

## Conceitos e Convenções

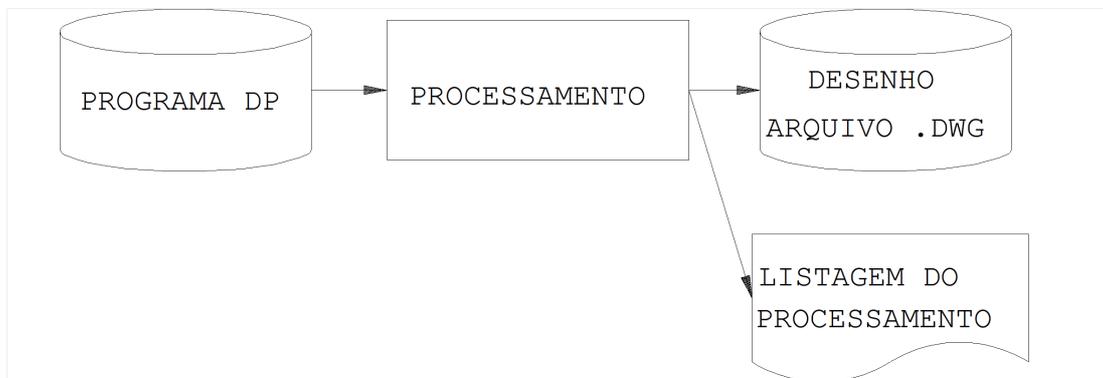
Neste capítulo descreveremos alguns conceitos do DP e as convenções adotadas no manual para a descrição da linguagem.

### Visão geral

Vamos mostrar o esquema de processamento de arquivos DP usando a notação:

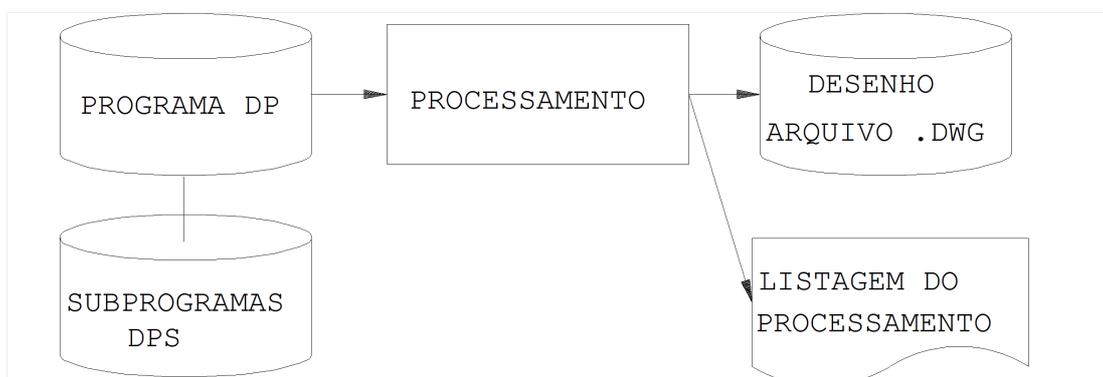


O DP é um interpretador de linguagem de programação, que lê um arquivo de comandos com tipo .DP (programa DP) e gera desenhos. Usualmente temos a geração de um desenho por arquivo .DP:



O desenho gerado pode ser modificado ou completado através do editor gráfico EAG. Quando um programa DP tem erros de codificação, estes erros serão acusados e mostrados na listagem do processamento.

Para maior modularidade, o DP permite a divisão de programas em subprogramas (também chamados de subrotinas ou procedimentos), que são codificados em arquivos .DPS. Os arquivos .DPS são interpretados da mesma maneira que os DP. Por convenção, a interpretação de linguagem começa em um programa .DP e pode ou não ser desviada para subprogramas .DPS:



Na prática, organiza-se os programas DP da seguinte maneira:

Arquivos de programa tipo .DP são usados exclusivamente para a entrada de dados de desenho; o programa .DP chama então um subprograma .DPS para processamento do desenho;

Um desenho paramétrico é inteiramente definido por subprogramas tipo .DPS. O nome do programa de desenho paramétrico é o nome do primeiro subprograma executado para processar o desenho;

Programas de desenho paramétrico (arquivos .DPS) são armazenados em bibliotecas (pastas) que podem ser acessadas por todos os projetistas;

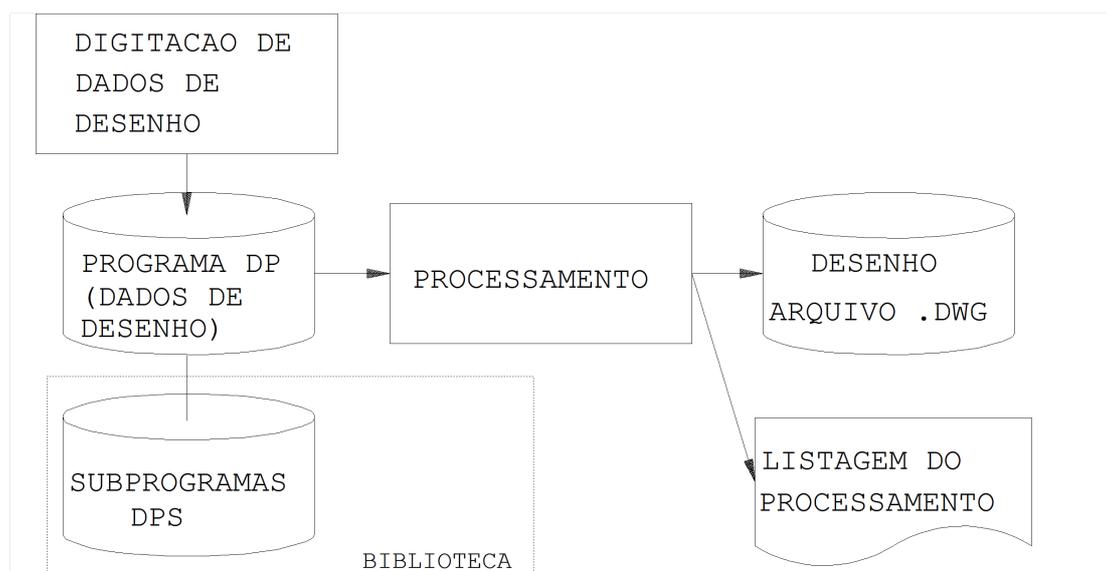
Em um pasta de projeto, codifica-se dados em um arquivo tipo .DP, que chama um subprograma .DPS e gera um desenho. Assim, o pasta de projeto terá apenas dados e desenhos.

Em linguagens de programação usuais, a programação da entrada de dados consome mais de 60% do tempo de programação. Isto não ocorre na linguagem DP, pois:

A entrada de dados é feita por um programa de digitação de dados de desenho. A partir do nome do subprograma de desenho DPS fornecido pelo projetista, o programa automaticamente monta telas de entrada de dados que devem ser preenchidas.

O resultado da digitação de dados é um arquivo .DP com a definição dos parâmetros de desenho e com uma chamada ao subprograma que fará o desenho. Este programa DP é então processado.

O esquema de processamento fica assim:



O projetista que usa o DP para fazer desenhos terá uma rotina do seguinte tipo:

Digitar dados associados a um determinado desenho paramétrico;

Processar estes dados;

Editar graficamente para dar acabamento ao desenho;

Imprimir e plotar o resultado.

## Codificação

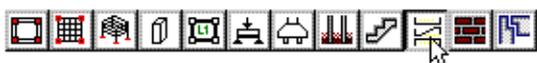
Os arquivos .DP, usualmente são codificados diretamente pelo programa de digitação de dados de desenho.

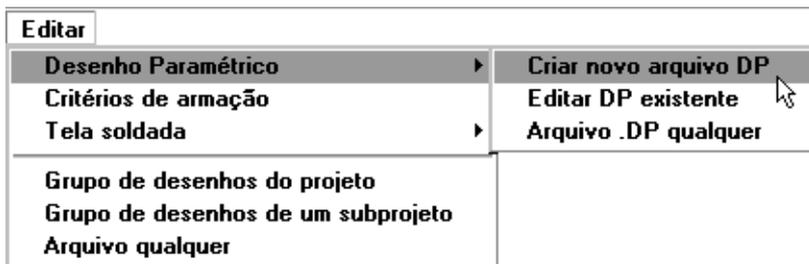
Arquivos .DP e .DPS podem ser editados por qualquer editor de textos, tal como o EDITW, tendo nomes de até 8 caracteres, conforme a convenção do MS-DOS, e tipo .DP ou .DPS.

A programação se faz com o uso de palavras chaves análogas a adjetivos / advérbios. Comandos como "LISTAR TODAS VARIABEIS" são um exemplo do tipo de codificação aceita na linguagem.

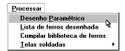
## Processamento

O acionamento dos programas é feito através do gerenciador, chamado pelo ícone do TQS-AGC&DP, usando uma pasta integrada de um edifício ou uma pasta qualquer como pasta de projeto:

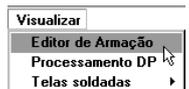




O menu "Editar" "Desenho Paramétrico" "Criar novo arquivo DP" tem o comando para digitação de dados de desenho. Este comando abrirá um quadro que pedirá os dados em uma tela e gerará o arquivo .DP correspondente. Este arquivo por sua vez é processado dentro do menu "Processar" "Desenho Paramétrico"



Como resultado teremos um desenho, que poderá ser editado graficamente através do editor gráfico do AGC&DP, ou pela terceira janela do gerenciador:



O manual de comandos e funções do TQS-AGC & DP mostra com mais detalhes a operação de todo o sistema.

## Grupo de desenhos do projeto

Para facilitar o agrupamento de desenhos, existe o conceito de "Projeto". Projeto é um grupo de desenhos identificados por um número de 4 dígitos, ou "Número de Projeto". O uso do número de projeto é semelhante ao de outros sistemas TQS.

Por convenção, associa-se uma pasta do disco a um número de projeto, e agrupa-se desenhos dentro desta pasta. O gerenciador facilita a criação automática de pastas do edifício.

Com a utilização de projeto, é possível de uma vez só carregar um grupo de desenhos no editor de plantas ou plotar desenhos na impressora.

## Unidades de desenho

Independentemente da unidade de trabalho escolhida, o desenho é (por convenção) sempre armazenado na escala 1:1. A transformação de escala é efetuada somente quando o desenho precisa ser transferido para o plotter ou impressora. Na verdade, a escala precisa ser previamente escolhida, para que os textos colocados no desenho saiam do tamanho esperado.

Os tamanhos de texto colocados no desenho, são sempre definidos programa ajusta o texto em relação ao resto do desenho, a partir de um fator de escala. Por convenção:

Uma unidade de desenho dividida pelo fator de escala resulta no número de centímetros que esta unidade representa no papel

Por esta convenção, o fator de escala depende da unidade de desenho. A tabela abaixo exemplifica os fatores de escala usados para a plotagem de um desenho em escala 1:50:

| Unidade de desenho | Escala desejada | Fator de escala | Centímetros de plotagem por unidade de desenho |
|--------------------|-----------------|-----------------|------------------------------------------------|
| metros             | 1:50            | 0.50            | 2.0000                                         |
| centímetros        | 1:50            | 50.00           | 0.0200                                         |
| polegadas          | 1:50            | 19.68           | 0.0508                                         |

Durante a plotagem, todas as medidas são divididas pelo fator de escala. Os textos são multiplicados pela escala durante o desenho e divididos por ela durante a plotagem, ficando com a dimensão estabelecida pelo projetista no início da codificação.

Uma vez definida a escala, esta é associada ao desenho e usada automaticamente pelo editor gráfico e programas de plotagem. Por default, o fator de escala vale 50.

## Sistema de coordenadas

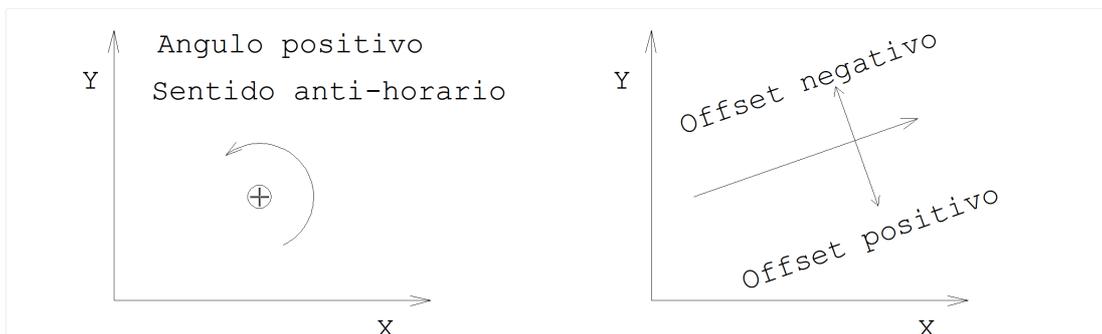
Todo o desenho é feito num plano X-Y, com origem no ponto (0,0). O DP permite a definição de sistemas locais em qualquer posição e ângulo, para a definição de qualquer parte do desenho; no entanto, internamente as coordenadas são sempre armazenadas no sistema global.

## Posição em relação à origem

O desenho pode estar em qualquer lugar em relação à origem (0,0). Esta posição não tem influência na hora de posicionar o desenho no papel para plotagem. Na verdade, o posicionamento efetuado pelo editor de plantas leva em consideração apenas um retângulo envolvente imaginário sobre o desenho.

## Direções

Ângulos são positivos no sentido anti-horário, medidos em relação ao eixo X global. Alguns modos de entrada de pontos permitem também a entrada de um ângulo relativo a outro já existente. Distâncias quando fornecidas em relação a uma direção são positivas à direita da direção e negativas à esquerda (estas distâncias são chamadas aqui de Offsets).



## Precisão

Todas as coordenadas e valores armazenados pelo programa tem 15 dígitos significativos (precisão dupla). Números com mais de 15 dígitos perderão a precisão (a exemplo de coordenadas UTM).

## Convenções de descrição da linguagem

O manual descreve a linguagem DP através de um conjunto de convenções:

Palavras em *letra de impressora* correspondem a dados codificados pelo projetista e saídas de tela.

Palavras em letras maiúsculas são rótulos que o projetista fornece ao programa. A parte sublinhada de cada rótulo corresponde a abreviação mínima do rótulo. Por exemplo:

DESENHO

Apenas as 3 primeiras letras são necessárias para descrever o comando; no entanto, a codificação de modo não abreviado torna a entrada de dados mais "clara".

Por via de regra, todas as palavras chaves usadas no programa são abreviáveis pelas 3 primeiras letras; a única exceção se aplica às palavras chaves utilizadas no comando DEFINE, onde são usadas as 6 primeiras letras.

Palavras em letra minúscula: correspondem a uma informação que o projetista deve fornecer ao programa. Por exemplo:

DISTANCIA nno1 nno2

A palavra DISTANCIA está em letra maiúscula, fazendo parte da descrição; "nno1" e "nno2" neste caso representam números de nós, que devem ser codificados nesta posição. Por exemplo, são codificações válidas:

DISTANCIA 10 20

DIStancia 4 32

DIST 7 20

Não devemos confundir a descrição do comando com o comando em si; as palavras chaves podem ser escritas em letra minúscula ou maiúscula sem distinção.

Algumas palavras em letra minúscula são usadas consistentemente ao longo do manual. Por exemplo, "valor", "ângulo", "coord" são descritos na medida em que aparecem no manual.

Palavras entre colchetes [ ]: é informação que pode ser codificada ou não, a critério do projetista. Por exemplo, na definição de um novo sistema local de coordenadas:

ORIGEM [coord] [ANG angulo]

A sintaxe acima significa: codificar a palavra ORIGEM, seguida opcionalmente por um par de coordenadas, e seguida opcionalmente pela palavra ANG. Se ANG for codificado, então o fornecimento de um ângulo a seguir será obrigatório. São definições válidas:

ORIGEM

```
ORI 0,0 ANG 45
ORI 100,100
ORIGEMANG 30
```

Note que os colchetes usados na descrição do comando não são codificados.

Definição em várias linhas: são opções que o projetista pode escolher para uma determinada entrada. Por exemplo:

```
nno coord
OFFSET valor coord1 coord2
INTERSECCAO linha1 linha2
PROJECAO coord1 coord2 coord3
```

A sintaxe acima descreve as formas alternativas de definição de um nó. Por exemplo:

```
15 0,0
27 OFF 12.5 1 2
39 INTERSECCAO 1 10 20 3
```

... significa que um item de dados pode ser repetido. Por exemplo:

```
POLIGONAL coord1; coord2; ...
```

Estamos descrevendo uma poligonal; ela é construída a partir de 2 ou mais nós, separados pelo sinal ";".

## Convenções de entrada de dados

### Formato de entrada

Todas as informações codificadas pelo usuário são fornecidas em formato livre, isto é, em qualquer coluna da linha de entrada, separadas por um ou mais brancos ou caracteres de tabulação. Letras maiúsculas e minúsculas podem ser usadas livremente e sem distinção. Os comandos a seguir têm todos o mesmo significado:

```
100 5@%DISTAN<DIR20 30-45
100 5 @ %DISTAN < DIR 20 30 - 45
100 5 @ %DISTAN < DIR 20 30 - 45
```

Em geral, a boa prática de codificação é alinhar itens correlatos em linhas diferentes, pois isto facilita a leitura. Para isto, pode-se usar o caractere de tabulação para avançar o cursor. Por exemplo, a codificação:

```
100 5@%DISTAN<DIR20 30-45
101 100@%DISTAN<DIR5 6
```

se tornaria mais legível numa listagem, se estivesse na forma:

```
100 5 @%DISTAN < DIR 20 30 - 45
101 100 @%DISTAN < DIR 5 6
```

## Abreviações

Como foi dito anteriormente, todas as palavras chaves usadas pelo programa são abreviáveis pelas 3 primeiras letras, com exceção das palavras usadas pelo comando `DEFINE`.

## Tamanho da linha de entrada

O programa lê até os primeiros 80 caracteres de cada linha de entrada. Se o projetista tiver necessidade de fornecer mais dados do que cabem numa linha, poderá continuar na linha seguinte desde que coloque um sinal de "-" como último caractere da linha atual. Por exemplo:

```
POL 2 3 22 35 25 -
23 16 14 13 10 -
7 29
```

O comando poligonal deve ser definido numa única linha. Neste exemplo, escolheu-se dividir o comando em 3 linhas diferentes, colocando-se o sinal de continuação entre uma linha e outra.

O sinal de "-" pode ser colocado em qualquer coluna, desde que seja o último elemento codificado na linha.

## Comentários

Em qualquer parte da codificação podem ser colocados comentários. Comentários não são interpretados pelo DP, mas servem para documentar o arquivo e orientar o usuário na análise e verificação da codificação.

Comentários são definidos pelo caractere "\$"; quando o DP encontra um "\$" numa linha de comando, todo o resto da linha passa a ser considerada como comentário.

Faz parte da boa prática de codificação colocar comentários em pontos importantes. Por exemplo, no desenho de uma peça com cortes A-A e B-B:

```
$
$ Corte A-A
$
```

(definição de desenho)

§

§ Corte B-B

§

(definição de desenho)

Nem sempre é visível ou trivial descobrir o que se está desenhando dentro de um arquivo DP (principalmente quando é necessário alterar um arquivo). A colocação de comentários pode ajudar a analisar o conteúdo do arquivo.

## Cópia de linha anterior

O sistema de entrada de dados do DP permite alguns "truques", que podem ser usados para acelerar a codificação: trata-se de um sistema que permite copiar partes da linha anterior para atual. Por exemplo:

```
INSERE 'X' 10 ESC 1
" " 20 " 2
" " 30 " 3
```

Na primeira linha definiu-se a inserção do bloco 'X' no nó 10 e escala 1. Nas linhas seguintes inseriu-se o mesmo bloco, mas em nós e escalas diferentes. O caractere de aspas (") copia o elemento da linha acima.

Outra possibilidade para copiar elementos da linha anterior:

```
INSERE 'X' 10 ESX 1 ESY 2 ANG 45
INSERE 'Y' ...
INSERE 'Z' ...
```

Na primeira linha foi definida a inserção do bloco 'X' em um certo nó, escalas X e Y e ângulo de rotação. Nas duas linhas seguintes, foram colocados 2 outros blocos, na mesma posição e escala; o resto da linha, a partir dos "..." foram copiados da linha de cima.

## Seções de programa

Todos os comandos são agrupados logicamente em "Seções de Programa" terminadas pela palavra FIM. Por exemplo:

```
DESENHO 'nome'
(comandos de definição de desenho)
```

FIM

O comando DESENHO define o início da descrição de um desenho. Esta descrição forma uma seção de programa terminada pela palavra FIM.

Seções de programa podem conter outras seções dentro. Por exemplo, dentro da seção de desenho, para definirmos um conjunto de variáveis globais:

```
DESENHO 'nome'  
  
GLOBAIS  
  
(definição de variaveis globais)  
  
FIM  
  
(comandos de definição de desenho)  
  
FIM
```

Neste exemplo, a primeira palavra FIM encontrada marca o fim da seção de variáveis globais do desenho 'nome', e não a seção de DESENHO. Existem muitos outros casos onde se pode definir seções de programa. Como regra geral, a palavra FIM fecha a seção aberta mais recentemente.

O próprio programa DP pode ser considerado como uma seção de programa; a palavra FIM, quando não fecha nenhuma outra seção, fecha o programa e termina o processamento do desenho.

## Ordem de entrada de dados

Não existe uma ordem pré-estabelecida para a entrada de dados. O DP interpreta comandos do projetista na medida que os lê. A única restrição é que quando um elemento referencia outro, o outro precisa ser criado primeiro. Por exemplo, para a definição de uma linha entre 2 nós é necessário definir os 2 nós primeiro.

## Organização de um arquivo .DP

O arquivo .DP contém a descrição de um ou mais desenhos. A seção de programa DESENHO delimita o início e o fim da descrição de cada desenho:

```
DESENHO [ 'nome' ]  
  
...  
  
FIM
```

Entre a palavra DESENHO e FIM é feita a descrição do desenho de nome 'nome' (entre apóstrofes, até 8 caracteres). Se o nome do desenho for omitido, 'TMP' é assumido. Um típico arquivo .DP teria a seguinte organização:

```
DESENHO 'nome1'  
(definição dos elementos do desenho 'nome1')  
FIM
```

```
DESENHO 'nome2'  
(definição dos elementos do desenho 'nome2')  
FIM
```

.....

Assim, "n" desenhos podem ser descritos dentro de um arquivo .DP. Existem muitos modos para se definir um desenho, sendo o mais importante o de chamada de subprogramas com a passagem de parâmetros.