SENAI

*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

**SÉRIE TI - SOFTWARE**

# LÓGICA DE PROGRAMAÇÃO

## ALGORITMOS DE BUSCA

SENAI

*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

## ALGORITMOS DE BUSCA

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

**SENAI Departamento Nacional**
Unidade de Educação Profissional e Tecnológica - UNIEP

**SENAI Departamento Regional de Santa Catarina**
Gerência de Educação

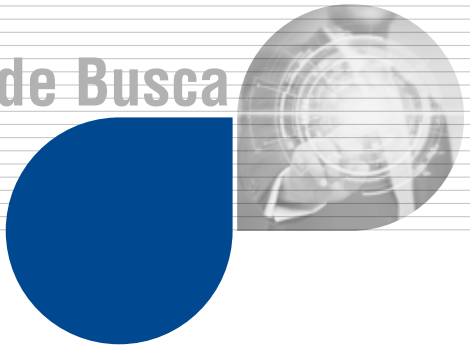
**SENAI**
Serviço Nacional de
Aprendizagem Industrial
Departamento Nacional

**Sede**
Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-9001 Fax: (0xx61) 3317-9190 • http://www.senai.br

# Lista de Ilustrações

# Sumário

# Algoritmos de Busca



## APRESENTAÇÃO

Olá! Sejam bem-vindos à unidade de Algoritmos de Busca!

Praticamente para tudo que se quer saber, as pessoas recorrem a um grande recurso computacional existente no mundo, chamado Google. E fazem isso com tanta frequência porque sabem que, quando pesquisam um determinado termo, o Google irá buscar resultados relevantes e retornará uma coleção de resultados pertinentes ao assunto pesquisado.

Na vida real, as pessoas também executam diversas vezes seus próprios algoritmos de busca quando, por exemplo, querem encontrar determinados contatos em suas agendas de telefones, ou quando querem encontrar um livro com um determinado título na estante.

Nesta unidade, será apresentado o Algoritmos de busca, para que você possa entender melhor sua definição, estrutura e obviamente sua a aplicação prática implementada em linhas de código.

Bons estudos!

## DEFINIÇÃO

Uma busca se refere ao rastreamento de um determinado elemento que está disposto em uma coleção de dados. Essa busca se faz a partir de uma lógica de programação, que evite, por exemplo, a execução de muitas linhas de código e também que impeça a necessidade da inspeção item a item por uma pessoa.

O intuito maior de um algoritmo de busca é que o usuário forneça o que quer encontrar e o próprio programa desenvolvido encontre o resultado correto (ou mais apropriado), sem que ele (usuário) tenha que procurar de forma manual.

As vantagens de uma busca a partir de um algoritmo de busca são muitas:

a) Confiabilidade no resultado;

b) Velocidade;

c) Estabelecimento de critérios de busca;

d) Padronização;

e) Menor esforço.

Imagine que você está diante do desafio de encontrar o livro "Senhor dos Anéis: a sociedade do anel" em uma estante de livros como essa da imagem a seguir.

mpafoto ((20--?))

Figura 1 - Estante de Livros

Certamente essa tarefa daria um certo trabalho para encontrar esse livro, ainda mais se essa estante cheia de livros não possuir nenhuma lógica de ordenação e/ou identificação dos livros por algum código. Isso seria uma tremenda dor de cabeça, não é? Muito provavelmente você perderia um precioso tempo procurando esse livro em meio de tantos outros. Porém, poderia implementar um algoritmo que fosse capaz de encontrar o livro de forma mais rápida e precisa.

## ESTRUTURA

Basicamente, o que se pode fazer é criar um algoritmo que percorra a totalidade de extensão de itens e, caso encontrar o item que se busca, retorne à posição do livro encontrado e pare a execução do algoritmo.

Observe como ficaria essa proposta ao se implementar esse algoritmo na linguagem de programação Java.

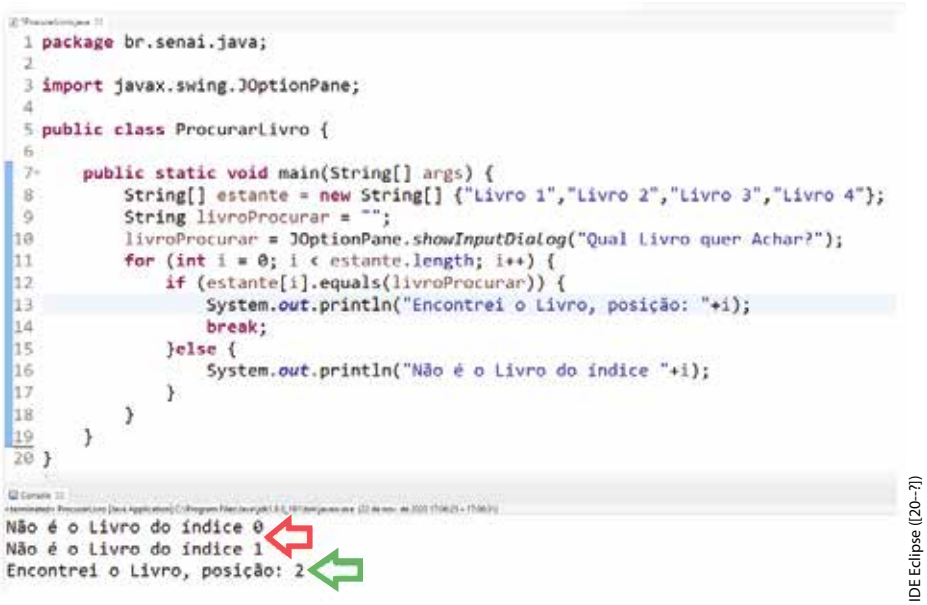



Figura 2 - *Screenshot* da IDE Eclipse – Algoritmo de busca do livro na estante
Fonte: do Autor (2020)

Para melhor compreender a proposta, analise as linhas de código a partir da tabela, a seguir.

| LINHA | ALGORITMO DE BUSCA DE LIVROS NA ESTANTE – IDE ECLIPSE |
|---|---|
| 8 | Declaração do vetor do tipo String chamado "estante" e a definição de quatros livros dentro desse vetor (Livro 1, Livro 2, Livro 3 e Livro 4). |
| 9 | Variável criada para que o usuário informe qual o nome do livro que ele deseja buscar dentro do vetor "estante". |
| 10 | Usuário informa, na janela apresentada, qual o título do Livro. |
| 11 até 18 | Estrutura de repetição FOR, que fará um laço do índice 0 até o tamanho de registros existentes no vetor "estante". |
| 12 até 15 | Estrutura IF que verifica se o livro naquele índice é igual ao título do livro informado pelo usuário. Caso seja verdadeiro, irá executar as linhas 13 e 14. |
| 15 até 17 | Estrutura ELSE, que executa a linha 16, caso a comparação da estrutura IF (linha 12 até 15) seja falsa, ou seja, o título não é igual. |
| 13 | Programa apresenta o índice do Livro encontrado na busca. |
| 14 | O programa é encerrado no comando BREAK, garantindo que assim a busca se encerre, visto que o livro desejado foi encontrado. |

Tabela 1 - Interpretação das linhas de código em Java da Figura Algoritmo de busca do livro na estante
Fonte: do Autor (2020)

Na execução do exemplo, foi informado que se desejava buscar "Livro 3". Observe, na figura, que, na seta vermelha, aparece a mensagem que o programa não encontrou o livro pesquisado no índice 0 e no índice 1. Porém, na seta verde, ele mostra que encontrou o livro que foi colocado na busca, apresentando

assim o índice 2. Note também que o programa se encerra, não sendo mais dada continuidade à estrutura FOR, visto que não se apresenta mais nenhuma mensagem sobre o livro 4.

## APLICAÇÃO NA INDÚSTRIA

Na indústria, encontra-se a aplicação do algoritmo de busca em diversas situações. Imagine que uma empresa que realiza os testes químicos de determinados reagentes deverá sempre excluir os reagentes que tiveram seus testes iguais a falso.

Seria possível aplicar a mesma lógica utilizada no algoritmo de busca, porém com a única diferença: nesse caso, é que não se parasse a verificação de busca e comparação. Ou seja, se for necessário verificar 1000 testes de reagentes, seriam buscados todos os resultados existentes nessa coleção que estejam com a situação igual a falso, registrando-os e excluindo-os de análises futuras.

Novamente, é possível perceber que, caso essas análises fossem feitas meramente de forma manual, com pessoas olhando reagente a reagente para verificar o resultado (verdadeiro ou falso), muito provavelmente se teria um processo muito demorado e também de baixa confiabilidade.

## EXEMPLOS

Para fechar esse tópico, será implementado o caso apontado anteriormente, em que se apresentou o cenário de um algoritmo que fosse capaz de analisar reagentes e classificá-los a partir do resultado de seus testes. Então, imagine que será desenvolvido um programa em Java que verificará a cor dos reagentes e definirá se eles são Verdadeiros (verde) ou Falsos (vermelhos). Assim, a partir dessas informações, o programa permitirá que se buque os testes Falsos, removendo-os da bancada, além de adicioná-los a uma outra fila para análise futura.

Basicamente, é possível representar graficamente a simulação do que se quer fazer a partir do seguinte cenário:

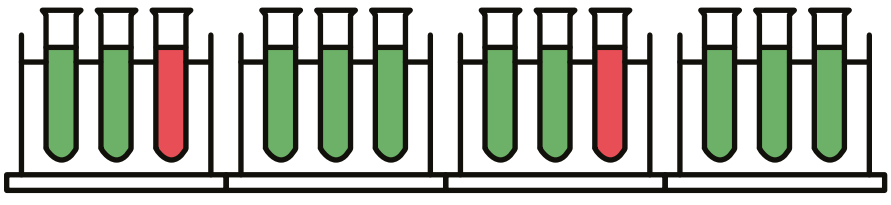



Figura 3 - Bancada para testes de reagentes
Fonte: do Autor (2020)

Ao observar essa bancada, verifica-se que há doze testes, que vão do índice 0 até o índice 11. É possível ver também que os reagentes do índice 2 e 8, por algum motivo, não passaram nos testes, ficando com a coloração vermelha. Já todos os demais foram positivos, ficando com a coloração verde.


IDE Eclipse ([20--?])

Figura 4 - *Screenshot* da IDE Eclipse – Algoritmo de busca de testes falsos
Fonte: do Autor (2020)

Observe, nessa implementação, que, no resultado, o programa apresenta a coleção de reagentes que deram errado, mostrando que os reagentes 2 e 8 apontaram resultado falso, pois suas cores estavam vermelhas.

**SAIBA MAIS**

Existem outras formas de busca que utilizam técnicas muito interessantes e avançadas para trazer performance e outras características aos critérios de programação. Nos exemplos anteriores, foi utilizada uma técnica de algoritmo de busca sequencial, porém você poderá encontrar outras formas de busca, tal como a busca binária.

O artigo da Khan Academy pode ajudar a compreender o conceito de busca binária, disponível em: https://pt.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search.

Procure pesquisar sobre os algoritmos de busca e, quem sabe, busque também como funciona o complexo sistema de busca do Google. Você ficará impressionado pela complexidade, técnica e infraestrutura que existe por trás dessa incrível ferramenta de busca mundial.

***

# PALAVRA DO DOCENTE

Você estudou, nesta unidade, os algoritmos de busca e já está se tornando um programador com uma boa bagagem de conhecimentos. O desenvolvimento da lógica é alcançado somente com a prática. Faça muitos exercícios práticos, aplique o mesmo algoritmo em diversas linguagens e vá se especializando a cada dia. O conhecimento de lógica de programação é cumulativo, isto é, você precisa aprender uma estrutura antes da outra, ainda que de forma granular.

# REFERÊNCIAS

ARAÚJO, Everton Coimbra de. **Algoritmos**: fundamentos e prática. Florianópolis: Visual Books, 2005.

PREISS, Bruno R. **Estruturas de dados e algoritmos**: padrões de projetos orientados a objetos com Java. Rio de Janeiro (RJ): Campus, 2001.

SOUZA, Marco Antonio de; GOMES, Marcelo Marques; SOARES, Márcio José; CONCILIO, Ricardo. **Algoritmos e lógica de programação**. São Paulo (SP): Thomson Pioneira, 2005.

SENAI

*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**