

# Manual de Programação

Manual de Programação

## Introdução

O sistema de desenho de ferros do CAD/TQS consiste em textos que descrevem ferros em níveis específicos seguindo certa convenção, chamada de “convenção de ferros do NGE”. Este sistema foi projetado em 1986 com o objetivo de ser simples e independente de editores gráficos. Não apenas o editor de armação CAD/AGC segue a convenção, como também o usuário deve seguir. Isto dá margem a diversos tipos de erros na geração de tabela e na transmissão de informações para a central de corte e dobra.

Um novo sistema foi desenvolvido para o desenho de ferros, chamados de “Ferros Inteligentes” (nome provisório), baseados no IPOEAG – a interface de programação de objetos do editor gráfico. No novo sistema, ferros são objetos com atributos, gerados através de uma única biblioteca, e editáveis através de operações de alto nível. O usuário passará a editar entidades que representam ferros, em vez de textos dentro de uma convenção. A biblioteca que cuida da geração, leitura e gravação de ferros é a IPOFER.DLL, com interface definida em \LIB\IPOFERX.H e linkedição com \LIB\IPOFER.LIB.

## Sobre a programação IPOEAG

A biblioteca IPOFER.DLL cria objetos de desenho que são reconhecidos pelos editores gráficos TQS como objetos comuns, que podem ser criados, editados (copiar, mover, escalar, espelhar, alterar), apagados e recuperados através de operação UNDO e REDO. O sistema IPOEAG permite que novos objetos sejam criados externamente e manipulados por DLL separada, sem que os editores conheçam o objeto criado ou precisem ser modificados. Para isto, são definidas funções “virtuais”, equivalentes às funções virtuais das linguagens de programação a objeto, que são chamadas pelos editores gráficos nas DLLs associadas aos respectivos objetos IPO (como IPOFER.DLL).

## Programa exemplo de programação

O programa TSTIPOFER.CPP distribuído com a biblioteca demonstra a criação de um desenho com todos os tipos de ferros e diversos tipos de atributos. A linkedição do programa é esta:

```
/OUT:\TQSW\EXEC\TSTIPOFER.EXE
```

```
/INCLUDE:_WinMain@16
```

```
TSTIPOFER.OBJ
```

```
\LIB\GEOM.LIB
```

```
\LIB\IPOFER.LIB
```

```
\LIB\MDWG.LIB
```

```
\LIB\ACESSOL.LIB
```

```
\LIB\WINLIB.LIB
```

```
LIBF.LIB
```

```
LIBC.LIB
```

Em resumo: o programa roda em TQSW\EXEC e declara IPOFER para gerar ferros, MDWG para gravar desenhos, ACESSOL para dados de edifício. A biblioteca GEOM foi declarada pelo uso da rotina georod, que gira coordenadas

dentro do programa.

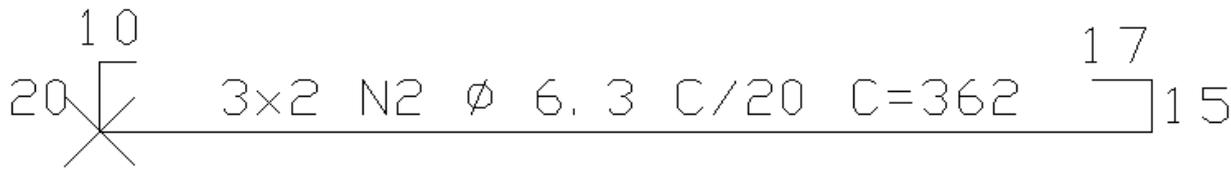
Este programa é a melhor referência de código testado para a geração de ferros, e o resultado é um arquivo TESTE.DWG com ferros (cerca de 340) gravado na pasta atual.

O principal header a ser incluído no programa é o \LIB\IPOFERX.H, que contém toda a interface de programação.

## Nomenclatura de ferros

Os tipos de ferros e principais atributos usados são descritos a seguir. Temos quatro tipos de ferros. Os “X” mostrados nas figuras são os pontos de inserção do ferro nas chamadas de criação.

### Ferros retos



Consiste em um trecho principal ou “horizontal” e zero, uma ou duas dobras ortogonais à esquerda e/ou direita.

O texto “3x2 N2 Ø 6.3 C/20 C=362” é o “texto de descrição dos ferros”. Se as coordenadas do texto de descrição não forem fornecidas, uma será assumida.

“3” é o multiplicador do número de ferros (ex: 3 nervuras com duas barras cada)

“2” é o número de ferros

“N2” é o número de posição. Cada ferro de características diferentes deve ter um número de posição diferente. Ferros diferentes, mas com mesmas características podem ter o mesmo número de posição.

“Ø 6.3” é a bitola em mm do ferro.

“C/20” é o espaçamento. Se um ferro não é distribuído, o valor do espaçamento deve ser zero.

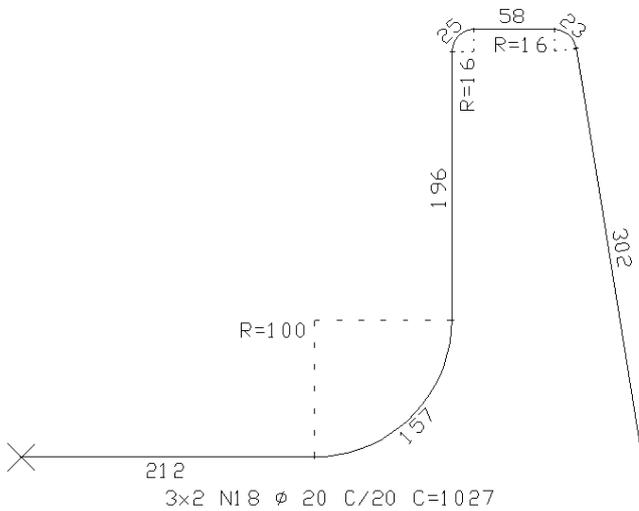
“C=320” é o comprimento total. Este comprimento, conforme critérios, pode ser afetado por descontos devido a raio de curvatura das dobras. Como este comprimento é mostrado e depois passado para a tabela de ferros é controlado pela biblioteca.

As dobras recebem a “identificação de dobras”, que é o seu comprimento. As identificações podem ser mostradas ou não.

Ferros retos são definidos por sua faces externas, e as dobras são as medidas destas faces.

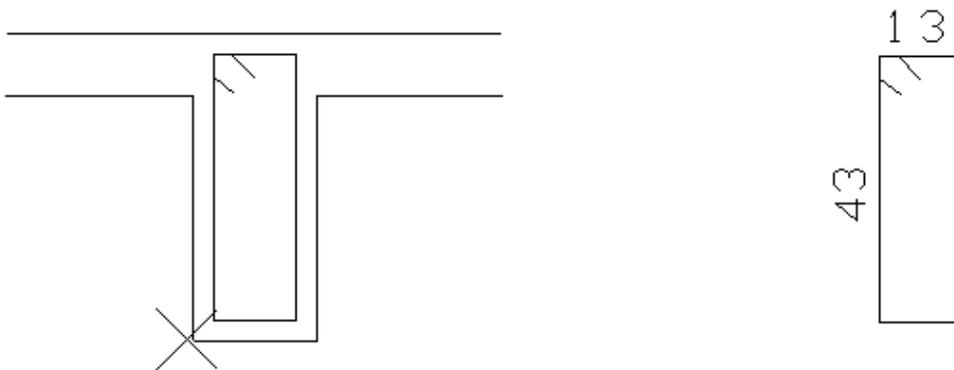
## Ferros genéricos

São ferros definidos por uma poligonal qualquer formada por trechos retos e/ou em arco.



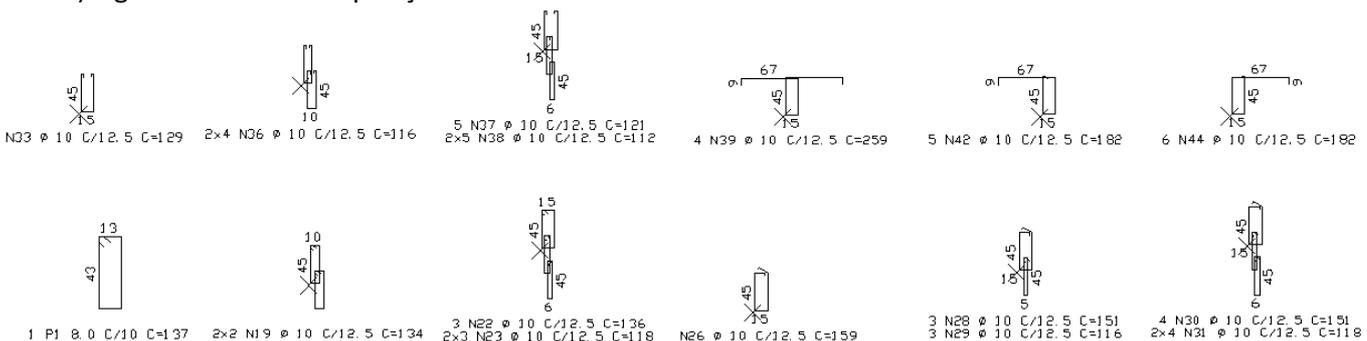
Ferros genéricos são definidos por seu eixo.

## Estribos de vigas



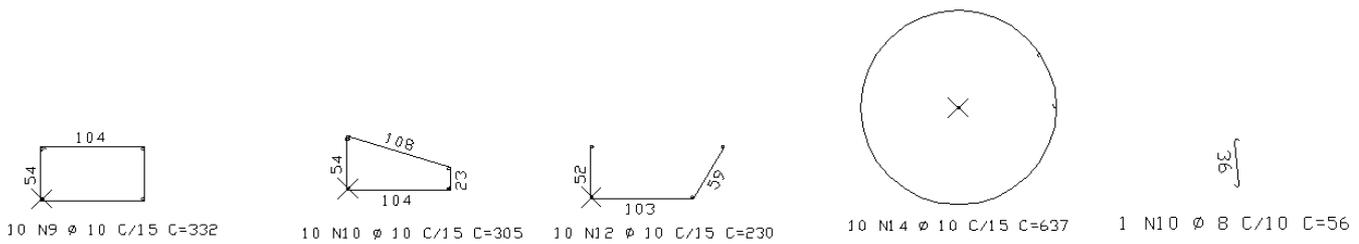
1 P1 8.0 C/10 C=137

Estribos inseridos por um canto da seção transversal. O cobrimento para posicionar o estribo é armazenado junto com o elemento. Estribos podem ser normais, fechados (para torção) ou abertos, e ainda ter a pata superior alongada para servir como negativo na laje. Estes estribos podem combinar com o número de ramos (2, 4 ou 6 ramos) e gerar uma ou duas posições de ferros.



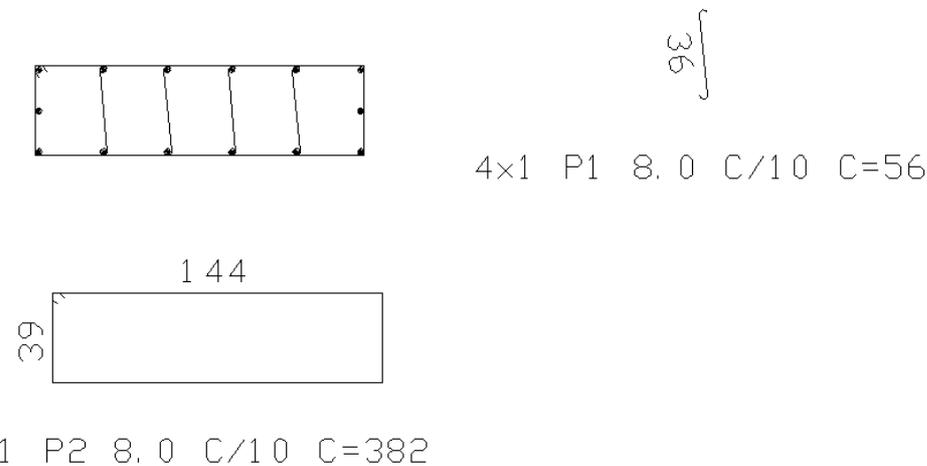
## Estribos genéricos (de pilares)

Estribos de pilares têm como referência os pontos de centro das armaduras longitudinais abraçados pelo estribo. Temos estribos fechados, abertos, grampos (por 2 pontos) e estribos circulares:



### Inserção de linhas de ferros

Um ferro pode ser representado por várias linhas no desenho, independente da quantidade final ou multiplicador do número de ferros. O novo Editor de Armação tem um comando para duplicar linhas, que também incrementa o multiplicador de um ferro. Na figura abaixo, o grampo com posição P1 tem sua linha inserida cinco vezes no desenho:

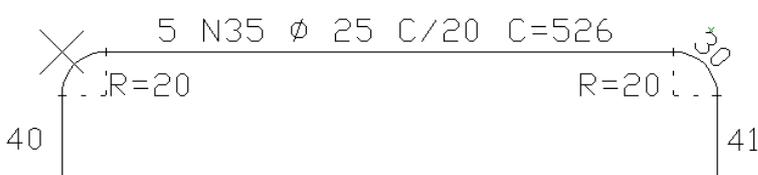


Quando um ferro tem várias linhas no desenho, por padrão a última linha inserida terá o texto de descrição do ferro.

### Raio de curvatura

Ferros em geral são dobrados na central de corte e dobra com o uso de pinos definidos pela norma NBR6118:2003 9.4.2.3. Mas ferros para resistir à força cortante ou nó de pórtico (principalmente de vigas) devem receber raio de dobra bem maior, definido pela NBR6118:2003 18.2.2. O tipo de uso do ferro deve ser informado à biblioteca para cálculo correto do raio.

Ferros com estes raios de dobra podem ser representados ou não, conforme critério:



As informações de curvatura são somente atributos, o programa que gera o ferro informa somente trechos retos. A representação é controlada pela biblioteca.

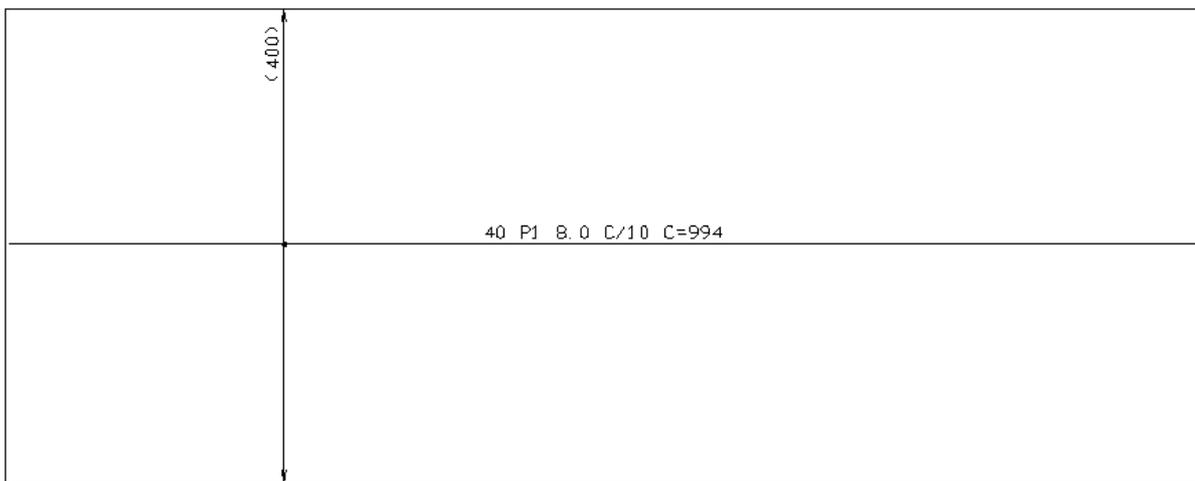
## Ganchos

Também a NBR6118:2003 18.2 define comprimentos padrão de ancoragem nas extremidades das barras chamados de “ganchos”. Podem ser especificados ganchos nas pontas de ferros retos ou genéricos, a 180, 135 e 90 graus. Os ganchos são desenhados em escala real com o comprimento de acordo com a norma. O comprimento total é calculado pela biblioteca.



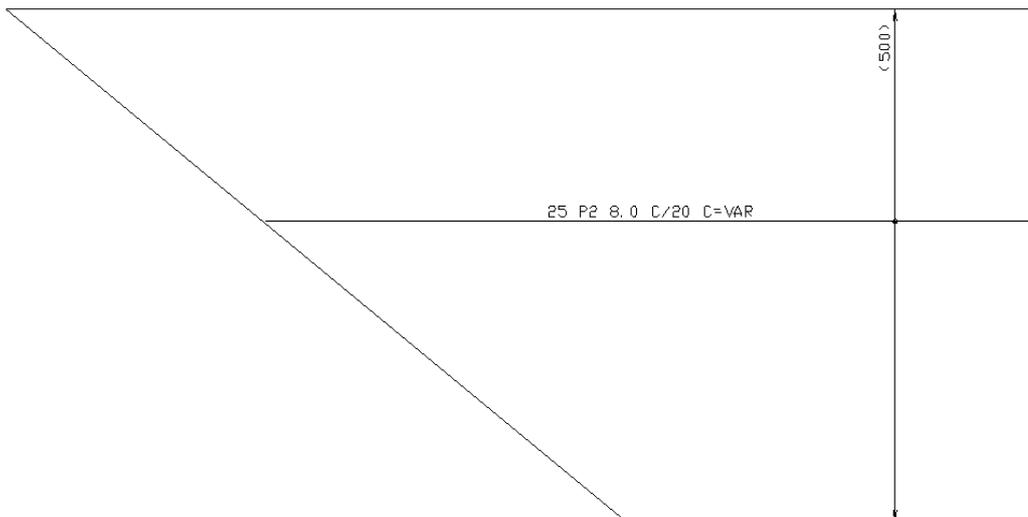
## Faixas de distribuição

Ferros distribuídos com espaçamento podem ter a faixa de distribuição indicada. Se a faixa for associada ao ferro, e se o usuário alterar o comprimento da faixa, o número de ferros será alterado junto, automaticamente.



Um ferro pode ter múltiplas faixas de distribuição. O número de ferros será igual à soma dos ferros de todas as faixas. A associação de um ferro com as faixas é opcional.

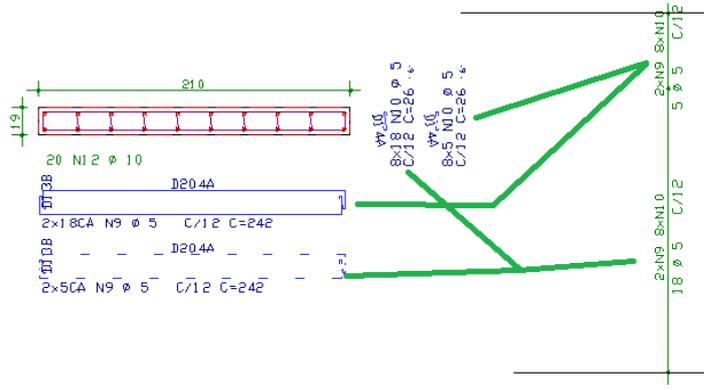
## Faixas de distribuição com comprimento variável



25 P2 8.0 C/20	
P2A	406
P2B	420
P2C	454
P2D	478
P2E	502
P2F	526
P2G	550
P2H	574
P2I	598
P2J	622
P2K	646
P2L	670
P2M	694
P2N	718
P2O	742
P2P	766
P2Q	790
P2R	814
P2S	838
P2T	862
P2U	886
P2V	910
P2W	934
P2X	958
P2Y	982

Ferros retos ou genéricos podem ser distribuídos com variação de um dos trechos. A biblioteca fornece facilidades para gerar automaticamente as posições de ferro dentro de um contorno ou permite a entrada de cada comprimento de trecho variável. A tabela de ferros de comprimentos variáveis também é gerada automaticamente, mas deve ser inserida pelo usuário ou programa.

## Faixas de distribuição de múltiplas posições

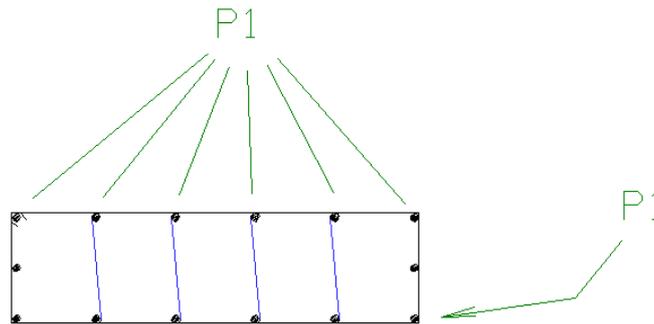


As faixas de distribuição de vigas e lajes são armazenadas junto com os dados do ferro. Isto é, podemos ter um ferro com múltiplas faixas. Faixas de múltiplas posições tem uma relação diferente: vários ferros podem ser associados a uma única faixa. Um exemplo são pilares, onde uma faixa de distribuição abrange diversos estribos e grampos da mesma seção do pilar.

Faixas de múltiplas posições são criadas independentemente dos ferros, e depois associada a cada um deles.

### Identificação de posição

“Identificação de posição” pode ser usada para identificar ferros em corte e outros. Se associadas a um ferro, alteram automaticamente o número da posição em caso de renumeração automática.



Um ferro pode ter várias identificações de posição

### Lógica de criação de um ferro

Um ferro é um objeto representado por uma estrutura de dados na memória. Para criar um ferro é necessária a declaração de IPOFERX.H e a seguinte chamada:

```
SUBREXPIPO IPOFER_CRIARFERRO (// Inicia a definição de um ferro
```

```
void **pedmv, // <- Pnt p/handle de desenho
```

```
void**ferobjv, // -> Handle do ferro
```

```
int*istat); // -> Ret <> 0 se não criou
```

Esta chamada cria a estrutura de um ferro na memória e associa o ferro criado com um desenho aberto. As chamadas de desenho que usam pedmv estão declaradas em MDWG.H. Uma chamada típica abrindo um desenho novo e criando um ferro é como esta:

```

void *pedmv= NULL;

int istat= 0;

g_desopn (arqdwg, seed, NULL, &pedmv, &istat);

void *ferobjv= NULL;

IPOFER_CRIARFERRO (&pedmv, &ferobjv, &istat);

```

Todas as rotinas de IPOFER.DLL tem prefixo IPOFER\_ e um argumento ferobjv, que representa um ferro na memória. Os argumentos são sempre passados por endereço, para facilitar a interface em Fortran.

O ferro criado não tem nenhuma informação. As demais chamadas definem atributos do ferro definido por ferobjv. Você pode manter múltiplos ferros abertos e guardar os handles de cada um para associar atributos posteriormente. A biblioteca garante que os handles são únicos e válidos enquanto o desenho definido por pedmv estiver aberto.

Algumas constantes usadas nas chamadas estão declaradas em IPOFERX.H (IPOFERX.CMM no caso Fortran).

## Dados gerais

São dados comuns a todo o tipo de ferro.

```

SUBREXPIPO IPOFER_ICFTPF (
void**ferobjv, // <- Handle do ferro
int*icftpf); // <- Tipo de ferro
// (ICPFRT) ferro reto
// (ICPFGN) ferro generico
// (ICPSTR) estribo
// (ICPSTRGEN) estribo
// (ICPGRA) grampo
//
//Número da posição: é mais racional não fornecer, e no final chamar
//IPOFER_POS_RENUMERAR
//
SUBREXPIPO IPOFER_ICFPOS (
void**ferobjv, // <- Handle do ferro
int*icfpos); // <- Número da posição
//
//Número de ferros. Se o ferro tiver faixas de distribuição c/cálculo
//automático, este número será ignorado e recalculado
//

```

```

SUBREXPIPO IPOFER_ICFNFR (
void**ferobjv,// <- Handle do ferro
int *icfnfr);// <- Número de ferros
//
//Multiplicador do número de ferros
//
SUBREXPIPO IPOFER_ICFMUL (
void**ferobjv,// <- Handle do ferro
int*icfmul);// Multipl do numero de ferros
//
//Bitola
//
SUBREXPIPO IPOFER_CFRBIT (
void**ferobjv,// <- Handle do ferro
double *cfrbit);// <- Bitola mm
//
//Espaçamento cm
//
SUBREXPIPO IPOFER_CFRESP (
void**ferobjv,// <- Handle do ferro
double*cfresp);// <- Espaçamento cm
//
//Mostrar espaçamento "C/Nerv"
//
SUBREXPIPO IPOFER_ICFNRV (
void**ferobjv,// <- Handle do ferro
int*icfnrv);// <- Espaçamento em laje nervurada
// (ICPNNV) Espacamento normal
// (ICPNRV) Espacamento em nervuras
//
//Observação opcional
//
SUBREXPIPO IPOFER_CFROBS (
void**ferobjv,// <- Handle do ferro
const char *cfrobs, int ARG_FAN);// Observação do texto do ferro

```

```

//
//Ganchos à esquerda e direita. São ganchos para ferros que não
//sejam nó de pórtico ou submetidos à força cortante, com
//raios de dobramento padrão conforme NBR6118:2003 9.4.2.3
//
SUBREXPIPO IPOFER_ICGTGE (
void**ferobjv,// <- Handle do ferro
int*icgtge,// <- Gancho padrão à esquerda
// (ICPSGA) Sem gancho a esquerda
// (ICP090) Gancho a 90
// (ICP135) Gancho a 135
// (ICP180) Gancho a 180
int*icgtgeneg);// <- (1) desenhar como negativo

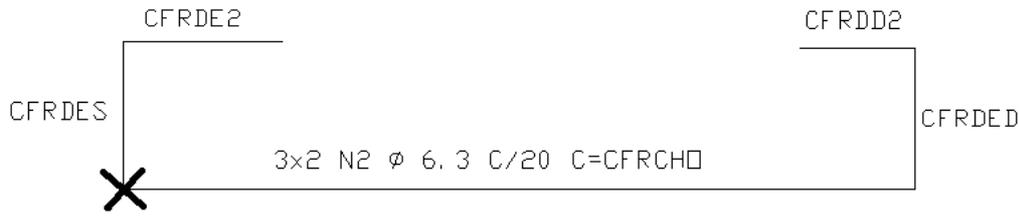
SUBREXPIPO IPOFER_ICGTGD (
void**ferobjv,// <- Handle do ferro
int*icgtgd,// <- Gancho padrão à direita
// (ICPSGA) Sem gancho a direita
// (ICP090) Gancho a 90
// (ICP135) Gancho a 135
// (ICP180) Gancho a 180
int*icgtgdneg);// <- (1) desenhar como negativo
//
//Lance do CAD/Pilar
//
SUBREXPIPO IPOFER_DEFLANCE (
void**ferobjv,// <- Handle do ferro
int*ilance);// <- Lance do CAD/Pilar
//
//Definição de cobrimento atual
//O cobrimento é usado p/remontar estribos ICPENR/ICPEFC/ICPEAB/ICPENC
//Também é usado com ferros retos p/descontar cobrimento de faces
//
SUBREXPIPO IPOFER_CFRREB (
void**ferobjv,// <- Handle do ferro

```

double\*cfrrdeb);// <- Cobr cm p/geom estrib e f.var

## Ferros retos

São definidos pelos comprimentos das dobras.



```
SUBREXPIPO IPOFER_CFRCHO (  
void**ferobjv, // <- Handle do ferro  
double*cfrrcho); // <- Comprimento horizontal (cm)  
SUBREXPIPO IPOFER_CFRDES (  
void**ferobjv, // <- Handle do ferro  
double*cfrrdes); // <- Dobra a esquerda  
SUBREXPIPO IPOFER_CFRDED (  
void**ferobjv, // <- Handle do ferro  
double*cfrrded); // <- Dobra a direita  
SUBREXPIPO IPOFER_CFRDE2 (  
void**ferobjv, // <- Handle do ferro  
double*cfrrde2); // <- 2a dobra a esquerda  
SUBREXPIPO IPOFER_CFRDD2 (  
void**ferobjv, // <- Handle do ferro  
double*cfrrdd2); // <- 2a dobra a direita  
SUBREXPIPO IPOFER_ICFIPN (  
void**ferobjv, // <- Handle do ferro  
int*icfipn); // <- Tipo de ferro reto  
// (ICPPOS/ICPNEG) Positiva/Negativa  
SUBREXPIPO IPOFER_ICFDSU (  
void**ferobjv, // <- Handle do ferro  
int*icfdsu); // Tipo da 2a dobra do ferro reto  
// (ICPDSU) dobra de suspensao  
// (ICPNSU) dobra normal  
SUBREXPIPO IPOFER_IDIRDOBR (  
void**ferobjv, // <- Handle do ferro
```

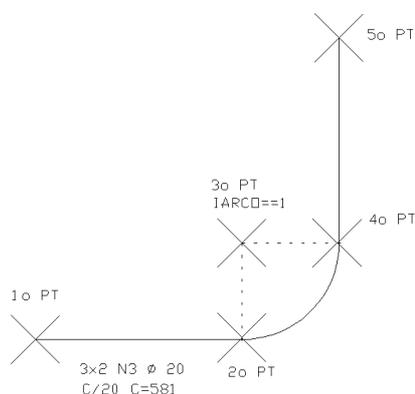
```

int*idir dobr); // <- Direção dos textos de dobras
// (0) // trecho (1) // direção do fer
SUBREXPIPO IPOFER_IPOSTREPRI (
void**ferobjv, // <- Handle do ferro
int*ipostrepri); // <- Texto do trecho principal
// (0) não (1) acima (2) abaixo

```

## Ferros genéricos

As coordenadas dos ferros genéricos (e também estribos genéricos) são fornecidas ponto a ponto no sistema local, com as coordenadas Z atualmente ignoradas. As coordenadas são transformadas na inserção de cada linha de ferro. Para a definição de arcos, 3 pontos são definidos em sequência: o ponto inicial do arco (e final do trecho anterior), o centro (com `iarco==1`) e o ponto final (e início do próximo trecho).



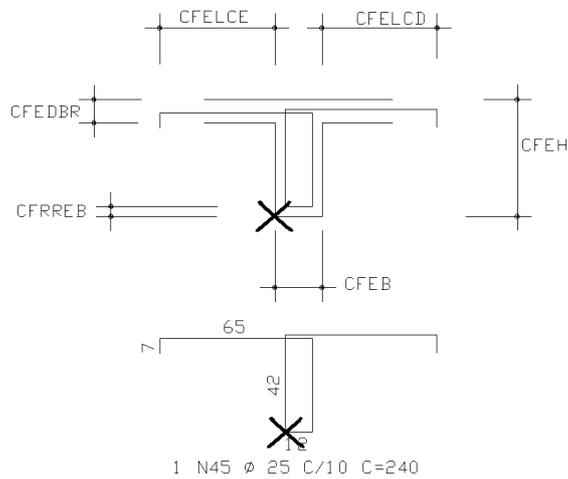
```

SUBREXPIPO IPOFER_PTFERGEN (
void**ferobjv, // <- Handle do ferro
double*xpt, // <- X
double*ypt, // <- Y
double*zpt, // <- Z
int*iarco, // <- (1) se centro de arco
int*ident dobr, // <- (1) p/identificar a dobra
int*indfrt); // <- Índice de trecho equiv ou (-1)

```

## Estribos de vigas

Estribos de vigas tem referência nas medidas da seção menos o cobrimento. O cobrimento deve ser passado através dos parâmetros gerais. Cada linha deve ser inserida por IPOFER\_LINHAFER nas posições mostradas abaixo.



```

SUBREXPIPO IPOFER_ICFNRE (
void**ferobjv,// <- Handle do ferro
int*icfnre);// <- Número de ramos de estribo
// (ICPNR2) Estribo de 2 ramos
// (ICPNR4) Estribo de 4 ramos
// (ICPNR6) Estribo de 6 ramos
//
//- Estribos padrão dependem de chamar IPOFER_ICFTPE, IPOFER_DADESTRIB,
//IPOFER_CFRREB, e IPOFER_LINHAFER
//- Todos os estribos devem definir IPOFER_NFIANE ou usar default
//
SUBREXPIPO IPOFER_ICFTPE (
void**ferobjv,// <- Handle do ferro
int*icftpe);// <- Formato de estribo
// ICPENR/ICPEFC/ICPEAB/ICPENC

SUBREXPIPO IPOFER_DADESTRIB (// Dados de estribo
void**ferobjv,// <- Handle do ferro
double*cfeb,// <- Base da seção cm / Base inicial
double*cfeh,// <- Altura da seção cm / Alt inicial
double*cfeb2,// <- Base final variável
double*cfeh2,// <- Altura final varriavel
double*cfelce,// <- Largura colab a esquerda
double*cfelcd,// <- Largura colab a direita
double*cfedbr);// <- Dobra do estribo na laje

SUBREXPIPO IPOFER_NFIANE (

```

```
void**ferobjv,// <- Handle do ferro
```

```
int*nfiane);// <- Num de bitolas da pata do estribo
```

## Grampos de vigas

Poderiam ser descritos como ferros retos, mas são representados em perspectiva.



```
SUBREXPIPO IPOFER_DADGRAMPO (
```

```
void**ferobjv,// <- Handle do ferro
```

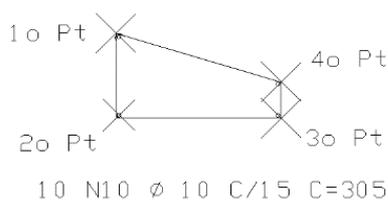
```
int*igrdir,// <- direção (1) esquerda (-1) direita
```

```
double*grclong,// <- Comprimento longitudinal cm
```

```
double*grctran);// <- Comprimento transversal cm
```

## Estribos genéricos (de pilares)

Estribos genéricos são definidos tendo como referência o centro da armadura longitudinal que abraçam. Para isto precisam da definição da bitola da armadura longitudinal em IPOFER\_BITLESTRGEN. O tipo de estribo é definido em IPOFER\_ICFTPEG. Os pontos podem ser definidos por IPOFER\_PTFERGEN, ou pela IPOFER\_PTESTRGEN, mas simples. Em estribos fechados, o ponto de fechamento não deve ser fornecido.



```
SUBREXPIPO IPOFER_ICFTPEG (
```

```
void**ferobjv,// <- Handle do ferro
```

```
int*icftpeg);// <- Formato de estribo
```

```
// ICPEGENFEC/ICPEGENABR/ICPEGENGRA/ICPEGENCIR
```

```
//
```

```
//Definição de um estribo genérico de pilar. 3 tipos de ferros:
```

```
//- Grampo, Aberto, Fechado
```

```
//
```

```
//- Os pontos fornecidos são do centro da armadura longitudinal
```

```
// de bitola bitlnest que o estribo abraça.
```

```
//- Valem as coordenadas absolutas definidas em IPOFER_PTESTRGEN.
```

```
//- No estribo fechado, não fechar o último ponto.
```

```
//
```

```
SUBREXPIPO IPOFER_PTESTRGEN (
```

```

void**ferobjv,// <- Handle do ferro
double*xlong,// <- X centro de armadura longitudinal
double*ylong);// <- Y centro de armadura longitudinal
//
//Define a bitola longitudinal de referência para a definição dos pontos
//de um estribo genérico
//
SUBREXPIPO IPOFER_BITLESTRGEN (
void**ferobjv,// <- Handle do ferro
double*bitlnest);// <- Bitola longitudinal de referên mm
//
//Define raio de estribo circular. (COBRIMENTO DESCONTADO DESTES FERRO)
//
SUBREXPIPO IPOFER_RAIOESTR (
void**ferobjv,// <- Handle do ferro
double*raioestr);// <- Raio de estribo circular cm
//
//Diversos critérios que modificam o grampo de pilar
//- Critério Paulo Sayeg: somar ao comprimento do grampo de pilar
//- Critério K83 do CAD/Pilar: desenho de grampo em S ou C
//- Critério K87 do CAD/Pilar: Soma duas bitolas ao compr de grampo
//
SUBREXPIPO IPOFER_COMPRADGRA (
void**ferobjv,// <- Handle do ferro
double *compradgra,// <- Compr adic cm a grampo de pilar
int*igrasck83,// <- (0) Grampo em S (1) Grampo em C
int*igrak87);// <- (1) Soma duas bitolas ao grampo

```

## Posição do texto de identificação

Textos de identificação são posicionados no desenho em posições padrão, não precisando a princípio ser fornecidos. Estas posições podem ser controladas pelas rotinas a seguir. As rotinas são redundantes para simular algumas chamadas da antiga biblioteca ARMLIB. Para posicionamento e escolha do texto de identificação, é suficiente a chamada às rotinas IPOFER\_ARM\_LAJMNTX e IPOFER\_ARM\_POSTEXX.

```

//
//Seleciona os textos de identificação a serem mostrados
//

```

```

SUBREXPIPO IPOFER_ARM_LAJMNTX (
void**ferobjv, // <- Handle do ferro
int*iflnfr, // <- (1) número de ferros
int*iflpos, // <- (1) número da posição
int*iflbit, // <- (1) bitola
int*iflesp, // <- (1) espaçamento
int*iflcmp); // <- (1) comprimento
//
//Mesma função de IPOFER_ARM_LAJMNTX
//
SUBREXPIPO IPOFER_ARM_LAJMNT (
void**ferobjv, // <- Handle do ferro
int*icond); // <- (0) Completo, c/delimitadores
// <- (1) posic, bitola e espaç
// <- (2) posic
// <- (3) posic, bitola
// <- (4) posic, bitola, espaç
// <- (5) posic, bitola, espaç, nf
// <- (6) posic, nf
// <- (7) comprimento
//
//Fornece posição e formatação do texto de identificação
//
SUBREXPIPO IPOFER_ARM_TEXTOFER (
void**ferobjv, // <- Handle do ferro
double*xtex, // <- Coordenadas do texto
double*ytex, // <- Coordenadas do texto
double*htex, // <- Altura de texto cm plotados
double*angulo, // <- Ângulo do texto em graus
int*imod, // <- (0) normal
// <- (1) bit/espac para tabela var
// <- (2) posição
// <- (3) posição centrada, levant circ
int*icentr, // <- (0) centrado
// (1) esquerda

```

```

// (2) direita
int*iquebr); // <- (0) sem quebra
// (1) salto de linha no C/ ou C=
// (2) salto de linha no {
// (3) depois do C/
//
//Mesmo que IPOFER_ARM_TEXTOFER
//
SUBREXPIPO IPOFER_ARM_POSTEXX (
void**ferobjv, // <- Handle do ferro
double*xtex, // <- Coordenadas do texto
double*ytex, // <- Coordenadas do texto
double*htex, // <- Altura de texto cm plotados
double*angulo, // <- Ângulo do texto em graus
int*imod, // <- (0) normal
// <- (1) bit/espac para tabela var
// <- (2) posição
// <- (3) posição centrada, levant circ
int*icentr, // <- (0) centrado
// (1) esquerda
// (2) direita
int*iquebr); // <- (0) sem quebra
// (1) salto de linha no C/ ou C=
// (2) salto de linha no {
// (3) depois do C/
//
//Mesmo que IPOFER_ARM_TEXTOFER com texto centrado e sem quebra de linha
//
SUBREXPIPO IPOFER_ARM_POSTEX (
void**ferobjv, // <- Handle do ferro
double*xtex, // <- Coordenadas do texto
double*ytex, // <- Coordenadas do texto
double*htex, // <- Altura de texto cm plotados
double*angulo, // <- Ângulo do texto em graus
int*imod); // <- (0) normal

```

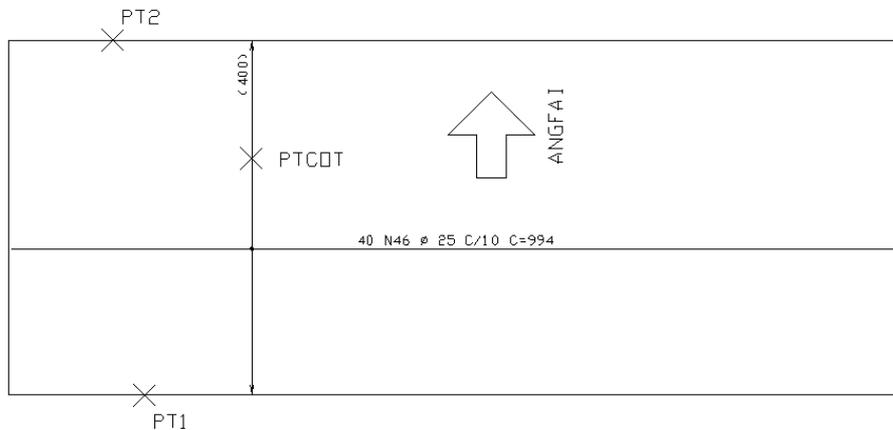
```
// <- (1) bit/espac para tabela var
```

```
// <- (2) posição
```

```
// <- (3) posição centrada, levant circ
```

## Faixas de distribuição

Faixas são associadas a ferros e podem fazer com que o cálculo de quantidade seja automático.



```
//  
//Cálculo do número de ferros dado comprimento da faixa e espaçamento.
```

```
//Retorna ISTAT <> 0 se espaçamento zero
```

```
//
```

```
//
```

```
SUBREXPIPO IPOFER_ARM_CALNFR (
```

```
int*icfes1, // <- Número de ferros distribuídos
```

```
// (ICPE1P) Espaçamen + 1 ferro
```

```
// (ICPE1M) Espaçamen - 1 ferro
```

```
// (ICPESP) Espaçamen = n de ferros
```

```
double*faixa, // <- Comprimento da faixa, cm
```

```
double*espac, // <- Espaçamento da faixa, cm
```

```
int*nf, // -> Número de ferros calculado
```

```
int*istat); // -> (!=0) se *espac==0.
```

```
//
```

```
//Adiciona uma faixa de distribuição ao ferro. O número de ferros
```

```
//passará a ser calculado pela soma dos ferros das faixas
```

```
//
```

```
SUBREXPIPO IPOFER_FAIXADIST (
```

```
void**ferobjv, // <- Handle do ferro
```

```
int*icfes1, // <- Número de ferros distribuídos
```

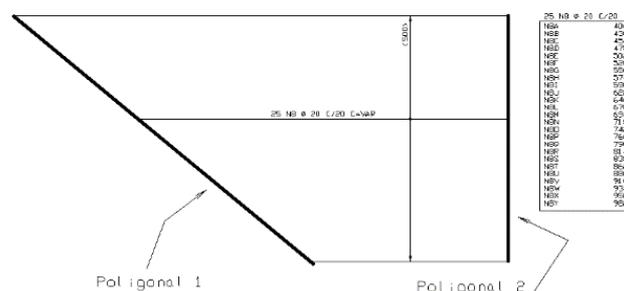
```
// (ICPE1P) Espaçamen + 1 ferro
```

```
// (ICPE1M) Espaçamen - 1 ferro
// (ICPESP) Espaçamen = n de ferros
double*angfai,// <- Âng faixa distrib princ XY glob
// Ortogonal ao ferro
double*xpt1,// <- Pt1 X
double*ypt1,// <- Pt1 Y
double*xpt2,// <- Pt2 X
double*ypt2,// <- Pt2 Y
double*xcot,// <- X de passagem da linha de cotagem
double*ycot,// <- Y de passagem da linha de cotagem
int*ifdcotc,// <- (1) p/cotar compr da faixa
int*iflnfr,// <- (1) descrever número de ferros
int*iflpos,// <- (1) descrever número da posição
int*iflbit,// <- (1) descrever bitola
int*iflesp,// <- (1) descrever espaçamento
int*ilinexten,// <- (1) linha de extensão automática
int*ilinchama,// <- (1) se linha de chamada
int*itpponta,// <- (0) flexa (1) círculo (2) traço
double*espac);// <- Espaçamento cm se dif do ferro
//
//Define cálculo automático do número de ferros da faixa
//
```

```
SUBREXPIPO IPOFER_FAIXDISTCALCNFAUTO (
void**ferobjv,// <- Handle do ferro
int*icalcnfauto);// <- (1) Se calcula núm de ferr autom
```

## Faixas de distribuição variável

Ferros variáveis podem ser definidos de maneira completamente manual ou com ajuda da biblioteca de ferros. No modo mais automático, o programa calcula as quantidades e comprimentos da seguinte maneira:



- Defina dados gerais, incluindo espaçamento

- Defina o cobrimento que será subtraído do contorno por IPOFER\_CFRREB.
- Defina a direção da faixa de distribuição por IPOFER\_ARM\_DIRFEI.
- Defina as poligonais de contorno como na figura acima. Use IPOFER\_ARM\_INITRNSFP seguida de IPOFER\_ARM\_TRNSFP para cada poligonal de contorno.
- Nos ferros genéricos, defina a dobra variável por IPOFER\_FERVAR\_IDOBRAVAR.
- Chame IPOFER\_FAIXADIST para definir a extensão da faixa de distribuição.
- Chame IPOFER\_ARM\_FERVAR para calcular cada comprimento variável e obter um comprimento médio e coordenadas para representar a linha do ferro.
- Chame IPOFER\_LINHAFER para inserir uma linha que representa o ferro no desenho. Para desenhar com o comprimento médio obtido no ferro reto, use IPOFER\_CFRCHO.
- Chame IPOFER\_ARM\_TABVAR se quiser inserir a tabela de comprimentos variáveis no desenho.

Se quiser definir os comprimentos manualmente, deixe de usar IPOFER\_ARM\_INITRNSFP, IPOFER\_ARM\_TRNSFP e IPOFER\_ARM\_FERVAR. Em vez disto, entre cada comprimento com IPOFER\_FERVAR\_ENTRARCOMPRVAR.

```
//
//Inicializa os contornos p/cálculo de ferro variável
//
SUBREXPIPO IPOFER_ARM_INITRNSFP (void);
//
//Acumula mais uma poligonal de contorno
//
SUBREXPIPO IPOFER_ARM_TRNSFP (
double xy [[2], // <- Poligonal a acumular
int *np); // <- Núm de pts da poligonal
//
//Entrada dados de ferro variável
//
SUBREXPIPO IPOFER_FERVAR_IDOBRAVAR (
void**ferobjv, // <- Handle do ferro
int*idobrar); // <- (-1) Não (>=0) Índ dobra var
// (0) p/ferro reto reto
SUBREXPIPO IPOFER_FERVAR_ENTRARCOMPRVAR (
void**ferobjv, // <- Handle do ferro
double*compvar); // <- Um comprimento variável
//
```

```

//Geração automática de ferro variável
//
//Calcula todos os comprs da dobra definida em IPOFER_FERVAR_IDOBRAVAR
//Distribui entre os contornos definidos por IPOFER_ARM_TRNSFP
//A direção da faixa de distrib deve ser definida em IPOFER_ARM_DIRFEI
//
SUBREXPIPO IPOFER_ARM_FERVAR (
void**ferobjv,// <- Handle do ferro
double*xinsvar,// -> Pos X inserção ferro médio
double*yinsvar,// -> Pos y inserção ferro médio
double*compvar,// -> Compr do ferro médio cm
double*anglvar,// -> Ângulo graus do ferro
int *istat);// -> (0) Ok ou:
// (1) contorno inválido
// (2) Número gerado não bate c/espac
// (3) espaçamento zero
//
//Retorna o número de ferros gerados
//
SUBREXPIPO IPOFER_ARM_FERVARNUM (
void**ferobjv,// <- Handle do ferro
int*numfer);// -> Número de ferros gerados
//
//Define manualmente os limites da faixa de distribuição
//
SUBREXPIPO IPOFER_ARM_DEFFAIXA (
double*xmin,// <- X mínimo
double*ymin,// <- Y mínimo
double*xmax,// <- X máximo
double*ymax);// <- Y máximo
//
//Insere tabela de ferros variável no desenho
//
SUBREXPIPO IPOFER_ARM_TABVAR (
void**ferobjv,// <- Handle do ferro

```

```
double*xins,// <- Ponto de inserção, canto esq sup
double*yins);// <- Ponto de inserção, canto esq sup
```

## Faixa de distribuição de múltiplas posições

Faixas para múltiplas posições são faixas independentes, definidas como se fossem um ferro, e depois associadas aos ferros distribuídos na faixa. A criação de uma faixa é como esta:

```
int istat= 0;
void *ferobjvf= NULL;
IPOFER_CRIARFERRO (&pedmv, &ferobjvf, &istat);
if(istat)return;
int icftpf= ICPFAIMUL;
IPOFER_ICFTPF (&ferobjvf, &icftpf);
int icfes1= ICPESP;
IPOFER_FAIXADIST (&ferobjvf, &icfes1, &angfai,
&xpt1, &ypt1, &xpt2, &ypt2, &xcot, &ycot,
&ifdcotc, &iflnfr, &iflpos, &iflbit, &iflesp,
&ilinexten, &ilinchama, &itpponta, &espac);
```

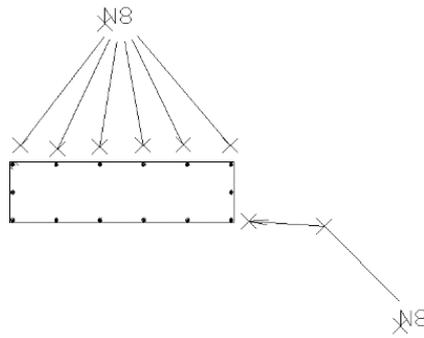
Ou seja: cria-se um ferro com tipo ICPFAIMUL, e define-se a geometria da faixa. Em seguida, os ferros são associados à faixa em chamadas como esta:

```
void *ferobjv= NULL;
int istat= 0;
IPOFER_CRIARFERRO (&pedmv, &ferobjv, &istat);
if(istat)return;
IPOFER_ASSOCFAIXAMULT (&ferobjvf, &ferobjv);
```

## Identificação de posições

Identificação de posições são cotagens que identificam linhas ou ferros em corte no desenho associados a uma posição de ferro. Um ferro pode ter uma ou mais identificações associadas. Se o número da posição do ferro se altera, a identificação também se altera de maneira automática.

O mesmo comando gera identificação com dois formatos diferentes: múltiplas linhas definidas pelos pontos da extremidade ou uma única linha com uma flecha na ponta:



A chamada IPOFER\_IDENTPOS cria a identificação com zero linhas. Para entrar os pontos das múltiplas linhas ou de uma poligonal, usamos IPOFER\_IDENTPOS\_ENTRARPT.

```

SUBREXPIPO IPOFER_IDENTPOS (
void**ferobjv,// <- Handle do ferro
int*imultipla,// <- (0) polig+flexa (1) múltiplas lin
double*xtex,// <- Posição do texto de identificação
double*ytex,// <- "
int*iflifr,// <- (1) descrever número de ferros
int*iflpos,// <- (1) descrever número da posição
int*iflbit,// <- (1) descrever bitola
int*iflesp);// <- (1) descrever espaçamento
//
//Entra novo ponto de linha cotagem. O primeiro está na posição de texto
//
SUBREXPIPO IPOFER_IDENTPOS_ENTRARPT (
void**ferobjv,// <- Handle do ferro
double*xid,// <- Ponta da poligonal ou linha
double*yid);// <- "

```

## Extensão do DP para gravação de ferros

Para adaptação dos desenhos de fundação usando o DP, fica definida a seguinte extensão:

- Um ferro é definido pelo bloco:

IPOFER

FIM

- A rotina IPOFER\_CRIARFERRO é chamada automaticamente dentro do bloco IPOFER.

- As chamadas às rotinas de IPOFER são feitas da mesma maneira que em C++, com as seguintes regras:

Coloca-se o nome da rotina sem o prefixo IPOFER\_.

Todos os caracteres "\_" são retirados da chamada

Os nomes das rotinas não tem abreviação.

Definem-se todos os parâmetros da respectiva rotina, com exceção do primeiro, ferobjv, que é implícito.

Usam-se espaços em branco para separar parâmetros, e não vírgulas.

Parâmetros que retornam valores não são declarados.

As constantes definidas em IPOFERX.H estão disponíveis para uso dentro do DP, com o prefixo "\_".

Exemplo de criação de um ferro em DP:

```
DESENHO 'TESTE'
```

```
IPOFER
```

```
ICFTPF %_ICPFRT
```

```
ICFIPN %_ICPPOS
```

```
ICFPOS 1
```

```
ICFNFR 10
```

```
CFRBIT 12.5
```

```
CFRESP 20.
```

```
CFRCHO 500.
```

```
CFRDES 15.
```

```
CFRDED 15.
```

```
LINHAFER 1000. 1000. 90. 1. 1 0 0 -1 -1 -1
```

```
FIM
```

```
FIM
```

### Usando o handle FEROBJV no DP

É possível descrever o ferro em partes, armazenando o handle de um ferro para uso posterior. Basta criar um nome arbitrário de variável (até 8 caracteres) e defini-lo na declaração de IPOFER:

```
DESENHO 'TESTE'
```

```
IPOFER FEROBJV
```

```
.....
```

```
FIM
```

```
IPOFER
```

```
.....
```

FIM

IPOFER FEROBJV

.....

FIM

FIM

Neste exemplo criamos a variável FEROBJV que conterà o handle para o ferro descrito na mesma seção IPOFER. Na próxima seção com a declaração da mesma variável, nenhum ferro novo será criado, e todas as declarações modificarão o ferro existente definido por FEROBJV.