



Lógica de Programação

O que é, importância e como aprender?



Carta de apresentação

Aprender sobre **lógica de programação** é um dos primeiros passos para quem deseja se tornar um desenvolvedor web.

Mas, afinal, o que é lógica de programação? Basicamente, é uma forma de **pensar sobre como resolver algo**.

Ok, mas é preciso ser um gênio da matemática?

Não, não é preciso. A lógica de programação não está relacionada a cálculos complexos, mas, sobre resolver problemas por etapas. Pode até parecer intimidador, mas o segredo é dar um passo de cada vez.

Como qualquer outra atividade, lógica pode ser treinada.

O praticamente adquire uma capacidade de **pensar cada vez mais de forma estruturada**, principalmente para a elaboração de algoritmos, fundamentais no desenvolvimento de soluções.

O que precisa? Uma boa rotina de estudos e exercícios.

A seguir, vamos falar sobre o que é lógica de programação e indicar **bons caminhos para você praticar!**



Índice

O que é lógica de programação? **04**

Quais os conceitos básicos de lógica de programação? **06**

Lógica de programação e algoritmos: qual a relação? **08**

O que é pseudocódigo? **10**

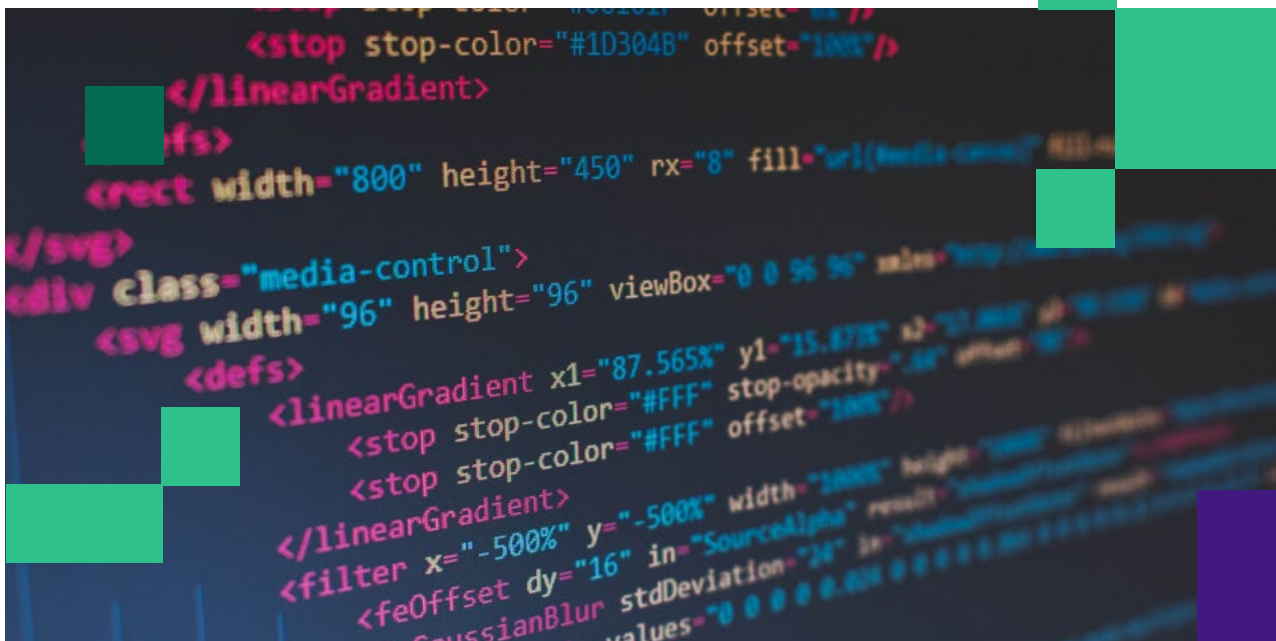
Dicas práticas sobre como aprender lógica de programação **11**

Sites e exercícios para aprender lógica de programação! **13**



01

O que é lógica de programação?



A lógica de programação diz respeito à forma como a pessoa que trabalha com programação escreve uma solução ou um algoritmo utilizando as [linguagens de programação](#) existentes no mercado.

De um modo geral, é como se ela fosse um passo a passo para a execução de uma tarefa (algoritmo), usando para isso uma língua ([linguagem de programação](#)) a fim de codificar como ela deve ser realizada. Essa lógica deve ser feita por quem trabalha com programação, considerando pontos importantes que abordaremos ao longo deste conteúdo.

As regras definidas pela lógica de programação permitem que, por meio de um código escrito, uma máquina consiga converter essas instruções em informação. Uma vez convertida, essa informação poderá ser compreendida por pessoas ou outros dispositivos, por meio das regras estabelecidas por quem desenhou a ideia.



Para conseguir implementar a programação e garantir um fluxo de informação feito de forma lógica, é importante que o sistema contenha as seguintes ferramentas:

■ **Estruturas de controle**

São elas que definem se uma ação pode ou não ocorrer internamente e quais são os parâmetros que definem sua execução ou interrupção;

■ **laço de repetição:**

essa ferramenta é justamente a determinação da condição para que uma ação seja executada ou interrompida, por exemplo: “ande até chegar à Avenida Paulista”;

■ **armazenamento de dados:**

toda ferramenta precisa de dados que alimentam o sistema e que permitem que o algoritmo trabalhe. Diz respeito aos vídeos, imagens, arquivos de texto, memórias de dispositivos eletrônicos, entre outros;

■ **tipos de dados:**

é a categorização das informações internas, mostrando como elas devem ser processadas – por exemplo, de que forma se apresentam o texto, números, frações etc.;

■ **variáveis:**

para quem não está habituado com termos matemáticos, as variáveis são semelhantes a termos matemáticos (por exemplo, “ $x = 5$ ” ou “ $x = y$ ”). Auxiliam no armazenamento de valores que são trabalhados durante a execução dos algoritmos.

A lógica de programação precisa ser nítida e eficiente para evitar ambiguidades: lembre-se de que as máquinas não apresentam subjetividade. Então, caso você não faça as ações da maneira adequada, pode gerar algumas questões problemáticas. Vamos a um exemplo prático que pode ser aplicado na vida cotidiana para compreender as implicações de ambiguidades:

■ você pede para que a pessoa que mora na sua casa vá ao mercado e compre 1kg de carne. Caso tenha limões, também pede para que traga seis;

■ a pessoa retorna para casa com 6kg de carne, porque tinham limões no mercado.

Apesar de ser uma situação anedótica, algo semelhante é passível de ocorrer durante as ações de programação. Caso você não delimite bem a lógica a ser aplicada, o sistema poderá oferecer uma resposta diferente do esperado ao ser executado.



02

Quais os conceitos básicos de lógica de programação?



Para compreender melhor tudo que envolve a lógica de programação, é fundamental conhecer alguns [termos importantes](#) que são utilizados nessa área. Vejamos os principais deles a seguir!

Linguagens de programação

A lógica sozinha não adianta se você não tiver como expressá-la – e isso é possível por meio das linguagens de programação. As linguagens são responsáveis por fazer as ações serem convertidas em linguagem de máquina (bits) e serem gravadas em arquivo, bem como a conversão em ações que podem ser compreendidas pelo usuário (imagens, vídeos, textos etc.).

Por isso, é importante que, além de dominar a lógica de programação, a pessoa que trabalha com esse tipo de função domine o uso das principais linguagens utilizadas atualmente.

Algumas delas são:

JavaScript;

CSS;

PHP;

Java;

C#;

Python;

HTML;

Rails.



Variáveis e constantes

Em termos simples, pense em um armário de arquivos, no qual há pastas com uma etiqueta cada, que armazenam dados sobre determinado tema (idade, nacionalidade, naturalidade, data de nascimento). Essas são as variáveis.

Elas são espaços de memória reservados dentro dos programas. Esse espaço pode ser

alterado ao longo do tempo, bem como as informações podem variar. Por exemplo, quando falamos em “idade”, ela não se mantém igual daqui a um ano, variando ao longo do tempo.

As constantes, em contrapartida, dizem respeito aos espaços de memória reservados dentro dos sistemas, mas que não vão ser alterados ao longo do tempo. O dado “data de nascimento” não mudará ao longo do tempo, sendo uma constante dentro do seu algoritmo.

Tipos de dados

Os tipos de dados, de forma simples, dizem respeito ao formato da informação. Voltemos ao exemplo da pasta no armário: ao abrir uma pasta, você estará em contato com um texto, uma imagem, um CD com vídeo, áudio, números inteiros, números fracionados, entre outros.

Quando falamos em linguagem de programação, dizemos se estamos tratando de um texto, números inteiros ou frações. Seguindo o que mencionamos acima, na variável “idade” teremos apenas números inteiros, bem como na constante “data de nascimento”.

Estruturas de repetição e seleção

Abordamos esse tema anteriormente, mas vale relembrar e aprofundar um pouco mais sobre ele! A estrutura de repetição é o ponto que permite executar, de forma redundante, um mesmo comando ou um conjunto de ações predefinidas no algoritmo, segundo uma determinada condição preexistente.

Pode-se definir que a aplicação gerará uma notificação toda vez que uma pessoa usuária receber uma mensagem. Isso é uma ação redundante, feita por uma estrutura de seleção: 100% das vezes que uma mensagem for recebida, a aplicação deverá gerar uma notificação. Enquanto a condição for verdadeira, essa função deverá ser estabelecida.

Já as estruturas de seleção são aquelas que geram um condicional para as situações, ou seja, apenas nos casos determinados é que a ação deverá ser executada. Dessa forma, torna-se necessário estabelecer afirmações de verdadeiro ou falso, para que a função seja realizada.

Vamos supor que, em sua aplicação, ocorra a venda de um produto em que só é possível garantir frete grátis a partir de R\$99. A partir do momento que o valor final oferecido no carrinho estiver abaixo dessa quantia, a eliminação do valor da tarifa de entrega não poderá ser realizada – trata-se, assim, de um condicional, sendo uma estrutura de seleção.

Em outros termos, entende-se que, sendo o valor acima de R\$ 99 uma condição verdadeira, subtrai-se a taxa. Estando a quantia abaixo do limite estimado, isso caracteriza uma condição falsa e, portanto, a tarifa deve ser incluída.



03

Lógica de programação e algoritmos: qual a relação?



Lógica de programação e [algoritmos](#) são dois conceitos que estão intimamente ligados. Por isso é importante que, ao conhecer o primeiro, você entenda o que é o segundo.

De uma forma simples, os algoritmos dizem respeito a um conjunto de instruções bem definidas, finitas e que, ao serem executadas, precisam gerar um resultado. Por exemplo, em um aplicativo como o Waze, é possível observar os caminhos disponíveis e indicar o mais curto ou com menor tempo de duração, realizar os cálculos de estimativas de tempo de chegada de acordo com a velocidade média, entre outros pontos.

Nesse caso, o resultado é aquele apontado para o motorista: encontrar o melhor trajeto para conseguir maior eficiência no seu trajeto. Isso é possível por meio das diretrizes definidas pelos algoritmos do aplicativo. Ou seja, eles são o roteiro para as ações que serão executadas.

Podemos simplificar ainda mais para tornar o processo mais entendível. Os algoritmos são como uma espécie de receita, com o passo a passo, em lista, que organiza o processo desde o uso do primeiro ingrediente, até obter o resultado – o seu prato pronto.



Esse exemplo ajuda a compreender que há a receita padrão (uma de bolo básico) e podem ocorrer variações de acordo com as necessidades (novo recheio, novo sabor, uma cobertura diferenciada etc.). A mesma situação ocorre com os algoritmos, logo, eles podem mudar de acordo com o que você precisa.

Agora que entendemos como funciona um algoritmo, precisamos conhecer sua base, ou seja, de que forma ele é constituído. Para que o algoritmo consiga executar suas funções é necessário ter uma linguagem de programação de fundo, a fim de que o dispositivo entenda o que deve ser executado.

Afinal, as máquinas não têm uma interpretação subjetiva, sendo necessário determinar as diretrizes para que ela realize o seu processo. Retornando ao exemplo, a pessoa que compõe o algoritmo é quem escreve a receita para que a outra pessoa (no caso, a máquina) possa realizá-la tal como foi mandada (executando o algoritmo).

A lógica de programação, nesse caso, é fundamental para realizar uma codificação do algoritmo realmente eficiente. É preciso que se leve em consideração esse conceito, pois, seu algoritmo pode ser falho e levar ao fracasso da solução caso não o conheça.

Se você, por exemplo, não introduz no seu algoritmo que ele deverá analisar a velocidade média dos carros ao determinar o tempo de duração de uma viagem, bem como introduzir de que forma deverá ser feito esse cálculo, como o sistema fará isso quando for executado? Por isso, a lógica de programação é fundamental para uma boa criação de soluções.

A pessoa que trabalha com esse tipo de atividade precisa ter a capacidade de entender de que modo pode ser possível resolver os problemas que aparecem com os algoritmos, dividindo-os em partes menores e conseguindo solucionar as questões.



04

O que é pseudocódigo?

Quando falamos em programação de algoritmos, estamos caminhando dentro da área de programação. A questão é: a representação deles nessa área pode ser feita de muitas formas. Não há, de fato, uma padronização ou uma regra formal que mostre de que forma eles devam ser feitos – algo que gera um afastamento do conceito formal de programação.

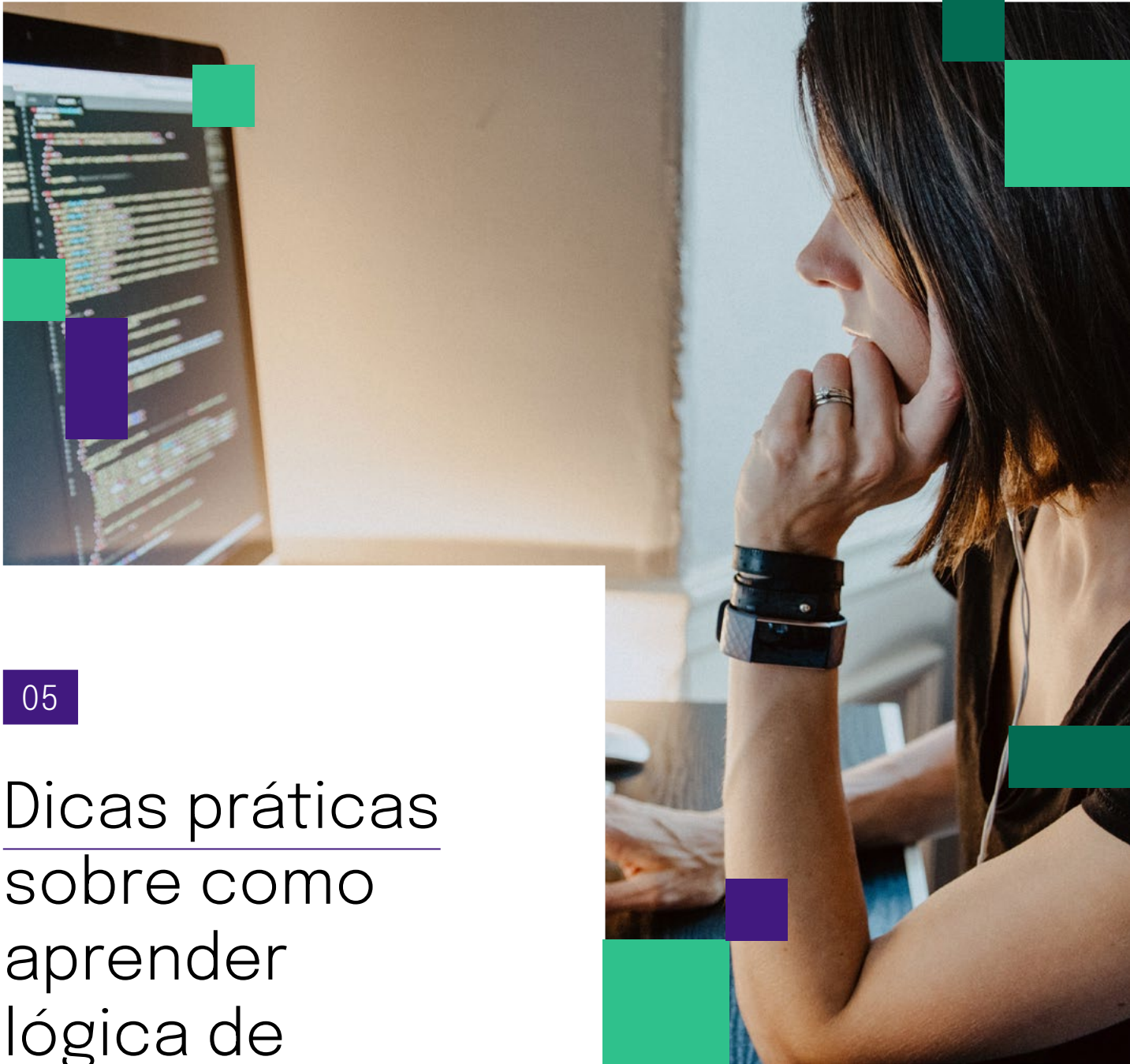
O **pseudocódigo**, também chamado de Linguagem Estruturada, cumpre uma função de intermediação entre esses dois pontos. Ele cria uma espécie de código cujo objetivo é gerar uma aproximação do que é visto nas linguagens padronizadas e tradicionais (como Java, C, C#, entre outras).

Ele transforma as regras definidas anteriormente, utilizadas em nossa linguagem corrente, em algo passível de ser compreendido no processamento das linguagens de programação. No Brasil, um dos pseudocódigos mais conhecidos e utilizados é o Portugol. Entretanto, esse não é o único, já que hoje são encontrados diversos outros no mercado capazes de cumprir com os mesmos objetivos.

Cada tipo de pseudocódigo terá as suas regras específicas para a elaboração da codificação. Por exemplo, no Portugol, as ações devem sempre ser inseridas no infinitivo. Portanto, trazem um conjunto restrito de palavras-chave que são elaboradas na língua nativa da pessoa que opera a programação – e que tenha termos equivalentes nas linguagens de programação padrões.

Assim, a pessoa consegue focar na lógica em si, sendo o pseudocódigo, posteriormente, uma espécie de “facilitador de tradução” para as linguagens de programação. Por exemplo, termos como “escrever”, “ler”, “então” e “isto” são palavras-chave que tendem a estar presentes em todas elas, fazendo com que quem está desenhando o algoritmo consiga fazer uma boa adaptação, sem perder detalhes da lógica de programação.





05

Dicas práticas sobre como aprender lógica de programação



1 - Comece pelos conceitos básicos

Em primeiro lugar, pode ser preciso rever alguns conceitos de lógica matemática, para acompanhar alguns termos que podem ser não muito usuais para quem não está acostumado. É possível revisar esses conteúdos em videoaulas em canais interessantes, como Portal da Matemática, Equaciona Matemática, entre outros.

Após se ambientar com os conceitos, você estará mais confortável para enfrentar a adoção de muitos dos caminhos que são utilizados no ambiente da programação. É aí que chega a hora de conseguir se especializar nessa área!

Os bons cursos de formação para programação incorporam, direta ou indiretamente, os ensinamentos sobre esse tipo de lógica em seus currículos. Isso porque não é possível falar sobre uma boa construção de algoritmos sem passar por tais questões.

Além disso, nesses locais você terá o suporte com monitores e professores especializados no mercado, que poderão dar todo o apoio, encarando as questões e auxiliando em suas dificuldades – conseguindo, assim, melhorar a sua capacidade de aprendizado.



2 - Foque na prática

Outro ponto fundamental é verificar se os conteúdos dos cursos de formação escolhidos permitem que você aprenda sobre lógica de programação na prática.

Afinal, estamos falando de um tema que, naturalmente, é muito abstrato. Logo, ficar com os conceitos parados em sua mente pode gerar dificuldades de colocá-los em execução quando for necessário.

Ao conseguir realizar essa aplicação prática nos cursos, é possível tirar suas dúvidas sobre como executar os processos de lógica de programação. Assim, caso ainda reste alguma dúvida ou tenha alguma falha, os professores poderão orientar sobre a melhor forma de solucionar as questões, melhorando seus estudos e formação.



3 - Rotina de estudos

Além de cursos, é preciso treinar todos os dias. Algumas ferramentas podem ajudar você neste processo, auxiliando a compreender, no dia a dia, como funciona a lógica de programação. Aplicativos específicos como o LightBot podem ser úteis durante sua rotina de estudos.

O jogo em formato de puzzle game é baseado em codificação: por meio das decisões realizadas durante o jogo, você vai aprendendo a lógica de programação, com sequenciamento, sobrecarga, procedimentos, loops recursivos e condicionais. Por ser simples, ele pode ser um auxílio também para pais que querem introduzir seus filhos também nesse cenário. Interessante, não é mesmo?

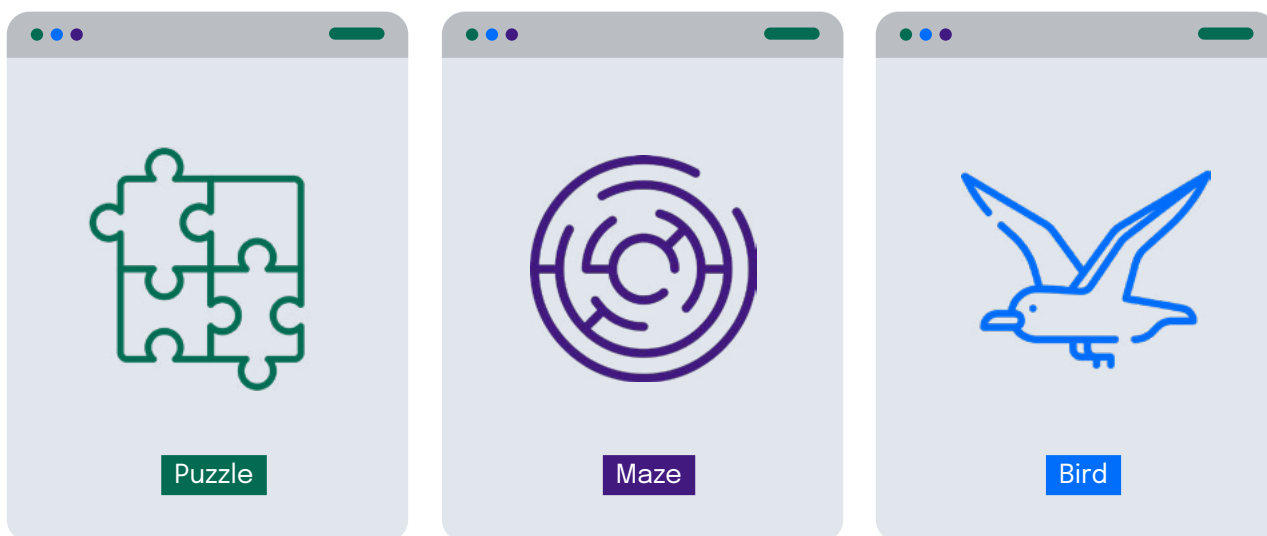


06

Sites e exercícios para aprender lógica de programação!



1 - Site para aprender lógica de programação: Blockly Games



(clique no ícone para acessar a página)



2 - 6 exercícios para aprender lógica de programação

a) Crie um algoritmo que recebe seu nome, último nome e idade e exibe estas informações no seguinte formato: Meu nome é “nome” “sobrenome” e tenho “x” anos de vida.

Resposta:

```
let name = 'Ronaldo';
let lastName = 'Lima';
let age = 36;

console.log('Meu nome é ' + name + ' ' + lastName + ' e tenho ' + age + ' anos de vida.');
```

b) Desenvolva um algoritmo que recebe dois números inteiros, executa as quatro operações matemáticas básicas nestes números (soma,

subtração, multiplicação e divisão) e exibe os resultados no console.

Resposta:

```
let num1 = 10;
let num2 = 4;

console.log(num1 + ' + ' + num2 + ' = ' + (num1 + num2));
console.log(num1 + ' - ' + num2 + ' = ' + (num1 - num2));
console.log(num1 + ' x ' + num2 + ' = ' + (num1 * num2));
console.log(num1 + ' / ' + num2 + ' = ' + (num1 / num2));
```

c) Você foi jantar em um restaurante chique, em uma cidade turística. O prato que você pediu custou R\$100,00. Porém o restaurante possui uma taxa de 10% sobre o valor da conta

por ser um lugar turístico. Crie uma solução para calcular o valor final da conta sendo que você também quis dar 5% de gorjeta para a pessoa que lhe atendeu.

Resposta:

```
let account = 100;

let tip = account * 0.05;

let tourismTax = account * 0.1;

let finalAccount = account + tip + tourismTax;

console.log(finalAccount);
```




d) Você está planejando uma viagem até a praia com colegas. Uma das tarefas para a viagem acontecer é calcular quantos dias ela irá levar. Você está em São Paulo, o João está em Belo Horizonte e a Joana está em Salvador. Levando em conta que o destino do grupo é Natal, no Rio Grande do Norte, crie um algoritmo que calcula a duração de dias de uma viagem levando em conta a quantidade total de horas. O total de dias deve ter precisão de 3 dígitos.

Informações:

São Paulo até Belo Horizonte = 7 horas e 23 minutos;

Belo Horizonte até Salvador = 19 horas e 57 minutos;

Salvador até Natal = 15 horas e 32 minutos.

Resposta:

```
let spBh = 7.23;
let bhSal = 19.57;
let salNat = 15.32;

let totalHours = spBh + bhSal + salNat;

var days = totalHours / 24;

console.log(days.toFixed(3));
```

e) Crie um programa para uma loja de ração, onde ao informar o preço do produto sejam exibidas as formas de pagamento disponíveis. Pagamento à vista possui 10% de desconto e parcelado em 4x possui juros de 15%.

Exemplo:

- Preço do produto: R\$80

- À Vista: R\$72

- 4x de: R\$23

Resposta:

```
let price = 100;

let inCash = price - (price * 0.1);

let creditCard = price + (price * 0.15);

console.log('Preço do produto: R$' + price);
console.log('- À Vista: R$' + inCash);
console.log('- 4x de: R$' + creditCard / 4)
```



f) Crie um programa para gerenciar as contas de um financiamento de veículo. A pessoa que quer fazer o financiamento precisa informar o valor total do carro, o valor da entrada a ser dada e a quantidade de parcelas que ela deseja pagar. A loja em questão usa uma taxa de juros fixa de 5% sobre o valor do financiamento.

Desenvolva a solução e exiba conforme o exemplo (o valor da parcela precisa ter 2 casas decimais após a vírgula):

- Valor Financiado: R\$35000
- Valor do Juros: R\$1750
- Quantidade de Parcelas: 36
- Valor da Parcela: R\$1020.83
- Valor Total do Carro: R\$46750

Resposta:

```
let carPrice = 45000;

let entranceValue = 10000;

let quota = 36;

let financing = carPrice - entranceValue;

let tax = 0.05;

console.log('Valor Financiado: R$' + financing);
console.log('Valor do Juros: R$' + (financing * tax));
console.log('Quantidade de Parcelas: ' + quota);
console.log('Valor da Parcela: R$' + (((financing * tax) + financing) /
quota).toFixed(2));
console.log('Valor Total do Carro: R$' + ((financing * tax) + financing +
entranceValue));
```

@betrybe

