

6 Algoritmo: Controle de Fluxo

Até o momento os algoritmos estudados utilizam apenas instruções primitivas de atribuição, e de entrada e saída de dados. Qualquer conjunto de dados fornecido a um algoritmo destes será submetido ao mesmo conjunto de instruções, executadas sempre na mesma seqüência.

No entanto, na prática muitas vezes é necessário executar ações diversas em função dos dados fornecidos ao algoritmo. Em outras palavras, dependendo do conjunto de dados de entrada do algoritmo, deve-se executar um conjunto diferente de instruções. Além disso, pode ser necessário executar um mesmo conjunto de instruções um número repetido de vezes. Em resumo, é necessário controlar o fluxo de execução das instruções (a seqüência em que as instruções são executadas num algoritmo) em função dos dados fornecidos como entrada ao mesmo.

Neste capítulo serão estudadas as estruturas básicas de controle do fluxo de instruções de um algoritmo. De acordo com o modo como este controle é feito, estas estruturas são classificadas em:

- Estruturas seqüenciais;
- Estruturas de decisão;
- Estruturas de repetição.

6.1 Comandos Compostos

Um comando composto é um conjunto de zero ou mais comandos (ou instruções) simples, como atribuições e instruções primitivas de entrada ou saída de dados, ou alguma das construções apresentadas neste capítulo.

Este conceito é bastante simples e será útil e conveniente nos itens seguintes, na definição das estruturas básicas de controle de execução.

6.2 Estrutura Seqüencial

Na estrutura seqüencial os comandos de um algoritmo são executados numa seqüência pré-estabelecida. Cada comando é executado somente após o término do comando anterior.

Em termos de fluxogramas, a estrutura seqüencial é caracterizada por um único fluxo de execução (um único caminho orientado) no diagrama. Em pseudocódigos, a estrutura seqüencial caracteriza-se por um conjunto de comandos dispostos ordenadamente. Como exemplos de aplicação desta estrutura de controle tem-se os algoritmos do capítulo anterior, onde não há estruturas de decisão ou de repetição.

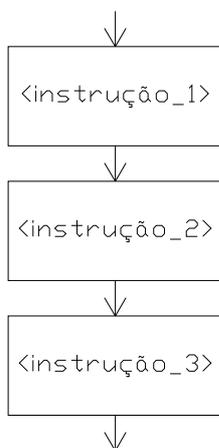


Figura 41: Trecho de um algoritmo seqüencial.

As linhas de código abaixo, apresentam um trecho de um algoritmo seqüencial em português estruturado, para o exemplo da Fig. 41.

```
...  
<instrução_1>  
<instrução_2>  
<instrução_3>  
...
```

6.3 Estruturas de Decisão

Neste tipo de estrutura o fluxo de instruções a ser seguido é escolhido em função do resultado da avaliação de uma ou mais condições. Uma condição é uma expressão lógica.

A classificação das estruturas de decisão é feita de acordo com o número de condições a serem testadas. Segundo esta classificação, têm-se dois tipos de estruturas de decisão:

- Se;
- Escolha.

6.3.1 Estruturas de Decisão do Tipo Se

Nesta estrutura uma única condição (expressão lógica) é avaliada. Se o resultado desta avaliação for verdadeiro (V), então um determinado conjunto de instruções é executado. Caso contrário, ou seja, quando o resultado da avaliação for falso (F), um comando diferente é executado.

Em termos de fluxogramas, uma construção do tipo **Se** pode ser encarada como uma bifurcação onde há dois caminhos que podem ser seguidos. Este tipo de estrutura é denominada de estrutura condicional composta, a Fig. 42 mostra sua sintaxe. A execução do algoritmo prosseguirá necessariamente por um deles. Esta escolha é feita em função do resultado da expressão: um dos caminhos é rotulado com (V) e será seguido quando a condição for falsa.

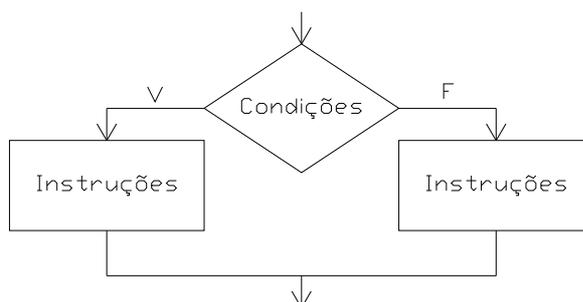


Figura 42: Fluxograma da estrutura condicional composta.

As linhas de código abaixo, apresentam a sintaxe de uma estrutura condicional composta em português estruturado, para o exemplo da Fig. 42.

```

Se (<condição>) então
{Instruções}
Senão
{Instruções}
Fim Se
  
```

Observação: As palavras **Se**, **Então**, **Senão** e **Fim Se**, são reservadas.

A semântica desta construção é a seguinte: a condição é avaliada. Se o resultado for verdadeiro, um grupo de instruções será executado. Ao término de sua execução o fluxo do algoritmo prossegue pela instrução seguinte à construção, ou seja, o primeiro instrução

após o **Fim Se**. Nos casos em que a condição é avaliada como falsa, o outro grupo de instruções será executado.

Há casos particulares deste tipo de construção, onde apenas um ramo do teste condicional será responsável pela realização de alguma ação, assim a porção relativa ao **Senão** pode ser omitida, resumindo a sintaxe da construção à forma mostrada na Fig. 43.

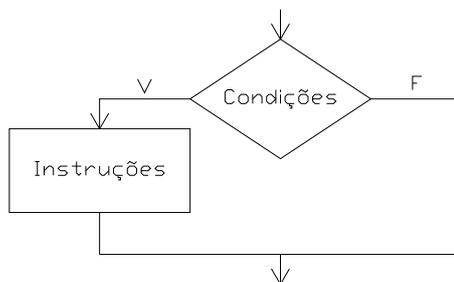


Figura 43: Fluxograma de uma estrutura condicional simples.

A sintaxe de uma estrutura condicional simples em português estruturado é:

```
Se (<condição>) então
{Instruções}
Fim Se
```

A semântica desta construção para o caso da condição ser verdadeira, as instruções serão executadas. Após seu término, o fluxo de execução prossegue pela próxima instrução após o **Fim Se**. Quando a condição é falsa, o fluxo de execução prossegue normalmente pela primeira instrução após o **Fim Se**.

Exemplo 5. Algoritmo para determinar se uma pessoa é maior ou menor de idade, utilizando a estrutura de decisão composta **Se**.

O algoritmo, em português estruturado, tem a seguinte forma:

```
Programa Exemplo_5
  Var Idade: Inteiro;
Início
  Leia(Idade);
  Se (Idade >= 18) então
    Escreva("Maior de idade")
  Senão
    Escreva("Menor de idade")
  Fim Se
Fim
```

O fluxograma para este problema utilizando a estrutura **Se** é mostrado na Fig. 44.

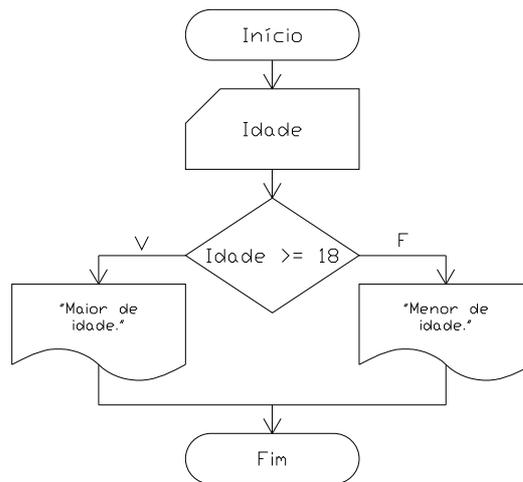


Figura 44: Exemplo de fluxograma de uma estrutura condicional.

6.3.2 Estruturas de Decisão do Tipo Escolha

Este tipo de estrutura é uma generalização da estrutura Se, onde a condição avaliada conduz a no máximo duas possibilidades de ações. Na estrutura de decisão do tipo Escolha podem existir uma ou mais possibilidades de ações a serem tomadas. A Figura 45 mostra a sintaxe em fluxograma para a estrutura escolha.

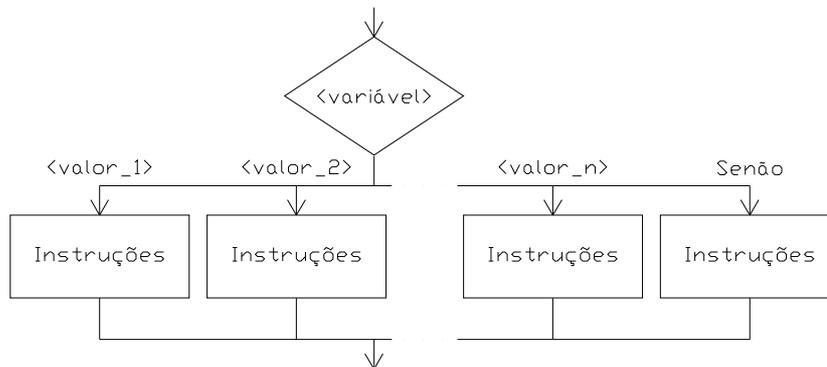


Figura 45: Fluxograma de uma estrutura condicional escolha.

A sintaxe de uma estrutura condicional do tipo escolha em português estruturado é:

```

Escolha (<variável>)
  Caso(<valor_1>): {Instruções}
  ⋮
  Caso(<valor_n>): {Instruções}
  Senão: {Instruções}
Fim Escolha
  
```

Para a estrutura condicional **Escolha**, será testado o valor da <variável>, se o valor estiver presente em alguma das condições(Caso(<valor>)), as instruções a ele associadas serão executadas. O usuário pode entrar com um valor não contido na lista de condições, então será executado as instruções associadas ao **Senão**. O uso do **Senão** é facultativo podendo ser excluído da estrutura do **Escolha**.

Exemplo 6. Algoritmo para calcular o reajuste salarial em função da profissão, utilizando a estrutura de decisão **Escolha**.

O fluxograma para uma aplicação desta construção é mostrado na Fig. 46.

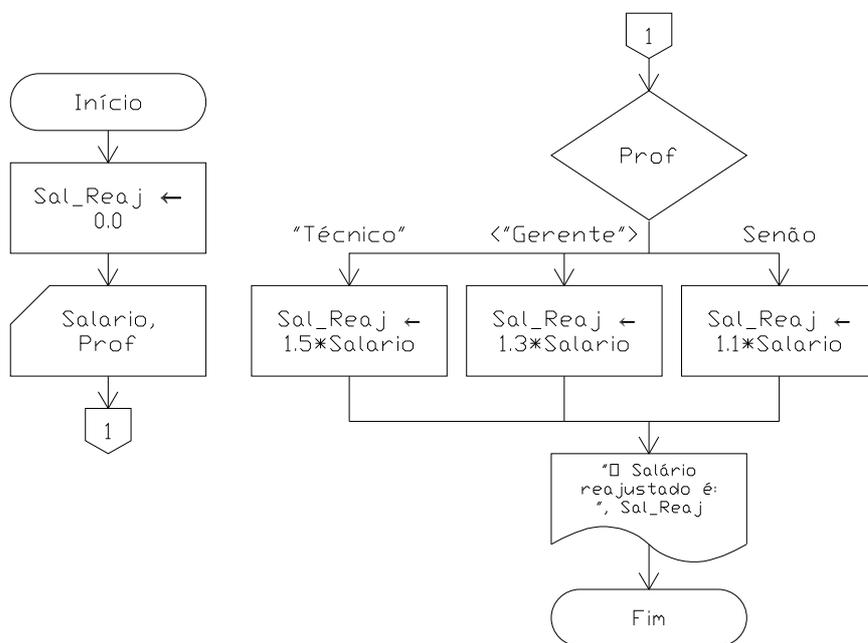


Figura 46: Exemplo de fluxograma que utiliza Escolha.

As linhas de código abaixo, apresentam o exemplo da Fig. 46 em português estruturado.

```

Programa Exemplo_6
  Var Sal_Reaj, Salario: Real;
    Prof: Char[10];
Início
  Sal_Reaj ← 0.0;
  Leia(Salario, Prof);
  Escolha(Prof)
    Caso("Técnico"): Sal_Reaj ← 1.5*Salario;
    Caso("<Gerente>"): Sal_Reaj ← 1.3*Salario;
    Senão: Sal_Reaj ← 1.1*Salario;
  Fim Escolha
  Escreva("O salário reajustado é: ", Sal_Reaj)
Fim
  
```

6.4 Estruturas de Repetição

São muito comuns as situações em que se deseja repetir um determinado trecho de um programa certo número de vezes. Por exemplo, pode-se citar o caso em que se deseja realizar um mesmo processamento para conjuntos de dados diferentes. Exemplo: processamento de folha de pagamentos de uma empresa em que o mesmo cálculo é efetuado para cada um dos funcionários. As estruturas de repetição são também denominadas Laços (*Loops*).

Baseado no conhecimento prévio do número de vezes que o conjunto de instruções será executado, as estruturas de repetição podem ser classificadas em:

Laços contados: quando se conhece previamente quantas vezes o comando composto no interior da construção será executado;

Laços condicionais: quando não se conhece o número de vezes que o conjunto de instruções no interior do laço será repetido, pois a condição testada é modificada pelas instruções do interior do laço.

6.4.1 Estrutura de Repetição Para-faça

Este tipo de estrutura é útil quando se conhece previamente o número de vezes que se deseja executar um determinado conjunto de comandos. Então, este tipo de laço nada mais é que uma estrutura dotada de mecanismos para contar o número de vezes que o corpo do laço é executado. A sintaxe usada é mostrada na Fig. 47.

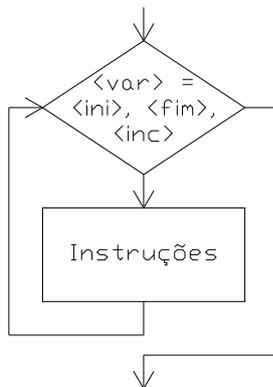


Figura 47: Fluxograma de uma estrutura de repetição Para-faça.

A sintaxe de uma estrutura de repetição Para-faça em português estruturado é:

```
Para(<var> ← <ini> até <fim> passo <inc>) faça
{Instruções}
Fim Para
```

A semântica do laço contado é a seguinte: no início da execução da estrutura o valor <ini> é atribuído à variável <var>. A seguir, o valor da variável <var> é comparado com o valor <fim>. Se <var> for maior que <fim>, então a execução do algoritmo prossegue a partir da primeira instrução posterior ao Fim Para. Por outro lado, se o valor de <var> for menor ou igual a <fim>, as instruções contidas no laço serão executadas, incrementando o valor da variável <var>. Isto repete-se até que <var> tenha um valor maior que <fim>.

Algumas observações interessantes devem ser feitas:

- <var> é necessariamente uma variável, uma vez que seu valor é alterado a cada iteração (repetição do laço);
- <ini>, <fim> e <inc> podem ser constantes ou variáveis. Quando variáveis, algumas linguagens de programação proíbem que seus valores sejam modificados durante a execução do laço;
- <inc> é o valor que é adicionado à variável <var> ao final de cada iteração do laço. Há linguagens de programação que permitem que lhe seja atribuído um valor negativo, de modo que o valor da variável <var> diminui a cada iteração. Neste caso, deve-se atentar ao sinal da comparação, se deve ser utilizado > ou <, para o correto funcionamento do algoritmo;
- Na grande maioria dos casos <inc> tem o valor 1, portanto, admite-se a omissão do trecho de incremento (<inc>) da sintaxe do estrutura Para-Faça, quando isto ocorre, assume-se que o incremento é 1.

Exemplo 7. Calcule a média aritmética entre duas notas para trinta alunos de uma sala.

O fluxograma para este problema utilizando a estrutura de repetição **Para-faça** é mostrado na Fig. 48.

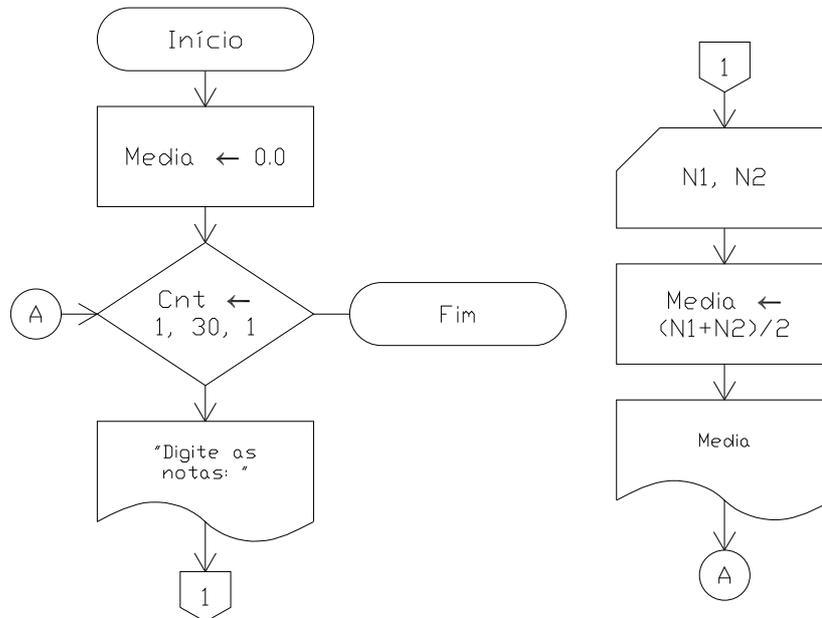


Figura 48: Exemplo de fluxograma de uma estrutura **Para-faça**.

As linhas de código abaixo, apresentam o fluxograma da Fig. 48 em português estruturado.

```

Programa Exemplo_7
  Var N1, N2, Media: Real;
  Cnt: Inteiro;
Início
  Media ← 0.0;
  Para(Cnt ← até 30 passo 1) faça
    Escreva("Digite as notas:");
    Leia(N1, N2);
    Media ← (N1+N2)/2;
    Escreva(Media);
  Fim Para
Fim
  
```

6.4.2 Estrutura de Repetição Enquanto

Sua semântica é a seguinte: ao início da estrutura de repetição **Enquanto** a condição é testada. Se o resultado do teste for falso, então as instruções no seu interior não serão executadas e a execução prossegue normalmente pela instrução seguinte ao **Fim Enquanto**.

Se a condição for verdadeira as instruções serão executadas e ao seu término retorna-se ao teste da condição. Assim, o processo acima será repetido enquanto a condição testada for verdadeira.

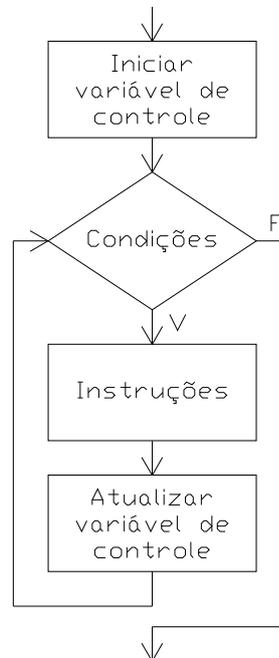


Figura 49: Fluxograma de uma estrutura de repetição **Enquanto**.

A sintaxe de uma estrutura de repetição **Enquanto** em português estruturado é:

```
{Iniciar variável de controle}
Enquanto(condição for verdadeira) faça
{Instruções}
{Atualizar variável de controle}
Fim Enquanto
```

Uma vez dentro do corpo do laço, a execução somente abandonará o mesmo quando a condição for falsa. O usuário deste tipo de construção deve estar atento à necessidade de que em algum momento a condição deverá ser avaliada como falsa. Caso contrário, o programa permanecerá indefinidamente no interior do laço, o que é conhecido como laço infinito.

Exemplo 8. Semelhante ao feito no Exemplo 7, calcule a média aritmética entre duas notas para trinta alunos de uma sala utilizando a estrutura de repetição **Enquanto**.

O fluxograma para este problema utilizando a estrutura de repetição **Enquanto** é mostrado na Fig. 50.

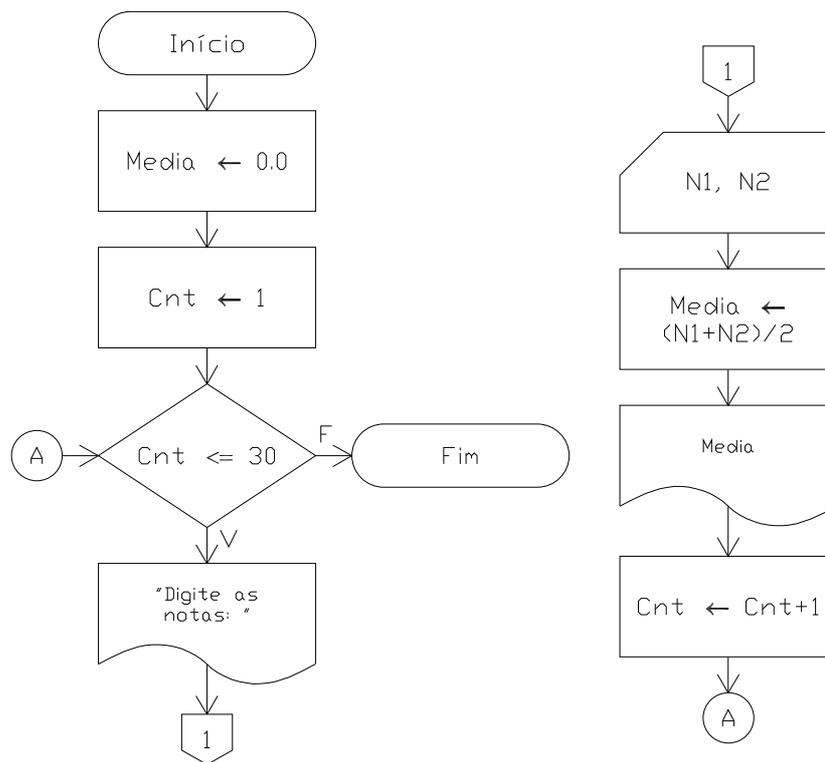


Figura 50: Exemplo de fluxograma de uma estrutura Para-faça.

As linhas de código abaixo, apresentam o fluxograma da Fig. 50 em português estruturado.

```

Programa Exemplo_8
  Var N1, N2, Media: Real;
      Cnt: Inteiro;
Início
  Media ← 0.0;
  Cnt ← 1;
  Enquanto(Cnt <= 30) faça
    Escreva("Digite as notas:");
    Leia(N1, N2);
    Media ← (N1+N2)/2;
    Escreva(Media);
    Cnt ← Cnt+1;
  Fim Enquanto
Fim
  
```

6.4.3 Estrutura de Repetição Repita

Seu funcionamento é bastante parecido ao da construção Enquanto, mas as instruções contidas no interior do laço serão executadas pelo menos uma vez. Em seguida a condição é testada, caso ela seja falsa, as instruções serão executadas novamente. Este processo é repetido até que a condição seja verdadeira, então a execução prossegue pela instrução

imediatamente posterior ao final da estrutura.

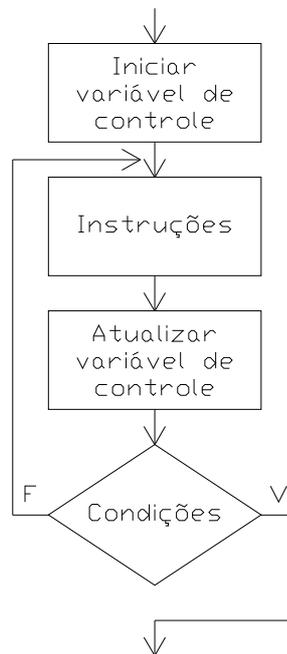


Figura 51: Fluxograma de uma estrutura de repetição Repita.

A sintaxe de uma estrutura de repetição Repita em português estruturado é:

```
{Iniciar variável de controle}
Enquanto(condição for verdadeira) faça
{Instruções}
{Atualizar variável de controle}
Fim Enquanto
```

Exemplo 9. Semelhante ao feito no Exemplo 7, calcule a média aritmética entre duas notas para trinta alunos de uma sala utilizando a estrutura de repetição Repita.

O algoritmo, em português estruturado, tem a seguinte forma:

```
Programa Exemplo_9
  Var N1, N2, Media: Real;
      Cnt: Inteiro;
Início
  Media ← 0.0;
  Cnt ← 1;
  Repita
    Escreva("Digite as notas:");
    Leia(N1, N2);
    Media ← (N1+N2)/2;
    Escreva(Media);
    Cnt ← Cnt+1;
  Até(Cnt > 30);
Fim
```

O fluxograma para este problema utilizando a estrutura Repita é mostrado na Fig. 52.

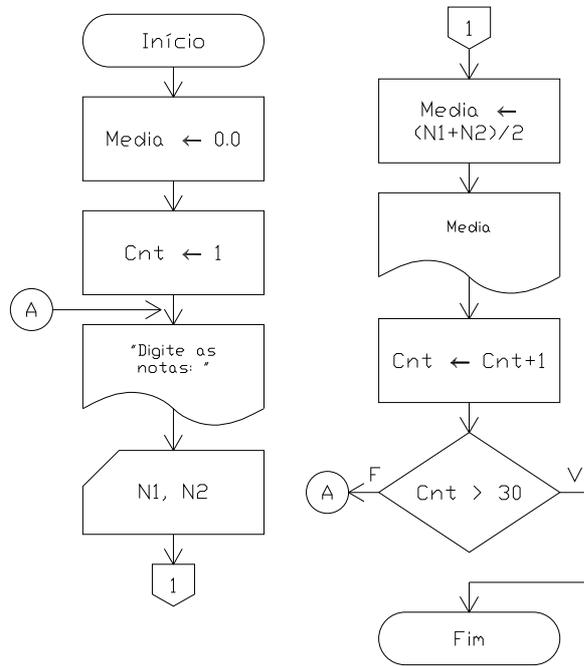


Figura 52: Exemplo de fluxograma de uma estrutura Repita.