

# Algoritmos

Expressões e Funções Matemáticas.

Prof. Dr. Osmar Betazzi Dordal

3

# Expressões Matemáticas

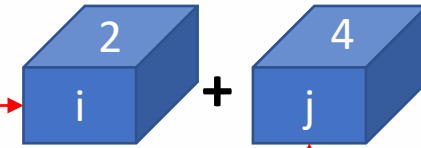
- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;  
2 var  
3   i: integer;  
4   j: integer;  
5   r: real;  
6   m: real;  
7 Begin  
8  
9   i := 2;  
10  j := 4;  
11  
12  r := 20.5;  
13  m := 40.7;  
14  
15  writeln('i+j: ', i+j);  
16  writeln('r+m: ', r+m);  
17  writeln('j-i: ', j-i);  
18  writeln('i-j: ', i-j);  
19  writeln('j/i: ', j/i);  
20  writeln('j*i: ', j*i);  
21  writeln('r-m: ', r-m);  
22  writeln('r/m: ', r/m);  
23  writeln('r/j: ', r/j);  
24  writeln('r*m: ', r*m);  
25  
26 End.
```

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;  
2 var  
3   i: integer;  
4   j: integer;  
5   r: real;  
6   m: real;  
7 Begin  
8  
9   i := 2;  
10  j := 4;  
11  
12  r := 20.5;  
13  m := 40.7;  
14  
15  writeln('i+j: ', i+j);  
16  writeln('r+m: ', r+m);  
17  writeln('j-i: ', j-i);  
18  writeln('i-j: ', i-j);  
19  writeln('j/i: ', j/i);  
20  writeln('j*i: ', j*i);  
21  writeln('r-m: ', r-m);  
22  writeln('r/m: ', r/m);  
23  writeln('r/j: ', r/j);  
24  writeln('r*m: ', r*m);  
25  
26 End.
```

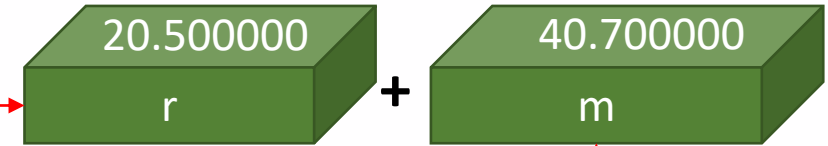


```
15 i+j: 6  
16 r+m: 61.200000  
17 j-i: 2  
18 i-j: -2  
19 j/i: 2.000000  
20 j*i: 8  
21 r-m: -20.200000  
22 r/m: 0.503686  
23 r/j: 5.125000  
24 r*m: 834.350000
```

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;  
2 var  
3   i: integer;  
4   j: integer;  
5   r: real;  
6   m: real;  
7 Begin  
8  
9   i := 2;  
10  j := 4;  
11  
12  r := 20.5;  
13  m := 40.7;  
14  
15  writeln('i+j: ', i+j);  
16  writeln('r+m: ', r+m);  
17  writeln('j-i: ', j-i);  
18  writeln('i-j: ', i-j);  
19  writeln('j/i: ', j/i);  
20  writeln('j*i: ', j*i);  
21  writeln('r-m: ', r-m);  
22  writeln('r/m: ', r/m);  
23  writeln('r/j: ', r/j);  
24  writeln('r*m: ', r*m);  
25  
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```



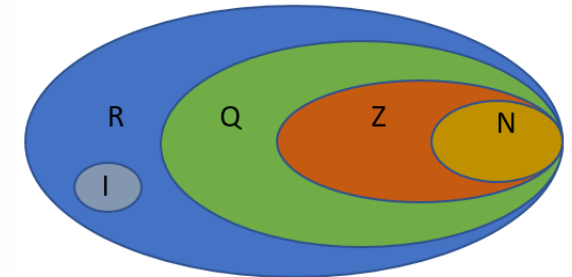
15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```

15 i+j: 6  
16 r+m: 61.200000  
17 j-i: 2  
18 i-j: -2  
19 j/i: 2.000000  
20 j\*i: 8  
21 r-m: -20.200000  
22 r/m: 0.503686  
23 r/j: 5.125000  
24 r\*m: 834.350000






# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000



# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```




15	i+j: 6
16	r+m: 61.200000
17	j-i: 2
18	i-j: -2
19	j/i: 2.000000
20	j*i: 8
21	r-m: -20.200000
22	r/m: 0.503686
23	r/j: 5.125000
24	r*m: 834.350000

# Expressões Matemáticas

- Soma (+)
- Subtração (-)
- Multiplicação (\*)
- Divisão (/)

```
1 Program Expressoes_Matematica;
2 var
3   i: integer;
4   j: integer;
5   r: real;
6   m: real;
7 Begin
8
9   i := 2;
10  j := 4;
11
12  r := 20.5;
13  m := 40.7;
14
15  writeln('i+j: ', i+j);
16  writeln('r+m: ', r+m);
17  writeln('j-i: ', j-i);
18  writeln('i-j: ', i-j);
19  writeln('j/i: ', j/i);
20  writeln('j*i: ', j*i);
21  writeln('r-m: ', r-m);
22  writeln('r/m: ', r/m);
23  writeln('r/j: ', r/j);
24  writeln('r*m: ', r*m);
25
26 End.
```

15 i+j: 6  
16 r+m: 61.200000  
17 j-i: 2  
18 i-j: -2  
19 j/i: 2.000000  
20 j\*i: 8  
21 r-m: -20.200000  
22 r/m: 0.503686  
23 r/j: 5.125000  
24 r\*m: 834.350000



# Funções Matemáticas.

# Funções Matemáticas

- **DIV**: retorna o **quociente** inteiro da divisão;
- **MOD**: retorna o **resto** da divisão;
- **INT(x)**: resulta a parte **inteira** de um número x;
- **FRAC(x)**: resulta a parte **fracionária** de x;
- **ABS(x)**: resulta o valor **absoluto** de x;
  - Não importa se o número é negativo ou positivo, com ABS sempre retorna positivo.
- **SQR(x)**: resulta no **quadrado** de x;
- **SQRT(x)**: resulta na **raiz quadrada** de x (para x real não negativo).

```
1 Program Funcao_Matematica;
2 var
3     h, i, j, k, l: integer;
4     r, m, n, o: real;
5
6 Begin
7     h := -20;
8     i := 15;
9     j := 7;
10    k := 40;
11    l := 4;
12
13    r := 20.5;
14    m := 40.7;
15    n := 100.0;
16    o := 50.0;
17
18    writeln(' (15)mod(7) ... ', (i)mod(j));
19    writeln(' (15)div(7) ... ', (i)div(j));
20    writeln(' (40)mod(4) ... ', (k)mod(l));
21    writeln(' (40)div(4) ... ', (k)div(l));
22    writeln('int(40.7) ... ', int(m));
23    writeln('frac(40.7) ... ', frac(m));
24    writeln('abs(-20) ... ', abs(h));
25    writeln('sqr(4) ... ', sqr(l));
26    writeln('sqrt(4) ... ', sqrt(l));
27
28 End.
```

# Funções Matemáticas

- **DIV**: retorna o **quociente** inteiro da divisão;
- **MOD**: retorna o **resto** da divisão;
- **INT(x)**: resulta a parte **inteira** de um número x;
- **FRAC(x)**: resulta a parte **fracionária** de x;
- **ABS(x)**: resulta o valor **absoluto** de x;
  - Não importa se o número é negativo ou positivo, com ABS sempre retorna positivo.
- **SQR(x)**: resulta no **quadrado** de x;
- **SQRT(x)**: resulta na **raiz quadrada** de x (para x real não negativo).

```
1 Program Funcao_Matematica;
2 var
3   h, i, j, k, l: integer;
4   r, m, n, o: real;
5
6 Begin
7   h := -20;
8   i := 15;
9   j := 7;
10  k := 40;
11  l := 4;
12
13  r := 20.5;
14  m := 40.7;
15  n := 100.0;
16  o := 50.0;
17
18  writeln(' (15)mod(7) ...: ', (i)mod(j));
19  writeln(' (15)div(7) ...: ', (i)div(j));
20  writeln(' (40)mod(4) ...: ', (k)mod(l));
21  writeln(' (40)div(4) ...: ', (k)div(l));
22  writeln('int(40.7) ...: ', int(m));
23  writeln('frac(40.7) ...: ', frac(m));
24  writeln('abs(-20) ...: ', abs(h));
25  writeln('sqr(4) ...: ', sqr(l));
26  writeln('sqrt(4) ...: ', sqrt(l));
27
28 End.
```

18	(15)mod(7) ...: 1
19	(15)div(7) ...: 2
20	(40)mod(4) ...: 0
21	(40)div(4) ...: 10
22	int(40.7) ...: 40
23	frac(40.7) ...: 0.700000
24	abs(-20) ...: 20
25	sqr(4) ...: 16
26	sqrt(4) ...: 2.000000



# Funções Matemáticas

- **DIV**: retorna o **quociente** inteiro da divisão;
- **MOD**: retorna o **resto** da divisão;
- **INT(x)**: resulta a parte **inteira** de um número x;
- **FRAC(x)**: resulta a parte **fracionária** de x;
- **ABS(x)**: resulta o valor **absoluto** de x;
  - Não importa se o número é negativo ou positivo, com ABS sempre retorna positivo.
- **SQR(x)**: resulta no **quadrado** de x;
- **SQRT(x)**: resulta na **raiz quadrada** de x (para x real não negativo).

```
1 Program Funcao_Matematica;
2 var
3   h, i, j, k, l: integer;
4   r, m, n, o: real;
5
6 Begin
7   h := -20;
8   i := 15;
9   j := 7;
10  k := 40;
11  l := 4;
12
13  r := 20.5;
14  m := 40.7;
15  n := 100.0;
16  o := 50.0;
17
18  writeln(' (15)mod(7) ...: ', (i)mod(j));
19  writeln(' (15)div(7) ...: ', (i)div(j));
20  writeln(' (40)mod(4) ...: ', (k)mod(l));
21  writeln(' (40)div(4) ...: ', (k)div(l));
22  writeln('int(40.7) ...: ', int(m));
23  writeln('frac(40.7) ...: ', frac(m));
24  writeln('abs(-20) ...: ', abs(h));
25  writeln('sqr(4) ...: ', sqr(l));
26  writeln('sqrt(4) ...: ', sqrt(l));
27
28 End.
```

15 7  
1 2

18	(15)mod(7) ...: 1
19	(15)div(7) ...: 2
20	(40)mod(4) ...: 0
21	(40)div(4) ...: 10
22	int(40.7) ...: 40
23	frac(40.7) ...: 0.700000
24	abs(-20) ...: 20
25	sqr(4) ...: 16
26	sqrt(4) ...: 2.000000

# Expressões e Funções Matemáticas

```
1 Program Area_circunferencia ;
2 var
3     raio, area: real;
4 Begin
5     writeln('Digite o valor do raio da circunferência: ');
6     readln(raio);
7     area := PI * raio * raio;
8     writeln('A área da circunferência de raio ', raio, ' é: ', area);
9 End.
```

```
Digite o valor do raio da circunferência:
10
A área da circunferência de raio 10.000000 é: 314.159265
```

# Expressões e Funções Matemáticas

```
1 Program Area_circunferencia ;
2 var
3   raio, area: real;
4 Begin
5   writeln('Digite o valor do raio da circunferência: ');
6   readln(raio);
7   area := PI * sqr(raio);
8   writeln('A área da circunferência de raio ', raio:0:1, ' é: ', area:0:2);
9 End.
```

Digite o valor do raio da circunferência:

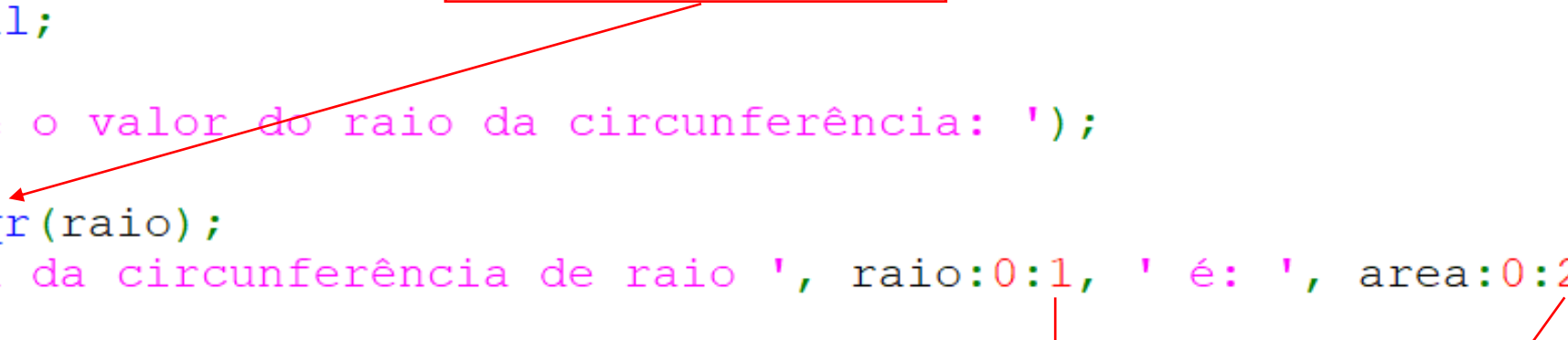
10

A área da circunferência de raio 10.0 é: 314.16

# Expressões e Funções Matemáticas

```
1 Program Area_circunferencia ;
2 var
3   raio, area: real;
4 Begin
5   writeln('Digite o valor do raio da circunferência: ');
6   readln(raio);
7   area := PI * sqr(raio);
8   writeln('A área da circunferência de raio ', raio:0:1, ' é: ', area:0:2);
9 End.
```





Expressão matemática



```
Digite o valor do raio da circunferência:
10
A área da circunferência de raio 10.0 é: 314.16
```

# Cálculos

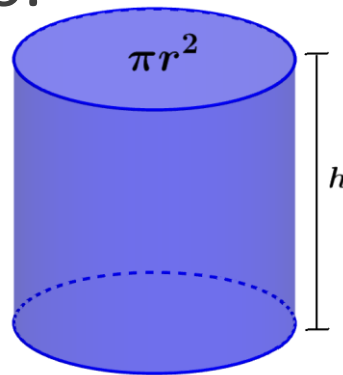
- Fazer o algoritmo que faça o cálculo de quantos litros de água caberá em cada caixa de água da figura ao lado.
- Lembrando que esse cálculo se chama volume do cilindro.

 Vídeo	<b>TBL 303</b> <b>3000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 4,40 Diametro(m) Ø 0,95	<input type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 304</b> <b>3000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 2,60 Diametro(m) Ø 1,27	<input checked="" type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 501</b> <b>5000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 5,40 Diametro(m) Ø 1,11	<input type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 502</b> <b>5000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 1,80 Diametro(m) Ø 1,91	<input type="checkbox"/> Selecionar para orçamento





# Cálculos

- Fazer o algoritmo que faça o cálculo de quantos litros de água caberá em cada caixa de água da figura ao lado.
- Lembrando que esse cálculo se chama volume do cilindro.

OBSERVAÇÃO: o dono da empresa que produz tais produtos irá entregar um produto que tenha a quantidade aproximada e sempre um pouco mais e não menos.



$$V = \pi r^2 \times h$$

 Vídeo	<b>TBL 303</b> <b>3000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 4,40 Diametro(m) Ø 0,95	<input type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 304</b> <b>3000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 2,60 Diametro(m) Ø 1,27	<input checked="" type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 501</b> <b>5000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 5,40 Diametro(m) Ø 1,11	<input type="checkbox"/> Selecionar para orçamento
 Vídeo	<b>TBL 502</b> <b>5000 LITROS</b> TUBULAR ALTO Projeto	Altura(m) 1,80 Diametro(m) Ø 1,91	<input type="checkbox"/> Selecionar para orçamento

# Referências

- FORBELLONE, André L. **Lógica de Programação**. Prentice Hall Brasil, 3ª edição, 2005.
- VELOSO, Paulo; et al. **Estrutura de dados**. Rio de Janeiro: Campus, 4ª edição, 1996.
- LAGES & GUIMARAES. **Algoritmos e Estrutura de dados**. Ed. LTC, 1994.
- FARRER, H. **Algoritmos estruturados**. Rio de Janeiro: Guanabara Koogan, 3ª edição, 1989.
- LUIZ, Jaime. **Estrutura de dados e seus algoritmos**. Editora LTC.
- GUEDES, S. **Lógica de Programação Algorítmica**. Editora Pearson, 2014.
- MANZANO, José Augusto N. G. **Algoritmos lógica para desenvolvimento de programação de computadores**. Ed. 1, São Paulo, Erica 2016.
- MANZANO, José Augusto N. G. **Algoritmos técnicas de programação**. Ed 2, São Paulo, Erica, 2016.