

---

---

PROGRAMAÇÃO DE COMPUTADORES I – BCC701

CONTEÚDO TEÓRICO EM LINGUAGEM PYTHON

**MÓDULO 1 – INTRODUÇÃO**

---

---

2020/1

ELABORADO PELA COMISSÃO DE UNIFICAÇÃO DA DISCIPLINA BCC701,  
COM A COLABORAÇÃO DE PROFESSORES E ESTAGIÁRIOS DOCENTES  
<http://www.decom.ufop.br/bcc701/>

DECOM – DEPARTAMENTO DE COMPUTAÇÃO  
ICEB – INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
UFOP – UNIVERSIDADE FEDERAL DE OURO PRETO

# Sumário

|          |                                  |          |
|----------|----------------------------------|----------|
| <b>1</b> | <b>Introdução</b>                | <b>2</b> |
| 1.1      | Como funciona um computador?     | 2        |
| 1.2      | Programas de Computadores        | 2        |
| 1.2.1    | Linguagem Python                 | 3        |
| 1.3      | Programação Estruturada          | 3        |
| 1.3.1    | Sequência                        | 3        |
| 1.3.2    | Decisão                          | 4        |
| 1.3.3    | Iteração                         | 4        |
| 1.4      | Algoritmos e Fluxogramas         | 5        |
| 1.4.1    | Exemplo 1                        | 5        |
| 1.5      | Exercícios Propostos             | 7        |
| 1.6      | Solução dos Exercícios Propostos | 9        |

# Introdução

O presente material foi elaborado pela Comissão de Unificação da disciplina de Programação de Computadores I (BCC701) com a colaboração de professores e estagiários docentes do DECOM durante o semestre letivo 2019-02. Nele procuramos apresentar todos os conceitos e recursos mínimos a serem abordados no conteúdo programático da disciplina. Professores podem abordar os assuntos à sua maneira, expandindo determinado conteúdo quando considerar necessário, assim como alterando a ordem de apresentação dos mesmos.

Neste módulo introduzimos alguns conceitos importantes sobre o funcionamento dos computadores, programação de computadores, programação estruturada e algoritmos.

## 1.1 Como funciona um computador?

Para compreender melhor o conteúdo da disciplina, é preciso entender o que são os **programas de computadores** e o que significa **programação estruturada**.

Um computador funciona baseado em 3 principais elementos: **Processador**, **Memória** e **Armazenamento**. Assista a este vídeo para compreender melhor este assunto: <https://youtu.be/MpKbTNonIwc>.

## 1.2 Programas de Computadores

Um programa de computador é uma sequência de instruções que um computador segue para realizar uma tarefa. Diversas tarefas do nosso dia-a-dia seguem esse tipo de lógica: receitas culinárias; instruções de um manual; execução de uma peça de música; etc.

De maneira similar a um músico que executa uma peça a partir de uma partitura, um computador precisa de um conjunto de instruções para executar alguma tarefa. Essas instruções estão contidas em um arquivo denominado **código fonte**. Assista a este vídeo para compreender melhor este assunto: <https://youtu.be/hD42nKsCimQ>.

Existem diversas linguagens nas quais podemos desenvolver nosso código fonte. Entre elas estão: C; C++; Java; Javascript; Ruby; Matlab/Octave; **Python**; etc.

Nessa disciplina utilizaremos a linguagem de programação **Python** devido à simplicidade e extensa documentação na internet.

### 1.2.1 Linguagem Python

A linguagem **Python**, utilizada nessa disciplina, é uma das linguagens mais populares e utilizadas do mundo. Além disso, ela é simples de entender e mais ainda de usar. Abaixo, vamos ensinar um jeito fácil de começar a usar a linguagem.

**Python** é uma linguagem muito ampla e flexível, utilizaremos neste curto um conjunto limitado de seus recursos, com o intuito de aplicar e fixar o conhecimento de estruturas e conceitos de programação de computadores em nível introdutório, compreendendo o conteúdo programático da disciplina.

Escrever um código em **Python** é semelhante a escrever qualquer lista ou sequência de instruções. Contudo, um computador não consegue executar qualquer tipo de instrução. As instruções aceitas são lógicas, aritméticas ou de controle. A seguir vemos um exemplo possível de programa em **Python**, feito para elevar um número ao quadrado:

```
1 # Entrada
2 x = float(input('Digite um número para elevar ao quadrado: '))
3 # Processamento
4 y = x * x
5 # Saída
6 print(f'O resultado do processamento é {y}')
```

No código acima, as linhas indicadas por # são comentários. Isso indica ao computador que tudo que estiver escrito nessas linhas, a partir de #, não deve ser interpretado como instruções a serem executadas. As demais linhas são maneiras de dar instruções ao computador usando a linguagem **Python**. Durante o decorrer da disciplina, aprenderemos a função de cada uma delas.

É importante **praticar** e **aprimorar** os seus conhecimentos na ferramenta de programação escolhida. Utilizaremos o **Thonny**, uma plataforma simples de aprendizado. Está disponível para Mac, Linux e Windows, e é super simples de instalar. Basta baixar no link <https://thonny.org/> e executar. No linux, é só seguir a instrução para instalação via linha de comando do terminal.

## 1.3 Programação Estruturada

Conhecendo o conceito de programas de computador, agora é importante saber também o que significa **Programação Estruturada**. Conforme dito, um programa de computador executa uma sequência de instruções para realizar uma tarefa. Na programação estruturada, instruções podem ser classificadas em três tipos de estrutura: **Sequência**; **Decisão**; **Iteração**.

A seguir apresentamos estas três estruturas em fluxogramas. Os fluxogramas são uma forma de representar um programa de forma mais “alto nível” para entendermos melhor a lógica dos programas implementados. Não se preocupe se você ainda não entender tão bem essas três estruturas. Elas serão trabalhadas novamente e com mais detalhes ao longo da disciplina.

### 1.3.1 Sequência

Como o nome sugere, essa estrutura é composta por uma **sequência** enfileirada de instruções que são executadas uma após a outra, como demonstrado na Figura 1. Com esse tipo de estrutura, podemos representar a visão geral do problema que desejamos resolver.

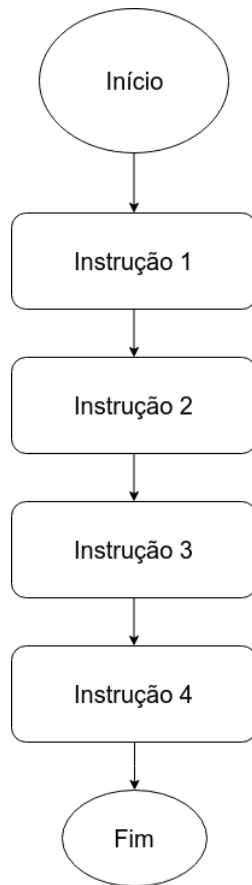


Figura 1: Estrutura do tipo **sequência**.

### 1.3.2 Decisão

Estruturas de **decisão** são utilizadas para realizar um controle e efetuar ações de acordo com uma determinada condição, como demonstrado na Figura 2.

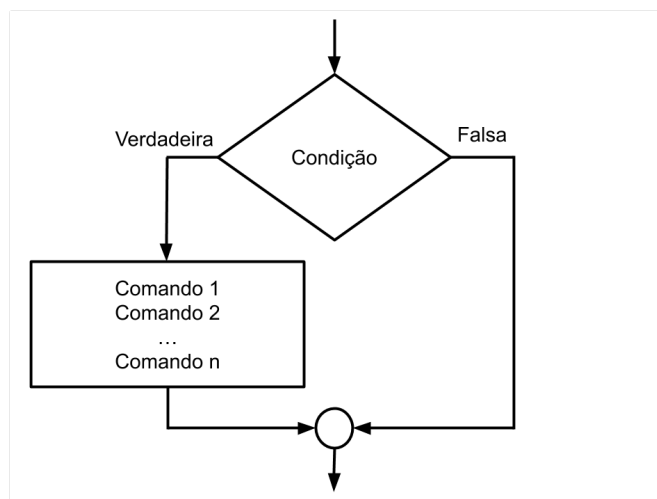


Figura 2: Estrutura do tipo **Decisão**.

### 1.3.3 Iteração

Estruturas de **iteração** (**repetição** ou **laço**) são utilizadas para realizar uma tarefa repetitiva de forma que alguma condição seja cumprida, como demonstrado na Figura 3.

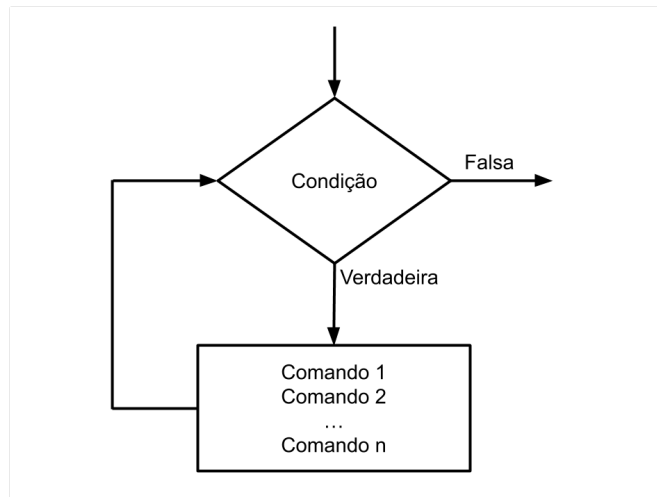


Figura 3: Estrutura do tipo **Iteração**.

## 1.4 Algoritmos e Fluxogramas

Para criar um programa de computador, precisamos de uma sequência de instruções claras e sem ambiguidade, usando os conceitos apresentados na Seção 1.3 – Programação Estruturada. O conjunto dessas instruções e como elas são estruturadas é o que chamamos de **Algoritmo**. Utilizamos algoritmos para resolver os mais diversos problemas do dia-a-dia, da nossa vida acadêmica ou profissional.

Para organização, dividiremos os algoritmos em **3 etapas**:

1. Entrada
2. Processamento
3. Saída

Para representar um algoritmo, uma ferramenta interessante são os **fluxogramas**, da mesma maneira que representamos anteriormente as estruturas de *sequência*, *decisão* e *iteração*. Eles representam cada um dos passos realizados por um algoritmo em caixas. O caminho tomado pelo algoritmo é representado por setas interligando as caixas. Esse tipo de representação permite criar visualizações amigáveis e fáceis de entender. Vejamos um exemplo a seguir.

### 1.4.1 Exemplo 1

Suponha que na aula de física você aprendeu a converter uma temperatura de **Celsius** para **Fahrenheit**. A fórmula usada é:

$$T_F = (T_C \cdot \frac{9}{5}) + 32$$

Em uma sequência sistemática de passos, o raciocínio pode ser o seguinte:

1. Entrada:
  - 1.1. Ler um valor de temperatura em Celsius;
2. Processamento
  - 2.1. Multiplicar o valor por 9 e dividir por 5;
  - 2.2. Somar 32 ao valor encontrado;
3. Saída:
  - 3.1. Imprimir o valor da saída.

Vamos agora organizar esse programa em um fluxograma, para visualizar os passos que o programa vai tomar. Observe na Figura 4, que o fluxograma apenas organiza os passos antes representados por uma lista textual.

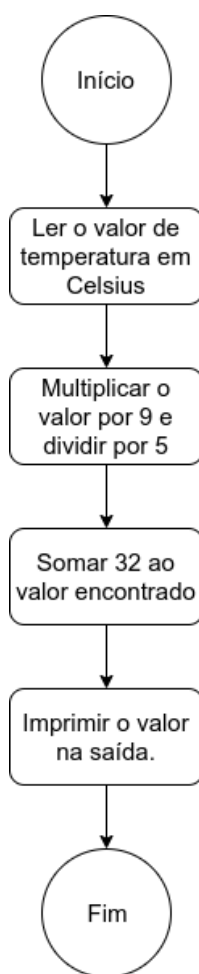


Figura 4: Fluxograma para o Exemplo 1.

O fluxograma mostra a sequência de passos a ser tomada para atingir o objetivo final. Abaixo, podemos ver a aplicação e o funcionamento disso em um programa em **Python**. Tente utilizar o **Thonny** para criar este programa, salvando-o em um arquivo e em seguida executando o código para ver o que acontece.

*Solução em Python para o exemplo 1:*

```
1 # Entrada
2 C = float(input('Digite a temperatura em Celsius: '))
3 # Processamento
4 F = C * 9 / 5
5 F = F + 32
6 #Saída
7 print(f'A temperatura em Fahrenheit é {F:.1f}')
```

Observe que para este exemplo foram apresentadas 3 representações para se resolver algoritmicamente o problema. Primeiro de forma textual, depois em um fluxograma e, por fim, em um programa em **Python**. Todas estas representações estão corretas e resolvem o problema, porém, quando utilizamos uma linguagem de programação é possível executar o programa e ver o resultado final. Nesta disciplina o foco principal será o programa em linguagem **Python**, mas você pode gerar representações

intermediárias, ou em mais “alto nível”, para entender melhor como resolver o problema e somente depois implementar seu programa na linguagem **Python**.

A seguir alguns exercícios propostos, para serem resolvidos através de fluxogramas.

## 1.5 Exercícios Propostos

### Exercício 1

Dada a equação que mostra a conversão de Fahrenheit para Celsius:

$$T_C = (T_F - 32) \cdot \frac{5}{9}$$

Crie um fluxograma que indique as etapas de um algoritmo que representa essa transformação. Veja a sequência na lista abaixo:

1. Ler um valor em Fahrenheit;
2. Subtrair 32 no valor;
3. Multiplicar por 5 e dividir por 9;
4. Imprimir o valor de saída;

### Exercício 2

Faça a lista de instruções e o fluxograma para calcular a associação de dois resistores em série. As entradas são os valores dos resistores em ohms.

Fórmula:

$$R_e = R_1 + R_2$$

### Exercício 3

Faça a lista de instruções e o fluxograma para calcular a associação de dois resistores em paralelo. As entradas são os valores dos resistores em ohms.

Fórmula:

$$R_e = \frac{R_1 \cdot R_2}{R_1 + R_2}$$

### Exercício 4

Apresente o fluxograma para calcular a aceleração de um corpo sendo puxado por uma mola, considerando como entrada a constante elástica da mola ( $k$ ), a massa do corpo ( $m$ ) e a extensão a partir da posição inicial ( $x$ ).

Fórmulas:

$$F = k \cdot x$$

$$a = \frac{F}{m}$$



### Exercício 5

Dado um objeto em queda livre, calcular a velocidade em um instante de tempo e a distância total percorrida desde o início da queda, tendo como a entrada do usuário o instante de tempo  $t$ .

Fórmulas:

$$v = g \cdot t$$

$$s = \frac{g \cdot t^2}{2}$$

$$g = 9,807$$

## 1.6 Solução dos Exercícios Propostos

### Exercício 1

Dada a equação que mostra a conversão de Fahrenheit para Celsius:

$$T_C = (T_F - 32) \cdot \frac{5}{9}$$

Crie um fluxograma que indique as etapas de um algoritmo que representa essa transformação. Veja a sequência na lista abaixo:

1. Ler um valor em Fahrenheit;
2. Subtrair 32 no valor;
3. Multiplicar por 5 e dividir por 9;
4. Imprimir o valor de saída;

*Solução em fluxograma: Figura 5:*

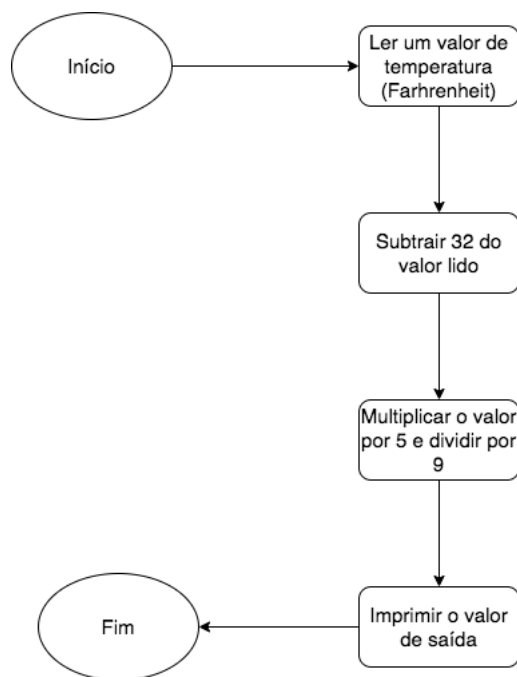


Figura 5: Fluxograma de solução do exercício 1.

### Exercício 2

Faça a lista de instruções e o fluxograma para calcular a associação de dois resistores em série. As entradas são os valores dos resistores em ohms.

Fórmula:

$$R_e = R_1 + R_2$$

Solução em fluxograma: Figura 6:

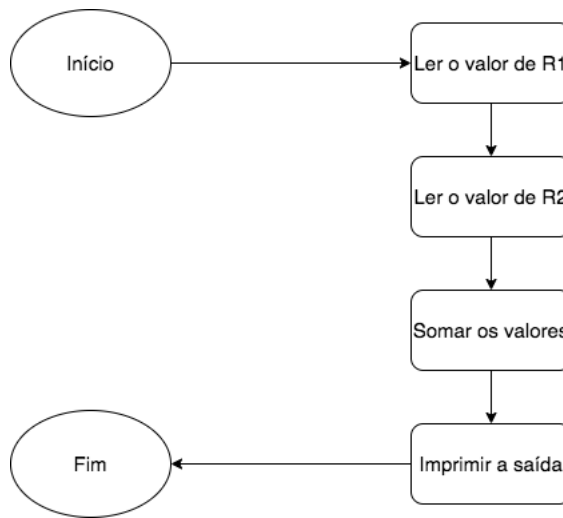


Figura 6: Fluxograma de solução do exercício 2.

### Exercício 3

Faça a lista de instruções e o fluxograma para calcular a associação de dois resistores em paralelo. As entradas são os valores dos resistores em ohms.

Fórmula:

$$R_e = \frac{R_1 \cdot R_2}{R_1 + R_2}$$

Solução em fluxograma: Figura 7:

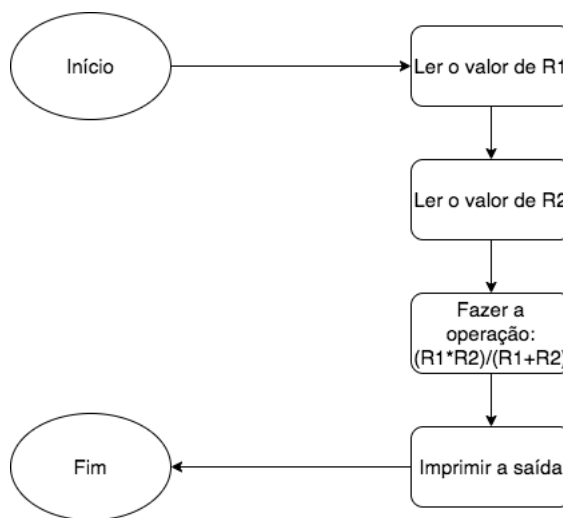


Figura 7: Fluxograma de solução do exercício 3.

#### Exercício 4

Apresente o fluxograma para calcular a aceleração de um corpo sendo puxado por uma mola, considerando como entrada a constante elástica da mola ( $k$ ), a massa do corpo ( $m$ ) e a extensão a partir da posição inicial ( $x$ ).

Fórmulas:

$$F = k \cdot x$$

$$a = \frac{F}{m}$$

Solução em fluxograma: Figura 8:

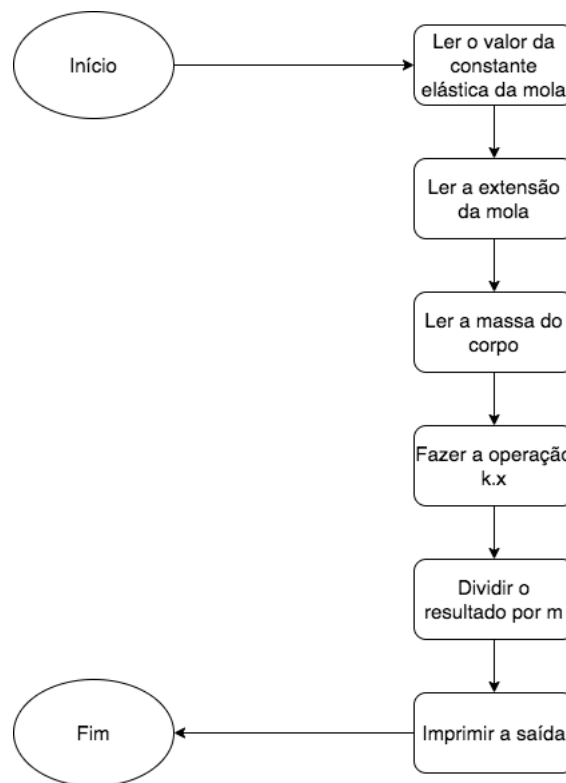


Figura 8: Fluxograma de solução do exercício 4.

#### Exercício 5

Dado um objeto em queda livre, calcular a velocidade em um instante de tempo e a distância total percorrida desde o início da queda, tendo como a entrada do usuário o instante de tempo  $t$ .

Fórmulas:

$$v = g \cdot t$$

$$s = \frac{g \cdot t^2}{2}$$

$$g = 9,807$$

Solução em fluxograma: Figura 9:

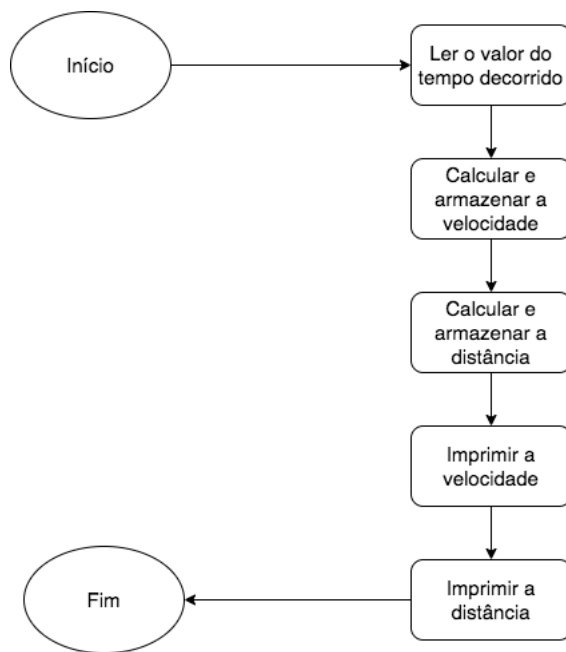


Figura 9: Fluxograma de solução do exercício 5.