

## Ferros Inteligentes

Ferros inteligentes são objetos dentro de um desenho DWG, mas a lógica de criação difere da dos demais elementos gráficos. É uma característica dos chamados "objetos inteligentes de edição gráfica", que são DLLs a parte, com lógica própria.

Nos elementos gráficos comuns, chamamos um método de um objeto `Dwg` para que o elemento gráfico seja criado dentro do desenho. Todos os atributos são assumidos (atributos atuais) ou passados como parâmetro da chamada. Por exemplo,

```
dwg.draw.Line (500, 300, 0, 300)
```

cria uma linha com coordenadas fornecidas, e nível, estilo e cor com valores atuais. Ferros inteligentes por sua vez são criados como um objeto independente e vazio:

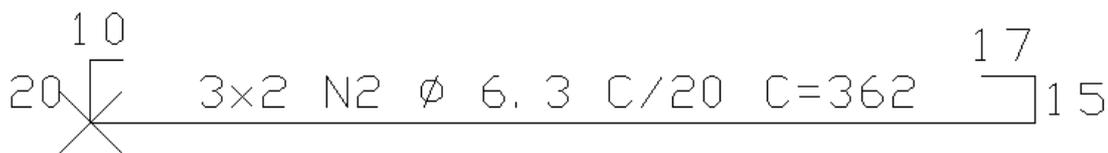
```
rebar = TQSDwg.SmartRebar (dwg)
```

Criamos acima o objeto `rebar`, instância da classe `TQSDwg.SmartRebar`, que é um ferro inteligente sem nenhuma informação. Os métodos e atributos para definir um ferro serão mostrados em seguida. Cada ferro diferente deve ter uma instância diferente da classe `TQSDwg.SmartRebar`.

## Nomenclatura de ferros

Para entender os dados a fornecer, vamos ilustrar a nomenclatura usada. 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



Consistem 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.

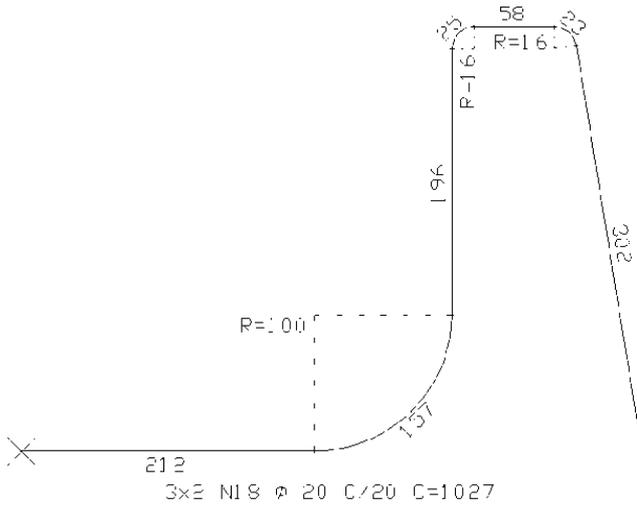
"C/20" é o espaçamento. Se um ferro não é distribuído, o valor do espaçamento deve ser zero.

"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 suas faces externas, e as dobras são as medidas por suas faces externas.

## Ferros genéricos

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

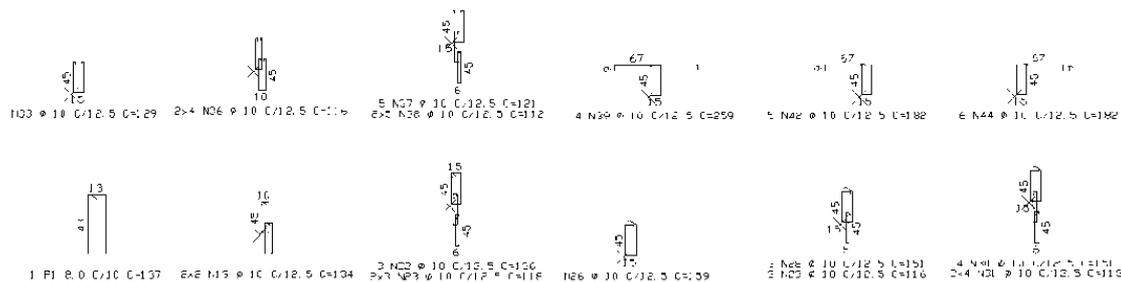


Ferros genéricos são definidos por seu eixo.

## Estribos de vigas



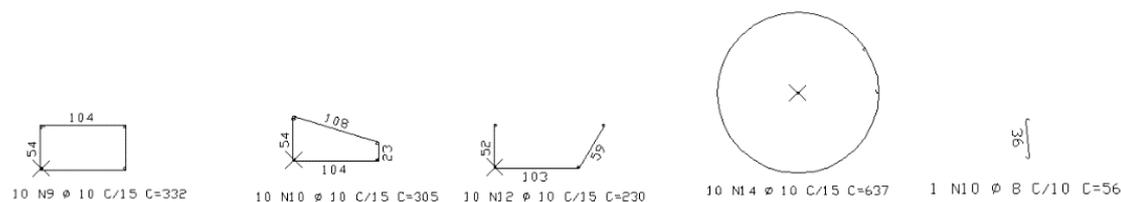
Estribos inseridos por um canto da seção transversal. O cobrimento para posicionar o estribo é armazenado junto com o elemento. Estribos podem ser normais, fechados (para torção) ou abertos, e ainda ter a pata superior alongada para servir como negativo na laje. Estes estribos podem combinar com o número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



## Estribos genéricos (de pilares)

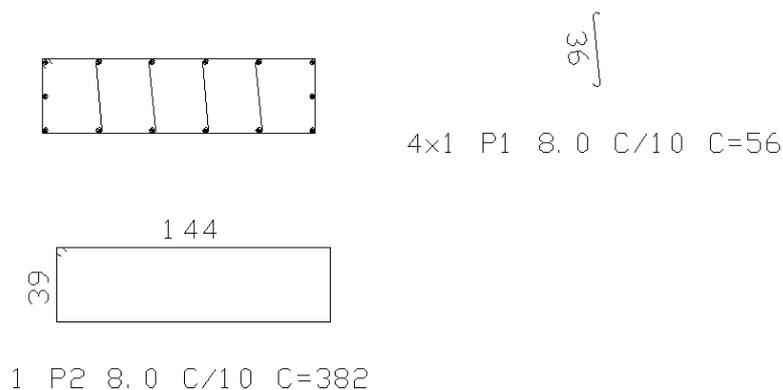
Estribos de pilares têm como referência os pontos de centro das armaduras longitudinais abraçados pelo estribo.

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, independentemente da quantidade final ou multiplicador do número de ferros. O 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 quatro 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 existem armaduras longitudinais que devem 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.

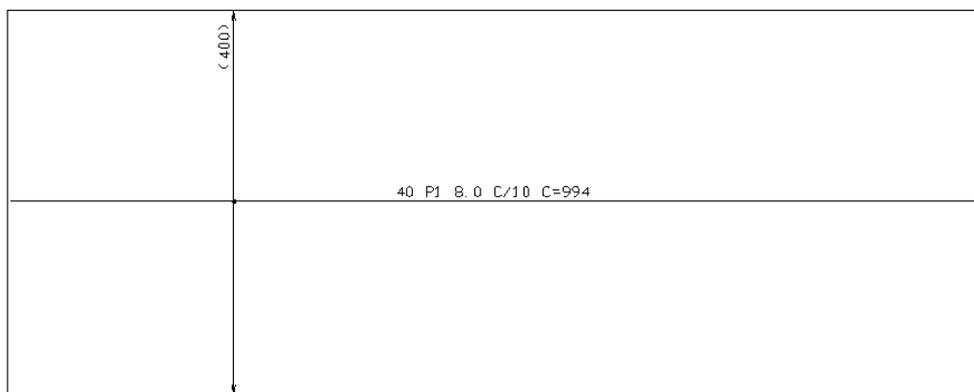
## Ganchos

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.



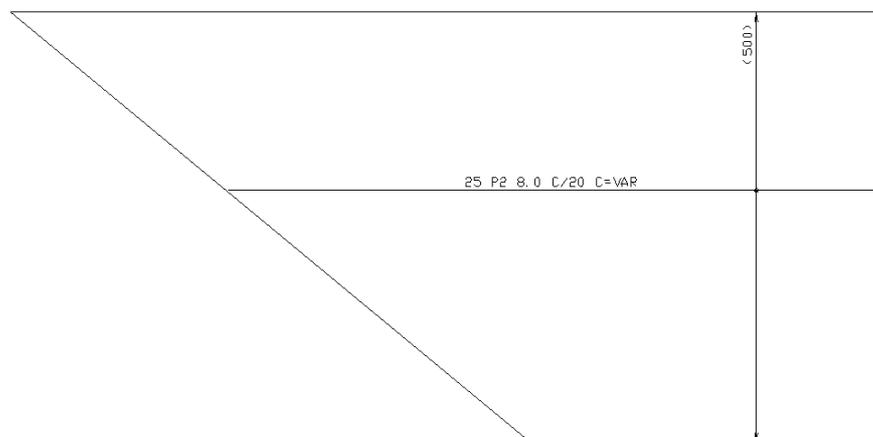
## Faixas de distribuição

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.

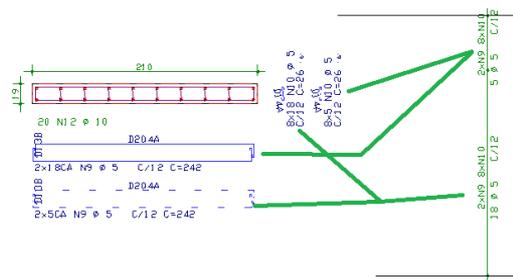
### Faixas de distribuição com comprimento variável



25_P2_8.0_C/20	
P2A	406
P2B	430
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

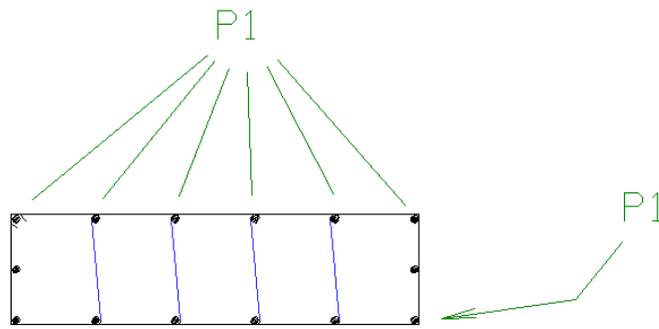


As faixas de distribuição de vigas e lajes são armazenadas junto com os dados do ferro. Isto é, podemos ter um ferro com múltiplas faixas. Faixas de múltiplas posições tem uma relação diferente: vários ferros podem ser associados a uma única faixa. Um exemplo são pilares, onde uma faixa de distribuição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

Faixas de múltiplas posições são criadas independentemente dos ferros, e depois associadas 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

## Atributos assumidos e arquivo de critérios

Diversos critérios e tabelas de bitolas são definidas no arquivo de critérios DESARM.DAT, sendo editado no gerenciador, na aba "Edifício", "Critérios gerais, Aço e desenhos de armação". Muitos destes critérios podem ser definidos por ferro no programa. Nos critérios não definidos, o programa assume como padrão os valores do arquivo de critérios mais próximo.

## Definição de ferros

Como mostramos, um ferro é criado na memória e depois definido:

```
rebar = TQSDwg.SmartRebar (dwg)
```

Como parâmetro da criação, passamos o objeto `TQSDwg` do desenho onde o ferro está sendo inserido. Após a chamada, o ferro já está no desenho, faltam agora seus dados. Você pode manter múltiplos ferros abertos e guardar os objetos de cada um para associar atributos posteriormente. Estes objetos são válidos enquanto o desenho representado por `dwg` estiver aberto.

As constantes que podem ser fornecidas nas chamadas serão mostradas na primeira vez em que aparecerem.

Os métodos e propriedades mostrados a seguir devem ser chamados com o objeto do ferro, e pertencem à classe `TQSDwg.SmartRebar`.

## Dados gerais

A primeira informação a ser fornecida com o ferro é o seu tipo:

**type**

Tipo do ferro, uma das constantes abaixo:

```
TQSDwg .ICPFRT
```

Ferro reto.

```
TQSDwg .ICPFGN
```

Ferro genérico.

```
TQSDwg .ICPSTR
```

Estribo.

```
TQSDwg .ICPGRA
```

Grampo de vigas.

```
TQSDwg .ICPSTRGEN
```

Estribo genérico, pilar.

TQSDwg .ICPFAIMUL

Faixa múltipla – a ser associada com outros ferros.

O fornecimento dos demais dados de ferros deve ser compatível com o tipo definido. O programa não verifica se a definição é inconsistente, e o comportamento poderá ser indefinido. A seguir, os outros dados comuns:

**mark**

Número da posição do ferro

**quantity**

Número de ferros

**multiplier**

Multiplicador de ferros

**diameter**

Bitola do ferro, mm

**spacing**

Espaçamento de ferros, cm

**ribbed**

(0) Em geral

(1) Em lajes nervuradas

**showRibbed**

(0) Não (1) Mostrar status de laje nervurada (C/NERV)

**comment**

Comentário de um ferro (texto que aparece com a descrição do ferro)

**leftHook**

Tipo de gancho à esquerda, uma das constantes a seguir:

TQSDwg .ICPSGA

Sem gancho

TQSDwg .ICP090

Gancho a 90 graus

TQSDwg .ICP135

Gancho a 135 graus

TQSDwg .ICP180

Gancho a 180 graus

**leftHookInvert**

(1) Gancho à esquerda invertido

**rightHook**

Tipo de gancho definido à direita. Mesmas constantes do gancho à esquerda.

## rightHookInvert

(1) Gancho à direita invertido

### columnLevel

Número de lance de pilar

### cover

Cobrimento, cm. Usado para remontar estribos tipo `ICPENR/ICPEFC/ICPEAB/ICPENC`.

## Linha de ferro

Todo ferro é representado por uma ou mais linhas, cada uma inserida no desenho pela chamada abaixo.

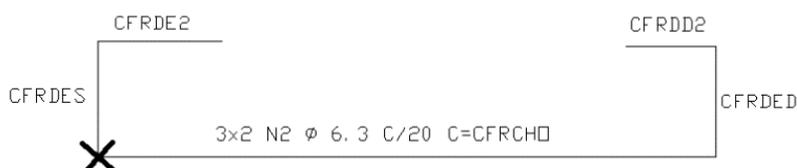
**RebarLine** (`xins,yins,angle,scale,identify,identifyBends,ipatas,iexplode,ilevel,iestilo,icolor`)

Entra linha de ferro de qualquer tipo com dados atuais. Um mesmo ferro pode ser representado por mais de uma linha no desenho.

<code>xins</code>	Ponto de inserção
<code>yins</code>	Ponto de inserção
<code>angle</code>	Ângulo de inserção graus
<code>scale</code>	Escala de inserção
<code>identify</code>	(1) Identificar o ferro
<code>identifyBends</code>	(1) Identificar dobras
<code>ipatas</code>	0 não 1 sim 2 45° 3 225° 4 invertido 0 e 1 valem para <code>ICPFRT</code> , <code>ICPSTR</code> , <code>ICPSTRGEN</code> e <code>ICPGRA</code> 2, 3 e 4 valem para <code>ICPFRT</code>
<code>iexplode</code>	(1) Explodir se estribo
<code>ilevel</code>	Nível 0..255 EAG (-1) padrão
<code>iestilo</code>	Estilo 0..5 EAG (-1) padrão
<code>icolor</code>	Cor 0..255 EAG (-1) padrão

## Ferros retos

São definidos pelos comprimentos das dobras.



### straightBarMainLength

Comprimento principal de ferro reto, cm

### **straightBarRightLength**

Comprimento da dobra esquerda de ferro reto, cm

### **straightBarRightLength2**

Comprimento da 2a dobra direita de ferro reto, cm

### **straightBarLeftLength**

Comprimento da dobra esquerda de ferro reto, cm

### **straightBarLeftLength2**

Comprimento da 2a dobra esquerda de ferro reto, cm

### **straightBarZone**

Posição de ferro reto, definida por uma das constantes:

TQSDwg .ICPPOS

Armadura positiva (face inferior)

TQSDwg .ICPNEG

Armadura negativa (face superior)

### **straightBarBendType**

Tipo de dobra de um ferro reto, definida por uma das constantes:

TQSDwg .ICPNSU

Dobra normal

TQSDwg .ICPDSU

Dobra de suspensão

TQSDwg .ICPDS2

Dobra de suspensão do mesmo lado

### **straightBarTextDirection**

Direção dos textos de dobras:

(0): Direção do trecho

(1): Direção do ferro

### **straightBarTextPosition**

Texto principal de ferros:

(0) Não colocar

(1) Acima da linha de ferro

(2) Abaixo da linha de ferro

### **straightBarContinuous**

Ferro corrido: (0) Não (1) Sim

### **startCoupler**

Luva no início da barra (0) Não (1) Sim

`endCoupler`

Luva no fim da barra (0) Não (1) Sim

`mirrorMode`

Espelhamento (0) restrito (1) completo

`alternatingMode`

Alternância de ferros retos, conforme as constantes:

`TQSDwg .ICPSAL`

Não

`TQSDwg. ICPCAL`

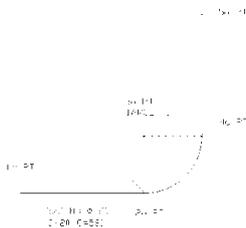
Sim

`alternatingFactor`

Fator de alternância de ferro reto

## 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).



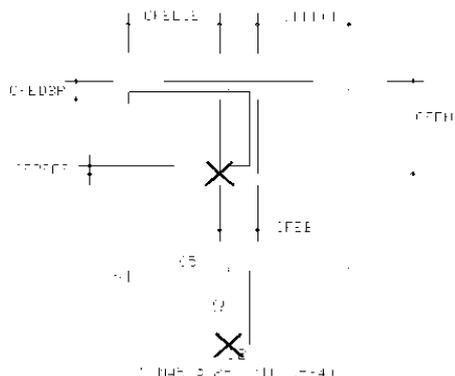
`GenRebarPoint` (`xpt`, `ypt`, `zpt`, `iarc`, `identifyBends`, `indfrt`)

Entra com um ponto de uma poligonal de ferro genérico. A poligonal será transformada na inserção em `RebarLine`. O Índice de trecho principal será o trecho variável caso o ferro seja distribuído com variação.

<code>xpt</code>	Ponto de inserção
<code>ypt</code>	Ponto de inserção
<code>zpt</code>	Ponto de inserção (ignorado)
<code>iarc</code>	(0) trecho reto (1) centro de arco
<code>identify</code>	(1) Identificar o ferro
<code>identbend</code>	(1) Identificar a dobra
<code>indfrt</code>	Índice de trecho principal (-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_RebarLine` nas posições mostradas abaixo.



### **stirrupType**

Tipo de estribo de vigas, uma das constantes abaixo:

TQSDwg `.ICPENR`

Normal

TQSDwg `.ICPEFC`

Fechado

TQSDwg `.ICPEAB`

Aberto

TQSDwg `.ICPENC`

Normal com largura colaborante

### **stirrupLegs**

Número de ramos de estribos, uma das constantes abaixo:

TQSDwg `.ICPNR2`

Estribo de 2 ramos

TQSDwg `.ICPNR4`

Estribo de 4 ramos

TQSDwg `.ICPNR6`

Estribo de 6 ramos

TQSDwg `.ICPNR4B`

Estribo de 4 ramos config B

### **stirrupSectionWidth**

Largura da seção transversal que contém o estribo, cm

### **stirrupSectionHeight**

Altura da seção transversal que contém o estribo, cm

### **stirrupSectionWidth2**

Largura final da seção transversal variável que contém o estribo, cm

## **stirrupSectionHeight2**

Altura final da seção transversal variável que contém o estribo, cm

## **stirrupEffectiveLeftWidth**

Largura esquerda colaborante da seção transversal que contém o estribo, cm

## **stirrupEffectiveRightWidth**

Largura direita colaborante da seção transversal que contém o estribo, cm

## **stirrupSlabBendLength**

Largura direita colaborante da seção transversal que contém o estribo, cm

## **stirrupHookLength**

Comprimento da pata de estribos em número de bitolas

## **stirrupHookType**

Tipo de pata de estribo, uma das constantes a seguir:

TQSDwg **.ICPTPPATA45**

Pata de estribo a 45 graus

TQSDwg **.ICPTPPATA90**

Pata de estribo a 90 graus

## **stirrupInternalLegDiameter**

Bitola em mm diferente para estribo interno de 4 ou 6 ramos

## Grampos de vigas

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



## **crosstiesDirection**

Direção do grampo de vigas (1) esquerda (-1) direita

## **crosstiesLongLength**

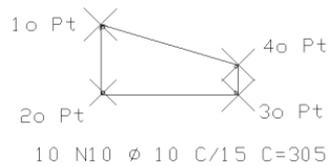
Comprimento longitudinal do grampo de vigas, em cm

## **crosstiesTransvLength**

Comprimento transversal do grampo de vigas, em

## 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 `genericStirrupLongDiameter`. O tipo de estribo é definido em `genericStirrupType`. Os pontos podem ser definidos por `GenRebarPoint`, ou pela `GenericStirrupPoint`, mas simples. Em estribos fechados, o ponto de fechamento não deve ser fornecido.



### **genericStirrupType**

Tipo de estribo genérico, conforme uma das constantes:

TQSDwg .**ICPEGENFEC**

Estribo genérico fechado

TQSDwg .**ICPEGENABR**

Estribo genérico aberto

TQSDwg .**ICPEGENGRA**

Grampo de pilar

TQSDwg .**ICPEGENCIR**

Estribo genérico circular

### **genericStirrupEntryMode**

Modo de definição de estribo genérico, uma das constantes:

TQSDwg .**ICPEGPONTOSLONG**

Pontos longitudinais

TQSDwg .**ICPEGPONTOSSECA**

Pontos da seção longitudinal

TQSDwg .**ICPEGPONTOSEXTR**

Pontos externos

### **genericStirrupLongDiameter**

Bitola longitudinal mm de referência para a definição dos pontos

**GenericStirrupPoint** (xlong, ylong)

Entrada de um ponto de estribo genérico. Os pontos são fornecidos no centro da armadura longitudinal. A bitola de armadura longitudinal é definida por **genericStirrupLongDiameter**. No estribo fechado, não fornecer o último ponto.

### **circularStirrupRadius**

Raio de estribo circular, cm

**StirrupAdditionalCriteria** (compradgra, igrasck83, igrak77)

Diversos critérios que modificam um grampo de pilar:

compradgra	Somar valor em cm ao comprimento do grampo de pilar
igrasck83	Desenho de grampo (0) em S ou (1) C

## 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.

```
RebarTextDisplaySelection (iflnfr, iflpos, iflbit, iflesp, iflcmp,
ialign, ibreak)
```

Seleciona os textos de identificação a serem mostrados:

<code>iflnfr</code>	(1) Número de ferros
<code>iflpos</code>	(1) Número da posição
<code>iflbit</code>	(1) Bitola
<code>iflesp</code>	(1) Espaçamento
<code>iflcmp</code>	Comprimento
<code>ialign</code>	Alinhamento – ICPCENTR_xxx abaixo
<code>ibreak</code>	Salto de linha – ICPQUEBR_xxx abaixo

```
TQSDwg .ICPCENTR_CENTRAD
```

Centrado

```
TQSDwg .ICPCENTR_ESQUERD
```

Esquerda

```
TQSDwg .ICPCENTR_DIREITA
```

Direita

```
TQSDwg .ICPQUEBR_SEMQUEBRA
```

Sem quebra

```
TQSDwg .ICPQUEBR_SALTOCBCI
```

Salto no espaçamento ou comprimento C/ ou C=

```
TQSDwg .ICPQUEBR_SALTOBITO
```

Salto na posição da bitola

```
TQSDwg .ICPQUEBR_SALTODECD
```

Salto no espaçamento C/

```
TQSDwg .ICPQUEBR_SALTONPOS
```

Salto no número de ferros

```
RebarTextDisplayPosition (xtex, ytex, htex, angle, imod, ialign, ibreak)
```

Fornece a posição e formatação do texto de identificação

<code>xtex</code>	Coordenadas do texto
<code>ytex</code>	Coordenadas do texto
<code>htex</code>	Altura de texto cm plotados
<code>angle</code>	Ângulo do texto em graus
<code>imod</code>	(0) normal(1) bitola/espaçamento para tabela variável(2) posição(3) posição centrada, levanta o círculo da posição
<code>ialign</code>	Alinhamento – ICPCENTR_xxx acima
<code>ibreak</code>	Salto de linha – ICPQUEBR_xxx acima

## Reposicionando textos de identificação existentes

Textos de identificação são posicionados em posição padrão, mas podem ter posição alterada pela rotina

`RebarTextDisplayPosition` ou interativamente por edição gráfica. Para reposicionar textos de identificação nas posições padrão, use a rotina:

**`ResetTextPosition`** ()

Reposiciona os textos identificadores do ferro em posição padrão

## Leitura de linha de ferro

Estas rotinas permitem tanto a leitura da linha de ferro desenvolvida em coordenadas absolutas e seus atributos, quanto a linha de um ferro genérico no sistema local, com descrição de arcos.

**`GetInsertionNumber`** ()

Retorna o número de linhas inseridas associadas a um ferro

`GetInsertionData` (**`indexins`**)

Retorna dados de uma linha inserida

`indexins` Índice da linha inserida, 0..`GetInsertionNumber()`-1

Retorna:

`xins` Ponto de inserção

`yins` Ponto de inserção

`angins` Ângulo de inserção

`escxy` Escala de inserção

`identfer` (1) Identificar o ferro

`identdobr` (1) Identifica dobras

`ipatas` (0) não (1) sim (2) a 45°

`iexplodir` (1) Se representação explodida

`inivel` Nível EAG (-1) default

`iestilo` Estilo EAG (-1) default

`icor` Cor EAG (-1) default

**GetInsertionPoints** (`indexins`)

Retorna o número de pontos de uma inserção

`indexins` Índice da linha inserida, 0..GetInsertionNumber()-1

Retorna:

`np` Número de pontos da linha de ferro

`istat` (!=0) se não leu

**GetInsertionPoint** (`indexins`, `ipt`)

Retorna as coordenadas de um ponto de um ferro inserido

`indexins` Índice da linha inserida, 0..GetInsertionNumber()-1

`ipt` Ponto a ler 0..np-1

Retorna:

`X` Abcissa X

`Y` Ordenada Y

`istat` (!=0) se não leu

**GetGenRebarPoints** ()

Retorna o número de pontos da poligonal de ferro genérico. Retorna zero para todos os outros tipos de ferros.

**GetGenRebarPoint** (`indexpoint`)

Retorna um ponto de ferro genérico

`indexpoint` Índice do ponto a ler, 0..GetGenRebarPoints()-1

Retorna:

`xpt` X

`ypt` Y

`zpt` Z

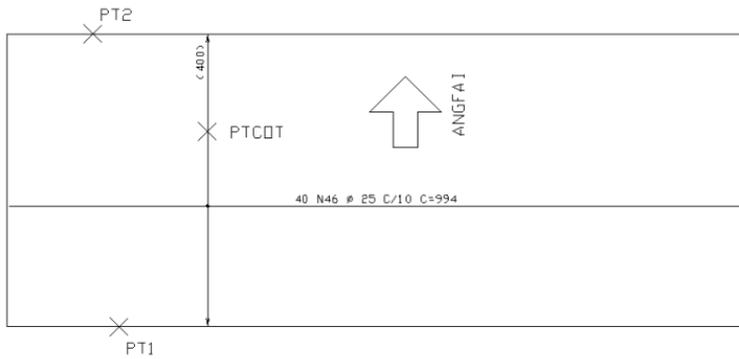
`iarco` (1) se centro de arco

`identdobr` (1) p/identificar a dobra

`indfrt` Índice de trecho equiv ou (-1)

## Faixas de distribuição

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



### RebarDistrQuant (icfes1, distrLength, spacing)

Cálculo do número de ferros dado comprimento da faixa e espaçamento. Retorna o número de ferros calculado e status (!=0 se espaçamento zero).

icfes1	Número de ferros em função dos espaçamentos, conforme constantes TQSDwg.ICPxxx abaixo.
distrLength	Comprimento da faixa de distribuição, cm
spacing	Espaçamento da faixa de distribuição, cm

TQSDwg .ICPE1P

Espaçamentos + 1 ferro

TQSDwg .ICPE1M

Espaçamentos - 1 ferro

TQSDwg .ICPESP

Espaçamentos = número de ferros

**RebarDistrAdd** (icfes1, angdist, xpt1, ypt1, xpt2, ypt2, xcot, ycot, ifdcotc, iflnfr, iflpos, iflbit, iflesp, ialign, ibreak, ordem, k32vigas, k41vigas, ilinexten, ilinchama, itpponta, spacing, scale)

Adiciona uma faixa de distribuição ao ferro. Temos dois tipos de faixas de distribuição:

Embutida em um ferro. Um ferro pode ter várias faixas. Não é usada em pilares. Uma faixa pertence a um ferro. Independente do ferro. Uma faixa pode ter vários ferros. Usadas em pilares, onde estribos e grampos partilham faixa única. Uma faixa é definida primeiro como um ferro tipo TQSDwg.ICPFAIMUL. Depois são definidos os dados de faixas ( RebartDistAdd) e depois associados os ferros ligados à faixa através da rotina RebarDistrLink. Uma vez que um ferro tem uma ou mais faixas associadas, o número de ferros passa a ser calculado pela soma dos ferros das faixas.

icfes1	Número de ferros em função dos espaçamentos, conforme constantes TQSDwg.ICPxxx
angdist	Ângulo em graus da faixa de distribuição ortogonal ao ferro
xpt1	Ponto 1 da faixa
ypt1	Ponto 1 da faixa

xpt2	Ponto 2 da faixa
ypt2	Ponto 2 da faixa
xcot	X de passagem da linha de cotagem
ycot	Y de passagem da linha de cotagem
ifdcotc	(1) p/cotar compr da faixa
iflnfr	(1) descrever número de ferros
iflpos	(1) descrever número da posição
iflbit	(1) descrever bitola
iflesp	(1) descrever espaçamento
ialign	Alinhamento TQSDwg.ICPCENTR_XXXXXXX
ibreak	Salto de linha TQSDwg.ICPQUEBR_XXXXXXXXXX
ordem	Ordem dos textos (" padrão) "N" Número de ferros "n" só o número "M" só o multiplicador "P" Posição "B" Bitola "E" Espaçamento "C" Comprimento "F" Comprimento de faixa
k32vigas	Critério K32 CAD/Vigas p/m_ordem=="
ilnexten	(1) linha de extensão automática
ilinchama	(1) se linha de chamada
itpponta	(0) flexa (1) círculo (2) traço
spacing	Espaçamento cm se diferente do ferro
scale	Escala de inserção (multiplica todas as dimensões)

## Leitura de faixa de distribuição

Os mesmos dados usados na criação de faixas associadas a um ferro podem ser lidos.

**GetRebarDistrNum** ()

Retorna o número de faixas de distribuição

**GetRebarDistrInfo** (ifaixa)

Retorna dados da faixa de distribuição ifaixa == 0..GetRebarDistrNum()-1

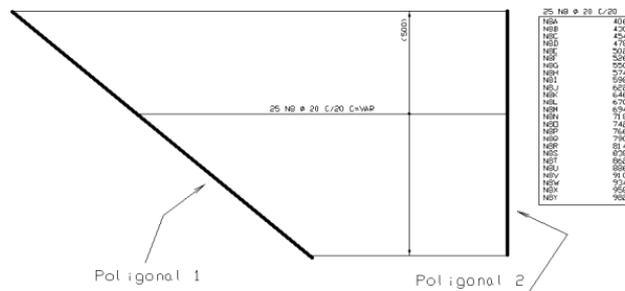
Retorna:

icfes1	Número de ferros distribuídos
angfai	Ângulo da faixa de distribuição sistema global

xpt1	Ponto 1 da faixa
ypt1	Ponto 1 da faixa
xpt2	Ponto 2 da faixa
ypt2	Ponto 2 da faixa
xcot	X de passagem da linha de cotagem
ycot	Y de passagem da linha de cotagem
ifdcotc	(1) p/cotar comprimento da faixa
iflnfr	(1) descrever número de ferros
iflpos	(1) descrever número da posição
iflbit	(1) descrever bitola
iflesp	(1) descrever espaçamento
icentr	Alinhamento TQSDwg.ICPCENTR_XXXXXXX
iquebr	Salto de linha TQSDwg.ICPQUEBR_XXXXXXXXXX
ordem	Ordem dos textos (" padrão) "N" Número de ferros "n" só o número "M" só o multiplicador "P" Posição "B" Bitola "E" Espaçamento "C" Comprimento "F" Comprimento de faixa
k32vigas	Critério K32 CAD/Vigas p/m_ordem=="(0) K32=0 (1) K32=1 (2) K32=2
k41vigas	Critério K41 CAD/Vigas p/m_ordem=="(0) ã cotar faixa (1) cotar (2) cota+muda Pn
ilinexten	(1) linha de extensão automática
ilinchama	(1) se linha de chamada
itpponta	(0) flexa (1) círculo (2) traço
espac	Espaçamento cm se diferente do ferro
escxy	Escala de inserção (multiplica dimensões)

## 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 `cover`.
- Defina as poligonais de contorno como na figura acima. Use `VariableRebarContoursInit`, `VariableRebarOneContourInit`, `VariableRebarOneContourPoint` e `VariableRebarOneContourEnd` para a definição dos contornos.
- Nos ferros genéricos, defina a dobra variável por `VariableRebarMainSegIndex`.
- Chame `RebarDistrAdd` para definir a extensão da faixa de distribuição.
- Chame `VariableRebarGenerate` para calcular cada comprimento variável e obter um comprimento médio e coordenadas para representar a linha do ferro.
- Chame `RebarLine` para inserir uma linha que representa o ferro no desenho. Para desenhar com o comprimento médio obtido no ferro reto, use `straightBarMainLength`.
- Chame `VariableRebarScheduleInsert` se quiser inserir a tabela de comprimentos variáveis no desenho.

**VariableRebarContoursInit** ()

Inicializa lista de contornos para a definição de ferros variáveis.

**VariableRebarOneContourInit**()

Inicializa um novo contorno atual para ferros variáveis.

**VariableRebarOneContourPoint** (x, y)

Entra um ponto no contorno atual

**VariableRebarOneContourEnd** ()

Finaliza a entrada de pontos no contorno atual.

**VariableRebarMainSegIndex** (idobraravar)

Define o índice da dobra variável (dobras numeradas a partir de zero).

**VariableRebarGenerate** ( icfes1, angdist, scale)

Geração automática de ferros variáveis.

`icfes1` Número de ferros distribuídos: `TQSDwg.ICPE1P/ICPE1M/ICPESP`

`angdist` Ângulo em graus da faixa de distribuição ortogonal ao ferro

`scale` Escala de inserção

Retorna:

`xins` Posição X média de inserção do ferro

`yins` Posição y inserção ferro médio

`meanLengthCompr` do ferro médio cm

`angle` Ângulo do ferro em graus

`istat(0)` Ok

(1) Contorno inválido

(2) O número gerado não bate com o número de espaçamentos

(3) Espaçamento zero

**VariableRebarQuantity** ()

Retorna o número de ferros variáveis gerados.

**VariableRebarScheduleInsert** (xins,yins)

Insere tabela de ferros variável no desenho.

**VariableStirrupGenerate** ()

Gera a lista de comprimentos variáveis de estribos.

Somente estribos retangulares (estribos de vigas).

Somente na largura ou altura

Variação entre **stirrupSectionWidth** e **stirrupSectionWidth2** ou **stirrupSectionHeight** e **stirrupSectionHeight2**

A quantidade de ferros é calculada pela faixa ou por número fixo.

Se não tem faixa definida, `icfes1==TQSDwg.ICPE1P`.

## 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:

```
rebar = TQSDwg.SmartRebar (dwg)
rebar.type = TQSDwg.ICPFAIMUL
rebar.RebarDistrAdd (TQSDwg.ICFESP, angdist, xpt1, ypt1, xpt2, ypt2, xcot,
ycot, ifdcotc, iflnfr, iflpos, iflbit, iflesp, ialign, ibreak, ordem, k32vigas,
k41vigas, ilinexten, ilinchama, itponta, spacing, scale)
```

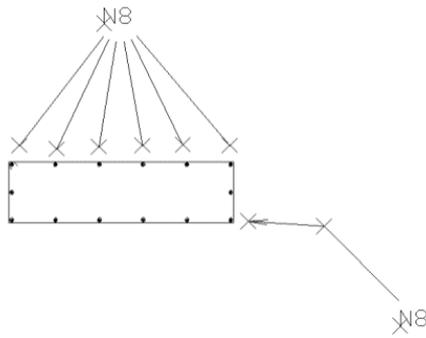
Ou seja: cria-se um ferro com tipo `TQSDwg.ICPFAIMUL`, e define-se a geometria da faixa. Em seguida, os ferros são associados à faixa em chamadas como esta:

```
rebar2 = TQSDwg.SmartRebar (dwg)
rebar.RebarDistrLink (rebar2)
```

## 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 função `RebarMarkIdentify` cria a identificação com zero linhas. Para entrar os pontos das múltiplas linhas ou de uma poligonal, usamos `RebarMarkIdentifyPoint`.

**RebarMarkIdentify** (`imultiple`, `xtex`, `ytex`, `iflnfr`, `iflpos`, `iflbit`, `iflesp`)

Identificação de uma posição de ferro. Uma ou mais linhas de chamada com um texto de descrição de ferro. Cada chamada a `RebarMarkIdentify` cria uma nova identificação. As chamadas `RebarMarkIdentifyPoint` em seguida se referem à última identificação criada.

```

10 P1
+ 2+-->3
| /
+---+
0 1
(imultiple == 0)

10 P1
+ + + +
/ | \ \
/ | \ \
0 1 2 3
(imultiple == 1)

```

`imultiple` poligonal e flexa (1) múltiplas linhas

`xtex` Posição do texto de identificação

`ytex` Posição do texto de identificação

`iflnfr` (1) descrever número de ferros

`iflpos` (1) descrever número da posição

`iflbit` (1) descrever bitola

`iflesp` (1) descrever espaçamento

**RebarMarkIdentifyPoint** (`xid`, `yid`)

Entra novo ponto de linha cotagem. O primeiro está na posição de texto.

## Cotagem de ponta de ferro

É uma cotagem automática colocada em uma ou duas pontas de uma linha de ferro.

**DimRebarEndPoint** (`insindex`, `idimpoint`, `xpt`, `ypt`):

Cotagem de uma ponta de ferro, com o índice (0..n-1) da linha do ferro.

`insindex` Índice da da linha a cotar (0..)

`idimpoint` (0) Não (1) cotar 1o ponto (2) cotar pontos 1 e 2

`xpt`X

`ypt`Y

## Linhas adicionais de ferro

São linhas arbitrárias associadas a um ferro, são movimentadas e apagadas junto com o ferro. Não precisam ter relação com o ferro, e não tem comprimento alterado com ele.

**AdditionalLineInit** (*inivel, iestilo, icor*)

Abre linha adicional de ferro. Os pontos serão definidos por `AdditionalLineInitPoint`. Para abrir nova linha, chame novamente esta rotina.

*inivel* Nível EAG (-1) default

*iestilo* Estilo EAG (-1) default

*icor* Cor EAG (-1) default

**AdditionalLineInitPoint** (*xpt, ypt, zpt, iarco*)

Entra ponto na linha adicional aberta.

*xpt*X

*ypt*Y

*zpt*Z (atualmente sem uso)

*iarco*(1) se centro de arco

## Duplicação de um ferro

Você pode duplicar um ferro existente e alterar seus atributos para facilitar a programação.

**DeepCopy** ()

Retorna uma cópia SmartRebar do objeto atual

## Controle de posições e o objeto TQSDwg. GlobalRebar

Embora o número da posição de um ferro possa ser atribuído, o mais fácil é usar a renumeração de ferros que atua no desenho inteiro, ordena as posições de cima para baixo e da direita para a esquerda, e agrupa posições iguais.

O objeto do tipo `TQSDwg` tem o subobjeto `globalrebar` da classe `TQSDwg.GlobalRebar`, que trata os ferros do desenho como um conjunto. Por exemplo, o número da primeira posição livre de ferros para ser usada é obtida com:

```
iprimpos = dwg.globalrebar.FreeMark
```

**firstMark**

Primeira posição para renumeração.

**RenumerateMarks** ()

Renumeração de posições, a partir da primeira posição definida em `firstMark`.

**FreeMark** ()

Retorna próximo número de posição livre para uso

## Iteração com ferros inteligentes

É possível ler o conteúdo de todos os ferros de um desenho, até mesmo gerar uma tabela de ferros. Para isto, itera-se no desenho até encontrar ferros, e lista-se os atributos de cada um.

Para identificar um ferro no desenho, é necessário saber que é um elemento gráfico tipo

TQSDwg.DWGTYPE\_OBJECT, e que o nome do objeto (também sua DLL) é " IPOFER". A rotina RebarScheduleInfo facilita a leitura dos dados do ferro, somando quantidades e comprimentos de acordo com os critérios de aplicação de raios de curvatura. Uma iteração típica seria como esta:

```
dwg.iterator.Begin ()
while True:
itipo = dwg.iterator.Next ()
if itipo == TQSDwg.DWGTYPE_EOF:
break
if itipo != TQSDwg.DWGTYPE_OBJECT:
continue
if dwg.iterator.objectName != "IPOFER":
continue
rebar = dwg.iterator.smartRebar
numdescr = rebar.RebarScheduleNumDescr ()
for idescr in range (0, numdescr):
ipos, bitola, nfer, mult, itpcorbar, ivar, rdval, rddsc, compr, igane, igand,
observ, ilance, itipo99, icftppata, icorrido, iluvai, iluvaf =
rebar.RebarScheduleInfo (idescr)
```

A classe TQSDwg.Iterator tem uma propriedade que retorna um objeto tipo SmartRebar caso o elemento gráfico extraído seja um, ou None:

```
iterator .smartRebar
```

Retorna objeto SmartRebar se o objeto extraído é um ferro inteligente, ou None

A Função abaixo, de SmartRebar, retorna o "número de descrições de um ferro". Este número é necessário apenas para estribos de 4 ou 6 ramos, que agrupam ferros diferentes.

```
RebarScheduleNumDescr ()
```

Retorna o número de descrições de um ferro.

E a função RebarScheduleInfo, retorna informações de um ferro usadas para gerar uma tabela de ferros;

```
RebarScheduleInfo (idescr)
```

idescr é o número da descrição de 0.. RebarScheduleNumDescr()-1

Retorna:

ipos Número da posição

bitola Bitola mm

nfer Número de ferros

mult Multiplicador de ferros

itpcorbar Tipo de ferro CORBAR

ivar (1) Se ferro variável

`rdval` Raio de dobra

`rddsc` Deconto total raio de dobra

`compr` Comprimento total sem desconto

`igane` Gancho esquerdo tipo `TQSDwg.ICPSGA/ICP090/ICP135/ICP180`

`igand` Gancho direito tipo `TQSDwg.ICPSGA/ICP090/ICP135/ICP180`

`observ` Observação associada

`ilance` Lance de pilar

`itipo99`(1) Se ferro tipo 99 é estribo (informação para o G-Bar)

`icftppata` Pata de estribo `ICPTPPATA45/90`

`icorrido`(1) Ferro corrido

`iluvai`(1) Luva inicial

`iluvaf`(1) Luva final

`InUse` ()

Retorna (`!=0`) se a posição do ferro está em uso por outro ferro.

## Programa TSTFer.py de teste de geração de ferros inteligentes no TQSDwg

Este programa é uma adaptação de um programa em C++ desenvolvido para testar ao máximo todas as funcionalidades da biblioteca de ferros inteligentes. Assim, é um pouco extenso (e mais difícil de entender), gerando no total mais de 300 ferros. Entretanto, é uma referência para entender como os ferros são gerados.

Ele grava o desenho `TESTE2.DWG`, usando as seguintes rotinas:

Rotina	Função
<code>DesFerroReto()</code>	Todas as variações de ferros retos
<code>FerroGeneri()</code>	Ferros genéricos
<code>FerroEstrib()</code>	Estribos de vigas
<code>FerroGrampo()</code>	Grampos de vigas
<code>FerroVariavel()</code>	Ferros retos com variação de comprimento
<code>EstribGen()</code>	Estribos de pilares com faixa de distribuição comum
<code>EstribRetVar()</code>	Estribo de viga com variação de seção
<code>FerroCorrido()</code>	Ferros corridos
<code>DesenharEscada()</code>	Desenho simples de escadas para teste interativo no EAGFER
<code>DesenharArco()</code>	Ferros genéricos com trecho em arco

<code>DesenharLinhas ()</code>	Linhas arbitr�rias para teste interativo de linhas em ferros.
<code>DesenharFerCor ()</code>	Identifica�o de ferros em corte, usando ferros retos como exemplo.
<code>ExtrairTabela ()</code>	Exemplo de extra�o de tabela de ferros.