

Curso

Algoritmos & Programação em JAVA

Atualizado até o Java 21 & Eclipse 2023-09



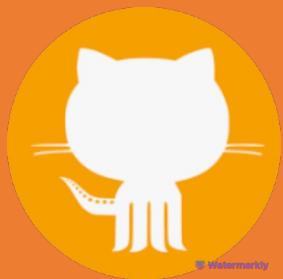
Prof. Msc. Antonio B. C. Sampaio Jr
ENGENHEIRO DE SOFTWARE & PROFESSOR

@abctreinamentos
@amazoncodebr

www.abctreinamentos.com.br
www.amazoncode.com.br



REPOSITÓRIO GITHUB



antonio-sampaio-jr / **algoritmos-programacao-java**

CONTEÚDO PROGRAMÁTICO



- UNIDADE 1 – INTRODUÇÃO
- UNIDADE 2 – CONTRUÇÃO DE ALGORITMOS
 - Lógica de Programação
 - Conheça os Algoritmos que Estão por Toda Parte!
 - UBER, Google, Google Maps, NETFLIX, E-COMMERCE, NISSAN ProPILOT
 - Etapas de Desenvolvimento [NOVO]

CONTEÚDO PROGRAMÁTICO



- UNIDADE 2 – CONSTRUÇÃO DE ALGORITMOS
(Continuação)
 - Instalação do JAVA 21 & ECLIPSE 2023-09 [NOVO]
 - Arquitetura Computacional na Qual Construímos Algoritmos
 - Operadores Matemáticos, Relacionais e Lógicos
 - Comandos de Entrada e Saída
 - Exercícios

UNIDADE 2

CONSTRUÇÃO DE ALGORITMOS

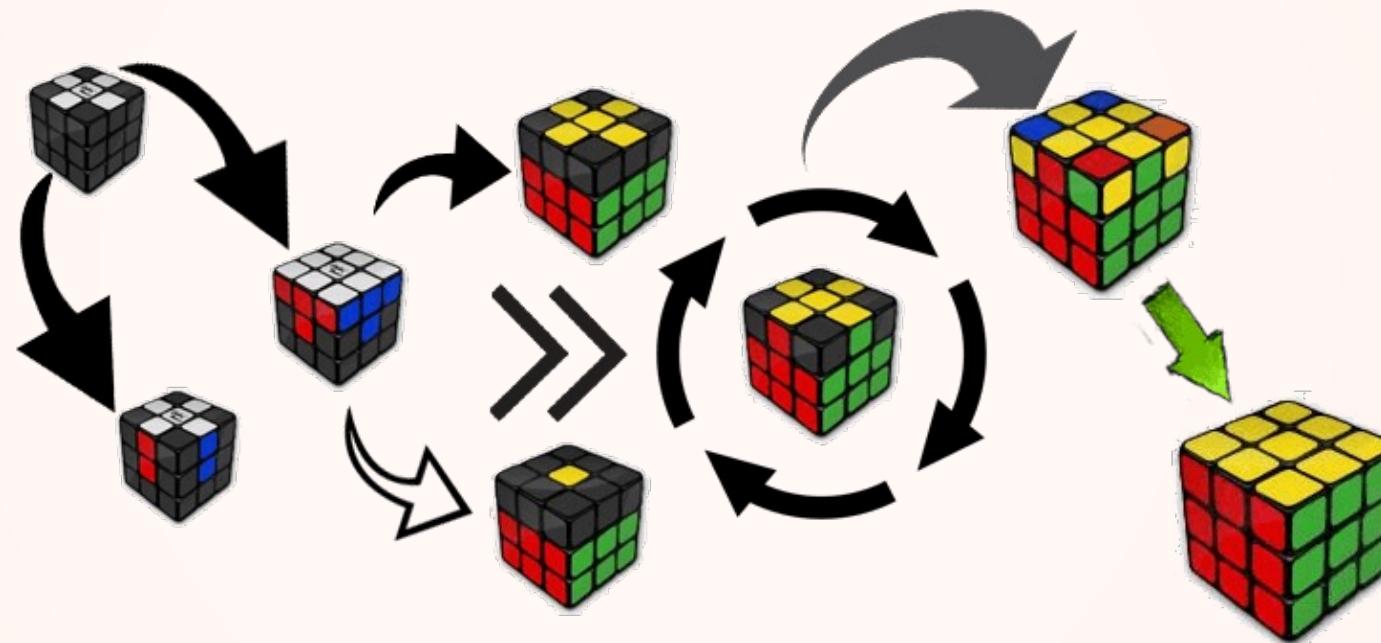
Lógica de **Programação**

O que é lógica?

- Do grego “logos”, significa ideia, argumento, pensamento;
- É arte do bem pensar;
- É a correção do pensamento;
- É a correção do raciocínio;
- Segundo Forbellone, o uso corriqueiro da palavra lógica está normalmente relacionado à coerência e racionalidade. Também está muito relacionado à matemática, mas não somente a ela, pois a lógica tem relação com todas as demais ciências.

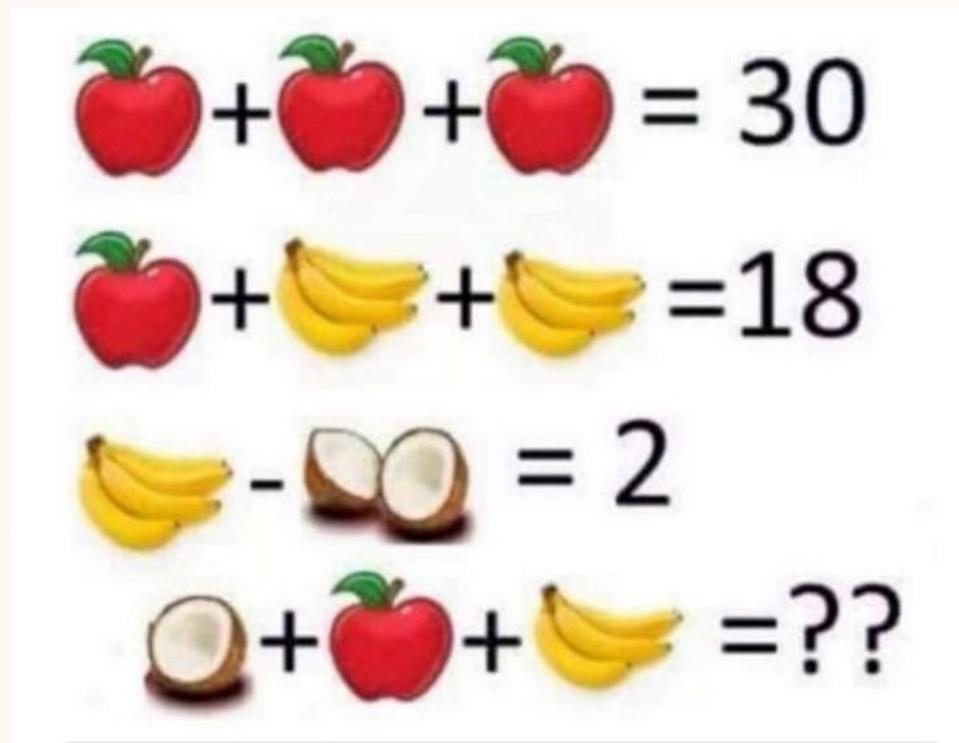
Fonte: <https://filosofandoehistoriando.blogspot.com/2015/08/o-nascimento-da-logica.htm>

O uso da lógica no dia a dia



Fonte: <http://blog.triadworks.com.br/novo-curso-de-logica-de-programacao>

O uso da lógica no dia a dia



Fonte: <https://revistaglamour.globo.com/Lifestyle/Must-Share/noticia/2016/02/problema-com-frutas-faz-sucesso-nas-redes-sociais.html>

O uso da **lógica no dia a dia**

O resultado é ...

... 14!!!



O uso da **lógica no dia a dia**

- Sempre que pensamos, o raciocínio e a lógica nos acompanham;
- A palavra falada é a representação do pensamento. Portanto, quando falamos usamos a lógica;
- A palavra escrita é a representação da palavra falada. Portanto, quando escrevemos também usamos a lógica;
- Resumindo, quando pensamos, falamos ou escrevemos corretamente, utilizamos a **LÓGICA!**

O uso da lógica no dia a dia

Exemplo:

Problema: Desejo pegar uma caneta.

A caneta está dentro da gaveta;

A gaveta está fechada;

Como Resolver?

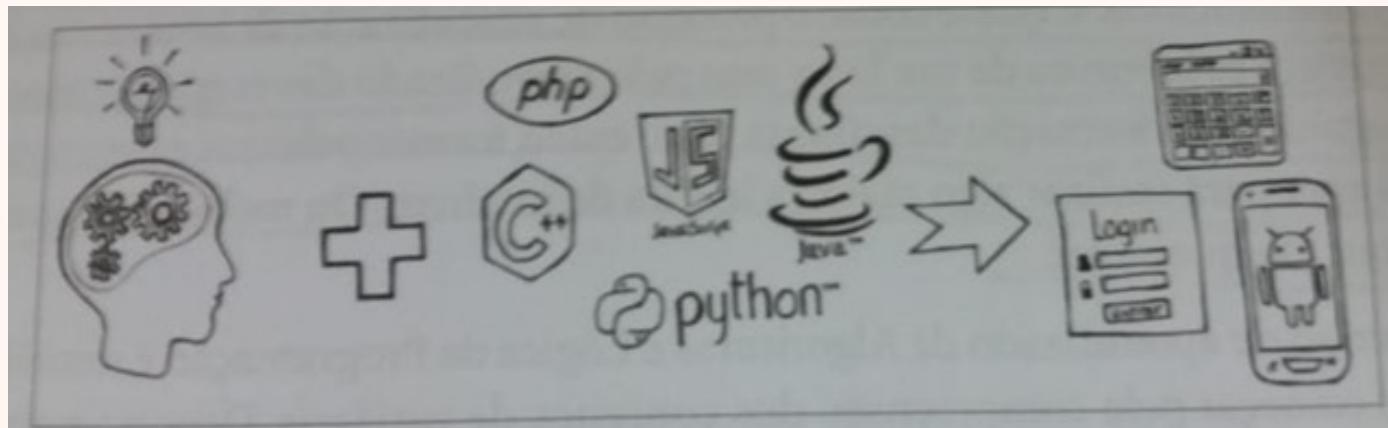
Usando a LÓGICA...

Primeiro abrimos a gaveta....depois
pegamos a caneta!



O que é **lógica de programação?**

- Lógica de Programação é a técnica de desenvolver um conjunto de instruções em uma sequência lógica (algoritmo) para solucionar um problema de forma automatizada (utilizando dispositivos computacionais).



Fonte: Edécio Fernando Iepsen

- Com o crescimento rápido do acesso a informações digitais, **os algoritmos estão por toda a parte!**

Conheça os **Algoritmos** que Estão por **Toda a Parte!**

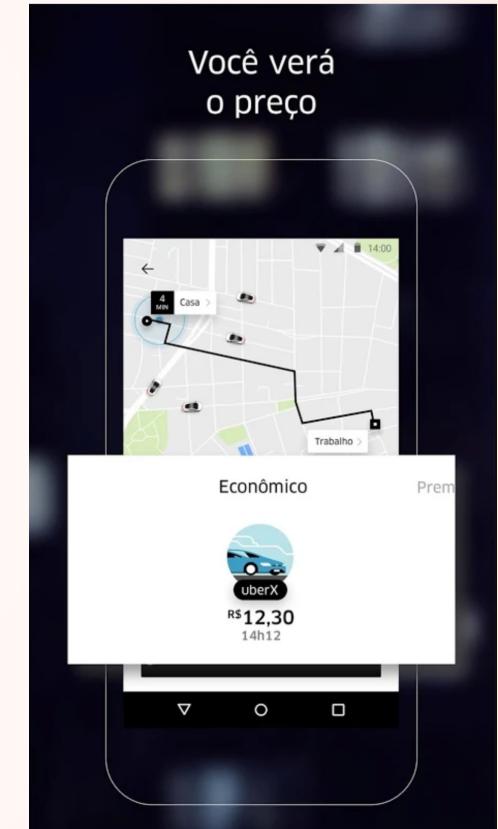
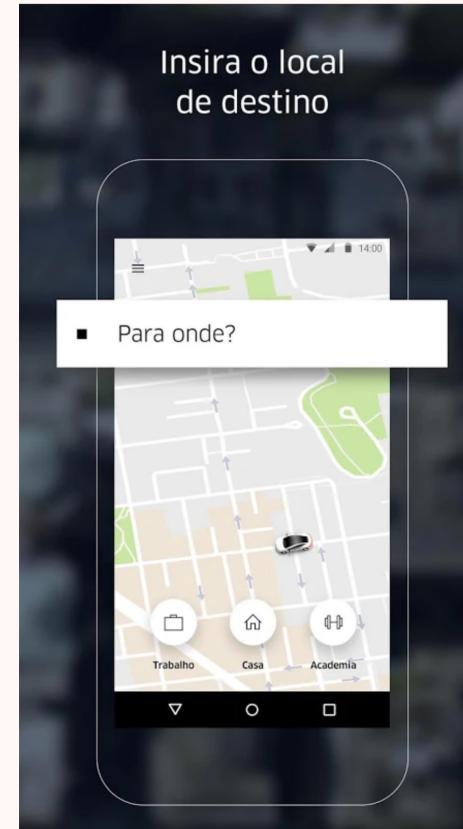
UBER



O que é o **UBER?**

- A Uber é uma plataforma de tecnologia que conecta motoristas parceiros com usuários por meio de um aplicativo (APP) para smartphones (iOS, Android e Windows App).

Como o UBER funciona?



Como o UBER funciona?

Seguindo uma sequência de **PASSOS LÓGICOS**:

1º Passo: Abrir o Aplicativo no celular;

2º Passo: Informar o destino;

3º Passo: Calcular o preço da viagem (utiliza como PRINCIPAIS PARÂMETROS:
TEMPO, DISTÂNCIA e DEMANDA);

4º Passo: Solicitar um dos serviços disponíveis (UBERX ou SELECT);

Como o UBER funciona?

5º Passo: A solicitação de viagem será enviada para os motoristas parceiros nas proximidades. Se algum motorista parceiro aceitar, o aplicativo irá mostrar o tempo estimado para que o motorista chegue ao ponto de embarque. O aplicativo também notificará quando o motorista parceiro estiver prestes a chegar.

6º Passo: Depois que a viagem é solicitada, o aplicativo informará o nome do motorista, o tipo de veículo, placa do automóvel na tentativa de garantir a segurança do usuário e do motorista. Uma previsão do tempo de chegada também é informada. Caso o motorista esteja em outra corrida, esta informação é repassada para o cliente.

Como o UBER funciona?

7º Passo: Ao chegar, o motorista informa ao aplicativo o início da corrida, e, através do GPS, mostra o caminho correto de acordo com localização atual;

8º Passo: Chegando ao destino, faz-se necessário informar a finalização da viagem. A tarifa será exibida ao motorista informando o método de pagamento. Caso seja cartão, confirmará o débito efetuado; no caso de dinheiro, cobrará o valor informado ao cliente. Os valores da corrida podem sofrer alteração caso o percurso da corrida tenha sofrido alteração e/ou o tempo percorrido tenha sido muito maior do que o previsto pelo aplicativo.

Como o UBER funciona?

21

9º Passo: Na finalização do serviço, o aplicativo irá solicitar ao usuário uma avaliação do motorista. O motorista parceiro também é solicitado a avaliar o usuário.

Como o UBER funciona?

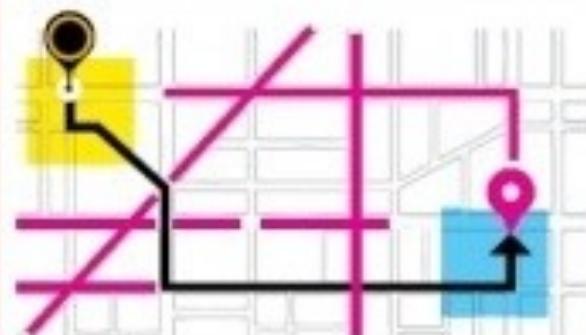
22



Como o UBER funciona?

ALGORITMOS Y COSTO

3 Cada empresa utiliza algoritmos con parámetros que establecen, además de la mejor ruta, el costo del servicio.



Uber



- Considera como factores para el precio el tiempo y la distancia. Cuando la demanda supera la oferta de vehículos, la tarifa sube.

Fonte: <https://www.linkedin.com/pulse/un-algoritmo-calcula-las-tarifas-de-aplicaciones-le%C3%B3n-trisoglio>

Como o UBER funciona?

```
12 function computeRoutes(
13   riders, route, used, n) {
14   if (n == riders.length) {
15     saveRoute(route);
16     return;
17   }
18   for (var i = 0;
19       i < riders.length;
20       i++) {
21     if (used[i]) continue;
22     route.push(
23       riders[i].location);
24     used[i] = true;
25     computeRoutes(
26       riders,
27       route,
28       used,
29       n + 1);
30     used[i] = false;
31     route.pop();
32   }
33 }
```

Novidades do UBER

A Uber anunciou nesta terça-feira, 10 de abril de 2018, que começou a usar algoritmos de inteligência artificial para proteger motoristas em sua plataforma de carros particulares sob demanda. O objetivo é evitar que parceiros sejam atraídos para emboscadas.

"A Uber tem adotado a tecnologia de machine learning para identificar riscos com base na análise, em tempo real, dos dados das milhões de viagens realizadas diariamente por meio do aplicativo", explicou a empresa por meio de uma nota oficial.

Google

Como o Google funciona?

- Os algoritmos do Google são programas de computador que buscam, por meio de processos e de fórmulas, “pistas” ou sinais para responder com exatidão as perguntas feitas em seu mecanismo de busca.

The screenshot shows a Google search results page. At the top, there's a logo with a soccer ball in front of a goal, followed by the word "GOOGLE". Below the logo is a search bar containing the query "curso de programação". To the right of the search bar are icons for device selection, microphone, and search. Below the search bar is a navigation menu with tabs: Todas (selected), Vídeos, Shopping, Notícias, Maps, Mais, Configurações, and Ferramentas. A message indicates approximately 47,500,000 results found in 0,55 seconds. The first result is a sponsored link for Udemy, titled "Site Oficial da Udemy™ | Mais de 65 mil Cursos Online" with a link to www.udemy.com/Cursos-Online/Site-Oficial. It includes a snippet: "Junte-se a mais de 20 milhões de alunos pelo mundo aprendendo online na Udemy! Cursos de Desenvolvimento · Cursos de TI · O que é a Udemy? · Cursos de Negócios". The second result is another sponsored link for NextU, titled "Cursos De Programação Web | Seja Freelancer Com A Next U" with a link to www.nextu.com.br/Curso-Online/ProgramaçãoWeb. It includes a snippet: "Destaque-se Entre Os Concorrentes. Estude Desenvolvimento Web. 500 Horas De Prática. Aulas Ilimitadas. Tutores Online. Certificados. Tutores. Alavanque Sua Carreira. Aulas 24h/dia. Torne-se Um Expert. Cursos em Tecnologia. Ganhe Seu Certificado. Android · Comece Hoje · Marketing Digital · Desenvolvedor Front End". The third result is a link for Udacity, titled "Introdução à Programação | Aprenda com Quem Faz o Futuro" with a link to br.udacity.com/Programação. It includes a snippet: "Prepare-se para Trabalhar em Grandes Empresas Aprendendo a Programar na Udacity! Cursos do Vale do Silício. Parceria Com Google. Cursos Em Português. Acelere Sua Carreira. Cursos: Marketing Digital, Startup Founder, Analista de Dados, Google AdWords, Programação. Acompanhe os Eventos · Fundamentos de Front-End · Imagine o Seu Futuro · Desenvolvedor iOS".

Como o Google funciona?

- Para isso, esses algoritmos são atualizados diariamente — às vezes, até mais de uma vez por dia — e utilizam mais de 200 códigos para filtrar, entre as milhões de páginas disponíveis na web, exatamente aquilo que você procura.
- O principal rastreador do Google é o **Googlebot**, usado para descobrir e analisar sites públicos. Seguindo os links, de uma página para outra automaticamente, e aos mapas de sites fornecidos, esse robô vasculha e lê todo o conteúdo disponível, visitando os mais diversos endereços, agregando informações sobre essas páginas, descobrindo páginas novas e atualizadas, bem como alterações em sites existentes e links inativos, todas os dados serão detectados e usados para a atualização do índice.

Como o Google funciona?

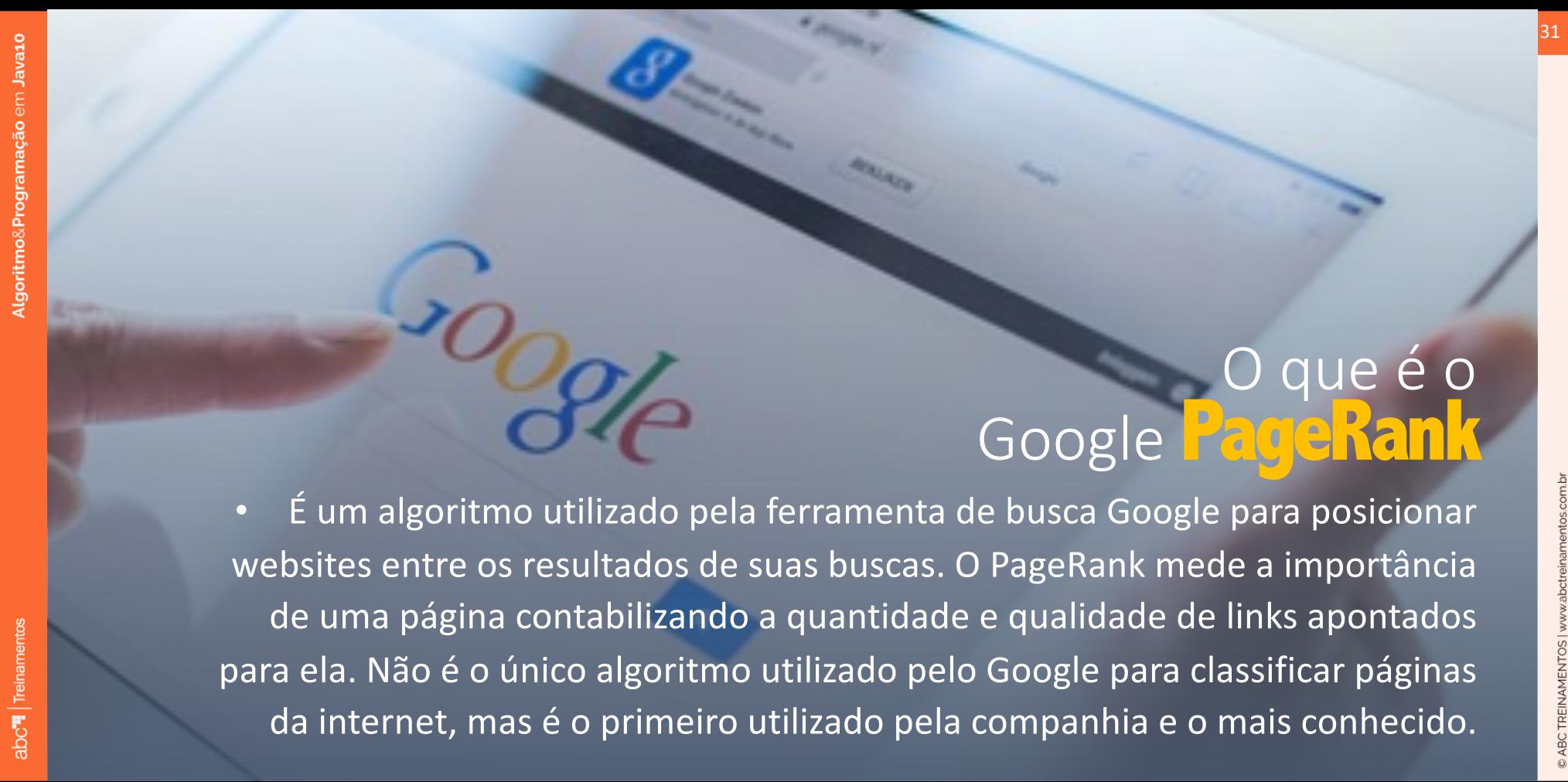
Indexação e Rastreamento

- A indexação possibilita o armazenamento de grande parte, e talvez até todo, o conteúdo da Internet, na enorme base de dados do Google. Os sistemas usados na confecção deste imenso índice levam em consideração aspectos diferentes das páginas: data de publicação, conteúdo, imagens e vídeos, autores, entre outros, para a definição de importância e especificidade do material publicado, de acordo com as definições de busca do usuário.
- O processo de rastreamento e a indexação permitem, que ao realizar a busca, o Google seja capaz de localizar exatamente as páginas adequadas e saiba exatamente onde encontrar o que procura, para que a sua busca seja mais que uma mera associação de palavras.

Como o Google funciona?

Google Bot

- Realiza 03 Atividades:
 - 1) Rastreamento de bilhões de sites na Internet;
 - 2) Cálculo do PageRank desses sites e destacar os mais relevantes para cada palavra-chave, ou seja, quanto maior a nota (PageRank) de um site, melhor o seu site será classificado pelo Google;
 - 3) Quando é feita alguma pesquisa (são bilhões a cada segundo!), o Google Bot visita a base de índice montada anteriormente e exibe os sites relacionados mais bem avaliados pelo PageRank.

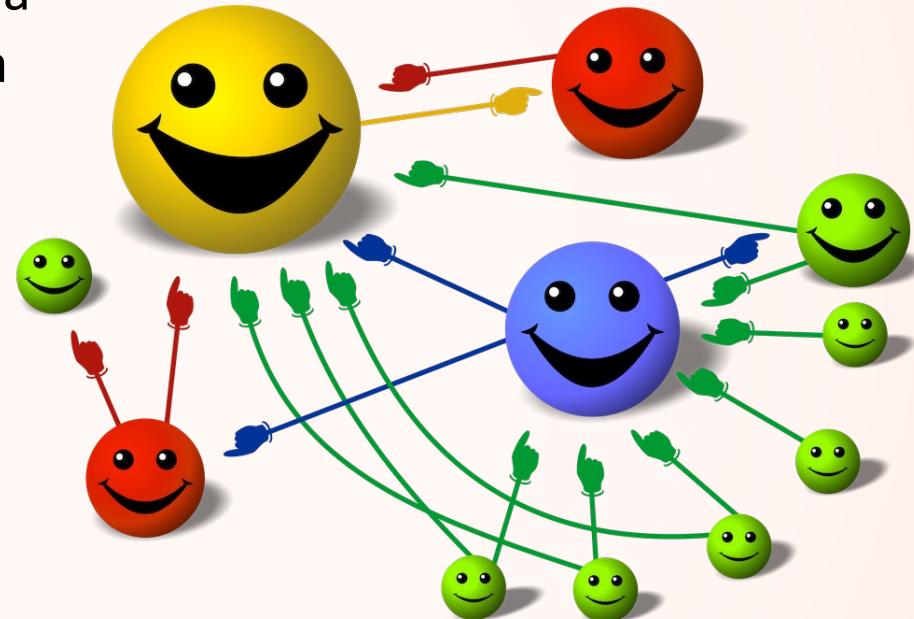


O que é o Google **PageRank**

- É um algoritmo utilizado pela ferramenta de busca Google para posicionar websites entre os resultados de suas buscas. O PageRank mede a importância de uma página contabilizando a quantidade e qualidade de links apontados para ela. Não é o único algoritmo utilizado pelo Google para classificar páginas da internet, mas é o primeiro utilizado pela companhia e o mais conhecido.

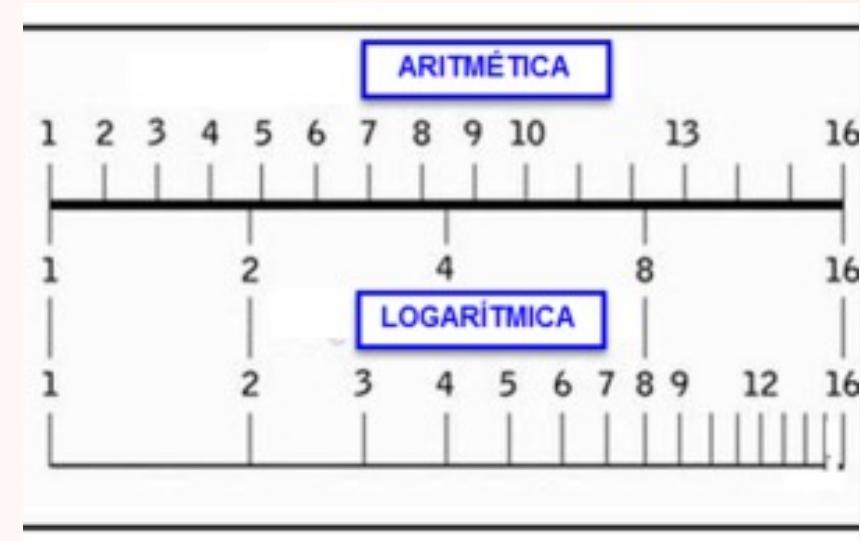
Entendendo *o conceito*

- Basicamente, o PageRank é uma avaliação da **relevância** de uma determinada página. Essa relevância é divulgada ao público em uma escala de 0 a 10, através da Toolbar do Google ou de ferramentas que simulam a ferramenta para obter este número.



Entendendo o conceito

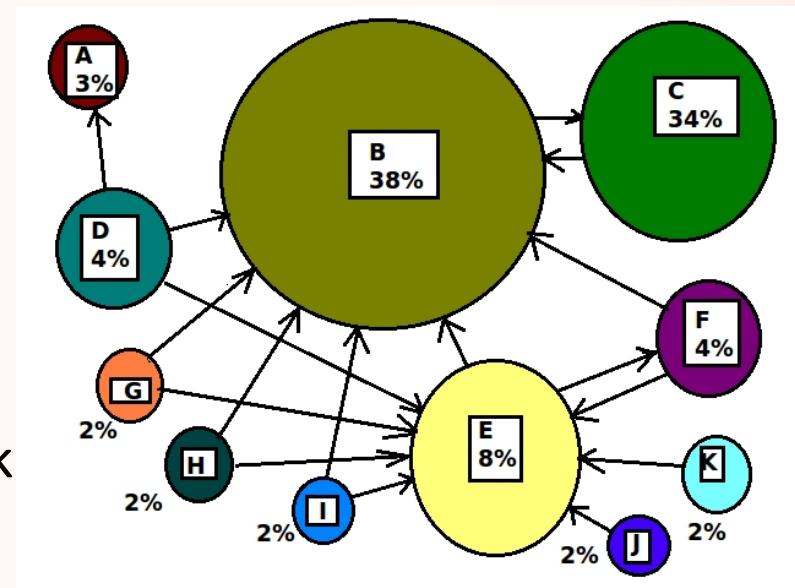
- O Google considera este número com várias casas decimais (para facilitar o cálculo) e utiliza a escala logarítmica para a atribuição de valores. Assim, a diferença entre os números 1 e 2 é muito maior do que os números 3 e 4.



Uma **escala logarítmica** é uma **escala** que usa o logaritmo de uma grandeza em vez da grandeza propriamente dita.

Entendendo o conceito

- O cálculo do PageRank é dado conforme a imagem ao lado. O valor de cada página é dado pelo volume de links recebidos, bem como da autoridade de quem faz essas referência. Assim, páginas com mais autoridade passariam mais PageRank para outras, fazendo com que este seja um ecossistema bem completo.



O processo do PageRank foi patenteado pela Universidade de Stanford nos Estados Unidos sob o número 6.285.999. Somente o nome PageRank é uma marca registrada do Google.

Simplificando o algoritmo

- Imagine uma rede de apenas 4 páginas: **A**, **B**, **C** e **D**. As ligações de uma página a si própria e as ligações múltiplas entre duas páginas são ignoradas.
- O PageRank utiliza valores entre 0 e 1 (representa uma distribuição probabilística) para poder acessar uma determinada página.

Simplificando o algoritmo

- **No primeiro passo** do processo de cálculo iterativo do PageRank, todas as páginas têm o mesmo valor de PageRank.
- *No nosso exemplo de 4 páginas (A, B, C e D), o primeiro passo consiste em atribuir o valor 0,25 de PageRank a cada uma das quatro páginas. Note-se que a soma dos valores de PageRank de todas as páginas é 1.*

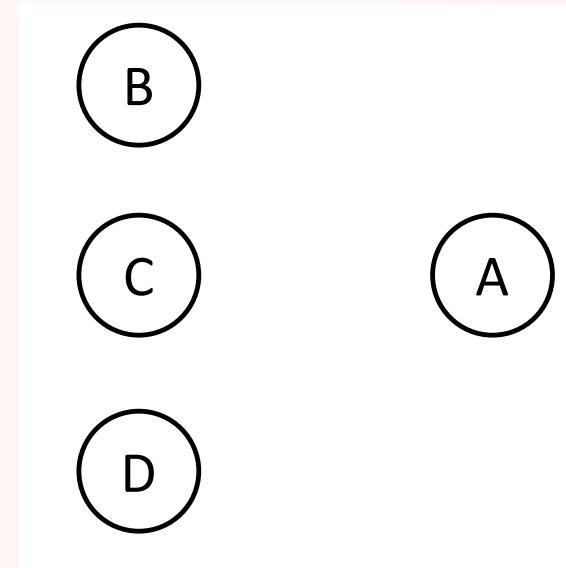


Figura 1- Todas as páginas o mesmo valor de PageRank.

Simplificando o algoritmo

- Numa rede com a configuração da figura ao lado, na **segunda iteração**, cada ligação transfere o valor 0,25 para o PageRank de A, ou seja:

$$PR(A) = PR(B) + PR(C) + PR(D).$$

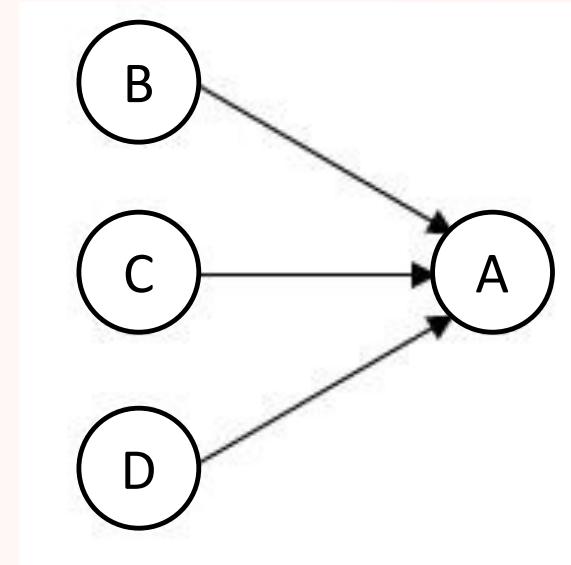


Figura 2- Todas as páginas têm apenas uma referência para a página A.

Simplificando o algoritmo

- No caso da rede ao lado, **na terceira iteração**, o valor de **B** é transferido metade para **A** (0,125) e outra metade para **C** (0,125). Como **D** referencia 3 páginas, o seu valor a transferir é dividido por três, neste caso o PageRank de **A** recebe os seguintes valores.

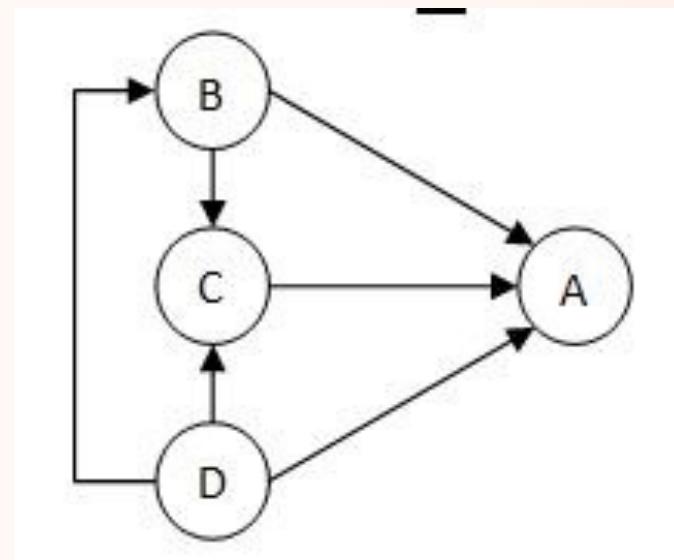


Figura 3- Páginas que referenciam mais de uma página.

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

Simplificando o algoritmo

- Portanto, a contribuição de uma ligação para o PageRank da página referenciada é *igual ao valor de PageRank da página com a ligação, dividido pelo número de ligações que a página contém*. Se representarmos por $L()$ o número de ligações de uma página, podemos reescrever a expressão anterior, para a nossa rede de 4 páginas:

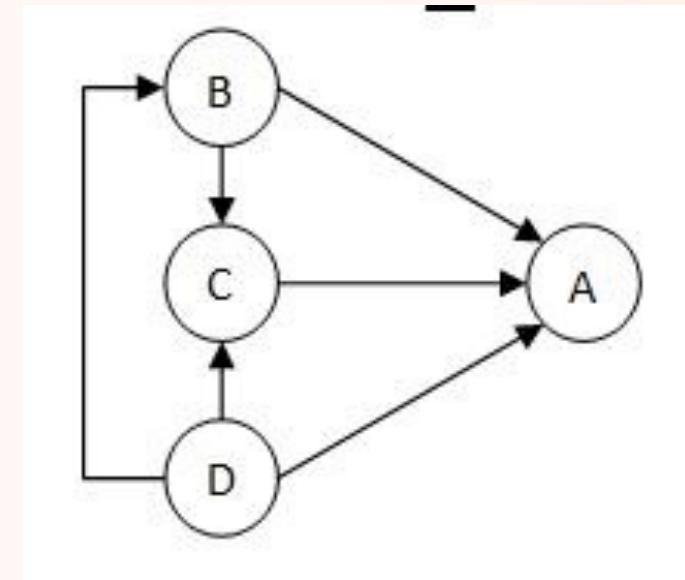


Figura 3- Páginas que referenciam mais de uma página.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

Simplificando o algoritmo

- Generalizando, o valor de PageRank para uma página u pode ser expresso da seguinte forma:
- *O valor de PageRank de uma página u , depende dos valores de PageRank de cada página v contida no conjunto B_u (conjunto de todas as páginas que referenciam u), dividido pelo número de referências $L(v)$ existentes em v .*

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

Simplificando o algoritmo

- Para o cálculo do valor do PageRank, o Google **considera basicamente a quantidade, a qualidade e o contexto de links que a página recebe.**
- As 3 frentes juntas irão informar ao Google o valor exato em que cada link irá reforçar o valor de PageRank de cada página.

Sem qualidade ou contexto, não adianta receber muitos links, que você **nunca será popular** no seu segmento. Sem contexto e quantidade, você **nunca terá a amplificação** necessária. Sem quantidade e qualidade você **nunca terá a força** suficiente para competir com outros concorrentes.

Google Maps



O que é o **Google Maps**

- É um serviço gratuito de pesquisa e visualização de mapas e imagens de satélite da Terra desenvolvido pelo Google.

Como **funciona?**

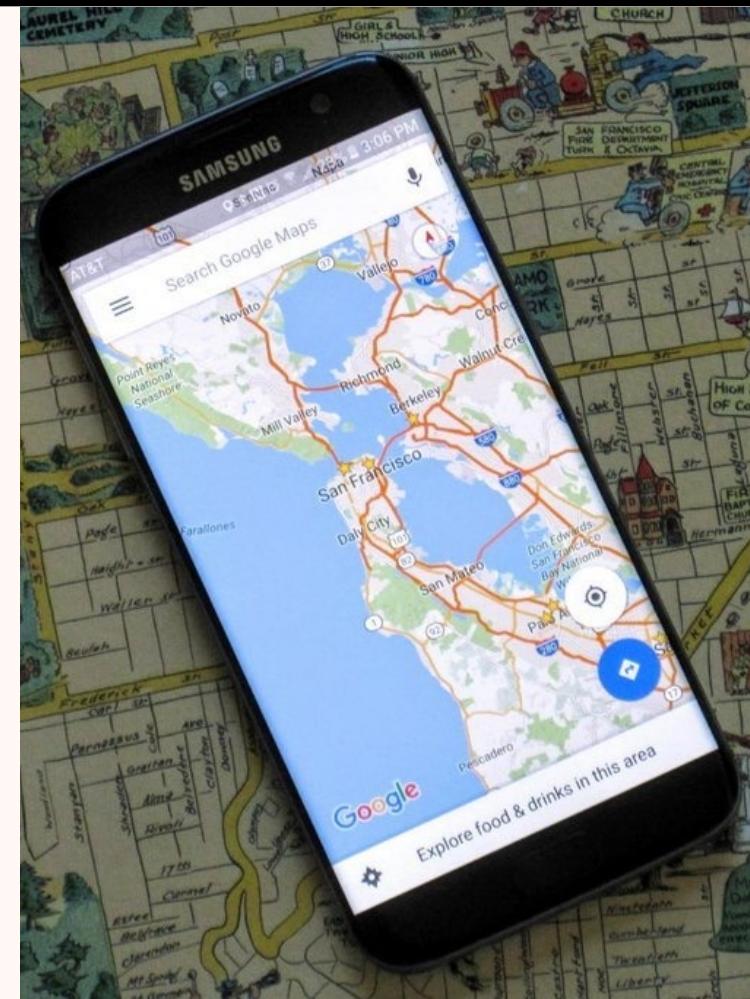
O sistema de mapas do **google** funciona com **vários algoritmos**:

- Algoritmos de **indexação espacial**;
- Algoritmos de **geometria computacional**;
- Algoritmos para **desenhar mapas**;
- Algoritmos para entender as **consultas dos usuários**;
- Algoritmos para **processar sinais de GPS**;
- Algoritmos para **executar o que é chamado de geocodificação**;
- **Algoritmos para o cálculo de rotas.**



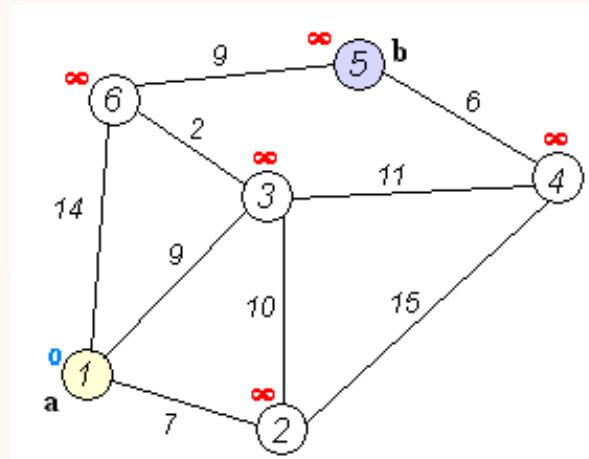
Como **funciona?**

- Vamos focar no **algoritmo de cálculo de rota** que é baseado no **algoritmo Dijkstra**, o principal responsável pelo funcionamento do **Google Maps**.
- **Ele permite realizar o cálculo da rota mais curta entre dois (2) pontos**, não considerando as variáveis do dia a dia, como ruas fechadas, tráfego intenso, entre outros.
- **Este algoritmo também pode ser aplicado para:** Soluções em redes de computadores; Sistemas rodoviários; Ferroviários; Aéreos e vários outros problemas de logística.



Como funciona?

- O algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959, soluciona o problema do caminho mais curto.



Como **funciona?**

```
1 function Dijkstra(Graph, source):
2
3     create vertex set Q
4
5     for each vertex v in Graph:
6         dist[v] ← INFINITY
7         prev[v] ← UNDEFINED
8         add v to Q
9
10    dist[source] ← 0
11
12    while Q is not empty:
13        u ← vertex in Q with min dist[u]
14
15        remove u from Q
16
17        for each neighbor v of u:
18            alt ← dist[u] + length(u, v)
19            if alt < dist[v]:
20                dist[v] ← alt
21                prev[v] ← u
22
23    return dist[], prev[]
```

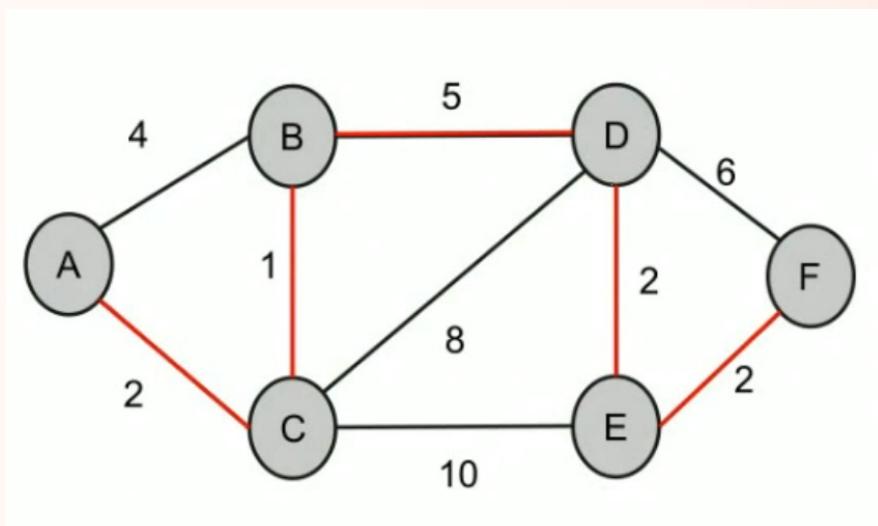
Como **funciona?**

- O algoritmo de Dijkstra (e suas variações) é o que faz o Google Maps funcionar. Traçar rotas com engenhosidade!



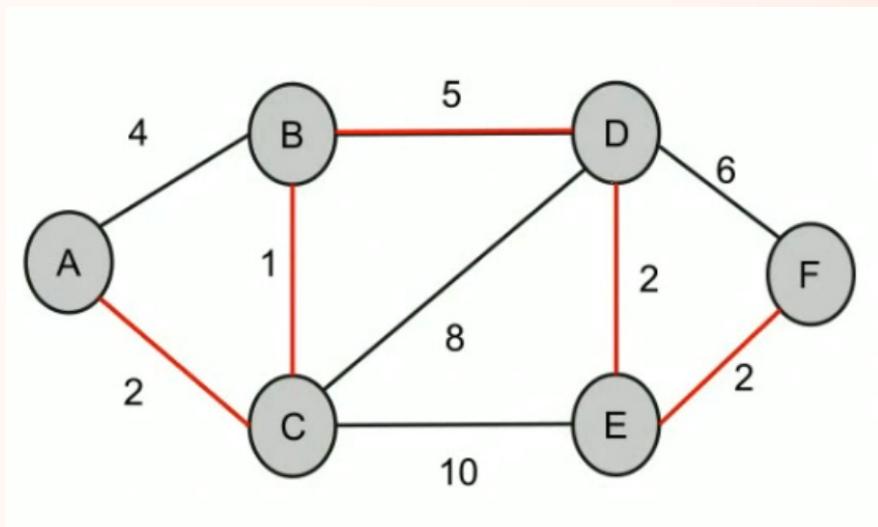
Como funciona?

- Como o algoritmo de Dijkstra ajuda o Google Maps a achar a melhor rota? Veja o gráfico ao lado. Os números são os pesos de cada conexão. Um peso pode ser uma distância simples ou qualquer custo associado ao percorrido de uma rota.
- “A” será o nosso ponto de partida. Os pontos vizinhos de “A” (“B”, “C”, “D”, “E”, “F”) são os destinos a serem atingidos.
- **OBJETIVO:** Sair de “A” e chegar até “F” no menor tempo/custo possível.



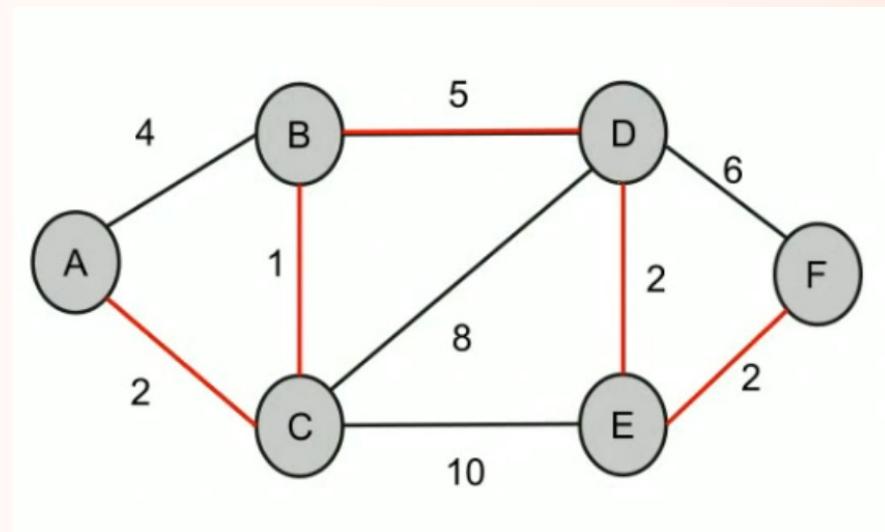
Como **funciona?**

- **No primeiro passo**, o algoritmo identifica quais são os pontos vizinhos à “A” e quais os seus pesos. O ponto “C” possui o menor peso (**2 unidades**);
- **Segundo passo** é achar o proximo ponto cuja a soma das unidades de distâncias anterior e as seguintes sejam as menores possíveis, logo, a distância do ponto “B” ao ponto “C” é de **3 unidades**.



Como **funciona?**

- **O terceiro passo**, será ir do ponto “B” (**3 pontos**) em direção ao ponto “D”, cujo total será de **8 unidades** de distância;
- **No passo seguinte**, identifica-se que o menor percurso a seguir será do ponto “D” ao ponto ”E”, pois este totalizará **10 unidades** de distância pecorrida até agora.



- **O passo final** será do ponto “E” ao “F”, cuja a somatória de pontos será de **12 unidades** de distância.

NETFLIX



O que é a **NETFLIX**

- Netflix é uma provedora global de filmes e séries de televisão via streaming, atualmente com mais de 100 milhões de assinantes. Fundada em 1997 nos Estados Unidos, a empresa surgiu como um serviço de entrega de DVD pelo correio.

Mas por que tanto *sucesso?*

- Uma das principais tecnologias que fazem a NETFLIX manter os seus espectadores colados na sua plataforma, é um **Algoritmo que aconselha filmes, documentários e séries aos usuários.**
- O Algoritmo é tão preciso que, além de sugerir os *blockbusters* que mais se encaixam nos gostos dos seus usuários, apostava em títulos pouco conhecidos, mas que pareciam ser perfeitos para os seus clientes.



*Mas por que tanto **sucesso?***

- Este Algoritmo possibilita oferecer um catálogo variado de produtos sob medida para os seus mais de **125 milhões de assinantes em todo mundo**, permitindo à NETFLIX investir dinheiro minimizando os seus riscos de não aceitação.



Como **funciona?**

- Este Algoritmo é baseado em **seus gostos e preferências**.
- Os assinantes são agrupados em “*clusters*” (ou comunidades globais) definidos quase que exclusivamente por seus gostos em comum.
- **As páginas iniciais dos usuários** de cada “cluster” são preenchidas por uma pequena fatia de conteúdo que a maior parte do grupo já assistiu ou gosta.



Como **funciona?**

- Logo que o usuário começa a usar o Netflix, ele ou uma pessoa na Groelândia podem ter a “home” bem parecidas.
- À medida que o algoritmo vai reconhecendo os **seus gostos e as suas preferências**, tudo muda.
- Quanto mais programas este usuário assiste, mais ele se encaixa em uma comunidade, ajudando a Netflix a sugerir para ele aquilo que as pessoas com gostos semelhantes assistem.



Como **funciona?**

- Se muitas pessoas que assistem “comédia” também assistem determinado filme de ação, esse filme vai ser sugerido para quem assiste a mesma fatia de produtos.



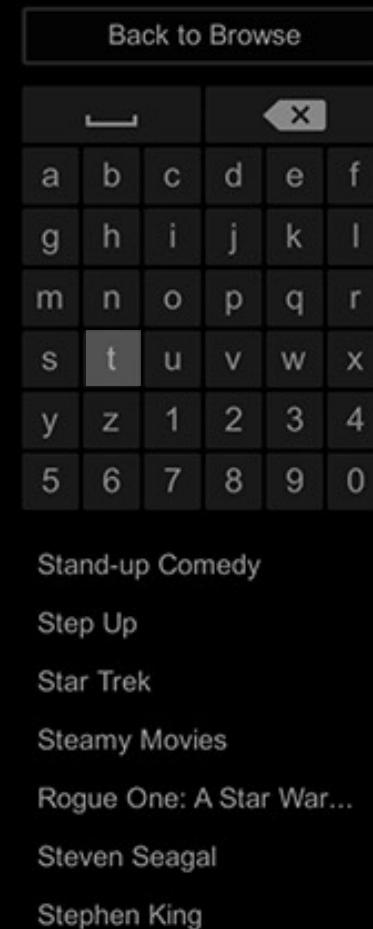
Como **funciona?**

- O **número de estrelas** que um usuário atribui a um filme, uma série ou a um documentário faz com que o algoritmo aprenda mais sobre ele, permitindo sugerir/bloquear os mesmos de acordo com o seu gosto.
- Atores, diretores e produtores também são levados em conta. Caso um usuário assiste a um filme do ***Tom Cruise*** e no final dê apenas duas estrelas, é muito provável que outro filme do ator não aparece tão cedo na sua *home*.



Como **funciona?**

- Agora, se o usuário constantemente pesquisa por nomes de diretores ou filmes consagrados, a Netflix vai lembrar disso e sugerir vários filmes do Woody Allen – a não ser que o cliente tenha atribuido uma nota péssima para *Meia Noite em Paris*.
- Tamanho é o sucesso do algoritmo da NETFLIX que o seu CEO, Reed Hastings, chegou a afirmar que “o único rival do algoritmo é o sono”.



st



E-COMMERCE

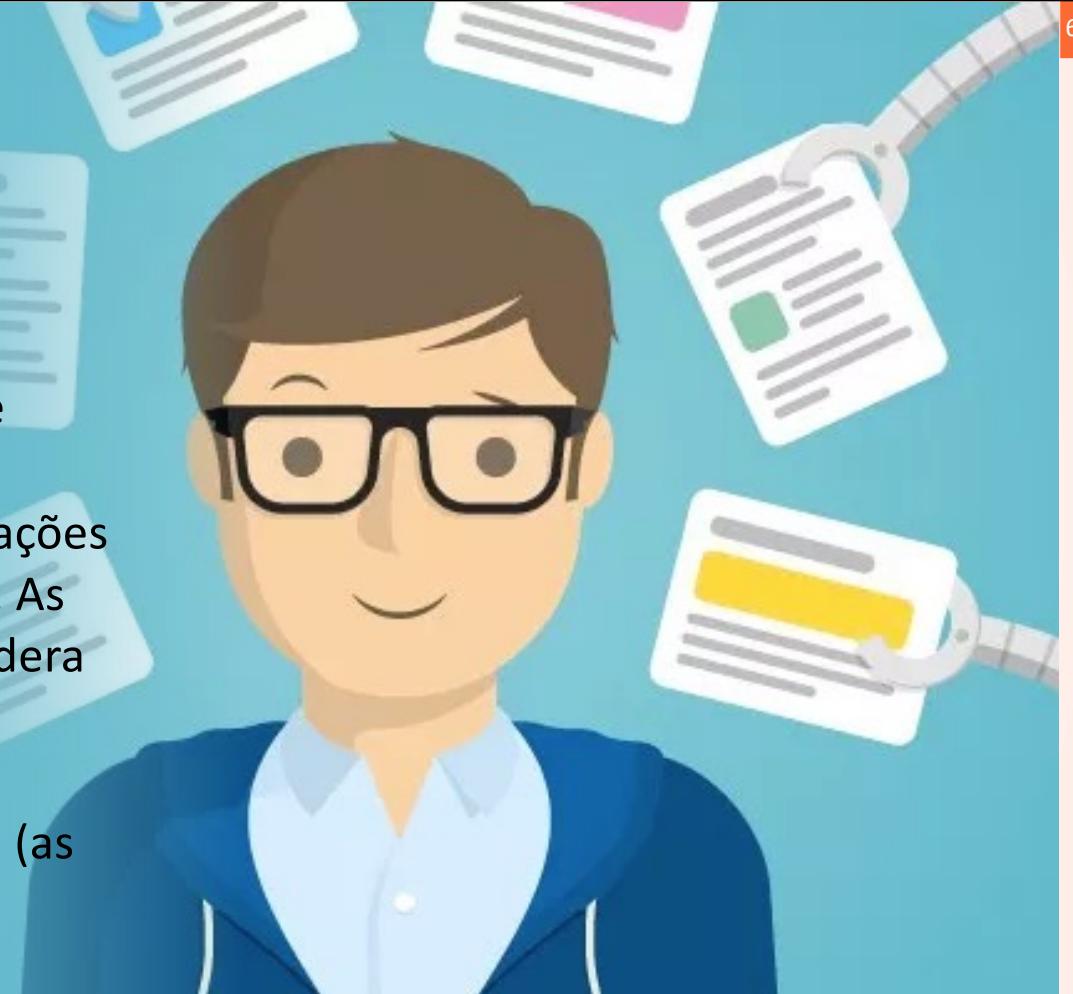
O que todas essas lojas têm em comum?



Simples! Todas utilizam os Algoritmos
Sistemas de Recomendação.

O que são os Algoritmos *Sistemas de Recomendação*?

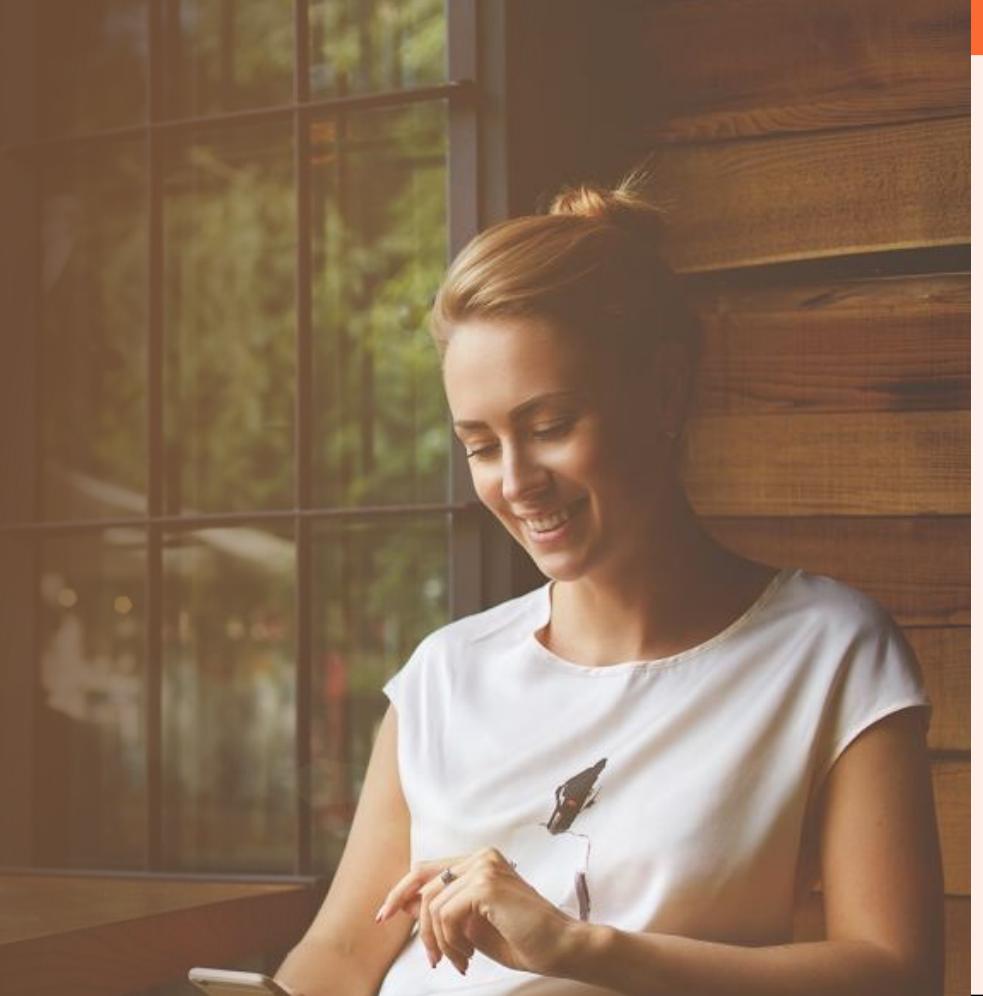
São um conjunto de algoritmos que utilizam técnicas de Aprendizagem de Máquina (AM) e Recuperação da Informação (RI) para gerar recomendações baseadas em algum tipo de filtragem. As mais comuns são: colaborativa (considera a experiência de todos os usuários), baseada em conteúdo (considera a experiência do usuário alvo) e híbrida (as duas abordagens são consideradas).





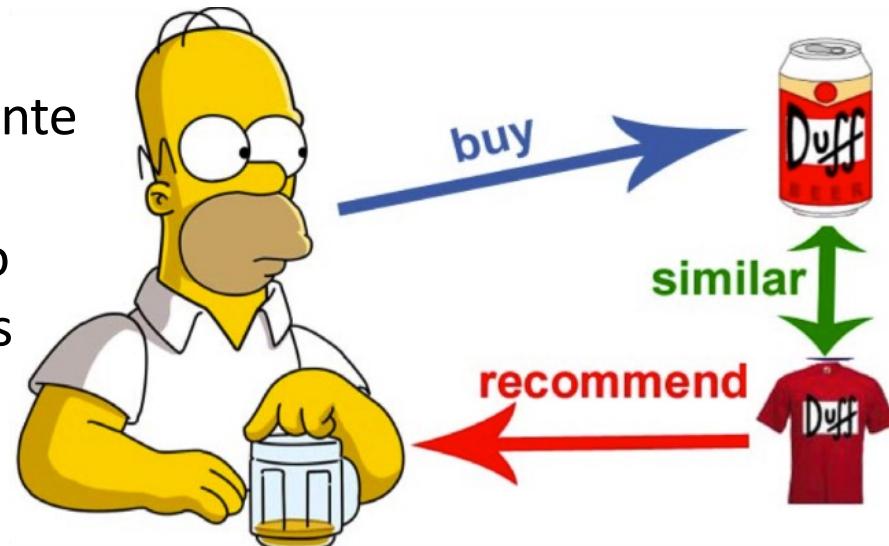
O *sistema de recomendação* da Amazon é baseado em um número simples de elementos. O que o usuário comprou, quais itens estão guardados em sua "lista de desejos", itens que qualificou e gostou e quais outros clientes que viram e compraram aqueles produtos.

Amazon chama este sistema de "filtro colaborativo item a item". Isso é muito usado para trazer a melhor experiência de compra para quem retorna à loja.



Algoritmo de Filtragem **colaborativa**

Os algoritmos de recomendação de Filtragem Colaborativa são amplamente utilizados no ramo de e-commerce e também em Redes Sociais. O usuário recebe recomendações baseadas nas avaliações passadas de todos os usuários, coletivamente.



Por exemplo: um fã de televisores pode receber várias *recomendações* de tv's conforme seu gosto; uma mãe recebe *recomendações* (nas mesmas páginas) de produtos para crianças.

Algoritmo de Filtragem **colaborativa**

As avaliações dos itens podem ser feitas de diversas maneiras. Como uma escala de pontuação baseado em estrelas, um ícone caracterizando gostei e não gostei (avaliação binária), ou até mesmo através da postagem de comentários sobre o item.



Smart TV Samsung 32" LED UN32J4300 Wide HD HDMI/USB Preto

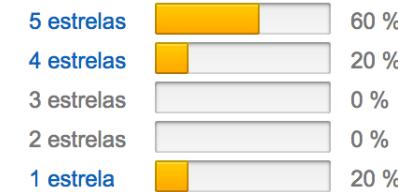
por SAMSUNG

R\$ 1.641⁵⁴ R\$ 1.849,00
R\$ 145,35 de frete

10 opções de compra a partir de R\$1.199,00



3.8 de 5 estrelas



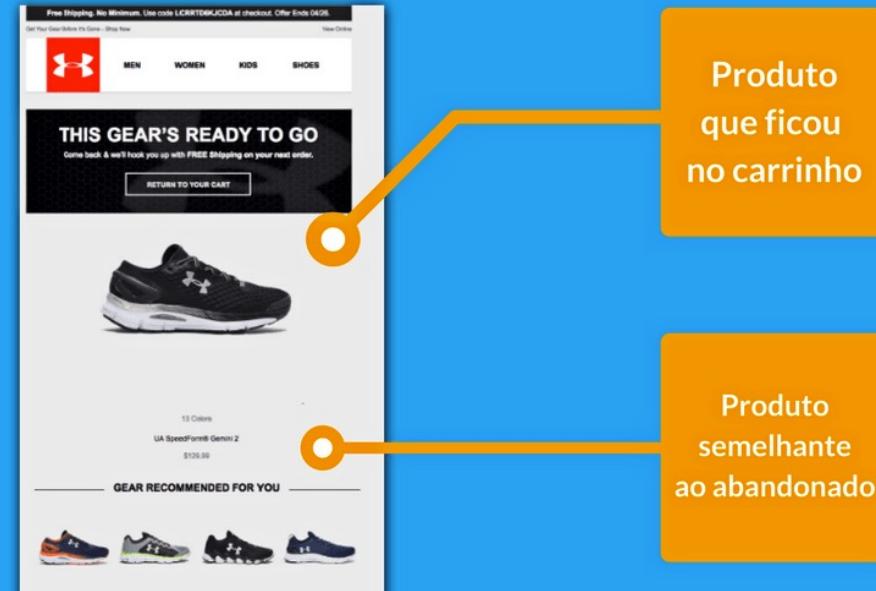
[Ver todas as 5 avaliações ▶](#)

GXZD

9 opções de compra a partir de R\$1.149,00

Algoritmo de Filtragem por conteúdo

Já os SISTEMAS DE RECOMENDAÇÃO que utilizam Filtragem Baseada em Conteúdo fazem a sugestão de itens que sejam semelhantes aos que o usuário demonstrou interesse no passado, e/ou sobre as configurações de preferências do usuário. Assim, as recomendações são personalizadas para cada usuário.



Algoritmo de **similaridade de item/usuário**

Um dos principais conceitos utilizados nos SISTEMAS DE RECOMENDAÇÃO é a similaridade. Esta consiste em descobrir itens similares aos que o usuário já adquiriu ou usuários mais similares entre si.

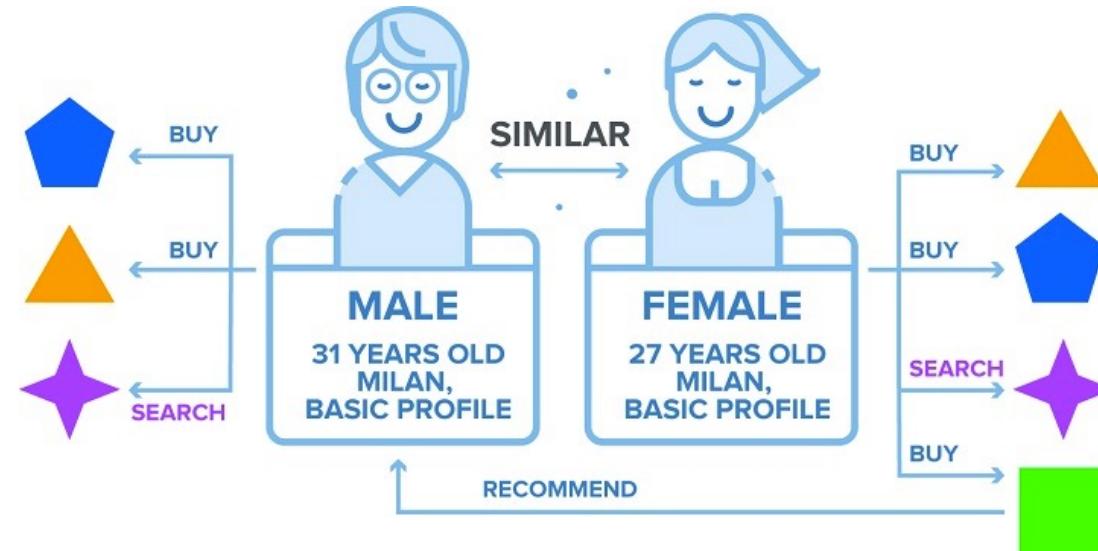
Relacionado com as suas compras em Acessórios para Veículos [Ver compras](#)



Utilizando as avaliações e preferências de todos os usuários, pode-se calcular a similaridade entre esses usuários e o usuário alvo.

Algoritmo de **similaridade de item/usuário**

A similaridade dos itens pode ser determinada pela busca booleana, métodos probabilísticos e também modelos vetoriais. Nessa abordagem, os algoritmos de Classificação mais utilizados são:



Vizinho K-mais-próximo; Árvores de Decisão; Classificador de Bayes; SVM e Redes Neurais.

NISSAN
ProPILOT



O que é o **ProPILOT?**

- É uma tecnologia revolucionária de acionamento autônomo projetada para uso em rodovias em tráfego de pista única, permitindo que o condutor mantenha a pista alinhada com os fluxos de tráfego, controlando a direção, o acelerador e os freios, reduzindo o estresse do dia dia no trânsito.

E como **funciona?**

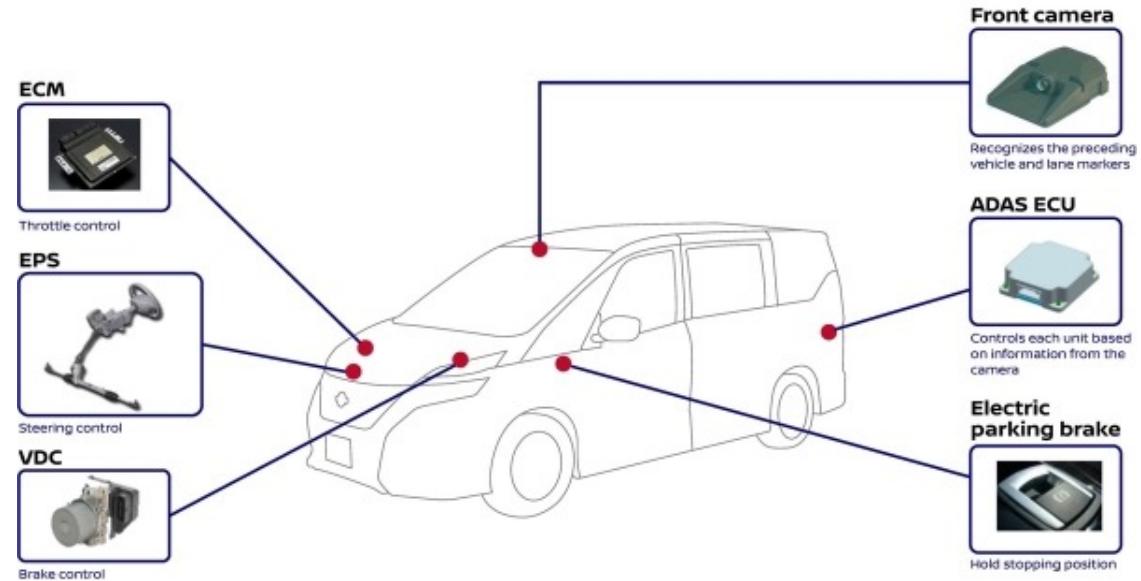
Através de uma tecnologia avançada de processamento de imagem, o algoritmo do sistema **ProPILOT** do carro comprehende as situações da estrada e do trânsito e executa uma direção precisa, permitindo que o veículo funcione naturalmente.

A tecnologia ProPILOT é extremamente fácil de usar, graças a um interruptor no volante que permite ao motorista ativar e desativar facilmente o sistema. A interface fácil de entender e ajustar para dirigir do ProPILOT inclui um monitor pessoal que mostra o status de operação.



E como *funciona*?

O ProPILOT usa duas tecnologias inovadoras, a primeira é a miniatura High-Spec Laser Scanner, que permite que o veículo circule pelos espaços mais apertados. A outra é uma câmera com visão 360° posicionada em 08 pontos estratégicos, que guia o carro através de cruzamentos e outras situações complexas na direção.





E como *funciona*?

Havendo perigos na condução, o módulo de controle é informado, fazendo com que os algoritmos do sistema executem pequenos ajustes e corrijam com segurança, por exemplo, distâncias entre veículos à frente e/ou mantenha o veículo dentro das faixas da pista e caso se aproxime de outra pista, faz as alterações necessárias para corrigir o percurso, sempre mantendo uma distância seguro em relação aos veículos ao redor.

E como *funciona?*

Outro algoritmo identifica se há fadiga no condutor, e, caso necessário, inicia o assistência de condução, reduzindo gradativamente o trabalho do condutor, corrigindo e mantendo a direção e velocidade do veículo, tornando a condução menos cansativa, o que ajuda a aumentar a segurança.



E como *funciona?*

Em caso de extrema necessidade, caso seja percebida pelo sistema perigo iminentente, outro algoritmo realiza a parada total do veículo.

Há também um algoritmo de pressão no volante do carro, caso o condutor não esteja segurando de forma adequada, o sistema emite sinais para lembrar o usuário de segurar com mais firmeza o volante do veículo.



E como *funciona?*

77

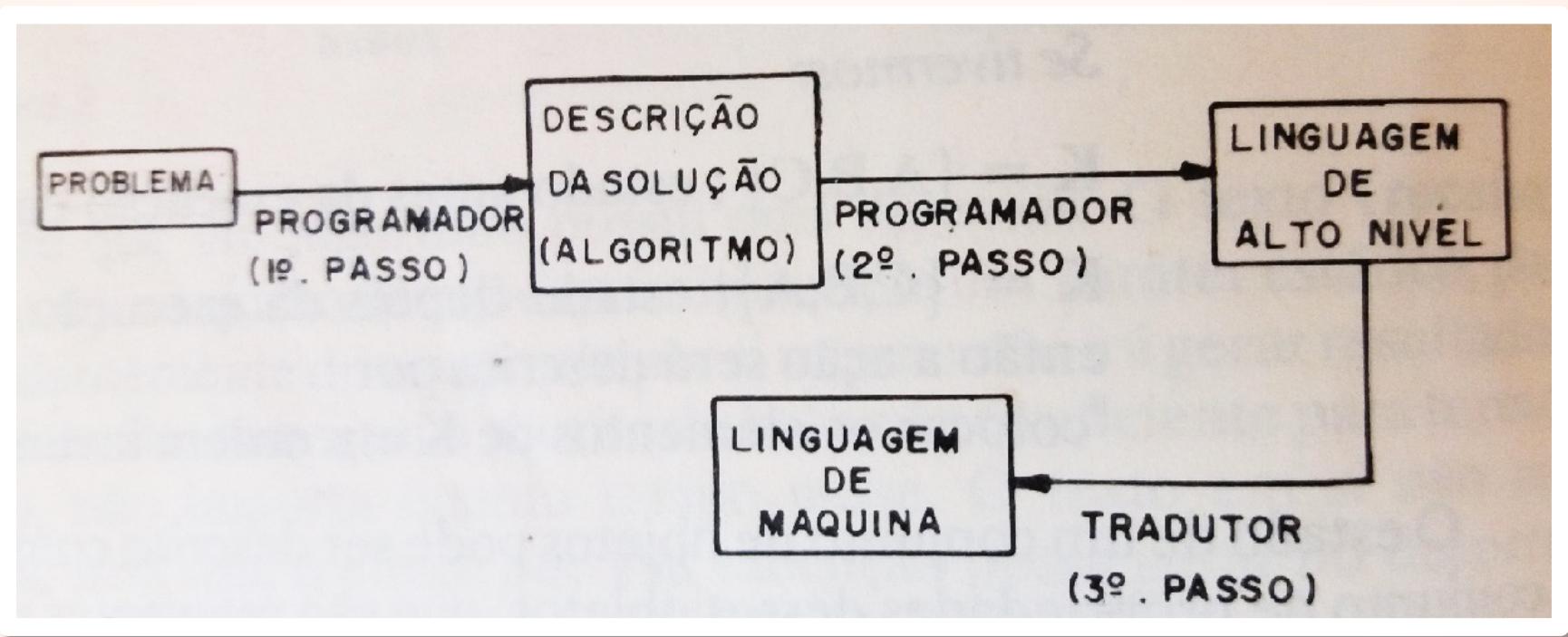


<https://www.youtube.com/watch?v=ImarUTZIEb0>

Etapas de Desenvolvimento

Passos Necessários para a Transformação de um Algoritmo em um **Programa (Software)**

79

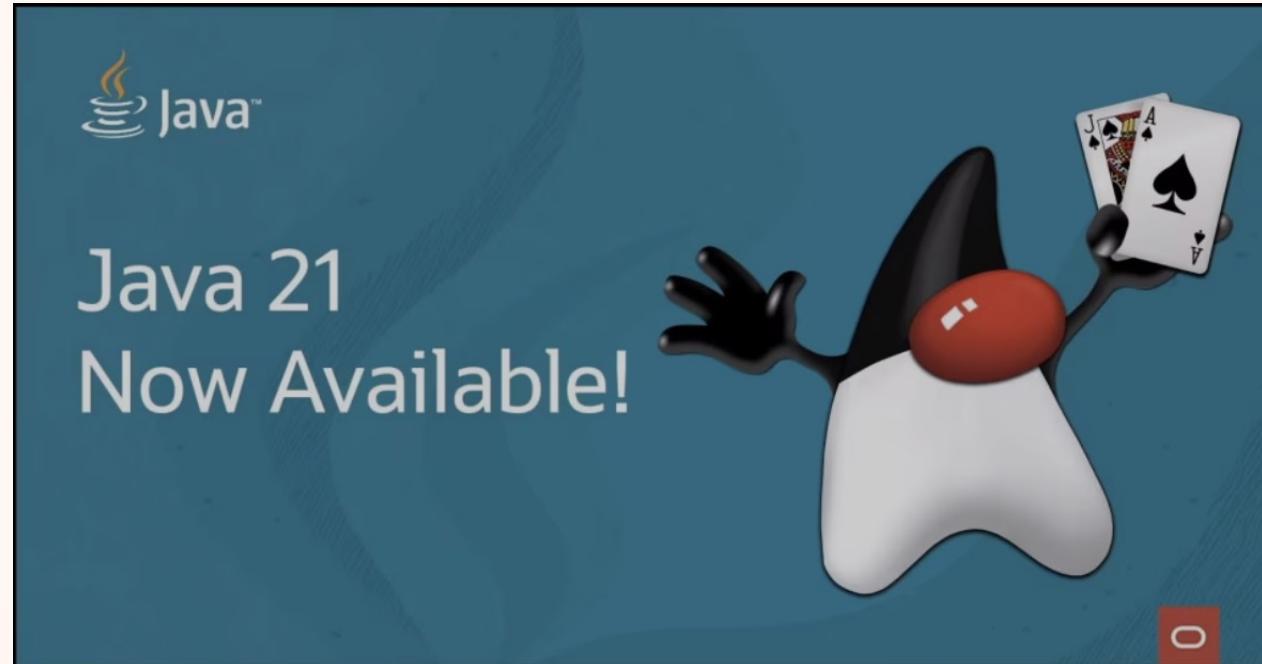


Fonte: Wilson Silva Pinto

**Instalação do JAVA 21
&
ECLIPSE 2023-09**

Ambiente de *Desenvolvimento*

81



<https://www.oracle.com/br/java/technologies/downloads/>

Ambiente de Desenvolvimento



<https://www.eclipse.org/downloads/packages/>

Ambiente de *Desenvolvimento*

```
jshell> int a = 5;  
a ==> 5  
  
jshell> int b = 10;  
b ==> 10  
  
jshell> int sum = a + b;  
sum ==> 15  
  
jshell> System.out.println("A soma é: " + sum);  
A soma é: 15
```

Arquitetura Computacional na Qual Construímos Algoritmos

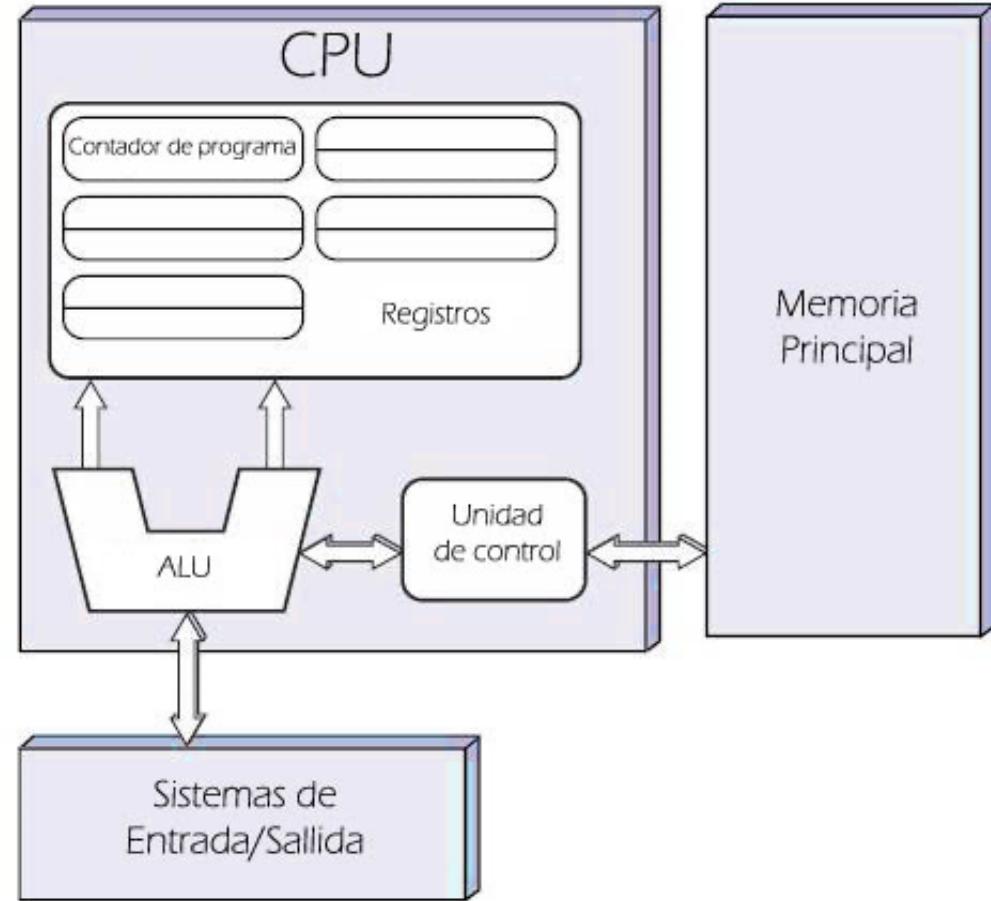
ARQUITETURA DE JOHN VON NEUMANN

- A Arquitetura de John von Neumann é uma arquitetura de computador que se caracteriza pela possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas;
- Esta arquitetura é um projeto modelo de um computador digital de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de armazenamento ("memória") para comportar, respectivamente, instruções e dados;
- Concebida a partir de 1946 → A base de todas as máquinas atuais.



ARQUITETURA DE JOHN VON NEUMANN

- A máquina proposta por Von Neumann reúne os seguintes componentes:
 - (i) uma memória;
 - (ii) uma unidade aritmética e lógica (ALU);
 - (iii) uma unidade central de processamento (CPU), composta por diversos registradores;
 - (iv) uma Unidade de Controle (CU), cuja função é buscar um programa na memória, instrução por instrução, e executá-lo sobre os dados de entrada.



Memória e o Conceito de Variáveis

- O conceito de **MEMÓRIA** da Arquitetura de John Von Neumann é o local onde as instruções a serem executadas e os dados a serem manipulados são armazenados (temporariamente ou não).
- A **MEMÓRIA** pode ser enxergada como um “armário” composto por muitas “gavetas” que são chamadas de **VARIÁVEIS**.



Fonte: Gustavo Furtado de Oliveira Alves - <https://dicasdeprogramacao.com.br/o-que-e-variavel-e-constante/>

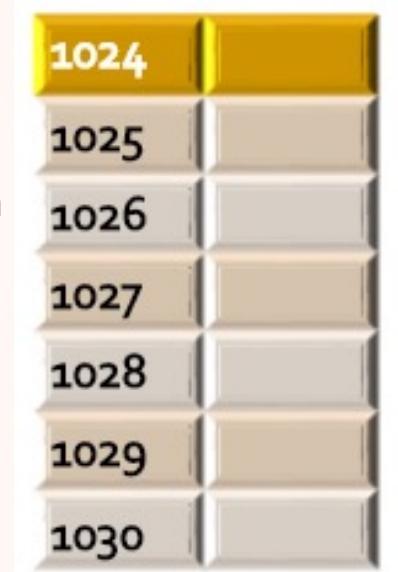
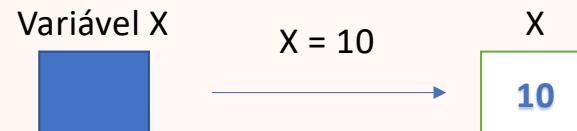
Memória e o Conceito de Variáveis

- Todas as **VARIÁVEIS (gavetas)** alocam uma espaço na memória do computador.

Cada célula de memória é caracterizada pelo seu **endereço** e seu **conteúdo**.

- Para facilitar a sua manipulação, foram criados os **identificadores** que fazem referência a esse espaço de memória.

EXEMPLO:



Memória e o Conceito de Variáveis



Fonte: Joaquim José Hangalo - <http://slideplayer.com.br/slide/3533490/>

Variáveis

VARIÁVEIS

- Declarar uma Variável significa reservar uma “gaveta” na memória, atribuir-lhe um rótulo (identificador);
- O Tipo de Dados com o qual a variável é declarada serve para especificar qual é o conjunto de valores que ela pode armazenar;
- Esses valores são passados para variáveis através das **operações de atribuição e leitura**.

SINTAXE BÁSICA

```
Declaração de Variáveis  
Inteiro: X;  
Real: Soma;  
Caracter: Nome;  
Lógico: Sexo;
```

Variáveis – Tipos de Dados

REAL

- Pode armazenar qualquer valor pertencente ao conjunto dos Números Reais;

Exemplo: 3,14; -8,74; 10.000; etc.

INTEIRO

- Pode armazenar qualquer valor pertencente ao conjunto dos Números Inteiros;

Exemplo: 3; -8; 100; etc.

CARACTER

- Pode armazenar apenas 1 elemento pertencente a um dos conjuntos abaixo:

Exemplo: {A..Z}, {0..9}, {*,&, ?, +, -, %, etc.}

LÓGICO

- Pode armazenar apenas as constantes Verdadeiro ou Falso.

Exemplo: V ou F.

Variáveis – Identificadores

- Servem para nomear as variáveis;
 - Podem conter letras e/ou dígitos, “_” e “\$”;
 - Não podem ser iniciados por dígito;
 - Não podem ser palavras reservadas;
 - Não tem tamanho máximo.
- Exemplo de nomes de **IDENTIFICADORES VÁLIDOS**:
- prova, nota, total_geral, nome, endereço, etc.
- Exemplo de nomes de **IDENTIFICADORES INVÁLIDOS**:
- 1prova, total geral, class, 5X, E(13), A:B, X-Y, P&AA, etc.

Variáveis – Atribuição de Valores

- Um comando de atribuição permite-nos fornecer um valor a uma variável (guardar um objeto na gaveta), em que o tipo do dado deve ser compatível com o tipo da variável, isto é, somente podemos atribuir um valor lógico a uma variável capaz de comportá-lo.
- EXEMPLO:**

TOTAL ← 78;



Declaração de Variáveis

```
Inteiro: TOTAL;  
...  
TOTAL ← 78;
```

Variáveis – Atribuição de Valores

EXEMPLO

Declaração de Variáveis

Inteiro: X;

Real: Soma;

Caracter: Nome;

Lógico: Sexo;

Início

X ← 10;

Soma ← 15,57;

Nome ← "Antonio";

Sexo ← M;

Variáveis em JAVA – Atribuição de Valores

SINTAXE BÁSICA

```
<TIPO_VARIABEL> nomeVariavel;
```

Exemplos:

```
int x;  
double soma;  
char nome;  
boolean sexo;
```

A ATRIBUIÇÃO É REALIZADA COM O OPERADOR ‘=’

- ‘=’ serve apenas para atribuição – não pode ser usado em comparações (que usa ‘==’)!
• Copia o valor da variável ou constante do lado direito para a variável do lado esquerdo.
• **EX: x = 10; // copia a constante inteira 10 para x**

Variáveis em JAVA – Atribuição de Valores

- As declarações podem ser exibidas em qualquer posição do código-fonte.

```
int x, y;
float z = 3.144f;
double w = 3.1415;
boolean verdade = true;
char c, d;
c = 'A';
d = '\u0013';
x = 6;
y = 1000;
```

INICIALIZAÇÃO PADRÃO JAVA

- variáveis numéricas com **0**;
- variáveis booleanas com **false**;
- outras variáveis com **null**.

Variáveis – Tipos de Dados em JAVA

	TIPO	MEMÓRIA FAIXA	
Lógico	boolean	1 bit	true ou false
Texto	char	2 bytes	\u0000 a \uFFFF
	String	variável	\u0000 a \uFFFF em cada localização
Integral	byte	1 byte	-128 a 127
	short	2 bytes	-32,768 a 32,767
	int	4 bytes	-2,147,483,648 a 2,147,483,647
	long	8 bytes	-9,223,372,036,854,775,808L a 9,223,372,036,854,775,807L
Ponto flutuante	float	4 bytes	aproximadamente +/-3.40282347E+38F (7 dígitos decimais significantes)
	double	8 bytes	aproximadamente +/-1.79769313486231570E+308 (15 dígitos decimais significantes)

Variáveis em JAVA

SINTAXE BÁSICA

- **LÓGICO**

```
boolean terminou;  
terminou = true;  
terminou = false;
```

- **TEXTO**

- **char:** representa um caracter Unicode de 16 bits (exemplos: 'a', 'M', '\t');
- **String:** representa uma seqüência de caracteres.

```
char opcao;  
opcao = 's'; opcao = 'n' ;  
String frase;  
frase = "Ordem e Progresso";
```

Variáveis em JAVA

SINTAXE BÁSICA

- **INTEIRO**
- Representações: 2 (decimal) / 077 (octal) / 0xBA (hexadecimal)

byte	8 bits	$2^7 \dots 2^7-1$
short	16 bits	$2^{15} \dots 2^{15}-1$
int	32 bits	$2^{31} \dots 2^{31}-1$
long	64 bits	$2^{63} \dots 2^{63}-1$

```
byte index = 50;
short soma = 2000;
int num_carros = 5;
long valor = 0XDADAL;
```

Variáveis em JAVA

SINTAXE BÁSICA

TIPOS PONTO FLUTUANTE (REAL)

- float (32 bits)
- double (64 bits)
- Representações: 3.14 / 6.02E23 / 2.718F / 123.4E+306D

```
float pi = 3.1415f;  
double pi = 3.1415;
```

Constantes

- Uma Constante é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução do algoritmo (programa);
- Por exemplo, se tivermos 04 avaliações no decorrer de um curso, a média aritmética das notas será dada pela expressão:
$$(n1+n2+n3+n4) / 4$$
- As notas são variáveis, mas o denominador (4) é invariável, podendo ser declarado por uma constante (num_avaliacoes).

$$(n1+n2+n3+n4) / num_avaliacoes$$

SINTAXE BÁSICA

Declaração de Constantes

```
num_avaliacoes = 4;
```

Constantes em JAVA

SINTAXE BÁSICA

```
final int num_avaliacoes = 4;
```

Exercícios

1) Dada as proposições abaixo, identifique o Tipo de Dado a ser utilizado:

- a) O saldo bancário do André é de R\$4.531,50;
- b) João possui 3 irmãos;
- c) Constava na prova: “Use somente caneta!”;
- d) A porta pode estar “aberta” ou “fechada”;
- e) O número de avaliações é 4.

2) Faça a Declaração das Variáveis/Constantes definidas no exercício anterior. Faça também a atribuição dos seus valores.

3) Faça a Declaração das Variáveis/Constantes definidas no exercício anterior na Linguagem Java. Faça também a atribuição dos seus valores.

4) As informações de um Aluno (matrícula, nome, endereço, nota, sexo) precisam ser utilizadas por um programa. Faça a declaração dessas variáveis em linguagem Algorítmica e em Linguagem Java e atribua valores a esses dados.

Operadores Matemáticos, Relacionais e Lógicos

Operadores

- Representam todos os tipos de operações que podem ser feitos nos dados para gerar as informações desejadas.
- Podem ser de 03 Tipos:
 - **MATEMÁTICOS**
 - **RELACIONAIS**
 - **LÓGICOS**

Operadores Matemáticos

- São usados para realizar as operações matemáticas nos dados do Tipo INTEIRO ou REAL.
- (+) Soma
- (-) Subtração
- (*) Multiplicação
- (/) Divisão
- (**) Exponenciação
Exemplo: $5^{**}2 = 25$
- (%) Resto
Exemplo: $5 \% 2 = 1$

Operadores Matemáticos

- Ordem de prioridade: exponenciação, multiplicação/divisão e soma/subtração.

Exemplo:

$$\rightarrow 5 + 8 * 3 = 29$$

$$\rightarrow (5 + 8) * 3 = 39$$

$$\rightarrow 5 + 9 + 7 + 8/4 = 23!$$

$$\rightarrow 1 - 4 * 3/6 - 3^{**}2 = -10$$

Operadores Relacionais

- São usados para comparar variáveis ou expressões;
- Sempre produzem um resultado lógico (Verdadeiro ou Falso).

- **(==) Igual**
- **(!=) Diferente**
- **(<) Menor**
- **(<=) Menor ou Igual**
- **(>) Maior**
- **(>=) Maior ou Igual**

Operadores Relacionais

- Exemplo:

→ $2 * 4 == 24/3$ (V)

→ $(15\%4) < (19\%6)$ (F)

→ $2 + 8\%7 >= 3 * 6 - 15$ (V)

Operadores Lógicos

- São usados para avaliar expressões lógicas;
 - Sempre produzem um resultado lógico (Verdadeiro ou Falso).
-
- **(&&) E**
 - **(||) OU**
 - **(!) Negação**

TABELAS-VERDADE

Negação

P	$\sim P$
V	F
F	V

Conjunção

P	Q	$P \wedge Q$
V	V	V
V	F	F
F	V	F
F	F	F

Disjunção

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

Operadores Lógicos

- Exemplo:

→ **2 < 5 E 15/3 == 5 (V)**

→ **2 < 5 OU 15/3 == 5 (V)**

→ **NÃO (2 < 5) OU NÃO (15/3 == 5) (F)**

Operadores em JAVA

OPERADOR	FUNÇÃO	OPERADOR	FUNÇÃO
<code>+</code>	Adição	<code>~</code>	Complemento
<code>-</code>	Subtração	<code><<</code>	Deslocamento à esquerda
<code>*</code>	Multiplicação	<code>>></code>	Deslocamento à direita
<code>/</code>	Divisão	<code>>>></code>	Desloc. a direita com zeros
<code>%</code>	Resto	<code>=</code>	Atribuição
<code>++</code>	Incremento	<code>+=</code>	Atribuição com adição
<code>--</code>	Decremento	<code>-=</code>	Atribuição com subtração
<code>></code>	Maior que	<code>*=</code>	Atribuição com multiplicação
<code>>=</code>	Maior ou igual	<code>/=</code>	Atribuição com divisão
<code><</code>	Menor que	<code>%=</code>	Atribuição com resto
<code><=</code>	Menor ou igual	<code>&=</code>	Atribuição com AND
<code>==</code>	Igual	<code> =</code>	Atribuição com OR
<code>!=</code>	Não igual	<code>^=</code>	Atribuição com XOR
<code>!</code>	NÃO lógico	<code><<=</code>	Atribuição com desl. esquerdo
<code>&&</code>	E lógico	<code>>>=</code>	Atribuição com desloc. direito
<code> </code>	OU lógico	<code>>>>=</code>	Atrib. C/ desloc. a dir. c/ zeros
<code>&</code>	AND	<code>? :</code>	Operador ternário
<code>^</code>	XOR	<code>(tipo)</code>	Conversão de tipos (cast)
<code> </code>	OR	<code>instanceof</code>	Comparação de tipos

Comandos de Entrada e Saída

COMANDOS DE *ENTRADA E SAÍDA*

114



COMANDO DE **ENTRADA**

115

- Os algoritmos precisam ser “alimentados” com dados para efetuarem as operações e cálculos que são necessários a fim de alcançar o resultado desejado;
- Para que o algoritmo possa receber os dados necessários a serem informados pelas pessoas, utiliza-se o comando **LEIA (VARIÁVEL)**.



COMANDO DE ENTRADA

- EXEMPLO:



```
Declaração de Variáveis  
Inteiro: TOTAL;  
...  
LEIA(TOTAL);
```

COMANDO DE SAÍDA

- Para que o algoritmo possa mostrar os dados que calculou, como resposta ao problema que resolveu, adota-se o comando **ESCREVA (VARIÁVEL)**.



COMANDO DE SAÍDA

- EXEMPLO:



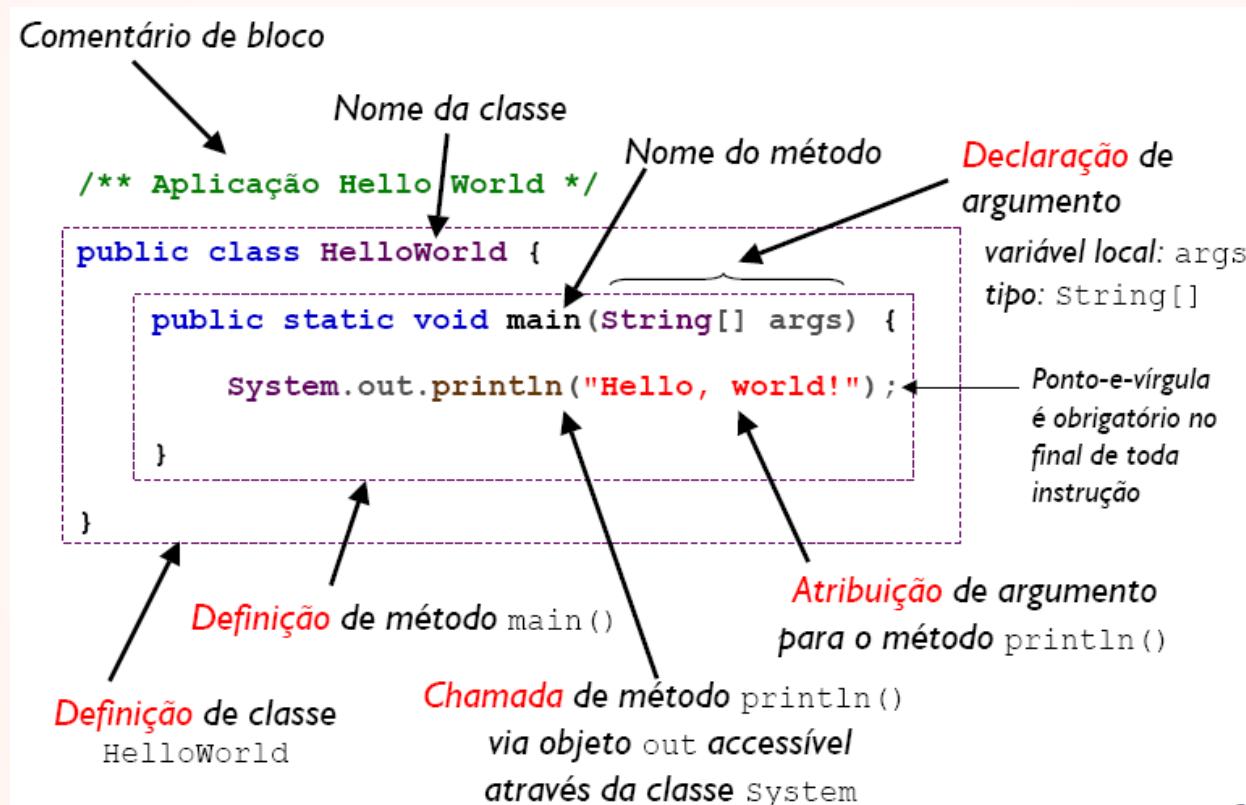
```
Declaração de Variáveis  
Inteiro: TOTAL;  
...  
ESCREVA(TOTAL) ;
```

BLOCOS

- Um bloco pode ser definido como um conjunto de ações (comandos de atribuição, entrada, processamento e saída) que delimitam as funcionalidades de um algoritmo (programa).
- **SINTAXE BÁSICA**

```
/** Primeiro Algoritmo **/  
Algoritmo <Nome>  
    Declaração de Constantes  
    ...  
    Declaração de Variáveis  
    ...  
    INÍCIO  
        LEIA(...);  
        {OPERAÇÕES ARITMÉTICAS/RELACIONAIS/LÓGICAS}  
        ESCREVA(...);  
    FIM.
```

BLOCOS EM JAVA



COMANDOS DE ENTRADA E SAÍDA EM JAVA

- A classe **Scanner** implementa as operações de entrada de dados pelo teclado.
- A classe **Formatter** implementa as operações de saída dos dados gerados pela aplicação.

```
Formatter fmt = new Formatter(System.out);
Scanner scn = new Scanner(System.in);
int n1 = scn.nextInt();
fmt.format("n1: %d", n1);
```

MÉTODO PRINTLN

- Escreve na tela o conteúdo nos parêntesis.

```
System.out.println("Olá!");
```

COMANDOS DE ENTRADA E SAÍDA EM JAVA

- Os principais caracteres de formatação estão apresentados abaixo:

%c	caractere simples
%d	decimal
%e	notação científica
%f	ponto flutuante
%o	octal
%s	cadeia de caracteres
%u	decimal sem sinal
%x	hexadecimal

- Para cada um dos tipos primitivos, há um método **nextXxx()** correspondente.

```
int n = ler.nextInt();
float preco = ler.nextFloat();
double salario = ler.nextDouble();
String palavra = ler.next();
```

ALGORITMO Nº1

- Escrever um Algoritmo para escrever os valores de duas variáveis.

```
/** Primeiro Algoritmo **/  
Algoritmo EscreveVariaveis  
    Declaração de Variáveis  
        caractere: fruta;  
        real: valor;  
    INÍCIO  
        fruta ← "Banana";  
        valor ← 3.50;  
        escreva("O valor da ",fruta, " é de", valor);  
    FIM.
```

PROGRAMA JAVA Nº1

- Escrever um Programa Java para escrever os valores de duas variáveis.

```
String fruta = "Banana";
double valor = 3.50;
System.out.println("O valor da "+fruta+" é de "+valor);
```

```
jshell> String fruta = "Banana";
fruta ==> "Banana"

jshell> double valor = 3.50;
valor ==> 3.5

jshell> System.out.println("O valor da "+fruta+" é de "+valor);
O valor da Banana é de 3.5
```

ALGORITMO Nº2

- Escrever um Algoritmo para ler um nome e escrever uma mensagem com o nome lido.

```
/** Segundo Algoritmo **/  
Algoritmo LeituraEscrita  
Declaração de Variáveis  
    character: nome;  
INÍCIO  
    escreva("Digite o seu Nome");  
    leia(nome);  
    escreva("Seja bem-vindo ", nome);  
FIM.
```

PROGRAMA JAVA Nº2

- Escrever um Programa para ler um nome e escrever uma mensagem com o nome lido.

```
Scanner teclado = new Scanner(System.in);
System.out.println("Digite o seu Nome");
String nome = teclado.next();
System.out.println("Seja bem-vindo "+nome);
```

```
jshell> Scanner teclado = new Scanner(System.in);
teclado ==> java.util.Scanner[delimiters=\p{javaWhitespace}+] ... \E][infinity string=\Q?\E]

jshell> System.out.println("Digite o seu nome");
Digite o seu nome

jshell> String nome = teclado.next();
Antonio Benedito
nome ==> "Antonio"

jshell> System.out.println("Seja bem-vindo "+nome);
Seja bem-vindo Antonio
```

ALGORITMO Nº3

- Escrever um Algoritmo para ler um número, calcular o seu dobro e escrever o seu resultado.

```
/*** Terceiro Algoritmo ***/
Algoritmo CalculaDobro
Declaração de Variáveis
    real: numero,dobro;
INÍCIO
    escreva("Digite o Número");
    leia(numero);
    dobro ← numero * 2;
    escreva("O dobro do número é ",dobro);
FIM.
```

PROGRAMA JAVA Nº3

- Escrever um Programa para ler um número, calcular o seu dobro e escrever o seu resultado.

```
Scanner teclado = new Scanner(System.in);
System.out.println("Digite o seu Número");
double numero = teclado.nextDouble();
double dobro = numero * 2;
System.out.println("O dobro do número é "+dobra);
```

PROGRAMA JAVA Nº3

- Escrever um Programa para ler um número, calcular o seu dobro e escrever o seu resultado.

```
jshell> Scanner teclado = new Scanner(System.in);
teclado ==> java.util.Scanner[delimiters=\p{javaWhitespace}+] ... \E][infinity string=\Q?\E]

jshell> System.out.println("Digite o seu numero");
Digite o seu numero

jshell> double numero = teclado.nextDouble();
10,5
numero ==> 10.5

jshell> double dobro = numero * 2;
dobra ==> 21.0

jshell> System.out.println("O dobro do numero e "+dobra);
O dobro do numero e 21.0
```

Exercícios

- 1) Elaborar um programa que leia o valor de um jantar. Calcule e informe o valor da taxa do garçom (10%) e o valor total a ser pago.
- 2) Escrever um programa que leia a duração de uma viagem em dias e horas. Calcule e informe a duração total da viagem em número de horas.
- 3) Elaborar um programa que leia um número. Informar os números anterior e posterior.
- 4) Escrever um programa para uma pizzaria que leia o valor total de uma conta e quantos clientes vão pagá-la. Calcule e informe o valor a ser pago por cada cliente.
- 5) Elaborar um programa para uma loja que leia o preço de um produto e informe as duas opções de pagamento: à vista com 10% de desconto ou o mesmo valor em 3x.
- 6) Escrever um programa que leia as 04 notas de um aluno e calcule a sua média.

Exercícios

- 7) Elaborar um programa para uma videolocadora que leia o título e a duração de um filme em horas e minutos. Exibir o título do filme com a sua duração apenas em minutos.
- 8) Escrever um programa para uma revenda de veículos. O programa deve ler o modelo e o preço do veículo. Apresentar como resposta o valor de entrada (50%) e o saldo restante em 12x sem juros.
- 9) Elaborar um programa para uma lanhouse, cujo custo por 15 minutos de uso seja uma constante de R\$2,00, que leia o tempo que o cliente utilizou o computador e quanto ele deve pagar, sabendo que as frações extras de 15 minutos devem ser cobradas de forma integral.