

Tabela de conteúdo

Desvio usando controle de fluxo	1
Desvio usando controle de fluxo: Introdução	1
Uso de pares de controle.....	2
If / End If.....	2
Else If / End Else If.....	3
Else / End Else.....	6
While / End While.....	7
Do / Until	9
Select / End Select.....	11
Case / End Case	14
Default Case / End Default Case	15
Uso de loops genéricos	16
Usos da execução em loop.....	18
Criação de um loop.....	19
Uso do comando Imprimir em um loop	20
Uso de rótulos.....	22
Ir para um rótulo usando IRPARA.....	23
Redefinir loop.....	26
Desvio em um erro	26
Interfaces suportadas On Error	29
Desvio com subrotinas	30
Criação de uma nova subrotina	31

Edição de uma subrotina existente	36
Compreensão da caixa de diálogo Edição de argumento	37
Chamada de uma subrotina	38
Exemplos de subrotina.....	46
Finalização de uma rotina de medição	53

Desvio usando controle de fluxo

Desvio usando controle de fluxo: Introdução

Suponha que tenha uma peça com muitos elementos, mas queira medir apenas alguns deles várias vezes para obter um conjunto de dados estatísticos completo desses elementos. Suponha que queira saltar para uma peça específica na rotina de medição, dependendo da resposta do usuário. É possível executar essas tarefas e muitas outras usando os comandos de controle de fluxo. É possível controlar o fluxo da rotina de medição pela configuração de condições de determinados comandos.

Este capítulo fornece as informações necessárias para executar essas tarefas. Ela explica as declarações condicionais para sintaxe, loops e subrotinas. Ela também fornece muitos exemplos de código.

Os principais tópicos abordados neste capítulo incluem:

- Uso de pares de controle
- Uso de loops genéricos
- Uso de rótulos
- Desvio em um erro
- Desvio com sub-rotinas
- Finalização de uma rotina de medição

Comandos do modo Comando após comentários

Como vários dos exemplos de código nesse capítulo usam comandos `COMENTÁRIO` digitados, considere o seguinte:



Após inserir um comentário do PC-DMIS, para digitar comandos adicionais do PC-DMIS no modo Comando, primeiro pressione *duas vezes* o Enter após o comando

COMENTÁRIO. Isso informa ao PC-DMIS que você não quer mais adicionar texto ao comentário, mas deseja adicionar um novo comando.

Uso de pares de controle

O submenu **Inserir | Comando de controle de fluxo | Pares de controle** oferece diversos comandos em pares que funcionam dentro da janela Edição para comandar ou "controlar" o fluxo adequado da rotina de medição. Para inserir um comando do tipo par de controle na janela Edição, basta digitar o comando ou escolhê-lo nesse submenu.



Quando você usa uma declaração de desvio condicional para testar o valor de um comentário SIM / NÃO, esteja ciente de que o teste deve procurar pelo valor "SIM" ou "NÃO" em maiúsculas. "Sim" ou "Não" em minúsculas não funcionará. Para obter informações sobre comentários, consulte o tópico "Inserção de comentários do programador" no capítulo "Inserção de comandos de relatório".

If / End If

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | If / End If** permite adicionar um bloco condicional à rotina de medição. Os itens entre os comandos IF e END IF executam somente se a expressão para o comando IF for verdadeira (diferente de zero). Caso contrário, o fluxo de execução salta para o primeiro comando após o comando **END/IF**.

A linha de comandos da janela Edição de uma declaração IF / END IF é:

IF/expressão

END_IF/

Para inserir os comandos If / End if, siga estes passos:

1. Coloque o cursor na posição desejada da Janela Edição.
2. Selecione **If / End if** a partir da barra de menu. As declarações IF / END IF aparecerão na Janela Edição.

Amostra de código de If / End if

Considere o seguinte exemplo para perguntar ao usuário se deseja medir um elemento de ponto.



```
C1= COMENT/SIMNÃO,Gostaria de medir o elemento de ponto,
PNT1?
IF/C1.INPUT=="SIM"
    PNT1=FEAT/POINT,RECT
    ...
    ...
    FIMMED/
END_IF/
```

Explicação do código da amostra

C1=COMENT/SIMNÃO

Essa linha obtém e armazena a resposta SIM ou NÃO do usuário.

IF/C1.ENTRADA=="SIM"

Essa linha é a expressão. Ela testa se a entrada do comentário 1 é um SIM. Se for um SIM, então a declaração IF será VERDADEIRA e continuará a executar as declarações depois da declaração IF, nesse caso, ela mede o elemento **PNT1**. Se for um NÃO, ela irá para a declaração **END_IF**.

END_IF

Essa linha finaliza a execução de comandos dentro do bloco de código IF / END IF. O PC-DMIS irá para qualquer comando depois dessa linha se o usuário clicar em **Não** no comentário.

Else If / End Else If

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | Else If / End Else If** permite adicionar um bloco condicional à rotina de medição. Os itens entre os comandos ELSE IF e END ELSE IF executam somente se a expressão para o comando ELSE IF for verdadeira (diferente de zero). O bloco ELSE IF / END ELSE IF deve ser posicionado diretamente *após* um bloco IF /END IF ou após outro bloco ELSE IF / END ELSE IF. Se todas as expressões IF / ELSE IF acima do bloco ativo tiverem sido avaliadas como falsas, a expressão será avaliada. Se a expressão for avaliada como falsa (zero), a execução saltará para o próximo comando após o comando END ELSE IF. Se alguma das expressões IF / ELSE IF acima do bloco atual for avaliada

como verdadeira, todos os blocos ELSE IF / END ELSE IF seguintes da sequência serão saltados.

A linha de comandos da janela Edição de uma declaração ELSE IF / END ELSE IF é:

```
ELSE_IF/expressão
```

```
END_ELSE_IF/
```

Para inserir os comandos ELSE IF / END ELSE IF:

1. Coloque o cursor no local desejado na Janela Edição, após uma declaração IF/END IF ou ELSE IF/END ELSE IF existente.
2. Selecione **Else If/End Else If** a partir da barra de menu. As declarações ELSE IF /END ELSE IF aparecerão na janela Edição.



Esse tipo de bloco é válido somente quando posicionado *após* um bloco IF / END IF ou ELSE IF / END ELSE IF. Pares de controle posicionados de forma inválida são mostrados em vermelho na janela Edição.

Amostra de código de Else If / End Else If

Considere o exemplo a seguir, que exibe uma mensagem notificando o usuário de quando um dos valores X, Y ou Z de um ponto medido excede as tolerâncias definidas:



```
PNT2=ELEM/PONTO,CARTES
...
...
FIMMEDS/
SE/PNT2.X<6.9 OR PNT2.X>7.1
COMENT/OPER,"O valor X medido de PNT2: " + PNT2.X + "
está fora da tolerância."
FIM_SE/
OUTRO_SE/PNT2.Y<3.3 OR PNT2.Y>3.5
    COMENT/OPER,"O valor Y medido para PNT2: " + PNT2.Y +
" está fora da tolerância."
FIM_OUTROSE/
```

```
OUTRO_SE/PNT2.Z<.9 OU PNT2.Z>1.1  
  COMENT/OPER,"O valor Z medido para PNT2: " + PNT2.Z +  
  " está fora da tolerância."  
FIM_OUTROSE/
```

Explicação do código da amostra

Este código testa primeiro o valor X do ponto. Se a condição for avaliada como falsa, então o código testará o valor Y. Se a condição de Y for avaliada como falsa, então ele testará o valor Z.

Se todas essas condições forem avaliadas como verdadeiras, o PC-DMIS exibirá o comentário associado a elas e ignorará as declarações condicionais restantes.

IF/PNT2.X7.1

Essa linha é a expressão. Ela testa se o valor X medido é menor que 6,9 ou maior que 7,1. Se exceder um desses limites, ela executará o primeiro comentário.

END_IF

Essa linha finaliza a execução de comandos dentro do bloco de código IF / END IF. O PC-DMIS irá para qualquer comando após essa linha se a condição IF THEN for avaliada como falsa.

ELSE_IF/PNT2.Y3.5

Essa linha é a expressão do primeiro comando ELSE_IF. Ela será executada somente se o bloco IF / END IF acima for falso. Ela testa se o valor Y medido é menor que 3,3 ou maior que 3,5. Se exceder um desses limites, ela executará o segundo comentário.

END_ELSEIF/

Essa linha finaliza a execução de comandos dentro do primeiro bloco de código ELSE IF / END ELSE IF.

ELSE_IF/PNT2.Z1.1

Essa linha é a expressão do segundo comando ELSE_IF. Ela será executada somente se o bloco ELSE IF/END ELSE IF acima for falso. Ela testa se o valor Z medido é menor que 0,9 ou maior que 1,1. Se exceder um desses limites, ela executará o terceiro comentário.

END_ELSEIF/

Essa linha finaliza a execução de comandos dentro do segundo bloco de código ELSE IF / END ELSE IF.

Else / End Else

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | Else / End Else** permite adicionar um bloco condicional à rotina de medição. Os itens entre os comandos ELSE e END ELSE executam somente se todos os outros blocos if /end if e else if / end else if acima do bloco else tiverem falhado (todos avaliados como zero). Para que sejam válidos, os blocos ELSE / END ELSE devem ser posicionados no fim de um conjunto de blocos IF / END IF ou ELSE IF / END ELSE IF.

A linha de comandos da janela Edição de uma declaração ELSE / END ELSE é:

```
ELSE/
```

```
END_ELSE/
```

Para inserir comandos Else / End Else:

1. Coloque o cursor na posição desejada da *janela Edição*. Observe que os blocos ELSE / END ELSE devem ser posicionados após um bloco IF / END IF ou ELSE IF / END ELSE IF.
2. Selecione **Else/End Else** a partir da barra de menu. As declarações OUTRO/FIM OUTRO aparecerão na janela Edição.

Amostra de código de Else / End Else

Considere o seguinte exemplo para perguntar ao usuário se deseja medir um elemento de ponto.



```
C1= COMENT/SIMNÃO,Gostaria de medir o elemento de ponto,
PNT1? Clicar em Não significa que o próximo elemento será
medido.
SE/C1.INPUT=="SIM"
    PNT1=ELEM/PONTO,CARTES
...
...
FIMMED/
```



```
FIM_SE/  
OUTRO  
    PNT2=ELEM/PONTO,CARTES  
    ...  
    ...  
    ENDMEAS/  
END_ELSE
```

Explicação do código da amostra

C1=COMENT/SIMNÃO

Essa linha obtém e armazena a resposta SIM ou NÃO do usuário.

IF/C1.ENTRADA=="SIM"

Essa linha é a expressão. Ela testa se a entrada do comentário 1 é um SIM. Se for um SIM, então a declaração IF será VERDADEIRA e continuará a executar as declarações depois da declaração IF, nesse caso, ela mede o elemento **PNT1**. Se for um NÃO, ela irá para a declaração END_IF.

END_IF

Essa linha finaliza a execução de comandos dentro do bloco de código IF / END IF. O PC-DMIS irá para qualquer comando depois dessa linha se o usuário clicar em **Não** no comentário.

ELSE

Se o bloco IF / END IF acima for avaliado como falso, as linhas de comandos que estiverem após essa linha e antes da linha END_ELSE serão executadas. Nesse caso, PNT2 é executado.

END_ELSE

Essa linha finaliza a execução de comandos dentro do segundo bloco de código ELSE / END_ELSE.

While / End While

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | While / End While** permite adicionar um loop condicional à rotina de medição. Os itens entre os comandos WHILE e END WHILE continuam executando em loop até que a

condição (ou expressão) que mantém o loop ativado não seja mais atingida, o que significa que a expressão do loop while é avaliada como FALSA (isto é, zero). O comando WHILE pode ser adicionado em qualquer local da rotina de medição. A expressão é testada no início de cada loop.

A linha de comandos da janela Edição de uma declaração WHILE / END WHILE é:

```
WHILE/expressão
```

```
END_WHILE/
```

Para inserir uma opção While / End While, siga estes passos:

1. Coloque o cursor na posição desejada da Janela Edição.
2. Selecione **While / End While** na barra de menu. A instrução WHILE / END WHILE aparecerá na janela Edição.

Amostra de código de While / End While

Considere o exemplo a seguir, que mede um elemento de acordo com o número de vezes especificado pelo usuário da rotina de medição.



```
C1=COMENT/ENTRADA,Quantas vezes gostaria de medir PNT1?
Digite somente um inteiro.
ATRIBUIR/CONTAGEM=0
WHILE/COUNT<C1.INPUT
    PNT2=ELEM/PONTO,CARTES
    ...
    ...
    ...
    FIMMED/
    ASSIGN/COUNT=COUNT+1
    COMENT/OPER,"Medido " + CONTAGEM + " do total de " +
    C1.INPUT+ " vezes."
END_WHILE/
```

Explicação do código da amostra

C1=COMENT/ENTRADA

Essa linha obtém e armazena a entrada de inteiro do usuário para a variável `C1.INPUT`.

ATRIBUIR/CONTAGEM=0

Essa linha inicializa `CONTAGEM`, uma variável definida pelo usuário, e atribui um valor inicial de 0. O código usa esta variável para contar o número de vezes que o PC-DMIS mede o elemento no interior do loop.

WHILE/COUNT<C1.INPUT

Essa linha é a expressão. Ela testa se o valor de `CONTAGEM` (inicialmente definido como 0) é menor que o inteiro selecionado pelo usuário. Se o teste for verdadeiro, as declarações depois de `WHILE/` e antes de `END_WHILE/` são executadas.

ASSIGN/COUNT=COUNT+1

Essa linha incrementa a variável `CONTAGEM` em um, de forma que saia eventualmente do loop quando não passar no teste de condição.

`COMENT/OPER,"Medido " + CONTAGEM + " do total de " + C1.INPUT+ " vezes."`

Essa linha exibe uma mensagem mostrando o número de vezes, fora do total, que o loop está sendo executado.

END_WHILE

Essa linha encerra a execução dos comandos dentro do bloco `WHILE / END WHILE`, desde que a condição seja falsa. Caso contrário, quando o PC-DMIS encontra esse comando, ele faz o loop novamente para a declaração `WHILE`.

Do / Until

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | Do / Until** permite adicionar um loop condicional à rotina de medição. Os itens entre os comandos `DO` e `UNTIL` continuam executando em loop até que a expressão do comando `UNTIL` seja VERDADEIRA (diferente de zero). Os comandos `DO / UNTIL` podem ser adicionados em qualquer local da rotina de medição. A expressão é testada no fim de cada loop.

A linha de comandos da janela Edição de uma declaração `DO / UNTIL` é:

DO/

UNTIL/ expressão

Para inserir comandos DO / UNTIL:

1. Coloque o cursor na posição desejada da Janela Edição.
2. Selecione **Do/Until** a partir da barra de menu. As declarações DO/UNTIL aparecerão na Janela Edição.

Amostra de código de Do / Until

Considere o exemplo a seguir, que mede um elemento de acordo com o número de vezes especificado pelo usuário da rotina de medição. Este exemplo é semelhante ao apresentado no tópico While / End While, com exceção de que o PC-DMIS testa a condição no final do loop em vez de testar no início.



```
C1= COMENT/ENTRADA,Digite o número de vezes que o PC-DMIS
deve medir o elemento PNT1:(digite somente um número
inteiro)
ATRIBUIR/CONTADOR=0
FAZER/
    PNT1=ELEM/PONTO, RETANG
    ...
    ...
    FIMMED/
    ATRIBUIR/CONTADOR=CONTADOR+1
    COMENT/OPER, "Medido"+CONTADOR+" de "+C1.INPUT+"
vezes."
ATÉ/CONTAGEM==C1.INPUT
```

Explicação do código da amostra

C1=COMENT/ENTRADA

Essa linha obtém e armazena a entrada de inteiro do usuário para a variável C1.INPUT.

ATRIBUIR/CONTAGEM=0

Desvio usando controle de fluxo

Essa linha inicializa `CONTAGEM`, uma variável definida pelo usuário, e atribui um valor inicial de 0. O código usa esta variável para contar o número de vezes que o PC-DMIS mede o elemento no interior do loop.

```
DO/
```

Inicia o loop `DO / UNTIL`. Todas as declarações são executadas pelo menos uma vez e o fluxo da rotina de medição sai do loop se a expressão for avaliada como falsa.

```
ATRIBUIR/CONTAGEM=CONTAGEM + 1
```

Essa linha incrementa a variável `CONTAGEM` em um, de forma que saia eventualmente do loop quando não passar no teste de condição.

```
COMENT/OPER,"Medido " + CONTAGEM + " do total de " + C1.INPUT +  
" vezes."
```

Essa linha exibe uma mensagem mostrando o número de vezes, fora do total, que o loop está sendo executado.

```
UNTIL/CONTAGEM == C1.INPUT
```

Essa linha finaliza a execução de comandos no loop `DO / UNTIL`, desde que a condição seja avaliada como falsa. Caso contrário, quando o PC-DMIS encontra esse comando, ele faz o loop novamente para a declaração `DO`.

Select / End Select

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | Select / End Select** permite a adição de um bloco condicional usado em conjunto com os pares `CASE / END CASE` e `Default Case / End Default Case`. A expressão do comando `Select` fornece dados que são comparados com a expressão das declarações `Case`. Se duas expressões forem avaliadas como a mesma coisa, as declarações no bloco `Case / End Case` serão executadas. O bloco `SELECT / END SELECT` cerca os conjuntos de blocos `CASE / END CASE` e `DEFAULT CASE / END DEFAULT CASE`.

A linha de comandos da janela Edição de uma declaração `SELECT / END SELECT` é:

```
SELECT/expressão
```

```
END_SELECT/
```

Para inserir os comandos `Select / End Select`:


1. Coloque o cursor na posição desejada da Janela Edição.
2. Selecione **Select/End Select** a partir da barra de menu. As declarações SELECT /END SELECT aparecerão na Janela Edição.

Amostra de código de Select / End Select

Os pares SELECT / END_SELECT, CASE / END_CASE, DEFAULT CASE / END_DEFAULT CASE trabalham todos juntos para avaliar múltiplas condições e fornecer uma ampla variedade de alternativas.

Suponha que tenha cinco círculos, rotulados CIR1 a CIR5, e deseja que o operador seja capaz de medir um círculo simplesmente pressionando uma tecla no teclado. Poderia ser usado um código semelhante a este:

Código completo



```
DO/
  C1=COMENT/ENTRADA,Digite um número para medir esse
  círculo:
  ,PARA CIR1 - Digite 1
  ,PARA CIR2 - Digite 2
  ,PARA CIR3 - Digite 3
  ,PARA CIR4 - Digite 4
  ,PARA CIR5 - Digite 5
  ,Qualquer outro caractere sai do loop
  SELECT/C1.INPUT
    CASE/1
      CIR1=ELEM/CÍRCULO
      ...
      ...
      FIMMED/
    END_CASE
  CASE/2
    CIR2=ELEM/CÍRCULO
    ...
    ...
    FIMMED/
  END_CASE
  CASE/3
    CIR3=ELEM/CÍRCULO
```

```
...
...
FIMMED/
END_CASE
CASE/4
    CIR4=ELEM/CÍRCULO
...
...
FIMMED/
END_CASE
CASE/5
    CIR5=ELEM/CÍRCULO
...
...
FIMMED/
END_CASE
DEFAULT CASE
    COMENT/OPER,Saindo do loop agora.
END_DEFAULT CASE
END_SELECT
UNTIL C1.INPUT < 1 OU C1.INPUT > 5
```

Explicação do código da amostra

SELECT/C1.INPUT

Essa linha de código obtém um número ou valor de sequência (nesse caso, um número) digitado pelo usuário e determina qual bloco `CASE/END_CASE` será executado a partir da entrada. Observe que o par `SELECT/END_SELECT` circunda toda a lista de código. Todos os pares `CASE/END_CASE` e `DEFAULT CASE/END_DEFAULT CASE` devem residir nessas duas linhas.

END_SELECT

Marca o fim do código mantido no par `SELECT / END SELECT`.

CASE/1 até CASE/5

Dependendo do valor de `C1.INPUT`, um dos blocos de código `CASE` é executado. Por exemplo, se `C1.INPUT` for avaliada como 1, o bloco de código `CASE 1` será executado, medindo `CIR1`. Se for avaliada como 2, o bloco de código `CASE 2` será executado, medindo `CIR2`, e assim por diante.

END_CASE

Essas linhas finalizam os blocos de código Case específicos.

DEFAULT CASE

Se o valor de `C1.INPUT` não corresponder a nenhuma das declarações CASE definidas (se o valor não for um número de um a cinco), o bloco de código **DEFAULT CASE** será executado. Nesse caso, ele exibe uma mensagem informando que está saindo do loop.

Observe como o loop `DO / UNTIL` circunda toda a amostra de código. Isso permite que o usuário continue a escolher no menu criado a partir da linha `COMENT/ENTRADA` até que selecione um caractere não reconhecido pelas declarações CASE.

Case / End Case

A opção de menu **Inserir | Comando de controle de fluxo | Pares de controle | Case / End Case** permite adicionar um bloco condicional à rotina de medição. Os itens entre os comandos CASE e END CASE serão executados se a expressão da declaração Case for avaliada com um valor igual ao da expressão do comando SELECIONAR correspondente. Caso contrário, o bloco de declarações é saltado. O bloco de declarações CASE / END CASE deve ser posicionado diretamente após um comando SELECIONAR ou um comando END CASE de um bloco CASE / END CASE anterior. Além disso, o PC-DMIS não pode comparar várias expressões em uma única declaração Case.

A linha de comando da janela Edição de uma declaração CASE / END CASE é:

`CASE/expressão`

`END_CASE`

Para inserir a opção Case / End Case:

1. Coloque o cursor na posição desejada da Janela Edição. Observe os requisitos de posição mencionados acima.
2. Selecione **Case / End Case** a partir da barra de menu. As declarações CASE / END CASE aparecem na janela Edição.

Default Case / End Default Case

A opção de menu Inserir | Comando de controle de fluxo | Pares de controle | Default Case / End Default Case permite adicionar um bloco condicional à rotina de medição. Os itens entre os comandos DEFAULT CASE e END DEFAULT CASE executam se todas as outras expressões dos blocos CASE / END CASE anteriores dentro do bloco SELECT / END SELECT correspondente forem avaliadas como falsas. Somente um bloco DEFAULT CASE / END DEFAULT CASE é permitido dentro de um bloco SELECT / END SELECT. O bloco DEFAULT CASE / END DEFAULT CASE deve ser posicionado após todos os blocos CASE / END CASE dentro do bloco SELECT / END SELECT.

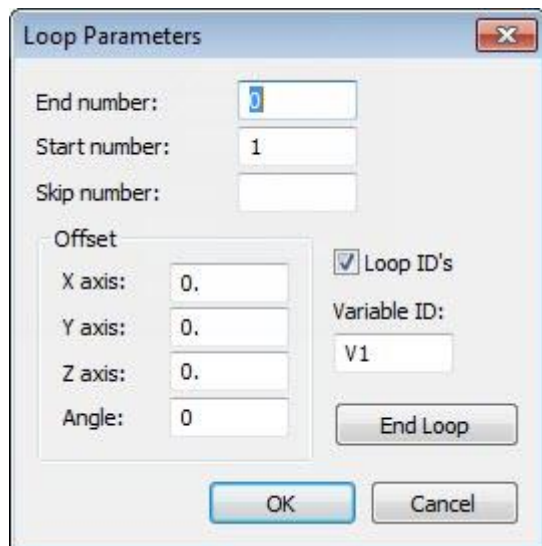
A linha de comandos da janela Edição de uma declaração DEFAULT CASE / END DEFAULT CASE é:

```
DEFAULT CASE/  
  
END_DEFAULT_CASE/
```

Para inserir comandos DEFAULT CASE/ END DEFAULT CASE:

1. Coloque o cursor na posição desejada na Janela Edição, observando as limitações de posição indicadas acima.
2. Selecione **Default Case/End Default Case** a partir da barra de menu. As declarações DEFAULT CASE / END DEFAULT CASE aparecem na janela Edição.

Uso de loops genéricos



Caixa de diálogo Parâmetros de execução em loop

A opção de menu **Inserir | Comando de controle de fluxo | Execução em loop** exibe a caixa de diálogo **Parâmetros da execução em loop**. Você pode usar essa caixa de diálogo para criar um comando **LOOP** que repete a rotina de medição (ou partes da rotina de medição) com ou sem qualquer dos deslocamentos. O comando **LOOP** pode ser adicionado em qualquer posição da rotina de medição, apesar de essa função ser mais útil no início e no fim da rotina de medição.

Número inicial - Esta caixa indica ao PC-DMIS o número da posição inicial em uma série de peças.



Suponha que tem 10 peças e você deseja iniciar com a posição número 5. Você inseriria 10 para o número total de peças e 5 para a posição inicial.

Número final - Esta caixa indica ao PC-DMIS quantas vezes executar o loop durante a rotina de medição. Esse número normalmente é o mesmo que o número de peças (ou padrões na peça) que o dispositivo de fixação prende, na direção X, Y ou Z. O PC-DMIS também pede o número da peça inicial (padrão).



Suponha que você tenha 10 peças na direção X e deseje iniciar com a posição número 5. Digite 10 na caixa **Número final**. Na caixa **Número inicial**, digite 5.

Ignorar número - Em um loop, o PC-DMIS repete uma rotina de medição o número de vezes indicado. A caixa **Ignorar número** permite ignorar um incremento específico. Observe que não é possível *ignorar o primeiro* incremento usando essa caixa, mas é possível alterar o **Número Inicial** para 2 e efetivamente ignorá-lo dessa maneira.



Suponha que deseja definir o parâmetro a ser ignorado a cada três incrementos do loop. Se o número três estiver indicado, o PC-DMIS mede a primeira e a segunda peça e pula para a quarta peça.

Deslocamento - A área deslocamento contém estas caixas:

Eixo X, Eixo Y, Eixo Z - Essas caixas definem as distâncias de deslocamento X, Y e Z entre as peças, ou padrões na mesma peça. Quando o loop é executado, esses valores deslocam a peça nas distâncias de deslocamento definidas.

Ângulo - Esta caixa configura o deslocamento angular entre as peças ou padrões na mesma peça. O primeiro deslocamento é baseado na origem da peça. O PC-DMIS desloca a peça no valor do ângulo quando o loop é executado.

O software baseia as distâncias de deslocamento na origem da peça. Ao executar o loop pela primeira vez, o PC-DMIS usa a origem da peça. Ao executar o loop pela segunda vez, ele move a peça conforme as distâncias de deslocamento. Ao executar o loop pela terceira vez, ele move a peça conforme as distâncias de deslocamento multiplicadas por dois, e assim por diante.



Se você tiver um comando de alinhamento dentro de um loop e o loop utilizar deslocamentos, você *tem que* definir todos os eixos para esse alinhamento. Além disso, o alinhamento dentro do loop deve usar os elementos medidos dentro do loop.

IDs do loop - Esta caixa de seleção determina se o PC-DMIS exibe o atual valor do loop em colchetes na ID do elemento.



Suponha que você tenha um elemento Círculo de CÍR1. Ele torna-se CÍR1[1] na primeira vez que o loop é executado, CÍR1[2] na segunda vez. e assim sucessivamente.

IDs de elementos em bancos de dados estatísticos

Se você marca a caixa de seleção **IDs de loop** e envia dados estatísticos a um banco de dados, em alguns casos o PC-DMIS pode não exibir esses IDs de loop no banco de dados.

Considere o seguinte:

Se houver um comando `ESTATS/LIG` e um comando `ESTATS/ATUALIZAR`, *dentro de um bloco de loop*, as IDs do loop não serão exibidas dentro do banco de dados.

Se houver um comando `ESTATS/LIG` *fora de um bloco de loop* e um comando `ESTATS/ATUALIZAR` *dentro de um bloco de loop*, as IDs do loop não serão exibidas dentro do banco de dados.

Se houver um comando `ESTATS/LIG` *fora de um bloco de loop* e um comando `ESTATS/ATUALIZAR`, *fora de um bloco de loop*, as IDs do loop serão exibidas dentro do banco de dados.

Se houver um comando `ESTATS/LIG` *dentro de um bloco de loop* e um comando `ESTATS/ATUALIZAR` *fora de um bloco de loop*, as IDs do loop serão exibidas dentro do banco de dados.

ID da variável - Esta caixa permite definir o nome da variável usada para rastrear a iteração atual do loop (ou loop atual dentro do número de loops especificados). Durante a execução da rotina de medição, essa variável é igual ao número atual da iteração do loop.

Finalizar - Este botão conclui o processo de loop. Você tem que seguir o comando `LOOP/INÍCIO` com o comando `LOOP/FIM` na janela Edição.

Usos da execução em loop

Existem três usos principais para a funcionalidade de execução em loop:

- Você tem um dispositivo de fixação de peças múltiplo que prende uma grade de peças. O dispositivo de fixação deve usar espaçamento consistente entre as

linhas. Os deslocamentos de translação / rotação permitem indexar de uma peça para a próxima da grade.

- Você tem um dispositivo de fixação que prende uma peça e deseja trocar por uma nova peça antes de cada loop da rotina de medição. Um comando [COMENTÁRIO](#) é útil para parar a CMM quando a peça está sendo substituída por uma nova. O comando pode estar no início ou no fim do loop.
- Você quer usar a funcionalidade de execução em loop para rotacionar a rotina de medição a fim de medir uma parte diferente da mesma peça. Por exemplo, você pode criar uma rotina de medição para medir um padrão de furo complicado que foi duplicado dez vezes na peça. A sua rotina de medição precisa medir somente um dos padrões de furo. Você pode usar a funcionalidade de execução em loop para deslocar a rotina de medição e medir as outras nove ocorrências do padrão.



Se você estiver usando um alinhamento dentro de um loop, o PC-DMIS permite usar o alinhamento ativo na linha de comando [ALINHAMENTO/ INICIAR](#) em vez de sempre recuperar um alinhamento armazenado anteriormente. Consulte o tópico "Uso de um alinhamento dentro de loops" no capítulo "Criação e uso de alinhamentos".

Criação de um loop

1. Selecione **Inserir | Comando de controle de fluxo | Execução em loop** na barra de menus para exibir a caixa de diálogo [Parâmetros da execução em loop](#).
2. Defina os valores na caixa de diálogo.
3. Selecione os parâmetros necessários (como **Número de peças**, **Número inicial**, **Ignorar número**, **Deslocamento angular**).
4. Coloque o cursor na posição da Janela Edição na qual deseja iniciar o loop.
5. Clique no botão **OK** para inserir o bloco de comando [LOOP/ INÍCIO](#).

A linha de comandos da janela Edição para a execução em loop é:



```
NOMEVAR = LOOP/INICIAR, ID = S/N, NÚMERO = 0, INICIAR
= 1, IGNORAR = , DESLOCAMENTO: EIXOX = 0, EIXOY = 0, EIXOZ
= 0, ÂNGULO = 0
```



Para concluir o procedimento de loop, você tem que finalizar o loop com um comando `LOOP/FIM`. O PC-DMIS faz o loop de quaisquer comandos da janela Edição englobados pelos comandos `LOOP/INÍCIO` e `LOOP/FIM`.

Você pode inserir o comando `LOOP/FIM` em uma das seguintes maneiras:

- Na janela Edição, digite `LOOP/FIM`.
- Selecione o item de menu **Inserir | Comando de controle de fluxo | Finalizar loop**.
- Na caixa de diálogo **Parâmetros de execução em loop**, clique no botão **Finalizar loop**.

Uso do comando Imprimir em um loop

Para usar o comando `IMPRIMIR` em um loop no PC-DMIS, faça o seguinte:

1. Insira os comandos `LOOP/INÍCIO` e `LOOP/FIM` na rotina de medição.
2. Insira o comando `IMPRIMIR` onde você deseja no meio dos comandos de loop.
3. Na janela Edição, clique no comando `IMPRIMIR`. Em seguida, pressione F9 para abrir a caixa de diálogo **Configurações de saída** para o comando `IMPRIMIR`. Para mais informações sobre as opções de saída e impressão da janela Relatório, consulte o tópico "Configuração de saída e de opções da impressora para a janela Relatório" no capítulo "Uso das opções de arquivo Basic".
4. Selecione uma das guias de formato de saída (**Relatório**, **DMIS** ou **Excel**) para definir o formato para o comando `IMPRIMIR`.
5. Selecione a caixa de seleção no canto superior esquerdo da página da guia para ativar as opções da saída.
6. Defina para onde o relatório gerado deve ser enviado:

- a. Clique no botão do navegador (...) à direita do caminho e na caixa do nome do arquivo para exibir a caixa de diálogo **Salvar como**.
 - b. Na caixa de diálogo **Salvar como**, defina o nome do arquivo e clique em **Salvar**.
7. Faça quaisquer outras alterações que você precise. Por exemplo, se deseja visualizar cada relatório depois de o PC-DMIS gerá-lo, marque a caixa de seleção **Mostrar relatório**.
8. Se deseja gerar saídas em diversos formatos, pressione a próxima guia e repita os passos 4 a 7.
9. Clique no botão **OK** para salvar as alterações que você fez no comando `IMPRIMIR`.

O comando `IMPRIMIR` funciona praticamente do mesmo jeito dentro e fora dos loops. A diferença é que internamente, o PC-DMIS monitora as saídas, e somente deixa sair dados obtidos após a execução do último comando `IMPRIMIR`.



Por exemplo, considere o pedaço de código abaixo. Quando o PC-DMIS executa o comando `IMPRIMIR` pela primeira vez, o software cria o primeiro arquivo de saída com *todos* os dados coletados até aquele ponto. Isso significa que o primeiro arquivo de saída inclui também dados coletados antes do comando `LOOP/INÍCIO`. A próxima que o comando `IMPRIMIR` é usado, o PC-DMIS somente executa o conteúdo dentro do loop. Isso significa que o segundo arquivo de dados somente contém os dados de dentro do loop. Isso é repetido até a rotina de medição ser concluída.

.
. .
.

`Início do loop`

```
CÍR1 = CÍRCULO (CONTATO)
```

```
CÍR2 = CÍRCULO (CONTATO)
```

```
Mover ponto
```

```
Mover ponto
```

```
Mover ponto
```

```
Mover ponto
CÍR1 = CÍRCULO (CONTATO)
Mover ponto
Mover ponto
Mover ponto
CÍR1 = CÍRCULO (CONTATO)
CÍR1 = CÍRCULO (CONTATO)
CÍR1 = CÍRCULO (CONTATO)
CÍR1 = CÍRCULO (CONTATO)
CÍR1 = CÍRCULO (CONTATO)
LOC1 aprovada: CÍR1
Relatório de impressão
```

```
Final do loop
```

```
.
.
.
```

Uso de rótulos

Você pode usar o comando **RÓTULO** com [IRPARA](#) ou um comando [SE IRPARA](#) para controlar onde o fluxo de execução é enviado, com base em determinadas condições. O nome do rótulo não pode ter espaços ou mais de 230 caracteres. O PC-DMIS exibe o nome rótulo usando letras maiúsculas.

Para criar um comando **RÓTULO**:

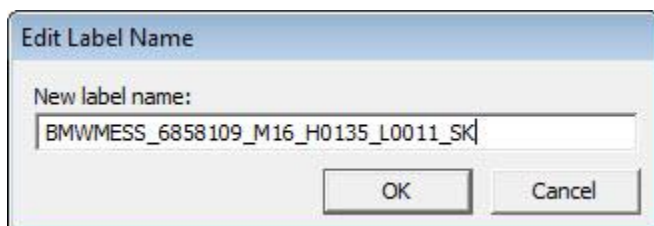
Digite o Comando.

1. Posicione o cursor em uma linha em branco na janela Edição.
2. Digite RÓTULO e pressione a tecla Tab.
3. No campo realçado, digite um novo nome de rótulo.

ou

Escolha um Comando.

1. Posicione o cursor na janela Edição.
2. Na barra de menu, selecione **Inserir | Comando de controle de fluxo | Rótulo** para acessar a caixa de diálogo **Editar nome do rótulo**.
3. Digite o nome do rótulo na caixa **Novo nome do rótulo**.



Caixa de diálogo Editar nome do rótulo

4. Clique em **OK** para inserir o nome do rótulo na próxima localização possível na janela Edição.

O comando RÓTULO no modo Comando da janela Edição é:

ID = RÓTULO/

Onde ID representa o nome especificado ao rótulo.

Para informações em como usar o comando [IRPARA](#) com rótulos, consulte o tópico "Ir para um rótulo usando IRPARA".

Para informações em como usar o comando [SE_IRPARA](#) com rótulos, consulte o tópico "Ir para um rótulo com base em condições".

Ir para um rótulo usando IRPARA

Você pode criar um comando IRPARA para dizer a sua rotina de medição para pular a um rótulo específico.

Para criar um comando IRPORA:

Digite o Comando.

1. Posicione o cursor em uma linha em branco na janela Edição.
2. Digite IRPORA e pressione a tecla Tab.
3. Digite o nome do rótulo para onde pular.

ou

Escolha um Comando.

1. Posicione o cursor na janela Edição.
2. Selecione a opção de menu **Inserir | Comando de controle de fluxo | IrPara** para abrir a caixa de diálogo **IrPara**.



Caixa de diálogo GoTo

3. Se os rótulos já existem na rotina de medição, eles aparecem na caixa **Rótulos atuais**.
4. Na caixa **Rótulos atuais**, selecione o rótulo desejado, ou digite o nome do rótulo na caixa **IrPara rótulo**.
5. Clique em **OK** para inserir o comando IRPORA na próxima localização possível na janela Edição.

O comando GOTO, no modo Comando da janela Edição, é:

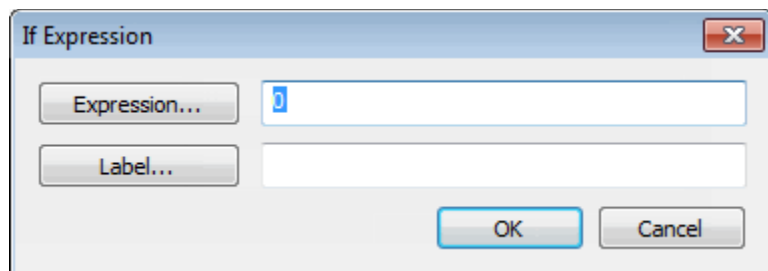
`GOTO/rótulo_ID`

Desvio usando controle de fluxo

Onde rótulo_ID representa o nome do rótulo específico. O nome do rótulo não pode ter espaços ou mais de 230 caracteres.



Se o rótulo ainda não tiver sido criado, o nome do rótulo aparece realçado em **vermelho** no modo Comando da janela Edição, e o comando GOTO é ignorado durante a execução.



Caixa de diálogo Expressão If

Com a opção **Inserir | Comando de controle de fluxo | If Goto**, você pode usar a caixa de diálogo **Expressão If** para criar declarações IF GOTO na sua rotina de medição. Quando a rotina de medição é executada e o PC-DMIS encontra uma declaração IF GOTO, o fluxo da rotina de medição se move para a identificação do rótulo se a expressão especificada for avaliada com um valor diferente de zero.

A linha de comandos da janela Edição de uma declaração IF_GOTO é:

```
IF_GOTO/expressão, GOTO=Rótulo
```

Expressão - Se você clica em **Expressão**, a expressão construtor abre. Com o construtor de expressões, é possível criar diversas opções diferentes que podem ser necessárias dentro da rotina de medição. Após você criar a expressão, ela aparece na caixa **Expressão**. Para obter informações sobre a criação de expressões, consulte o capítulo Uso de expressões e variáveis.

Rótulo - Se você clica em **Rótulo**, a caixa de diálogo **Goto** abre. A partir da caixa de diálogo **Goto**, você pode escolher para qual rótulo deseja que o fluxo da rotina se mova quando é encontrada a expressão definida. O rótulo escolhido aparece na caixa **Rótulo**. Ou se você souber o nome do rótulo, pode digitá-lo na caixa. Para mais informações sobre rótulos, veja "Salto para um rótulo".

Redefinir loop

O comando Redefinir loop redefine a árvore de execução em um loop. Você pode usar os pares de comando abaixo para adicionar o comando Redefinir loop e assim suportar várias medições de peças

- Enquanto / Fim enquanto
- Fazer/Até
- Loop/Fim loop

Você tem que usar o comando Redefinir loop dentro de um desses loops. O comando não tem efeito se você o coloca fora do loop. Para inserir um comando Redefinir loop, selecione a opção de menu **Inserir | Comando de controle de fluxo | Redefinir loop**. A linha de comando da janela Edição para um comando Redefinir loop, seria:

REDEFINIR LOOP/

Para várias medições de peça, você pode obter um relatório da última peça medida no loop. Para fazer isso, insira um comando Redefinir loop no início do loop, como mostrado nesta imagem:

```

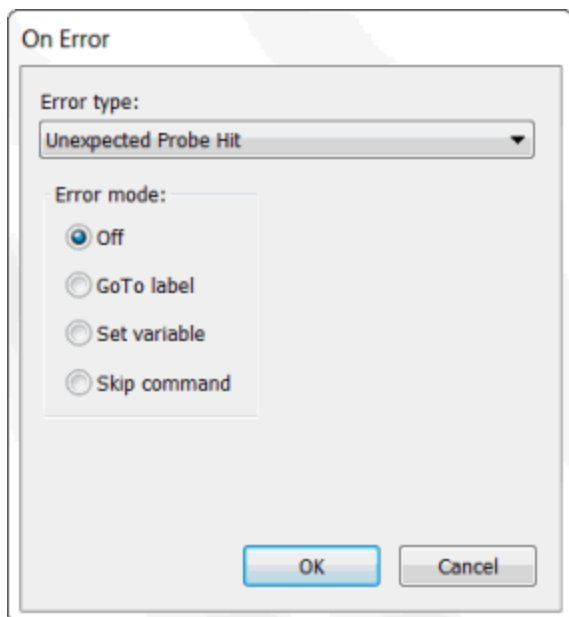
WHILE/COUNTER<=3
  REDEF LOOP/
  COMMENT/COPES,NO,FULL SCREEN=YES,AUTO-CONTINUE=YES,TIME DELAY=1,OVC=NO,
  Wait!
  F1
  <GCODE/C/CIRCLE,DEPENDENT,CARTESIAN,OUT,$
  NOM/XYI,<0,0,0>,$
  MEAS/XYI,<0,0,0>,$
  NOM/XYZ,<0,0,1>,$
  MEAS/XYZ,<0,0,1>,$
  DIAMETER/$,COUNTER
  DIM LOC1= LOCATION OF CIRCLE F1 UNITS=MM,$
  GRAPH=OFF TEXT=OFF MOLT=10.00 OUTPUT=BOTH HALF ANGLE=NO
  AX NOMINAL +TOL -TOL MEAS DEVI OUTTOL
  D 3.000 0.050 0.050 3.000 0.000 0.000 ----|----
  END OF DIMENSION LOC1
  PRINT/REPORT,EXEC MODE=END,$
  TO FILE=ON,AUTO=4, OUTPUT FORMAT=TXT,$
  BEST REPORT=YES,AUTO OPEN=OFF,$
  TO PRINTER=OFF,COPIES=1,$
  TO DMIS REPORT=ON,FILE OPTION=INDEX,FILENAME=C:\Users\Sudhanshu.Trivedi\
  REPORT THEORETICALS=NONE,REPORT FEATURE WITH DIMENSIONS=NO,$
  TO EXCEL=ON,AUTO=5,FILENAME=C:\Users\Sudhanshu.Trivedi\downloads\Whil
  PREVIOUS RUNS=DELETE,INSTANCES
  EXCEL FORM1 =EXCELFORSEREPORT/TEMPLATE=Form Row Wise, TYPE=EXCEL
  RESULT FOLDER=C:\Users\Public\Documents\Hexagon\PC-DMIS\2024.2\ExcelFor
  ASSIGN/COUNTER=COUNTER+1
  END WHILE/
  END OF MEASUREMENT FOR

```

O comando de redefinir loop remove todos os objetos da árvores de execução.

Desvio em um erro

A opção **Inserir | Comando Controle de fluxo | On Error** abre a caixa de diálogo **On Error**:



Caixa de diálogo Em erro

Você pode usar essa caixa de diálogo para dizer ao PC-DMIS a ação a tomar quando ocorre um erro da máquina.

Tipo de erro - O PC-DMIS rastreia esses tipos de condições de erro:

- Toque inesperado da sonda
- Falha de toque da sonda
- Refletor não encontrado - É usado com o rastreador no PC-DMIS Portable.
- Erro do laser - Se você está usando uma configuração de laser, consulte o tópico "Como resolver erros de sensor do laser com comando Em erro" na documentação do PC-DMIS Laser.
- Temperatura fora do limite - O comando Compensação de temperatura na rotina de medição aumenta esse erro se uma ou mais das temperaturas para a escala do eixo X, eixo Y ou eixo Z está acima do limite máximo ou abaixo do limite mínimo definido pelo comando Compensação de temperatura.



O comando On Error tem que ser colocado acima do comando Compensação da temperatura na rotina de medição.

- Calibração fora dos limites - O comando Verificar limites de calibração na rotina de medição indica esse erro se uma ou mais verificações de erro de calibração está fora dos limites. Para mais informações sobre o comando Verificar limites

de calibração, consulte "Sobre Verificar limites de calibração na documentação dos módulos do PC-DMIS Toolkit.



O comando Em erro tem que ser colocado acima do comando Verificar limites de calibração na rotina de medição.

Modo Erro - O PC-DMIS pode agir das seguintes maneiras, dependendo do tipo de erro:

- **Desativado** - O PC-DMIS não realiza nenhuma ação.
- **IrPara rótulo** - O fluxo da rotina de medição move-se para um rótulo definido (consulte "Uso de rótulos"). Essas opções ficam disponíveis:
 - **ID do rótulo** - Digite uma referência a um rótulo que ainda não existe.
 - **Rótulos atuais** - Lista todos os rótulos da rotina de medição.
- **Definir variável** - Define o valor da variável para um.
- **Ignorar comando** - O fluxo da rotina de medição salta sobre o comando atual e faz o movimento para o próximo comando marcado na rotina de medição.



Para mais detalhes sobre como o PC-DMIS registra no relatório as dimensões que usam elementos ignorados durante a execução, consulte o tópico "Adição em relatório de dimensão que usa um elemento ignorado durante a execução" nessa documentação.

Por padrão, todas as rotinas de medição são iniciadas com a ação para os dois tipos de erros definidas para **Desativado** (o PC-DMIS não faz nada). O modo de ação para cada tipo de erro pode ser alterado em qualquer lugar da rotina de medição.



Durante a execução, se o PC-DMIS encontrar um comando **ON ERROR/TOQUE INESPERADO / IR PARA RÓTULO**, quaisquer toques inesperados ocorridos após esse ponto da rotina de medição fazem com que a execução salte para o rótulo especificado. A ação de "definir valor da variável como um" faz com que a variável seja definida assim que o erro especificado ocorre. Em seguida, esse valor da variável pode ser testado usando uma declaração SE para fazer com que a execução salte para um novo ponto da rotina de medição.

Uso do comando On Error

1. Selecione **Inserir | Comando Controle de fluxo | On Error** para abrir a caixa de diálogo **On Error**.
2. Na lista **Tipo de erro**, selecione o tipo de erro como a condição para desvio.
3. Na área **Modo de erro**, selecione a opção que define a ação a ser executada quando a condição é atendida.
4. Para adicionar o comando On Error, clique em **OK**. Para fechar a caixa de diálogo sem aplicar quaisquer alterações, clique em **Cancelar**.

Interfaces suportadas On Error

Algumas interfaces suportam o comando On Error. Para ver se sua interface suporta o comando, consulte a tabela a seguir.

- Se a interface está na tabela, uma pequena caixa preta indica o tipo de erro que a interface suporta.
- Se a sua interface não está na tabela, não é possível usar o comando On Error.

Interfaces suportadas	Toque inesperado do sensor	Falha de toque do sensor	Refletor não encontrado
DEA	-	■	-
FDC	■	■	-
Cliente I++ DME	■	■	-
Johansson	■	■	-

Rastreador Leica	-	-	■
Leitz	■	■	-
LK Direct (também conhecida como LKRS232)	■	■	-
Driver LK	■	■	-
Mora	■	■	-
Sharpe	■	■	-
Sheffield	■	■	-
Wenzel	■	■	-
Zeiss	■	■	-

Desvio com subrotinas

Subrotinas são blocos de código na sua rotina de medição ou em uma rotina de medição externa que são normalmente referenciados repetidamente permitindo uma programação mais concisa. O PC-DMIS permite a passagem de "argumentos" (ou de variáveis locais) para a subrotina e a partir dela. Os tipos de argumentos que podem ser passados em uma subrotina são valores numéricos, variáveis, cadeias de texto e nomes de elementos.



Os blocos das sub-rotinas de comando são fechados dentro de comandos SUBROTINA e SUBFINAL.

Uma vez que você criou uma subrotina em sua rotina de medição, é possível "chamá-la" a partir da sua rotina de medição atual ou de outra rotina de medição, levando o fluxo da execução da rotina de medição para uma subrotina especificada e executando os comandos contidos no bloco de comando da subrotina. O fluxo da rotina de medição retornará para a declaração imediatamente após a declaração de chamada.



As sub-rotinas são chamados através da utilização do comando CALLSUB.

subrotinas externas

Subrotinas externas ou localizadas em uma rotina de medição fora da rotina de medição chamando não possuem acesso aos recursos, variáveis ou alinhamentos a partir da rotina de medição chamando. A subrotina continuará a ter acesso a itens dentro de sua própria rotina de medição. A rotina de medição externa e a rotina de medição chamando têm que usar as mesmas unidades de medida.

Subrotinas aninhadas

É possível guardar subrotinas dentro de outras subrotinas. A única limitação quanto ao número de argumentos e sub-rotinas aninhadas é a quantidade de memória disponível.

Criação de uma nova subrotina

Criação de uma subrotina digitando subrotina

É possível inserir esse comando digitando-se simplesmente SUB-ROTINA no modo Comando da Janela Edição e pressionando-se TAB. Uma vez inserido o comando, é necessário especificar o nome da sub-rotina e qualquer argumento que ela possua. Consulte a sintaxe da subrotina e os exemplos abaixo para obter essa informação.

Digite o comando FIMSUB e pressione TAB para finalizar o bloco de comando. Quaisquer comando da Janela Edição colocado dentro deste bloco de comando será considerado parte de uma subrotina e será executado quando a subrotina for chamada.

Criação de uma subrotina usando o item de menu Chamar subrotina

1. Selecione **Inserir | Comando de controle de fluxo | Subrotina** no submenu. Isto exibe a caixa de diálogo **Criação de subrotina**. Para obter informações sobre essa caixa de diálogo, consulte o tópico "Compreensão da caixa de diálogo Criação de subrotina".



Caixa de diálogo Criação de subrotina

2. Dê para a subrotina um nome, digitando-o na caixa **Nome**. O nome está limitado até 180 caracteres. Se introduzir 181 ou mais caracteres, aparece a mensagem de erro "Linha muito longa".

Quando clica no botão **OK**, o nome é automaticamente reduzido para os primeiros 180 caracteres e a caixa de diálogo **Criação de subrotina** fecha.

3. Se sua subrotina usar argumentos (colocar indicadores para informações passadas para a subrotina), inclua então uma a uma clicando o botão **Adicionar argumento**. A caixa de diálogo **Edição de argumento** aparece. Para obter informações sobre essa caixa de diálogo, consulte o tópico "Compreensão da caixa de diálogo Edição de argumento".



Caixa de diálogo Edição de argumento


4. Dê para o seu argumento um nome, digitando-o na caixa **Nome**.
5. Dê para o seu argumento um valor padrão, digitando-o na caixa **Valor**. A subrotina utilizará o valor padrão se nenhum valor for passado para a subrotina

a partir da declaração `CHAMARSUB`. Os argumentos válidos podem ser valores numéricos, variáveis, sequências de texto e nomes de elemento.

6. Se deseja dar uma descrição ao argumento, digite-a na caixa **Descrição**.
7. Clique em **OK** na caixa de diálogo **Editar argumento** para criar o argumento.
8. Repita as etapas 3 a 7 para cada argumento que deseja em sua subrotina.
9. Clique no botão **OK** na caixa de diálogo **Criação de subrotina** para encerrar a criação de sua subrotina. Esta subrotina aparecerá dentro da janela Edição com qualquer argumento definido.
10. Termine sua subrotina selecionando a opção de menu **Inserir | Comando de controle de fluxo | Finalizar subrotina**. Isto coloca um comando "`FIMSUB/`" na janela Edição, completando o bloqueio do comando de subrotina. Qualquer outro comando da rotina de medição que você desejar em sua subrotina deve ser incluído dentro do bloco de comando da subrotina, antes do comando `FIMSUB`.

Sintaxe para um bloco de comando da subrotina

A linha de comandos da Janela Edição de uma amostra de bloco de comando da subrotina mostraria:



```
SUBROTINA/<Nome>,  
<A1> = <Arg1> : <Descrição>,  
<A2> = <Arg2> : <Descrição>,  
=  
<Comandos>  
FIMSUB/
```

SUBROTINA é o comando usado para iniciar o bloco de comando de subrotina.

<Nome>= O nome da subrotina. O nome está limitado a 256 caracteres. Se você introduzir 257 ou mais caracteres, o nome é automaticamente reduzido para 256 caracteres.

<A1>= O primeiro argumento (ou variável de local) usado na subrotina. Geralmente, essa variável não pode ser acessada fora da subrotina.

<A2>= segundo argumento usado na subrotina. Geralmente, essa variável não pode ser acessada fora da subrotina. Argumentos adicionais podem ser incluídos conforme haja necessidade.

<Arg1> = o valor padrão do primeiro argumento.

<Arg2>= O valor padrão do segundo argumento.

<Descrição> = A descrição para o argumento.

<Comandos>= Outros comandos da janela Edição podem ser inseridos conforme haja necessidade após os argumentos e antes do comando FIMSUB.

FIMSUB/ é o comando usado para finalizar o bloco de comando da subrotina.

Exemplo de um bloco de comando da subrotina

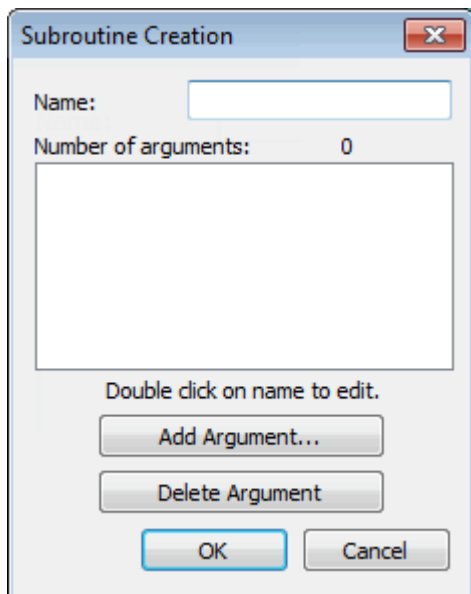
Por exemplo, uma subrotina finalizada que obtêm dados do operador e os exibe para que o relatório possa se parecer com o seguinte:



```
SUBROTINA/OBTER_INFO_OPERADOR
  NOMEOP = <Operador> : NOME DO OPERADOR,
  TURNO = <Turno> : HORÁRIO DO TURNO,
  =
  COMENT/RELAT,NOMEOP
  COMENT/RELAT,TURNO
FIMSUB/
```

Compreensão da caixa de diálogo Criação de sub-rotina

Para acessar a caixa de diálogo **Criação de subrotina**, selecione **Inserir | Comando de controle de fluxo | Subrotina**.



Caixa de diálogo Criação de subrotina

Os itens a seguir estão disponíveis na caixa de diálogo **Criação de subrotina**:

Nome - Esta caixa define a sua subrotina. Esse é o nome que deve ser usado quando você chama a subrotina mais tarde. Se você tem múltiplas subrotinas em uma única rotina de medição, cada nome tem que ser único.

O nome está limitado até 180 caracteres. Se introduzir 181 ou mais caracteres na caixa **Nome**, aparece a mensagem de erro "Linha muito longa".

Quando clica no botão **OK**, o nome é automaticamente reduzido para os primeiros 180 caracteres e a caixa de diálogo **Criação de subrotina** fecha.

Número de argumentos - Esta lista mostra os argumentos da subrotina que está sendo criada. Os argumentos aparecem nessa da seguinte forma:



Por exemplo, se um de seus argumentos é nomeado "Diâmetro" com o valor padrão 3, o argumento nesta lista pode aparecer como:



DIÂMETRO = 3 : O diâmetro do orifício

O PC-DMIS usa o valor padrão não importa qual outro valor não seja passado do comando **CHAMARSUB**.

Para editar um argumento, clique duas vezes sobre o argumento que deseja alterar. A caixa de diálogo **Editar argumento** abre, permitindo que efetue alterações. Para obter informações sobre essa caixa de diálogo, consulte o tópico "Compreensão da caixa de diálogo Edição de argumento".

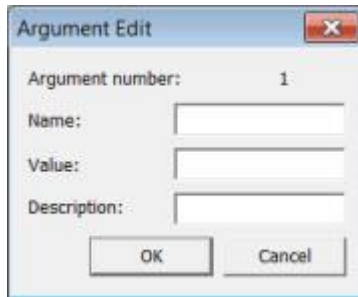
Adicionar argumento - Este botão adiciona novos argumentos na subrotina. Clique neste botão para abrir a caixa de diálogo **Edição de argumento**. Para mais informações sobre essa caixa de diálogo, consulte "Compreensão da caixa de diálogo Edição de argumento".

Excluir argumento - Este botão permite a exclusão de argumentos da subrotina. Selecione o argumento na lista e, em seguida, clique no botão **Excluir argumento** para removê-lo.

Edição de uma subrotina existente

Para editar uma sub-rotina existente, é necessário usar sempre o modo Comando da Janela Edição e editar a sub-rotina diretamente. Como alternativa, é possível acessar a caixa de diálogo **Criação de sub-rotina** colocando-se o cursor sobre a sub-rotina na Janela Edição e pressionando-se F9. Isso abre a caixa de diálogo **Criação de sub-rotina**. Para mais informações sobre esta caixa de diálogo, consulte o tópico "Compreensão da caixa de diálogo Criação de sub-rotina".

Compreensão da caixa de diálogo Edição de argumento



Caixa de diálogo Edição de argumento

A caixa de diálogo **Edição de argumento** aparece sempre que você escolhe criar ou editar um argumento dentro da caixa de diálogo **Criação de subrotina (Inserir | Comando de controle de fluxo | Subrotina)** ou da caixa de diálogo **Chamar subrotina (Inserir | Comando de controle de fluxo | Chamar subrotina)**.

A caixa de diálogo **Edição de argumento** pode ser usada nestes dois contextos:

- Para definir argumentos de subrotina e seus valores padrão em um bloco de comando [SUBROTINA](#).
- Para definir os valores que serão passados para a subrotina de um comando [CHAMARSUB](#).

As opções a seguir estão disponíveis na caixa de diálogo **Edição de argumento**:

Nome - Esta caixa define o nome do argumento que está sendo criado ou editado.

Valor - Esta caixa define o valor do argumento.

Se estiver criando ou editando o comando [SUBROTINA](#), este será o valor padrão usado se nenhum valor for passado para a subrotina pela declaração [CHAMARSUB](#).

Se estiver editando ou criando o comando [CHAMARSUB](#), este será o valor passado para a subrotina.

Os valores válidos podem ser:

- Numérico
- Variável

- Sequência de texto - As sequências de texto devem estar fechadas por aspas duplas.
- Nome do elemento - O nome do elemento deve estar fechado pelo sinal de chaves, como por exemplo {F1}.

Descrição - Define a descrição do argumento para a subrotina. Essa descrição aparece ao lado do argumento na janela Edição do bloco de comando da [SUBROTINA](#).

Chamada de uma subrotina

Para chamar uma subrotina, é necessário inserir um comando [CHAMARSUB](#) em sua rotina de medição para chamar uma subrotina existente a partir da rotina de medição atual ou uma subrotina a partir de uma rotina de medição externa.



Quando um comando [CHAMARSUB](#) é executado, o PC-DMIS executa todos os comandos nas subrotinas. Isso até inclui mesmo comandos desmarcados.

A subrotina recuperada tem que ter sido criada em uma versão válida do PC-DMIS (versões 2020 R2 a 2026.1).

Chamar uma subrotina digitando CHAMARSUB

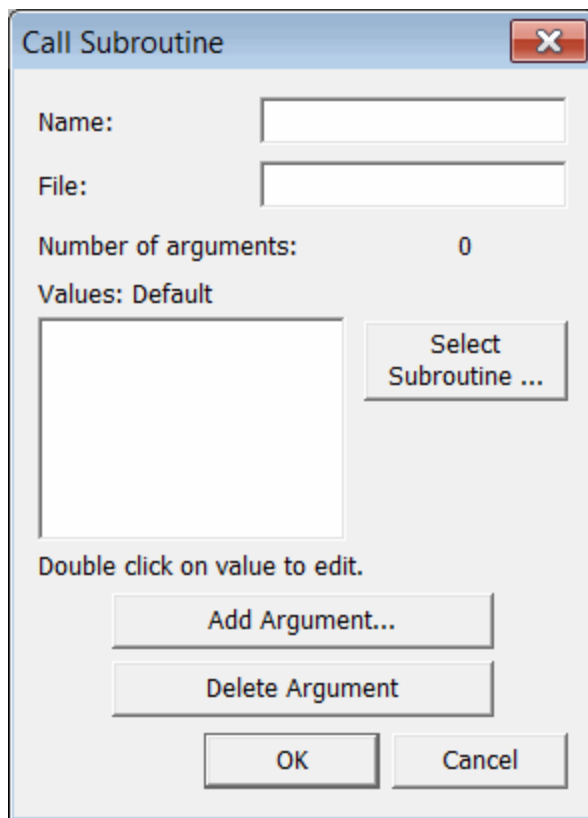
É possível inserir esse comando digitando [CHAMARSUB](#) na janela Edição e então pressionar a tecla Tab no teclado onde você deseja que o comando apareça na janela Edição.

Uma vez que o comando seja inserido, será necessário especificar o nome da subrotina sua localização, se é uma rotina de medição bem como qualquer valor a ser passado para argumentos disponíveis. Consulte "Passagem de argumentos para uma subrotina" para exemplos de passagem de argumentos.

Chamada de subrotina usando o item de menu Chamar subrotina

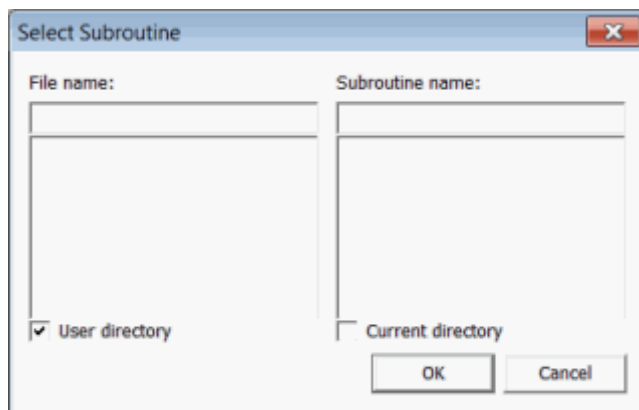
1. Selecione a opção **Inserir | Comando de controle de fluxo | Chamar subrotina** no submenu. A caixa de diálogo *Chamar subrotina* é aberta. Consulte "Compreensão da caixa de diálogo Chamar subrotina" para obter informações sobre essa caixa de diálogo.

Desvio usando controle de fluxo



Caixa de diálogo Chamar sub-rotina

2. Clique no botão **Selecionar subrotina**. A caixa de diálogo *Selecionar subrotina* é aberta.



Caixa de diálogo Selecionar subrotina


3. Selecione a caixa de seleção **Diretório do usuário**, a caixa de seleção **Diretório atual** ou ambas. Se a rotina de medição da subrotina residir no diretório especificado para procurar subrotinas, selecione a caixa de seleção

Diretório do usuário. Se for do diretório atual, marque a caixa de seleção **Diretório atual**. O PC-DMIS lista todas as rotina de medição disponíveis para seleção.

4. Selecione a rotina de medição que contém a subrotina desejada. Todas as subrotinas associadas com a rotina de medição selecionada aparecem na caixa **Nome da subrotina**.
5. Selecione a subrotina que deseja chamar.
6. Clique no botão **OK**. As informações sobre a subrotina que será chamada aparecem nas caixas **Nome** e **Arquivo** da caixa de diálogo **Chamar subrotina**.
7. Se deseja passar informações para a subrotina, clique no botão **Adicionar argumento** e use o caixa de diálogo **Editar argumento** para definir argumentos e valores para passagem. Consulte "Compreensão da caixa de diálogo Edição de argumento" para informações sobre esta caixa de diálogo. Consulte "Passagem de argumentos para uma subrotina" para exemplos de passagem de argumentos.
8. Clique novamente no botão **OK**. O PC-DMIS adiciona o comando `CHAMARSUB` ao local selecionado da janela Edição.

A sintaxe do comando **CHAMARSUB**

A sintaxe da linha de comandos da Janela Edição para uma subrotina chamada mostraria:



```
CS1 =CHAMARSUB/<Nome>, <Arquivo>:<Arg1>,<Arg2>,
```

CS1 = a ID do rótulo dada ao comando `CHAMARSUB`.

<Nome> = o nome da subrotina a ser chamada.

<Arquivo> = o caminho completo para a rotina de medição que contenha a subrotina para chamar. Se esse campo está vazio, o PC-DMIS procura a subrotina na rotina de medição atual.

<Arg1> = o valor a ser passado para o primeiro argumento na subrotina. Se esse campo está vazio, o PC-DMIS usa o valor padrão definido para o primeiro argumento na subrotina.

<Arg2> = o valor a ser passado para o segundo argumento na subrotina. Se esse campo está vazio, o PC-DMIS usa o valor padrão definido para o segundo argumento na subrotina. Essa amostra de sintaxe mostra apenas dois argumentos. Se necessário, você pode passar outros argumentos para a subrotina.



No comando `CHAMARSUB`, você deve manter um conjunto de ponteiros para todos os objetos feitos para a sub-rotina, de forma que possa se referir a eles posteriormente usando a Ident. da sub-rotina. Para obter mais informações sobre apontadores, consulte "Apontadores" no capítulo "Uso de expressões e variáveis".

Exemplo do Comando CHAMARSUB

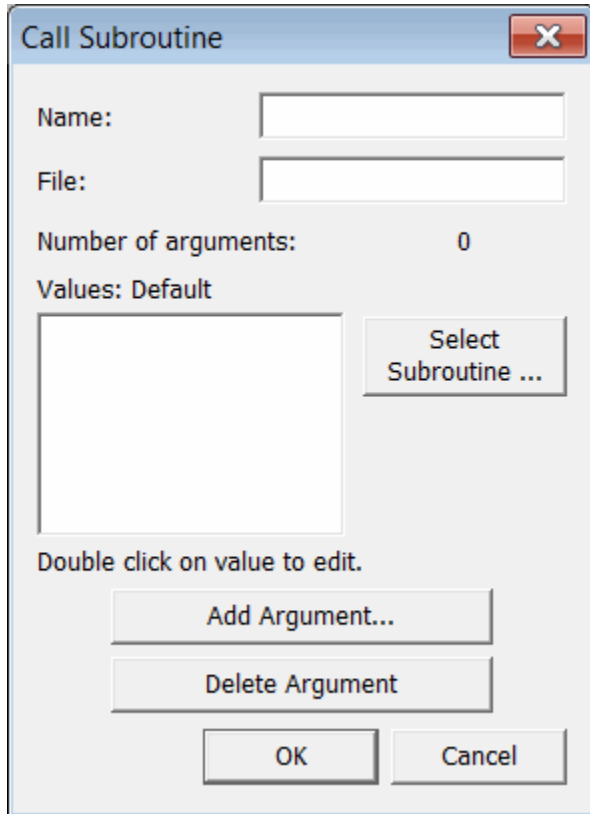


```
CS1=CHAMARSUB/OBTERR_INFO_OPERADOR,D:\ROTINASMEDIÇÃO\V42TESTESUBROUTINA.PRG:V1,V2,,
```

Neste exemplo do comando `CHAMARSUB`, CS1 chama uma subrotina nomeada `OBTERR_INFORMAÇÃO_OPERADOR` localizada dentro da rotina de medição `V42SUBROUTINETEXT.PRG` localizada no diretório `D:\MEASROUTINES\`.

Haverá a passagem de dois valores - neste caso, as variáveis `V1` e `V2` - para a subrotina.

Compreensão da caixa de diálogo Chamar sub-rotina



Caixa de diálogo Chamar sub-rotina

Os itens a seguir descrevem as várias opções disponíveis na caixa de diálogo **Chamar sub-rotina**.

Caixa **Nome**

A caixa **Nome** contém o nome da sub-rotina selecionada após o uso do botão **Selecionar subrotina....**

O nome está limitado até 180 caracteres. Se introduzir 181 ou mais caracteres na caixa **Nome**, aparece a mensagem de erro "Linha muito longa".

Quando clica no botão **OK**, o nome é automaticamente reduzido para os primeiros 180 caracteres e a caixa de diálogo **Chamar subrotina** fecha.

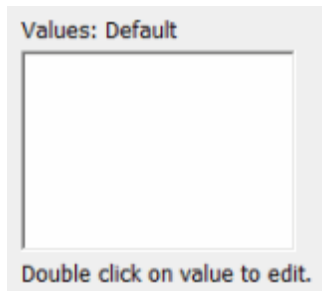
Caixa **Arquivo**

Desvio usando controle de fluxo



A caixa **Arquivo** contém o caminho do diretório para o arquivo da subrotina chamada.

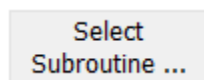
Caixa **Valores**



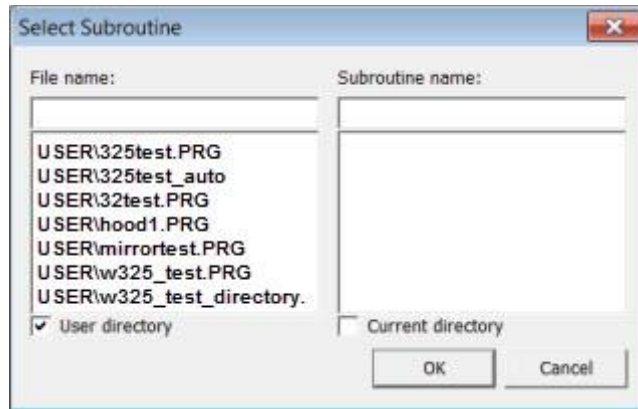
A caixa **Valores** contém uma lista dos valores de cada argumento associado com a subrotina. Esses valores serão passados para a subrotina quando a esta for executada.

Para alterar esses valores, clique duas vezes no valor que deseja alterar. A caixa de diálogo **Edição de argumento** aparece. Para obter informações sobre essa caixa de diálogo, consulte o tópico "Compreensão da caixa de diálogo Edição de argumento".

Botão **Selecionar sub-rotina**



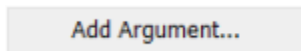
O botão **Selecionar subrotina** exibe a caixa de diálogo **Selecionar subrotina**.



Caixa de diálogo Selecionar subrotina

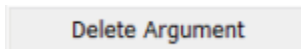
Essa caixa de diálogo permite chamar sub-rotinas criadas anteriormente, procurando no diretório do usuário ou no diretório ativo. Consulte "Chamada de subrotina" para informações sobre a caixa de diálogo **Selecionar subrotina**.

Botão **Adicionar argumento**



Esse botão define um valor a ser passado aos argumentos para a sub-rotina.

Botão **Excluir argumento**



Esse botão permite excluir os argumentos da caixa **Valores**. Selecione o valor exibido e, em seguida, clique no botão **Excluir argumento**. O argumento associado ao valor é excluído.

Para adicionar um novo argumento usando a caixa de diálogo Chamar sub-rotina:

Para incluir um novo argumento ao seu comando [CHAMARSUB](#) para passagem até a subrotina:

1. Dentro da caixa de diálogo **Chamar subrotina**, clique no botão **Adicionar argumento** para abrir a caixa de diálogo **Edição de argumento**.
2. Clique na caixa **Valor**.

Desvio usando controle de fluxo

3. Na caixa **Valor**, digite o valor do argumento.
4. Clique em **OK**.

Para editar argumentos existentes usando a caixa de diálogo Chamar subrotina:

Para editar um argumento existente dentro do seu comando [CHAMARSUB](#):

1. Na caixa de diálogo **Chamar subrotina**, dê um clique duplo no valor do argumento que deseja alterar. Aparece uma caixa de valor mostrando o valor padrão da subrotina chamada.
2. Digite um novo valor.
3. Clique no botão **OK**.

Consulte "Criação de uma subrotina" e "Edição de uma subrotina existente" para obter informações adicionais sobre como editar ou criar novos argumentos para uma subrotina.

Para excluir argumentos de um comando Chamar subrotina:

1. Coloque o cursor no comando [CHAMARSUB](#).
2. Pressione F9 para acessar a caixa de diálogo **Chamar subrotina**.
3. A partir da lista de argumentos, selecione um ou mais argumentos.
4. Clique no botão **Excluir argumento**.
5. Clique em **OK**.

Um argumento também pode ser excluído diretamente no texto da Janela Edição. Para isso:

1. Coloque o PC-DMIS no Modo Comando.
2. Coloque o cursor sobre o comando [CHAMARSUB](#) e pressione TAB até realçar o argumento desejado.
3. Digite as letras "del" para excluir o argumento. Observe que pressionar as teclas Delete ou Backspace não exclui realmente o argumento, elas simplesmente mudam o argumento para um argumento vazio.

Uso de instruções CHAMARSUB no modo Vários braços

Se é atribuída uma instrução CHAMARSUB para o Braço1, todos os comandos na subrotina são atribuídos ao Braço1 quando a subrotina é chamada.

Se é atribuída uma instrução CHAMARSUB para o Braço2, todos os comandos na subrotina são atribuídos ao Braço2 quando a subrotina é chamada.

Se marcar uma declaração CHAMARSUB para ambos os braços, o PC-DMIS deixará as marcações de subrotina da forma como foram originalmente configuradas.

Se uma subrotina contiver um comando MOVER/SINC e atribuir a declaração CHAMARSUB para o Braço1 ou Braço2, no momento da execução, o PC-DMIS exibirá um erro indicando que isso é inválido e a subrotina não será chamada.

Para obter informações sobre a atribuição de um comando a ser executado em um braço específico, consulte o tópico "Atribuição de um comando a um braço" no capítulo "Uso do modo Múltiplos braços".

Exemplos de subrotina

Considere as informações nos tópicos a seguir para obter alguns exemplos de passagem de argumentos e de subrotinas de um modo geral.

- Passagem de argumentos para uma subrotina
- Amostra de código de uma subrotina

Passagem de argumentos para uma subrotina

Os tipos de argumentos que você pode passar para uma sub-rotina são valores numéricos, variáveis, cadeias de texto e nomes de elementos. Para passar valores para argumentos, digite o valor dentro da caixa **Valor** da caixa de diálogo **Editar argumento**, ou diretamente em um comando CHAMARSUB inserido no modo Comando.

Para mais detalhes, veja "Criação de uma nova subrotina" e "Compreensão da caixa de diálogo Edição de argumento".


Passagem de variáveis para uma subrotina

Os argumentos que podem retornar dados são variáveis. Quando você usa uma variável como argumento para uma subrotina, todas as alterações ocorridas na variável correspondente da subrotina são retornadas e tornam-se o valor da variável que foi passada.



Exemplo de variáveis de passagem:

Este exemplo mostra como o valor de uma variável, usada como argumento de uma subrotina, é modificada e passada de volta para uma subrotina:



```
CS4      ATRIBUIR/V1=6
          =CHAMARSUB/MINHASUB, :V1,,
          SUBROTINA/MINHASUB,
          A1 = 0 : PRIMEIRO ARGUMENTO,
          =
          ATRIBUIR/A1=A1+1          COMENT/OPER,
NÃO,TELA INTEIRA=NÃO,AUTOCONTINUAR=NÃO,OVC=NÃO,
          O valor de A1 é:
          A1
          FIMSUB/
```

Explicação do código com fluxo de execução

Primeiro, a variável V1 recebe um valor de 6.

A declaração `CHAMARSUB` chama então a subrotina e passa a variável V1 como o primeiro argumento.

A declaração `SUBROTINA/MINHASUB` define MINHASUB como uma subrotina.

A1 é o nome do primeiro argumento, portanto, quando é feita a chamada, A1 tem inicialmente o mesmo valor que V1 tem no momento da chamada, 6.

A subrotina executa então suas duas declarações:

- A primeira declaração é um comando `ATRIBUIR` que incrementa o valor de A1 para 7.
- A segunda declaração é um comando `COMENT/OPER` que mostra o valor de A1 atualizado.

A subrotina é então finalizada com o comando `FIMSUB/`.

Em seguida, o fluxo de execução retorna para a declaração diretamente após o comando `CHAMARSUB`. Há um outro comando `COMENT/OPER` que mostra o valor final de V1.



Quando a execução retorna, todas as variáveis usadas como argumentos, V1 neste caso, são atualizadas para o valor das variáveis correspondentes na sub-rotina, A1 neste caso. Portanto, após retornar da subrotina, o V1 tem agora tem o valor de 7. O valor foi passado de volta a partir da subrotina.

Passagem de valores de número para uma subrotina

Os argumentos também podem ser utilizados como caracteres numéricos.



Exemplo de passagem dos valores de número:

Este exemplo mostra como passar valores de número para uma subrotina. Ele passa até dois números e então os inclui juntos.



```
CS1      =CHAMARSUB/SOMA_NÚMEROS, :,
CS2      =CHAMARSUB/SOMA_NÚMEROS, :5, 10, ,
          SUBROTINA/SOMA_NÚMEROS,
          NÚM1 = 1 : PRIMEIRO NÚMERO,
          NÚM2 = 1 : SEGUNDO NÚMERO,
          =
          ATRIBUIR/TOTAL=NÚM1+NÚM2
          COMENT/OPER,NÃO,TELA INTEIRA=NÃO,
          AUTOCONTINUAR=NÃO,OVC=NÃO,
          NÚM1 + "+" + NÚM2 + "=" + TOTAL
          FIMSUB/
```

No primeiro comando `CHAMARSUB` (CS1), nenhum valor de número é passado para a subrotina. Os números padrão, 1 para NÚM1 e 1 para NÚM2 são usados em substituição, e a soma total é 2.

No segundo comando `CHAMARSUB` (CS2), são passados dois valores, 5 e 10. Portanto, NÚM1 é 5 e NÚM2 é 10, e a soma total é 15.

Passagem de seqüências de texto para uma subrotina

Os argumentos também podem empregar caracteres numéricos. Para efetuar a passagem de uma seqüência de texto, verifique se os caracteres alfanuméricos são colocados dentro de aspas duplas.



Exemplo de passagem de sequências de texto:

Este exemplo mostra como passar valores de seqüência para uma subrotina. Ele passa dois valores de seqüência para dois parâmetros e então os exibe no relatório:



```
CS1          =CHAMARSUB/OBTÉR_INFO_OPERADOR, : "BOB
JONES", "MANHÃ", , ,
              SUBROTINA/OBTÉR_INFO_OPERADOR,
              NOMEOP = "<NENHUM OPERADOR PASSOU>"
: NOME DO OPERADOR,
              TURNO = "<NENHUM TURNO PASSOU>" :
HORÁRIO TURNO,
              =
              COMENT/RELAT,
              NOMEOP
              COMENT/RELAT,
              TURNO
              FIMSUB/
```

O primeiro argumento, `OPNOME`, recebe o valor passado de "BOB JONES" e o segundo argumento, `DESLOCAMENTO`, recebe "MANHÃ". Os dois comandos `COMENTÁRIO/RELAT` passam, então, as sequências para o relatório de inspeção.

Passagem de nomes de elemento para uma subrotina

Você pode passar os nomes do elemento entre chaves. Por exemplo, no comando `CHAMARSUB`, se você usar {PNT1} como um parâmetro, o software passa PNT1 para a subrotina. Quando você passa o nome do elemento desta maneira, a subrotina tem acesso total ao elemento.



Exemplo da passagem de nomes de elemento:

Este exemplo passa o nome do elemento PNT1 para a subrotina, dando para sua subrotina acesso completo ao elemento. Se nenhum valor for passado, o nome do elemento padrão F1 será usado. Esta subrotina realizará uma consulta em busca de um novo valor e alterará o valor de X teórico para o elemento.



```
CS1          =CHAMARSUB/MUDAR_TEÓR_X, : {PNT1}, , ,
C1           =COMENT/ENTRADA, NÃO, TELA INTEIRA=NÃO,
```

```

Elemento passado:
ELEM1
O valor teórico de X atual é:
ELEM1.TX
Digite um novo valor teórico de X:
ATRIBUIR/ELEM1.TX=C1.INPUT
COMENT/OPER,NÃO,TELA INTEIRA=NÃO,AUTO-
CONTINUAR=NÃO,OVC=NÃO,
"Elemento atualizado para " + ELEM1.TX

```

Devido ao fato da subrotina possuir acesso aos elementos passados, a declaração `ATRIBUIR/ELEM1.TX = C1.INPUT ?>` modifica o valor do X teórico do elemento PNT1 real. PNT1 terá agora seu valor de X teórico permanentemente alterado.

Amostra de código de uma subrotina

A amostra de código a seguir permite que o operador tenha a opção de alterar os valores teóricos X, Y e Z de um elemento após sua medida. As execuções subsequentes usam os valores teóricos atualizados.

```

PNT1      =GENÉRICO/PONTO,DEPENDENTE,CARTESIANO,$
          NOM/XYZ,<5,10,15>,$
          MED/XYZ,<7,12,17>,$
          NOM/IJK,<0,0,1>,$
          MEAS/IJK,<0,0,1>
C1         =COMENT/SIMNÃO,NÃO,TELA INTEIRA=NÃO,AUTO-
CONTINUAR=NÃO,
          Deseja mudar os valores teóricos para o PNT1?
          SE/C1.INPUT=="SIM"
CS1        =CHAMARSUB/MUDARTEÓR,:,
          FIM_SE/
          COMENT/OPER,NÃO,TELA INTEIRA=NÃO,AUTO-
CONTINUAR=NÃO,
          Os valores XYZ teóricos e reais para o PNT1
          são:
          "Teór X= "+PNT1.TX
          "Teór Y= "+PNT1.TY
          "Teór Z= "+PNT1.TZ
          -----
          "Real X= "+PNT1.X
          "Real Y= "+PNT1.Y

```

```
        "Real Z= "+PNT1.Z
        ROTINA/FIM
        SUBROTINA/MUDARTEÓR,
            PONTO1={ PNT1 } : ,
            =
            INFODIM/PNT1;ICON,IDDIM,IDELEM,VERT,HORIZ,, $
            CABEÇALHOS,;MED,,,,,,,,,
C2      =COMENT/ENTRADA,NÃO,TELA INTEIRA=NÃO,
        Digite o novo valor X teórico para o PNT1.
        "Seu valor atual é "+PNT1.TX
        ATRIBUIR/PNT1.TX=C2.INPUT
C3      =COMENT/ENTRADA,NÃO,TELA INTEIRA=NÃO,
        Digite o novo valor teórico de Y para O PNT1.
        "Seu valor atual é "+PNT1.TY
        ATRIBUIR/PNT1.TY=C3.INPUT
C4      =COMENT/ENTRADA,NÃO,TELA INTEIRA=NÃO,
        Digite o novo valor teórico de Z para O PNT1.
        "Seu valor atual é "+PNT1.TZ
        ATRIBUIR/PNT1.TZ=C4.INPUT
        FIMSUB/
```

Explicação do código da amostra

C1=COMENT/SIMNÃO

Essa linha obtém e armazena a resposta SIM ou NÃO do usuário.

IF/C1.ENTRADA=="SIM"

Essa linha é a expressão. Ela testa se a entrada do comentário 1 é um SIM. Se for um SIM, então a declaração IF será VERDADEIRA e continuará a executar as declarações depois da declaração IF, nesse caso, ela mede o elemento `PNT1`. Se for um NÃO, ela irá para a declaração `END_IF`.

CS1=CHAMARSUB/ALTERARTEÓR, : ,

Essa linha invoca a sub-rotina chamada `ALTERARTEÓR`. O fluxo da rotina de medição salta agora para a linha `SUBROTINA/ALTERARTEÓR`.

SUB-ROTINA /ALTERARTEÓR,

Essa linha inicia a sub-rotina `ALTERARTEÓR`. O fluxo da rotina de medição continua com a execução do código entre essa linha e a linha `FIMSUB/`.

PONTO1={ PNT1 } : ,

Esse é o único argumento da sub-rotina. Ele permite que a subrotina acesse informações do elemento PNT1.

C2=COMENT/ENTRADA, C3=COMENT/ENTRADA, C4=COMENT/ENTRADA

Todos esses comentários de entrada obtêm os novos valores teóricos de X, Y e Z do usuário e os armazenam em C2.INPUT, C3.INPUT e C4.INPUT, respectivamente.

ATRIBUIR/PNT1.TX=C2.INPUT

Essa linha obtém o valor teórico de X de C2.ENTRADA e o atribui à variável **PNT1.TX**. **PNT1.TX** é uma variável do PC-DMIS que mantém o valor teórico de X (indicado por TX) para o ponto com o rótulo de ID PNT1.

ATRIBUIR/PNT1.TY=C3.INPUT

Essa linha obtém o valor teórico de Y de C3.ENTRADA e o atribui à variável **PNT1.TY**. **PNT1.TY** é uma variável do PC-DMIS que detém o valor Y teórico (denotado por TY) para o ponto com o rótulo de ID de PNT1.

ATRIBUIR/PNT1.TZ=C4.INPUT

Essa linha obtém o valor teórico de Z de C4.ENTRADA e o atribui à variável **PNT1.TZ**. **PNT1.TZ** é uma variável do PC-DMIS que detém o valor Z teórico (denotado por TZ) para o ponto com o rótulo de ID de PNT1.

FIMSUB/

Essa linha encerra a sub-rotina e o fluxo da rotina de medição retorna à linha imediatamente após a invocação da sub-rotina. Nesse caso, a declaração **FIM_SE/**. O fluxo da rotina de medição continua com o próximo comentário do operador que exibe os valores teóricos e reais de X, Y e Z e, em seguida, a rotina de medição finaliza com o comando **ROTINA/FIM**.

Comandos do modo Comando após comentários



Após inserir um comentário do PC-DMIS, para digitar comandos adicionais do PC-DMIS no modo Comando, primeiro pressione *duas vezes* o Enter após o comando **COMENTÁRIO**. Isso informa ao PC-DMIS que você não quer mais adicionar texto ao comentário, mas deseja adicionar um novo comando.

Finalização de uma rotina de medição

O item de menu **Inserir | Comando de controle de fluxo | Finalizar rotina** insere um comando **ROTINA/FIM** na janela Edição. Sempre que o PC-DMIS encontra esse comando durante a execução da rotina, ele interrompe imediatamente a execução da rotina de medição.

Esse comando é útil se desejar parar uma rotina de medição antes do usual, com base em condições definidas.