



Sumário

1	Apresentação	1
2	Algoritmos	1
2.1	L ^A T _E X	2
2.2	Arquivos texto	8
2.3	Outros editores	8
A	O arquivo algcap.sty	9

Lista de Algoritmos

1	Estruturas essenciais do algoritmo.	2
2	Cálculo da média de dois valores reais.	3
3	Verificação de aprovação de aluno dadas duas notas.	4
4	Exemplo da estrutura if .	4
5	Exemplo da estrutura if com else if .	5
6	Exemplo da estrutura switch .	5
7	Exemplo da estrutura while .	5
8	Exemplo da estrutura repeat .	5
9	Exemplo da estrutura for e suas variantes.	6
10	Exemplo de função, procedimento e chamadas.	6
11	Exemplo genérico, com aninhamentos.	7
12	Verificação de aprovação de aluno dadas duas notas.	8

1 Apresentação

Este documento se propõe a apresentar algumas diretrizes gerais para a escrita de algoritmos para a disciplina 25070 – Construção de Algoritmos e Programação.

2 Algoritmos

Todos os algoritmos para a disciplina devem seguir um padrão formal. Esse padrão determina o comprometimento da escrita com os seguintes pontos:

- Especificação completa;
- Uso correto da estruturação do código;
- Atenção à organização visual (indentação);
- Uso adequado das estruturas de controle e demais especificações;
- Ausência de numeração de linhas.

A **especificação é completa** quando são identificadas corretamente as entradas e saídas, usando-se a formalização **Input** e **Output**. O Algoritmo 1 ilustra a estrutura básica de um algoritmo.

*Jander Moreira – Universidade Federal de São Carlos – Departamento de Computação – Rodovia Washington Luis, km 235 – 13565-905 - São Carlos/SP – Brasil – jander@dc.ufscar.br

Algoritmo 1 Estruturas essenciais do algoritmo.

Input: especificação das entradas**Output:** especificação das saídas*sequência de comandos*

Tabela I: Estruturas de controle usadas nos algoritmos.

Início	Término	Comentário
if	end if	Execução condicional; inclui elsif e else
switch	end switch	Condicional de seleção; inclui blocos case e otherwise
while	end while	Repetição com teste no início (superior)
repeat	until	Repetição com teste no final (inferior)
for	end for	Repetição determinística
for all	end for	Variante do for
for each	end for	Variante do for
procedure	end procedure	Procedimento
function	end function	Função; inclui return

A **estruturação do código** é refletida pelo uso das estruturas de controle de fluxo (como **if/end if**, **repeat/until** e **for/end for**, por exemplo), bem como das sub-rotinas (**procedure** e **function**).

A clareza e a compreensão dos comandos de um algoritmo estão diretamente associadas à sua **organização visual**. Em outras palavras, a indentação é essencial para a qualidade do código escrito. Desta forma, seguem-se algumas regras básicas:

- Todos os comandos de mesmo nível se iniciam na mesma coluna;
- Comandos internos às estruturas de controle de fluxo devem ser indentados mais à direita;
- Comandos ou sentenças longas podem ser quebradas em várias linhas, desde que as linhas adicionais também sejam indentadas em um nível;
- O início e o fim de uma estrutura devem estar sempre alinhados;
- Linhas em branco devem ser usadas para separar blocos de comandos correlatos; porém deve-se evitar seu excesso;
- Comentários devem ser empregados sempre que puderem trazer maior clareza à compreensão do código.

As **estruturas de controle de fluxo** são os elementos que *estruturam* o algoritmo. Instruções referenciando “o comando acima” ou “os próximos três comandos” não fazem sentido na estruturação (exemplo, “volte para o comando acima”). O elenco de estruturas de controle está representado na Tabela I.

Adicionalmente, as especificações de entrada e saída de dados podem ser expressas por **read** e **write**, respectivamente. A atribuição usa o símbolo \leftarrow , uma vez que $=$ é usado para comparações de igualdade. Na seção 2.2 são apresentadas as alternativas de símbolos quando se usa um editor que não tenha disponibilidade de símbolos “gráficos” (como \leftarrow , \leq , \neq , por exemplo).

Os algoritmos não devem ter **linhas numeradas**. Este recurso deve ser usado exclusivamente quando um algoritmo for apresentado e forem necessárias referências específicas às linhas. Um exemplo seria o texto: “... no **while** da linha 17...”, sendo que há várias estruturas **while** similares no algoritmo.

As seções seguintes apresentam uma orientação geral para o uso de algoritmos usando-se o \LaTeX , quando apenas se dispõe de um editor de texto simples (sem formatação) e para o caso de editores com formatação, como Microsoft Word e Libre Office Write.

2.1 \LaTeX

A estruturação de algoritmos no \LaTeX é uma forma prática e com resultado de alta qualidade. A formatação é dada pelo estilo **algcap**, que é uma adaptação simples do estilo **algpseudocode** do pacote **algorithmix**. O apêndice A apresenta o arquivo **algcap.sty**.

O código deve ser especificado em um ambiente **algorithmic**, provido pelo estilo. A indentação é tratada automaticamente pelo ambiente.

As entradas são especificadas pelo comando $\text{\textbackslash Input}$ e as saídas por $\text{\textbackslash Output}$.

Cada linha de comando deve ser precedida pelo comando $\text{\textbackslash State}$. Há uma versão $\text{\textbackslash StateLine}\{\}$ que gerencia comandos longos, com mais de uma linha. O comando $\text{\textbackslash Statex}$ produz uma linha em branco, a qual não é considerada na numeração de linhas.

Os comandos de entrada e saída são obtidos por $\text{\textbackslash Read}$ e $\text{\textbackslash Write}$, respectivamente.

Os comandos `\If` e `\EndIf` definem a estrutura condicional. Internamente, podem ser usados os comandos `\ElsIf` e `\Else`. Não existe o comando `\Then`.

A seleção múltipla do **switch** usa o par `\Switch` e `\EndSwitch`. Internamente, cada cláusula **case** é estruturada pelos comandos `\Case` e `\EndCase`, sendo o caso não coincidência dado pelos comandos `\Otherwise` e `\EndOtherwise`. Uma macro para a especificação de intervalos é também fornecida: `\Interval{início}{fim}`.

As repetições com teste superior usam `\While` e `\EndWhile` e as com teste inferior são obtidas pelo par `\Repeat` e `\Until`.

Repetições com **for** são iniciadas por `\For` ou, alternativamente, `\ForAll` ou `\ForEach`. Todas as variações são terminadas com `\EndFor`. As macros `\To`, `\DownTo` e `\Step` também estão disponíveis.

Os comentários usam o comando `\Comment{}`, produzindo a formatação à direita. A variação `\CommentLocal{}` posiciona o comentário no local especificado.

Exemplos de algoritmos e seus respectivos códigos em \LaTeX são providos nos Algoritmos 2 a 11.

Algoritmo 2 Cálculo da média de dois valores reais.

```
\Input Dois valores reais quaisquer dados pelo usuário
\Output Um valor real equivalente à média aritmética dos valores fornecidos
\Statex
\State \Read $v_1, v_2$
\State $m \leftarrow (v_1 + v_2)/2$ \Comment{cálculo}
\State \Write $m$
```

Input: Dois valores reais quaisquer dados pelo usuário

Output: Um valor real equivalente à média aritmética dos valores fornecidos

```
read  $v_1, v_2$ 
 $m \leftarrow (v_1 + v_2)/2$ 
write  $m$ 
```

▷ *cálculo*

Algoritmo 3 Verificação de aprovação de aluno dadas duas notas.

```

\Input Duas notas $n_1, n_2 \in \mathbb{R}$, ambas no intervalo $[0, 10]$
\Output Uma mensagem $$$, conforme a média seja maior ou igual a 6, com $$ \in
    \{\mathit{aprovado}, \mathit{reprovado}\}$
\Statex
\State\CommentLocal{Obtenção das notas e cálculo da média}
\State \Read $n_1, n_2$
\State $m \gets \dfrac{n_1 + n_2}{2}$
\Statex
\State\CommentLocal{Resultado}
\If{$m \geq 6$}
    \State $s \gets \mathit{aprovado}$
\Else
    \State $s \gets \mathit{reprovado}$
\EndIf
\State \Write $$$

```

Input: Duas notas $n_1, n_2 \in \mathbb{R}$, ambas no intervalo $[0, 10]$

Output: Uma mensagem s , conforme a média seja maior ou igual a 6, com $s \in \{\text{aprovado}, \text{reprovado}\}$

▷ Obtenção das notas e cálculo da média

$$m \leftarrow \frac{n_1 + n_2}{2}$$

▷ *Resultado*

```

if  $m \geq 6$  then
     $s \leftarrow \text{aprovado}$ 

```

```

else
     $s \leftarrow reprovado$ 

```

end if

write s

Algoritmo 4 Exemplo da estrutura if.

```
\If{$a$}
  \State Faça A
\If{$b$}
  \State Faça B
\Else
  \State Faça C
\EndIf
\State Faça D
\Else
  \State Faça E
\EndIf
```

```

if a then
    Faça A
if b then
    Faça B
else
    Faça C
end if
    Faça D
else
    Faça E
end if

```

Algoritmo 5 Exemplo da estrutura **if** com **else if**.

```
\If{$a$}
  \State Faça A
\ElseIf{$b$}
  \State Faça B
\ElseIf{$c$}
  \State Faça C
\ElseIf{$d$}
  \State Faça D
\ElseIf{$e$}
  \State Faça E
\Else
  \State Faça F
\EndIf
```

```
if  $a$  then
  Faça A
else if  $b$  then
  Faça B
else if  $c$  then
  Faça C
else if  $d$  then
  Faça D
else if  $e$  then
  Faça E
else
  Faça F
end if
```

Algoritmo 6 Exemplo da estrutura **switch**.

```
\Switch{n}
  \Case{1}\Comment{$n = 1$?}
    \State Faça A
    \State Faça B
  \EndCase
  \Case{2, 4, 6, 8}
    \Comment{$n$ \in \{2, 4, 6, 8\}$?}
    \State Faça C
  \EndCase
  \Case{3, 5, 7}
    \Comment{$n$ \in \{3, 5, 7\}$?}
    \State Faça D
    \State Faça E
    \State Faça F
  \EndCase
  \Case {\Interval{9}{100}}
    \Comment{$9 \leq n \leq 100$?}
    \State Faça G
    \State Faça H
  \EndCase
  \Otherwise\Comment{n.d.a.?}
    \State Faça I \CommentLocal{default}
  \EndOtherwise
\EndSwitch
```

```
switch  $n$  of
  case 1 do
    Faça A
    Faça B
  end case
  case 2, 4, 6, 8 do
    Faça C
  end case
  case 3, 5, 7 do
    Faça D
    Faça E
    Faça F
  end case
  case 9..100 do
    Faça G
    Faça H
  end case
  otherwise do
    Faça I  $\triangleright$  default
  end otherwise
end switch
```

$\triangleright n = 1?$
 $\triangleright n \in \{2, 4, 6, 8\}?$
 $\triangleright n \in \{3, 5, 7\}?$
 $\triangleright 9 \leq n \leq 100?$
 $\triangleright n.d.a.?$

Algoritmo 7 Exemplo da estrutura **while**.

```
\While{$n > 0$}
  \State  $n$  \gets  $n - 1$ 
  \State \Write  $n$ 
\EndWhile
```

```
while  $n > 0$  do
   $n \leftarrow n - 1$ 
  write  $n$ 
end while
```

Algoritmo 8 Exemplo da estrutura **repeat**.

```
\Repeat
  \State  $n$  \gets  $n - 1$ 
\Until{$n = 0$}
```

```
repeat
   $n \leftarrow n - 1$ 
until  $n = 0$ 
```

Algoritmo 9 Exemplo da estrutura **for** e suas variantes.

```
\State\CommentLocal{Formato padrão}
\For{$i \gets 0$ \To $k - 1$ \Step $s$}
  \State\Write $i$
\EndFor
\Statex
\State\CommentLocal{Formato alternativo: for all}
\ForAll{caracteres do nome da cidade}
  \State Converta o caractere para maiúsculo
\EndFor
\Statex
\State\CommentLocal{Formato alternatio: for each}
\ForEach{$\textit{mês}$ \in \{Jan, Mar, Out, Nov, Dec\}}
  \State\Call{Processa}{$\textit{mês}$}
\EndFor
```

▷ *Formato padrão*

for $i \leftarrow 0$ **to** $k - 1$ **step** s **do**
 write i
end for

▷ *Formato alternativo: for all*

for all caracteres do nome da cidade **do**
 Converta o caractere para maiúsculo
end for

▷ *Formato alternatio: for each*

for each $mês \in \{Jan, Mar, Out, Nov, Dec\}$ **do**
 PROCESSA($mês$)
end for

Algoritmo 10 Exemplo de função, procedimento e chamadas.

```
\Procedure{Apresente}{$c, f$}
  \State Apresente a mensagem: $c$ é
    igual a $f$
\EndProcedure
\Statex
\Function{CelsiusParaFahrenheit}{$c$}
  \State $t \gets \frac{9}{5}c + 32$
  \State \Return $t$
\EndFunction
\Statex
\For{$c \gets 0$ \To $100$}
  \State $f \gets$
    \Call{CelsiusParaFahrenheit}{$c$}
  \State \Call{Apresente}{$c, f$}
\EndFor
```

procedure APRESENTE(c, f)
 Apresente a mensagem: c é igual a f
end procedure

function CELSIUSPARAFAHRENHEIT(c)
 $t \leftarrow \frac{9}{5}c + 32$
 return t
end function

for $c \leftarrow 0$ **to** 100 **do**
 $f \leftarrow$ CELSIUSPARAFAHRENHEIT(c)
 APRESENTE(c, f)
end for

Algoritmo 11 Exemplo genérico, com aninhamentos.

```
\State Faça A
\While{$c$ não ocorre}
  \State \Read $n$
  \If{$n \leq 0$}
    \State\Call{Erro}{$n$}
  \Else
    \State Faça B
    \Switch{$n$}
      \Case{1}
        \State Faça C
      \EndCase
      \Case{2}
        \State Faça D
      \EndCase
      \Case{3}
        \State Faça E
      \EndCase
      \Case{\Interval{4}{10}}
        \State Faça F
      \EndCase
    \EndSwitch
  \Repeat
    \State Faça G
    \State $n$ \gets \Call{ComputeNovo}{$n+1$}
  \Until{$n \neq -1$ e $n < m$}
\EndIf
\State Avalie a condição $c$
\EndWhile
```

```
Faça A
while c não ocorre do
  read n
  if  $n \leq 0$  then
    ERRO( $n$ )
  else
    Faça B
    switch n of
      case 1 do
        Faça C
      end case
      case 2 do
        Faça D
      end case
      case 3 do
        Faça E
      end case
      case 4..10 do
        Faça F
      end case
    end switch
  repeat
    Faça G
     $n \leftarrow \text{COMPUTENOVO}(n + 1)$ 
  until  $n \neq -1$  e  $n < m$ 
  end if
  Avalie a condição c
end while
```

2.2 Arquivos texto

Uma das melhores formas de se ter um algoritmo visualmente bem estruturado é pelo uso de um editor de texto simples, ou seja, sem formatação. Em geral as fontes são mono-espaçadas, o que permite um ajuste correto das indentações.

A dificuldade é a ausência de alguns símbolos, como \leftarrow e \neq , por exemplo. Neste caso, é possível fazer substituições, como as da Tabela II. Essas substituições são de emprego amplo na comunidade.

Tabela II: Símbolos no modo texto para substituição dos símbolos gráficos.

Gráfico	Texto
\leftarrow	<-
\triangleright (comentário)	//
\neq	<> ou !=
\leq	<=
\geq	>=
Expressões	Texto
\sqrt{x}	raiz(x)
x^5	x**5 ou x^5 ou pot(x, 5)
Gregos	Texto
α	alfa
τ	tau
Δ	delta
Constantes	Texto
π	pi ou 3,1415

O Algoritmo 12 é uma versão texto do Algoritmo 3.

Algoritmo 12 Verificação de aprovação de aluno dadas duas notas. Confrontar esta versão com a do Algoritmo 3.

Input: Duas notas reais n1 e n2, ambas no intervalo [0, 10]
Output: Uma mensagem s em {aprovado, reprovado}, conforme a
médica das notas seja maior que 6

```
// Obtenção das notas e cálculo da média
read n1, n2
m <- (n1 + n2)/2

// Resultado
if m >= 6 then
    s <- aprovado
else
    s <- reprovado
endif
write s
```

2.3 Outros editores

Em outros editores de texto, é possível ter os símbolos especiais, bem como letras gregas e outros recursos. Para este editores, sugere-se uso de fonte mono-espaçada, como Courier.

Em especial, é preciso atenção para as situações em que o comprimento da linha é grande e a quebra automática desobedece os padrões de indentação.

A marcação de palavras reservadas (**if**, **for**, **function**) em negrito é totalmente desnecessária.

Apêndice

A O arquivo algcap.sty

O arquivo **algcap.sty** está disponível em <http://cap.dc.ufscar.br> e está transcrito na sequência.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Algoritmos CAP:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Jander Moreira, 2018
%
\usepackage{algorithm} % algoritmos numerados
\usepackage{algpseudocode} % formatador do pseudocódigo
\usepackage{icomma} % ajusta vírgula como separador decimal

\algnewcommand\algorithmicinput{\textbf{Input:}}
\algnewcommand\Input{\item[\algorithmicinput]}
\algnewcommand\algorithmicoutput{\textbf{Output:}}
\algnewcommand\Output{\item[\algorithmicoutput]}

%switch
\algblockdefx[NAME]{Switch}{EndSwitch}[1]%
    {\textbf{switch} #1 \textbf{of}}
    {\textbf{end~switch}}
%case
\algblockdefx[NAME]{Case}{EndCase}[1]%
    {\textbf{case} $#1$ \textbf{do}}%
    {\textbf{end~case}}
%otherwise
\algblockdefx[NAME]{Otherwise}{EndOtherwise}%
    {\textbf{otherwise do}}%
    {\textbf{end~otherwise}}
%foreach
\algblockdefx[NAME]{ForEach}{EndFor}[1]%
    {\textbf{for~each} #1 \textbf{do}}%
    {\textbf{end~for}}
\newcommand{\Interval}[3][ ]{
    \ensuremath{#2\mathcal{\ldotp\ldotp}#3}%
    \def\temp{#1}%
    \ifx\temp\empty{}%
    \else{\mathcal{\ldotp\ldotp}#1}%
    \fi
}

\newcommand{\CommentSymbolLeft}{$\triangleright$} % ou \texttt{//} ???
\newcommand{\CommentSymbolRight}{$\triangleleft$}
\algnewcommand{\CommentIn}[1]{\CommentSymbolLeft\textsl{#1}\CommentSymbolRight}
\algnewcommand{\CommentLocal}[1]{\CommentSymbolLeft~\textsl{#1}}
\algrenewcommand{\algorithmiccomment}[1]{\hfill\CommentLocal{#1}}
\algrenewcommand{\alglinenumber}[1]%
    {\hspace{-1em}\color{black!35}\scriptsize#1\blacktriangleright}

\algnewcommand{\Read}{\textbf{read}~}
\algnewcommand{\Write}{\textbf{write}~}
\algnewcommand{\To}{\textbf{to}~}
\algnewcommand{\DownTo}{\textbf{downto}~}
\algnewcommand{\Step}{\textbf{step}~}

\usepackage{tcolorbox} % to save verbatim
\usepackage{fancyvrb} % to load verbatim preserving tabs
\newenvironment{definecode}[1]
    {\begingroup\tcbverbatimwrite{\jobname_code_#1.tmp}}
    {\endtcverbatimwrite\endgroup}
```

```

\newcommand{\algcode}[1]{\input{\jobname_code_#1.tmp}}
\newcommand{\sourcecode}[1]{\small\VerbatimInput[tabsize=4]{\jobname_code_#1.tmp}}

\usepackage{listings}
\lstset{
  basicstyle = \ttfamily,
  tabsize = 4,
  literate=
    {á}{\`a}}1 {é}{\`e}}1 {í}{\`i}}1 {ó}{\`o}}1 {ú}{\`u}}1
    {Á}{\`A}}1 {É}{\`E}}1 {Í}{\`I}}1 {Ó}{\`O}}1 {Ú}{\`U}}1
    {à}{\`a}}1 {è}{\`e}}1 {ì}{\`i}}1 {ò}{\`o}}1 {û}{\`u}}1
    {À}{\`A}}1 {È}{\`E}}1 {Î}{\`I}}1 {Ò}{\`O}}1 {Û}{\`U}}1
    {ä}{\`a}}1 {ë}{\`e}}1 {ï}{\`i}}1 {ö}{\`o}}1 {ü}{\`u}}1
    {ã}{\`a}}1 {õ}{\`o}}1
    {Ã}{\`A}}1 {Õ}{\`O}}1
    {Ä}{\`A}}1 {Ë}{\`E}}1 {Ï}{\`I}}1 {Ö}{\`O}}1 {Ü}{\`U}}1
    {â}{\`a}}1 {ê}{\`e}}1 {î}{\`i}}1 {ô}{\`o}}1 {û}{\`u}}1
    {Â}{\`A}}1 {Ê}{\`E}}1 {Î}{\`I}}1 {Ô}{\`O}}1 {Û}{\`U}}1
    {ç}{\`c}}1 {Ç}{\`C}}1
    {ø}{\`o}}1 {å}{\`r}}1 {Å}{\`r}}1
    {œ}{\`oe}}1 {Œ}{\`OE}}1 {æ}{\`ae}}1 {Æ}{\`AE}}1 {ß}{\`ss}}1
    {ü}{\`H{u}}1 {Ů}{\`H{U}}1 {ơ}{\`H{o}}1 {Ỗ}{\`H{O}}1
    {£}{\`pounds}}1 {«}{\`guillemotleft}}1
    {»}{\`guillemotright}}1 {ñ}{\`n}}1 {Ñ}{\`N}}1 {¿}{\`?}}1
}

% fim: algoritmos para CAP
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```