



# Programação: guia completo para os primeiros passos

Aprenda os principais conceitos,  
linguagens e entenda o melhor caminho  
para entrar na área

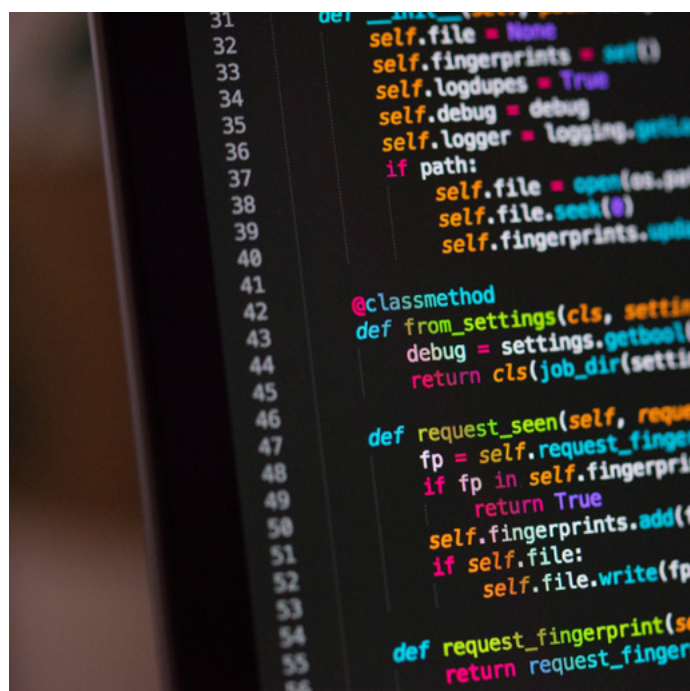


# Carta de apresentação

A programação vem trazendo resultados importantes para o desenvolvimento da área tecnológica [desde 1943](#), quando o conceito da primeira linguagem de programação começou a ser desenvolvido.

Já no início, o objetivo era explícito: **criar uma forma de modelar o mundo real para automatizar algumas tarefas e obter agilidade e comodidade**. A área evoluiu até os dias de hoje, quando seus efeitos podem ser vistos em praticamente todos os lugares.

Em vista disso, **a programação se tornou uma profissão muito visada**. Ela começou a atrair muitas pessoas e a criar uma grande demanda no mercado, principalmente na proporção em que a computação ia se tornando popular. Desde já, aprender mais sobre essa área será um grande diferencial para quem deseja conseguir bons trabalhos e destaque profissional.





# Índice

---

O que é programação?

**04**

Quais os principais conceitos da programação?

**06**

Quais são as 6 linguagens mais usadas e por qual começar?

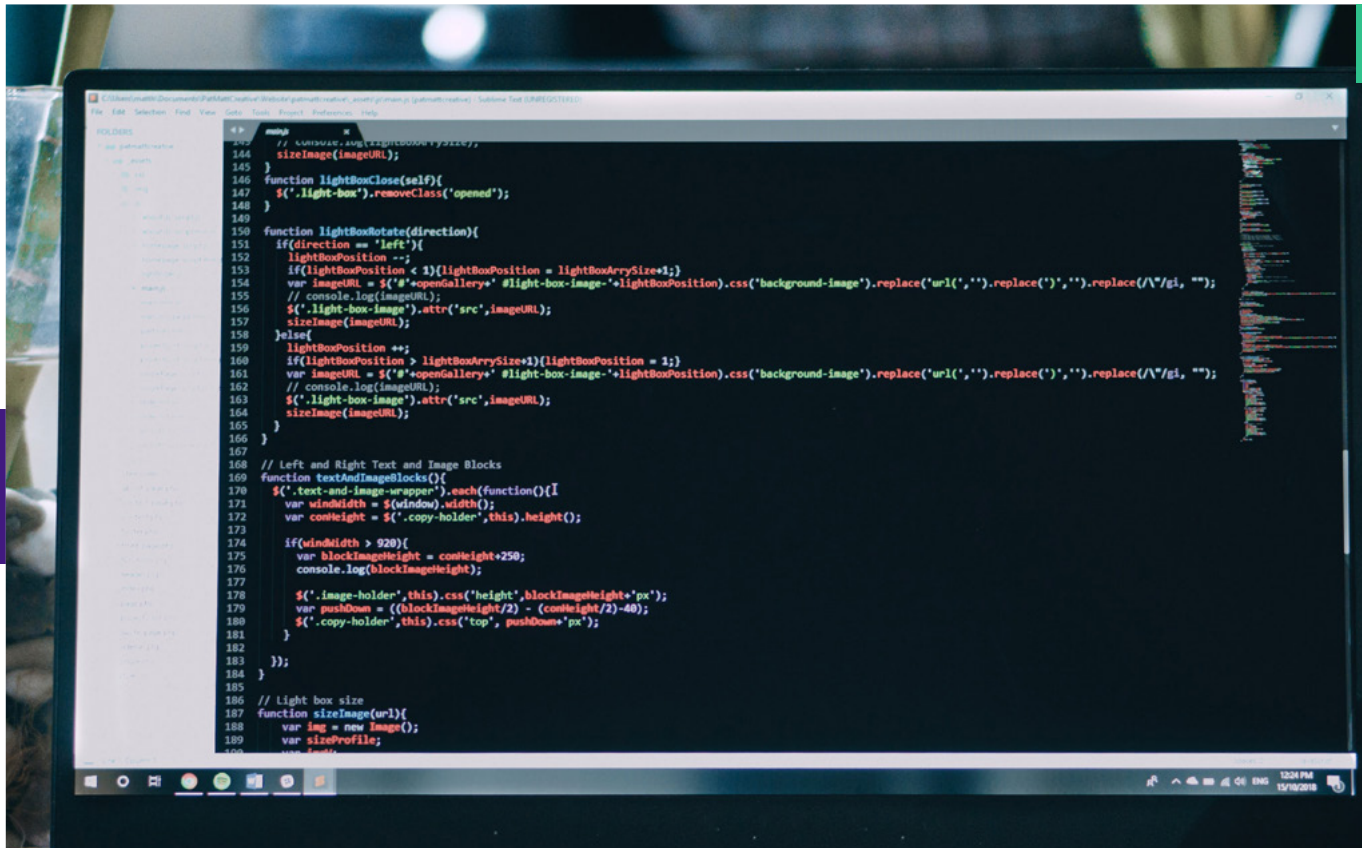
**09**

Dicas para iniciantes: como aprender programação?

**12**

Como é o mercado de trabalho para programação?

**14**



01

# O que é programação?





Podemos definir programação como **o ato de escrever instruções para que máquinas executem uma determinada função ou várias funções encadeadas**. Ou seja, a partir de um conjunto de códigos, as pessoas se comunicam com os computadores e os configuram para a execução de alguma tarefa específica.

Quando paramos para analisar, os computadores não são de fato inteligentes, eles apenas executam processos com uma velocidade extremamente maior do que a nossa. Por isso, precisamos definir regras bem detalhadas e instruí-los cuidadosamente para que realizem algum trabalho, que pode ser calcular números como em uma calculadora ou simplesmente exibir um nome na tela.

**A programação lida com os diversos componentes do computador: o processador, a memória, os elementos de entrada e saída, etc.** Os mesmos componentes serão manipulados, seja para desenvolver um sistema de streaming de música ou para criar um pequeno programa que mostra um número digitado.

Se precisássemos que um computador calculasse dois números, utilizaríamos algumas instruções específicas para esse fim. Por exemplo: em uma instrução, guardamos um número na memória do computador; em outra, salvamos outro número. Por fim, realizamos a soma com mais uma instrução.

Essas regras escritas ajudam as pessoas a dialogarem com as máquinas na linguagem delas. Contudo, isso é feito de uma maneira inteligente: as pessoas programadoras não precisam lidar com os códigos binários que os computadores entendem, mas com linguagens baseadas no idioma humano. **Essas são chamadas linguagens de programação e são responsáveis por permitir essa interação.**



02

# Quais os principais conceitos da programação?



Vamos explorar alguns conceitos principais da noção de programar. Acompanhe atentamente.

## Algoritmo

A **ideia de algoritmo** é estrutural para o que entendemos como programação. Trata-se de uma **sequência de passos para atingir um objetivo**, como se fosse uma receita estipulando todas as etapas para cozinhar algo.

Voltando a este exemplo, se o objetivo é somar dois números, o algoritmo poderia ser: primeiro, escolha um número; segundo, selecione outro número; então, some os dois; em seguida, exiba o resultado da soma. Ou seja, todo

programa é um conjunto de etapas conectadas por uma lógica para gerar um determinado resultado esperado. Cada etapa procede em uma relação de causa e consequência.

A **lógica algorítmica também compreende a ideia de “se-então”, que é tão comum na hora de programar**. As condições criam ramificações que permitem testar possibilidades e estabelecer caminhos a depender do cenário: se Y é maior que 2, então subtraia uma unidade; se Y é menor do que 2, some uma unidade, por exemplo.



## Funções/métodos

Antes de avançar, precisamos solidificar a ideia de funções ou procedimentos. Esses termos variam a depender da linguagem e do tipo de programação. Basicamente, **eles são um conjunto de códigos ou instruções que realizam um trabalho específico**. Tais instruções são agrupadas em um bloco e renomeadas para que sejam chamadas sempre que for conveniente.

Digamos que você já tenha o algoritmo de soma de dois números que mencionamos

anteriormente. Quando for construindo outro sistema, o de uma calculadora completa, você pode precisar realizar a soma várias vezes.

Uma das soluções para isso é reescrever o código de soma toda vez que for necessário. A outra – mais eficiente – é utilizando as funções. Como o código já está escrito e tem um nome (você pode chamá-la de “soma”), é só referenciá-lo no momento específico e obter o resultado.

## Paradigma

O **paradigma** é a maneira de construir sistemas com uma linguagem de programação. É a forma como os códigos são estruturados e modelados, de modo a gerar uma solução para um determinado problema. **Temos três principais: imperativo, orientado a objetos e funcional.**

O **imperativo** é marcado por instruções sequenciais, em que há um grande foco na sequência em si. Os programas são executados seguindo essa lógica direta, linha por linha. É um tipo simples, que não envolve grandes abstrações e que, portanto, materializa a noção de algoritmo que descrevemos.

O **paradigma orientado a objetos** busca modelar o mundo real em classes e objetos. Cada classe se comporta como um modelo que representa algo do mundo real, e cada objeto é uma instância única da classe. Além disso, existem métodos que são responsáveis por tarefas específicas dentro dos programas. A orientação a objetos envolve programas estruturados em diversas pastas e arquivos.

É comum que cada classe esteja configurada em um arquivo.

Para entender a modelagem orientada a objetos, vamos utilizar um exemplo de sistema gerenciador de vendas de uma loja, o chamado ponto de venda. O objetivo desse software é registrar vendas, gerar notas, assim como permitir que as pessoas colaboradoras consigam controlar o estoque. Portanto, a compra pode ser uma classe, ao passo que produtos podem ser outra.

Em cada classe, definimos as características, que são os atributos. As características de uma compra podem ser os produtos comprados, o preço, a data, entre outros.

Por fim, temos o **paradigma funcional**.

Ele é baseado em funções e trabalha de maneira mais direta e intuitiva, com foco em resolução de problemas específicos. Logo, é uma forma de programar que evita estados e que transforma o fluxo em chamadas de pedaços de código que se responsabilizam por determinadas tarefas.



## Compilação/interpretação

Em programação, compilação é bem diferente de interpretação. Na compilação, existe um software intermediário (o compilador) que lê todo o código de um arquivo e o transforma em um executável

capaz de ser gerenciado pela máquina.

Já o interpretador analisa linha por linha e executa cada uma por vez. A compilação pressupõe análises anteriores à execução, o que não existe na interpretação.

## Variáveis e constantes

Uma variável é **um espaço de memória para armazenar uma ideia ou um dado específico**. Por ser armazenado, esse dado pode ser sempre resgatado quando referenciado pelo nome da variável.

São úteis para realizar operações: em uma calculadora, é possível salvar números em variáveis antes de fazer os cálculos entre eles.

Dessa forma, não é necessário citar o valor da variável o tempo todo, só mesmo o nome dela.

Em um arquivo com várias instruções, uma variável pode ser alterada a qualquer momento, recebendo novos valores na medida em que for conveniente. **Contudo, existem variáveis que não mudam seus valores, e a essas damos o nome de constantes.**

## Front-end e back-end

**Front-end e back-end** são terminologias para a programação web. **O front corresponde ao desenvolvimento realizado no lado de clientes**, isto é, na parte em que as alterações são visíveis para a pessoa usuária. Os botões de um menu, as interações do mouse e pop-ups são exemplos de mudanças de visual, portanto, específicas do lado de clientes.

**O back-end cuida das operações do lado do servidor, que entrega os dados a clientes**

**por meio da internet.** Manipulações de bancos de dados, validações de formulários e outras funções mais complexas são típicas desse lado.

O front-end é a parte que todo mundo vê. Em uma analogia com um programa de televisão, é o que está à mostra e o que é filmado pela câmera. O back-end então seria o sustentáculo ou, na TV, o conjunto de atividades que acontecem nos bastidores – como as pessoas que configuram as câmeras.

## Framework

Um framework é um pedaço de código que realiza alguma função específica.

**São padrões de codificação que resolvem um problema de uma forma simples e que podem ser importados para agilizar o processo em outros programas.**





03

Quais são as  
6 linguagens de  
programação  
mais usadas e por  
qual começar?



*Nesta seção, vamos conhecer as principais linguagens de programação do mundo moderno e suas características. É importante ressaltar que novas linguagens surgem sempre, mas essas são as mais conhecidas atualmente.*



## 1 - JavaScript

A linguagem mais utilizada do momento e ideal para

**começar.** O JavaScript é uma tecnologia interpretada, voltada para o desenvolvimento web, principalmente em aplicações no lado de clientes. Quando surgiu, o objetivo da tecnologia foi trazer interatividade às páginas, com a criação de pop-ups, por exemplo.

Nos dias atuais, ela **é aplicada para gerenciar toda a lógica da web no front-end**, já que trabalha com o HTML e o CSS, e essas duas não são linguagens de programação. Portanto, a lógica fica sempre concentrada nos códigos de JS, o que envolve modificação de componentes HTML e CSS em muitos momentos.

Além do lado cliente, o JavaScript atualmente se tornou ainda mais versátil por conta dos frameworks – que surgem com uma frequência muito alta. Já existem frameworks que estendem a linguagem para o lado do servidor também, bem como para aplicativos desktop.



## 2 - PHP

O PHP é uma linguagem **portável e versátil, voltada para desenvolvimento web**. É uma tecnologia para back-end de projetos online.

Para isso, ela **permite programação orientada a objetos ou imperativa, sempre com ótima performance**. Outra de suas características é a simplicidade para aprendizado, bem como

ser de código aberto – as próprias pessoas que usam fazem mudanças na linguagem de forma comunitária.

É interessante ressaltar que o PHP pressupõe o domínio de outra linguagem, que não é de programação: o HTML. O PHP trabalha diretamente com essa tecnologia, inclusive manipulando seus componentes. A boa notícia é que o HTML também é um padrão bem simples de aprender.



## 3 - C

A linguagem C é uma das mais tradicionais que estão em uso atualmente. Criada em 1972, por

Dennis Ritchie, foi a base de outras linguagens comuns, como o Java e Python. **É imperativa, de alto nível, mas também une alguns recursos de baixo nível para oferecer maior versatilidade e poder de gerenciamento.**

Quando surgiu, era diretamente comparada com Assembly, uma linguagem de baixo nível bem complexa que existia. Por isso, rapidamente se tornou um sucesso, já que é muito mais simples que Assembly.

Além das linguagens que mencionamos acima, sistemas operacionais e programas diversos foram desenvolvidos com C. Até hoje, **é muito usada para programação de sistemas de hardware com pouca memória, que são os projetos embarcados.**



#### 4 - C++

Essa linguagem multiparadigma e portátil (executada em diversas plataformas diferentes) foi desenvolvida nos anos 80 para ser um complemento à C. Afinal, com o avanço da orientação a objetos, a C ficou um pouco obsoleta.

A linguagem C++ surgiu, então, para **acrescentar paradigmas e permitir versatilidade no desenvolvimento de soluções.**



#### 5 - Java

Surgiu e ficou famosa nos anos 90. A linguagem Java **impressionou o mercado por ser orientada a objetos, simples e cheia de recursos que ajudavam quem programava**, como bons **depuradores** (identificadores de erros), ferramentas para construção de interfaces, testes, etc. É conhecida também por ser portátil e apresentar bom desempenho.



#### 6. Python

O Python é uma linguagem muito simples, moderna e poderosa. Surgiu com uma filosofia voltada a facilitar a vida de quem programa com melhor legibilidade e praticidade. Assim, **é uma ferramenta que permite uma curva de aprendizado menor, com ganho de clareza, portabilidade e versatilidade.** Além disso, é multiparadigma.

**A tecnologia é muito usada na atualidade para aplicações de inteligência artificial e estatística.** Contudo, pode ser aplicada para web, **games** e desktop também. Existem bibliotecas que auxiliam bastante a depender da necessidade, o que agiliza o processo.



04

## Dicas para iniciantes: como aprender programação?



**O aprendizado de programação é uma jornada. É preciso foco, disciplina e saber o que é importante priorizar.**

Uma boa dica é saber onde buscar a informação para chegar lá. **Livros são sempre uma boa recomendação**, já que ensinam bastante os fundamentos teóricos cruciais e ajudam com exemplos práticos. Uma formação baseada em bons livros ajuda profissionais a desenvolverem boas práticas e se adaptarem bem ao [mercado](#).

Além disso, é recomendado procurar bons [cursos sobre o assunto](#). A vantagem dos cursos é que existem vários, desde os mais gerais aos mais específicos que tratam de tecnologias especiais.

**Vale citar também os fóruns**, que são importantes por conta do apoio de colegas e outros estudantes. Pessoas respondem dúvidas e problemas o tempo todo, incluindo tópicos complexos e questões técnicas bem pontuais. A interação com outras pessoas é



a chave para o sucesso na programação. Por essa razão, é importante também criar uma rede de [networking](#).

Você pode estar se perguntando: quanto tempo leva para realmente aprender a programar? A resposta requer algumas considerações.

Aprender uma [habilidade](#) como essa significa ser capaz de solucionar problemas com uma certa fluência, e isso se consegue com um esforço, em média, de um ano. Contudo, o aprendizado vai além do período de exposição inicial e se estende ao resto da carreira das pessoas que atuam nessa profissão.

Sem falar que novas tecnologias e padrões surgem a todo instante. Por conta disso, é preciso estar sempre se atualizando para buscar novas referências e aprender melhores estratégias para lidar com as questões do dia a dia.

Logo, sempre haverá oportunidade para crescer e evoluir. O ideal é sempre que possível se expor a desafios diversos e situações novas para dominar a linguagem e se tornar ainda mais versátil.





05

## Como é o mercado de trabalho para programação?



Com a transformação digital, as companhias precisam ainda mais de digitalização de seus processos e de sistemas de gestão e automação.

Por esse motivo, **o mercado de programação encontra-se bastante aquecido**. Praticamente todas as empresas apresentam demandas de software em diversos setores diferentes.

Ademais, como já falamos, novas tecnologias estão surgindo a todo tempo

e, conseqüentemente, pessoas que saibam lidar com elas. Isso faz com que novos cargos surjam no mercado.

Seja em [startups](#) ou fábricas de software, sempre tem espaço para profissionais de qualidade na programação. As empresas **valorizam profissionais que dominam aspectos de gestão e saibam [trabalhar em equipe](#)**.

**O desenvolvimento web também está em alta devido ao aumento do interesse de**



**organizações por mais espaço na internet.**

Da mesma forma, o uso de smartphones pelas pessoas tem gerado uma demanda maior por [desenvolvimento mobile](#).

A programação é uma área riquíssima e poderosa, que tem conquistado diversas pessoas e [mudado carreiras](#). Como vimos, ela carrega vários conceitos e princípios fundamentais que devem ser dominados por quem quer se dedicar a essa profissão. A

sua importância é percebida em todo lugar, com diversas linguagens protagonizando o mundo digital.

Para [avançar na carreira](#), é preciso foco e esforço, bem como busca por cooperação com colegas. A questão que deve ser lembrada é: sempre é necessário continuar aprendendo e evoluindo.

@betrybe

