

Tabla de contenido

Usar expresiones y variables	1
Usar expresiones y variables: Introducción	1
Usar expresiones en una rutina de medición.....	2
Ver valores de las expresiones	2
Mantener valores de las expresiones solamente	2
Usar expresiones con ramificación	3
Usar expresiones con Entrada/Salida de archivos	3
Crear expresiones con el constructor de expresiones	4
Crear una expresión tecleándola	4
Crear una expresión con el constructor de expresiones	5
Verificar si la expresión es correcta	6
Tipo de elemento de expresión	7
ID	7
Extensión	7
Segunda extensión	9
Botón Añadir	10
Cuadro de edición	11
Área de descripción	11
Usar variables con expresiones.....	11
Asignar valores a variables mediante el cuadro de diálogo Asignación	12
Comprender los componentes de las expresiones	13
Tipos de operandos	13

Literales	13
Referencias	14
Variables	21
Estructuras	25
Punteros	27
Matrices	29
Operadores para expresiones	43
Precedencia	45
Funciones	46
Coerción de operandos	96
Expresiones de ID	99
Acceder a las propiedades de objeto de un informe	102
Acceder a la información de un círculo mínimo de escaneado construido	106

Usar expresiones y variables

Usar expresiones y variables: Introducción

Una expresión es una condición definida por el usuario y utilizada con los comandos de control de flujo de PC-DMIS. Utilizando instrucciones de control de flujo puede comprobar estas condiciones en la rutina de medición. El usuario puede determinar qué debe hacer PC-DMIS cuando se cumple o no se cumple la condición.

Las expresiones son un componente importante para que PC-DMIS lleve a cabo sus tareas específicas. Cuando utilice las expresiones junto con comandos de control de flujo, podrá desarrollar muchas más de las grandes funciones de PC-DMIS.

En este capítulo se explica cómo crear y utilizar expresiones dentro de la ventana de edición de PC-DMIS. Cuando trabaje con expresiones, debe cambiar la ventana de edición de PC-DMIS al modo Comando. De este modo podrá visualizar directamente el código de la ventana de edición.

En este capítulo se tratan los siguientes temas principales:

- Usar expresiones en una rutina de medición
- Crear expresiones con el constructor de expresiones
- Usar variables con expresiones
- Comprender los componentes de las expresiones
- Acceder a las propiedades de objeto de un informe
- Acceder a la información de un círculo mínimo de escaneado construido



Si busca información sobre expresiones de informe, consulte "Acerca de las expresiones de los informes" en el capítulo "Informes de los resultados de las mediciones".

Usar expresiones en una rutina de medición

La ventana de edición de PC-DMIS permite utilizar expresiones en la mayoría de los campos editables. Los campos editables son los que se resaltan en amarillo cuando se pulsa la tecla del tabulador en la ventana de edición, estando esta en el modo Comando. Los campos que cambian el tipo de elemento no permiten utilizar expresiones.



No es posible utilizar expresiones en los cuadros de los elementos automáticos que especifican el tipo del elemento automático; por ejemplo, Punto de superficie, Círculo automático, Ranura redonda automática, etc.

Los subtemas que componen este tema ofrecen una referencia completa a las expresiones disponibles.

Ver valores de las expresiones

Para ver el valor de una expresión, coloque el cursor dentro de la expresión y déjelo allí al menos un segundo. Se evaluará la expresión y debajo del cursor del ratón aparecerá una ventana emergente de color amarillo para mostrar la expresión y su valor actual.

Mantener valores de las expresiones solamente

Para resolver inmediatamente una expresión en la ventana de edición y conservar solo el valor, siga estos pasos:

1. Seleccione el texto de la expresión en la ventana de edición.

Usar expresiones y variables

2. Coloque una tilde (`) antes de la expresión.



Supongamos que teclea la expresión ``1/7` en un campo numérico. La expresión se resolverá inmediatamente y solo se insertará en el campo el valor **(0,143)**.

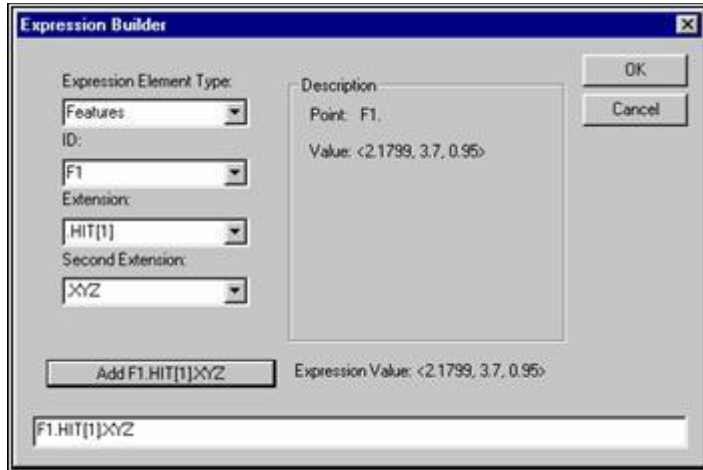
Usar expresiones con ramificación

Los comandos de control de flujo utilizan expresiones para determinar el flujo de ejecución de la rutina. Consulte el capítulo "Ramificación mediante control de flujo" para obtener más información acerca de cuándo puede o no producirse la ramificación.

Usar expresiones con Entrada/Salida de archivos

Al escribir datos en un archivo de datos externo o al leer datos de un archivo de datos externos con frecuencia se utilizan variables y otras expresiones para gestionar y almacenar o mostrar dichos datos de la forma más eficaz. Consulte el capítulo "Usar entrada/salida de archivos" para obtener más información.

Crear expresiones con el constructor de expresiones



Cuadro de diálogo Constructor de expresiones

PC-DMIS permite crear y añadir expresiones en la ventana de edición tecleándolas o mediante la interfaz del cuadro de diálogo **Constructor de expresiones**. La opción de menú **Edición | Expresión** abre el cuadro de diálogo **Constructor de expresiones**.

Este cuadro de diálogo le permite crear una expresión e insertarla en un campo editable. También se abrirá el cuadro de diálogo **Constructor de expresiones** si pulsa la tecla F2 mientras el cursor se encuentra en un campo que acepta expresiones.

El cuadro de diálogo **Constructor de expresiones** presenta todos los tipos de operadores y funciones que se pueden utilizar en las expresiones.

Crear una expresión tecleándola

Para crear una expresión tecleándola directamente en la ventana de edición:

1. Abra la ventana de edición (**Ver | Ventana de edición**).
2. Coloque la ventana de edición en modo Comando.
3. Pulse la tecla TAB para mover el cursor al campo editable donde desea insertar la expresión. Los campos marcados en amarillo son "editables".
4. Teclee la expresión.

Crear una expresión con el constructor de expresiones



Debe estar en modo Comando para que se active la opción Expresión.

Para introducir una expresión mediante el cuadro de diálogo Constructor de expresiones (**Edición | Expresión**):

1. Abra la ventana de edición (**Ver | Ventana de edición**).
2. Coloque la ventana de edición en modo Comando (**Ver | Modo Comando**).
3. Desplace el cursor hasta el campo editable donde desea insertar la expresión.
4. Pulse la tecla F2 mientras el cursor se encuentra en un campo que acepta expresiones. Se abre el cuadro de diálogo **Constructor de expresiones**. El cuadro de diálogo **Constructor de expresiones** presenta todos los tipos de operadores, operandos y funciones. Es posible especificar lo siguiente en este cuadro de diálogo:
 - Tipos de expresiones disponibles
 - Variables
 - Elementos
 - Dimensiones
 - Alineaciones
 - Comentarios
5. Seleccione el elemento de la expresión en la primera lista desplegable. Según su selección, aparecerán otros cuadros de opciones.
6. Seleccione la ID deseada en la lista desplegable **ID**.
7. Seleccione una extensión en la lista desplegable **Extensión**.
8. Seleccione otra extensión en la lista desplegable **Segunda extensión**. Si la expresión puede utilizarse, se activará el botón **Añadir**.
9. Haga clic en el botón **Añadir**. La expresión aparecerá en un cuadro de edición.
10. Haga clic en el botón **Aceptar**. Ahora, la expresión aparecerá en el punto en el que está el cursor en la ventana de edición.



También puede abrir el cuadro de diálogo **Constructor de expresiones** desde estos otros cuadros de diálogo:

- El cuadro de diálogo **Expresión If**: Seleccione **Insertar | Control de flujo | Si Ir a**. Haga clic en el botón **Expresión**.
- El cuadro de diálogo **Asignación**: seleccione **Insertar | Asignación**. Haga clic en el botón **Asignar a** o **Asignar desde**.

Una vez creada la expresión, PC-DMIS automáticamente inserta la expresión en la siguiente posición válida en la ventana de edición.

Verificar si la expresión es correcta

Cuando el cursor salga del campo en el que ha incluido la expresión, PC-DMIS tratará de verificar que la expresión es correcta. Si hay algún problema en la expresión, es posible que aparezca un mensaje de error para indicar que hay un número no válido, o bien que la expresión aparezca en color rojo. También aparecerán en color rojo las expresiones que incluyen referencias a objetos inexistentes.

Dado que la verificación de si la expresión es correcta se realiza en el momento de salir del campo, un campo que esté en rojo debido a una referencia a un objeto inexistente (por ejemplo, CÍRCULO1.X) permanecerá en rojo incluso si se añade el nuevo objeto (por ejemplo, CÍRCULO1). El campo permanece de color rojo hasta que se vuelve a verificar si la expresión es correcta.

Para volver a verificar si la expresión es correcta:

1. Desplace el cursor hasta el campo de la expresión.
2. Pulse la tecla F2 . Se volverá a abrir el cuadro de diálogo **Constructor de expresiones**. Los cambios que haga en la expresión aparecerán en el cuadro de edición.
3. Pulse la tecla INTRO para cerrar el cuadro de diálogo.

Tipo de elemento de expresión

La lista desplegable Tipo de elemento de expresión del cuadro de diálogo **Constructor de expresiones (Edición | Expresión)** contiene los diferentes tipos de elementos de expresión que pueden insertarse en las expresiones. Son estos:

- Funciones
- Operadores
- Alineaciones
- Comentarios
- Dimensiones
- Elementos
- Variables

ID

La lista desplegable **ID** del cuadro de diálogo **Constructor de expresiones (Edición | Expresión)** enumera el conjunto de elementos disponibles según el tipo de elemento de expresión seleccionado en la lista desplegable **Tipo de elemento de expresión**.



La lista de elementos disponibles depende del elemento de expresión seleccionado:

- Cuando se selecciona **Funciones y operadores** en la lista desplegable **Tipo de elemento de expresión**, la lista desplegable **ID** contiene una lista de funciones y operadores disponibles.
- Cuando se selecciona **Elementos** en la lista desplegable **Tipo de elemento de expresión**, la lista desplegable **ID** muestra las identificaciones de los elementos de la rutina de medición.

Extensión

La lista desplegable **Extensión** del cuadro de diálogo **Constructor de expresiones (Edición | Expresión)** está disponible cuando el elemento seleccionado en la lista desplegable **ID** requiere que se añada una extensión para poder formar un elemento de

expresión utilizable. La lista desplegable **Extensión** enumera las extensiones, según el elemento seleccionado en la lista desplegable **ID**.



Supongamos que selecciona un elemento en la lista desplegable **ID**. PC-DMIS muestra las posibles extensiones que puede utilizar para hacer referencia a datos de dicho elemento (por ejemplo, "X", "Y", "Z", "Diám", "Long", etc.) en la lista desplegable **Extensión**.

Las extensiones posibles son los tipos de datos medidos o teóricos:

Medido

- All: Todos los valores del elemento se asignan a la variable. Consulte el ejemplo siguiente.
- X: Valores X medidos de los contactos
- Y: Valores Y medidos de los contactos
- Z: Valores Z medidos de los contactos
- XYZ: Valores XYZ medidos de los contactos
- I: Valores I medidos de los contactos
- J: Valores J medidos de los contactos
- K: Valores K medidos de los contactos
- IJK: Valores IJK medidos de los contactos

Teórico

- TX: Valores X teóricos de los contactos
- TY: Valores Y teóricos de los contactos
- TZ: Valores Z teóricos de los contactos
- TXYZ: Valores XYZ teóricos de los contactos
- TI: Valores I teóricos de los contactos
- TJ: Valores J teóricos de los contactos
- TK: Valores K teóricos de los contactos
- TIJK: Valores IJK teóricos de los contactos



Supongamos que este fragmento de código forma parte de una rutina de medición:

```
F1=GENÉRICO/PUNTO,DEPENDIENTE,CARTESIANA,$
```

```
NOM/XYZ,<8,9,10>,$
```

```
MED/XYZ,<7.98,8.98,9.98>,$
```

```
NOM/IJK,<1,0,0>,$
```

```
MED/IJK,<1,0,0>
```

```
ASIGN/MIELEMENTO=F1.ALL
```

```
ASIGN/V1=MIELEMENTO.XX
```

```
ASIGN/V2=MIELEMENTO.TX
```

Cuando la rutina de medición ejecuta el fragmento de código anterior, PC-DMIS asigna las variables con estos valores:

```
V1 = 7.98
```

```
V2 = 8
```

Segunda extensión

La lista desplegable **Segunda extensión** se activa solamente si el ítem elegido en la lista desplegable **Extensión** necesita que se añada una segunda extensión para poder formar un elemento de expresión utilizable.



Supongamos que hace referencia al valor nominal del eje de la ubicación X de una dimensión llamada "D1". Haría lo siguiente:

1. Elegir **D1** en la lista desplegable **ID**.
2. Seleccione **X** en la lista desplegable **Extensión**.
3. Seleccione **Nom** en la lista desplegable **Segunda extensión**.

Botón Añadir

Cada vez que selecciona un elemento de expresión utilizable o completo en una lista, se activa el botón **Añadir**. Con este botón se muestra el texto que debe añadirse a la expresión.

Por ejemplo, si ha elegido lo siguiente para la expresión:

- Dimensiones en la lista **Tipo de elemento de expresión**
- D1 en la lista **ID**
- X en la lista **Extensión**
- Nom en la lista **Segunda extensión**

Se activa el botón **Añadir** y aparece el siguiente texto: **Add D1.X.NOM**.

Al hacer clic en el botón **Añadir**, el texto aparecerá en el cuadro de edición, en la parte inferior del cuadro de diálogo.



Cuando haga clic en el botón **Aceptar**, PC-DMIS añadirá el texto del cuadro de edición al campo de expresión de la ventana de edición donde está situado el cursor. Si selecciona un elemento en el campo de expresión de la ventana de edición y el texto que desea añadir contiene paréntesis, el elemento seleccionado se colocará dentro de los paréntesis del texto añadido.

Cuadro de edición

En la parte inferior del cuadro de diálogo **Expresión (Edición | Expresión)** aparece un cuadro de edición con la expresión actual. Puede teclear la expresión directamente en este cuadro, o bien introducirla mediante el botón **Añadir**.

Área de descripción

El cuadro de diálogo **Constructor de expresiones (Edición | Expresión)** también contiene un área llamada **Descripción**, que muestra información acerca de los elementos seleccionados en las listas desplegables. El campo al lado del botón **Añadir** también muestra el valor actual de la expresión.



Las expresiones no válidas tienen un valor 0.

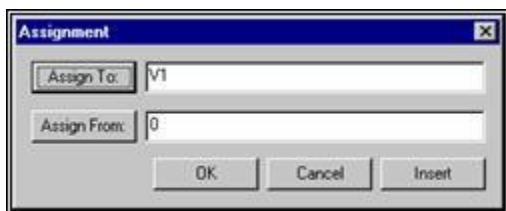
Usar variables con expresiones

Las variables son objetos que contienen valores. Las variables hacen referencia a diversos tipos de operandos, como enteros, reales, cadenas o puntos. Las variables son esenciales para utilizar expresiones. Las variables tienen un nombre y un valor. El nombre se utiliza para tener acceso al valor de la variable. El nombre es constante y el valor puede cambiar. Puede utilizar el comando `ASIGN/` para asignar un valor a una variable.

Por ejemplo, la instrucción `ASIGN/V1=2` crea una variable con el nombre V1 y el valor 2. `ASIGN/V2=V1+2` accede al valor de V1. Si V1 todavía tenía el valor de 2 al ejecutarse la instrucción ASIGN, V2 tendría entonces el valor de 4.

Para obtener más información acerca de las variables, consulte el tema "Variables".

Asignar valores a variables mediante el cuadro de diálogo Asignación



Cuadro de diálogo Asignación

La opción de menú **Insertar | Asignación** abre el cuadro de diálogo **Asignación**. Utilice este cuadro de diálogo para asignar un valor a una variable o elemento de datos de una alineación, dimensión o elemento en una rutina de medición. El uso del comando de asignación requiere conocimientos básicos de las expresiones de PC-DMIS.

Botón Asignar a: Este botón designa la variable que recibe el valor calculado en el cuadro **Asignar desde**. La información elegida con el botón **Asignar a** se introduce en el cuadro **Asignar a**. Este valor puede incluir el nombre de una variable o una referencia a un elemento de datos de una alineación, dimensión o elemento.

Por "evaluado" se entiende el resultado de resolver la expresión matemática para obtener un valor.

Botón Asignar desde: Este botón coloca el valor asignado en el cuadro **Asignar desde**. Si el cuadro contiene una expresión, esta se evalúa en el momento de la ejecución y el resultado o valor del cálculo se asigna al objeto especificado en el cuadro **Asignar a**.

Botón Insertar: Este botón inserta un comando de asignación en la rutina de medición a la vez que se mantiene abierto el cuadro de diálogo **Asignación**. Se pueden insertar una serie de comandos de asignación sin cerrar el cuadro de diálogo.

Temas relacionados:

Comprender los componentes de las expresiones

Acceder a las propiedades de objeto de un informe

Comprender los componentes de las expresiones

En las expresiones se utilizan los siguientes tipos de operandos:

- Números enteros
- Números reales
- Cadenas
- Puntos
- Punteros a elementos
- Matrices
- Funciones

Estos operandos se explican en detalle más adelante.

Tipos de operandos

Existen varios tipos de operandos:

- Literales
- Referencias
- Variables
- Estructuras
- Punteros
- Matrices

Literales

***Enteros:** 1, -6, 209

Reales: 1, -6, 2.4, -0.1, 345.6789

Cadenas: "Hola a todos", "47", "CIRCULO1"

Puntos: No existe una representación literal para los puntos. Sin embargo, la función `MPOINT` permite crear puntos a partir de otros literales; por ejemplo: `MPOINT(0,0,1)`, `MPOINT(2.2,3.1,4.0)`.

Puntero: Nombre de un elemento incluido entre llaves: {CIR1}, {LIN2}, {F3}

Matrices: No existe una representación literal para las matrices. Sin embargo, la función ARRAY permite crear matrices a partir de otros literales; por ejemplo: `ARRAY(3,5,6)`, `ARRAY("Hola",2.3,9)`. Estas funciones crean matrices de 3 elementos con los elementos enteros 3, 5 y 6 en el primer ejemplo y la cadena del elemento "Hola", el elemento doble 2.3 y el componente entero 9 en el caso del segundo ejemplo.

Funciones: No existe una representación literal para las funciones. Las funciones se definen mediante la palabra clave `FUNCTION` y se accede a ellas mediante las ID de las variables. Por ejemplo, `ASIGN/Añadir2=FUNCTION(X,X+2)` define una función que toma un argumento y le suma 2. A la variable Añadir2 se le asigna la función. Es posible invocar la función utilizando la variable Añadir2 de la siguiente manera. `ASIGN/Resultado=Añadir2(5)`. Al resultado se le asigna el valor 7.



Los literales numéricos se interpretan como números reales, salvo si el operador o la función implica el uso de números enteros. Por ejemplo, la evaluación de la expresión `10 / 8` da como resultado 1,25 en lugar de 1. Tenga en cuenta asimismo que las divisiones discretas también se pueden realizar por medio de los operadores de coerción de operandos. La expresión `INT(10) / INT(8)` sí arroja 1.

Referencias

Las referencias permiten incluir referencias a datos específicos de otros objetos de una rutina de medición. Las referencias usan la ID de un objeto de la rutina de medición, seguida de un punto y una extensión que hace referencia a datos específicos del objeto.



Dado un círculo medido en la rutina de medición llamado CIRCULO1, `CIRCULO1.X` haría referencia al valor que se ha medido para el componente X de CIRCULO1. Todas las referencias se evalúan en coordenadas de pieza respecto a la alineación actual.

Referencias de tipo Doble

Puede utilizar las siguientes expresiones de referencia:

Ejemplos de extensiones válidas para referencias a elementos de tipo Doble

Formato: <ID elemento>.<extensión> -> `CIRCULO1.X`

CIRCULO1.X Valor X medido para CIRCULO1

CIRCULO1.Y Valor Y medido para CIRCULO1

CIRCULO1.Z Valor Z medido para CIRCULO1

CIRCULO1.TX Valor X teórico (nominal) para CIRCULO1

CIRCULO1.TY Valor Y teórico (nominal) para CIRCULO1

CIRCULO1.TZ Valor Z teórico (nominal) para CIRCULO1

LÍNEA1.SX Valor X medido para el punto de inicio de LÍNEA1

LÍNEA1.SY Valor Y medido para el punto de inicio de LÍNEA1

LÍNEA1.SZ Valor Z medido para el punto de inicio de LÍNEA1

LÍNEA1.TSX Valor X teórico para el punto de inicio de LÍNEA1

LÍNEA1.TSY Valor Y teórico para el punto de inicio de LÍNEA1

LÍNEA1.TSZ Valor Z teórico para el punto de inicio de LÍNEA1

LÍNEA1.EX Valor X medido para el punto final de LÍNEA1

LÍNEA1.EY Valor Y medido para el punto final de LÍNEA1

LÍNEA1.EZ Valor Z medido para el punto final de LÍNEA1

LÍNEA1.TEX Valor X teórico para el punto final de LÍNEA1

LÍNEA1.TEY Valor Y teórico para el punto final de LÍNEA1

LÍNEA1.TEZ Valor Z teórico para el punto final de LÍNEA1

PUNTO.I Valor medido para el componente I del vector de PUNTO

PUNTO.J Valor medido para el componente J del vector de PUNTO

PUNTO.K Valor medido para el componente K del vector de PUNTO

PUNTO.TI Valor teórico para el componente I del vector de PUNTO

PUNTO.TJ Valor teórico para el componente J del vector de PUNTO

PUNTO.TK Valor teórico para el componente K del vector de PUNTO

ELEM1.TYP Tipo de elemento (por ejemplo, círculo, ranura, cono). Puede utilizar esto para cambiar el tipo de un elemento genérico (`ASIGN/Gen1.TYP=Elem1.TYP`).

ELEM1.ALL hace referencia a todos los componentes del elemento. Es de gran utilidad para copiar información en un elemento genérico. (`ASIGN/Gen1.ALL=Elem1.ALL`)

Vector de superficie

BORDE.SURFI
BORDE.SURFJ
BORDE.SURFK
BORDE.TSURFI
BORDE.TSURFJ
BORDE.TSURFK

Vector de ángulo

CIR.ANGI
CIR.ANGJ
CIR.ANGK
CIR.TANGI
CIR.TANGJ
CIR.TANGK

Radio

CIRCULO1.R
CIRCULO1.TR
CIRCULO1.RAD
CIRCULO1.TRAD
CIRCULO1.RADIUS
CIRCULO1.PR : Radio polar
CIRCULO1.TPR : Radio polar teórico
CIRCULO1.TRADIUS (solo son significativos los primeros caracteres)

Diámetro

CIRCULO1.D
CIRCULO1.TD
CIRCULO1.DIAM
CIRCULO1.TDIAM
CIRCULO1.DIAMETER
CIRCULO1.TDIAMETER (solo son significativos los primeros caracteres)

Ángulo

CONO.A
CONO.TA
CONO.ANG
CONO.TANG
CONO.ANGLE
CONO.TANGLE
CONO.PA : Ángulo polar
CONO.TPA: Ángulo polar teórico (solo son significativos los primeros caracteres)

Longitud

LÍNEA.L
LÍNEA.TL
LÍNEA.LEN
LÍNEA.TLEN
LÍNEA.LENGTH
LÍNEA.TLENGTH (solo son significativos los primeros caracteres)

Altura

CILINDRO.PH : Altura polar
CILINDRO.TPH: Altura polar teórica

Radio, ángulo, altura

PUNTO.RAH: Punto con radio, ángulo y altura medidos
PUNTO.TRAH: Punto con radio, ángulo y altura teóricos

Área

BLOB1.AREA: Devuelve el valor de área medida de un elemento de tipo blob.
BLOB1.TAREA: Devuelve el valor de área teórica de un elemento de tipo blob.
Por ejemplo, `ASIGN/V1=BLOB1.AREA` devuelve el valor de área medida del elemento BLOB1 y lo asigna a la variable V1.
Actualmente solo el elemento blob funciona con estas extensiones de área. Para obtener información sobre el elemento blob, consulte el tema "Blob de Vision" en la documentación de "PC-DMIS Vision".

Ejemplos de extensiones válidas para referencias a dimensiones de tipo Doble

Formato: <ID de dimensión>.<EJE>.<Componente de dimensión> -> `DIM1.X.NOM`

DIM1.X.NOM	El valor nominal de la ubicación del eje X de DIM1
DIM1.X.MEAS	El valor medido de la ubicación del eje X de DIM1
DIM1.X.MAX	La desviación máxima de la ubicación del eje X de DIM1
DIM1.X.MIN	La desviación mínima de la ubicación del eje X de DIM1
DIM1.X.PTOL	El valor de tolerancia positivo de la ubicación del eje X de DIM1
DIM1.X.MTOL	El valor de tolerancia negativo de la ubicación del eje X de DIM1
DIM1.X.DEV	La desviación en la ubicación del eje X de DIM1
DIM1.X.OUTTOL	El valor de fuera de tolerancia de la ubicación del eje X de DIM1

DIM1.Y.NOM	El valor nominal de la ubicación del eje Y de DIM1
DIM1.Z.DEV	La desviación en la ubicación del eje Y de DIM1
DIM3.PA.MEAS	El valor medido de la ubicación del ángulo polar de DIM3
DIM4.M.PTOL	El valor de tolerancia positivo de la ubicación del eje M de DIM4
DIM4.PTOL	El valor de tolerancia positivo de la ubicación del eje M de DIM4 (consulte la *nota que aparece después de "Ejes válidos" a continuación).
DIM5.BTOL	El valor de tolerancia adicional en el que DIM5 es una posición.

Ejes válidos:

X, Y, Z, D, R, A, T, V, L, PR, PA, M, PD, RS, RT, S, H, DD, DF, TP



* Para las dimensiones que por definición tienen un solo eje (por ejemplo, redondez, concentricidad, etc.) se puede omitir el modificador del eje. Si utiliza el modificador de eje, tenga en cuenta que todos estos tipos de dimensiones (de un solo eje) utilizan el modificador de eje M, con la excepción de las dimensiones angulares de 2 y de 3 dimensiones, que utilizan el modificador de eje A.

Ejemplos de extensiones válidas para referencias a alineaciones de tipo Doble:

Formato: <ID alineación>.<Eje/origen alineación>.<Componente eje/origen alineación>
-> A1.ORIGIN.X

A1.ORIGIN.X	Componente X del origen medido de la alineación A1.
A2.ORIGIN.Y	Componente Y del origen medido de la alineación A2.
A1.ORIGIN.Z	Componente Z del origen medido de la alineación A1.
A1.XAXIS.I	Componente I del eje X medido de la alineación A1.
A1.YAXIS.J	Componente J del eje Y medido de la alineación A1.
A1.ZAXIS.K	Componente K del eje Z medido de la alineación A1.
A1.CORIGIN.X	Componente X del origen de la alineación A1, basado en datos teóricos (C viene de CAD)
A1.CXAXIS.J	Componente J del eje de la alineación A1, basado en datos teóricos (C viene de CAD)

Referencias de tipo Punto

Puede utilizar las siguientes expresiones de referencia:

Ejemplos de extensiones válidas para referencias a elementos de tipo Punto

Formato: <ID elemento>.<extensión> -> CIRCULO1.XYZ

CIRCULO1.XYZ	Centroide medido de CIRCULO1
CIRCULO1.TXYZ	Centroide teórico de CIRCULO1
LINEA1.SXYZ	Punto inicial medido de LINEA1
LINEA1.TSXYZ	Punto inicial teórico de LINEA1
LINEA1.EXYZ	Punto final medido de LINEA1
LINEA1.TEXYZ	Punto final teórico de LINEA1
CIRCULO1.IJK	Vector medido de CIRCULO1
CIRCULO1.TIJK	Vector teórico de CIRCULO1
BORDE.SURFIJK	Vector de superficie medido de BORDE

BORDE.TSURFIJK	Vector de superficie teórico de BORDE
CIRAUTO1.ANGIJK	Vector de ángulo medido de CIRAUTO1
CIRAUTO1.TANGIJK	Vector de ángulo teórico de CIRAUTO1

Ejemplos de extensiones válidas para referencias a alineación de tipo Punto

Formato: <ID alineación>.<Eje/origen alineación> -> A1.XAXIS

A1.ORIGIN	Origen medido de la alineación de A1
A1.XAXIS	Eje X medido de la alineación de A1
A1.YAXIS	Eje Y medido de la alineación de A1
A1.ZAXIS	Eje Z medido de la alineación de A1
A1.CORIGIN	Origen teórico de la alineación de A1
A1.CXAXIS	Eje X teórico de la alineación de A1
A1.CYAXIS	Eje Y teórico de la alineación de A1

A1.CZAXIS	Eje Z teórico de la alineación de A1
-----------	--------------------------------------

Referencias de tipo Cadena

Las referencias a comentarios son los únicos tipos de objeto de tipo Cadena. Solo es posible hacer referencia a comentarios de ENTRADA o de SÍ/NO. Estos tipos de comentario tienen una ID que puede utilizar para identificar el comentario.

Formato: <ID comentario>.INPUT -> [C1.INPUT](#)

C1.INPUT: Valor de la entrada (del operador) en el comentario C1

Los comentarios de tipo SÍ/NO establecen la entrada en la cadena sí o no en función del idioma de PC-DMIS. En PC-DMIS, si el operador pulsa el botón Sí, la cadena queda establecida en "SÍ". Si el operador pulsa el botón No, la cadena se establece en "NO". Cuando se comparan cadenas para probar los casos de "SÍ" o "NO", se distingue entre mayúsculas y minúsculas. Por ello, si compara con "sí" o "no", la comparación no funciona si la entrada de comentario SÍ/NO está establecida en "SÍ" o en "NO".

Variables

Las variables pueden ser de cualquiera de los siete tipos de operandos: enteros, reales, cadenas, puntos, punteros a elemento, matrices y función. Las variables cobran existencia y reciben sus valores y sus tipos por medio de la instrucción [ASIGN](#).

La ID de variable puede ser cualquier cadena alfanumérica que no empiece por un carácter numérico. Puede utilizar caracteres de subrayado en la ID de la variable, siempre y cuando el subrayado no sea el primer carácter.



Mientras la rutina de medición permanece abierta, PC-DMIS guarda los valores de las variables entre las ejecuciones. Esto significa que, cuando la ejecución finalice, PC-DMIS utilizará los valores como final de la rutina cuando se reinicie la rutina. Este comportamiento puede ser o no ser el deseado. Si desea valores de variables nuevos, una buena práctica consiste en borrar los valores con instrucciones ASIGN al principio de la rutina. Por ejemplo, si utiliza el valor de la variable V1 en algunos cálculos numéricos, puede utilizar `ASIGN/V1=0` para vaciar esa variable.



Si está activa la ventana de edición, PC-DMIS indica el valor actual de la variable cada vez que se sitúe el cursor en el campo. Durante la ejecución, los valores de las variables cambian en función del flujo de ejecución. Sitúe el puntero del ratón encima de una variable para saber qué valor tiene en ese momento.

```
ASIGN/V1=2.2+2
```

La variable V1 es un número real con el valor de 4.2.

```
ASIGN/VAR1=CIRCULO1.X
```

La variable VAR1 es un número real con un valor que equivale al valor medido de CIRCULO1.X en el momento de la asignación.

```
ASIGN/MIVAR=LINEA1.XYZ
```

La variable MIVAR es un punto que tiene el mismo valor del centroide de LINEA1 medido en el momento de la asignación.

```
ASIGN/VARC="Hola a todos"
```

La variable VARC es una cadena que tiene el valor "Hola a todos".

En estos ejemplos, a las variables se les asignan valores. Una vez que se asigna un valor a una variable, puede utilizar la variable como operando en cualquier campo de expresiones.

Aquí, V1 se utiliza en un campo numérico. Se utiliza como valor para el comando de precontacto.

```
ASIGN/V1=1/3PRECONT/V1
```




Debido a que es posible utilizar expresiones en la mayoría de los campos que puede modificar, la siguiente expresión también es válida y tiene idénticos efectos: PRECONT / 1/3.

Es posible hacer referencias individuales a los componentes de las variables de tipo Punto, por medio de la notación de punto y extensión que se utiliza para las referencias.

```
ASIGN/V1=MPOINT(3,4,5)
```

V1 es de tipo Punto y tiene un valor de 3, 4, 5.

```
ASIGN/VARX=V1.X
```

VARX es de tipo Doble y tiene un valor de 3.

```
ASIGN/VARY=V1.Y
```

VARY es de tipo Doble y tiene un valor de 4.

```
ASIGN/ VARI=V1.I
```

VARI es de tipo Doble y tiene un valor de 3.

```
ASIGN/VARREDUN=V1.XYZ
```

VARREDUN es de tipo Punto con el valor de 3, 4, 5.

Las siguientes extensiones son equivalentes. Se incluyen las dos para explicar mejor el significado de una expresión de una rutina de medición.

Dado que V1 es de tipo Punto.

V1.X es igual que V1.I

V1.Y es igual que V1.J

V1.Z es igual que V1.K.

V1.XYZ V1.IJK y V1 sin extensión.

Si una variable de tipo Cadena tiene un valor que equivale al nombre de una ID, una dimensión o una alineación de un elemento, puede utilizar la variable como objeto de referencia.

```
ASIGN/V1="CÍRCULO1"
```

Si existe un elemento denominado CIRCULO1, puede utilizar los siguientes operandos.

V1.X: Valor medido X en CIRCULO1

V1.TX: Valor teórico de X en CIRCULO1

V1.Diameter: Diámetro medido de CIRCULO1

V1.Radius: Radio medido de CIRCULO1

Este tipo de referencia indirecta sólo se puede utilizar en un nivel de referencia indirecta en las variables de cadena. Los ejemplos siguientes no funcionan.



```
ASIGN/V1="CIRCULO1"
ASIGN/V2="V1"
```

V2.X: En lugar del valor medido de CÍRCULO1.X, esto da como resultado 0.



La referencia V2.X *no* aparece indicada como error en texto rojo aunque una expresión situada por encima establezca que es de tipo cadena. Esto se debe a que el flujo de ejecución de la rutina de medición no se conoce hasta la hora de la ejecución.

No obstante, si se utilizan llaves, no funcionará lo siguiente:



```
ASIGN/V1={CIRCULO1} ASIGN/V2={V1}
```

V2.X: Esto le proporciona el valor de CÍRCULO1.X.

Considere el siguiente ejemplo:



```
ASIGN/V1="CÍRCULO1"
ASIGN/V2="V1" SI/CÍRCULO1.X>CÍRCULO1.TX,IR A,L2
L1=ETIQUETA/ ASIGN/V3=V2.X
IR A/ETIQUETA,L3 L2=ETIQUETA /
ASIGN/V2=MPOINT(2,5,7)
IR A/ETIQUETA,L1 L3=ETIQUETA/
```

Durante la ejecución de la rutina, si el valor de CÍRCULO1.X es superior al de CÍRCULO1.TX, la expresión V2.X es válida y arroja 2 como resultado. De lo contrario, la expresión V2.X arroja 0, puesto que en el momento en que se ejecuta ASIGN para V3, el valor de V2 es la cadena "V1". Es responsabilidad del programador de piezas asegurarse que en estos casos las expresiones produzcan los resultados esperados.



Puede utilizar casi todas las referencias a elementos en el lado izquierdo de la instrucción `ASIGN` para incluir un valor (medido o teórico) en un elemento. La única excepción a esta regla son los componentes I, J y K de los vectores. Para realizar asignaciones a los vectores, debe asignar el vector completo de una sola vez con una expresión que dé como resultado de su evaluación un punto. Los datos de vectores se normalizan a medida que se introduzcan en los datos de vector del elemento.



```
ASIGN/CÍRCULO1.I=2-no es válido  
ASIGN/CÍRCULO1.IJK=MPOINT(2,0,0)-válido (el vector  
se normaliza en 1,0,0)
```

Para obtener información sobre el uso de variables en las dimensiones, consulte el tema "Dimensionar variables" en el capítulo "Utilizar dimensiones heredadas".

Estructuras

Puede utilizar un tipo de variable denominado *estructura* para colocar extensiones en una variable con el fin de identificar un subelemento de esa variable. Lo puede comprobar en este fragmento de código:



```
ASIGN/V1.HEIGHT=6  
ASIGN/V1.WIDTH=4.3  
ASIGN/V1.MODE="CIRCULAR"  
ASIGN/V1.POINT  
= MPOINT(100.3,37.5,63.1)
```

Donde:

- `V1` es la estructura
- `HEIGHT`, `WIDTH`, `MODE` y `POINT` son subelementos de la estructura.

Reglas de estructuras

- Al igual que las variables, tampoco es necesario declarar las estructuras.
- Los subelementos de una estructura pueden ser cualquiera de los siguientes tipos de variable:
 - Entero

- Doble
- Punto
- Puntero a elemento
- Función
- Matriz
- Estructura

Por ejemplo, es posible tener elementos de estructura que sean matrices y elementos de matriz que sean estructuras. Puede ver las expresiones válidas en el fragmento de código siguiente:

```
ASIGN/COCHE.PUERTA.IZQUIERDA[2].CUADRANTE[3].ARTICULACIÓN[5]
.HIT[4]=MPOINT(558.89,910.12,42.45)
```

```
COMENTARIO/OPER,"Posición Z actual:
"+COCHE.PUERTA.IZQUIERDA[2].CUADRANTE[3].ARTICULACIÓN[5].HIT
[4].Z
```

```
ASIGN/ARTICACTUAL=PUERTA.IZQUIERDA[2].CUADRANTE[3].ARTICULAC
IÓN[5]
```

```
COMENTARIO/OPER,"Contacto siguiente: "+ARTICACTUAL.HIT[4]
```

Estructuras con variables de tipo Punto

Si una variable es de tipo punto, puede utilizar igualmente las extensiones X, .Y, .Z, .I, .J y .K para llegar a elementos individuales del punto. También puede utilizar cualquiera de las extensiones del ejemplo siguiente en sus estructuras, sin estar obligado por ello a utilizarlas como elementos de punto, tal como se muestra a continuación:




```
ASIGN/V1.X="Una
cadena" ASSIGN/V1.Y=ARRAY(1,3,5,9,7)
ASIGN/V1.Z=MPOINT(3,5,7)
```

COMENTARIO/INFORME,V1.X

El resultado es "Una cadena"

COMENTARIO/INFORME,V1.Y[2]	El resultado es 3, el segundo elemento de la matriz.
COMENTARIO/INFORME,V1.Z.Y	El resultado es 5, el valor Y de MPOINT.

Si se combinan estructuras con las funciones del lenguaje de las expresiones de PC-DMIS, es posible tener referencias de estructuras dinámicas como se muestra a continuación:



```

ASIGN/DYNAMICSTRUCT=FUNCTION ((X,Y),X.Y)
C1=COMENTARIO/ENTRADA,Introduzca un elemento
ASIGN/CADPRUEBA=C1.INPUT
ASIGN/FRENTE=LEFT (CADPRUEBA, INDEX (CADPRUEBA, ".") -1)
ASIGN/ATRÁS=MID (CADPRUEBA, INDEX (CADPRUEBA, ".") )
ASIGN/RESULTADO=DYNAMICSTRUCT (FRENTE,ATRÁS)

```

Esta parte del ejemplo le pide que introduzca una referencia de variable, divide la referencia en el primer '.', y luego asigna **RESULTADO** para que sea igual que esa referencia utilizando la función **DYNAMICSTRUCT**.

En consecuencia, si había escrito **V1.Y[4]** para la variable **C1.INPUT**, **RESULTADO** podría arrojar un resultado de 9 (el cuarto elemento de la matriz asignada a **V1.Y**).

La evaluación del tiempo de aprendizaje de expresiones se ha mejorado para mostrar con exactitud todos los elementos de una estructura o una matriz.

Punteros

Los punteros se denominan también “punteros a elementos”. Consulte el término “Punteros a elementos” en el glosario para obtener más información.

Los punteros ofrecen una manera de hacer referencia a un elemento por medio de una variable o de pasar objetos mediante el comando LLAMAR SUB. Los punteros son parecidos a las referencias indirectas realizadas por medio de nombres de cadenas. Sin embargo, la ventaja principal de utilizar punteros se da con las subrutinas. A diferencia de las cadenas, los punteros permiten modificar directamente el objeto al que apunta la subrutina cuando se pasan como argumento de una subrutina. Los punteros no se utilizan en expresiones complejas. Si se utilizan en expresiones complejas, arrojan 0.

Considere los ejemplos siguientes:

Ejemplo de utilización de puntero:

En este ejemplo, `V1` se define como un puntero que apunta a CIR1:

```
ASIGN/V1={CIR1}
```

En este ejemplo se asigna a `DIST` el valor de la distancia de CIR1 desde el origen:

```
ASIGN/DIST=DOUBLE (V1.XYZ)
```

También se puede poner una expresión entre llaves para obtener un puntero a elemento. Los ejemplos siguientes son todas formas válidas para llevar el puntero al elemento CIR1:

```
ASIGN/RECuento ELEM=1
```

```
ASIGN/V1={"CIR"+RECuento ELEM}
```

Asigna la expresión "CIR1" a V1.

```
ASIGN/V2="CIR1"
```

```
ASIGN/V3={V2}
```

Asigna la expresión "CIR1" de la variable V2 a la variable V3

```
C1=COMENTARIO/ENTRADA, Escriba un nombre de elemento.
```

```
ASIGN/V4={C1.INPUT}
```

Esto toma el nombre de elemento de C1.INPUT y lo coloca en la variable V4.

Ejemplo de subrutina:

En la rutina que realiza la llamada:

```
CS1=LLAMAR SUB/SUB.PRG, CHANGEX, {CIR1}
```

En la subrutina:

```
GEN1=GENÉRICO/ELEM
```

```
SUBROUTINA/CHANGEX, ARG1={GEN1}
```

Usar expresiones y variables

Durante la ejecución, cuando el código pasa CIR1 a la rutina, CIR1 toma el lugar de GEN1.

```
ARG1.X=5
```

(Establece el valor X medido de CIR1 en 5)

```
FIN/SUBROUTINA
```

Ejemplo de una expresión compleja:

```
ASIGN/V1={CIR1}+2
```

{CIR1} arroja 0, de forma que el resultado de la expresión completa es 2.

Matrices

Se pueden utilizar tres tipos de matrices: de elementos, de contactos y de variables.



Aunque las matrices multidimensionales se muestran como tales en el software, en realidad solo las puede utilizar como matrices de una sola dimensión hasta que las preceda con un comando ÍNDICES MATRIZ (consulte el tema "Objeto índices de matriz:").

Matrices de elementos

Cuando se mide un elemento varias veces durante la ejecución de una rutina debido a la existencia de algún tipo de bucle, el software crea automáticamente una matriz de elementos. La cantidad de componentes que aparecen en la matriz de elementos equivale a la cantidad de veces que se ha ejecutado el elemento.



Si un elemento de círculo medido está situado en un bucle del comando MIENTRAS que se ejecuta cinco veces, se crea una matriz de cinco círculos medidos. Si la ID del círculo medido es CIR1, puede utilizar una expresión de matriz para acceder a las instancias individuales del objeto Círculo medido. Utilice corchetes para indicar la instancia que desea, como se muestra a continuación:

```
ASIGN/V1=CIR1[3].X
```

V1 se asigna al valor medido X de la tercera instancia del círculo CIR1.



Cuando existe una matriz de elementos para un elemento determinado, pero no se utiliza la notación de matrices en una referencia a dicho elemento, se utiliza la instancia más reciente. En el ejemplo anterior, la referencia `CIR1.X` sería equivalente a `CIR1[5].X`, puesto que la quinta instancia sería la más reciente del objeto.

Puede incluir expresiones en los corchetes de una expresión de matriz.

Por este motivo, `CIR1[3].X` y `CIR1[2+1].X` serían equivalentes.

En el ejemplo siguiente se utilizan dos bloques de comandos de bucle Mientras / Terminar mientras. El primer bloque ejecuta el círculo CIR1 cinco veces. El segundo bloque utiliza la variable V1 en los corchetes (`CIR1[V1].XYZ`) para enviar el centroide medido de cada una de las cinco ejecuciones a la ventana de informe:




```

                                ASIGN/V1=1
                                MIENTRAS/V1<6
CIR1                            =ELEM/CÍRCULO,CARTESIANA,DENTRO,CUAD_MÍN
                                TEO/<40,30,-4.824>,<0,0,1>,30
                                REAL/<40.002,29.991,-4.836>,<0,0,1>,29.982
                                MED/CÍRCULO,4,Z+
                                CONTACTO/BASE,NORMAL,<41.984,44.868,-
2.885>,<-0.132272,-0.9912135,0>,<41.972,44.85,-
2.891>,USAR TEOR=SÍ
                                CONTACTO/BASE,NORMAL,<51.721,39.36,-
5.094>,<-0.781412,-0.6240155,0>,<51.706,39.375,-
5.107>,USAR TEOR=SÍ
                                CONTACTO/BASE,NORMAL,<54.792,32.491,-
5.44>,<-0.9861119,-0.1660821,0>,<54.775,32.474,-
5.453>,USAR TEOR=SÍ
                                CONTACTO/BASE,NORMAL,<52.526,21.748,-
5.879>,<-0.8350841,0.5501223,0>,<52.537,21.764,-
5.893>,USAR TEOR=SÍ
                                TERMINARMED/
                                ASIGN/V1=V1+1
                                TERMINAR MIENTRAS/
                                ASIGN/V1=1
                                MIENTRAS/V1<6
                                COMENTARIO/INFORME,
                                "Centroide de CIR1, instancia n.º" + V1
                                CIR1[V1].XYZ
                                COMENTARIO/INFORME,
                                -----
                                ASIGN/V1=V1+1
                                TERMINAR MIENTRAS/

```



Esta es la salida que se genera en la ventana de informe:

	PART NAME : Top Holes - Concentric		May 23, 2022	15:25
	REV NUMBER : Rev1	SER NUMBER : 12345	STATS COUNT : 1	

Centroid of CIR1, instance #1
<39.994, 30.016, -4.833>

Centroid of CIR1, instance #2
<40.039, 30.011, -4.821>

Centroid of CIR1, instance #3
<40.032, 30.013, -4.819>

Centroid of CIR1, instance #4
<39.991, 30.013, -4.819>

Centroid of CIR1, instance #5
<40.016, 30.003, -4.83>

Las matrices también existen para dimensiones y alineaciones que se han ejecutado varias veces en una ejecución concreta. Esto significa que se podría utilizar `Dim1[2].Nom` y `Alin1[4].Origin`, siempre que la dimensión "Dim1" se haya ejecutado al menos dos veces y la alineación "Alin1" se haya ejecutado al menos cuatro veces.

Si una referencia a una matriz de elementos excede los límites (por ejemplo, si el usuario pide `CIR1[2.5]` o `CIR1["Hola a todos"]`), se devolverá el límite superior o el límite inferior. Si `CIR1` tuviera tres instancias, `CIR1[4]` y cualquier valor superior devolverían el valor de `CIR1[3]`, y `CIR1[0]` y cualquier valor inferior devolverían el valor de `CIR1[1]`. Todas las expresiones incluidas entre corchetes son forzadas a considerarse números enteros, de forma que 2,5 pasará a ser 2 y "Hola a todos" tomará el valor 0.

Objeto índices de matriz

Por omisión, las matrices de elementos son siempre unidimensionales. Si resulta más práctico tratar una matriz de elementos como si fuera una matriz multidimensional, puede hacerlo con un objeto índices de matriz.

El objeto índices de matriz permite especificar las delimitaciones superior e inferior para múltiples dimensiones de matrices.

Usar expresiones y variables

- Al establecer las delimitaciones superior e inferior de la primera dimensión, PC-DMIS crea una matriz bidimensional en la línea donde la primera dimensión está delimitada y la segunda dimensión no lo está.
- Cuando se establecen las delimitaciones superior e inferior de las dos primeras dimensiones de una matriz, PC-DMIS crea una matriz tridimensional. La última dimensión siempre está sin delimitar.



Supongamos que el elemento F1 se ubica en un bucle MIENTRAS anidado. El bucle MIENTRAS interior se ejecuta cinco veces y el bucle MIENTRAS exterior se ejecuta tres veces. Al terminarse la ejecución, F1 se habrá ejecutado 15 veces, por lo que existirán 15 instancias de F1.

Considere el siguiente ejemplo de un segmento de rutina de medición:

```
ÍNDICES_MATRIZ/1..5, ..
```

```
ASIGN/V1=1
```

```
MIENTRAS/V1<=3
```

```
    ASIGN/V2=1
```

```
    MIENTRAS/V2<=5
```

```
        F1=ELEM/PUNTO, RECT
```

```
        TEO/V2, V1, 0, 0, 0, 1
```

```
        REAL/1, 1, 0, 0, 0, 1
```

```
        MED/PUNTO, 1
```

```
        CONT/BASE, V2, V1, 0, 0, 0, 1, 1, 1, 0
```

```
        TERMINARMED/
```

```
        ASIGN/V2=V2+1
```

```
        COMENTARIO/INFORME, "Ubicación de F1["+V2+", "+V1+"
        : "+F1[V2, V1].XYZ
```

```
    TERMINAR MIENTRAS/
```

```
    ASIGN/V1=V1+1
```

```
TERMINAR MIENTRAS/
```

Este segmento de código crea una malla de 3 X 5 de 15 puntos medidos.

El comando de índices de matriz ha delimitado la primera dimensión de la matriz de elementos entre 1 y 5, ambos incluidos. Consecuentemente, en el informe de inspección, en lugar de constar como F1[1] – F1[15], los objetos constarán como F1[1, 1] – F1[5, 3], una forma más coherente con la distribución de los elementos. Observe que el comentario también se refiere a la matriz de elementos que utiliza una sintaxis de matriz bidimensional.

Para insertar un objeto índices de matriz en una rutina de medición:

1. Teclee "**Matriz**" en una línea vacía en la ventana de edición.
2. Pulse la tecla Tab del teclado.



Si se desmarca la casilla de verificación **Mostrar paréntesis para matrices de elementos**, el elemento no aparecerá con el nombre entre paréntesis. Consulte la descripción "Mostrar paréntesis para matrices de elementos" en el tema "Opciones de configuración: ficha Configuración ID" del capítulo "Establecer preferencias".

Matrices de contactos

Los contactos de un elemento en particular están disponibles como matriz a la que es posible acceder mediante expresiones que utilizan la sintaxis de matrices con los formatos siguientes: <ID elemento>.Hit[<Expresión matriz>].<extensión> o <ID elemento>.RawHit[<Expresión matriz>].<extensión>. Si está activada la compensación de sonda, Hit devuelve datos de sonda compensados. RawHit siempre devuelve datos sin compensar. Las extensiones válidas son X, Y, Z, I, J, K, TX, TY, TZ, TI, TJ, TK, XYZ, TXYZ, IJK y TIJK.

`Círculo1.Hit[1].XYZ`

El centroide medido (con compensación de sonda) para el contacto 1 de "Círculo1"

`Círculo1.Hit[2].IJK`

El vector medido para el contacto 2 de "Círculo1"

Los datos de contactos están disponibles para todos los objetos que tienen contactos, independientemente de si se muestran los contactos en la ventana de edición o no. De esta forma se pueden obtener los contactos para los escaneados y los elementos automáticos.

Usar matrices de contactos para definir las entradas de los elementos contruidos

Puede utilizar matrices de contactos para definir las entradas de los elementos contruidos.

En el caso de elementos contruidos, cuando se utilizan los métodos elemento.CONTACTO[inicio..fin] o elemento.CONTS[inicio..fin], las propiedades "inicio" y "fin" son opcionales:

- Si el valor de inicio no se ha definido, PC-DMIS presupone que se trata del valor "1".
- Si el valor de fin no se ha definido, PC-DMIS presupone que es el número total de contactos del elemento.

Es igual que si hubiera introducido "elemento.CANTCONT".



Estos ejemplos muestran tres maneras distintas de utilizar los mismos contactos desde CIR1:

Ejemplo 1



En este ejemplo, los valores de inicio y fin se definen explícitamente:



```
CIR3      ==ELEM/CIRCULO,CARTESIANA,DENTRO,NO
          TEO/<20.97629,22.90352,0>,<0,0,-1>,20.97629
          REAL/<20.96578,22.89023,-0.01243>,<0,0,-
1>,20.96578
          CONST/CIRCULO,INV,CIR1.CONTACTO[1..CIR1.CANT
CONT]
```

Ejemplo 2



En este ejemplo, solo se ha definido el valor de inicio "1". Puesto que el valor de fin no se ha definido, PC-DMIS define el valor de fin como "CIR1.CANTCONT":



```
CIR3      ==ELEM/CIRCULO, CARTESIANA, DENTRO, NO
          TEO/<20.97629, 22.90352, 0>, <0, 0, -1>, 20.97629
          REAL/<20.96578, 22.89023, -0.01243>, <0, 0, -
1>, 20.96578
          CONST/CIRCULO, INV, CIR1.CONTACTO[1..]
```

Ejemplo 3



En este ejemplo, no se ha definido ni el valor de inicio ni el de fin. Por eso, PC-DMIS define el valor de inicio como "1" y el valor de fin como "CIR1.CANTCONT":



```
CIR3      ==ELEM/CIRCULO, CARTESIANA, DENTRO, NO
          TEO/<20.97629, 22.90352, 0>, <0, 0, -1>, 20.97629
          REAL/<20.96578, 22.89023, -0.01243>, <0, 0, -
1>, 20.96578
          CONST/CIRCULO, INV, CIR1.CONTACTO[. .]
```

En las secciones siguientes se describen algunas funciones de matrices adicionales que resultan de utilidad para localizar los puntos mínimos o máximos en un escaneado:

Asignar un rango de contactos a una matriz

También puede asignar un rango de contactos a una matriz mediante esta sintaxis:

<ID elemento>.<Tipo contacto>[<Número inicial>.<Número final>].<Extensión>

donde

<ID elemento> es el nombre del elemento.

<Tipo contacto> puede ser la palabra "HIT" para los datos compensados o "RAWHIT" para los datos no compensados. Si la compensación de sonda está desactivada, los valores devueltos siempre son sin compensar.

<Número inicial> es una expresión que identifica el primer valor de índice del rango de contactos.

<Número final> es una expresión que identifica el segundo valor de índice del rango de contactos.

<Extensión> identifica el tipo de datos. Las extensiones posibles son los tipos de datos medidos o teóricos que se enumeran en la sección siguiente.



Ejemplo de utilización de valores de inicio y fin implícitos en una expresión.

Supongamos que desea encontrar todos los puntos de contacto que definen un plano (PLN1). Puede escribirlo así:

```
PLN1.CONTACTO[1..PLN1.CANTCONT]
```

Una forma más corta de hacer esto es utilizar valores de inicio y fin de contacto inferidos. Puede escribir el mismo código así:

```
PLN1.CONTACTO[. .]
```

En este caso, como no hemos definido los valores de inicio y fin de contacto, PC-DMIS presupone que el valor de inicio de contacto es 1 y que el valor de fin de contacto es CANTCONT.

Medido

- X: Valores X medidos de los contactos
- Y: Valores Y medidos de los contactos
- Z: Valores Z medidos de los contactos
- XYZ: Valores XYZ medidos de los contactos
- I: Valores I medidos de los contactos
- J: Valores J medidos de los contactos
- K: Valores K medidos de los contactos
- IJK: Valores IJK medidos de los contactos

Teórico

- TX: Valores X teóricos de los contactos
- TY: Valores Y teóricos de los contactos
- TZ: Valores Z teóricos de los contactos
- TXYZ: Valores XYZ teóricos de los contactos
- TI: Valores I teóricos de los contactos
- TJ: Valores J teóricos de los contactos
- TK: Valores K teóricos de los contactos
- TIJK: Valores IJK teóricos de los contactos

Por ejemplo:

```
ASIGN/V1=ESCANEADO1.HIT[1..10].X
```

V1 se asigna a una matriz de 10 valores que son los valores X medidos a partir de los 10 primeros contactos de ESCANEADO1.

```
ASIGN/V2=ESCANEADO1.HIT[1..ESCANEADO1.CANTCONT].XYZ
```

V2 se asigna a una matriz de puntos procedentes de cada uno de los centroides de los contactos del escaneado.

Ordenar matrices

PC-DMIS permite ordenar las matrices en orden ascendente o descendente. Las dos expresiones siguientes toman una matriz y devuelven una matriz ordenada:

Para ordenar una matriz en orden *ascendente*, utilice:

```
SORTUP(<matriz>)
```

Para ordenar una matriz en orden *descendente*, utilice:

```
SORTDOWN(<matriz>)
```

Por ejemplo:

```
ASIGN/V1=ARRAY(5,8,3,9,2,6,1,7)
```

Se asigna a V1 la matriz "5,8,3,9,2,6,1,7".

```
ASIGN/V2=SORTUP(V1)
```

V2 contendrá los valores de la matriz en orden ascendente: "1,2,3,5,6,7,8,9".

```
ASIGN/V3=SORTDOWN(V1)
```

V3 contendrá los valores de la matriz en orden descendente: "9,8,7,6,5,3,2,1".

Devolver los valores de índice más alto y más bajo de una matriz

Puede utilizar una matriz como entrada para una función y obtener el número de índice del elemento que tiene el valor más alto o más bajo utilizando estas funciones:

Para obtener el valor de índice *más alto* del elemento, utilice:

```
MAXINDEX(<matriz>)
```

Para obtener el valor de índice *más bajo* del elemento, utilice:

```
MININDEX(<matriz>)
```

Por ejemplo:

```
ASIGN/V1=ARRAY(5,8,3,9,2,6,1,7)
```

Se asigna a V1 la matriz "5,8,3,9,2,6,1,7".

```
ASIGN/V2=MAXINDEX(V1)
```

V2 contendrá el valor de índice 4 de la matriz. El valor real de ese elemento de la matriz es 9.

```
ASIGN/V3=MININDEX(V1)
```

V3 contendrá el valor de índice 7 de la matriz. El valor real de ese elemento de la matriz es 1.

A continuación puede utilizar los valores de índice devueltos para obtener el valor real del elemento de la matriz.

Devolver los valores de índice clasificados de una matriz

Puede utilizar una matriz como entrada para una función, clasificar los valores de la matriz en orden ascendente o descendente y después obtener los valores de índice mediante las funciones siguientes:

Para devolver las posiciones de índice de la matriz con sus valores ordenados de más alto a más bajo, utilice:

```
MAXINDICES(<matriz>)
```

Para devolver las posiciones de índice de la matriz con sus valores ordenados de más bajo a más alto, utilice:

```
MININDICES(<matriz>)
```

Por ejemplo:

Usar expresiones y variables

```
ASIGN/V1=ARRAY (4,8,2,9,5,7)
```

Se asigna a V1 la matriz "4,8,2,9,5,7".

```
ASIGN/V2=MAXINDICES (V1)
```

V2 contendrá una matriz con estos valores: "4,2,6,5,1,3".

```
ASIGN/V3=MININDICES (V1)
```

V3 contendrá una matriz con estos valores: "3,1,5,6,2,4".

Ejemplo de utilización de funciones de matriz para localizar los puntos mínimo y máximo en un escaneado

El objetivo principal de las funciones de matriz de contactos descritas anteriormente es proporcionar al usuario una manera sencilla de localizar los puntos mínimo y máximo de un escaneado.

Para dimensionar el punto de ESCANEADO1 que tiene el valor X medido más alto, puede utilizar esta expresión:



```
ASIGN/INDPUNTMAX=MAXINDEX (ESCANEADO1.HIT[1..ESCANEADO1.NU  
MHITS].X)
```

```
D1=POSICIÓN DE ELEMENTO ESCANEADO1.HIT[INDPUNTMAX]
```

Para buscar los tres puntos más altos en el eje Z de ESCANEADO2, puede utilizar esta expresión:



```
ASIGN/MI=MAXINDICES (ESCANEADO2.HIT[1..ESCANEADO2.NUMHITS]  
.Z)
```

```
ASIGN/TRESPUNTOS=ARRAY (ESCANEADO2.HIT[MI[1]].XYZ,  
ESCANEADO2.HIT[MI[2]].XYZ, ESCANEADO2.HIT[MI[3]].XYZ)
```

Matrices de variables

No es necesario declarar las matrices de variables. Las matrices de variables cobran existencia por medio de la instrucción ASIGN cuando al evaluar la expresión situada en el lado derecho de la instrucción ASIGN se obtiene una matriz o cuando el lado izquierdo de la instrucción ASIGN hace referencia a un componente (o elemento) de una matriz de variables.

```
ASIGN/V1=Matriz(3,4,5,6,7)
```

Crea una matriz de 5 elementos y la asigna a V1.

```
ASIGN/V2=V1[3]
```

Asigna a V2 el valor del tercer elemento de la matriz V1: 5.

```
ASIGN/V1[4]=23
```

Asigna 23 al cuarto elemento de la matriz V1.

Las matrices se crean y se asignan de forma dinámica. Esto permite crear una matriz utilizando una referencia a matriz en el lado izquierdo de una instrucción ASIGN.

```
ASIGN/V3[5]=8
```

Crea dinámicamente una matriz donde el quinto elemento es igual a 8.

Cuando se crean referencias a un componente de una matriz que nunca ha recibido un valor, la evaluación de la expresión de la matriz arroja 0.

```
ASIGN/V3[5]=8
```

```
ASIGN/V4=V3[5]
```

V4 recibe el valor de 8.

```
ASIGN/V5=V3[6]
```

Si nunca se ha establecido un valor para el sexto componente de V3, V5 toma el valor 0.

Al igual que otros tipos de matrices, se pueden utilizar expresiones entre corchetes.

```
ASIGN/V3[5]=8
```

```
ASIGN/V4=V3[2+3]
```

V4 recibe el valor de 8.

Las matrices de variables pueden tener varias dimensiones.

```
ASIGN/V6=Array(Array(4,7,2),Array(9,2,6))
```

V6 se establece con una matriz dimensional de 2 por 3, donde V6[1, 1] es igual a 4, V6[1, 2] es igual a 7, V6[1, 3] es igual a 2, V6[2, 1] es igual a 9, V6[2,2] es igual a 2 y V6[2,3] es igual a 6.

```
ASIGN/V7=V6[2,1]
```

V7 recibe el valor 9.

Las matrices de variables pueden tener índices negativos:

Usar expresiones y variables

```
ASIGN/V8[-3]=5
```

El índice -3 de la matriz V8 se establece en 5.

La asignación de matrices prevalece sobre los valores anteriores:

```
ASIGN/V8="Hola"
```

La variable V8 es igual a la cadena "Hola".

```
ASIGN/V8[2]=5
```

V8 ya no es una cadena, sino una matriz, y su segundo componente tiene un valor de 5.

```
ASIGN/V8=9
```

V8 ya no es una matriz, sino un número entero con valor 9.

Las matrices se pueden componer con elementos de varios tipos:

```
ASIGN/V9=Array("Hola",3,2.9,{ELEM1})
```

Crea la matriz V9 con cuatro elementos. El primer componente es una cadena, el segundo un entero, el tercero un número real y el cuarto un puntero a ELEM1.

El tamaño de las matrices se puede aumentar para que incluya más elementos:



```
ASIGN/V10=ARRAY(3,1,5)
```

```
ASIGN/V10[LEN(V10)+1]=7
```

La primera instrucción crea una matriz inicial V10 con tres elementos (3, 1 y 5). A continuación, la segunda instrucción aumenta el tamaño de la matriz en V10 en un elemento, y asigna al elemento final el valor 7.

Operadores para expresiones

En PC-DMIS existen los operadores básicos siguientes:

+ Suma: *<Expresión> + <Expresión>*

Suma las dos expresiones. En el caso de cadenas, se concatenan.

- Resta: *<Expresión> - <Expresión>*

Resta la segunda expresión de la primera expresión.

* Multiplicación: *<Expresión> * <Expresión>*

Multiplica las dos expresiones.

/ División: *<Expresión> / <Expresión>*

Divide la primera expresión entre la segunda expresión.

^ Exponenciación: *<Expresión> ^ <Expresión>*

Eleva la primera expresión a la potencia de la segunda expresión.

% Módulo: *<Expresión> % <Expresión>*

Devuelve el resto de una expresión dividida entre la otra.

- Inverso aditivo *-<Expresión>*

Devuelve el inverso aditivo de la expresión.

! Negación lógica: *!<Expresión>*

Devuelve la negación lógica de la expresión.

== Igual a: *<Expresión> == <Expresión>*

Da como resultado 1 si las expresiones son iguales. De lo contrario, da como resultado 0. Se utilizan dos signos de igual para distinguirlo del operador = de la instrucción de asignación.

<> No igual a: *<Expresión> <> <Expresión>*

Da como resultado 1 si la expresión no es igual. De lo contrario, da como resultado 0.

> Mayor que: *<Expresión> > <Expresión>*

Da como resultado 1 si la primera expresión es mayor que la segunda expresión. De lo contrario, da como resultado 0.

>= Igual o mayor que: *<Expresión> >= <Expresión>*

Da como resultado 1 si la primera expresión es mayor o igual que la segunda expresión. De lo contrario, da como resultado 0.

< Menor que: *<Expresión> < <Expresión>*

Usar expresiones y variables

Da como resultado 1 si la primera expresión es menor que la segunda expresión. De lo contrario, da como resultado 0.

<= Igual o menor que: *<Expresión> <= <Expresión>*

Da como resultado 1 si la primera expresión es menor o igual que la segunda expresión. De lo contrario, da como resultado 0.

AND Conjunción lógica: *<Expresión> AND <Expresión>*

Da como resultado 1 si las dos expresiones no dan como resultado 0. De lo contrario, da como resultado 0.

OR Disyunción lógica: *<Expresión> OR <Expresión>*

Da como resultado 1 si una de las dos expresiones no da como resultado 0. De lo contrario, da como resultado 0.

() Paréntesis: *(<Expresión>)*

Da precedencia a la evaluación de la expresión incluida entre paréntesis.

Precedencia

Las expresiones se evalúan de acuerdo con la precedencia que se indica a continuación (la lista está ordenada desde la precedencia superior a la inferior).

Grado superior de precedencia

- Operandos
- (menos unario), !, (), funciones (por ejemplo, ABS, COS, STR, LEN, CROSS, etc.)
- ^
- *, /, %
- +, -
- ==, <>, <, <=, >, >=
- Y
- O

Grado inferior de precedencia

Funciones

Las funciones son expresiones propias de PC-DMIS o definidas por el usuario que normalmente toman parámetros y luego devuelven resultados. Los parámetros se insertan en la expresión antes de evaluarla.

Lista de funciones

La lista siguiente contiene todas las funciones disponibles para el lenguaje de expresiones de PC-DMIS.

- ABS (matemática)
- ACOS (matemática)
- ANGLEBETWEEN (punto)
- ARCSEGMENTENDINDEX (varios)
- ARCSEGMENTSTARTINDEX (varios)
- ARRAY (matriz)
- ASIN (matemática)
- ATAN (matemática)
- CHR (cadena)
- CONCAT (cadena)
- COS (matemática)
- CROSS (punto)
- DEG2RAD (matemática)
- DELTA (punto)
- DIST2D (puntero)
- DIST3D (puntero)
- DOT (punto)
- ELEMENT (cadena)
- EOF (varios)
- EOL (varios)
- EQUAL (matriz)
- EQUAL (cadena)
- EXP (matemática)
- FORMAT (cadena)
- FUNCTION (función)
- GETCOMMAND (puntero)
- GETPROGRAMINFO (cadena)
- GETROTABDATA (varios)
- GETSETTING (cadena)
- GETTEXT (cadena)

Usar expresiones y variables

- GETTRACEVALUE (cadena)
- IF (varios)
- INDEX (cadena)
- ISIOCHANNELSET (varios)
- LEFT (cadena)
- LEN (matriz)
- LEN (puntero)
- LEN (cadena)
- LINESEGMENTENDINDEX (varios)
- LINESEGMENTSTARTINDEX (varios)
- LN (matemática)
- LOG (matemática)
- LOWERCASE (cadena)
- MAX (matriz)
- MID (cadena)
- MIN (matriz)
- MPOINT (punto)
- ORD (cadena)
- PCDMISAPPLICATIONPATH (cadena)
- PCDMISUSERHIDDEN DATAPATH (cadena)
- PCDMISUSERVISIBLE DATAPATH (cadena)
- PCDMISSYSTEMHIDDEN DATAPATH (cadena)
- PCDMISSYSTEMVISIBLE DATAPATH (cadena)
- PCDMISSYSTEMREPORTINGPATH (cadena)
- PROBEDATA (varios)
- QUALTOOLDATA (varios)
- RAD2DEG (matemática)
- RIGHT (cadena)
- ROUND (matemática)
- SETROTABDATA (varios)
- SIN (matemática)
- SQRT (matemática)
- SYSTEMDATE (cadena)
- SYSTEMTIME (cadena)
- SYSTIME (cadena)
- TAN (matemática)
- TUTORELEMENT (varios)
- UNIT (punto)
- UPPERCASE (cadena)

Funciones de cadena

Con las cadenas de texto se utilizan las funciones siguientes.

CHR

Conversión de caracteres: *CHR(<Entero>)*

Esta función devuelve una cadena que contiene el carácter correspondiente al valor decimal en ASCII.

CONCAT

Esta función concatena en una única cadena todas las cadenas especificadas en las expresiones 1 a N: *CONCAT (<expresión1>, <expresión2>, & <expresiónN>)*

ELAPSEDEXECUTIONTIME

Tiempo de ejecución transcurrido con formato: *ELAPSEDEXECUTIONTIME()*

Esta función devuelve el tiempo que ha transcurrido desde que la rutina de medición o la minirrutina empezó a ejecutarse. El tiempo de ejecución transcurrido es el tiempo dedicado a la parte de DCC de la ejecución; no se hace un seguimiento del tiempo de las pausas cuando se precisa atención por parte del usuario. Dichas pausas corresponden a las pausas durante la ejecución de los comentarios o los mensajes de PC-DMIS, y los mensajes de error que pueden detener por completo la ejecución.

Puede registrar el tiempo de ejecución transcurrido en cualquier punto de la rutina de medición o de la minirrutina; para ello, asigne la función a una variable, como por ejemplo:



```
ASIGN/V1=ELAPSEDEXECUTIONTIME ()
```

El formato por omisión de tiempo que se devuelve es "hh:mm:ss". Puede medir también el tiempo de ejecución transcurrido en otros formatos:

- Utilice *ASIGN/V1=FORMATO(ELAPSEDEXECUTIONTIME(),"hh:mm:ss")* o *ASIGN/V1=ELAPSEDEXECUTIONTIME()* para obtener el tiempo en horas, minutos y segundos.

Usar expresiones y variables

- Utilice `ASIGN/V1=FORMATO(ELAPSEDEEXECUTIONTIME(),"mm:ss")` para obtener el tiempo en minutos y segundos.
- Utilice `ASIGN/V1=FORMATO(ELAPSEDEEXECUTIONTIME(),"ss")` para obtener el tiempo en segundos.

ELEMENT

Ubicación de subcadena delimitada: `ELEMENT(<Entero>, <Cadena1>, <Cadena2>)`

Esta función devuelve la enésima subcadena (elemento) de `cadena2` utilizando `cadena1` como texto delimitador que divide los elementos que forman `cadena2`.



supongamos que `cadena2` es "6, 12, 8, 4, 5" y `cadena1` es un carácter de coma (","). Los cinco elementos que es posible recuperar de forma individual por medio del comando `ELEMENT` son "6", "12", "8", "4" y "5".

EQUAL

Comparación de cadenas sin distinguir entre mayúsculas y minúsculas:
`EQUAL(<Cadena1>, <Cadena2>)`

Esta función compara dos cadenas (sin distinguir entre mayúsculas y minúsculas) para determinar si son idénticas. Devuelve un entero (1) si las cadenas son iguales, y 0 si no lo son.

FORMAT

Formato: `FORMAT(<Cadena>,<Entero, doble o punto>)`

Esta función toma dos expresiones y devuelve una cadena con formato, de forma similar a la función `sprintf` en C++.

- La expresión 1 debe ser de tipo *cadena* y contener uno o tres especificadores de formato. Si es de otro tipo, el evaluador de expresiones intenta convertirla en una cadena. La cadena debe contener *un* especificador de formato si la segunda expresión es de tipo entero o doble y *tres* especificadores de formato (consulte los párrafos siguientes) si es de tipo punto.
- La segunda expresión se espera que sea del tipo *entero, doble o punto*. Si se utiliza otro tipo, el valor de la expresión es 0.

Especificador de formato para función **FORMAT**:

El especificador de formato debe tener la misma sintaxis que los especificadores de formato utilizados en la función *sprintf* del lenguaje de programación C++.

Un especificador de formato consta de campos opcionales y obligatorios y tiene la sintaxis siguiente:

%[indicadores] [anchura] [.precisión] tipo

Cada campo del especificador de formato es un único carácter o un número que indica una opción de formato determinada. El especificador de formato más sencillo utiliza solamente el signo de porcentaje y un carácter de tipo (por ejemplo, %d). Si el signo de porcentaje va seguido de un carácter que no tiene ningún significado como campo de formato, el carácter se copia en STDOUT. Por ejemplo, para imprimir un carácter de porcentaje, utilice %%.

Los campos opcionales de indicador, anchura y precisión, que aparecen delante del carácter de tipo, controlan otros aspectos del formato. Se describen a continuación:

indicadores

Estos *caracteres opcionales* controlan la justificación de la salida y la impresión de símbolos, espacios en blanco, separadores decimales y prefijos octales o hexadecimales. En un especificador de formato puede aparecer más de un indicador.

Éstos son los indicadores posibles:

–

Significado: Alinea el resultado A la izquierda dentro de la anchura de campo dada.

Valor por omisión: Alineación por la derecha.

+

Significado: Coloca en el valor de salida un signo como prefijo (+ o –) si el valor de salida es del tipo con signo.

Valor por omisión: El signo solo aparece en el caso de los valores con signo negativo (–).

0

Significado: Si la anchura tiene el prefijo 0, se añaden ceros hasta que se alcanza la anchura mínima. Si aparecen 0 y –, los 0 se pasan por alto. Si se especifica 0 con un formato de enteros (i, u, x, X, o, d), el 0 se pasa por alto.

Por omisión: Sin relleno.

vacío (' ')

Significado: Coloca en el valor de salida un espacio en blanco como prefijo si el valor de salida es con signo y positivo; el espacio en blanco se pasa por alto si aparecen los indicadores espacio en blanco y +.

Valor por omisión: No aparece ningún espacio en blanco.

#

Significado 1: Cuando se utiliza con el tipo o, x o X, el indicador # añade 0, 0x o 0X como prefijo en cualquier valor de salida que no sea cero.

Valor por omisión 1: No aparece ningún prefijo.

Significado 2: Cuando se utiliza el tipo e, E o f, el indicador # hace que el valor de salida contenga un separador decimal en todos los casos.

Valor por omisión 2: El separador decimal sólo aparece si hay dígitos detrás.

Significado 3: Cuando se utiliza con el formato g o G, el indicador # hace que el valor de salida contenga un separador decimal en todos los casos e impide que se trunquen los ceros finales.

Valor por omisión 3: El separador decimal sólo aparece si hay dígitos detrás. Los ceros finales se truncan. Se pasa por alto cuando se utiliza con d, i o bien u.

Anchura

Este segundo campo opcional, o argumento, controla el número mínimo de caracteres impresos. Es un entero decimal no negativo.

- Si el número de caracteres del valor de salida es menor que la anchura especificada, se añaden espacios en blanco a la izquierda o a la derecha del valor, en función de si se especifica el indicador – (alineación izquierda), hasta que se alcanza la anchura mínima.
- Si la anchura tiene el prefijo 0, se añaden ceros hasta que se alcanza la anchura mínima (no se aplica a los números alineados por la izquierda).
- La especificación de anchura nunca hace que se trunque un valor. Si el número de caracteres del valor de salida es mayor que la anchura especificada o si no se proporciona ninguna anchura, se imprimen todos los caracteres del valor (teniendo en cuenta la especificación de precisión que aparece más adelante).

precisión

Este tercer campo opcional, o argumento, especifica el número de caracteres que se imprimirán, el número de posiciones decimales o el número de dígitos significativos. A diferencia de la especificación de anchura, la especificación de la precisión puede hacer que el valor de salida se trunque o que un valor de coma flotante se redondee. Es un entero decimal no negativo, con un punto (.) delante.

tipo

Este carácter obligatorio determina si el argumento asociado es entero, doble o un punto. Los tipos disponibles son los siguientes:

d: entero decimal con signo

ii: entero decimal con signo

o: entero octal sin signo

u: entero decimal sin signo

x: entero hexadecimal sin signo, utilizando "abcdef"

X: entero hexadecimal sin signo, utilizando "ABCDEF"

e: doble en formato exponencial [-]d.dddd e [signo]ddd

E: igual que e, con la diferencia de que utiliza E para introducir el exponente

f: doble con el formato [-]dddd.dddd

g: toma el formato e o f en función de cuál sea más compacto

G: igual que g, con la diferencia de que utiliza E para introducir el exponente

Ejemplo de FORMAT

En este ejemplo se muestran varias instrucciones que utilizan la función FORMAT en una rutina de medición:

ASIGN/V1=PROBEDATA("OFFSET")	V1 pasa a ser de tipo punto; representa los offsets de la sonda actual. Al utilizar los valores de la rutina de medición usados en este ejemplo, V1 será: <-1.8898, 1.8898, 5.704>
ASIGN/V3=FORMAT("%.5f,%.5f,%.5f",V1)	V3 pasa a ser de tipo cadena. La cadena toma un formato que utiliza el objeto de punto de la variable V1. V3 ahora contiene: -1.88976, 1.88976, 5.70403
ASIGN/V4=1.123456789	V4 pasa a ser de tipo doble.
ASIGN/V5=FORMAT("%.5f",V4)+FORMAT("%.6f",V4)+FORMAT("%.7f",V4)+FORMAT("%.8f",V4)	V5 pasa a ser de tipo cadena con este valor: 1.12346 1.123457 1.1234568 1.12345679

Usar expresiones y variables

ASIGN/V6A="El valor de V4 es: "+FORMAT("%.8f", V4)	V6A pasa a ser de tipo cadena con este valor: El valor de V4 es: 1.12345679
ASIGN/V6B=FORMAT("El valor de V4 es: %.8f", V4)	El resultado de la expresión es el mismo que el de V6A.
ASIGN/V7=4444	V7 pasa a ser de tipo doble, puesto que se presupone que todos los números son dobles a menos que se fuerce que sean enteros.
ASIGN/V8=FORMAT("%o",INT(V7))	V8 pasa a ser de tipo cadena con este valor: 10534
ASIGN/V9=FORMAT("%u",INT(-1))	V9 pasa a ser de tipo cadena con este valor: 4294967295
ASIGN/V10=FORMAT("%x",INT(2143))	V10 pasa a ser de tipo cadena con este valor: 85f
ASIGN/V11=FORMAT("%X",INT(9567))	V11 pasa a ser de tipo cadena con este valor: 255F
ASIGN/V12=FORMAT("%e",0.0005432)	V12 pasa a ser de tipo cadena con este valor: 5.432000e-004
ASIGN/V13=FORMAT("%E",145.3421)	V13 pasa a ser de tipo cadena con este valor: 1.453421E+002
ASIGN/V14=FORMAT(",%6d,",INT(1))	V14 pasa a ser de tipo cadena con este valor: , 1,
ASIGN/V15=FORMAT(",%-6d,",INT(1))	V15 pasa a ser de tipo cadena con este valor: ,1 ,

GETSETTING

Esta función devuelve varios valores de PC-DMIS dependiendo del parámetro de cadena insertado.

GETSETTING(<Cadena>)

Puede utilizar estos parámetros de cadena:

- "DCC Mode": Devuelve un 1 si PC-DMIS está en modo DCC y 0 en caso contrario.
- "Manual Mode": Devuelve un 1 si PC-DMIS está en modo Manual y 0 en caso contrario.
- "Current Alignment": Devuelve una cadena de la alineación actual.
- "Current Workplane": Devuelve una cadena del plano de trabajo actual.
- "Workplane Value": Devuelve un valor numérico del plano de trabajo actual.
- "PreHit": Devuelve el valor del precontacto actual como número de doble precisión.
- "Retract": Devuelve el valor de retracción actual como número de doble precisión.
- "Check": Devuelve el valor de revisión actual como número de doble precisión.
- "Touch Speed": Devuelve el valor de velocidad de toque actual como número de doble precisión.
- "Move Speed": Devuelve el valor de velocidad de movimiento actual como número de doble precisión.
- "Fly Mode": Devuelve un 1 si PC-DMIS utiliza el modo Fly y un 0 en caso contrario.
- "Ph9 present": Devuelve un 1 si Ph9/Ph10 está presente y un 0 en caso contrario.
- "Manual CMM": Devuelve un 1 si la CMM es manual y un 0 en caso contrario.
- "LangStr(<Número o ID>)" –: Devuelve una cadena de los recursos de PC-DMIS en el idioma actual a partir de un número o de una de estas ID:

"Yes", "No", "Oper", "Rept", "Input", "Doc", "YesNo", "Readout", "Internal", "External", "Rect", "Polr ", "Out", "In", "Least_Sqr", "Min_Sep", "Max_Insc", "Min_CircSc", "Fixed_Rad", "Workplane", "Xaxis", "YAxis", "ZAxis", "Xplus", "Xminus", "YPlus", "YMinus", "ZPlus", "ZMinus", "Point", "Plane", "Line", "Circle", "Sphere", "Cylinder", "Round_Slot", "Square_slot", "Cone", o "None".

Si el valor que utiliza es un número positivo, PC-DMIS extrae la cadena del archivo resource.dll. Si utiliza un número negativo, PC-DMIS extrae la cadena del archivo strings.dll (la tabla de cadenas).

- "Extended Sheet Metal": Devuelve un 1 si la casilla **Mostrar opciones extendidas de chapa metálica** está seleccionada en el cuadro de diálogo **Opciones de configuración** y un 0 en caso contrario.

Usar expresiones y variables

- "LastHitMove(X)": Devuelve el valor X del comando CONT/BASE o MOV/PUNTO más reciente. PC-DMIS debe estar en modo DCC para que funcione.
- "LastHitMove(Y)": Devuelve el valor Y del comando CONT/BASE o MOV/PUNTO más reciente. PC-DMIS debe estar en modo DCC para que este parámetro funcione.
- "LastHitMove(Z)": Devuelve el valor Z del comando CONT/BASE o MOV/PUNTO más reciente. PC-DMIS debe estar en modo DCC para que este parámetro funcione.

Puede utilizar la función GETSETTING para determinar si PC-DMIS está en modo manual o DCC, como se muestra a continuación:

`ASIGN/VARMODODCC=GETSETTING("DCC Mode")`: Asigna a la variable VARMODODCC el valor 1 si PC-DMIS está en modo DCC y el valor 0 en caso contrario.

`ASIGN/VARMODOMAN=GETSETTING("Manual Mode")`: Asigna a la variable VARMODOMAN el valor 1 si PC-DMIS está en modo manual y el valor 0 en caso contrario.

Puede utilizar la función GETSETTING para determinar el plano de trabajo actual, como se muestra a continuación:

`ASIGN/ID_PLANOTRABAJO=GETSETTING("Current Workplane")`: Asigna a la variable ID_PLANOTRABAJO el valor de cadena del plano de trabajo actual (Z+, Z-, etc.).

`ASIGN/VALOR_PLANOTRABAJO=GETSETTING("Workplane Value")`: Asigna a la variable VALOR_PLANOTRABAJO un valor numérico que representa el plano de trabajo. Los planos de trabajo tienen asociados estos valores: Z+ = 0, Z- = 3, X+ = 1, X- = 4, Y+ = 2 o Y- = 5.

GETTEXT

Esta función devuelve el texto actual del campo de datos especificado:

`GETTEXT(<Cadena o Entero>, <Entero>, <Puntero>)`

Esta función tiene tres campos.

Primer campo: Número o descripción del campo de datos

El primer campo puede ser una cadena que describa el campo de datos, indicada como (A) en la imagen que aparece a continuación, o bien el número del campo de datos, indicado como (C).



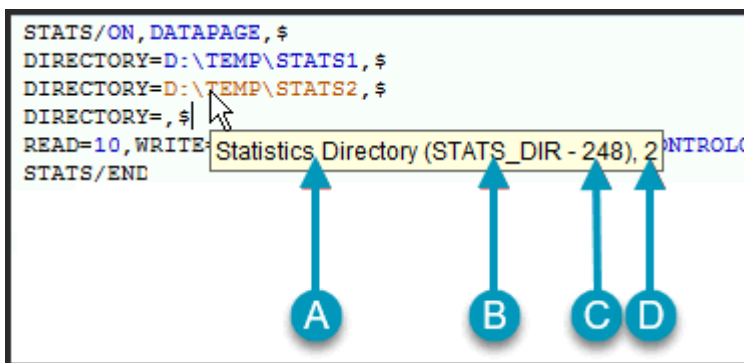
El elemento (B) de la imagen siguiente no se utiliza en esta función, pero a veces se emplea en las expresiones de informe o automatización.

Para obtener estos valores:

1. Coloque PC-DMIS en modo Comando. Haga clic con el botón derecho del ratón en cualquier punto de la ventana de edición. Aparecerá un menú de acceso directo.
2. En el menú de acceso directo, seleccione **Cambiar pantalla desplegable** y luego **Información de tipo de datos**.
3. En la ventana de edición, coloque el ratón sobre un campo de datos. Aparecen la descripción, el número y el índice del tipo del elemento de datos correspondiente.



La descripción del tipo puede ser diferente para cada idioma. Si la rutina de medición se ejecuta en una versión de PC-DMIS con un idioma distinto, utilice el número de tipo en su lugar.



Información de tipo de datos de ejemplo en la que se muestra: (A) la descripción del tipo, (B) el identificador de cadena del tipo, (C) el número del tipo y (D) el índice del tipo

Segundo campo: Índice del tipo

El segundo campo es el índice del tipo, indicado como (D) en la imagen anterior. Este campo suele ser cero a menos que tenga más instancias del mismo tipo de campo en el mismo comando, como es el caso de los varios campos DIRECTORY de la imagen anterior. El valor correcto de este campo puede obtenerse de igual forma que para el primer campo.

Tercer campo: Puntero a comandos

El tercer campo corresponde a un puntero a comandos. Apunta al comando que contiene el campo desde el que se obtiene el texto. Puede utilizar la anotación del puntero a comandos (es decir, {F15}) para especificar este campo, o puede utilizar la expresión GETCOMMAND como se muestra a continuación:



`ASIGN/V1=GETTEXT("Tipo de cálculo para mejor ajuste",0,{F15})` Este comando asigna a V1 el valor actual del alternante de tipo de cálculo para mejor ajuste del elemento F15.

`ASIGN/V2=GETCOMMAND("Comentario", "SUP", 1)`: Se asigna a V2 un puntero al primer comentario desde la parte superior de la rutina de medición.

`ASIGN/V3=GETTEXT("Tipo de comentario",1,V2)`: A V3 se le asigna el valor del campo alternante Tipo de comentario. Si el primer comentario de la rutina de medición es un comentario que debe mostrarse al operador, el valor de V3 será la cadena "OPER".

Consulte el tema "Funciones de punteros" para obtener información sobre la expresión GETCOMMAND utilizada para la asignación de punteros a los comandos.

GETTEXTEX

Esta función devuelve el texto actual del campo de datos especificado:

`GETTEXTEX(<Cadena o Entero>, <Entero>,<Cadena>,<Puntero>)`

Esta función tiene cuatro campos.

Primer campo: Número o descripción del campo de datos

El primer campo puede ser una cadena que describa el campo de datos o bien el número del campo de datos, indicado como (A) en la imagen siguiente.



Si utiliza el identificador de cadena del tipo en lugar del identificador numérico (elemento [A] en la imagen siguiente), PC-DMIS lo convertirá automáticamente al valor numérico correcto.

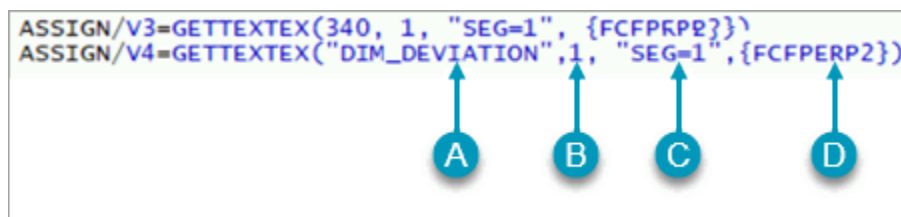
Por ejemplo, si pasa el identificador de cadena "DIM_DEVIATION", internamente PC-DMIS lo convertirá al valor numérico 340. Cuando se pasa el cursor del ratón por encima del comando en la ventana de edición, el comando emergente muestra la cadena de texto, así como el valor de identificador numérico real. En este ejemplo, pues, si pasa el cursor del ratón por encima del comando de la ventana de edición, el comando emergente mostrará (DIM_DEVIATION - 340), 1, SEG=1.

También puede pasar simplemente el valor numérico si lo conoce.

Para obtener estos valores:

1. Pase PC-DMIS a modo Comando y haga clic con el botón derecho del ratón en cualquier punto de la ventana de edición. Aparecerá un menú de acceso directo.
2. En el menú de acceso directo, seleccione **Cambiar pantalla desplegable** y luego **Información de tipo de datos**.
3. En la ventana de edición, coloque el ratón sobre un campo de datos. Aparecen la descripción, el número y el índice del tipo del elemento de datos correspondiente. Pase el cursor del ratón por un D_Type ampliado para que se muestre la cadena de contenido después de los dos puntos.

La descripción del tipo puede ser diferente para cada idioma. Si la rutina de medición se ejecuta en una versión de PC-DMIS con un idioma distinto, utilice el número de tipo en su lugar.



Información de tipo de datos de ejemplo que muestra la cadena del tipo o el identificador numérico (A), el índice del tipo (B), la cadena de contenido (C) y el puntero a comandos (D)

Segundo campo - Índice del tipo

El segundo campo es el índice del tipo, indicado como B en la imagen anterior. Este campo suele ser cero a menos que tenga más instancias del mismo tipo de campo en el mismo comando, como es el caso de los varios campos DIRECTORY de la imagen anterior. Puede obtener el valor correcto de este campo de igual forma que para el primer campo.

Tercer campo - Cadena de contenido

El tercer campo es la cadena de contenido del D_TYPE ampliado, indicada como C en la imagen anterior.

Cuarto campo - Puntero a comandos

El cuarto campo es un puntero a comandos, indicado como D en la imagen anterior. Apunta al comando del que la expresión extrae datos. Puede utilizar la anotación del puntero a comandos (es decir, {FCFPERP2}) o la expresión GETCOMMAND como se muestra a continuación:

```
ASIGN/V1=GETTEXTTEX("DIM_DEVIATION",1,"SEG=1",{FCFPERP2}):
```

Este comando asigna a V1 el valor actual de la desviación del elemento 1, segmento 1, dimensión FCFPERP2.

Consulte el tema "Funciones de punteros" para obtener información sobre la expresión GETCOMMAND utilizada para la asignación de punteros a los comandos.



La función GETTEXTTEX añade compatibilidad con los DTypes ampliados que contienen una cadena de CONTENIDO. Actualmente, solo los comandos de tolerancia geométrica de PC-DMIS utilizan DTypes ampliados.

GETPROGRAMINFO

Esta función devuelve la información de la rutina de medición según los parámetros pasados: `GETPROGRAMINFO(<Cadena>, <Cadena opcional>)`

Esta función tiene como máximo dos cadenas como parámetros. Para la mayoría de los elementos solamente se necesita el primer parámetro. Los campos de cadena no distinguen entre mayúsculas y minúsculas.

Primer campo - cadena

El primer campo es una entrada de cadena en la que se indica qué información se devolverá.

CADMODELFILE: Devuelve la ruta completa del archivo del modelo de CAD que ha importado a la rutina de medición.

CADMODELFILENAME: Devuelve solamente el nombre del modelo de CAD (sin la ruta) que ha importado a la rutina de medición.

DATE: Devuelve la fecha actual.

DRAWING: Al igual que REVISION, también devuelve el número de revisión según se define en el encabezado.

ELAPSEDTIME: Devuelve el tiempo que ha transcurrido desde el inicio de la ejecución.

FILENAME: Devuelve el nombre del archivo de la rutina de medición (.prg).

NUMMEAS: Devuelve el número de dimensiones ejecutadas.

NUMOOT: Devuelve el número de dimensiones fuera de tolerancia ejecutadas.

PARTNAME: Devuelve el nombre de la pieza según se define en el encabezado de la rutina de medición.

PARTPATH: Devuelve la ruta completa al archivo de la rutina de medición.

PCDMISVERSION: Devuelve un valor de cadena con el número de la versión de PC-DMIS que está instalada.

PRGSCHEMA: Devuelve un entero que corresponde al número de esquema de PC-DMIS del archivo de la rutina de medición. Se trata de un valor interno utilizado por PC-DMIS para indicar los comandos y las opciones que están serializados.

PRGVERSION: Devuelve un valor de cadena con el número de versión de PC-DMIS del archivo de la rutina de medición. Puede guardar un archivo de rutina de medición para que sea compatible con una versión concreta. Para obtener más información, consulte "Guardar como" en el capítulo "Usar opciones de archivo básicas".

PROBEFILE: Devuelve el nombre del archivo de sonda actual en uso.

REPORTNAME: Devuelve el nombre del archivo de salida actual.

REVISION: Devuelve el número de revisión tal como se define en el encabezado.

SERIALNUM: Devuelve el número de serie según se define en el encabezado.

SEQNUM: Al igual que STATSCOUNT, esta cadena también devuelve el número de cuenta de estadísticas actual.

SHRINK: Devuelve el factor de escala global.

STATSCOUNT: Devuelve la cuenta de estadísticas actual.

TEMP: Devuelve la temperatura para la segunda cadena de entrada (opcional). Consulte "Segundo campo - cadena opcional" a continuación.

TIME: Devuelve la hora actual.

TIPID: Devuelve el nombre de la punta actual en uso.

Segundo campo - cadena opcional

El segundo campo es una entrada de cadena opcional. Solamente es necesario si se utiliza TEMP en el primer campo de entrada. Las cadenas posibles proceden del comando Compensación de temperatura. Para obtener más información, consulte "Compensar la temperatura" en el capítulo "Establecer preferencias" para obtener más información.

HIGH_THRESHOLD: Devuelve el umbral superior de la temperatura,

LOW_THRESHOLD: Devuelve el umbral inferior de la temperatura.

REF_TEMP: Devuelve la temperatura de referencia.

TEMPP: Devuelve la temperatura del sensor de piezas-

TEMPX: Devuelve la temperatura del sensor del eje X.

TEMPY: Devuelve la temperatura del sensor del eje Y.

TEMPZ: Devuelve la temperatura del sensor del eje Z.

Ejemplo

\$\$ NO, Este ejemplo de código muestra el número de dimensiones totales y el número de dimensiones fuera de tolerancia.

```
ASIGN/V1=GETPROGRAMINFO("NUMMEAS")
```

```
ASIGN/V2=GETPROGRAMINFO("NUMOOT")
```

```
COMENTARIO/INFORME
```

```
"Dimensiones totales: "+V1
```

```
"Total fuera de tolerancia: "+V2
```

\$\$ NO, Este ejemplo de código devuelve la temperatura del sensor del eje Z.

```
ASIGN/V3=GETPROGRAMINFO("TEMP", "TEMPZ")
```

```
COMENTARIO/INFORME
```

```
"Temperatura en eje Z: "+V3
```

GETTRACEVALUE

Read Trace Value: *GETTRACEVALUE(<Cadena>)*

Esta función toma un único parámetro de cadena. Devuelve un valor de un comando **CAMPORAST** en la rutina de medición.

<Cadena> representa una cadena, en la que se distingue entre mayúsculas y minúsculas, del nombre de rastreo cuyo valor quiere devolver.

Si tiene varios campos de rastreo con el mismo nombre de rastreo, esta función devuelve el valor del campo de rastreo más reciente por encima de esta función. Si un campo de rastreo no contiene un valor, esta función devuelve el valor 0. En el ejemplo siguiente, "Operador" es el nombre del campo de rastreo en la rutina de medición:



ASIGN/V2=GETTRACEVALUE("Operador")

INDEX

Ubicación de subcadena: *INDEX(<Cadena>, <Cadena>)*

Esta función devuelve la ubicación de la segunda cadena dentro de la primera cadena. La primera letra de la cadena es 1. Si se devuelve cero, la cadena no contiene la subcadena.

Para ver un ejemplo de esta función, consulte el tema "Muestra de código de Leer línea" en el capítulo "Usar entrada/salida de archivos".

LASTEXECUTIONTIME

Tiempo de ejecución más reciente con formato: *LASTEXECUTIONTIME()*

Esta función devuelve el tiempo de ejecución más reciente que PC-DMIS ha registrado y almacenado en el archivo *<nombre de rutina de medición>.MiniRoutines.xml*. El tiempo de ejecución más reciente aparece en el cuadro de diálogo **Ejecución**. El tiempo se devuelve en el formato "hh:mm:ss".

LEFT

Caracteres a la izquierda de una cadena: *LEFT(<Cadena>, <n>)*

Esta función devuelve una cadena formada por un número concreto de los caracteres situados más a la izquierda en la primera expresión (Cadena); la segunda expresión (n) especifica cuántos caracteres se deben incluir en la cadena resultante.

La primera expresión (Cadena) toma necesariamente el tipo cadena. La segunda expresión (n) toma necesariamente el tipo entero.

Para ver un ejemplo de esta función, consulte el tema "Muestra de código de Leer línea" en el capítulo "Usar entrada/salida de archivos".

LEN

Longitud de la cadena: *LEN(<Cadena>)*

Esta función devuelve la cantidad de caracteres que la cadena contiene.

LOWERCASE

Crea una cadena en minúsculas: *LOWERCASE(<Cadena>)*

Esta función devuelve una cadena que es el equivalente en minúsculas de la cadena de la expresión.

MID

Los n caracteres del medio de una cadena: *MID(Cadena>, <Entero>, <Entero opcional>)*

Esta función devuelve una subcadena compuesta de los caracteres de la cadena especificada en el primer parámetro, empezando en la posición especificada por el segundo parámetro y en la longitud de caracteres n especificada en el tercer parámetro. Si no se incluye el tercer parámetro, devuelve el resto de la cadena.

Para ver un ejemplo de esta función, consulte el tema "Muestra de código de Leer línea" en el capítulo "Usar entrada/salida de archivos".

ORD

Conversión a ordinal: *ORD(<Cadena>)*

Esta función devuelve el valor ASCII (número entero) correspondiente a la primera letra de la cadena (0-255).

PCDMISAPPLICATIONPATH

Ruta de acceso completa visualizada: *PCDMISAPPLICATIONPPATH()*

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de aplicación en que se ha instalado PC-DMIS. Este directorio contiene el archivo ejecutable principal y otros archivos de programa necesarios para ejecutar PC-DMIS.

PCDMISUSERHIDDENDATAPATH

Ruta de acceso completa visualizada: *PCDMISUSERHIDDENDATAPATH()*

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de datos de usuario oculto que utiliza PC-DMIS. Consulte el tema "Explicación de las ubicaciones de los archivos" para saber qué archivos contiene este directorio.

PCDMISUSERVISIBLEDATAPATH

Ruta de acceso completa visualizada: PCDMISUSERHIDDENDATAPATH()

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de datos de usuario visible que utiliza PC-DMIS. Consulte el tema "Explicación de las ubicaciones de los archivos" para saber qué archivos contiene este directorio.

PCDMISSYSTEMHIDDENDATAPATH

Ruta de acceso completa visualizada: PCDMISSYSTEMHIDDENDATAPATH()

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de datos de sistema oculto que utiliza PC-DMIS. Consulte el tema "Explicación de las ubicaciones de los archivos" para saber qué archivos contiene este directorio.

PCDMISSYSTEMVISIBLEDATAPATH

Ruta de acceso completa visualizada: PCDMISSYSTEMVISIBLEDATAPATH()

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de datos de sistema visible que utiliza PC-DMIS. Consulte el tema "Explicación de las ubicaciones de los archivos" para saber qué archivos contiene este directorio.

PCDMISSYSTEMREPORTINGPATH

Ruta de acceso completa visualizada: PCDMISSYSTEMREPORTINGPATH()

Esta función devuelve el valor de la cadena que contiene la ruta completa al directorio de informes que utiliza PC-DMIS. Este directorio contiene las plantillas de informe y de etiqueta que se utilizan en la ventana de informes.

RIGHT

Los n caracteres a la derecha de la cadena: *RIGHT*(<Cadena>, <Entero>)

Esta función devuelve una cadena formada por los n caracteres más a la derecha de la cadena, según se especifique en el entero.

SYSTEMDATE

Fecha del sistema: SYSTEMDATE(<Cadena_formato_de_fecha>)

Esta función devuelve una cadena con formato de fecha que incluye los datos de la fecha del día. Por ejemplo, si la fecha actual es 12 de febrero de 2014, el comando `SYSTEMDATE("MM'/'dd'/'yy")` devuelve la cadena "02/12/14".

Utilice los siguientes componentes para crear la cadena de la fecha. Los componentes deben utilizar la combinación de mayúsculas o minúsculas que se indica a continuación (MM, en lugar de mm). Los caracteres no específicos a las fechas (como los espacios) que se incluyan entre los componentes de la cadena del formato de fecha aparecen en la cadena resultante en la misma ubicación que la cadena de entrada. Los caracteres de la cadena de entrada delimitados por comillas simples aparecen en la misma ubicación dentro de la cadena resultante, pero sin las comillas.

d: Día del mes en forma de dígito. No se antepone un cero a las fechas de una sola cifra.

dd: Día del mes en forma de dígitos. Se antepone cero a las fechas de una sola cifra.

ddd: Abreviatura de tres letras del día de la semana.

dddd: Nombre completo del día de la semana.

M: Mes en forma de dígitos sin anteponer ceros a los meses de una sola cifra.

MM: Mes en forma de dígitos; se antepone cero en los meses de una sola cifra.

MMM: Abreviatura de tres letras del mes.

MMMM: Nombre completo del mes.

y: Año en forma de dígitos; no se anteponen ceros en los años de una sola cifra.

yy: Año en forma de dígitos; se antepone cero en los años de una sola cifra.

yyyy: Año representado con cuatro dígitos.

SYSTEMTIME

Hora del sistema con formato: `SYSTEMTIME(<Cadena_formato_de_hora>)`

Esta función devuelve una cadena con formato de hora que incluye los datos de la hora actual. Por ejemplo, el comando `SYSTEMTIME("hh:mm:ss tt")` devuelve la hora en una cadena con formato, de este modo: "11:29:40 PM".

Utilice los siguientes componentes para crear la cadena de la hora. Los componentes deben utilizar la combinación de mayúsculas o minúsculas que se especifica en los ejemplos (**tt** en lugar de **TT**). Los caracteres no específicos a la hora (como los espacios) que se incluyan entre los componentes de la cadena de formato de hora aparecerán en la cadena resultante en la misma ubicación que en la cadena de entrada. Los caracteres de la cadena de entrada delimitados por comillas simples aparecerán en la misma ubicación dentro de la cadena resultante, pero sin las comillas.

h: Horas sin cero delante en las horas con un solo dígito; reloj de 12 horas

hh: Horas con cero delante en las horas con un solo dígito; reloj de 12 horas.

H: Horas sin cero delante en las horas con un solo dígito; reloj de 24 horas

HH: Horas con cero delante en las horas con un solo dígito; reloj de 24 horas

m: Minutos sin cero delante en los minutos con un solo dígito.

mm: Minutos con cero delante en los minutos con un solo dígito

s: Segundos sin cero delante en los segundos con un solo dígito

ss: Segundos con cero delante en los segundos con un solo dígito

t: Un carácter para la notación de antes del mediodía (A) o después del mediodía (P)

tt: Varios caracteres para la notación de antes del mediodía (AM) o después del mediodía (PM)

SYSTIME

Hora del sistema: `SYSTIME()`

Esta función devuelve una cadena con la hora actual del sistema. Esta función es diferente de la función `SYSTEMTIME` descrita anteriormente. Devuelve automáticamente el día, la fecha y la hora, seguidos del año.

Ejemplo: "Mié 12 de febrero 13:50:21 2014"



La cadena que se devuelve muestra la hora del sistema, configurada según las especificaciones de la hora local.

UPPERCASE

Crea una cadena en mayúsculas: *UPPERCASE(<Cadena>)*

Esta función devuelve una cadena que es el equivalente en mayúsculas de la cadena.

Funciones matemáticas

ABS

Valor absoluto: *ABS(<Doble>)*

Devuelve el valor absoluto del valor de entrada.

EXP

Exponencial: *EXP(<Doble>)*

Devuelve el exponencial de la expresión.

LOG

Logaritmo de base 10: *LOG(<Doble>)*

Devuelve el logaritmo de base 10 de la expresión.

LN

Logaritmo natural: *LN(<Doble>)*

Devuelve el logaritmo natural de la expresión.

ROUND

Redondeo: *ROUND(<Doble>)*

Devuelve la entrada redondeada al entero más próximo.

Usar expresiones y variables

SQRT

Raíz cuadrada: *SQRT(<Doble>)*

Devuelve la raíz cuadrada de la entrada.

Funciones de trigonometría

Por omisión, cada una de las funciones trigonométricas toma y devuelve radianes. Si desea que los valores se expresen en grados, utilice la función RAD2DEG descrita a continuación.

ACOS

Arco coseno: *ACOS(<Doble>)*

Devuelve el arco coseno de la expresión. Por ejemplo, ACOS(5.0) devuelve 0. En términos generales, ACOS(<expresión>) devuelve el arco coseno del valor de la expresión.

ASIN

Arco seno: *ASIN(<Doble>)*

Devuelve el arco seno del valor de entrada.

ATAN

Arco tangente: *ATAN(<Doble>)*

Devuelve el arco tangente del valor de la entrada.

COS

Coseno: *COS(<Doble>)*

Devuelve el coseno del valor de la entrada.

DEG2RAD

Grados a radianes: *DEG2RAD(<Doble>)*

Devuelve la entrada dividida por 360 y multiplicada por 2π . Convierte de grados a radianes.

RAD2DEG

Radianes a grados: *RAD2DEG(<Doble>)*

Devuelve la entrada multiplicada por 360 y multiplicada por 2π . Convierte de radianes a grados.

SIN

Sino: *SIN(<Doble>)*

Devuelve el seno del valor de la entrada.

TAN

Tangente: *TAN(<Doble>)*

Devuelve la tangente de la entrada.



Las funciones donde la entrada queda fuera de rango (por ejemplo, en las funciones ACOS, ASIN, LOG, LN, SQRT, etc., que causarían el bloqueo del equipo informático), devuelven el valor 0.

Funciones de puntos

ANGLEBETWEEN

Ángulo entre: *ANGLEBETWEEN(<vector>, <vector>)*

Devuelve en grados el valor del ángulo entre los dos vectores. Los dos parámetros deben ser expresiones que arrojan como valor un tipo de vector. Para obtener el vector desde un elemento, por ejemplo, tiene que utilizar la ID seguida de la extensión .IJK. Lo puede comprobar en el fragmento de código siguiente:


```
F1      =GENÉRICO/PUNTO,DEPENDIENTE,CARTESIANA,$
        NOM/XYZ,<3,3,3>,$
        MED/XYZ,<3,3,3>,$
        NOM/IJK,<1,0,0>,$
        MED/IJK,<1,0,0>

F2      =GENÉRICO/PUNTO,DEPENDIENTE,CARTESIANA,$
        NOM/XYZ,<10,10,10>,$
        MED/XYZ,<10,10,10>,$
        NOM/IJK,<0,0,1>,$
        MED/IJK,<0,0,1>
        ASSIGN/V1=F1.IJK
        ASSIGN/V2=F2.IJK
        ASSIGN/V3=ANGLEBETWEEN(V1,V2)
        COMENTARIO/OPER,NO,PANTALLA
COMPLETA=NO,CONTINUAR AUTO=NO,
        "El ángulo entre "+V1+" y "+V2+" es: "+V3
```

CROSS

Producto vectorial: *CROSS*(<Punto>, <Punto>)

El valor de retorno es del tipo punto y es el vector de unidad en la dirección del producto vectorial de la primera y la segunda expresión.

DELTA

Offset vectorial: *DELTA*(<Punto>, <Punto>, <Doble>)

La función toma la primera expresión (punto) y calcula un punto nuevo en la dirección indicada por la segunda expresión (vector) con el offset especificado en la tercera expresión. Por ejemplo, *DELTA*(*MPOINT*(0,0,0),*MPOINT*(1,0,0),10) devuelve el punto 10,0,0.

DOT

Producto de dos puntos: *DOT*(<Punto>, <Punto>)

Devuelve el punto que es producto de los otros dos puntos (vectores).

UNIT

Vector de unidad: *UNIT*(<Punto>)

Devuelve el punto dividido por su longitud. Por ejemplo, *UNIT*(*MPOINT*(0,0,0)) devuelve el punto 0,0,1.

MPOINT

Coerción de puntos: *MPOINT*(<Expresión1>, <Expresión2>, <Expresión3>)

Hace que las tres expresiones sean de tipo punto, como se muestra en el fragmento de código siguiente:

```
ASIGN/V1=MPOINT (2.5, 3.6, 4)
```

Donde:

V1.X = 2.5

V1.Y = 3.6

V1.Z = 4.0

Consulte "Coerción de puntos".

Funciones de punteros

DIST2D

Distancia bidimensional: *DIST2D*(<ELEM1>, <ELEM2>, <ELEM3>)



Los elementos deben estar encerrados entre llaves.

Calcula la distancia entre los dos primeros argumentos del comando (Elem1 y Elem2), perpendicular al tercer argumento (Elem3).

- Si el tercer elemento es un plano, PC-DMIS calcula la distancia entre los dos primeros argumentos perpendicular al plano.
- Si el tercer elemento es una línea o un cilindro, PC-DMIS calcula la distancia entre los dos primeros argumentos perpendicular al tercer argumento en el plano de trabajo activo.

Por ejemplo, si tiene el plano XY como tercer argumento, tendrá un vector Z+ (0,0,1) y la distancia notificada sólo estará en el eje Z.

Ejemplo



```
ASIGN/V3=DIST2D({CIR1},{CIR2},{PLN1})  
COMENTARIO/OPER,NO,PANTALLA  
COMPLETA=NO,CONTINUAR AUTO=NO,  
V3
```

DIST3D

Distancia tridimensional: DIST3D(<ELEM1>, <ELEM2>)

Calcula la distancia tridimensional entre Elem1 y Elem2.

Los elementos deben estar encerrados entre llaves.

Ejemplo



```
ASIGN/V3=DIST3D({CIR1},{CIR2})  
COMENTARIO/OPER,NO,PANTALLA  
COMPLETA=NO,CONTINUAR AUTO=NO,  
V3
```

GETCOMMAND

Obtiene un puntero al comando especificado por los parámetros:

```
GETCOMMAND(<Entero o Cadena>, <Cadena>,<Entero>
```

Primer parámetro: Campo Información de comandos

El primer parámetro corresponde al campo de información de comandos. Especifica el tipo de comando que hay que buscar. Puede introducirse lo siguiente:

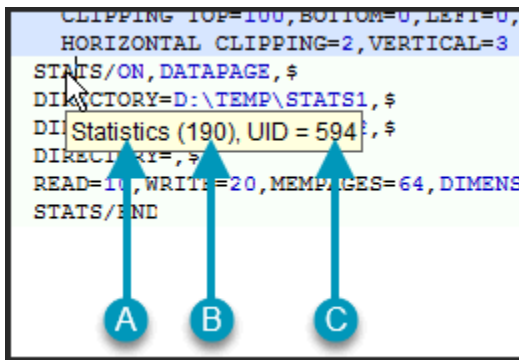
- Una cadena de descripción de comandos. Vea (A) en el gráfico que aparece a continuación.

- El número de un tipo de comando. Vea (B) en el gráfico que aparece a continuación.
- El identificador numérico exclusivo. Vea (C) en el gráfico que aparece a continuación.

Si se introduce la ID exclusiva del comando, no será necesario ningún otro argumento.

Para obtener la cadena de descripción de un comando, el número del tipo de un comando y el identificador exclusivo del número de un comando:

1. Haga clic con el botón derecho del ratón en la ventana de edición.
2. Elija **Cambiar pantalla desplegable | Información de comandos** (PC-DMIS debe estar en modo Comando).
3. Sitúe el ratón encima del comando deseado. PC-DMIS muestra la descripción del comando, el número de tipo y el identificador numérico exclusivo de dicho comando en la ventana emergente.



- A. Cadena de descripción de comandos
- B. Número de tipo de comando
- C. Identificador numérico exclusivo (UID)

Segundo parámetro: Dirección de la búsqueda

El segundo parámetro es la dirección de la búsqueda. Los valores legales incluyen:

Valor	Descripción
ARRIBA	Este valor significa que la búsqueda debe empezar en el comando actual y retroceder hacia arriba.
ABAJO	Este valor significa que la búsqueda debe empezar en el comando actual y avanzar hacia abajo.
SUPERIOR	Este valor significa que la búsqueda debe empezar al principio de la rutina de medición, en dirección hacia abajo.
INFERIOR	Este valor significa que la búsqueda debe empezar en el último objeto de la rutina de medición, en dirección hacia arriba.

Tercer parámetro: Instancia que hay que buscar

El tercer parámetro indica qué instancia del comando se encontrará si la rutina de medición contiene varias instancias del mismo comando.



Si la rutina de medición contiene dos instancias de un comando ESTAD/ACT y desea obtener un puntero a la segunda instancia empezando desde arriba, pase "2" como tercer parámetro y "TOP" como segundo parámetro tal como se muestra aquí.

```
ASIGN/V1=GETCOMMAND("Statistics","TOP",2)
```

Puede utilizar la función GETCOMMAND para indicar el tercer parámetro para la función de cadena GETTEXT. Consulte el tema "Funciones de cadena" para obtener información acerca de GETTEXT.

LEN

Recuento de bucles del puntero: *LEN(<PUNTERO>)*

Devuelve el número de veces que ha estado un puntero en un bucle. Por ejemplo, si el elemento CIR1 se encuentra en un bucle que se repite 10 veces, puede almacenar cuántas veces se ha medido CIR1 en una variable con una instrucción ASIGN como la siguiente: `ASIGN/V1=LEN({CIR1})`

Funciones de matriz

ARRAY

Crear matriz: *MATRIZ*(<EXPRESIÓN1>, <EXPRESIÓN2>, <EXPRESIÓN3>, ...)

Crea un objeto Matriz con los elementos de matriz indicados en los parámetros de la expresión. Los elementos de matriz están numerados, con un índice base de 1.

AVERAGE

Promedio de los elementos de la matriz: *AVERAGE*(<MATRIZ>)

Devuelve el valor promedio de los elementos de la matriz.

EQUAL

Comparación de matrices elemento por elemento: *EQUAL*(<MATRIZ>, <MATRIZ>)

Compara las dos matrices, elemento por elemento, para determinar si tienen los mismos elementos. Si las dos matrices no son del mismo tamaño o si algún elemento de la matriz no coincide con el elemento correspondiente en la otra, la función devuelve el valor de 0. De lo contrario, la función devuelve 1.

LEN

Recuento de los elementos de la matriz: *LEN*(<MATRIZ>)

Devuelve el número de elementos de la matriz.

MAX

Elemento mayor de la matriz: *MAX*(<MATRIZ>)

Devuelve el elemento mayor de la matriz. Los elementos de la matriz se comparan de forma numérica o alfabética.

MIN

Elemento menor de la matriz: *MIN*(<MATRIZ>)

Devuelve el elemento menor de la matriz. Los elementos de la matriz se comparan de forma numérica o alfabética.

SUM

Suma de los elementos de la matriz: *SUM(<MATRIZ>)*

Devuelve la suma de los elementos de la matriz.

Otras funciones

ARCSEGMENTENDINDEX

Esta función devuelve el número de índice del punto final de un segmento de arco especificado de un escaneado: *ARCSEGMENTENDINDEX(<ID>, <índice>, <tol1>, <tol2>)*

<ID>: El primer parámetro es un valor de cadena de la ID del escaneado en la que esta función extrae el número de índice del punto final correspondiente al arco. Este parámetro puede ser la ID entre comillas o cualquier expresión que, cuando se fuerza a adquirir el tipo cadena, se convierte en la ID de un escaneado.

<índice>: El segundo parámetro es el número de índice correspondiente al arco del cual desea obtener el número del punto final. Se trata de un valor con base uno. Por ejemplo, el número de índice del arco sería 3 si desea el número del punto final correspondiente al tercer arco del escaneado.

<tol1>: El tercer parámetro es la tolerancia de elemento general. Es un error de forma máximo utilizado para dividir el escaneado en líneas y arcos.

<tol2>: El cuarto parámetro es la tolerancia de precisión. Generalmente, esta tolerancia más estricta se utiliza para proyectar puntos desde cualquiera de los extremos del elemento hasta que el error de forma del segmento esté dentro de la tolerancia.

Una vez que tenga los índices inicial y final de un arco, puede utilizar estos puntos en un elemento construido para construir un elemento de arco por separado. Para ver un ejemplo similar consulte el tema "Ejemplo de elemento de línea creado a partir de un segmento de escaneado".

ARCSEGMENTSTARTINDEX

Esta función devuelve el número de índice del punto inicial de un segmento de arco especificado de un escaneado: *ARCSEGMENTSTARTINDEX(<ID>, <índice>, <tol1>, <tol2>)*.

<ID>: El primer parámetro es un valor de cadena de la ID del escaneado en la que esta función extrae el número de índice del punto inicial correspondiente al arco. Este parámetro puede ser la ID entre comillas o cualquier expresión que, cuando se fuerza a adquirir el tipo cadena, se convierte en la ID de un escaneado.

<índice>: El segundo parámetro es el número de índice correspondiente al arco del cual desea obtener el número del punto inicial. Se trata de un valor con base uno. Por

ejemplo, el número de índice del arco sería 3 si desea el número del punto inicial correspondiente al tercer arco del escaneado.

<tol1> : El tercer parámetro es la tolerancia de elemento general. Es un error de forma máximo utilizado para dividir el escaneado en líneas y arcos.

<tol2>: El cuarto parámetro es la tolerancia de precisión. Generalmente, esta tolerancia más estricta se utiliza para proyectar puntos desde cualquiera de los extremos del elemento hasta que el error de forma del segmento esté dentro de la tolerancia.

Hay dos parámetros adicionales, que controlan si un segmento de arco identificado en un escaneado es aceptable. Solo se pueden cambiar con el Editor de la configuración de PC-DMIS. Los segmentos de arco con un radio inferior al valor de la entrada

`MinimumArcSegmentRadiusInMM` se rechazan. El valor por omisión de este parámetro es 2 mm. Del mismo modo, los segmentos de arco con un radio superior al valor de la entrada `MaximumArcSegmentRadiusInMM` se rechazan. El valor por omisión de este parámetro es 2000 mm (no debería ser necesario cambiar este valor). Una vez que tenga los índices inicial y final de un arco, puede utilizar estos puntos en un elemento construido para construir un elemento de arco por separado. Para ver un ejemplo similar consulte el tema "Ejemplo de elemento de línea creado a partir de un segmento de escaneado".

EOF y EOL

Para obtener información acerca de estas funciones, consulte el tema "Comprobar si se ha llegado al final de un archivo o de una línea" en el capítulo "Usar entrada/salida de archivos".

FUNCTION

Crea una función: `FUNCTION((<PARÁM1>, <PARÁM2>...), <EXPRESIÓN>)`

Crea una función que toma el número de parámetros indicado en la lista de parámetros e inserta estos parámetros en la expresión.

- La lista de parámetros es el primer elemento cuando se utiliza la palabra clave `FUNCTION`.
- Esta lista contiene los nombres de los parámetros, separados por comas.
- La lista de parámetros aparece entre paréntesis.
- El segundo elemento es la expresión.
- La expresión contiene los nombres de los parámetros que deben ser sustituidos cuando se invoca la función.

Para ver un ejemplo consulte el tema "Ejemplo de una función genérica".

GETROTABDATA

Esta función devuelve los valores de vector, posición angular y centro correspondientes a la mesa giratoria especificada.

GETROTABDATA(<PARÁMETRO>[,<MESA>])

La función devuelve los valores correspondientes a las configuraciones siguientes:

- Mesa giratoria individual
- Dos mesas giratorias (independientes)
- Mesas giratorias apiladas

Los datos que la función devuelve coinciden con los datos del cuadro de diálogo **Configurar mesa giratoria (Edición | Preferencias | Configuración de mesa giratoria)**. Para obtener más información acerca de este cuadro de diálogo, consulte el tema "Definir mesa giratoria".

CENTRO

- "CENTER": Devuelve el centro XYZ actual de la mesa giratoria.
- "CENTER","V": Devuelve el centro XYZ actual de la mesa giratoria V en una configuración de dos mesas o de mesas apiladas.
- "CENTER","W": Devuelve el centro XYZ actual de la mesa giratoria W en una configuración de dos mesas o de mesas apiladas.

Ejemplos:

ASIGN/V1=GETROTABDATA ("CENTER")	V1 recibe el valor del centro XYZ actual de la mesa giratoria.
ASIGN/V1=GETROTABDATA ("CENTER", "V")	V1 recibe el valor del centro XYZ actual de la mesa giratoria V.
ASIGN/V1=GETROTABDATA ("CENTER", "W")	V1 recibe el valor del centro XYZ actual de la mesa giratoria W.

POSICIÓN ANGULAR

- "ANGLE": Devuelve la posición angular actual de la mesa giratoria.
- "ANGLE","V": Devuelve la posición angular actual de la mesa giratoria V en una configuración de dos mesas o de mesas apiladas.

- "ANGLE","W": Devuelve la posición angular actual de la mesa giratoria W en una configuración de dos mesas o de mesas apiladas.

Ejemplos:

<code>ASIGN/V2=GETROTABDATA ("ANGLE")</code>	V2 recibe la posición angular actual de la mesa giratoria.
<code>ASIGN/V2=GETROTABDATA ("ANGLE", "V")</code>	V2 recibe la posición angular actual de la mesa giratoria V.
<code>ASIGN/V2=GETROTABDATA ("ANGLE", "W")</code>	V2 recibe la posición angular actual de la mesa giratoria W.

VECTOR

- "VECTOR": Devuelve el vector IJK actual de la mesa giratoria.
- "VECTOR","V": Devuelve el vector IJK actual de la mesa giratoria V en una configuración de dos mesas o de mesas apiladas.
- "VECTOR","W": Devuelve el vector IJK actual de la mesa giratoria W en una configuración de dos mesas o de mesas apiladas.

Ejemplos:

<code>ASIGN/V3=GETROTABDATA ("VECTOR")</code>	V3 recibe el valor del vector IJK actual de la mesa giratoria.
<code>ASIGN/V3=GETROTABDATA ("VECTOR", "V")</code>	V3 recibe el valor del vector IJK actual de la mesa giratoria V.
<code>ASIGN/V3=GETROTABDATA ("VECTOR", "W")</code>	V3 recibe el valor del vector IJK actual de la mesa giratoria W.



El argumento [MESA] es opcional. Si no especifica la mesa V o W, PC-DMIS realiza una de las acciones siguientes:

- Si utiliza una configuración de una sola mesa o de mesas apiladas, devuelve los valores de la mesa giratoria W.
- Si utiliza una configuración de dos mesas, devuelve los valores de la mesa giratoria que está activada en la barra de herramientas **Mesa giratoria activa**. Para obtener más información acerca de la barra de herramientas, consulte el tema "Barra de herramientas de mesa giratoria activa".

PC-DMIS tiene dos definiciones de mesas internas para dar cabida a las configuraciones de dos mesas y de mesas apiladas. En el caso de una configuración de una sola mesa, la definición de la segunda mesa no se utiliza. Puesto que existe internamente, no se producirá un error si especifica la mesa V en una configuración de una sola mesa; sin embargo, esto no es recomendable. Los valores que la función devuelve por lo general no son útiles, ya que la mesa en realidad no existe.

IF

Evaluación de expresión condicional: IF(<EXPRESIÓN1>, <EXPRESIÓN2>, <EXPRESIÓN3>)

Si Expresión1 da como resultado VERDADERA (no igual a cero), esta función devuelve el valor de Expresión2; en caso contrario, la función devuelve el valor de Expresión3.

ISIOCHANNELSET

Esta expresión admite dos parámetros. El primer parámetro indica el canal E/S que se comprobará (el rango de números disponible se basa en la máquina utilizada). El segundo parámetro determina si el software enviará una consulta a la máquina de brazo 1 o de brazo 2. Si el segundo parámetro se establece en 1 (uno), se enviará una consulta al controlador del brazo 2. Si no se introduce el segundo parámetro (o si está establecido en cero), el canal E/S enviará la consulta al controlador del brazo 1. El controlador de brazo 1 es la única opción disponible si no se trabaja en modo de varios brazos.



Si se especifican valores no válidos para los tipos de datos de la sonda, la ID de la punta, el nombre del archivo de la sonda o el número de canal, la expresión arroja 0.

Ejemplo:

<code>ASIGN/V4=ISIOCHANNELSET(3,0)</code>	V4 será igual a 1 (dará verdadero) cuando el canal esté establecido; de lo contrario, arrojará 0 (dará falso).
---	--

LINESEGMENTENDINDEX

Esta función devuelve el número de índice del punto final de un segmento de línea especificado de un escaneado: `LINESEGMENTENDINDEX(<ID>, <índice>, <tol1>, <tol2>)`.

<ID>: El primer parámetro es un valor de cadena de la ID del escaneado en la que esta función extrae el número de índice del punto final correspondiente al segmento de línea. Este parámetro puede ser la ID entre comillas o cualquier expresión que, cuando se fuerza a adquirir el tipo cadena, se convierte en la ID de un escaneado.

<índice>: El segundo parámetro es el número de índice correspondiente al segmento de línea del cual desea obtener el número del punto final. Se trata de un valor con base uno. Por ejemplo, el número de índice del segmento de línea sería 3 si desea el número del punto final correspondiente a la tercera línea del escaneado.

<tol1>: El tercer parámetro es la tolerancia de elemento general. Es un error de forma máximo utilizado para dividir el escaneado en líneas y arcos.

<tol2>: El cuarto parámetro es la tolerancia de precisión. Generalmente, esta tolerancia más estricta se utiliza para proyectar puntos desde cualquiera de los extremos del elemento hasta que el error de forma del segmento esté dentro de la tolerancia.

Una vez que tenga los índices inicial y final de un segmento de línea, puede utilizar estos puntos en un elemento construido para construir un elemento de línea por separado. Para ver un ejemplo, consulte el tema "Ejemplo de elemento de línea creado a partir de un segmento de escaneado".

LINESEGMENTSTARTINDEX

Devuelve el número de índice del punto inicial de un segmento de línea especificado de un escaneado: `ARCSEGMENTSTARTINDEX(<ID>, <índice>, <tol1>, <tol2>)`.

<ID>: El primer parámetro es un valor de cadena de la ID del escaneado en la que esta función extrae el número de índice del punto inicial correspondiente al segmento de línea. Esto puede ser la ID entre comillas o cualquier expresión que, cuando se fuerza a adquirir el tipo cadena, se convierte en la ID de un escaneado.

<índice>: El segundo parámetro es el número de índice correspondiente al segmento de línea del cual desea obtener el número del punto inicial. Se trata de un valor con base uno. Por ejemplo, el número de índice del segmento de línea sería 3 si desea el número del punto inicial correspondiente a la tercera línea del escaneado.

<tol1>: El tercer parámetro es la tolerancia de elemento general. Es un error de forma máximo utilizado para dividir el escaneado en líneas y arcos.

<tol2>: El cuarto parámetro es la tolerancia de precisión. Generalmente, esta tolerancia más estricta se utiliza para proyectar puntos desde cualquiera de los extremos del elemento hasta que el error de forma del segmento esté dentro de la tolerancia. Hay un parámetro adicional que controla si un segmento de línea identificado en un escaneado es aceptable. Solo se puede cambiar con el Editor de la configuración de PC-DMIS. Los segmentos de línea con una longitud inferior al valor de la entrada `MinimumLineSegmentLengthInMM` se rechazan. El valor por omisión de este parámetro es 2 mm.

Una vez que tenga los índices inicial y final de un segmento de línea, puede utilizar estos puntos en un elemento construido para construir un elemento de línea por separado. Consulte el tema "Ejemplo de elemento de línea creado a partir de un segmento de escaneado" para ver un ejemplo.

PROBEDATA

Esta función devuelve datos acerca de la sonda actual o la especificada:

`PROBEDATA(<OPTPROBEDATATYPE>, <OPTTIPID>, <OPTPROBEFILENAME>)`.

Esta función utiliza un máximo de tres parámetros opcionales. Solo es necesario colocar comas entre los parámetros si utiliza más de uno. No es necesario utilizar comas entre los parámetros vacíos. Por ejemplo, para obtener el diámetro de la sonda actual utilizaría `ASIGN/V1=PROBEDATA ("DIAM")`.

TipoDatosSondaOpcional: Parámetro opcional que especifica qué datos de sonda debe devolver la expresión. Si no se especifica este parámetro, la función devuelve la ID de la punta actual. Este parámetro es una cadena. Cualquier expresión que se evalúe como una cadena válida se puede incluir en la posición de la primera expresión. Las expresiones de cadena válidas (sin distinguir entre mayúsculas y minúsculas) para el primer parámetro son las siguientes. Se trata de expresiones de cadena y deben aparecer entre comillas:

"A": Ángulo A de la punta. Devuelve un valor doble.

"B": Ángulo B de la punta. Devuelve un valor doble.

"C": "C": Ángulo de un cabezal de sonda CW43 Light. Devuelve un valor entero.

"Date": Fecha de la última ocasión en que se cualificó la punta. Devuelve un valor de cadena.

"Diam" o "Diameter": Diámetro medido de la punta. Es necesario incluir las cuatro primeras letras "Diam", pero también se pueden incluir más letras para poner el nombre completo, "Diameter". Devuelve un valor doble.

"ID": ID de punta. Parámetro por omisión. Devuelve un valor de cadena.

"Offset": Offset X,Y,Z para la punta medida. Devuelve un valor de punto.

"PrbRdv": Desviación radial de la sonda. Devuelve un valor doble.

"Rotation": Rotación alrededor del vector de punta en radianes. Devuelve un valor doble.

"Standarddeviation" - Desviación estándar de la sonda. Devuelve un valor doble.

"Thick" o "Thickness": Espesor medido de la punta. Es necesario incluir las cinco primeras letras "Thick", pero también se pueden incluir más letras para poner el nombre completo, "Thickness". Devuelve un valor doble.

"Time": Hora de la última ocasión en que se calificó la punta. Devuelve un valor de cadena.

"Vector": Vector de la punta. Devuelve un valor de punto.



Puede poner una **"T"** delante de **"Diameter"**, **"Offset"** o **"Thickness"** para que se devuelva la información teórica (por ejemplo, **TDIAMETER**, **TOFFSET** y **TTHICKNESS**).

OPTTIPID: Este parámetro opcional especifica qué punta se debe utilizar para obtener los datos de la sonda que se especifican en la primera expresión. Si no se especifica este parámetro, se utiliza la punta de la sonda actual. Este parámetro debe ser una cadena.

OPTPROBEFILENAME: Este parámetro opcional especifica el nombre del archivo de sonda que se debe utilizar al obtener los datos de la sonda. Si no se especifica este parámetro, se utiliza el archivo de la sonda actual.

Ejemplos:

<code>ASIGN/V1=PROBEDATA ()</code>	V1 se establece en el valor de la ID de la punta actual (por ejemplo, "T1A0B0")
<code>ASIGN/V2=PROBEDATA ("TOFFSET", "T1A45B0")</code>	V2 se establece en el valor teórico del offset de sonda de la punta T1A45B0
<code>ASIGN/V3=PROBEDATA ("Date", "T1A90B90", "MISONDA")</code>	V3 toma el valor de una cadena que representa la fecha de la última ocasión en que se calibró la punta T1A90B90 del archivo de sonda MISONDA.

QUALTOOLDATA

Esta función devuelve datos acerca de la herramienta de calibración actual o especificada. La sintaxis es la siguiente:

QUALTOOLDATA(<INFOHERRAM>, <IDHERRAM>, <NUMCARA>)

Esta función utiliza un máximo de tres parámetros. Necesita al menos un parámetro para devolver datos:

El primer parámetro, <INFOHERRAM>, es una *cadena* que indica el tipo de información que se devolverá acerca de la herramienta de calibración. Si no se utiliza este parámetro, esta función devuelve el nombre de la herramienta de calibración actual o especificada.

- **"CTE"** o **"COEFFICIENTOFTHERMALEXPANSION"**: Cualquiera de estas cadenas devuelve el coeficiente de expansión térmica como valor doble.
- **"DIAM"**: Esta cadena devuelve el diámetro de la herramienta como valor doble.
- **"ID"**: Esta cadena devuelve el nombre de la herramienta como valor de tipo cadena.
- **"LENGTH"**: Esta cadena funciona igual que "DIAM". También devuelve el diámetro de la herramienta como valor doble.
- **"OVERRIDEIJK"**: Esta cadena devuelve el vector IJK que sobrescribe el vector de búsqueda como valor de punto.
- **"POLYDIAM"**: Esta cadena devuelve el diámetro de la cara poliédrica especificada como valor doble.
- **"POLYIJK"**: Esta cadena devuelve el vector IJK de la cara poliédrica especificada como valor de punto.
- **"POLYXYZ"**: Esta cadena devuelve el centro XYZ de la cara poliédrica especificada como valor de punto.
- **"SHANKIJK"**: Esta cadena devuelve el vector IJK del vástago como valor de punto.
- **"TYPE"**: Esta cadena devuelve el tipo de herramienta como valor entero (0 para esfera, 1 para esfera de brazo 2, 2 para poliedro, 3 para poliedro de brazo 2).
- **"ANCHURA"**: Este parámetro ya no se utiliza.
- **"XYZ"**: Esta cadena devuelve la ubicación XYZ de la herramienta como valor de punto.

El segundo parámetro, <IDHERRAM>, es una *cadena* que especifica el nombre de la herramienta de calibración sobre la que desea recibir información. Si no se utiliza este parámetro, PC-DMIS presupone que desea información de la herramienta de calibración actual. La cadena no distingue entre mayúsculas y minúsculas.

El tercer parámetro, <NUMCARA>, solo es necesario cuando se trabaja con una herramienta de calibración poliédrica y únicamente cuando el primer parámetro es "POLIXYZ", "POLIJK" o "POLIDIAM". Es un valor entero que especifica la cara de la herramienta poliédrica que se utilizará para obtener datos.



La herramienta de calibración no tiene un valor global que se aplique automáticamente a todas las sondas en una rutina de medición. Cuando se calibra una sonda por primera vez, debe seleccionar la herramienta de calibración que se utilizará. PC-DMIS guarda la información sobre la herramienta de calibración para cada sonda. Cuando vuelva a calibrar la misma sonda, PC-DMIS utilizará la misma herramienta de calibración.

Ejemplos:

<code>ASIGN/VDIAM=QUALTOOLDATA ("DIAM", "SPHERE_1_IN")</code>	Asigna a la variable VDIAM el diámetro de la herramienta SPHERE_1_IN.
<code>ASIGN/VID=QUALTOOLDATA ("ID")</code>	Asigna a la variable VID el nombre de la herramienta actual.
<code>ASIGN/VTYPE=QUALTOOLDATA ("TYPE")</code>	Asigna a la variable VTYPE el tipo de la herramienta actual.
<code>ASIGN/VPOLYDIAM=QUALTOOLDATA ("POLYDIAM", "POLYTEST", 3)</code>	Asigna a la variable VPOLYDIAM el diámetro de la cara 3 de la herramienta poliédrica POLYTEST.

SETROTABDATA

Esta función asigna el nuevo valor de entrada al centro o al vector:

SETROTABDATA(<PARÁMETRO>,<Valor_nuevo>[,<MESA>])

Esta función se puede utilizar en las configuraciones siguientes:

- Mesa giratoria individual
- Dos mesas giratorias (independientes)
- Mesas giratorias apiladas

CENTRO

- "CENTER",<NewValue>: Asigna el valor nuevo al centro XYZ de la mesa giratoria.
- "CENTER",<NewValue>,"V": Asigna el valor nuevo al centro XYZ de la mesa giratoria V en una configuración de dos mesas o de mesas apiladas.
- "CENTER",<NewValue>,"W": Asigna el valor nuevo al centro XYZ de la mesa giratoria W en una configuración de dos mesas o de mesas apiladas.

Ejemplos:

<code>ASIGN/V1=SETROTABDATA ("CENTER",NewValue)</code>	V1 es un código de retorno (1=éxito, 0=error).
<code>ASIGN/V1=SETROTABDATA ("CENTER",NewValue, "V")</code>	V1 es un código de retorno (1=éxito, 0=error).
<code>ASIGN/V1=SETROTABDATA ("CENTER",NewValue, "W")</code>	V1 es un código de retorno (1=éxito, 0=error).

VECTOR

- "VECTOR",<NewValue>: Asigna el valor nuevo al vector IJK de la mesa giratoria.
- "VECTOR",<NewValue>,"V": Asigna el valor nuevo al vector IJK de la mesa giratoria V en una configuración de dos mesas o de mesas apiladas.
- "VECTOR",<NewValue>,"W": Asigna el valor nuevo al vector IJK de la mesa giratoria W en una configuración de dos mesas o de mesas apiladas.

Ejemplos:

<code>ASIGN/V2=SETROTABDATA ("VECTOR",NewValue)</code>	V2 es un código de retorno (1=éxito, 0=error).
<code>ASIGN/V2=SETROTABDATA ("VECTOR",NewValue,"V")</code>	V2 es un código de retorno (1=éxito, 0=error).
<code>ASIGN/V2=SETROTABDATA ("VECTOR",NewValue,"W")</code>	V2 es un código de retorno (1=éxito, 0=error).



El argumento [MESA] es opcional. Si no especifica la mesa V o W, PC-DMIS realiza una de las acciones siguientes:

- Si utiliza una configuración de una sola mesa o de mesas apiladas, establece el valor nuevo para la mesa giratoria W.
- Si utiliza una configuración de dos mesas, establece el valor nuevo para la mesa giratoria que está activada en la barra de herramientas **Mesa giratoria activa**. Para obtener más información acerca de la barra de herramientas, consulte el tema "Barra de herramientas de mesa giratoria activa".

PC-DMIS tiene dos definiciones de mesas internas para dar cabida a las configuraciones de dos mesas y de mesas apiladas. En el caso de una configuración de una sola mesa, la definición de la segunda mesa no se utiliza. Puesto que existe internamente, no se producirá un error si especifica la mesa V en una configuración de una sola mesa; sin embargo, esto no es recomendable. Los valores que la función establece por lo general no son útiles, ya que la mesa en realidad no existe.

TUTORELEMENT

Esta función toma un parámetro, ya sea un número o una cadena (una cadena sería la ID de un elemento).

TUTORELEMENT(<PARÁMETRO>)

Usar expresiones y variables

Esta función trabaja con el tipo de variable *Estructuras*. Para obtener más información sobre estructuras y subelementos, consulte "Estructuras".

Ejemplos:

<code>ASIGN/E=TUTORELEMENT (1)</code>	Crea una única estructura de TutorElement
<code>ASIGN/WM=TUTORELEMENT (n)</code>	Para cualquier número mayor que 1, crea una matriz de <code>n</code> estructuras de TutorElement.
<code>ASIGN/CIR1E=TUTORELEMENT ("CIR1")</code>	Copia datos del elemento CIR1 en las estructuras de TutorElement.

La estructura de TutorElement dispone actualmente de los siguientes subelementos:

Subelemento	Descripción
ID	Cadena de la ID del elemento
TIPO	ENTERO (TIPO F)
X, Y, Z	Valores de coordenada X, Y y Z
PR	Radio polar
PA	Ángulo polar
CX	I
CY	J
CZ	K
DM	Diámetro 1
DM2	Diámetro 2
DS	Distancia desde el origen
A	Ángulo
AXY	Ángulo en el plano XY
AYZ	Ángulo en el plano YZ

AZX	Ángulo en el plano ZX
F	Error de forma
SDEV	Desviación estándar
TP	Posición

Ejemplos de funciones

A continuación se proporcionan varios ejemplos de funciones que pueden servirle de ayuda para crear y utilizar sus propias funciones:

- Ejemplo de una función genérica
- Ejemplo de funciones pasadas como variables
- Ejemplo de una función con múltiples parámetros
- Ejemplo de funciones que crean otras funciones
- Ejemplo de funciones como miembros de una matriz
- Ejemplo de funciones definidas para recurrencia

Ejemplo de una función genérica

```
ASIGN/MIFUNC=FUNCTION ( (X,Y,Z) ,X*3+Y*2+Z)
```

Crea una función definida por el usuario y la asigna a la variable MIFUNC. La función toma tres parámetros: X, Y y Z.

- X se multiplica por 3.
- Y se multiplica por 2.
- Z contiene el valor pasado.

El total de $X + Y + Z$ es lo que se devuelve cuando se pasan valores a la función, como se muestra aquí:

```
ASIGN/V1=MIFUNC (7,2,5)
```

Asigna a V1 el valor 30 evaluando los parámetros pasados en la función MIFUNC(7,2,5).

Usar expresiones y variables

- 7 es el parámetro que se utiliza para la sustitución cuando aparece X en la sección de expresión de la definición de la función. Consecuentemente, X^3 se convierte en 7^3 , es decir, 21.
- Donde aparece Y se sustituye por 2. Por lo tanto, Y^2 se convierte en 2^2 , es decir, 4.
- Donde aparece Z se sustituye por 5.

A continuación se suman todos los valores ($21 + 4 + 5$) y se pasan a V1.

Ejemplo de funciones pasadas como variables

Las funciones pueden ser pasadas como variables. El ejemplo siguiente se basa en el tema Ejemplo de una función genérica anterior:

```
ASIGN/NEWFUNC=MIFUNC
```

Establece la variable NEWFUNC para que tenga la misma función que MIFUNC.

```
ASIGN/V3=NEWFUNC (12, 2, 3)
```

Asigna a V3 el valor 43, procedente de las expresiones evaluadas en la función ($36 + 4 + 3$).

Ejemplo de una función con múltiples parámetros

Un función puede tener múltiples parámetros:

```
ASIGN/ADDANDDOUBLE=FUNCIÓN ( (A, B) , 2 * (A+B) )
```

Crea una función y la asigna a la variable ADDANDDOUBLE. La función toma dos parámetros, los suma y multiplica el resultado por 2.

```
ASIGN/V2=ADDANDDOUBLE (4, 5)
```

Asigna a V2 el valor de 18. Los parámetros 4 y 5 son sustituidos en la sección expresión de la función, convirtiéndose en $2*(4+5)$.

Ejemplo de funciones que crean otras funciones

Las funciones pueden crear otras funciones.

```
ASIGN/COMPONER=FUNCTION ( (F, G) , FUNCTION ( (X) , G (F (X)) ) )
```

Convierte COMPONER en una función que utiliza dos funciones como parámetros y crea una nueva utilizando las dos.

`ASIGN/AÑADIR2=FUNCTION ((X) , X+2)`

Convierte AÑADIR2 en una función que suma dos al parámetro introducido.

`ASIGN/AÑADIR3=FUNCTION ((X) , X+3)`

Convierte AÑADIR3 en una función que suma tres al parámetro introducido.

`ASIGN/AÑADIR5=COMPOSE (AÑADIR2 , AÑADIR3)`

Convierte AÑADIR5 en una función compuesta de las funciones AÑADIR2 y AÑADIR3.

`ASIGN/V5=ADD5 (3)`

Asigna a V5 el valor V8.

Ejemplo de funciones como miembros de una matriz

Las funciones pueden ser miembros de una matriz.

`ASIGN/UNAMATRIZ=ARRAY (3, FACTORIAL, "Hola a todos", AÑADIR5)`

Asigna UNAMATRIZ a una matriz de 4 elementos: un número (3), una función (FACTORIAL), una cadena ("Hola a todos") y una función (Añadir5).

`ASIGN/V6=UNAMATRIZ [2] (4)`

El segundo elemento de UNAMATRIZ es la función FACTORIAL. El parámetro 4 se pasa a esta función, y el resultado de 24 se asigna a V6.

`ASIGN/V7=UNAMATRIZ [2] (UNAMATRIZ [4] (UNAMATRIZ [1]))`

De dentro afuera: el primer elemento de UNAMATRIZ (3) se pasa a la función del cuarto elemento de la matriz (Añadir5). El resultado, 8, se pasa a la función del segundo elemento de la matriz (FACTORIAL) y se asigna a V7. V7 recibe un valor de 40320.

Ejemplo de funciones definidas para recurrencia

Las funciones pueden definirse para recurrencia; es decir, que pueden ser definidas para que se invoquen a sí mismas.

`ASIGN/FACTORIAL=FUNCTION ((X) , IF (X<=1, 1, X*FACTORIAL (X-1)))`

Crea una función factorial que contiene un parámetro. Si el parámetro es menor o igual a 1, arrojará 1, de lo contrario arrojará X multiplicado por el FACTORIAL de X-1.

`ASIGN/V4=FACTORIAL (5)`

Asigna a V4 el valor de 120 (5*4*3*2*1).

Ejemplo de elemento Línea creado a partir de un escaneado de segmento

En este tema se muestra con un ejemplo cómo se utiliza el lenguaje de expresiones de PC-DMIS, y en especial las funciones de segmentos de línea, para exportar los números de los puntos inicial y final de los segmentos de línea dentro de un escaneado y luego crear un elemento de línea propio empleando los puntos extraídos dentro de un elemento construido. Puede servirse de los mismos principios que se tratan en este ejemplo para crear también un segmento de arco a partir de un escaneado.

Supongamos que su rutina de medición tiene un elemento de escaneado denominado SCN1 que tiene este aspecto:



```

SCN1=ELEM/ESCANEADO,LÍNEAABIERTA, MOSTRAR CONT=NO,
MOSTRAR TODOS PARÁMS=SÍ

MODULO EJEC=REAPRENDER, MODULO NOMS=BUSCAR
NOMINALES,PLANASEG=NO,UN PUNTO=NO,ESPES=0

BUSCARNOMS=5,SOLOSELECCIONADO=NO,USAMEJAJ=NO,

SONDACOMP=SÍ,MOVIMIENTO EVITACIÓN=NO,DISTANCIA=0,
Compensación CAD=NO

DIR1=VARIABLE,

TIPOCONT=VECTOR

VECCINIC=0,-1,0

VECDIR=1,0,0

VECCORTE=0,0,1

VECFINAL=0,-1,0

VECPLANO=1,0,0

PUNTO1=100,0,-5

PUNTO2=70,0,-5

MED/ESCANEADO

ESCANEADO BASE/LÍNEA,MOSTRARCONT=NO,MOSTRAR TODOS
PARÁMS=SÍ

<100,0,-5>,<70,0,-5>,VecCorte=0,0,1,VecDir=1,0,0

VecInic=0,-1,0,VecFinal=0,-1,0,ESPES=0

FILTRO/FILTRO NULO,

MODULO EJEC=REAPRENDER

LÍM/PLANO,<70,0,-5>,VecPlano=-1,0,0,Cruces=2

```


Usar expresiones y variables

```
TIPOCONT/VECTOR

MODULO NOMS=BUSCARNOMS,5

TERMINAR ESCANEADO

TERMINARMED/
```

Para crear una línea a partir de este escaneado tendrá que utilizar las funciones LINESEGMENTSTARTINDEX y LINESEGMENTENDINDEX a fin de extraer los datos, de este modo:



```
ASIGN/ÍNDICEINICIOLÍNEA=LINESEGMENTSTARTINDEX("SCN1",
1,0.4,0.1)

ASIGN/ÍNDICEFINLÍNEA=LINESEGMENTENDINDEX("SCN1",1,0.4
,0.1)
```

Esto indica a PC-DMIS que vaya al escaneado denominado "SCN1" y que de su primer segmento de línea extraiga los valores de los índices de inicio y fin que entren dentro de las tolerancias definidas. Luego asigna esos valores de índice a las variables denominadas ÍNDICEINICIOLÍNEA y ÍNDICEFINLÍNEA.

Una vez que haya asignado a las variables los valores de índice inicial y final del segmento de línea, puede utilizar esas variables dentro de una línea construida, de este modo:



```
LIN4=ELEM/LÍNEA,RECT,NODELIM

TEO/100.225,0,-5.011,1,0,0

REAL/100.225,-0.005,-5.011,1,-0.0000388,0

CONST/LÍNEA,MEJAJ,2D,SCN1.HIT[LINESTARTINDEX..LINEEND
INDEX],,

ELIMINACIÓN_OUTLIERS/DES,3

FILTRO/DES, LONGITUDONDA=0
```

Observe que en el código resaltado del elemento de línea anterior PC-DMIS utiliza los números de inicio y fin que extrajo del escaneado para crear el elemento:
SCN1.HIT[LINESTARTINDEX..LINEENDINDEX]

Coerción de operandos

Los operadores de coerción permiten forzar a los operandos para que sean de otro tipo:

Coerción de enteros

INT(<Expresión>): Fuerza el valor de una expresión para que sea entero.

INT(4)	Da como resultado 4
INT(4.5)	Da como resultado 4
INT("Hola a todos")	Da como resultado 0
INT("2")	Da como resultado 2
INT("2.2")	Arroja 2
INT("3 cerditos")	Da como resultado 3
INT("Los 3 cerditos")	Da como resultado 0
INT("3, 4, 5")	Da como resultado 3
INT(MPOINT(0, 0, 1))	Da como resultado, la distancia entre el punto y el origen, en este caso 1.
INT(MPOINT(3, 4, 5))	La distancia da como resultado 7,0711. Esta expresión da como resultado 7.

Coerción de dobles

DOUBLE(<Expresión>): Fuerza el valor de una expresión para que sea doble.

DOUBLE (4)	Da como resultado 4.0
DOUBLE (4.5)	Da como resultado 4.5
DOUBLE ("Cadena A")	Da como resultado 0.0

Usar expresiones y variables

<code>DOUBLE ("3.5")</code>	Da como resultado 3.5
<code>DOUBLE ("3,5 pulgadas")</code>	Da como resultado 3.5
<code>DOUBLE ("El círculo tiene un diámetro de 3,5 pulgadas")</code>	Da como resultado 0.0
<code>DOUBLE (MPOINT (0,0,1))</code>	Da como resultado 1,0
<code>DOUBLE (MPOINT (3,4,5))</code>	Da como resultado 7,0711

Coerción de cadenas

STR(<Expresión>): Fuerza el valor de una expresión para que sea una cadena.

<code>STR(4)</code>	Da como resultado "4"
<code>STR(4.5)</code>	Da como resultado "4.5"
<code>STR("Hola a todos")</code>	Da como resultado "Hola a todos"
<code>STR(MPOINT(3,4,5))</code>	Da como resultado "3, 4, 5"

Coerción de puntos

MPOINT(<Expresión1>, <Expresión2>, <Expresión3>): Fuerza los valores de las expresiones para que sean puntos después de forzar a todas las expresiones para que sean de tipo Doble.

<code>MPOINT (1, 1, 1)</code>	Da como resultado el punto 1,0,1.0,1.0
<code>MPOINT (1.1, 1.1, 1.1)</code>	Da como resultado el punto 1,1, 1,1, 1,1
<code>MPOINT ("1", "1", "1")</code>	Da como resultado el punto 1,0,1.0,1.0
<code>MPOINT (3, 4.5, "5.6")</code>	Da como resultado el punto 3.0, 4.5, 5.6
<code>MPOINT (MPOINT (1, 0, 0),</code> <code>MPOINT (0, 1, 0),</code> <code>MPOINT (3, 4, 5))</code>	Da como resultado 1.0, 1.0, 7.0711

Coerción de operandos y expresiones de tipo mixto

El evaluador de expresiones fuerza automáticamente las variables para que sean expresiones mixtas. Si debido a la coerción automática el resultado de una expresión no es el esperado, en algunos casos es posible utilizar los operadores de coerción para obtener el resultado deseado. A continuación se incluyen varios ejemplos de coerciones automáticas en expresiones mixtas.

"CIR" + 1

Da como resultado "CIR1".

"2" + 2

Da como resultado 4.

"El valor de 2+2 es " + 2 + 2

Da como resultado "El valor de 2+2 es 22". Ello se debe a que PC-DMIS evalúa las expresiones de izquierda a derecha, y la parte izquierda de la expresión es una cadena.

"El valor de 2+2 es " + (2 + 2)

Da como resultado "El valor de 2+2 es 4".

LINEA1.XYZ > 2

Da como resultado 1 si la distancia del centroide de LINEA1 desde el origen es mayor que 2.

LINEA1.XYZ > LINEA2.XYZ

Da como resultado 1 si el centroide de LINEA1 está más lejos del origen que el centroide de LINEA2.

Usar expresiones y variables

LINEA1.XYZ = LINEA2.XYZ

Da como resultado 1 si los centroides de LINEA1 y LINEA2 son los mismos (en este caso no se produce coerción).

DOUBLE(LINEA1.XYZ) = DOUBLE(LINEA2.XYZ)

Da como resultado 1 si los centroides están a la misma distancia del origen.

11% 3.1

Da como resultado 2 (% es el operador de módulo designado para funcionar con enteros). Devuelve el resto de una división discreta. $11\%3 = 2$.)

CIRCULO1.HIT[3.2].X

Da como resultado el valor X medido para el tercer contacto de Círculo1. El argumento 3.2 fuerza automáticamente a un entero para que adquiera el valor de 3.

Expresiones de ID

Muchos de los comandos de PC-DMIS utilizan las ID de los elementos como parámetro. Por ejemplo, los elementos construidos emplean las identificaciones para indicar qué elementos se deben utilizar como entradas para el elemento construido. Las expresiones de ID permiten hacer referencia a una instancia específica de un elemento, a un grupo de elementos con nombres parecidos, a una instancia de un elemento incluido en una llamada a subrutina o a un elemento de una rutina de medición externa.

ID de la matriz de elementos

Utilice una ID de matriz de elementos para hacer referencia a una instancia específica de un elemento o a un rango de instancias del elemento. Por ejemplo, si el elemento "Círculo1" estuviera situado en un bucle del comando MIENTRAS que se ejecuta cinco veces, se quedarían cinco instancias del círculo al salir del bucle. Para hacer referencia a una instancia en particular de las cinco instancias de "Círculo1", utilice la sintaxis de matriz de elementos que se describe en "Matrices de elementos:" donde "Círculo1[1]" haría referencia a la primera instancia, "Círculo1[2]" haría referencia a la segunda instancia, etc.

Para hacer referencia a un rango de instancias, utilice la notación de dos puntos seguidos (..). "Círculo1[1..3]" hace referencia a las instancias 1 a 3 de Círculo1. "Círculo1[3..5]" hace referencia a las instancias 3 a 5 de Círculo1. "Círculo1[1..5]" hace referencia a las instancias 1 a 5 de Círculo1. Cuando se hace referencia a un rango de elementos, el conjunto se considera como un conjunto construido y actúa como tal.

Comodines de ID

Utilice comodines de ID para hacer referencia a un conjunto de elementos con nombres parecidos. Los dos caracteres comodín son "*" y "?". Consulte el tema "Seleccionar elementos por comparación de metacaracteres" en el capítulo "Editar la presentación de modelos CAD" para obtener más información.

El asterisco "*" hace referencia a 0 o más instancias de cualquier carácter. Para hacer referencia al conjunto de todos los elementos que comienzan por las letras "CIR", utilice la expresión ID "CIR*". Esta sintaxis crea un conjunto de elementos que incluye todos los elementos cuya ID empieza por "CIR", como "CIRCULO1", "CIRCULO2", "CIRCULO3" o "CIR".



Si CIR3 se ha ejecutado varias veces, solo se utilizará la medición más reciente. Para obtener todas las instancias de las ejecuciones, podría utilizar la siguiente expresión: CIR?[1..3]

El signo de interrogación "?" hace referencia a una única instancia de cualquier carácter.



La expresión de ID "MI????1" crea un conjunto de los elementos que empiezan por "MI" y terminan en "1", como "MICIRC1", "MIRECT1", "MICON1", "MILÍN1" o "MIELM21".

ID de elementos en subrutinas, guiones BASIC o rutinas externas

Es posible encontrar subrutinas en la rutina de medición actual o en una rutina de medición externa. Si la subrutina se encuentra en la misma rutina que la llamada a la subrutina, puede utilizar la sintaxis de ID de matriz de elementos descrita en "Matrices de elementos:" para hacer referencia a instancias individuales de un elemento creadas en la subrutina. Sin embargo, si la subrutina se encuentra en una rutina de medición externa, puede utilizar la siguiente sintaxis para hacer referencia a los elementos creados en la subrutina: "<ID sub llam>:<ID elemento>".

Por ejemplo, si un elemento denominado "F1" se encuentra en una subrutina externa que recibe una llamada a través de un comando Llamar sub con la ID "CS1", podría utilizar la expresión de ID "CS1:F1" para hacer referencia a dicho elemento.



Este ejemplo simplemente ilustra cómo se utiliza la sintaxis CS1.F1 y no está diseñado para su utilización.

Rutina 1: PLUS1.PRG

```
SUBROUTINA/PLUS1, A1 = 0, A2 = 0, A3 = 0

F1=ELEM/PUNTO,RECT

TEO /A1+1,A2+1,A3+1,0,0,1

REAL/3,1,1,0,0,1

MED/PUNTO,1

CONT /BASE,A1+1,A2+1,A3+1,0,0,1,0,0,0

TERMINARMED/

TERMINAR SUB/
```

Rutina 2: TEST.PRG

```
CS1 =LLAMAR_SUB/PLUS1,D:\V30\WINDEBUEG\PLUS1.PRG: 3,3,3,,

DIM D1= UBICACIÓN DEL PUNTO CS1:F1 UNIDADES=PULG,$

GRÁFICO=DES TEXTO=DES MULT=10.00 SALIDA=AMBOS

EJE NOMINAL +TOL -TOL MED MÁX MÍN DESV FUERATOL

X 3.0000 0.0000 0.0000 3.0000 3.0000 3.0000 0.0000 0.0000

----#----

FIN DE DIMENSIÓN D1
```

Los guiones BASIC crean y suprimen objetos dinámicamente. Utilice la sintaxis "<ID guión BASIC>:<ID elemento>" para hacer referencia a un elemento creado por un guión BASIC. Por ejemplo, si un guión BASIC con ID "BS1" crea un elemento con ID "F2", utilice la expresión de ID "BS1:F2" para hacer referencia a dicho elemento.

Puede utilizar el comando ADJUNT para anexar rutinas externas a PC-DMIS. Para hacer referencia a los elementos de la rutina anexada, utilice la sintaxis siguiente: "<ID rutina anexada>:<ID elemento>". Si desea hacer referencia al elemento "F3" de la rutina de medición anexada con la ID "ENGRANAJE1", utilice la expresión "ENGRANAJE1:F3". Para obtener más información, consulte "Anexar una rutina de medición externa" en el capítulo "Añadir elementos externos".

Combinaciones de expresiones de ID

Puede combinar expresiones de ID de matriz, expresiones de ID con comodines y expresiones de subrutinas externas, guiones BASIC e ID de rutinas de medición externas. Por ejemplo, para hacer referencia a la tercera instancia de todos los elementos que empiezan por las letras "REC" de una rutina de medición externa anexada con la ID "PERNO", utilice la expresión de ID "PERNO:REC*[3]".

También puede utilizar expresiones de ID en expresiones regulares. Por ejemplo, el centroide medido para el conjunto de elementos antes mencionado se podría asignar a una variable por medio de la siguiente expresión:

```
ASIGN/V1=PERNO:REC*[3].XYZ
```

También es posible utilizar expresiones de ID en expresiones regulares. Por ejemplo, el centroide medido para el conjunto de elementos antes mencionado se podría asignar a una variable por medio de la siguiente expresión:

```
ASIGN/V1=PERNO:REC*[3].XYZ
```

Acceder a las propiedades de objeto de un informe

Puede crear sus propias plantillas de informe y de etiqueta personalizadas. PC-DMIS las utiliza para mostrar los datos de los informes en una ventana de informe (consulte **Ver | Ventana de informe**). Puede crear plantillas con los editores de plantillas. Los editores utilizan una interfaz de tipo Visual Basic que permite insertar componentes especiales, denominados "objetos", así como cambiar su ubicación y tamaño.

Cada objeto consta de "propiedades" que definen cómo aparece y qué información contiene. Algunas de estas propiedades se comparten con todos los demás objetos; otras se comparten solamente con los objetos relacionados, mientras que otras son exclusivas de ese objeto determinado.

El lenguaje de expresiones de PC-DMIS puede consultar el informe que esté cargado en un momento dado y almacenar en una variable los valores de las propiedades de un objeto determinado. Puede obtener valores de tipo cadena, entero y real con esta sintaxis:

Sintaxis de consulta de propiedades



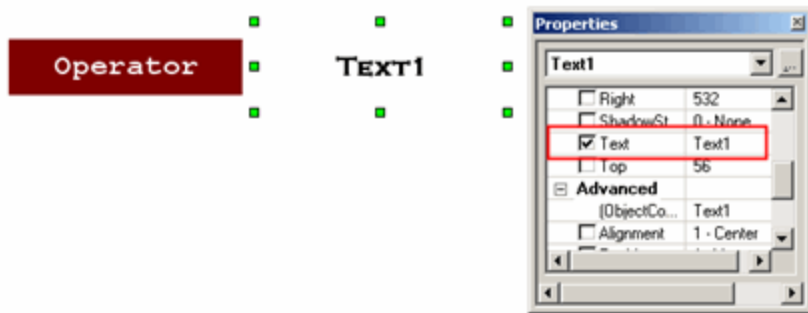
```
ASIGN/V1 = Report.<nombre objeto>.<nombre propiedad>
```

Report es una referencia al informe que está cargado. <Nombre de objeto> es el nombre exclusivo del objeto. <Nombre propiedad> es un nombre de propiedad válido para ese objeto.

Ejemplo



Supongamos que la plantilla de informe tiene un objeto de texto denominado "Text1" que desea utilizar en el informe final para mostrar el nombre del operador. PC-DMIS almacena la cadena de caracteres real que representa el nombre del operador se almacenará en la propiedad **Texto** del objeto. Por omisión, la propiedad de texto (el texto mostrado) tiene inicialmente el valor "Text1" (vea la figura siguiente). Como es una propiedad asignada por el usuario, el valor de la propiedad cambiará cuando escriba el nombre durante la ejecución.



Cuadro de diálogo Propiedades donde se muestra un objeto seleccionado y la propiedad que se va a consultar

Para utilizar el código del lenguaje de expresiones con el fin de consultar la propiedad de texto de este objeto de texto y obtener los datos tecleados, se utilizaría el comando siguiente:

```
ASIGN/V1 = Report.Text1.Text
```

En este código:

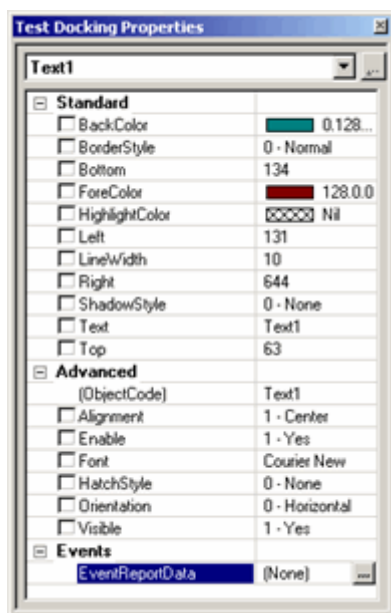
"Report" indica al código que busque el informe cargado en la ventana de informe.

"Text1" indica que se debe buscar el objeto denominado "Text1".

"Text" indica que se debe buscar la propiedad "Text" en ese objeto. A continuación, el valor de la propiedad "Text" se pasa a la variable V1, que posteriormente se puede manipular o mostrar mediante el lenguaje de expresiones de PC-DMIS.

Buscar propiedades

Para buscar las propiedades que están asociadas a un objeto determinado, acceda a la plantilla de informe en el editor de plantillas de informe (**Archivo | Generar informe | Edición | Plantilla de informe**), seleccione el objeto y pulse el botón derecho del ratón en el objeto para mostrar su hoja de propiedades.



Hoja de propiedades de un objeto de texto

La hoja de propiedades contiene dos columnas. La columna de la izquierda muestra el nombre de la propiedad. La columna de la derecha muestra el valor actual. Asegúrese de que utiliza el nombre exacto de la propiedad en el código de la expresión.



Al consultar los valores de las propiedades, es posible que algunas propiedades devuelvan un valor numérico que parece inservible. Normalmente, esto sucede cuando la propiedad tiene una lista establecida de opciones disponibles. PC-DMIS devuelve un valor interno para la propiedad seleccionada que no está relacionado con la propiedad mostrada.

Por ejemplo, el objeto **Text** tiene una propiedad **Orientation** con estos valores:

- 0 - Horizontal
- 1 - Vertical hacia arriba
- 2 - Vertical hacia abajo

Sin embargo, si obtiene el valor mediante el lenguaje de expresiones de PC-DMIS, el software devolverá en su lugar lo siguiente:

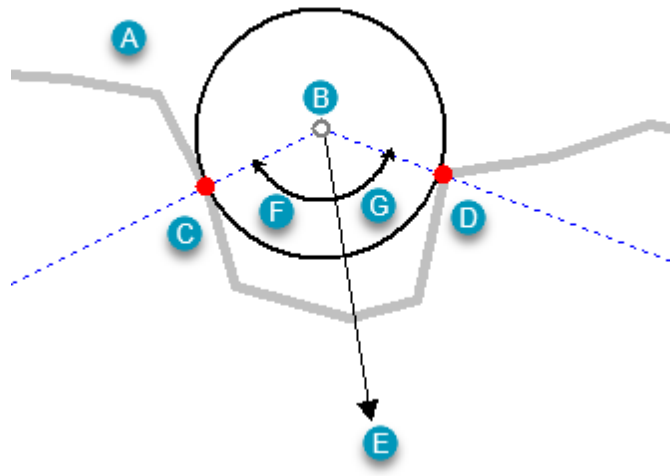
- 0 (para Horizontal)
- 900 (para vertical hacia arriba)
- -900 (para vertical hacia abajo)

Pueden ser necesarios varios intentos antes de poder determinar qué valores devueltos corresponden al valor mostrado en la hoja de propiedades.

Acceder a la información de un círculo mínimo de escaneado construido

Puede utilizar las expresiones de PC-DMIS para obtener información de un elemento de círculo que esté construido con un radio determinado en un punto mínimo a lo largo de un escaneado lineal. Consulte el tema "Construir un círculo en el punto mínimo del escaneado" en el capítulo "Construir nuevos elementos a partir de los ya existentes" para obtener más información.

Al construir un elemento de círculo mínimo de un escaneado, el círculo utiliza en último término un vector (denominado vector inferior) para entrar en contacto con la línea del escaneado. Sólo entra en contacto con la línea en dos lugares llamados puntos de contacto (PUNTOCONTACTO1 y PUNTOCONTACTO2). PC-DMIS puede utilizar luego estos puntos para determinar los ángulos del vector inferior con estos puntos de contacto (ÁNGULOCONTACTO1 y ÁNGULOCONTACTO2). Considere este ejemplo:



A: La línea del escaneado con la que se construye el círculo.

B: La posición XYZ final del centroide del círculo.

C: El punto de contacto a la izquierda del vector inferior. Se denomina CONTACTPOINT1.

D: El punto de contacto a la derecha del vector inferior. Se denomina CONTACTPOINT2.

E: El vector inferior.

F: El ángulo que va del vector inferior a CONTACTPOINT1. Se denomina CONTACTANGLE1.

G: El ángulo que va del vector inferior a CONTACTPOINT2. Se denomina CONTACTANGLE2.

Las expresiones que se detallan a continuación solamente funcionan con este tipo de elemento de círculo construido. También puede utilizar CONTACTPOINT2 en la sintaxis siguiente para que se devuelva en su lugar la información equivalente haciendo uso del segundo punto de contacto.

`ASIGN/V1=CIR1.PUNTOCONTACTO1.XYZ`

Devuelve la información de punto XYZ para el primer punto de contacto del círculo con la línea, PUNTOCONTACTO1.

`ASIGN/V1=CIR1.CONTACTPOINT1.X`

Devuelve la información de X correspondiente a CONTACTPOINT1.

`ASIGN/V1=CIR1.CONTACTPOINT1.Y`

Devuelve la información de Y correspondiente a CONTACTPOINT1.

`ASIGN/V1=CIR1.CONTACTPOINT1.Z`

Devuelve la información de Z correspondiente a CONTACTPOINT1.

`ASIGN/V1=CIR1.CONTACTPOINT1.IJK`

Devuelve el vector IJK de CONTACTPOINT1 al centroide del círculo.

`ASIGN/V1=CIR1.CONTACTPOINT1.I`

Devuelve el valor I del vector CONTACTPOINT1 IJK anterior.

`ASIGN/V1=CIR1.CONTACTPOINT1.J`

Devuelve el valor J del vector CONTACTPOINT1 IJK anterior.

`ASIGN/V1=CIR1.CONTACTPOINT1.K`

Devuelve el valor K del vector CONTACTPOINT1 IJK anterior.

`ASIGN/V1=CIR1.CONTACTANGLE1`

Devuelve el ángulo que va del vector inferior a CONTACTPOINT1.

`ASIGN/V1=CIR1.CONTACTANGLE2`

Devuelve el ángulo que va del vector inferior a CONTACTPOINT2.

`ASIGN/V1=CIR1.CONTACTANGLE`

Devuelve la suma de los valores absolutos de CONTACTANGLE1 y CONTACTANGLE2. No debe ser mayor que 180 grados.