

## **Sistema de Busca Otimizado para Catálogo de Produtos – MegaStore**

### **Autor:**

- Nome: Jorge T.U
- RA:\*\*\*\*\*
- Curso: Análise e Desenvolvimento de Sistemas

### **1. Introdução**

A "MegaStore", uma gigante do varejo, enfrenta dificuldades com seu sistema de busca para um catálogo extenso contendo milhões de produtos. O sistema atual apresenta lentidão e resultados imprecisos, prejudicando a experiência do usuário e as conversões de vendas.

#### **Objetivo do Projeto**

Desenvolver um sistema de busca otimizado, eficiente e escalável, utilizando Rust e estruturas de dados avançadas como tabelas hash e grafos. A solução visa melhorar a precisão das buscas, reduzir o tempo de resposta e oferecer recomendações personalizadas para os usuários.

---

### **2. Arquitetura do Sistema**

O sistema consiste em três módulos principais:

1. **Índice de Produtos:**
    - Implementado com tabelas hash para permitir acesso rápido aos produtos com base em critérios como categoria e marca.
  2. **Sistema de Cache:**
    - Armazena resultados frequentes para reduzir o tempo de resposta e otimizar o desempenho.
  3. **Grafo de Recomendações:**
    - Utiliza um modelo baseado em grafos para relacionar produtos frequentemente acessados ou comprados juntos.
- 

### **3. Estruturas de Dados e Algoritmos**

#### **Tabelas Hash**

- Permitem acesso direto aos produtos com base em termos de busca, garantindo eficiência em grandes volumes de dados.
- Implementadas para indexar produtos por categoria, marca e ID.

#### **Grafo de Recomendação**

- Modelado como um HashMap onde cada nó representa um produto e cada aresta representa uma relação com outros produtos.
- Exemplo: um "Smartphone" é conectado a "Capas de Celular" e "Carregadores" devido à alta relevância conjunta.

#### **Cache**

- Usado para armazenar resultados das buscas mais frequentes. Isso evita consultas repetitivas ao índice e melhora o desempenho geral.

## 4. Desafios Enfrentados

### 1. Problemas de Lifetimes no Rust:

- Resolver conflitos de referências entre o índice de produtos e o cache foi essencial para a implementação.

### 2. Tratamento de Categorias Vazia:

- Ajustar o sistema para evitar armazenamento no cache de resultados vazios.

### 3. Escalabilidade:

- Garantir que o sistema seja capaz de lidar com milhões de produtos sem perda de desempenho.
- 

## 5. Resultados Obtidos

### Eficiência

- Buscas rápidas mesmo em um catálogo extenso.
- Resultados precisos para consultas específicas, como filtros por categoria e marca.

### Escalabilidade

- O uso de tabelas hash e grafos garantiu que o sistema pudesse lidar com o crescimento contínuo do catálogo e número de clientes.

### Recomendações

- Sugestões personalizadas com base no produto consultado, aumentando a probabilidade de conversão de vendas.
- 

## 6. Instruções de Uso

### Como Executar o Sistema

1. Clone o repositório GitHub:
2. git clone < [https://github.com/JorgeToshio/busca\\_megastore.git](https://github.com/JorgeToshio/busca_megastore.git) >
3. cd < megastore\_search >
4. Compile o projeto:
5. cargo build
6. Execute o sistema:
7. cargo run

### Como Executar os Testes

1. Certifique-se de estar na pasta do repositório.
  2. Rode os testes:
  3. cargo test
-

## 7. Considerações de Desempenho

- **Tempo de resposta:** A execução é rápida devido ao uso de tabelas hash e cache.
  - **Precisão:** As buscas retornam produtos altamente relevantes para os critérios definidos.
  - **Escalabilidade:** Testado com grandes volumes de dados para garantir desempenho consistente.
- 

## 8. Conclusão

O sistema de busca desenvolvido para a "MegaStore" resolve os problemas enfrentados pela empresa, oferecendo:

- Buscas rápidas e precisas.
- Recomendações relevantes e personalizadas.
- Escalabilidade para lidar com o crescimento contínuo do catálogo.

Com essa solução, a "MegaStore" pode melhorar a experiência do cliente, aumentar a taxa de conversão e fortalecer sua marca.

---

## 9. Referências

- Documentação oficial de Rust: <https://www.rust-lang.org/>
  - Boas práticas em estruturas de dados e algoritmos.
  - Artigos acadêmicos sobre tabelas hash e grafos.
- 

## 10. Link do Repositório

Certifique-se de incluir o link do repositório GitHub no PDF:

Link do Repositório: [https://github.com/JorgeToshio/busca\\_megastore.git](https://github.com/JorgeToshio/busca_megastore.git)