

Fundamentos de Logica de Programação (I) - Descomplicada

“A Lógica nada mais é do que a concretização das ideias, reunidas através de sua abstração da realidade e da necessidade.” (MORAES, Rogério)

Princípios Elementares de um Algoritmo

A construção de um algoritmo nada mais é que reunir 3 (três) elementos básicos para que seja realizada uma determinada função, ou seja, **Entrada** (Através de variáveis e/ou dispositivos), **Processamento** (Realização de funções Matemáticas/Processos) e **Saída** (Representação na tela ou em arquivos, tanto de resultados, assim como textos informativos).

Importante! Sempre para haver um processamento, precisamos ter entrada de algum valor, a regra básica é **E + P = S**, entretanto, a **Entrada** pode ser um valor padrão (inicialização de variável, tanto com um valor pré-determinado, como em uma estrutura de looping [Para], onde a regra exige que seja inicializada dentro da própria estrutura, onde haverá sobre a mesma um auto incremento, através do qual haverá preenchimento de valores sequenciais). Pode uma estrutura apresentar múltiplas **Entradas** e **Processamentos** antes da saída.

Ex. **E + E + P = S** assim como **E + P + E = S** e também saída antes **S + E + E + P = S**.

Elementos básicos para Construção do Diagrama de Blocos

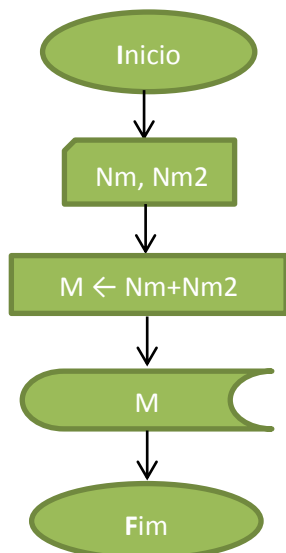
A Logica (Algoritmo) Possui muitas representações, entre elas o Clássico, o Diagrama de Blocos e o Português estruturado. Através dos quais construímos uma aplicação.

Elementos básicos para Construção do Diagrama de Blocos



Primeiro Exemplo de um Aplicativo (1)

A estrutura básica que utilizaremos em nosso aplicativo “soma” será construída em formato de algoritmo (Diagrama de Blocos e Português Estruturado).



- 1) Iniciamos nosso programa com o cabeçalho.
- 2) Entramos com valores via teclado, [Nm e Nm2].
- 3) Processamos A Soma de [Num e Nm2] e implicamos (armazenamos) “←” seu valor em [M].
- 4) Apresentamos o resultado que esta em [M] na tela.
- 5) Finalizamos o programa

Português Estruturado

Linguagem estrutural (semelhante) utilizada para passar das ideias do diagrama de blocos, para algo mais sólido, próximo a linguagem final (linguagem de programação), o português estruturado também conhecido com o termo “Portugol”, que é utilizado para trazer um melhor entendimento de seu funcionamento.

Extrutura do Primeiro Exemplo de um Aplicativo (1) em português estruturado.

Programa Soma; { cabeçalho do programa
var nm,num2,m:**Inteiro**; { onde iniciaremos (criaremos) as variáveis e determinamos seu formato
Início: { determina o início do programa
leia nm; { entrada de valores via teclado
leia nm2; { entrada de valores via teclado
m ← nm + nm2; { Processamento, onde “←” implica valor na variável. Recebe o resultado
Escreva “m”;; { Imprimimos o valor da variável na tela
Fim { finalizamos o programa

Principais formatos (Iniciais)

Inteiros = números sem casas decimais. (... -1 .. 0... 1...)

Real = Para números com casas decimais. (...-75,58 ... -1 .. 0... 1... 1,2...)

Caractere = Para caracteres (Texto). (a,b,c,2001)

Operadores Matemáticos Essenciais

+	-	↑	/	*	+
Adição	Subtração	Exponencial	Divisão	Multiplicação	Adição

Operadores Relacionais

=	<>	>	<	≥ ou >=	≤ ou <=
Igual	Diferente	Maior	Menor	Maior ou igual	Menor Ou Igual

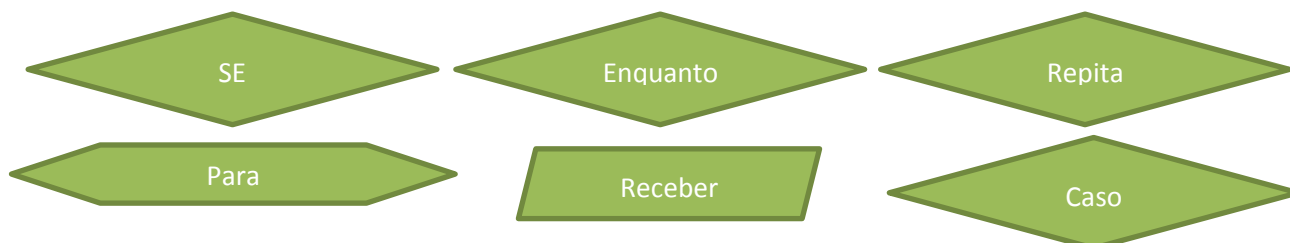
Operadores Lógicos

.e.			.ou.			.Não.	
Condição 1	Condição 2	Resultado	Condição 1	Condição 2	Resultado	Condição	Resultado
Falsa	Falsa	Falsa	Falsa	Falsa	Falsa	Verdadeira	Falsa
Verdadeira	Falsa	Falsa	Verdadeira	Falsa	Verdadeira	Falsa	Verdadeira
Falsa	Verdadeira	Falsa	Falsa	Verdadeira	Verdadeira		
Verdadeira	Verdadeira	Verdadeira	Verdadeira	Verdadeira	Verdadeira		

Estruturas Condicionais

São estruturas usadas para toda de decisões em logica de programação, onde controlamos determinada execução de processos e/ou saída de dados na tela por exemplo.

Elementos básicos para Construção do Diagrama Condicional e Looping

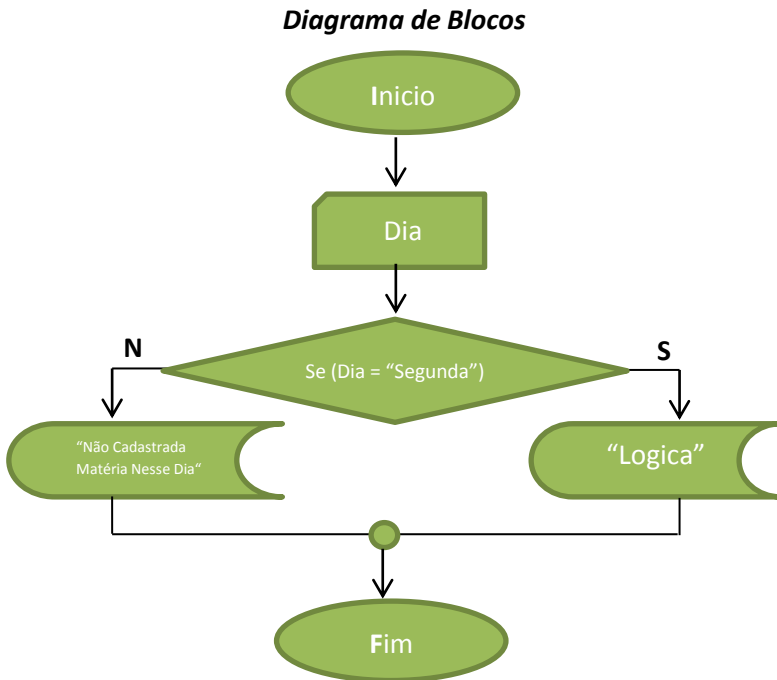


Estrutura SE

Através dela conseguimos determina Se hoje for segunda-feira, qual seria a matéria estudada e assim exibir a matéria correspondente ao dia. Se a condição for favorável exibirá o valor, senão ela exibira outra mensagem e/ou executara outra condição até que seja favorável.

Exemplo 1

- 1) Efetuar uma entrada de dados referente ao dia da semana, e logo em seguida exibir a matéria correspondente ao mesmo.

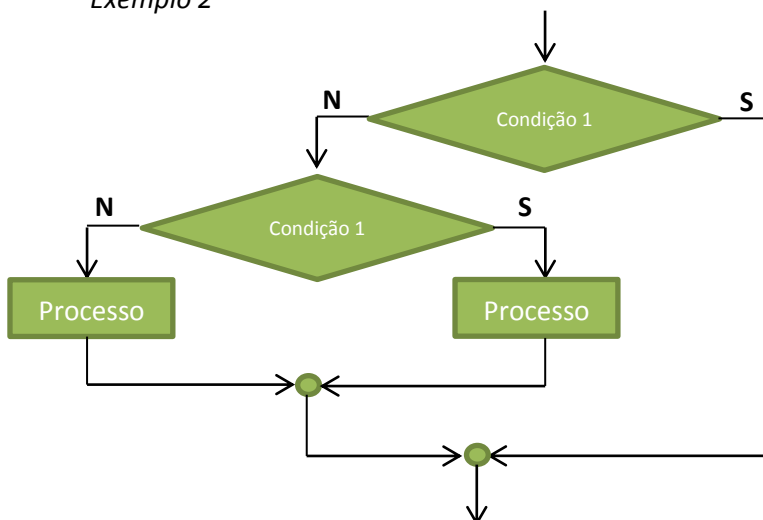


Português Estruturado

```
Programa Dia_Semana;  
var dia:caractere[10];  
Inicio:  
Leia dia;  
Se (dia = "Segunda") Então  
    Escreva "Logica";  
Senão  
    Escreva "Não Cadastrada  
Matéria Nesse Dia.";  
Fim_Se  
Fim
```

No nosso exemplo a estrutura **SE** irá validar se foi dada uma entrada igual a palavra "Segunda", caso tenha sido dada, ele informara a primeira parte da condição, onde apresentara a saída "Logica", senão ela exibira a segunda, que é realizada quando a primeira não é satisfatória.

Exemplo 2

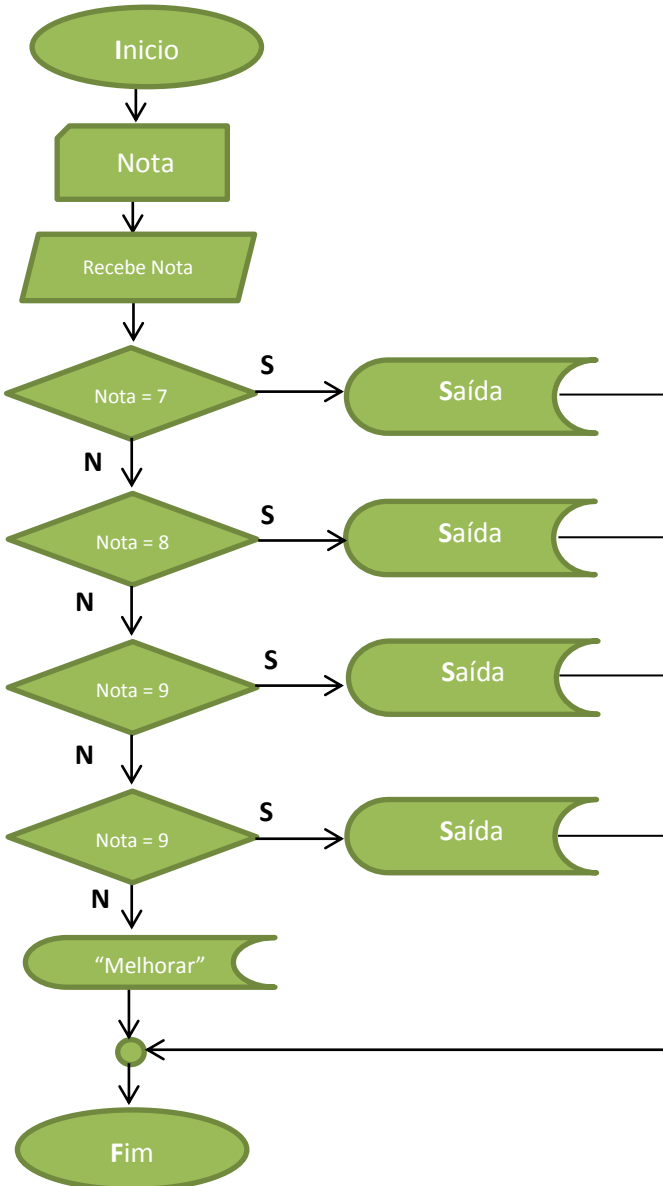


No caso do **SE** composto, informaremos o segundo **SE** logo após o senão. Conforme no exemplo ao lado. Com seu fechamento da forma correta. Ou seja, da ultima condição a ser processada para a primeira, em ordem decrescente.

Estrutura Caso

Utilizada geralmente para simplificar uma tomada de decisão, ou seja, sendo entrada igual a um numero na maior parte das vezes, ela apresenta o item ou passa para o seguinte, até que a mesma seja satisfatória a condição de acordo com o valor da entrada.

Diagrama de Blocos



Português Estruturado

```
Programa notas
Var nota: inteiro;
Inicio:
leia nota;
Recebe(nota)
caso 7: escreva "Regular";
parar
caso 8: escreva "Muito bem";
parar
caso 9: escreva "Parabéns";
parar
caso 10: escreva "Excelente";
parar
Padrão: escreva "Favor
Melhorar"
Fim_Caso
Fim
```

O Seleccionar, no caso "Switch" server para carregar a variável na condição, onde caso o valor seja igual ao atual ele executa e sai da função, ou ele segue até o padrão "default" onde por si só ele finaliza.

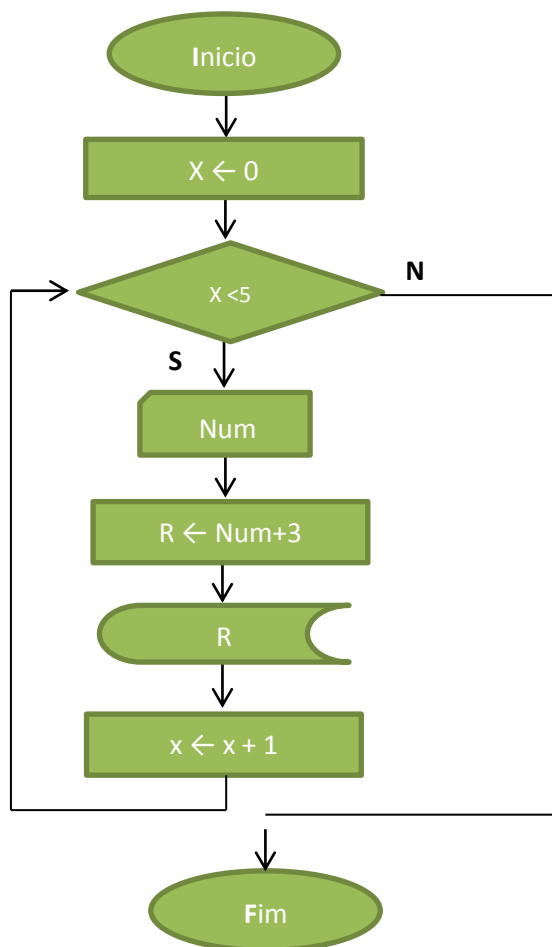
Estruturas de Loopings

São estruturas que são utilizadas geralmente para executar determinada função, ate que sua condição (gerada por contador) seja satisfatória, a mesma é utilizada para fazer cálculos e apresentar valor de variáveis (Tabelas / Matrizes).

Estrutura Enquanto e Repita

Estruturas de looping, usadas para fazer o mesmo processo ate que seu contador atinja o limite pré-determinado.

Estrutura Enquanto



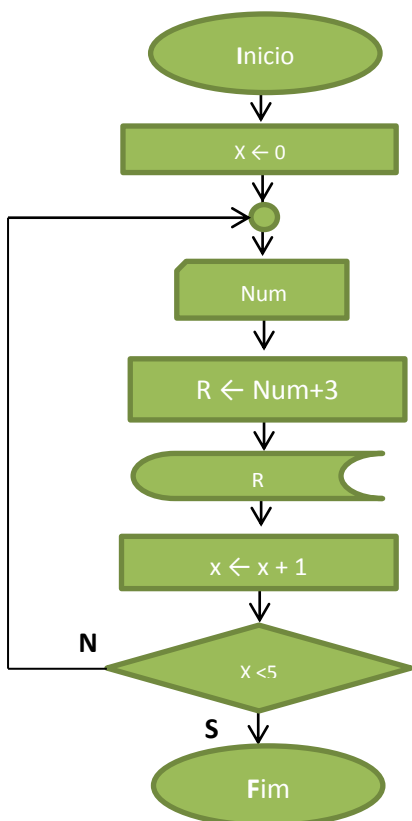
Português Estruturado

```

Programa Looping_1;
var x,num:inteiro;
Início:
x ← 0;
Enquanto (X < 5) Faça
leia num;
R ← num + 3;
escreva R;x
x=x+1;
Fim_enquanto;
Fim
    
```

No Enquanto é executado por uma condição onde apenas após o seu valor ser igual ao da variável de controle, termina o looping e passa para o próximo procedimento (função). Essa variável devera ser de auto incremento, ou seja, ela recebera seu próprio valor + 1 ($R \leftarrow num + 3$).

Estrutura Repita



Português Estruturado

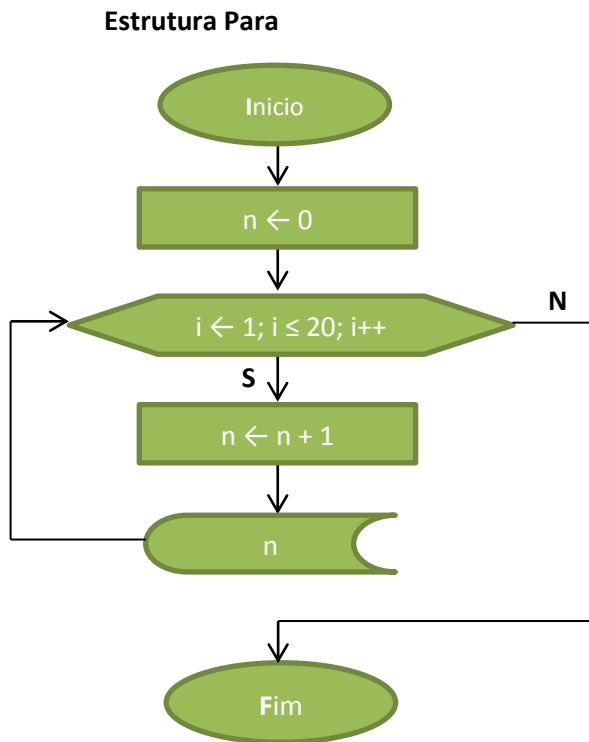
```

Programa Looping_2;
var x,num:inteiro;
Início:
x ← 0;
Repita
leia num;
R ← num + 3;
escreva R;
x=x+1;
Até que (x<5)
Fim
    
```

Semelhante ao enquanto, mas a condição apenas vem no final de sua estrutura. Ambas são estruturas utilizadas também em menus para repetir um procedimento.

Estrutura Para

A estrutura para é uma das mais utilizadas, pois já possui um contador próprio. Desde ordenar textos (classificar / ordem alfabética) a controle de manipulação de matriz é a mais indicada.



Português Estruturado

```
Programa Contagem_Regressiva;  
var i,n:inteiro;  
Início:  
n ← 0;  
Para i de 1 até 20 passos 1 faça  
n ← n + 1;  
escreva n;  
Fim_Para  
Fim
```

No para conseguimos fazer um looping de forma mais simples, onde usamos para manipular dados contínuos. A variável i recebe auto incremento de +1 a cada vez que passa por ela, até que chegue ao valor desejado.