

# PRIMEIROS PASSOS

---

E-Book

# Introdução

Vamos iniciar nosso estudo com os principais fundamentos de **lógica**. Talvez não pareça, mas ela está envolvida em praticamente todas as atividades de nosso cotidiano. Quando a sua mãe faz um bolo de chocolate, faz de acordo com uma sequência de passos (que conhecemos como "receita") que faz sentido para que o bolo realmente fique pronto e gostoso. Ela não pode, por exemplo, assar o bolo antes de colocar algum ingrediente imprescindível na massa!

Outro exemplo: quando você decide entre jogar videogame ou estudar, há um raciocínio lógico envolvendo premissas e conclusões. Seu raciocínio provavelmente lhe levará à conclusão de que, quanto mais estudar, maior será a sua chance de sucesso. Você também considerará que é um ser humano e que precisa de tempo para trabalho e diversão.

Esses exemplos foram para dizer que a lógica faz parte do nosso dia a dia, e quando desejamos desenvolver soluções computacionais, a lógica está presente todo o tempo. Além disso, o desenvolvimento delas envolve o uso de computadores, que são máquinas que requerem que instruções sejam expressas de maneira muito clara e não ambígua para que possam operar como esperamos e resolver os problemas que desejamos. O desenvolvimento de toda solução computacional (como um sistema acadêmico, uma rede social, um aplicativo de paqueras, um aplicativo de trocas de mensagens instantâneas etc.) envolve muita lógica. Animados?

**FIGURA 1 - É muita coisa que vamos aprender!**





Fonte: Elaboração dos autores, 2020.

## Lógica proposicional

Uma das variedades mais elementares de lógica é a proposicional. Nós a aplicamos diariamente para, por exemplo, tomar decisões sobre o que devemos fazer mediante determinadas circunstâncias ou fatos observados. Quando estudamos lógica, dizemos que uma proposição é um conteúdo ao qual podemos atribuir o valor verdadeiro ou o valor falso. Uma frase declarativa, por exemplo, pode expressar uma proposição. Há, é claro, frases que não expressam conteúdo, cujo valor pode ser avaliado como verdadeiro ou falso e que, portanto, não expressam proposições.

### Exemplos de frases que expressam proposições

1. Amanhã vai chover ou nevar.
2. Está muito quente para nevar.
3. Se hoje for domingo, então não tenho de ir trabalhar.
4. Hoje é domingo.
5. Eu vou trabalhar no escritório hoje ou amanhã.
6. Vou ficar em casa hoje.

### Exemplos de frases que não expressam proposições

1. Seja prudente.
2. O homem feliz.
3. Boa sorte!
4. Faça o bem sem olhar a quem.

Vamos nos concentrar somente no estudo de frases que expressam proposições. A **Lógica Proposicional** recebe esse nome por se tratar de um formalismo matemático cujos elementos fundamentais são as proposições.





**Lógica Proposicional é um assunto amplamente estudado com vasta bibliografia de qualidade. Reforce seus conhecimentos lendo as páginas 34 a 41. Bons estudos!**

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/150814>

SOUZA, J. A. L. de. Lógica Matemática. São Paulo: Pearson Education do Brasil, 2016.

---

## Raciocínio dedutivo

Como vimos, frases declarativas que expressam proposições são, muitas vezes, fatos do nosso cotidiano. Elas podem envolver condições climáticas, condições para ir ao trabalho ou não e assim por diante. Um mecanismo fundamental da lógica proposicional é o raciocínio dedutivo. Como seu nome sugere, ele é utilizado para, em função de proposições existentes, deduzir o valor lógico (verdadeiro ou falso) de outras proposições. Anteriormente, estudamos sobre frases que expressam proposições. Vejamos novamente os exemplos dados e frases que expressam proposições cujos valores lógicos podem ser deduzidos em função deles.

Do ponto de vista de lógica, os assuntos sobre os quais as frases que expressam proposições tratam têm pouca relevância. O que importa, de fato, é o valor lógico daquilo que elas expressam. Por essa razão, quando desejamos aplicar o raciocínio dedutivo para chegar a conclusões de interesse, é muito mais prático que os conteúdos que expressam proposições sejam apenas letras, como **P**, **Q**, **R** etc. De fato, é possível aplicar o raciocínio dedutivo e chegar ao valor lógico de uma proposição em função do valor lógico de outras, mesmo quando não há um contexto que possa ser inferido a partir de frases que tratam de eventos do cotidiano, como as que vimos até então.

Considere que **P** e **Q** são proposições. Perceba como a conclusão lógica obtida pela aplicação do raciocínio dedutivo faz sentido.



Proposição 1: Ou P é verdadeira ou Q é verdadeira.

Proposição 2: P é falsa.

Conclusão obtida por meio do raciocínio dedutivo: Q é verdadeira.

Acompanhe alguns exemplos ilustrados na Figura 3.

**FIGURA 2 - Exemplos de proposições**

Proposição 1	Proposição 2	Conclusão
Amanhã vai chover ou nevar.	Está muito quente para nevar.	obtida por meio do raciocínio dedutivo: Amanhã vai chover.
Proposição 1	Proposição 2	Conclusão
Se hoje for domingo, então não tenho que ir trabalhar.	Hoje é domingo.	obtida por meio do raciocínio dedutivo: Não tenho que ir trabalhar hoje.
Proposição 1	Proposição 2	Conclusão
Eu vou trabalhar no escritório hoje ou amanhã. .	Vou ficar em casa hoje.	obtida por meio do raciocínio dedutivo: Vou trabalhar no escritório amanhã.

Fonte: Elaboração dos autores, 2020.

## Conectivos Lógicos

O exemplo anterior mostra como as proposições podem ser simplificadas quando representadas apenas com letras. Podemos simplificar as expressões ainda mais introduzindo os **conectivos lógicos** que permitem realizar operações lógicas sobre proposições. As operações lógicas fundamentais são negação, conjunção, disjunção, condicional e bicondicional.

Como vimos, a cada sentença da lógica proposicional pode ser atribuído um único valor, chamado valor verdade, que pode ser verdadeiro ou falso. O valor verdade de uma sentença composta por conectivos lógicos é definido pela tabela verdade de cada conector.



**Para aprender mais sobre os conectivos lógicos e tabelas-verdade, leia as páginas 42 a 49 do nosso livro texto para este assunto.**

<https://plataforma.bvirtual.com.br/Leitor/Loader/150814>

SOUZA, J. A. L. de. Lógica Matemática. São Paulo: Pearson Education do Brasil, 2016.

Para desenvolvermos programas eficientes, a lógica proposicional é uma ferramenta poderosa. Existem vários conceitos importantes que nos permitem elaborar adequadamente várias sentenças da programação, independentemente da linguagem utilizada.

Às vezes fazemos perguntas que nem precisaríamos fazer, existem algumas afirmações que serão sempre verdadeiras ou sempre falsas. Conceitualmente estamos falando de:

- **Tautologia:** uma proposição composta cuja última coluna da tabela verdade possui somente valor **verdadeiro**, independentemente dos valores das proposições simples componentes.
- **Contradição:** uma proposição composta cuja última coluna da tabela verdade possui somente valor **falso**, independente dos valores verdade das proposições simples componentes.

E quase sempre é possível encontrar expressões mais simples que outras que representam uma condição num programa de computador, isso é possível porque existe equivalência lógica entre expressões.

Dizemos que duas proposições são **logicamente equivalentes** quando as colunas resultantes das tabelas-verdade de cada uma delas são idênticas.



Quer saber mais sobre essas ferramentas poderosas? Faça uma leitura atenta das páginas 82 a 86 do livro indicado.

<https://plataforma.bvirtual.com.br/Leitor/Loader/150814>

SOUZA, J. A. L. de. Lógica Matemática. São Paulo: Pearson Education do Brasil, 2016.



### Você Sabia?

George Boole nasceu em 1815, era filho de sapateiro, com pouca chance de ingressar em uma faculdade. Foi na Matemática, ensinada por seu pai, que ele encontrou sua verdadeira vocação. Ele passou a estudar as operações matemáticas de forma diferente, separando todos os símbolos das coisas sobre as quais eles operavam, com o intuito de criar um sistema simples e totalmente simbólico.

A Lógica de Boole (ou Lógica Booleana) utiliza um sistema numérico binário que incorpora as propriedades básicas do Cálculo Proposicional e da Teoria dos Conjuntos. Na época de sua descoberta não foi utilizada. Com o surgimento do computador, a utilização do sistema binário tornou-se indispensável.



# Algoritmos

Um **algoritmo** pode ser definido como uma sequência finita de passos lógicos que visam atingir um determinado objetivo. Um exemplo clássico é um algoritmo para trocar uma lâmpada pendurada no teto (Figura 4).

**FIGURA 3 - Substituição de lâmpada**



Fonte: Elaboração dos autores, 2020.



**Você pode ver mais algoritmos nas páginas 5 a 8 do livro sugerido.**

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/323>

FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005.

Existem ainda diversas formas de representação de algoritmos, além dos pseudocódigos mostrados: os **fluxogramas tradicionais**, os **diagramas de Chapin**, entre outros.



Veja exemplos nas páginas 10 e 11.

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/323>

FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005.

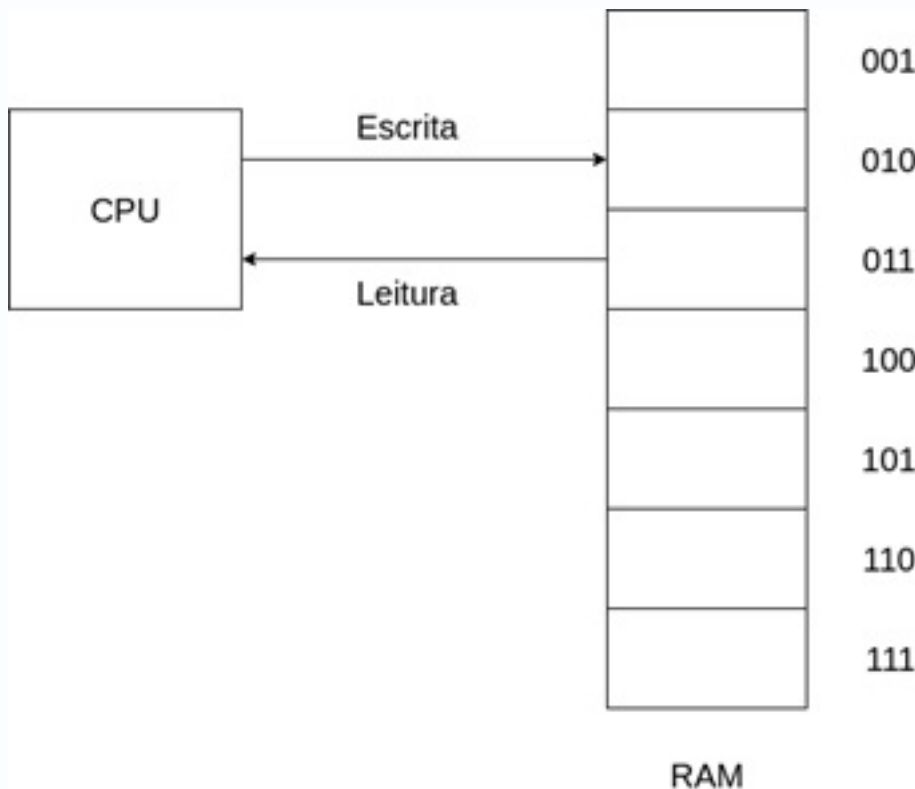
---

## Elementos fundamentais da arquitetura de computadores

Já estudamos a lógica e a representação de algoritmos que fundamentam o raciocínio lógico para o desenvolvimento de *softwares*. Agora vamos dar uma olhada em alguns elementos básicos do hardware que sustentam a programação de computadores.

Programar um computador significa especificar o que ele deve fazer. Para isso, precisamos conhecer as partes mais importantes dele, que são a memória e a CPU. Além disso, precisamos conhecer como ambas interagem, que é através de operações comumente chamadas de **leitura** e **escrita**. A Figura 5 ilustra essa interação. Note que a memória é dividida em pequenos blocos, cada qual com o seu endereço. Em cada um deles podemos guardar um dado de interesse de modo a viabilizar a solução computacional para algum problema.

**FIGURA 4 - Interação entre CPU e memória RAM**

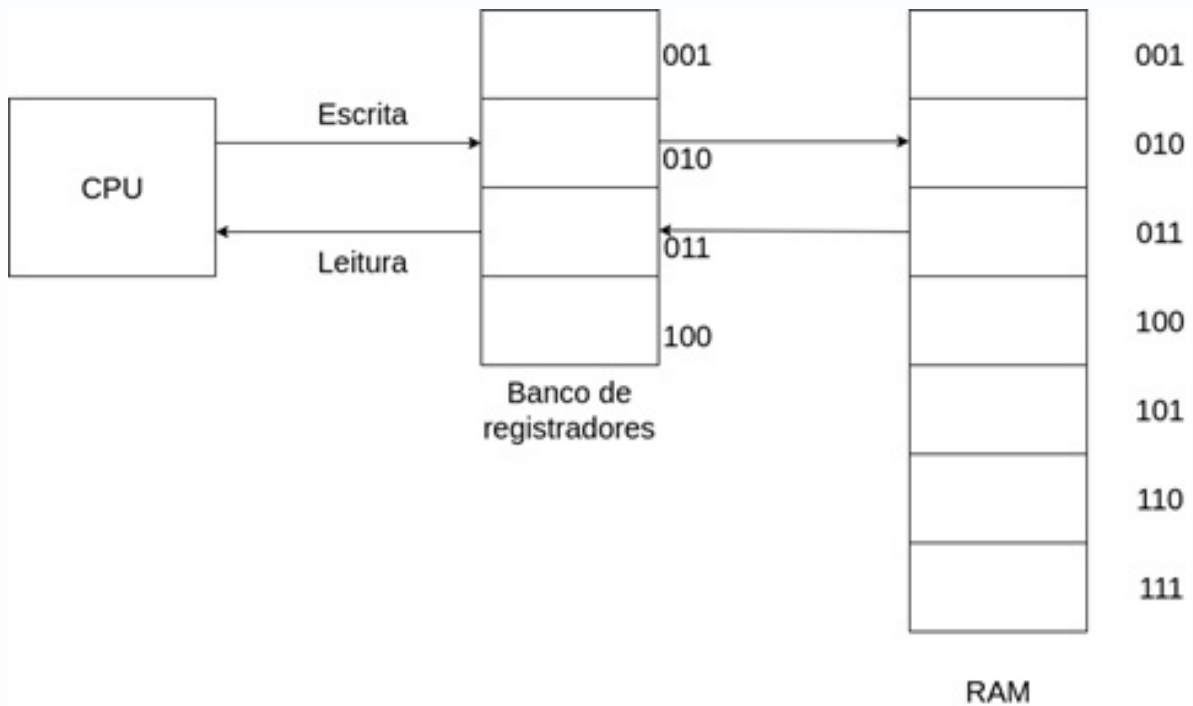


Fonte: Elaboração dos autores, 2020.

## O banco de registradores

A Figura 5 ilustra os componentes fundamentais de um computador. Porém, há um detalhe muito importante a ser citado. A velocidade de processamento da CPU é muito maior do que a velocidade com que a memória é capaz de disponibilizar os dados para que o processamento ocorra. Por exemplo, para calcular 10% de aumento sobre o salário de uma pessoa que está armazenado na memória, a CPU precisa fazer uma operação de leitura nela. Por ser muito mais rápida do que a memória, a tendência é que a CPU passe algum tempo ociosa aguardando até que o salário esteja disponível para ela poder processar, o que é um desperdício computacional. Para resolver esse problema, os computadores empregam um recurso conhecido como **hierarquia de memória**. O banco de registradores faz parte dele. Trata-se de um bloco de memória parecido com a memória RAM, porém menor e muito mais rápido. Ele fica entre a CPU e a memória. Veja a Figura 6, é possível perceber que a fim de realizar alguma computação, é preciso transportar dados da memória RAM para o banco de registradores para que só então a CPU possa manipulá-los.

**FIGURA 5 - Banco de registradores**



Fonte: Elaboração dos autores, 2020.

Também na próxima figura é possível verificar que cada posição da memória e dos registradores é identificada por um endereço bem definido. O bom é que o desenvolvedor que utiliza **linguagens de programação de alto nível** não tem que se preocupar com a existência do banco de registradores para fazer uso apropriado dele, o que seria muito inconveniente, difícil e propenso a erros, pois teria que lembrar os endereços em que está armazenado cada um dos valores de interesse para seu programa.

**Linguagens de programação de alto nível** permitem ao programador desenvolver seus programas sem se preocupar com muitos detalhes de *hardware*. Dizemos que elas são mais próximas do ser humano e mais distantes dos detalhes de funcionamento do computador.

## Variáveis

Um dos recursos mais importantes existentes em uma linguagem de alto nível é a **declaração de variáveis** que permite que sejam atribuídos nomes ou apelidos a posições de memória de modo que não seja necessário lembrar de seus endereços.



Quer saber mais? Você encontra exemplos e uma boa explicação nas páginas de 14 a 18.

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/323>

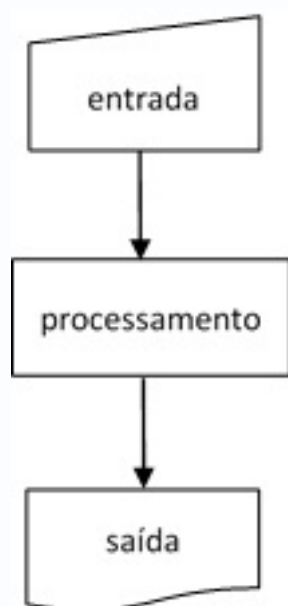
FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: A construção de algoritmos e estruturas de dados. 3. ed. [S.l.]: Prentice Hall, 2005.

Porém, é necessário que seja feita uma espécie de **tradução** do programa escrito em alto nível para uma linguagem que o computador entenda. Em geral, esse processo é feito por um componente de software chamado **compilador**.

## Algoritmos básicos

Um algoritmo básico ou de sequência simples tem a seguinte estrutura: entrada – processamento – saída. A Figura 7 ilustra um fluxograma para essa estrutura.

FIGURA 6 - Algoritmo básico





Fonte: Elaboração dos autores, 2020.

**Um exemplo simples:** somar 2 números escolhidos pelo usuário e exibir o resultado.

### Entrada

São as informações fornecidas ao programa para que ele seja executado.

Nesse exemplo, são os 2 números fornecidos pelo usuário.

Ler 2 números e armazená-los em variáveis:

*primeiroValor* e *segundoValor*

### Processamento

É a razão de ser do programa.

No exemplo é a realização da soma, também armazenando o resultado em uma variável.

*soma = primeiroValor + segundoValor*

### Saída

Exibição do resultado obtido.

Para o exemplo, exibir o valor da soma.

*exibir resultado*

Note que mesmo um algoritmo simples como esse utilizou recursos poderosos da programação:

- operadores aritméticos;
- operadores de atribuição;
- comando de entrada;
- comando de saída.



**Novamente, você pode ter uma boa ideia sobre esses elementos, consultando as páginas 25 a 27 da nossa bibliografia.**

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/323>

FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005.

A partir do que estudamos aqui, convidamos você a pensar em uma tarefa cotidiana. Anote essa tarefa em um papel e pense como você inicia, desenvolve e finaliza essa tarefa.

Refleta: é possível fazer diferente? Tem sentido ser assim? Quais são as variáveis dessa tarefa?

Analisar pequenas situações do cotidiano poderá desenvolver em você a capacidade de raciocínio lógico.

