



# Introdução aos princípios fundamentais da computação

Introdução aos conceitos primordiais acerca do funcionamento de computadores e como estes são ferramentas desenvolvidas pelos humanos para criar soluções úteis a partir da compreensão de códigos de computador.

Prof. Guilherme Dutra Gonzaga Jaime

## Objetivos

- Descrever a equação que representa a essência dos computadores.
- Reconhecer o processo de exploração do potencial dos computadores por meio de códigos e algoritmos.
- Empregar códigos simples de instruções de computadores.

## Introdução

Neste vídeo, você vai aprender conceitos básicos acerca do funcionamento de computadores e como estes são ferramentas desenvolvidas pelos humanos para criar soluções úteis a partir da compreensão de códigos de computador.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

## Equação fundamental

Enquanto os sistemas modernos processam vastas quantidades de informações rapidamente, a habilidade de reflexão e análise crítica, muitas vezes, fica em segundo plano. A tecnologia oferece acesso ilimitado ao conhecimento, mas a verdadeira compreensão requer contemplação e raciocínio profundo. É essencial equilibrar a eficiência do processamento com a qualidade do pensamento para alcançar uma sociedade verdadeiramente informada e consciente.

Vídeo sobre o acesso ilimitado ao conhecimento que a tecnologia nos proporciona.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Anotem a equação para não se esquecerem dela. O computador é igual a:

Imensa capacidade de processar



Terrível deficiência do pensar

Vamos entender o que é isso. Este conteúdo trata dos recursos básicos do funcionamento dos computadores. Isso é muito importante, pois, em pleno século XXI, não é desejável que um profissional pense no computador como algum tipo de caixa mágica com a qual todos interagem diariamente. De fato, no mercado de trabalho atual, é cada vez mais importante que os profissionais saibam como o computador faz o que faz.

É essencial perceber que o mercado de trabalho não é um ente determinador, mas, sem dúvida, a relação com a empregabilidade é fundamental. Você consegue perceber alguma função em que o uso tecnológico não é necessário? Pense bem: as salas de aula, os treinamentos... Enfim, tudo, de alguma forma, vivencia o uso da tecnologia.

Com isso, nós nos deparamos com um dilema: a adoção da tecnologia não é linear, nem todos têm acesso a ela nem possuem a mesma experiência como usuários.

Durante muito tempo, pensou-se que lidar com os computadores consistia em dominar a tecnologia vigente e aprender suas técnicas e seus fundamentos, pois, assim, você estaria seguro. Mas como isso é possível tendo em vista que parte das pessoas já estão no topo do uso da tecnologia, enquanto outros ainda estão dando os primeiros passos?





### Atenção

O desespero de diversas instituições gerou o processo mais absurdo: já que todos não podem dispor de tecnologia, então vamos abrir mão dela. Isso só aumentou a segregação, a fragilidade de grupos sociais que não têm acesso à tecnologia. Quando fazemos isso, negamos a vários profissionais a possibilidade de diminuir essas diferenças.

Porém, o que pode ser feito? Se não aprendermos a usar a tecnologia do momento, se renegarmos a tecnologia para garantir igualdade, qual a solução proposta? Uma ideia é entender o funcionamento básico dos computadores, perceber como funcionam essas máquinas e, independentemente da tecnologia atual, compreender qual o sentido de sua atualização — os computadores estão sempre sendo atualizados, uma vez que o homem precisa constantemente de novas respostas e possibilidades.

Felizmente, qualquer pessoa pode compreender, basicamente, como os computadores funcionam. Nosso objetivo é explorar as qualidades essenciais dos computadores, como eles funcionam, o que eles podem e o que não podem fazer. Este conteúdo não requer nenhum conhecimento prévio em informática.

A essência fundamental dos computadores é composta pelas seguintes características:

#### Extremamente poderosos

São poderosos, pois um computador pode fazer bilhões de operações por segundo e consegue gerar dados a uma velocidade inconcebível para nós, humanos.



#### Profundamente tolos

São tolos, pois as operações que ele pode realizar são extremamente simples. Por exemplo, adicionar dois números para obter um terceiro número ou verificar se um número é zero.

Você provavelmente deve pensar “portanto, operações tão simples como essas não são capazes de gerar discernimentos ou entendimentos mais elaborados, como, por exemplo, identificar a impressão digital de uma pessoa, certo?”. Embora você pense dessa forma, computadores fazem isso: identificam digitalmente um indivíduo.



### Saiba mais

Estudos revelaram que os humanos possuem impressões digitais singulares. Com base nisso, foi elaborada uma forma de retirar esses dados — primeiro, foi utilizada graxa para colher digitais; atualmente, usam-se luz e calor —, criando um banco de dados com essas informações. Portanto, o trabalho de discernimento é humano; o computador apenas cruza, de forma mais rápida e efetiva, os dados armazenados pelas pessoas.

Isso ocorre porque discernimento e compreensão são qualidades humanas! É importante compreender o seguinte: o computador é realmente um mundo mecânico, em que é possível executar operações muito simples a uma velocidade incrivelmente alta. Por exemplo, desde 2005, processadores são capazes de

executar mais de 1 bilhão de instruções por segundo. Já os processadores lançados em meados de 2019 são capazes de executar mais de 48 bilhões de instruções por segundo.

Ao longo do tempo, Hollywood tem retratado em seus filmes computadores que nunca são mostrados como mecânicos, mas sim como entidades com características essencialmente humanas, como discernimento, compreensão, criatividade e capacidade de se emocionar.

A realidade não poderia estar mais distante daquilo que é mostrado nos filmes. O que temos é uma combinação engraçada, que une a poderosa capacidade de executar instruções a um conjunto de instruções que, inicialmente, parecem tolas.



### Saiba mais

Filmes Na seção Explore+, indicaremos um vídeo da lendária cena do filme 2001: uma odisseia no espaço, e esperamos que, ao ver esse trecho, você entenda como computadores são bem diferentes do que é mostrado em filmes como esse. A descrição de filmes que mostram isso é interessante, como Matrix, Exterminador do futuro e Eu, robô. Basta ler uma sinopse para perceber como o senso comum olha para o desenvolvimento das máquinas.

Então, é necessário compreender como é possível, mesmo assim, criar recursos/soluções interessantes e poderosos. Qualquer pessoa que interage com computadores, provendo instruções em alguma linguagem de programação, experimenta uma sensação vívida dessa combinação.

## Atividade 1

Os processadores são capazes de executar muitas instruções (tarefas) e, para surpresa de muitos, não consegue fazer sozinho alguns tipos. Qual operação os processadores executam?

A Multiplicação

B Radiciação

C Interpretação de dados

D Decomposição

E Manipulação de dados



A alternativa E está correta.

Os processadores executam operações matemáticas básicas, como multiplicação (a), adição (b), subtração (c) e divisão (d), por meio de instruções de hardware específicas.

## Humano versus máquina

A relação entre homem e máquina é complexa e está em constante evolução, influenciada por avanços tecnológicos, questões sociais e éticas, e a maneira como projetamos, desenvolvemos e usamos tecnologias digitais em nossas vidas. Essa área estuda como as pessoas interagem com os sistemas computacionais e como projetar esses sistemas de forma que sejam eficientes e satisfatórios para os usuários. A relação entre homem e máquina é um tema amplo e multifacetado, abrangendo diversas áreas, desde a interação humano-computador até a ética e o impacto social das tecnologias digitais.

Vídeo sobre a relação complexa do homem com a máquina, e a sua evolução com os avanços tecnológicos.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

E a inteligência artificial? E os robôs que podem ter sentimentos? Em todos esses casos, a equação fundamental se mantém.

O computador possui a capacidade de fazer as ações sobre sequenciamento lógico definido, com um conjunto de variações e velocidades elevadas, conforme são ampliadas as capacidades de armazenamento e processamento. Por outro lado, se os comandos não forem estabelecidos, se os parâmetros não estiverem dispostos e estruturados, a máquina chega a um limite.

Um dos principais movimentos intelectuais trabalhados pelos gregos foi perceber que, no mundo, existe uma parte física, calculável, com possibilidades amplas, e que, sem essa percepção, não é possível entendê-la — trata-se do mundo físico. Também temos outra parte, que, embora seja proveniente das representações físicas do mundo, não depende delas para a construção dos sujeitos.

Estamos falando especificamente da metafísica, a construção de percepções mentais que olham pela lógica, pela argumentação, pelo estudo, pela dialogia, pela capacidade mental de o sujeito alcançar o entendimento universal. Essa parte não tem limites; suas representações são infinitas.

Uma velha alegoria de Platão ajuda a explicar isso. Trata-se da alegoria da caverna, em que é mostrado que todos nós vivemos em um mundo limitado, físico, de pedra, com as representações nas paredes iluminadas por um fogo contínuo atrás de nós. Quando libertos, o que encontramos é um mundo infinito, perfeito, é o mundo do pensamento, da abstração. Nossa capacidade de abstração e de pensar é infinita, mas nossa capacidade de materializar o que foi pensado é limitada. Esse é um dos sentidos da alegoria. Buscamos ser mais eficientes, mais próximos das múltiplas possibilidades de nossa mente.

O mundo perfeito, infinito, com possibilidades de construir e destruir, infelizmente, não mora em uma máquina, mas na capacidade humana. A máquina, o computador, é só mais um aprimoramento de nossos usos e nossas ferramentas, ainda que aprisionados, limitados, mas tentando e buscando ir mais longe.



## A alegoria da caverna

Veja uma explicação sobre a alegoria da caverna e entenda como o filme *Eu, robô* muda a nossa percepção de entendimento de computador x humano.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Agora ficou claro? Vejamos mais um caso. Enfrentar uma máquina em um jogo de xadrez era uma das maiores diversões do século XX. Durante muito tempo, os grandes nomes do xadrez foram os vencedores, mas, lenta e continuamente, o desempenho das máquinas foi melhorando, e os jornais proclamavam: "Máquina vence o homem!".

Será mesmo? Claro que não! Essa disputa nunca foi entre a máquina e o homem, mas sim entre o programador e o xadrezista.



Enquanto as máquinas não receberam todas as informações para neutralizar, calcular as probabilidades e dar o melhor resultado, o programador perdeu. Porém, quando a capacidade de processar, arquivar e perceber os melhores algoritmos foi atingida, quando a máquina caminhou o bastante para dar respostas no tempo esperado, tivemos a impressão de que a máquina venceu o homem. Entretanto, tudo o que ela fez foi executar aquilo que o programador pediu, só que com uma capacidade que o homem, de forma direta, jamais conseguiria.

## Atividade 2

A alegoria da caverna é uma metáfora utilizada por Platão, filósofo grego do século IV AEC, em seu diálogo *A República*. Nesta alegoria, Platão descreve um grupo de pessoas acorrentadas em uma caverna desde o nascimento, com as cabeças presas de modo que só podem olhar para frente. Atrás delas, há uma fogueira e, entre a fogueira e as pessoas, passa uma parede. Entre a parede e o fogo, objetos e pessoas são projetados, criando sombras na parede em frente aos prisioneiros.

Essas sombras são a única percepção da realidade desse grupo. Platão usa essa metáfora para ilustrar a jornada da alma em direção ao conhecimento verdadeiro e à compreensão da realidade, sugerindo que a maioria das pessoas está presa à ignorância e à ilusão, como os prisioneiros na caverna, até que se libertem e alcancem a luz do verdadeiro conhecimento. Como podemos relacionar a alegoria da caverna aos computadores?

A Representa uma metáfora para o surgimento da inteligência artificial.

B Ilustra a interconexão global proporcionada pela internet.

C Reflete o processo de aprendizado de máquina por meio da experiência.

D Demonstra a distinção entre a realidade física e a realidade virtual.

E Explora a ética da privacidade de dados e a vigilância digital.



A alternativa D está correta.

A **alegoria da caverna** de Platão pode ser relacionada aos computadores ao explorar a distinção entre a realidade física e a realidade virtual. Assim como os prisioneiros na caverna de Platão enxergavam apenas sombras projetadas na parede, muitas vezes os usuários dos computadores são imersos em mundos digitais, desconectados da realidade física. Essa alegoria oferece uma perspectiva para pensar sobre como a tecnologia pode criar realidades alternativas e nos afastar da verdadeira experiência da vida.



# Entre códigos e algoritmos: aprendendo a lógica do computador

Um computador opera por meio de uma série de instruções codificadas, executadas rapidamente, mas individualmente simples. A utilidade surge da combinação dessas instruções para atender às necessidades humanas, como edição de fotos ou envio de e-mails. Os programadores são essenciais nesse processo, transformando demandas em algoritmos executáveis, possibilitando diversas funcionalidades, desde operações bancárias até filtros de maquiagem. A evolução tecnológica reflete a evolução social, em que as demandas são captadas e traduzidas em código para solucionar problemas específicos, permitindo a criatividade na resolução de desafios reais.

Vídeo sobre os princípios da lógica de programação e como ele atende às necessidades humanas.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Vamos voltar à dinâmica do entendimento do pensamento computacional. Você está fazendo uma prova de matemática, e o conteúdo é análise combinatória; portanto, aprendeu todas as fórmulas e métricas, basta aplicá-las. Agora, é ler o problema e colocar tudo lá.



Se a máquina é extremamente rápida, mas muito tola, como é possível a implementação de tantos recursos úteis nos computadores?

Você repetiu o dia inteiro:  $C_{n,k} = \frac{(n+k-1)!}{k!(n-1)!}$

E na hora surge uma dúvida:

“É uma combinação simples ou com repetição? Não sei, eu só aprendi que essa é a fórmula. Não basta apenas colocar os números, e o resultado será alcançado?”.

Claro que não! Uma fórmula é uma simplificação, e cada um de seus elementos só tem sentido se for observado dentro de um contexto. Demos o exemplo matemático, porém a matemática é apenas uma leitura da vida.

Imagine um motorista que condicionou o seguinte: sinal (farol) amarelo é para parar. Porém, no trânsito, nem todos pensam da mesma forma. Logo, se esse motorista vê um sinal amarelo e para, pode ser que o condutor que vem atrás, por não pensar como ele, não consiga frear. O resultado é a colisão. O ponto é que, para que

um computador execute o que você deseja, ele precisa de informações, padrões e leituras, pois só assim ele será capaz de ampliar suas possibilidades de soluções.



### Comentário

Os computadores, e incluímos aí o seu celular ou o PC que você usa, têm a mesma dinâmica. Todos receberam informações — processos e procedimentos — para que pudessem executar o que se espera deles. É isso que vamos estudar agora.

## Atividade 1

Os ambientes computacionais têm vários elementos importantes para execução das tarefas. O código-fonte e o algoritmo possuem uma relação. Qual é a relação entre eles?

- A O código-fonte é uma linguagem de programação específica, enquanto o algoritmo é uma sequência de passos lógicos.
- B O código-fonte é uma representação visual do algoritmo.
- C O código-fonte é o resultado final do algoritmo.
- D O algoritmo é uma etapa intermediária na criação do código-fonte.
- E O código-fonte e o algoritmo são sinônimos e podem ser usados indistintamente.



A alternativa A está correta.

O código-fonte é uma representação escrita em uma linguagem de programação específica, enquanto o algoritmo é uma sequência de passos lógicos que descreve a solução para um problema. O algoritmo precede a codificação em código-fonte e é uma descrição abstrata do processo a ser seguido. O código-fonte, por sua vez, é a implementação concreta do algoritmo em uma linguagem de programação específica, tornando-o executável pelo computador. As demais opções não refletem adequadamente a relação entre código-fonte e algoritmo.

## O conceito de código

Para compreender o funcionamento do computador, é essencial entender o código, que consiste em instruções sequenciais. O computador executa essas instruções rapidamente, porém de forma simples. Os programadores criam algoritmos que transformam demandas humanas em funcionalidades, como a edição de fotos. A utilidade do código é direcionada pelo usuário, refletindo a evolução social. A conexão entre a demanda humana e a solução é mediada pelos programadores, que desenvolvem algoritmos para atender às necessidades da sociedade.

Vídeo sobre as instruções sequenciais (códigos) necessárias para a execução de tarefas no computador.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Para compreendermos o funcionamento de um computador, precisamos entender o que é um código. Então, dentro do computador, temos um corpo de códigos, e eles são apenas uma série de instruções, por exemplo: adicione dias e medidas e compare-os. Você se lembra da equação fundamental, não é mesmo?

Um computador corresponde à incapacidade de pensar à velocidade X. Então, o trabalho da máquina consiste em apenas uma série de instruções; ela só executa o que é pedido na ordem em que recebeu as orientações de quem escreveu o código. O computador é capaz de passar por uma sequência de instruções de uma forma incrivelmente rápida. Quem pensa e estrutura, podendo ser cada vez mais complexo ou simples, é o sujeito que programa.



Há uma série de instruções, e o computador apenas caminha através delas, executando cada uma exatamente como instruído. São as famosas palavras executar e rodar, como você já deve ter ouvido. Então, sim, um código é executado de forma muitíssimo rápida, mas as instruções individuais são bastante simples, o que leva a uma pergunta natural:

Se essas instruções são tão simples, como podemos ter um recurso útil? Ou seja, como tornar o computador capaz de realizar tarefas que dependam de algum nível de inteligência e discernimento?

Vamos tentar algo prático. Usaremos o recurso de redução de olhos vermelhos de fotografias como um exemplo de utilidade do computador. Existem infindáveis exemplos diferentes, como enviar um e-mail, assistir a um vídeo on-line ou qualquer outra tarefa que executamos em nossos computadores diariamente.

Que tal, agora, pensarmos nos filtros para maquiagem? Mas isso serve para quê? Para atender aos desejos do usuário. Diverte, faz rir, aumenta a autoestima. A programação não julga, não cria a demanda; ela atende à demanda. Embora receba as instruções e realize as tarefas, a utilidade é do usuário que precisava daquilo.



Sempre que você pensar em um código, deve imaginar o seguinte: se ele foi criado e, principalmente, se alcança um grande número potencial de utilização, não foi o computador que evoluiu, mas a sociedade. Alguém captou uma demanda e observou que a máquina, que a criação de um código, poderia suprir essa carência, nem que fosse fazer rir.

O que liga esses dois lados da essência de um computador: a demanda humana e a necessidade de criação de um código para alcançar a solução? A resposta é a seguinte: **programadores.**

Voltemos ao exemplo dos olhos vermelhos. Um programador de computador talvez tenha tido a seguinte ideia: “Seria legal se os computadores fossem

capazes de reduzir os olhos vermelhos de fotografias, pois esse defeito é meio macabro, e eu fico desconfortável ao vê-lo". Essa é uma ótima ideia! Trata-se de um discernimento, uma escolha humana que atende a seus anseios: deveríamos ter redução de olhos vermelhos. Assim, o programador de computador pensa em uma maneira de tornar isso possível. Resumidamente, temos as seguintes etapas:

#### Etapa 1

O programador procura a cor vermelha na foto e, talvez, checaríamos se ela tem uma forma arredondada.

#### Etapa 2

Em seguida, o programador precisa pensar em uma forma de desempenhar cada um desses passos e escrever um algoritmo em instruções tão simples, tornando possível que o computador execute cada uma.

#### Etapa 3

Ao final, o programador tem a função de redução de olhos vermelhos funcionando.

O resultado do sequenciamento de instruções compreensíveis ao computador para realizar uma dada tarefa (ex: redução de olhos vermelhos) é chamado de **algoritmo**.

## Atividade 2

Vários profissionais são importantes para as atividades relacionadas ao funcionamento dos ambientes computacionais e uma das funções mais relevantes é o papel do programador. Qual é o papel dos programadores no contexto do funcionamento do computador?

A Criar demandas para novas funcionalidades.

B Monitorar o desempenho do hardware.

C Desenvolver algoritmos para transformar demandas humanas em funcionalidades.

D Ajustar a velocidade de processamento do computador.

E Atender às demandas de hardware do usuário.



A alternativa C está correta.

Os programadores têm a responsabilidade de desenvolver algoritmos que traduzam as demandas humanas em funcionalidades executáveis pelo computador, permitindo que ele atenda às necessidades da sociedade. Esses algoritmos são a base para uma variedade de funcionalidades, desde a edição de fotos até o envio de e-mails. As demais opções não refletem adequadamente o papel específico dos programadores no contexto do funcionamento do computador.

## Algoritmo

Um algoritmo é um conjunto de instruções com uma finalidade útil, podendo ser expresso em linguagem comum. Por exemplo, calcular a idade com base na data de nascimento. Para implementá-lo, escreve-se um código em linguagem de programação. O algoritmo só é válido quando alcança seu propósito. Programadores implementam algoritmos que, por sua vez, serão executados em computadores. Já o computador se destaca por sua velocidade e economia.

Vídeo sobre o conjunto de instruções com uma finalidade específica, chamada algoritmo.



### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

## Definição

Para usar o e-mail, assistir a um vídeo, conversar pelo WhatsApp, realizar operações bancárias, ou qualquer outro recurso útil via computador, smartphones etc., foi preciso que, em algum momento, uma pessoa ou uma equipe pensasse: “Bem, deveríamos escrever um algoritmo para isso”.

Alguém estruturou os procedimentos necessários e, depois, transformou os passos de alto nível em instruções suficientemente simples, para que o computador ou o smartphone pudesse executá-las. Essa pessoa ou a equipe trabalhou, então, escrevendo o código para a solução. Em seguida, um bom tempo foi gasto para testar e melhorar o código, até que ele, de fato, representasse com fidelidade o algoritmo projetado originalmente.

O programador pode se concentrar em ser criativo, para, então, criar um algoritmo capaz de resolver um problema real.



## Programadores versus computadores

Programadores de computadores são os responsáveis por discernir um aproveitamento útil, transformar a ideia de aplicação (ex.: redução de olhos vermelhos em fotos) em um algoritmo e traduzir esse discernimento em uma sequência de instruções simples, compreensíveis para o computador. As máquinas, por outro lado, são incredivelmente rápidas. Assim, programadores e computadores formam uma combinação extremamente poderosa. Vamos entender agora a diferença entre código e algoritmo:

#### Código (definição simplificada)

É um conjunto qualquer de instruções simples escrito em alguma linguagem padrão compreensível para o computador. Deve ser escrito em uma linguagem de programação padrão, senão os computadores não conseguirão executá-lo.



#### Algoritmo (definição simplificada)

É um termo que reflete uma ideia mais completa, em que o conjunto de instruções possuem uma finalidade útil. Não precisa ser escrito em linguagem de computador, podendo ser escrito, por exemplo, em português.

Suponha que você tenha pensado em um algoritmo com a finalidade de calcular a idade de pessoas com base em sua data de nascimento. Então, você começa a escrever um código que deve seguir uma das linguagens de programação padrão. Enquanto não estiver funcionando, não poderá ser classificado como um algoritmo, pois ainda não alcança sua finalidade.

Então, no que o computador é bom? Bem, o computador é bom em ser **rápido e barato**.

Com a lei de Moore, os computadores têm ficado mais baratos, e isso ocorre há várias décadas. Como resultado, eles podem ser mais difundidos e é possível embutir mais pesquisa científica para desenvolvê-los, tornando-os ainda mais rápidos. Conheça mais a seguir.

## Atividade 3

O algoritmo tem a função de atender às necessidades do usuário e é importante para a escrita do código-fonte. Qual é a definição de um algoritmo?

A

Um conjunto de instruções expresso apenas em linguagem de programação.

B

Uma sequência de passos complexos sem propósito definido.

C

Um conjunto de passos, escrito em linguagem normal.

D

Uma série de comandos sem relação entre si.

E

Um processo que não requer implementação em linguagem de programação.



A alternativa C está correta.

Um algoritmo é definido como um conjunto de instruções com uma finalidade útil, que pode ser expresso em linguagem comum. Ele é projetado para resolver um problema específico e pode ser implementado em linguagem de programação para ser executado por um computador. As demais opções não correspondem adequadamente à definição de algoritmo apresentada no texto.

# Como os computadores evoluem

A previsão de Gordon Moore, feita há mais de 50 anos, destaca a contínua evolução tecnológica, impulsionando a interconexão e a substituição rápida de sistemas. Cada nova geração amplia o acesso a tecnologias mais antigas, enquanto a capacidade de troca de dados aumenta e os custos diminuem. A popularização dos computadores exemplifica esse processo. Apesar das mudanças constantes, a dinâmica subjacente permanece. É importante destacar a importância de entender a lógica do funcionamento dos computadores.

Vídeo que mostra a evolução tecnológica contínua, na qual cada nova geração amplia o acesso tecnológico.



## Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

## Lei de Moore

A previsão feita há mais de 50 anos por Gordon Moore é fantástica. A tecnologia está em evolução e desenvolvimento contínuos. Afinal, cada vez mais, sistemas dependem da tecnologia, em virtude de sua interconexão, permitindo a ampliação da tecnologia e dos produtos e aumentando a velocidade de sua substituição.

A cada geração, um conjunto geracional antigo de tecnologia está presente no mercado, sendo consumido por públicos que não teriam acesso antes. A cada evolução, porém, a capacidade de troca de dados e informações seriam maiores, e a necessidade de uso de material, menor, permitindo um barateamento relativo. No limite, a evolução tecnológica acaba barateando os computadores.

A popularização dos computadores é uma das maiores provas daquilo que foi apontado por Moore.



Gordon Moore.

Nos anos 1980, nos Estados Unidos (o que só ocorreu no Brasil a partir dos 1990), a primeira onda de computadores pessoais, os PC, ocupou os espaços nas lojas lentamente. Essas máquinas deixaram de ser enormes e pesadas.



Nos anos 2000, as telas e as conexões eram a novidade.



Nos anos 2010, as bandas largas se multiplicaram.



Com o passar do tempo, o computador foi diminuindo, podendo ser encontrado em forma de tablets, laptops, smartphones.



O acesso à tecnologia se deu em um tempo muito curto, o que é impressionante. Quando achamos que dominamos determinada tecnologia, tudo parece mudar de repente. É assustador! Mas sabe o que de alguma forma não mudou? A **dinâmica**.

## Reflexão

Você aprendeu a lógica do funcionamento do computador. Vamos revisar.



### Relembrando

Um computador é apenas mais uma das históricas tentativas humanas de facilitar e resolver demandas que surgem de forma recorrente. Na busca dessas demandas, cada inovação que aparece torna-se uma estrutura — quer dizer, parte da sociedade, ao mesmo tempo, é estruturante e força novas buscas, transformando toda a sociedade. Nós, humanos, temos essa característica, e as máquinas vivem para atender a demandas que são pensadas e estruturadas por nós. Depois que uma demanda é pensada, precisamos focar o processamento das informações e a capacidade de armazená-las. Isso, aliás, é a origem de todo o processo da computação. Homens precisavam ampliar sua capacidade de armazenamento, de reprodução, de execução.

Agora, pense sobre a questão. Em seguida, construa um texto que conte um pouco da história de como a tecnologia (em suas fases de desenvolvimento) impactou sua vida escolar, sua casa etc.

Esse exercício é para você pensar em velocidade, mas é importante que também perceba: continuamos desenhando códigos e executando algoritmos, permanecemos como um operador de máquina da Revolução Industrial, ou como uma criança aprendendo a ler e a escrever, que necessita entender a mecânica de funcionamento e a busca de melhoria da execução.

Pense nisso!

## Atividade 4

A todo momento, a tecnologia vem avançando e, conseqüentemente, assumindo novas atividades e desafios, os quais necessitam de mais recursos. Qual é o principal impacto da evolução tecnológica?



A A redução da interconexão entre sistemas.

B A diminuição da dependência da tecnologia.

C O aumento da velocidade de substituição de produtos.

D A estagnação do desenvolvimento tecnológico.

E A ampliação do acesso à tecnologia.



A alternativa E está correta.

A evolução tecnológica permite a ampliação do acesso à tecnologia, possibilitando que públicos que antes não tinham acesso a determinadas tecnologias agora possam consumi-las. Isso é evidenciado pela popularização dos computadores. As demais opções não refletem adequadamente o impacto descrito no texto.

## Contextualização

Para compreender a natureza dos computadores, é importante interagir com códigos em execução. Este tutorial introduz uma abordagem simples para escrever e executar código de computador, oferecendo uma primeira experiência na linguagem Python. Apesar da simplicidade inicial, essa jornada visa familiarizar você com os fundamentos da programação, proporcionando uma compreensão básica de como os computadores operam. Assim como usamos linguagens humanas para comunicação, as linguagens de programação são usadas para instruir os computadores. Esse processo, gradualmente, expande as possibilidades de manipulação de códigos, fornecendo uma base para explorar conceitos mais complexos no futuro. Assista agora ao vídeo sobre as instruções de computadores e seus conceitos básicos.



#### Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Agora queremos chegar ao momento em que você possa escrever algum código de computador, executá-lo e ver o que ele faz. No computador, tudo se resume realmente a códigos em execução (rodando). É assim que as coisas acontecem. Então, para que a natureza dos computadores seja entendida, nada melhor do que rodar um pouco de código e ver como ele funciona.

## Atividade 1

**De acordo com o texto, qual é a analogia empregada para descrever os elementos de código apresentados?**

A

Blocos de construção

B

Quebra-cabeças

C

Objetos de arte

D

Ferramentas complexas

E

Pacotes de software



A alternativa B está correta.

No texto, é apresentada uma analogia que correlaciona os elementos de código a **um brinquedo de peças de encaixe**, sugerindo que, embora individualmente eles possam parecer simples, dispersos e complexos, com o tempo e mais conhecimento, as possibilidades e os usos desses elementos se modificam. A analogia de **blocos de construção** é a mais precisa, pois cada peça isolada pode parecer simples, mas, quando combinada de diversas maneiras, pode criar estruturas maiores e mais complexas, refletindo a progressão gradual na compreensão e na manipulação dos códigos de computador.

## Entendendo o código na prática

Sintaxe refere-se às regras e à estrutura gramatical específicas que devem ser seguidas ao escrever códigos em uma linguagem de programação. Cada linguagem de programação tem sintaxe única, que define como o código deve ser escrito para que seja considerado válido e compreensível pelo compilador ou interpretador. Lidar com erros de sintaxe é uma parte fundamental do processo de desenvolvimento de software e a resolução desses erros é uma habilidade essencial para o dia a dia do desenvolvedor. Com a grande quantidade de linguagens e ferramentas de desenvolvimento, pode acontecer a utilização de sintaxe de uma linguagem, enquanto desenvolve, com outra diferente. Outro caso comum é a ausência de caracteres necessários para o entendimento das instruções pelo computador. Hoje em dia, as interfaces de desenvolvimento fazem sugestões de sintaxe e uma verificação prévia, a qual já aponta os erros e possíveis soluções, minimizando, assim, os erros de sintaxe.

Utilizaremos, nas práticas a seguir, códigos-fonte simples. Os elementos de código que serão mostrados aqui não são chiques, chamativos ou complexos. Eles são muito simples. Talvez seja um pouco como um brinquedo de peças de encaixe, isoladas, dispersas, complexas; porém, com o tempo e a familiaridade, as possibilidades e o uso vão se modificando.

Aqui, utilizaremos uma das linguagens de programação mais usadas na atualidade, a linguagem Python. Usaremos somente os principais recursos, para que você possa escrever e manipular códigos bem simples e brincar com a ideia-chave: compreender de fato como funcionam os computadores.

Para compreender como imprimir **strings**, veja a Prática 1 a seguir.

### Strings

Strings e números são os dois tipos de dados muito comuns no código de computador. Por exemplo, uma string é usada para lidar com nomes de pessoas, endereços, endereços de páginas web, endereços de e-mail, ou até mesmo um parágrafo, quando alguém está redigindo uma redação. Todos esses exemplos de tipos de textos são armazenados na memória do computador como strings. Esse é apenas um segundo tipo de dados que seremos capazes de usar para informar ao computador como fazer as coisas. Existem muitos outros tipos de dados, mas, por enquanto, nós nos limitaremos aos vistos até agora. Outra novidade presente no código-fonte da Prática 1 é o uso de comentários na segunda linha.

### Prática 1

Clique em Executar e observe o resultado apresentado em Saída.



#### Conteúdo interativo

esse a versão digital para executar o código.

## O que temos de novidade aqui?

O que fizemos foi usar a sintaxe padrão para informar ao computador que o que está sendo impresso é um texto (string). Para isso, cercamos as strings **Teste**, **Bom dia** e **Tchau** com aspas duplas.

Ao clicar em **Executar**, você pode ver os textos corretamente impressos no **output do code compiler**. A string `"\n"` pula uma linha, ou seja, faz com que a próxima impressão seja feita na linha seguinte.



### Atenção

Um comentário em Python começa com o sinal de jogo da velha (`#`). Outra possibilidade é colocar todo o comentário entre 3 aspas duplas (`""" ... """`) ou 3 aspas simples (`'''...'''`). Em seguida, você pode escrever notas para você mesmo, para lembrá-lo, no futuro, do que você pretendia ou do que está tentando fazer neste ponto do seu código. O computador sabe ignorar comentários, então é apenas uma maneira de adicionar decorações/lembretes ao código com pequenas observações. O uso de comentários é considerado bastante importante, pois ajuda o programador a compreender um código que não foi escrito por ele, ou até mesmo um código que foi escrito por ele há tanto tempo, que ele não lembra mais como raciocinou para escrever aquele trecho de código.

Um exemplo intrigante que podemos citar aqui é colocar a palavra **print** no lugar de **Tchau**. Isso é intrigante, pois **print** também é o nome da instrução de impressão reconhecida pela linguagem de programação que estamos usando.

O que ocorrerá ao substituírmos **Tchau** por **print** e clicarmos e Rodar?

Se você mesmo fizer este teste na Prática 1, obterá o seguinte resultado:

```
Teste
Bom dia
print
```

Portanto, quando sequências de caracteres são colocadas entre aspas duplas, como neste exemplo, o computador entende que se trata apenas de dados passivos, e lida com eles conforme esperado.

Note que temos uma lição importante aqui sobre sintaxe. Nós já sabemos que ela é restrita e mecânica. No entanto, é importante entender que você está seguindo um padrão/convenção ao escrever um código de computador que envolve a sintaxe, o que é característico da linguagem de programação. Então, é muito comum, mesmo para programadores muito profissionais, a ocorrência de pequenos erros de sintaxe.

Quando há um erro de sintaxe, ao clicarmos no botão de execução, haverá alguns erros de sintaxe listados em Saída. Então, é parte da tarefa de quem escreveu o código corrigir esses erros de sintaxe até que o computador aceite que as instruções estão de acordo com o padrão de sintaxe da linguagem de programação em uso. Esse é apenas um processo rápido, superficial e comum.

A razão pela qual estamos discutindo isso neste módulo é que, quando alguém está apenas começando a aprender a escrever códigos (programar), será bastante comum se deparar com erros de sintaxe, pois isso é realmente comum. Então, jamais se permita ter a impressão de que você não está entendendo como escrever

código. Apenas entenda que todo mundo passa por erros de sintaxe, inclusive programadores com décadas de experiência. Então, quando se deparar com erros de sintaxe, apenas faça uma checagem rápida e cuidadosa para corrigir os erros e seguir adiante.

Para promover a ideia de que o erro de sintaxe não é grande coisa, queremos mostrar como consertar isso. Veremos alguns exemplos práticos a seguir. São apenas alguns exemplos de código, todos com erros de sintaxe. Queremos que você treine o processo de identificá-los e corrigi-los.

## Prática 2

Todo mundo passa por erros de sintaxe.

Vamos aprender a identificá-los e corrigi-los?

O resultado de cada um dos seis exemplos a seguir deve ser imprimir em Saída as letras destacadas em **verde**. Você precisa corrigir a sintaxe de cada um dos seis exemplos. Depois que você corrigir a sintaxe das instruções, cada um dos 4 exemplos imprimirá exatamente as três linhas indicadas em vermelho mais à frente.

Saída esperada para cada um dos cinco exemplos práticos a seguir:

A

B B

C C C



Conteúdo interativo

esse a versão digital para executar o código.



Conteúdo interativo

esse a versão digital para executar o código.



Conteúdo interativo

esse a versão digital para executar o código.



Conteúdo interativo

esse a versão digital para executar o código.

### Chave de resposta

1. No primeiro exemplo, o segundo print está escrito errado. Há um l no lugar do i.
2. No segundo exemplo, faltam aspas após o segundo B.
3. No terceiro exemplo, está faltando o primeiro B na segunda linha. Além disso, o terceiro print está escrito errado (pront).
4. No quarto exemplo, está faltando o parêntese esquerdo na primeira linha e o parêntese direito na última linha.

Como podemos perceber, devemos tomar cuidado com a sintaxe ao escrever os nossos códigos, porque, por um pequeno detalhe, podemos escrever códigos com erros de sintaxe (SyntaxError).

## Atividade 2

Uma pessoa iniciante em programação escreveu o seguinte código:

Código-Fonte 05:

```
python  
print ("A")  
print ("B", "B")  
print ("C", "C", "C"
```

O que acontecerá quando o código for compilado? Ele vai executar sem erros ou algum erro ocorrerá?

Assinale a alternativa que corresponde ao que acontecerá quando o código for executado e qual consequente ação o aluno precisará fazer caso o código apresente erros:

☐ A Ocorrerá um erro de sintaxe (SyntaxError) por causa da segunda linha do código. Para corrigir esse erro, essa linha deverá ser alterada para: print (B, B);

☐ B Não ocorrerá nenhum erro quando o código for executado. Logo, nenhuma modificação no código precisará ser feita.

☐ C A primeira linha do código possui um erro de sintaxe e fará com que o código não seja executado corretamente. Essa linha deverá ser alterada para: print (A);

☒ D Ocorrerá um erro de sintaxe quando o código for executado. Para que o código seja executado sem erros, a terceira linha deverá ser alterada para: print ("C", "C", "C");

**E** O código possui erros de sintaxe em suas duas primeiras linhas. Elas devem ser alteradas, respectivamente para: `print (A);` e `print (B, B, B);`



A alternativa D está correta.

O código apresenta um erro de sintaxe apenas na última linha, pois, na invocação da função `print`, está faltando um parêntese de fechamento. Portanto, ao tentar executar o código, o interpretador lançará um erro de `SyntaxError`. As outras duas linhas do código estão corretas e imprimirão na tela os caracteres *A* e *B*.

## Demonstrando o uso de variáveis

Variáveis são como caixas de armazenamento no computador, nas quais guardamos valores para uso posterior. Elas ocupam espaços na memória, permitindo a troca fácil de valores no código. Ao simplesmente alterar o valor atribuído à variável, todo o código adotará o novo valor. Isso oferece flexibilidade ao programador para ajustar e adaptar o software conforme necessário, simplificando o processo de desenvolvimento e manutenção do código.

Então, com relação ao conceito de variável, se tivermos uma linha de código como descrito a seguir, no qual atribuímos o valor 7 a uma variável **V**, o que isso significa, metaforicamente falando, que há uma 'caixa' no computador chamada *V*. Nessa caixa, nós simplesmente podemos armazenar um valor, como um número 7. Também poderíamos armazenar qualquer outro valor que quiséssemos.

Veja o código para atribuir valor a uma variável *V*.

```
python
V=7
```

O que significa na prática?

“Caixa” de nome *V*, onde atribuímos o valor numérico 7

**"Caixa" *V* ← 7**

Sendo um pouco mais específico, essa “caixa” é, na verdade, um espaço na memória do computador, em que podemos armazenar valores, e *V* é o nome que damos a esta área reservada para armazenar o valor que desejarmos.

Em linhas posteriores, no código, se um *V* aparecer, o computador sabe o que colocamos dentro da caixa (em nosso exemplo, o 7). A variável se torna uma espécie de abreviação conveniente para qualquer valor que eu queira usar.

Veja o exemplo a seguir.

1. Na primeira linha, atribuímos o valor 7 à variável que optamos por chamar de V.
2. Na segunda linha, damos um comando para imprimir o valor da variável, que será 7 neste exemplo.
3. Na terceira linha, imprimimos uma string informando "O valor da variável é:" e, depois, indicamos "V" para que o computador imprima de fato o valor recuperado da variável.

Clique em **Executar** e observe os resultados. Notou que, sempre que **V** é referenciado no código, o computador o substitui pelo valor **7**? Este é o equivalente a ler o **valor armazenado na caixa**.



#### Conteúdo interativo

esse a versão digital para executar o código.

Uma grande vantagem do uso de variáveis é que, se o programador quiser usar um valor diferente em seu código, basta trocar o valor na linha de atribuição de valor à variável, e todo o restante do código passará a usar o novo valor sempre que houver referência à variável, que, em nosso exemplo, é V.

Vamos ver na prática o uso de variáveis nas duas práticas a seguir.

### Prática 1



#### Conteúdo interativo

esse a versão digital para executar o código.

Observou que aconteceu quando substituímos o 7 por 1980 e clicamos em Executar?

Todos os locais em que a variável V foi referenciada resultaram em 1980 na saída!

### Prática 2

Escreva, no quadro a seguir, um código que atribua a uma variável chamada nome1 a string "Mariah" e a uma variável chamada nome2 a string "Luiza". Depois, clique em Executar para obter a Saída a seguir destacada em vermelho:

***Mariah Luiza***

***Mariah e Luiza são irmãs***



#### Conteúdo interativo

esse a versão digital para executar o código.

Chave de resposta



É importante destacar que o sinal de igual “=” usado em código de computador significa atribuição de valor.

```
python

nome1 = "Mariah"
nome2 = "Luiza"

print(nome1, nome2)
print(f"{nome1} e {nome2} são irmãs")
```

Brinque à vontade com as práticas anteriores, até que fique à vontade para definir mais do que uma variável e imprimir mais do que um valor, até estar confortável com a ideia de que, para computadores, o uso do sinal de igual (=) significa atribuição e é usado quando se deseja atribuir valor a uma variável. Isso é importante, pois, em Matemática, o sinal de igual possui um significado diferente.

## Atividade prática

Imagine que um desenvolvedor deseja escrever um código que faça com que as seguintes frases sejam impressas no console do computador:

*Guilherme Guilherme Guilherme*

*Eu conheço um amigo chamado Guilherme*

Como seria o código que ele deveria escrever para alcançar o seu objetivo?



### Conteúdo interativo

esse a versão digital para executar o código.

### Chave de resposta

Seria o seguinte:

```
python

N="Guilherme"
print(N,N,N)
print(f"Eu conheço um amigo chamado {N}")
```

## Considerações finais

### O que você aprendeu neste conteúdo?

- Utilização do potencial dos computadores.
- Código e algoritmo de computadores.
- Códigos-fonte de computadores.
- Tratamento de erros de sintaxes.
- Manipulação de variáveis.

### Explore +

A literatura e o cinema de ficção contribuíram muito para discutir a percepção da máquina (e do computador) como superior e possível inimiga da humanidade. Esta é uma das funções da arte: provocar nossa reflexão! Assim, além das obras citadas ao longo de nosso material, sugerimos algumas outras.

Leia os seguintes livros:

- **Admirável mundo novo**, de Aldous Huxley (1932).
- **Eu, robô**, de Isaac Asimov (1950).
- **Guerra dos mundos**, de H.G. Wells (1898).
- **O homem do castelo alto**, de Philip K. Dick (1962).
- **1984**, de George Orwell (1949).

Assista:

- **A.I. – Inteligência artificial**, de Steven Spielberg (2001).
- **Animatrix**, de Shinichiro Watanabe, Takeshi Koike, Yoshiaki Kawajiri, Peter Chung, Koji Morimoto, Mahiro Maeda e Andrew R. Jones (2003).
- **Blade Runner – O caçador de androides**, de Ridley Scott (1982).
- **Ex-Machina: Instinto artificial**, de Alex Garland (2015).
- **Gattaca – A experiência genética**, de Andrew Niccol (1997).

- **Minority report – A nova lei**, de Steven Spielberg (2002).
- **O homem bicentenário**, de Chris Columbus (1999).
- **Wall-E**, de Andrew Stanton (2008).
- **Jornada nas estrelas: Picard**, de Akiva Goldsman, Michael Chabon, Kirsten Beyer e Alex Kurtzman (2020–atualmente)

## Referências

CARVALHO, A.; LORENA, A. **Introdução à computação**: hardware, software e dados. Rio de Janeiro: LTC, 2017.

DALE, N.; LEWIS, J. **Ciência da computação**. 4. ed. Rio de Janeiro: LTC, 2011.

FEDELI, R. D.; POLLONI, E. G. F.; PERES, F. E. **Introdução à ciência da computação**. 2. ed. São Paulo: Cengage, 2010.

FLANAGEN, D. **Javascript**: o guia definitivo. 6. ed. Porto Alegre: Bookman, 2013.

GLENN, J. **Ciência da computação**: uma visão abrangente. 11. ed. Porto Alegre: Bookman, 2013.