



Introdução aos princípios fundamentais da computação

Prof. Guilherme Dutra Gonzaga Jaime

Descrição

Conceitos primordiais sobre o funcionamento de computadores; computadores como ferramentas desenvolvidas pelos humanos para criar soluções úteis; soluções úteis a partir da compreensão de códigos de computador.

Propósito

Iniciar as bases do pensamento computacional demonstrando o seu protagonismo no mercado de trabalho e construir uma compreensão introdutória dos conceitos primordiais que norteiam o mundo dos computadores.

Objetivos

Módulo 1

Equação essencial dos computadores

Descrever a equação que representa a essência dos computadores.

Módulo 2

Processo de exploração do potencial dos computadores

Reconhecer o processo de exploração do potencial dos computadores por meio de códigos e algoritmos.

Módulo 3

Código de instrução de computadores

Empregar códigos simples de instruções de computadores.



Introdução

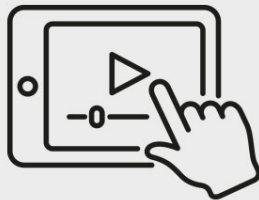
Você sabe como os computadores surgiram? Muitos pensarão em teorias da conspiração, grupos secretos, investimentos maciços de governos em tecnologia. Outros se lembrarão dos mais velhos comentando sobre os cartões perfurados e dos primeiros computadores, que eram do tamanho de uma sala. Um grupo de apaixonados por tecnologia comentará o uso dessa tecnologia durante as guerras do século XX, quando essas máquinas estavam relacionadas a arquivo e trânsito de informações.

No entanto, o processo é, definitivamente, mais longo e mais complexo. Os computadores não são chips, bits e afins. Computadores são o que as máquinas sempre representaram: homens em busca de soluções. Ao longo dos séculos, essa necessidade nos permitiu criar diques para contenção de água, prédios e até estradas para tornar as grandes viagens mais rápidas.

Trata-se de entender que essas máquinas e seus componentes são nossos filhos; elas reproduzem e aceleram o volume de cálculos, armazenamentos e comunicações que nós idealizamos. Dessa forma, cada vez que precisamos de algo a mais, a máquina trabalha para atender nossas demandas. Entender isso é perceber que não estamos em uma maratona contra a tecnologia, com medo de nos tornarmos obsoletos. Novas tecnologias surgirão, e isso é algo que nós mesmos buscamos. Cada máquina, cada software, cada hardware tem o propósito de atender, acelerar e realizar um desejo do coletivo. Algo que não tem essa funcionalidade é abandonado, é esquecido.

Seja bem-vindo ao mundo da tecnologia e lembre-se: você verá aqui o que é necessário para lidar com ela.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



1 - Equação essencial dos computadores

Ao final deste módulo, você será capaz de descrever a equação que representa a essência dos computadores.

Equação fundamental

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

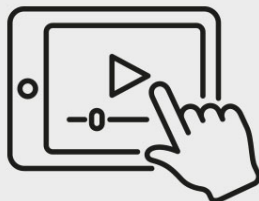


Imensa capacidade de processar



Terrível deficiência do pensar

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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

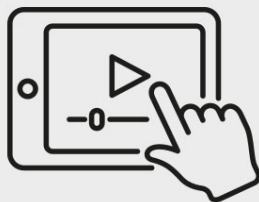
São tolos, pois as operações que ele pode realizar são extremamente simples. Por exemplo, adicionar dois números para obter um terceiro número ou verificar se um número é zero.

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.



Computador: poderosamente rápidos, porém tolos

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.



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.

Humano *versus* máquina

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.

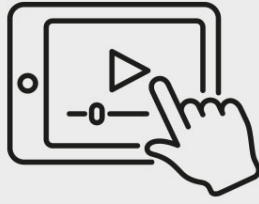
O mundo perfeito, infinito, com possibilidades de construir e destruir, infelizmente, não mora em uma máquina, mas na capacidade humana. A máquina, o computador, é só mais um aprimoramento de nossos usos e nossas ferramentas, ainda que aprisionados, limitados, mas tentando e buscando ir mais longe.



A alegoria da caverna

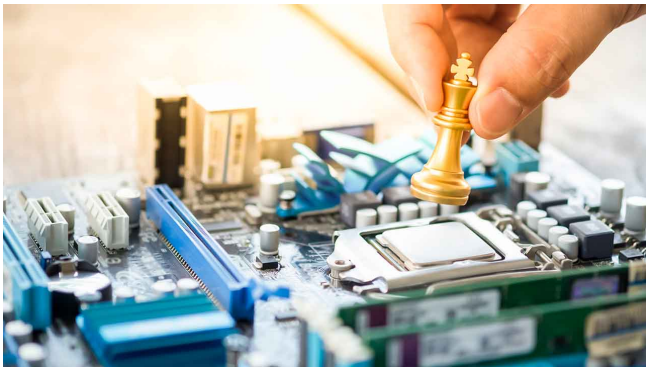
Veja uma explicação sobre a alegoria da caverna e entenda como o filme *Eu, robô* muda a nossa percepção de entendimento de computador x humano.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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 discursiva

Após ter assistido ao vídeo, coloque aqui o seu relato. Agora é o momento de você organizar suas ideias a partir dos seguintes pontos:

1. A equação fundamental dos computadores a partir de suas narrativas.
2. Algo que, de alguma forma, não valida essa equação fundamental.
3. Provoque alguns amigos e familiares sobre a opinião deles. Não precisa sair daqui, pois as redes sociais e os computadores já lhe dão condição de, em minutos, ter suas respostas.

Digite sua resposta aqui...

Chave de resposta ▾

Vamos fazer o relato de uma experiência. No filme *Eu, robô*, ocorre o debate filosófico que estamos propondo. Será que, pela inteligência artificial, o algoritmo pode ser compreendido de forma equivocada? Repare que o filme é perfeito no uso da equação primordial, pois, em momento algum, trata de uma mente coletiva para as máquinas. Nem mesmo a Vick, pois tudo o que ela faz é executar uma ordem humana, que não foi feita corretamente. O robô criado pelo doutor segue regras de programação e acúmulo e não cria nada que vá além de sua programação. Embora o detetive busque encontrar algo que comprove a falha — essa nossa expectativa de que as máquinas têm alma, vida própria e são uma ameaça —, isso não acontece. É utilizada a equação de formulação — máquinas impressionantemente velozes e capazes de processar, mas totalmente burras, incapazes de “pensar” ou agir fora de sua programação.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Estudamos sobre a equação essencial dos computadores. De acordo com o que foi visto, os computadores

- A possuem capacidade de discernimento.
- B possuem capacidade de compreensão.
- C são incrivelmente rápidos e tolos.
- D admitem instruções escritas segundo determinadas línguas, como a portuguesa ou a inglesa.
- E são incrivelmente inteligentes.

Parabéns! A alternativa C está correta.

Conforme vimos, computadores são incrivelmente rápidos, porém tolos, pois não possuem capacidade de compreensão e discernimento. Inclusive, as instruções que são dadas às máquinas precisam estar de acordo com as linguagens específicas de programação, que definem instruções muito simples e um conjunto limitado de palavras que o computador é capaz de compreender.

Questão 2

Analise as afirmações a seguir:

- I – Computadores são capazes de executar bilhões de instruções por segundo.
- II – Computadores são capazes de discernir e compreender, por exemplo, usando inteligência artificial.
- III – Toda e qualquer capacidade de discernimento/compreensão é ausente em computadores. São as pessoas, dotadas de tais habilidades essencialmente humanas, que escrevem listas de instruções para os computadores, que, embora não façam ideia do objetivo da execução das instruções em questão, geram resultados que fazem sentido para as pessoas.

Está correto o que se afirma em

- A I apenas.
- B II apenas.
- C III apenas.
- D I e III.
- E I, II e III.

Parabéns! A alternativa D está correta.

A questão é um retorno à equação primordial que adotamos: computadores são tremendamente rápidos em sua capacidade de processar e armazenar, mas são tolos, ou seja, são incapazes de estabelecer discernimento e compreensão. Portanto, as afirmações I e III estão corretas.



2 - Processo de exploração do potencial dos computadores

Ao final deste módulo, você será capaz de reconhecer o processo de exploração do potencial dos computadores por meio de códigos e algoritmos.

Entre códigos e algoritmos: aprendendo a lógica do computador

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.

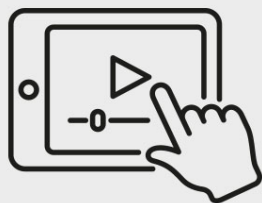
Imagine um motorista que condicionou o seguinte: sinal (farol) amarelo é para parar. Porém, no trânsito, nem todos pensam da mesma forma. Logo, se esse motorista vê um sinal amarelo e para, pode ser que o condutor que vem atrás, por não pensar como ele, não consiga frear. O resultado é a colisão. O ponto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

Código

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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:

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

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.

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**.

Algoritmo

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”.



Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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 inacreditavelmente 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.

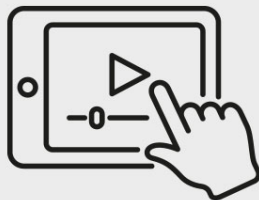
Então, no que o computador é bom? Bem, o computador é bom em ser **rápido e barato**.



Códigos e algoritmos

Entenda melhor os códigos e os algoritmos.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.

Como os computadores evoluem

Lei de Moore

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.



Gordon Moore.

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

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Sobre a diferença entre algoritmos e códigos de computadores, analise as afirmações abaixo e depois marque a opção que relaciona corretamente as afirmações a um código ou a um algoritmo.

- I – Sequência de passos simples e bem projetados para realizar uma tarefa ou resolver um problema.
- II – Pode ser em uma linguagem de programação padrão.
- III – Sequência de instruções que computadores podem executar.
- IV – Deve ser escrito em uma linguagem de programação padrão, senão os computadores não conseguirão executá-lo.

- A I – código; II – código; III – algoritmo; IV – algoritmo.
- B I – algoritmo; II – código; III – código; IV – algoritmo.
- C I – código; II – algoritmo; III – algoritmo; IV – código.
- D I – algoritmo; II – algoritmo; III – código; IV – código.
- E I – algoritmo; II – algoritmo; III – código; IV - algoritmo.

Parabéns! A alternativa D está correta.

A associação entre códigos, algoritmo e construção da linguagem da programação é uma operação relativamente simples, mas vívida para a compreensão. Ao longo do módulo, essas definições foram apontadas.

Questão 2

Analise as afirmações a seguir:

I – O ser humano, que escreve códigos em alguma linguagem para/através de uma sequência de instruções bastantes simples, consegue gerar uma aplicação útil.

II – A inteligência artificial, que escreve códigos em alguma linguagem para/através de uma sequência de instruções bastantes simples, consegue gerar uma aplicação útil.

III – O processador do computador, que processa instruções escritas em alguma linguagem para/através de uma sequência de instruções bastantes simples, consegue gerar uma aplicação útil.

IV – O ser humano, que escreve códigos em alguma linguagem para/através de uma sequência de instruções complexas, consegue gerar uma aplicação útil.

Está correto o que se afirma em

A I e II.

B I e III.

C II e IV.

D I apenas.

E II apenas.

Parabéns! A alternativa D está correta.

Programadores de computadores são os responsáveis por discernir uma aplicação útil e 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. Os computadores, por outro lado, são inacreditavelmente rápidos. Então, programadores e computadores formam uma combinação extremamente poderosa.



3 - Código de instrução de computadores

Ao final deste módulo, você será capaz de empregar códigos simples de instruções de computadores.

Códigos em execução

Preparação

Aqui queremos chegar ao ponto 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. Assim, para que a natureza dos computadores seja entendida, nada melhor do que rodar um pouco de código e ver como ele funciona.

Atenção!

Não se preocupe, pois não veremos casos complexos com um milhão de linhas de código, mas apenas um pouco de código. Somente o mínimo, para que você tenha o primeiro contato com o que é escrever código de computador.

Antes de chegar ao código, será necessário um pouco de paciência. 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.

Assim como usamos alguma língua (português, inglês etc.) para conversar com as pessoas, é necessário usar uma das linguagens disponíveis para escrever códigos que o computador compreenda. Existem várias linguagens de computador diferentes para informar ao computador o que fazer.

Aqui, utilizaremos uma das mais usadas na atualidade, uma versão simplificada da linguagem [Javascript](#), por exemplo, empregando instruções simplificadas para impressão na tela e nas estruturas de repetição. 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.

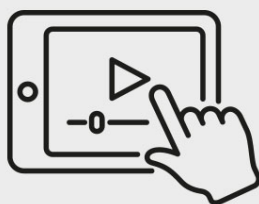
JavaScript

No mundo de computadores, alguns termos, como JavaScript, são nomes próprios e, por isso, não possuem tradução. Não se preocupe, pois rapidamente você estará bem ambientado com esse termo. Situação semelhante ocorre com as instruções reconhecidas por linguagens de programação. Por exemplo, a instrução *print* é reconhecida pela linguagem JavaScript, e ambos os termos não possuem tradução.

Prática 1

Assista ao vídeo para o nosso primeiro exemplo prático.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Para compreender como imprimir [strings](#), veja a Prática 2 a seguir.

trings

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 2 é o uso de comentários na segunda linha.

Prática 2

Clique em Executar e observe o resultado apresentado em Saída.

Código-Fonte

```
print(6, "Teste");  
  
//Imprimir uma string, um núm. e outra  
string  
  
print("Bom dia",2,"Tchau");
```

Rodar/Executar

Saída

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 e valores numéricos corretamente impressos em Saída.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Dica

Um comentário, em JavaScript, começa com duas barras. 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 Executar? Se você mesmo fizer este teste na Prática 2, obterá o resultado ilustrado no emulador a seguir.

Código-Fonte

```
print(6, "Teste");  
  
//Imprimir uma string, um núm. e outra  
string  
  
print("Bom dia",2,"print");
```

Rodar/Executar

Saída

Como a *string* `print` está delimitada por aspas na terceira linha, o computador sabe que se trata apenas de mais um texto a ser impresso, e não de outra instrução *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.

Atenção!

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, alguns serão listados em Saída. Então, é parte da tarefa de quem escreveu o código corrigir esses erros 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 é que, quando alguém está apenas começando a aprender a escrever códigos (programar), será bastante comum se deparar com erros de sintaxe. 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 3

Como vimos, 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 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 seis exemplos imprimirá exatamente as três linhas indicadas em vermelho mais à frente.

```
A
B B
C C C
```

Resultado da saída dos exemplos.

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

Código-Fonte

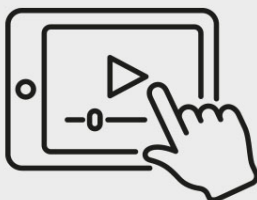
```
print("A");

print("B","B");
print("C","C","C");
```

Rodar/Executar

Saída

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Código-Fonte


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

Rodar/Executar

Saída

Código-Fonte

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

Rodar/Executar

Saída

Código-Fonte

```
print("A");
```

```
print("B""B");  
print("C", "C", "C");
```

Rodar/Executar

Saída

Código-Fonte

```
print("A");
```

```
print(, "B");  
pront("C", "C", "C");
```

Rodar/Executar

Saída

Código-Fonte

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

Rodar/Executar

Saída

A seguir, veja a resolução de cada exemplo:

Solução

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, falta o parêntese direito na terceira instrução.
4. No quarto exemplo, falta vírgula entre as duas letras B.
5. No quinto exemplo, está faltando o primeiro B na segunda linha. Além disso, o terceiro *print* está escrito errado (pront).
6. No sexto exemplo, está faltando o parêntese esquerdo na primeira linha e o parêntese direito na última linha.

Por fim, precisamos abordar mais um conceito básico: **variáveis**. Uma variável no computador é como uma caixa, ou seja, um local onde podemos armazenar valores para uso futuro. Então, se tivermos um código em que atribuímos o valor 7 a uma variável V, o que isso significa é que há uma caixa no computador chamada **V**, conforme ilustrado na tabela a seguir, nós simplesmente podemos armazenar um valor, como um 7, nessa caixa. Também poderíamos armazenar qualquer valor que quiséssemos.

O que significa na prática?

“Caixa” de nome V, em que atribuímos o valor numérico 7:

"Caixa" \leftarrow 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 essa 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.

- Na primeira linha, atribuímos o valor 7 à variável que optamos por chamar de V.
- Na segunda linha, damos um comando para imprimir o valor da variável, que será 7 neste exemplo.
- 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.

Código-Fonte

```
V=7;

print(V);
print("O valor da variável V é:",V);
```

Rodar/Executar

Saída

Notou que, sempre que V é referenciado no código, o computador o substitui pelo valor 7? Esse é o equivalente a ler o valor armazenado na caixa.

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.

Prática 4

Observe o que acontece quando substituímos 7 por 1980 e clicamos em Executar. Note que todos os locais em que V foi referenciado resultaram em 1980 na saída.

Código-Fonte

```
V=1980;

print(V);
print("O valor da variável V é:",V);
```

Rodar/Executar

Saída

Prática 5

Escreva, a seguir, um código que atribua a uma variável chamada N a *string* Guilherme, e depois clique em Executar para obter a seguinte Saída:

- Guilherme Guilherme Guilherme
- Eu conheço um amigo chamado Guilherme

Código-Fonte

Rodar/Executar

Saída

A seguir, veja sua resolução

Solução



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

Java



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 igual e de instruções significa atribuição. Isso é importante, pois, em matemática, o sinal de igual possui um significado diferente.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Para que um código reproduza a saída "13 35 Brasil", qual das alternativas estaria correta?

- A `print("treze","trinta e cinco",Brasil)`
- B `print(13,35,Brasil)`
- C `print(13,35,"Brasil")`
- D `print(13,35,"Brasil"`
- E `print("13", "35", Brasil)`

Parabéns! A alternativa C está correta.

Na opção A, teríamos um erro informando que Brasil não está definido. Além disso, os valores estão escritos por extenso. Na opção B, estão faltando aspas duplas cercado a *string* Brasil, o que geraria uma mensagem de erro. Na opção D, falta o parêntese direito ao final da instrução. Na opção E, os valores estão por extenso e sem aspas duplas.

Questão 2

Qual será a saída do código abaixo?

```
NOME="João"  
print(NOME, "NOME", "NOME");
```

- A João João João
- B NOME João João
- C João NOME João
- D João NOME NOME
- E NOME, "NOME", NOME

Parabéns! A alternativa D está correta.

Em primeiro lugar, vamos analisar o código:

NOME="João" -> Esta instrução atribui o texto passivo (*string*) "João" à variável NOME

Print(NOME, "NOME","NOME"); -> Esta instrução ordena que o computador imprima o conteúdo da variável NOME, depois o texto passivo NOME e novamente o texto passivo NOME. Então, a saída resultante será: João NOME NOME.

Considerações finais

Pensamento computacional parece um tema futurístico. Inclusive, muitos alunos devem ter se lembrado dos filmes de catástrofe, em que a tecnologia se levanta contra o homem. Neste material, você foi tranquilizado sobre essas questões. Primeiro, descobriu que máquinas são terrivelmente ignorantes e profundamente capazes. Somente a partir da interação e da estruturação de um conjunto de linguagens específicas, o grande potencial das máquinas — armazenamento e processamento — pode ser efetivamente estruturado.

Sendo assim, pensamento computacional é um convite para que você entenda como funciona essa relação entre máquinas e homens e, com isso, perceba que não é preciso dominar tecnicamente uma ferramenta apenas em determinado momento, pois, embora elas sejam atualizadas constantemente, a dinâmica e os fins permanecem os mesmos.

Nesse sentido, o que você precisa conhecer, então, são as linguagens que compõem esse novo universo: códigos e algoritmos. Se você entendeu que código são as instruções que o ser humano dá à máquina — mostrando que computadores executam nossas demandas — e que algoritmo é a linha desse comando estruturado para que a máquina interprete e execute, você entendeu a essência da dinâmica comando, forma de comando, execução, novas demandas, comandos e forma de comando, sempre impulsionada pelas demandas humanas.

Para concluir, sugerimos que você faça inúmeros testes (em alguma das práticas acima) a fim de perceber como se relacionou com o assunto, criou e executou a dinâmica proposta. Como uma criança que aprende a engatinhar e a ficar em pé, daqui por diante, busque dar os primeiros passos, entendendo a mecânica, e, em breve, você estará correndo.



Podcast

Ouçá um resumo sobre os principais assuntos abordados no tema.

Para ouvir o *áudio*, acesse a versão online deste conteúdo.



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 andróides**, 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.

Material para download

Clique no botão abaixo para fazer o download do conteúdo completo em formato PDF.

Download material

O que você achou do conteúdo?



Relatar problema

