causas da lentidão e imprecisão do sistema de busca atual da "MegaStore"?**
  - Esta questão convida os alunos a analisar o sistema de busca existente, identificando gargalos e pontos de melhoria. É importante considerar fatores como a estrutura de dados utilizada, os algoritmos de busca implementados e a escalabilidade do sistema.
2. **Quem são os principais afetados pela baixa performance do sistema de busca e como eles são afetados?**
  - Compreender o impacto do problema nos diferentes atores envolvidos (clientes, funcionários da "MegaStore", etc.) é fundamental para dimensionar a importância da solução e definir os critérios de avaliação.
3. **Quais são as possíveis soluções para otimizar o sistema de busca da "MegaStore" e quais são os prós e contras de cada uma?**
  - Esta questão centraliza o objetivo do estudo de caso, incentivando os alunos a explorar diferentes abordagens para solucionar o problema. É crucial que sejam consideradas soluções que envolvam o uso de tabelas hash em Rust, explorando seus benefícios para otimização de buscas.

# ORIENTAÇÕES

## Estudo de caso



4. **Quais são as barreiras ou desafios para implementar cada uma das soluções propostas?**
  - A implementação de qualquer solução envolve desafios técnicos, como a escolha da função hash ideal, o tratamento de colisões e a garantia da escalabilidade do sistema. É importante que os alunos ponderem esses desafios ao avaliar as diferentes soluções.
5. **Como a utilização de tabelas hash em Rust pode contribuir para a otimização do sistema de busca da "MegaStore"?**
  - Esta questão direciona o foco para a tecnologia central do estudo de caso, incentivando os alunos a pesquisarem e compreenderem as vantagens do uso de tabelas hash em Rust para indexar e buscar dados de forma eficiente.
6. **Quais são os requisitos de hardware e software necessários para implementar a solução escolhida?**
  - A implementação de um sistema de busca otimizado pode exigir investimentos em infraestrutura. É importante que os alunos considerem esses aspectos ao propor suas soluções.
7. **Como a solução proposta se integra com o restante da plataforma de e-commerce da "MegaStore"?**
  - A integração com os sistemas existentes é um aspecto crucial a ser considerado. Os alunos devem analisar como a solução de busca se encaixa na arquitetura da plataforma e como ela interage com outros componentes, como o banco de dados de produtos e o sistema de recomendação.

# ORIENTAÇÕES

## Estudo de caso



### 8. Quais são os indicadores de desempenho (métricas) que podem ser utilizados para avaliar a eficácia da solução implementada?

- Definir métricas claras e objetivas é fundamental para avaliar o sucesso da solução. Os alunos devem propor indicadores que permitam medir o tempo de resposta das buscas, a precisão dos resultados, a escalabilidade do sistema e a satisfação dos clientes.

### 9. Quais são as implicações a longo prazo para a "MegaStore" caso o problema do sistema de busca não seja resolvido?

- Esta questão visa conscientizar os alunos sobre a importância de solucionar o problema, mostrando como a baixa performance do sistema de busca pode impactar negativamente a reputação da empresa, a fidelização dos clientes e o desempenho das vendas.

## Sugestão de Soluções (Mínimo 3)

Com base nas questões norteadoras, os alunos podem propor diferentes soluções para otimizar o sistema de busca da "MegaStore". Abaixo, apresentamos algumas sugestões, que podem ser exploradas e adaptadas:

#### 1. Implementação de um índice de produtos utilizando tabelas hash em Rust:

- Esta solução consiste em criar um índice de produtos na memória principal, utilizando tabelas hash para mapear os termos de busca (palavras-chave) para os IDs dos produtos correspondentes. Essa abordagem permite realizar buscas rápidas e precisas, pois o acesso aos produtos é feito

# ORIENTAÇÕES

## Estudo de caso



diretamente através da tabela hash, sem a necessidade de percorrer grandes estruturas de dados.

### **2. Utilização de técnicas de pré-processamento de texto para otimizar as buscas:**

- Esta solução visa melhorar a relevância dos resultados da busca, aplicando técnicas de pré-processamento de texto aos termos de busca e aos dados dos produtos. Essas técnicas podem incluir a remoção de palavras irrelevantes (stop words), a normalização de palavras (lematização e stemming) e a correção de erros de ortografia.

### **3. Implementação de um sistema de cache para armazenar os resultados de buscas frequentes:**

- Esta solução visa reduzir o tempo de resposta das buscas, armazenando em cache os resultados de buscas frequentes. Quando um usuário realiza uma busca que já foi realizada anteriormente, o sistema pode retornar o resultado diretamente do cache, sem a necessidade de acessar o índice de produtos.

## **Observações**

- As soluções apresentadas são apenas sugestões e podem ser combinadas ou adaptadas de acordo com as necessidades do estudo de caso.
- É importante que os alunos justifiquem a escolha da solução, apresentando argumentos técnicos e demonstrando o conhecimento sobre as estruturas de dados e algoritmos envolvidos.



# ORIENTAÇÕES

## Estudo de caso



- Os alunos devem levar em consideração os critérios de avaliação ao propor suas soluções, buscando um equilíbrio entre eficiência, escalabilidade, segurança e qualidade do código.

### 5. ENTREGÁVEL:

O entregável deste estudo de caso será um repositório GitHub contendo o sistema de busca otimizado para o catálogo de produtos da "MegaStore", implementado na linguagem de programação Rust e um PDF contendo toda a documentação desenvolvida. O repositório deverá seguir as seguintes diretrizes:

#### Estrutura do Repositório

- **src/**: Contém o código fonte do sistema de busca, organizado em módulos e arquivos Rust.
- **tests/**: Contém os testes unitários e de integração para o sistema de busca, garantindo a qualidade e a confiabilidade do código.
- **Cargo.toml**: Arquivo de configuração do projeto Rust, contendo as dependências e informações sobre o projeto.
- **README.md**: Arquivo de documentação do projeto, contendo instruções de como executar o sistema de busca, os testes e exemplos de uso.

#### Conteúdo do README.md

O arquivo README.md deve conter as seguintes informações:

- **Título do projeto**: "Sistema de Busca Otimizado para Catálogo de Produtos - MegaStore"
- **Descrição do projeto**: Uma breve descrição do sistema de busca, seus objetivos e funcionalidades.
- **Tecnologias utilizadas**: Lista das tecnologias utilizadas no projeto, incluindo Rust, crates (bibliotecas) e ferramentas de teste.
- **Instruções de como executar o sistema de busca**: Passo a passo de como compilar e executar o sistema de busca, incluindo os comandos Rust necessários.
- **Instruções de como executar os testes**: Passo a passo de como executar os testes

# ORIENTAÇÕES

## Estudo de caso



unitários e de integração, incluindo os comandos Rust necessários.

- Exemplos de uso: Exemplos de como utilizar o sistema de busca, incluindo diferentes tipos de consultas e filtros.
- Arquitetura do sistema: Uma descrição da arquitetura do sistema, incluindo os módulos e componentes principais.
- Algoritmos e estruturas de dados utilizados: Uma descrição dos algoritmos e estruturas de dados utilizados na implementação do sistema de busca, com ênfase na utilização de tabelas hash.
- Considerações sobre desempenho e escalabilidade: Uma análise do desempenho e da escalabilidade do sistema de busca, incluindo os resultados dos testes de desempenho.
- Contribuições: Informações sobre como contribuir para o projeto, caso seja um projeto colaborativo.
- Licença: Informações sobre a licença do projeto.

### Critérios de Avaliação

- Funcionalidade: O sistema de busca deve funcionar corretamente, indexando o catálogo de produtos e realizando buscas rápidas e precisas.
- Desempenho: O sistema de busca deve apresentar bom desempenho em termos de tempo de resposta e utilização de recursos computacionais.
- Escalabilidade: O sistema de busca deve ser escalável para lidar com o crescimento do número de produtos e clientes da "MegaStore".
- Qualidade do código: O código implementado deve seguir as boas práticas de programação em Rust e ser bem documentado.
- Qualidade dos testes: Os testes unitários e de integração devem cobrir as principais funcionalidades do sistema de busca e garantir a qualidade do código.
- Qualidade da documentação: O arquivo README.md deve ser claro, conciso e completo, descrevendo todos os aspectos relevantes do projeto.

### Observações

- O repositório GitHub deve ser público para que os avaliadores possam acessar o código e a documentação.
- Os alunos podem utilizar crates (bibliotecas) Rust para implementar o sistema de busca, mas devem garantir que as dependências estejam listadas no arquivo Cargo.toml.
- Os alunos são incentivados a utilizar ferramentas de teste Rust, como **cargo test**, para garantir a qualidade do código.

# ORIENTAÇÕES

## Estudo de caso



### Entrega

- link para o repositório GitHub contendo o sistema de busca otimizado e toda a documentação necessária. Os alunos deverão enviar o link do repositório.
- PDF contendo toda a documentação desenvolvida.

Este formato de entrega permite que os alunos demonstrem suas habilidades de programação em Rust, conhecimento sobre estruturas de dados e algoritmos, e capacidade de documentar e apresentar seus projetos de forma clara e organizada. Além disso, a utilização do GitHub facilita o compartilhamento e a colaboração entre os alunos.

# ORIENTAÇÕES

## Estudo de caso



### Roteiro do estudante

#### 1. Leia o cenário:

Sua primeira tarefa é se tornar um detetive. Use os recursos que fornecemos, mas também não tenha medo de sair em busca de suas próprias informações adicionais. Queremos que você compreenda o tópico e explore o problema a fundo.

#### 2. Encare o desafio:

Muita atenção para não perder o foco durante o estudo. Você precisa compreender qual é o desafio para não perder isso de vista durante todo o processo.

#### 3. Teoria na prática:

Este é o momento de pesquisar o que já existe no mercado e ler todas as indicações que o professor fizer. Afinal, antes de pensar em resolver o desafio, é preciso reunir as ferramentas necessárias e reconhecer o que já existe no mercado de trabalho da sua profissão para lidar com esse tipo de situação. Não esqueça de trazer um exemplo concreto de uma solução já existente.

#### 4. Levante hipóteses para resolver o problema:

Depois de reunir todas as suas informações, é hora de se tornar um(a) verdadeiro(a) analista. **Busque identificar padrões, relações, causas e efeitos.** Procure contradições ou conflitos de informação. **Seja crítico** e **questione** o que você encontrou durante sua investigação.

#### 5. Aponte a melhor solução:

Agora é o momento de se tornar um(a) solucionador(a) de problemas. Com base no que você descobriu e analisou, dentre as hipóteses, qual você escolheu como uma solução para o problema? Chegou o momento de explicar em detalhes: por que essa é a melhor solução neste cenário? Use seus conhecimentos para argumentar.

# ORIENTAÇÕES

## Estudo de caso



### **6. Crie um plano de ação:**

Depois de desenvolver sua solução, é hora de mostrar a viabilidade e a aplicabilidade dela. Explique em detalhes como o problema será resolvido, o passo a passo, em quanto tempo e quais as implicações disso. Lembre-se de que sua comunicação deve ser clara e eficaz para que todos entendam sua proposta. Nesta etapa, você terá que apresentar sua solução. Use a sua criatividade para apresentá-la da melhor maneira possível.

# ORIENTAÇÕES

# Estudo de caso



## **Como seu trabalho será avaliado?**

Sua avaliação será baseada em uma série de critérios, com uma pontuação total de até 4,0 pontos.

Lembre-se, o objetivo aqui não é apenas obter a pontuação máxima, mas aprender durante o processo. Aproveite essa jornada!