

# Representação de algoritmos

## Pacote `algorithm2e`

► **Definição algoritmo:** é um conjunto de instruções para obter uma solução para um determinado problema.

► **Representação de um algoritmo:** é uma forma genérica de escrever um algoritmo, utilizando uma **linguagem simples** sem necessidade de conhecer a sintaxe de **nenhuma linguagem de programação**. Uma dessas representações é conhecida como pseudo-código.

Para escrever um algoritmo no Rmarkdown vamos utilizar:

- Pacote do latex `\usepackage[portuguese,ruled]{algorithm2e}`, em que
  - `portuguese` altera `algorithm` para `algoritmo`;
  - `ruled` acrescenta as linhas horizontais.
- Arquivos
  - `algorithm2e.sty`;
  - `ifoddpage.sty`;
  - `relsize.sty`.

Os arquivos são necessários para gerar os algoritmos. Os mesmos vão estar junto com o `template`.

A seguir, temos dois exemplos de algoritmo.

---

### **Algoritmo 1:** Exemplo de como escrever algoritmo

---

**Entrada:**  $n$  - tamanho da amostra

$y$  - vetor de zeros

```
while condição do  
  ação  
  if condição then  
    | ação se a condição for satisfeita  
  else  
    | ação se caso contrário  
  end  
end  
for  $i=1$  até  $n$  do  
  | ação  
end
```

---

---

**Algoritmo 2:** Exemplo de como escrever algoritmo

---

```
função soma ( $x$ )  
 $n$  - comprimento do vetor  $x$   
 $i = 1$   
 $soma = 0$   
while  $i \leq n$  do  
   $soma = soma + x_i$   
   $i = i + 1$   
end  
retorna  $soma$ 
```

---

As funções para construir um algoritmo no Rmarkdown:

- Bloco para construir um algoritmo

```
\begin{algorithm}  
  algoritmo  
\end{algorithm}
```

- Posicionamento no texto

```
\begin{algorithm}[H]  
  algoritmo  
\end{algorithm}
```

A letra  $H$  entre colchetes indica que o algoritmo deve ficar na ordem em que foi escrito no texto. A ausência, irá ajustar o algoritmo de acordo com o espaço do texto.

- Título do algoritmo

```
\caption{Título}
```

- Control de fluxo: condicional if e else

```
\eIf{condição}{  
  ação se a condição for satisfeita  
}{  
  ação se caso contrário  
}
```

- Controle de fluxo: *loop for*

```
\For{variável=1 até n}{  
    ação  
}
```

- Controle de fluxo: *loop while*

```
\While{condição}{  
    ação  
}
```

- Função

```
\underline{função soma} (argumento)  
    corpo da função  
    retorna saída
```

- Declaração genérica de objetos.

```
\SetKwInOut{Entrada}{Entrada}  
\Entrada{variável - o que é}
```

Não é necessário utilizar **Entrada** quando for atribuído um valor fixo ao objeto.

- Espaçamento entre linhas

```
\SetAlgoLined
```

- Quando necessário, para mudar de linha utilizar: \\