

## Contexto

Uma varejista de médio porte opera uma loja online com milhares de produtos. A busca é um dos principais motores de receita: quando o cliente não encontra itens rapidamente, ele abandona a compra. A empresa tem vários consumidores internos e externos que precisam de busca:

- o storefront público (site e app)
- ferramentas de atendimento ao cliente (encontrar produtos para ajudar consumidores)
- ferramentas de merchandising (gerenciar promoções e “descoberta” de produtos)

Times diferentes implementaram funcionalidades de busca em lugares diferentes, o que levou a filtros inconsistentes, diferenças estranhas de ranqueamento e bugs repetidos em múltiplos sistemas. A empresa quer um **módulo de busca compartilhado** que encapsule a semântica da busca e consiga evoluir com segurança.

## Conceitos centrais do domínio

Um **Produto** é um item vendável. Cada produto possui:

- **SKU/ProductId** (identidade estável)
- **Nome**
- **Descrição** (opcional)
- **Marca**
- **Categorias** (uma ou mais)
- **Preço** (pode mudar; pode haver preço promocional)
- **Disponibilidade** (em estoque / fora de estoque / sob encomenda)
- **Atributos** (tamanho, cor, material etc., dependendo do tipo de produto)
- **Avaliações** (nota média + quantidade, opcional)

## Necessidades dos clientes (tarefas típicas)

Clientes geralmente querem:

- buscar por intenção: “mouse sem fio”, “tênis de corrida”, “moedor de café”
- filtrar por categoria, marca, faixa de preço, tamanho/cor, nota mínima
- excluir itens fora de estoque (ou ao menos mostrá-los com menor prioridade)
- ordenar por:
  - relevância (padrão)
  - preço (crescente/decrescente)
  - lançamentos/novidades
  - avaliação
- ver facetar/contagens para refinar resultados (ex.: “Nike (120), Adidas (95)”)
  - 
  - 
  - 
  -
- lidar com sinônimos e variações de escrita (ex.: “tv” vs “televisão”, “fone” vs “headset”)

## Requisitos funcionais

Seu módulo deve suportar:

### A) Indexação

- Adicionar/atualizar dados de produto (incluindo mudanças de preço e disponibilidade).
- Remover produtos descontinuados.
- Reconstruir o índice (reindexação).
- Suportar atualizações frequentes (disponibilidade/preço podem mudar várias vezes ao dia).

### B) Busca

- Consulta por palavras-chave em nome/descrição/marca/rótulos de categoria.
- Filtros:
  - categoria (uma ou mais)

- marca
- faixa de preço
- disponibilidade (somente em estoque vs incluir todos)
- atributos (tamanho/cor etc.)
- nota mínima (avaliação)
- Ordenação:
  - padrão por relevância
  - opcional por preço, lançamentos/novidades, avaliação
- Resultados devem incluir:
  - ProductId/SKU
  - campos para exibição (nome, preço, disponibilidade, categoria principal e/ou marca)
  - um artefato de explicação (ex.: campos que casaram ou um snippet)

**Opcional, mas desejável: facetas** para categoria/marca/faixas de preço.

# Entregáveis

## 1) Linguagem Ubíqua (Glossário)

Um documento curto com:

- lista de termos do domínio (ex.: “Publicação”, “Autor”, “Consulta”, “Filtro”, “Relevância”...)
- definição clara (1–2 frases)
- exemplo de uso (1 frase ou 1 caso)

## 2) Subdomínios e Bounded Context

Um diagrama simples (pode ser desenho) indicando:

- quais partes são **core** / **supporting** / **generic** (justificativa curta)
- qual é o **Bounded Context de Busca** (Search)
- o que está **dentro** e **fora** do contexto de Busca

## 3) Mapa de Contexto (Context Map)

Um desenho mostrando:

- quem consome o módulo de busca (portal, backoffice, etc.)
- de onde vêm os dados (catálogo/publicações)
- como a busca se integra (ex.: eventos, sincronização, chamadas diretas)

## 4) Modelo Tático (Tactical DDD)

Um diagrama (ou lista estruturada) contendo:

- **Entidades** (com identidade)
- **Regras/invariantes** importantes

## 5) Esboço da API pública (em linguagem do domínio)

Um documento com:

- nomes de operações
- tipos de entrada e saída **em termos do domínio** (não em termos de tecnologia)
- 2–3 exemplos de uso (pseudo-código)