

5.2 - Guia de apoio: Configurações Avançadas do Smart Finder

Sobre:

Guia de apoio: Configurações Avançadas do Smart Finder

Esta documentação complementa a página de **Configurações Avançadas do Smart Finder** e explica os principais campos técnicos apresentados na tela.

O **Smart Finder** utiliza o **Elasticsearch** como motor de busca da plataforma. Ele é responsável por consultar os produtos indexados, identificar correspondências com o texto pesquisado e calcular a relevância utilizada para ordenar os resultados.

As Configurações Avançadas permitem controlar diretamente campos, pesos e parâmetros utilizados nesse processamento. A estrutura disponibilizada contempla **Query Fields**, **Query Boosts** e **Extra Params** nos contextos Pesquisa, Sugestão, Categoria, Marca e Lista de Produtos.

***Atenção:** alterações incorretas podem reduzir a relevância, aumentar as buscas sem resultados ou fazer produtos pouco relacionados aparecerem nas primeiras posições. Para ajustes cotidianos, utilize preferencialmente as Configurações Básicas.*

Como o Smart Finder processa uma pesquisa

Quando o consumidor pesquisa, por exemplo:

tênis de corrida azul

o Smart Finder precisa definir:

- quais campos dos produtos serão consultados;
- quanto cada campo influenciará o resultado;
- se todas as palavras precisam ser encontradas;
- se erros de digitação serão tolerados;
- se as palavras precisam aparecer juntas;
- como caracteres especiais serão tratados;
- como as diferentes correspondências serão combinadas.

Essas definições são controladas principalmente por três grupos de configuração:

Grupo	Finalidade
Query Fields	Define os campos pesquisados.
Query Boosts	Define regras e pesos de relevância.
Extra Params	Inclui parâmetros adicionais na consulta.

***Dica:** as abas Pesquisa, Sugestão, Categoria, Marca e Lista de Produtos possuem estruturas semelhantes, mas aplicam as regras somente ao contexto selecionado.*

1. Query Fields

Os **Query Fields** definem quais informações dos produtos serão consideradas durante a pesquisa.

Um produto pode possuir diferentes campos indexados, como:

- nome do produto;
- palavras-chave;
- descrição;
- marca;
- SKU;
- nome preparado para correspondência exata;
- nome preparado para variações das palavras;
- nome preparado para similaridade fonética.

Um mesmo conteúdo pode existir em mais de uma forma no índice. O nome do produto, por exemplo, pode ter uma versão voltada para correspondência exata e outra preparada para localizar variações.

Como interpretar os nomes dos campos

Os nomes apresentados nessa tela são técnicos e normalmente utilizam inglês.

ProductNameExact

Representa o campo utilizado para pesquisar o nome do produto com uma correspondência mais exata.

Separando o nome:

- ProductName: nome do produto;
- Exact: versão destinada a correspondências mais exatas.

Para uma pesquisa como:

Tênis Nike Air Max

esse campo tende a valorizar produtos cujo nome esteja muito próximo da expressão pesquisada.

ProductNameStem

Normalmente representa uma versão do nome preparada para considerar variações da palavra.

Exemplo:

- camisa;
- camisas;
- camiseta;
- outras variações processadas pelo campo.

ProductNamePhonetic

Normalmente representa uma versão do nome preparada para considerar similaridade sonora.

Pode auxiliar em situações como:

- tenis e tênis;
- smartfone e smartphone;
- outras palavras com sons próximos.

SearchKeywordsExact

Representa as palavras-chave do produto em uma versão destinada a correspondências mais exatas.

Campos terminados em Term

Quando o nome possui o sufixo Term, como:

ProductNameExactTerm

a regra tende a considerar os termos individualmente, e não somente a expressão completa.

***Importante:** nomes como `ProductNameExact` não são campos nativos obrigatórios do Elasticsearch. Eles são definidos pela estrutura de indexação utilizada pela Linx Commerce.*

***Não traduza nem altere livremente esses nomes.** O campo informado precisa existir no índice. Substituir `ProductNameExact` por `NomeProdutoExato`, por exemplo, pode fazer a regra deixar de funcionar.*

2. Query Boosts

Os **Query Boosts** são regras utilizadas para influenciar a pontuação de relevância dos produtos.

Cada regra determina:

- qual campo será consultado;
- qual peso será aplicado;
- como o texto pesquisado será tratado;
- se haverá tolerância a variações;
- como caracteres especiais serão processados.

A tela pode apresentar os campos:

Campo	Finalidade
Name	Identificador da regra.
Field	Campo do produto pesquisado.
Boost	Peso de relevância.
Similarity	Tolerância para termos semelhantes.
Term	Indica se o texto pesquisado será utilizado.
Escape	Caracteres especiais que devem ser tratados.
Format	Formato utilizado para montar a consulta.
Operator	Relação lógica entre as palavras.
Type	Tipo técnico da consulta.

Name

O campo **Name** identifica a regra dentro da configuração.

Exemplo:

ProductNameExact

Ele:

- não é exibido ao consumidor;
- não representa o texto pesquisado;
- pode possuir o mesmo valor do Field;
- deve seguir o padrão técnico utilizado pela configuração.

Evite renomear regras existentes. A alteração pode impedir que a aplicação reconheça corretamente a configuração.

Field

O campo **Field** define em qual informação indexada do produto a consulta será executada.

Exemplo:

ProductNameExact

Nesse caso, a regra pesquisará no campo técnico relacionado ao nome exato do produto.

Utilize apenas campos existentes. Um nome digitado incorretamente pode fazer a regra não retornar produtos, ser ignorada ou gerar inconsistências.

Boost

O **Boost** representa o peso da regra.

Exemplo:

80

Quanto maior o Boost em comparação com as outras regras, maior tende a ser a influência daquela correspondência na posição do produto.

Exemplo de comparação

Regra	Boost
Nome exato do produto	80
Palavras-chave exatas	30
Nome do produto por termos	25
Similaridade fonética	10
Descrição	5

Nesse cenário, encontrar a expressão no nome exato tende a ser mais relevante do que encontrá-la somente na descrição.

O valor 80 não significa:

- 80% de correspondência;
- 80 posições acima;
- 80 pontos fixos;
- garantia de primeira posição.

O Boost é um **peso relativo**, combinado com as demais regras e critérios da busca.

***Como configurar:** aumente o Boost quando uma correspondência importante não estiver recebendo prioridade suficiente. Reduza quando uma regra estiver levando produtos pouco relevantes para as primeiras posições.*

***Altere apenas uma regra por vez.** Aumentar todos os Boosts simultaneamente pode manter praticamente a mesma prioridade entre eles.*

Similarity

O campo **Similarity** está relacionado à tolerância utilizada para localizar termos parecidos.

Exemplo:

0.60

A busca aproximada pode considerar diferenças como:

- uma letra incorreta;
- uma letra ausente;
- uma letra adicional;
- duas letras invertidas;
- pequenas variações na escrita.

Exemplos:

Pesquisa	Termo cadastrado
notbook	notebook
camizeta	camiseta
smartfone	smartphone

O valor 0.60 não deve ser interpretado literalmente como “60% das letras precisam ser iguais” ou “60% de relevância”.

Nas versões atuais do Elasticsearch, a busca aproximada normalmente utiliza uma quantidade de alterações permitidas entre as palavras. O decimal apresentado na tela pode ser convertido ou interpretado por uma camada própria do Smart Finder.

***Importante:** a interpretação exata de 0.60 depende de como a Linx Commerce transforma essa configuração na consulta enviada ao Elasticsearch.*

***Como configurar:** mantenha o valor atual quando erros de digitação estiverem sendo tratados corretamente. Revise somente quando termos muito diferentes forem considerados semelhantes ou quando pequenos erros não retornarem resultados.*

Term

A opção **Term** indica se o texto digitado pelo consumidor será utilizado naquela regra.

Para a pesquisa:

tênis azul

quando **Term** estiver marcado, esse conteúdo será incluído na consulta correspondente.

Não desmarque sem conhecer o impacto. A regra pode deixar de utilizar diretamente o termo pesquisado pelo consumidor.

3. Escape e caracteres especiais

O campo **Escape** define quais caracteres especiais devem ser tratados antes que o texto seja enviado ao Elasticsearch.

Exemplo:

`\+-&!(){}[]^"~*?:/`

Essa sequência não é uma senha, uma fórmula nem um termo de pesquisa.

Ela representa uma lista de caracteres que possuem funções especiais na linguagem de consultas do Elasticsearch e do Lucene.

Sem o tratamento adequado, esses símbolos podem ser interpretados como comandos em vez de texto comum.

O que é escape?

Escape é o processo de informar ao motor de busca:

“Interprete este símbolo como parte do texto, e não como uma instrução.”

Considere a pesquisa:

C++

O símbolo + pode ter uma função especial na consulta. Para tratá-lo como parte do nome, a expressão pode ser preparada internamente como:

C\+\+

A barra invertida \ indica que o caractere seguinte deve ser considerado literalmente.

Outros exemplos:

Texto digitado	Forma tratada internamente
C++	C\+\+
AC/DC	AC\/DC
kit (2+1)	kit \ (2\+1\)
TV 55"	TV 55\"
produto:azul	produto\:azul

O consumidor não precisa inserir essas barras. O tratamento é realizado pela aplicação.

O que significa cada caractere?

\ — Barra invertida

É o próprio caractere de escape.

Exemplo:

`\+`

Significa que o + deve ser tratado como texto comum.

+ — Termo obrigatório

Pode indicar que o termo seguinte precisa estar presente.

Exemplo técnico:

`+camiseta +azul`

Pode representar uma consulta que exige os dois termos.

Em um nome como C++, os sinais devem ser tratados como parte do texto.

- — Exclusão de termo

Pode indicar que determinado termo deve ser excluído.

Exemplo técnico:

`camiseta -infantil`

Pode significar pesquisar camiseta e excluir produtos relacionados a infantil.

Porém, o hífen também pode fazer parte de nomes e códigos:

`UV-50`

& — Operação AND

Na sintaxe completa, && pode representar a operação lógica **AND**, exigindo que duas condições sejam atendidas.

A configuração apresenta o caractere & individualmente porque a plataforma pode identificá-lo e tratá-lo antes da montagem final da consulta.

| — Operação OR

Pode representar a operação lógica **OR**.

Exemplo:

`camiseta | camisa`

Indica que a busca pode considerar camiseta ou camisa.

! — Negação

Pode representar uma negação ou exclusão.

Em uma pesquisa como:

Oferta!

o ponto de exclamação deve ser considerado pontuação, e não uma instrução.

(e) — Agrupamento

Os parênteses agrupam partes da consulta.

Exemplo técnico:

(camiseta OR camisa) AND azul

O conteúdo entre parênteses é processado como um grupo.

Quando fizerem parte do nome pesquisado, devem ser tratados como caracteres comuns.

{ e } — Intervalo exclusivo

Podem representar uma pesquisa por intervalo sem incluir os valores inicial e final.

Exemplo técnico:

price:{100 TO 200}

Pode indicar valores maiores que 100 e menores que 200.

[e] — Intervalo inclusivo

Podem representar intervalos que incluem os valores inicial e final.

Exemplo:

price:[100 TO 200]

Pode incluir produtos com valores iguais a 100 ou 200.

^ — Aumento de relevância

Pode atribuir maior peso diretamente a um termo.

Exemplo:

camiseta^5 camisa

Nesse caso, correspondências com “camiseta” recebem mais relevância que correspondências com “camisa”.

" — Pesquisa por frase

As aspas podem indicar que as palavras devem ser consideradas como uma frase.

Exemplo:

"tênis azul"

Isso tende a exigir ou valorizar a proximidade e a ordem das palavras.

Porém, as aspas também podem fazer parte de um produto:

TV 55"

~ — Similaridade ou proximidade

Depois de uma palavra, pode indicar busca aproximada:

notbook~1

Depois de uma frase, pode indicar tolerância de distância entre as palavras:

"tênis corrida"~3

* — Coringa para vários caracteres

Pode substituir zero ou vários caracteres.

Exemplo:

camis*

Pode encontrar:

- camisa;
- camiseta;
- camisaria.

Consultas com coringas muito amplos podem exigir mais processamento, principalmente quando o asterisco é utilizado no início da palavra.

? — Coringa para um caractere

Pode substituir um único caractere.

Exemplo:

te?is

Em uma pesquisa comum como tem tênis?, deve ser tratado apenas como pontuação.

: — Separação entre campo e valor

Pode indicar em qual campo a pesquisa deve ocorrer.

Exemplo:

`brand:Nike`

Significa pesquisar o valor Nike no campo brand.

Em um texto como Promoção: tênis, deve ser tratado como pontuação.

/ — Expressão regular

Pode delimitar uma expressão regular.

Exemplo técnico:

`name:/camis.*/`

Quando a barra fizer parte de um nome, como AC/DC, deve ser considerada texto comum.

O campo Escape deve ser alterado?

Em condições normais, não.

Mantenha:

`\+-&|!(){}[]^"~*?:/`

Alterar essa lista pode provocar:

- falhas em pesquisas;
- interpretação indevida de símbolos;
- exclusão involuntária de termos;
- consultas excessivamente abrangentes;
- resultados inesperados;
- problemas em nomes com hífen, barra, aspas ou parênteses.

Altere o campo Escape somente com orientação técnica.

A lista apresentada na tela pode não ser idêntica à lista completa documentada pelo Elasticsearch, pois parte do tratamento pode ocorrer em outras camadas da plataforma.

4. Format

O campo **Format** define como o termo pesquisado será inserido na expressão de busca.

Pode ser utilizado um marcador como:

`{0}`

O `{0}` representa o texto digitado pelo consumidor.

Exemplo de formato:

{0}

Para a pesquisa:

tênis azul

a expressão pode ser montada como:

(tênis azul)

Os parênteses mantêm os termos agrupados.

Não remova o marcador {0}. Sem ele, a regra pode deixar de incluir o texto pesquisado.

5. Operator

O **Operator** define a relação lógica entre as palavras.

Operador	Comportamento
AND	Exige maior correspondência entre os termos.
OR	Permite que apenas parte dos termos corresponda.
Em branco	Utiliza o comportamento padrão da configuração.

Considere a pesquisa:

camiseta azul masculina

Com AND, a busca tende a exigir os três termos.

Com OR, pode retornar produtos que correspondam apenas a uma ou duas palavras.

Orientação: utilize AND quando a precisão for mais importante. Utilize OR ou o padrão quando for importante reduzir buscas sem resultados.

6. Type

O campo **Type** informa o tipo técnico de consulta utilizado pela regra.

Ele pode estar relacionado a comportamentos como:

- correspondência por termo;
- correspondência por frase;
- busca aproximada;
- busca parcial;
- outro tipo aceito pela implementação.

Quando o campo estiver em branco, mantenha-o em branco, salvo quando existir uma orientação técnica específica. Um Type inválido pode fazer a regra ser ignorada ou gerar resultados inesperados.

7. Exemplo completo de Query Boost

Considere a seguinte configuração:

Campo	Valor
Name	ProductNameExact
Field	ProductNameExact
Boost	80
Similarity	0.60
Term	Marcado
Escape	\+-& !(){}[]^"~*?:/
Format	({0})
Operator	Em branco
Type	Em branco

Para uma pesquisa por:

Tênis Nike Air

a regra funciona, de forma simplificada, assim:

1. utiliza o termo porque **Term** está marcado;
2. trata os caracteres especiais indicados em **Escape**;
3. insere o termo no modelo definido em **Format**;
4. pesquisa no campo ProductNameExact;
5. aplica peso 80 à correspondência;
6. considera a configuração de Similarity;
7. combina a pontuação com as demais regras;
8. utiliza o comportamento padrão para Operator e Type.

Essa regra não garante isoladamente a primeira posição. A classificação final considera todas as regras, campos, ordenações e correspondências aplicadas à pesquisa.

8. Extra Params

Os **Extra Params** são parâmetros adicionais utilizados na montagem ou no processamento da consulta.

Na tela podem aparecer configurações como:

- mm;
- qf;
- qs;
- ps.

***Observação técnica:** esses nomes são conhecidos principalmente no ecossistema Lucene/Solr. No Smart Finder, eles podem ser processados por uma camada interna da plataforma antes da consulta ao Elasticsearch. Por isso, a função abaixo descreve o comportamento esperado, mas a implementação pode possuir adaptações próprias.*

mm — Minimum Should Match

Define a quantidade mínima de termos que precisa corresponder para que um produto seja considerado.

Exemplo:

mm = 75%

Para uma pesquisa com quatro termos:

camiseta masculina azul algodão

uma configuração de 75% representa, de forma simplificada, a exigência de correspondência de pelo menos três termos.

- valor maior: busca mais restritiva;
- valor menor: busca mais abrangente;
- valor muito alto: pode aumentar buscas sem resultados;
- valor muito baixo: pode apresentar produtos pouco relacionados.

qf — Query Fields

Define os campos pesquisados e a importância relativa de cada um.

Exemplo:

ProductNameExact^80 ProductName^25 SearchKeywordsExact^30

Separando:

- ProductNameExact^80: nome exato com peso 80;
- ProductName^25: nome do produto com peso 25;
- SearchKeywordsExact^30: palavras-chave exatas com peso 30.

O caractere ^ separa o nome do campo de seu peso.

Mantenha os nomes técnicos existentes. Não informe campos que não existam no índice.

qs — Query Phrase Slop

Controla a tolerância de distância em frases pesquisadas explicitamente.

Exemplo:

qs = 5

Quanto menor o valor, mais próximas as palavras precisam estar. Quanto maior, maior a tolerância de distância ou variação.

ps — Phrase Slop

Também controla a proximidade entre palavras, mas normalmente é utilizado em regras de frase voltadas ao aumento de relevância.

Para a pesquisa:

tênis corrida

um produto com:

tênis de corrida

pode receber maior pontuação porque os termos aparecem próximos.

O ps normalmente influencia a pontuação, e não necessariamente a inclusão ou exclusão do produto.

9. O que é mais seguro alterar?

Ajustes mais comuns

Os ajustes operacionais mais comuns costumam envolver:

- Boost;
- seleção de campos já conhecidos;
- prioridade entre nome, palavras-chave e descrição;
- correspondência mínima;
- Operator, quando o comportamento estiver bem definido.

Campos que devem ser preservados

Evite alterar sem orientação técnica:

- Name;
- Field desconhecido;
- Escape;
- Format;
- Type;
- parâmetros extras não documentados;
- Similarity sem validação da implementação.

10. Como testar uma alteração

Depois de modificar uma configuração:

1. registre os valores anteriores;
2. altere apenas uma regra ou parâmetro;
3. clique em **Salvar**;
4. selecione **Visualizar Preview**;
5. teste um nome exato de produto;
6. teste uma busca com várias palavras;
7. teste um erro de digitação;
8. teste um termo genérico;
9. teste pesquisas com caracteres especiais;
10. compare a ordem dos resultados;
11. publique somente após a validação.

Sugestões de pesquisas

Tipo de teste	Exemplo
Nome exato	Tênis Nike Air Max
Termo genérico	tênis
Várias características	tênis masculino azul corrida
Erro de digitação	teniz
Código com hífen	UV-50
Nome com barra	AC/DC
Nome com símbolo	C++
Medida com aspas	TV 55"
Uso de parênteses	kit (2+1)

Boa prática: teste as mesmas pesquisas antes e depois da alteração para comparar quantidade, qualidade e ordenação dos resultados.

11. Recomendações importantes

- Não traduza nomes técnicos.
- Não exclua regras sem registrar os valores atuais.
- Não aumente todos os Boosts simultaneamente.
- Não altere o Escape para tentar melhorar a relevância.
- Faça pequenas alterações.
- Utilize sempre o preview.
- Teste diferentes tipos de pesquisa.
- Monitore buscas sem resultados após a publicação.
- Restaure os valores anteriores caso o comportamento piore.

Não existe uma única configuração ideal para todas as lojas. A relevância depende do catálogo, da qualidade dos cadastros, dos campos indexados e da forma como os consumidores pesquisam.

Saiba mais

Para conhecer os conceitos técnicos utilizados pelo Elasticsearch, consulte a documentação oficial:

- [Documentação oficial do Elasticsearch](#)
- [Query DSL — linguagem de consultas](#)
- [Query string e caracteres reservados](#)
- [Simple query string](#)
- [Fuzzy query — busca aproximada](#)
- [Parâmetros DisMax e eDisMax — referência complementar](#)

Importante: a documentação do Elasticsearch explica os conceitos nativos da tecnologia. Os nomes de campos e alguns parâmetros apresentados no Smart Finder podem possuir adaptações específicas da Linx Commerce.