Linguagem e Técnica de Programação 1 – (LTP1)

CAPÍTULO – 3

OPERADORES

Prof. Arnaldo

SUMÁRIO

- Operadores
  - Aritméticos
  - Relacionais
  - Lógicos
    - E / AND
    - OU / OR
    - NÃO / NOT

OPERADORES

- São meios pelo qual se incrementa, decrementa, compara e avalia os dados dentro do computador.

- Tendo as variáveis como base da informação de uma linguagem, elas podem ser modificadas e comparadas com outras por meio dos chamados **operadores**.

- Temos 03 (três) tipos de operadores:
  - ✓ **Aritméticos;**
  - ✓ **Relacionais; e**
  - ✓ **Lógicos**.

Operadores Aritméticos

- São utilizados para realizar operações numéricas com os dados usados pelo programa.
- Existe dois operadores não muito convencionais, porém bastante úteis para resolver diversas situações na construção de um algoritmo: o **mod** e o **div**.

Operação	Símbolo
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	**
Radiciação	rad
Resto da Divisão	mod
Quociente da Divisão	div

Operadores Aritméticos

- Para resolver as operações aritméticas há uma hierarquia a ser seguida:

Prioridade	Operadores
1°	()
2°	** rad
3°	* / div mod
4°	+ -

Nota:
quando duas operações de mesmo nível de prioridade têm de ser avaliadas, a operação mais à esquerda será resolvida primeiro.

Operadores Aritméticos

Observação:
Um ponto importante que deve ser sempre levado em consideração, quando uma expressão for calculada, são os **tipos das variáveis**, porque eles **alteram** radicalmente os **resultados das expressões**. Se uma variável foi declarada como inteiro, a divisão entre inteiros trunca qualquer parte decimal que ocorra.

Operadores Relacionais

- Durante o desenvolvimento de um algoritmo, vamos sempre encontrar situações em que será necessário **comparar informações** para que o programa possa **tomar uma decisão**. Para isso, é necessário **testar uma condição**, que em geral consiste em verificar se uma determinada variável tem valor **verdadeiro** ou **falso**. Essa comparação é por exemplo: se "A é maior que B", ou se "A é igual a15", se "A é diferente de 50" .

Operadores Relacionais

- São utilizados para comparar duas expressões de qualquer tipo. Quando efetuamos uma comparação o resultado é sempre do tipo **lógico** (**booleano**), isto é resulta sempre em um valor **Verdadeiro** (**True**) ou **Falso** (**False**) .

Símbolo	Descrição
=	Igual a
<> ou !=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Operadores Relacionais

- Exemplo: Tendo duas variáveis A = 7 e B = 4, os resultados da expressões seriam:

Condições	Resultado
A = B	**Falso**
A <> B	**Verdadeiro**
A > B	**Verdadeiro**
A < B	**Falso**
A >= B	**Verdadeiro**
A <= B	**Falso**

Lembre-se:
Para estabelecer prioridades no que diz respeito a qual operação executar primeiro, utilize os parênteses.

Dica:
As operações lógicas só podem ser efetuadas com relação a valores do mesmo tipo.

Operadores Lógicos

- Quando existe a necessidade de trabalhar com duas ou mais condições ao mesmo tempo são utilizados os operadores lógicos que **são responsáveis para a formação de novas proposições lógicas compostas** a partir de outras proposições lógicas simples.
- São usados para **comparar mais de uma condição** em uma mesma expressão, ou seja, pode-se fazer mais de uma comparação (teste) ao mesmo tempo, retornando se o resultado da nova proposição é verdade ou falso.

Operador	Operação	Prioridade
não / not	Negação	1ª
e / and	Conjunção	2ª
ou / or	Disjunção	3ª

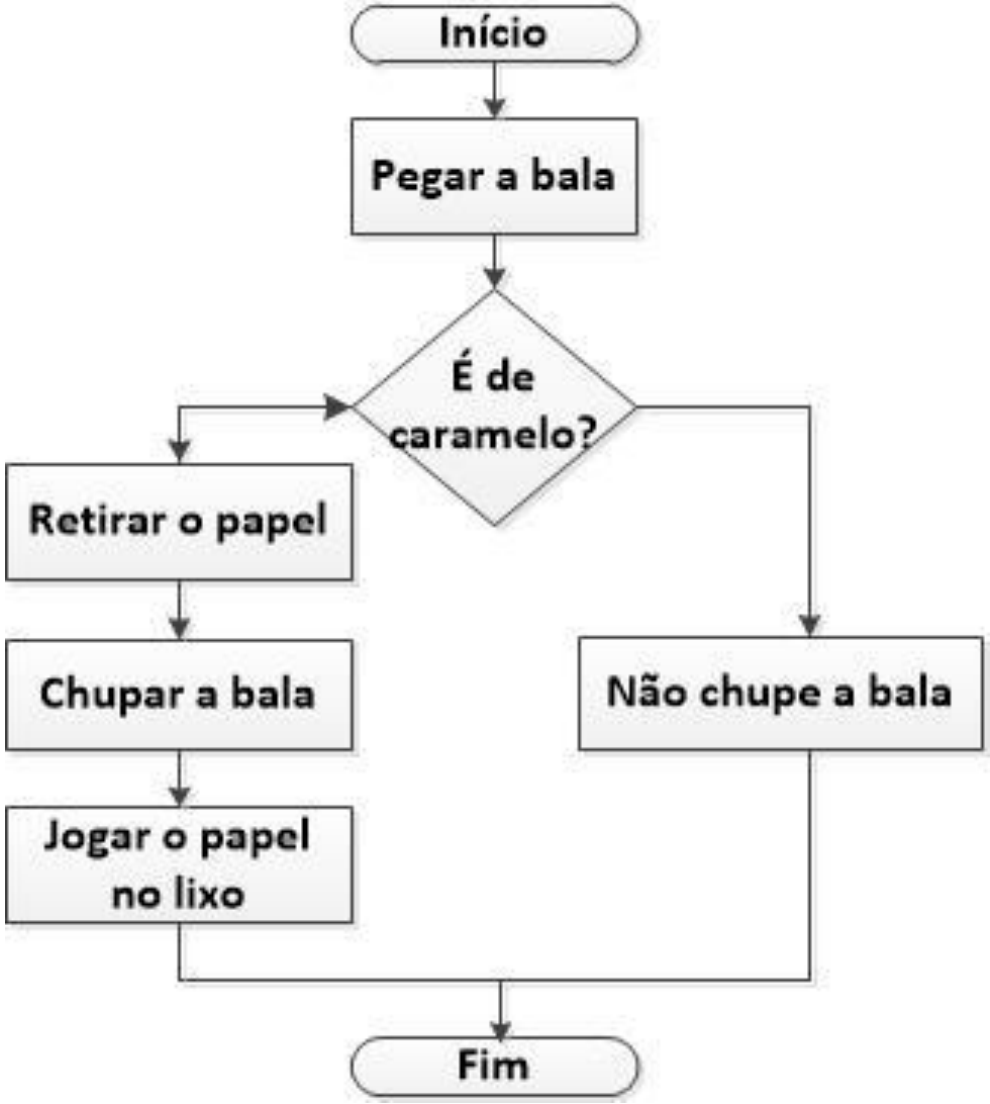
Operadores Lógicos

- **Exemplo**: algoritmo "Chupar uma bala" (imagine que algumas pessoas não gostem de chupar bala de caramelo, neste caso teremos que modificar o algoritmo para:

1- Pegar a bala.
2- A bala é de caramelo?
3- Se sim, não chupe a bala.
4- Se não, continue com o algoritmo.
5- Retirar o papel.
6- Chupara a bala.
7- Jogar o papel no lixo.

Operadores Lógicos

- **Fluxograma:**



Operador Lógico E / AND

- É utilizado quando todas as proposições lógicas de uma condição necessitam ser verdadeiras.
- A seguir, a **tabela verdade** para esse tipo de operador:

1ª condição	Operador	2ª condição	Resultado
V	E	V	V
V	E	F	F
F	E	V	F
F	E	F	F

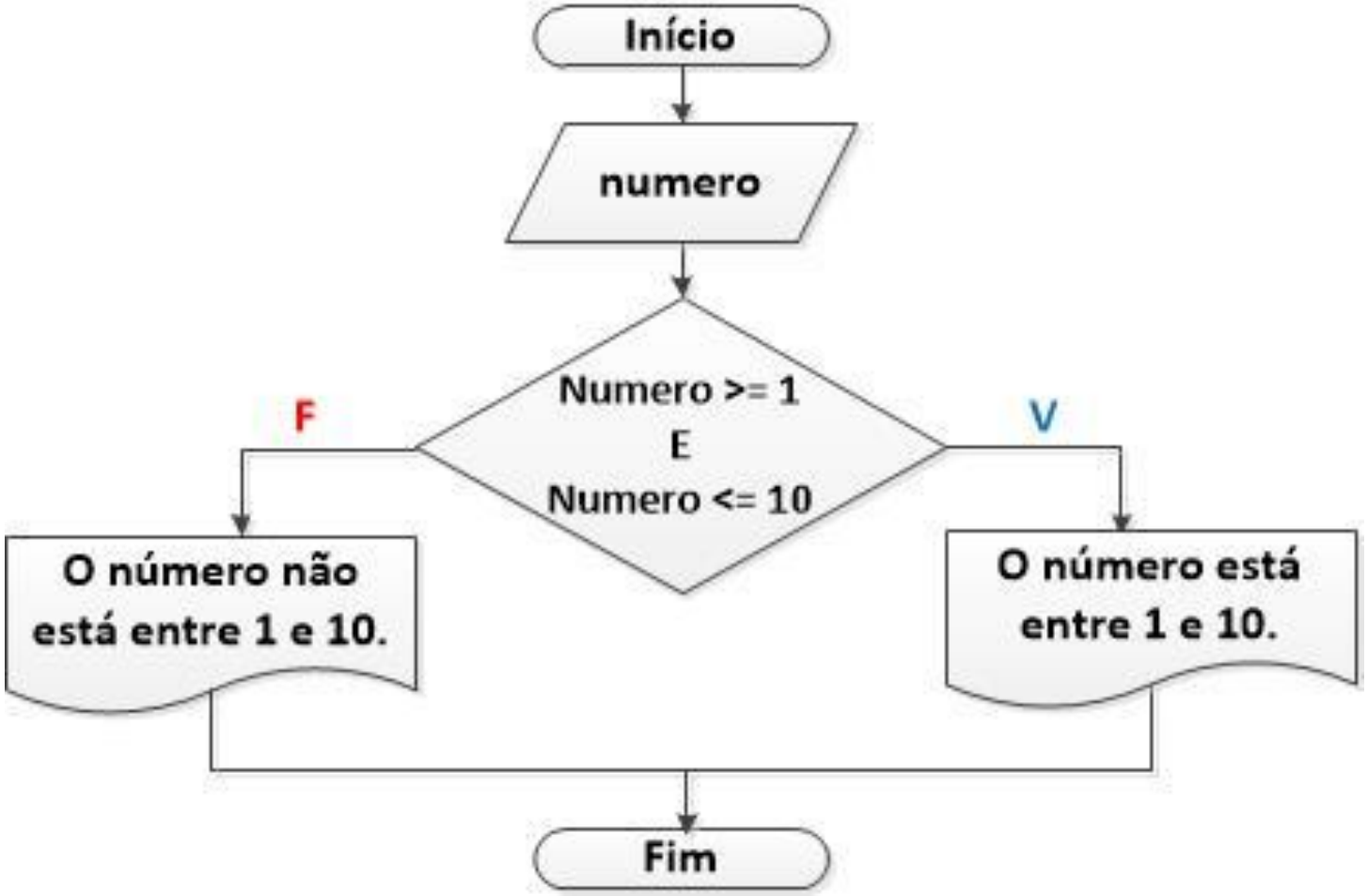
Operador Lógico E / AND

- **Portugol:**

```
PROGRAMA Exemplo_E
VAR
   numero: inteiro
INICIO
   Leia numero
   Se (numero >= 1) E (numero <= 10) entao
        Escreva "O número está entre 1 e 10"
     Senao
        Escreva "O número está entre 1 e 10"
   Fimse
FIM
```

Operador Lógico E / AND

- **Fluxograma:**



Operador Lógico OU / OR

- É utilizado quando pelo menos uma das proposições lógicas de uma condição necessitam ser verdadeiras.
- A seguir, a **tabela verdade** para esse tipo de operador:

1ª condição	Operador	2ª condição	Resultado
V	OU	V	V
V	OU	F	V
F	OU	V	V
F	OU	F	F

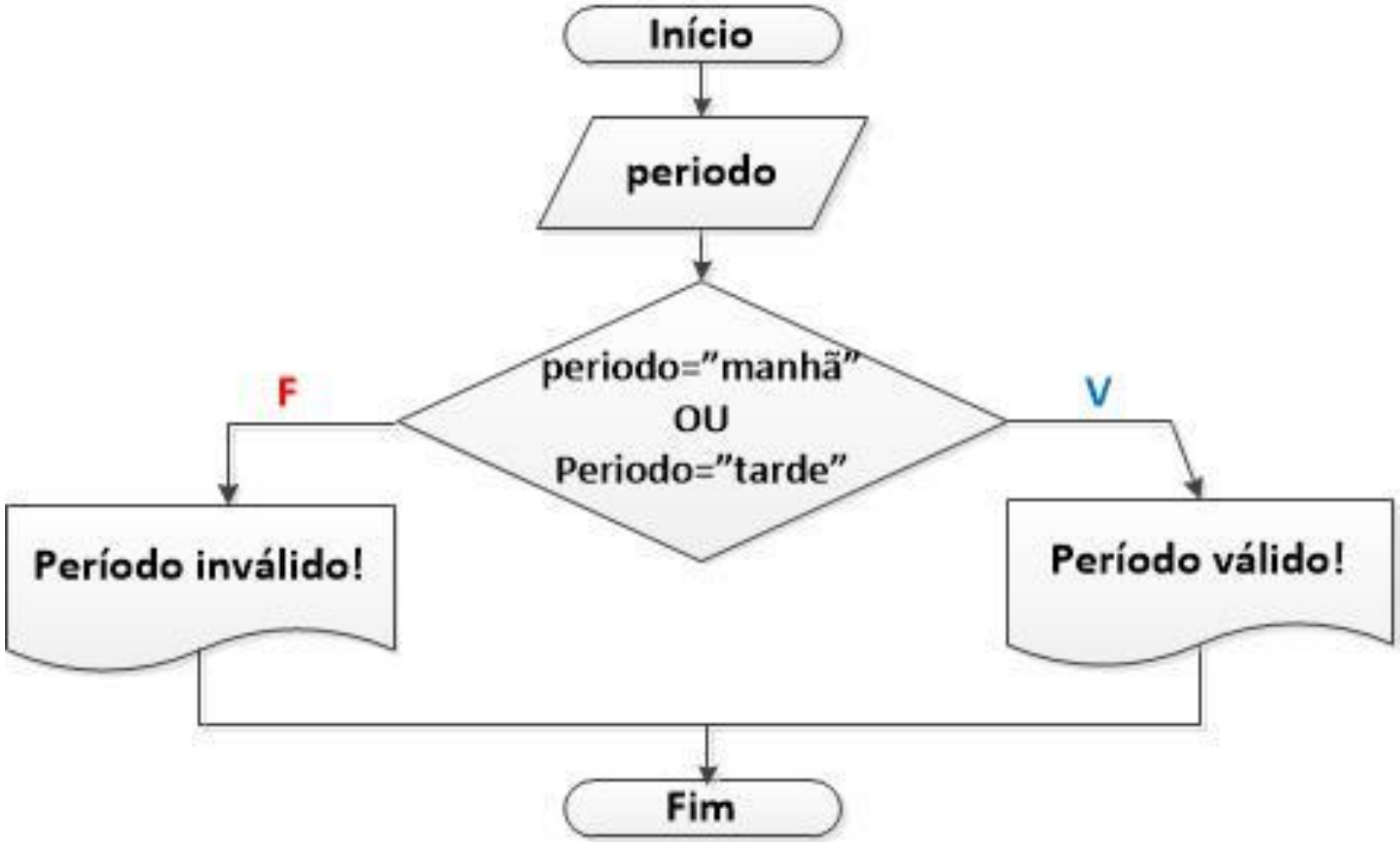
Operador Lógico OU / OR

- **Portugol:**

```
PROGRAMA Exemplo_OU
VAR
   periodo: caracter
INICIO
   Leia periodo
   Se (periodo = "manhã") OU (periodo = "tarde") entao
        Escreva "Período válido!"
      Senao
        Escreva "Período inválido!"
   Fimse
FIM
```

Operador Lógico OU / OR

- **Fluxograma:**



Operador Lógico NÃO / NOT

- É utilizado quando há necessidade de **inverter** o valor lógico de uma determinada condição. Se a condição for **verdadeira** assumirá o valor **falso**, e se a condição for **falsa** assumirá o valor **verdadeiro**.
- A seguir, a **tabela verdade** para esse tipo de operador:

1ª condição	Operador	Resultado
V	NÃO	F
F	NÃO	V

Operador Lógico NÃO / NOT

- **Portugol:**

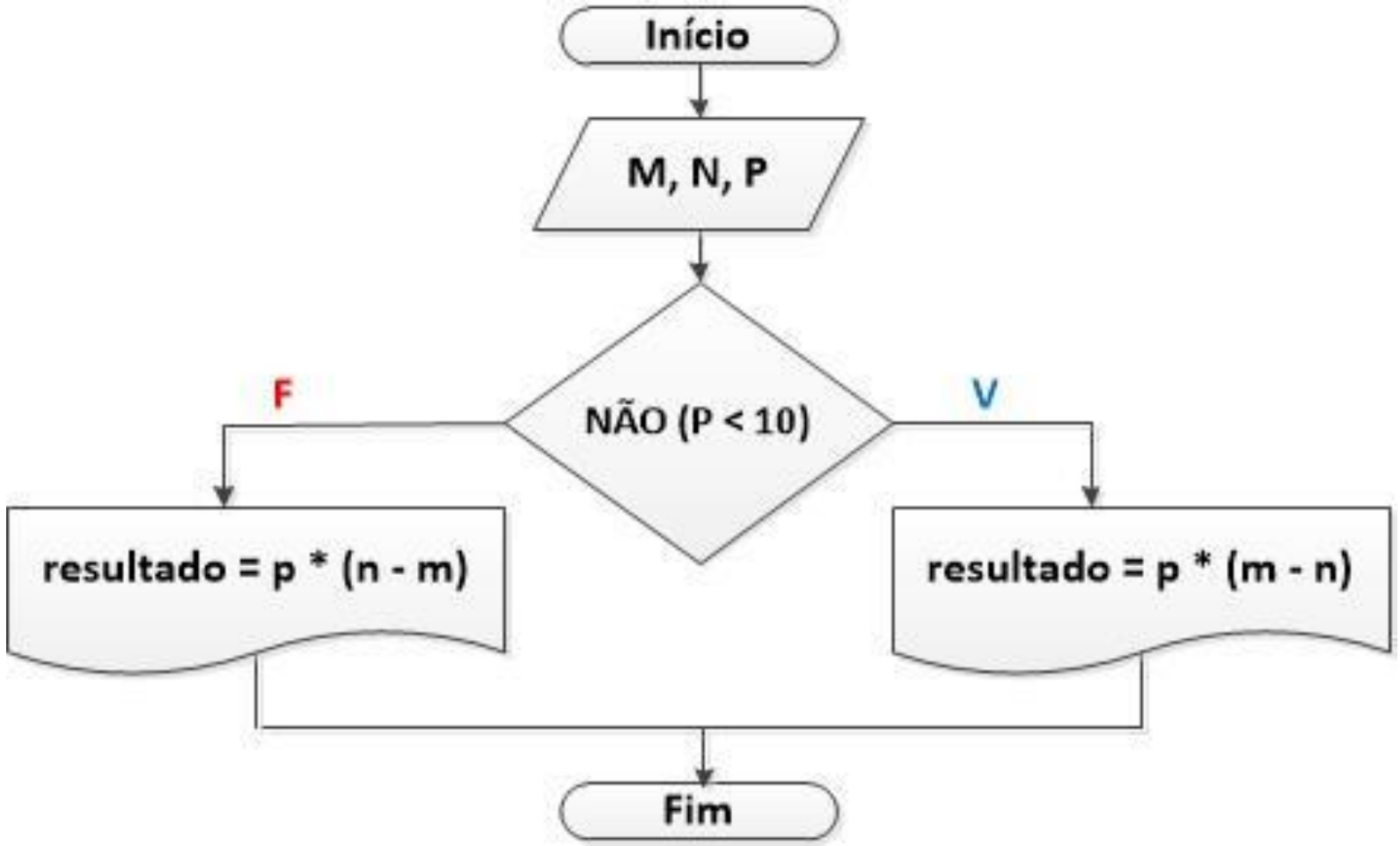
```
PROGRAMA Exemplo_nao
VAR
   m, n, p, resultado : inteiro
INICIO
   Leia m, n, p
   Se NAO (p < 10") entao
        resultado ← p * (m – n)
     Senao
        resultado ← p * (n – m)
   Fimse
FIM
```

Observação:

- ✓ o cálculo resultado ← p * (m – n) só é executado se p for maior que 10. Para qualquer valor abaixo de 10 será efetuado o cálculo resultado ← p * (n – m).

Operador Lógico NÃO / NOT

- **Fluxograma:**



Referência Bibliográfica

- SILVA, Camila Ceccatto da e PAULA, Everaldo Antônio de. Lógica de Programação: aprendendo a programar. Santa Cruz do Rio Pardo, SP: Viena, 2007.

- SIMÃO, Daniel Hayashida e REIS, Wellington José dos. Lógica de Programação: conhecendo algoritmos e criando programas. Santa Cruz do Rio Pardo, SP: Viena, 2015.