

MultiBase Cosmos

Notas a la versión 5.5

BASE100

BASE 100, S.A.
www.base100.com

Índice

1. IMPLEMENTACIONES	3
2. MEJORAS	4
2.1 ODBC	4
2.2 MONITOR.....	4
2.3 COSMOS.....	4
2.4 CTSQL.....	5
2.5 APIS	5
2.5.1 BMAPSDLL.....	5
2.5.2 TTCHARDLL.....	5
3. VARIABLES DE ENTORNO	6
4. FUNCIONES DE LA API TTCHARDLL	8
5. MÉTODOS DE LA CLASE SIMPLECONTROL	10
6. MÉTODOS DE LA CLASE DATE.....	12
7. FUNCIONES CTSQL	13
7.1 DECODE	13
7.2 NVL.....	13
7.3 ABS	14
7.4 TRUNC.....	14
7.4.1 Fecha.....	14
7.4.2 Números.....	15
7.5 SIGN.....	15
7.6 TO_DATE	16
7.7 TO_CHAR.....	16
7.7.1 Fechas	16
7.7.2 Números (smallint, integer, serial, money, decimal)	16
8. CORRECCIONES	19
8.1 RUNTIME.....	19
8.2 GESTOR BASE DE DATOS	20
8.3 ODBC	20
8.4 GATEWAY ORACLE	20
8.5 API TTCHARDLL.....	20
ANEXO I.....	21
ANEXO II	23
ANEXO III	24
ANEXO IV	25
ANEXO V	26
ANEXO VI	28

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

[NT_CO_5.5_v5]

1. Implementaciones

- En esta versión se han implementado las siguientes funciones en el gestor de base de datos CTSQL:
 - DECODE
 - NVL
 - ABS
 - TRUNC
 - SIGN
 - TO_CHAR
 - TO_DATE
- En la API TTCHARTDLL se ha implementado una serie de funciones que permiten personalizar el color de las líneas en el eje de ordenadas y abscisas (ChartSetXAxisColorLabel, ChartSetYAxisColorLine), personalizar el color del texto de las etiquetas del eje X (ChartSetAxisColorLabel), mostrar las label de los elementos como tooltips (ChartShowElementLabelAsTooltip), indicar los valores máximo y mínimo de la escala de valores (ChartSetViewDataRange), e indicar la precisión en los números de las etiquetas de éstos en el eje de ordenadas (CharSetYAxisLabelPrecision).
- Se han implementado dos métodos nuevos de la clase DATE para el formateo de fechas:
 - TOCHAR
 - FROMCHAR
- Se ha implementado una nueva variable de entorno (FETCHLASTBEFOREEDITNEW) que permite que el cursor se posicione en la última fila del grid de la tabla de líneas, en un cabeceras-líneas, cuando se ejecuta el método EditNew de la clase FormTable y el valor de la QUERYBUFFERING es TRUE. De esta manera se evitará la aparente pérdida de líneas.
- Posibilidad de cambiar el valor de las propiedades Background (color de fondo) y Foreground (color de texto) de los controles EditField Multilínea, tanto en diseño como en ejecución.
- Se podrá cambiar el color del texto de los controles Edit Field deshabilitados que no estén asociados a un FormTable (ver variable de entorno EDITDISABLEDCTRTEXTCOLOR).

2. Mejoras

2.1 ODBC

- Se ha cambiado el instalador del ODBC de 32 bits. Este instalador permitirá instalar y desinstalar el driver.
- Se han realizado modificaciones para mejorar la compatibilidad con LibreOffice.

2.2 Monitor

- Se ha añadido un control check "Auto Refresh" que permite al administrador de la base de datos elegir si el refresco de la información que se muestra en la pantalla se realizará de manera automática o manual pulsando el botón de refresco.
- Nueva funcionalidad que permite consultar en una lista las tablas y/o registros que se encuentran bloqueados en una base de datos.

2.3 Cosmos

- Se ha ampliado el tamaño de la memoria de intercambio en las aplicaciones cliente de Cosmos (cosrun, csq) para que la lectura de registros sea más rápida cuando el tamaño del registro leído supere los 1.024 bytes.
- Se ha optimizado el método Last de la clase FormTable cuando la lectura de datos de la base de datos se encuentra almacenada en el búffer.
- Se ha modificado el instalador del CD Máster de Cosmos para que permita instalar los drivers ODBC de 32 y 64 bits.
- La fuente que se utilizará en los controles Drop List en las listas editables será la misma que se haya asignado en la lista.
- Posibilidad de modificar la propiedad AutomaticSort de los controles grid en ejecución mediante el método SetProperty de la Clase Control.
- En las operaciones de funciones de agregados en listas agrupadas (ShowMultiColumnGroupDlg y ShowListAsMultiColumnGroup), si no se ha especificado el número de decimales para la columna sobre la que se calculan, el resultado se mostrará con 3 decimales.
- Listas Editables. En la edición de tipos calendar, al seleccionar un valor acepta automáticamente e invoca al evento ListAcceptEdit.
- En las lista de agrupados y agregados se ha implementado un nueva vista que permite elegir si los totales del último grupo de ruptura de columnas agrupadas se mostrarán en la misma fila del grupo, tanto en las listas como en la exportación a Excel, cuando no se muestre el detalle. Para poder realizar esta acción se han modificado los métodos ShowMultiColumnGroupDlg y ShowListAsMultiColumnGroup de la clase SimpleControl.

El método ShowMultiColumnGroupDlg mostrará primero un nuevo control Check con el texto "Compacto" en la barra. Cuando esté marcado este check y desmarcado el check del detalle, los totales de la última columna de agrupados se mostrarán en la misma fila del grupo.

Para habilitar esta acción en el método ShowListAsMultiColumnGroup se deberá indicar su valor con COMPACTMODE:YES/COMPACTMODE:TRUE o COMPACTMODE:NO/COMPACTMODE:FALSE.

- En las listas agrupadas (ShowMultiColumnGroupDlg y ShowListAsMultiColumnGroup), si se decide no mostrar las líneas de detalle (con el check *Detalle* deshabilitado), se podrá elegir si se desea o no que se muestren las columnas que no pertenezcan a una ruptura de grupo o a un cálculo de agregados. Para ello, se deberá definir la variable de entorno HIDECOLUMNIFNODETAIL.
- Se han añadido mensajes de aviso en algunas ventanas de las aplicaciones de Cosmos para indicar al usuario que un proceso está en ejecución:
 - En la parte inferior izquierda de la ventana del Sql interactivo (Csql) se muestra un texto “Running...” mientras se están ejecutando sentencias Sql.
 - En la exportación a csv, xls, unl, ods, pdf, etc. se muestra un mensaje en la ventana del Sql interactivo (Csql) o en la del método SelectWindow con el texto: “Exporting list, please wait...”.
 - En la exportación a Excel de los listados desde el Preview se muestra una ventana con una barra de progreso.
- En los procesos mencionados en el punto anterior, y también con el método ExportToExcel de la clase SimpleControl, el runtime de Cosmos refrescará automáticamente la pantalla, minimizando así el número de veces que se muestra el mensaje “No responde”.
- A partir de esta versión de Cosmos la exportación de datos con los métodos ExportToExcel, ExportToHTML, ExportToODS y ExportToPdf en los controles Grid será completa con independencia de que los datos estén almacenados en el búffer del control (variable de entorno QUERYBUFFERING=TRUE).
- A las variables de entorno DEACTIVATEDPKCOLOR y DEACTIVATEDTEXTCOLOR se les podrá asignar un valor con formato RGB(r,g,b). Hasta ahora solo se podía indicar el valor entero correspondiente a un color dado en formato RGB.
- Método AllowColumnHeaderFilter. Se ha optimizado el proceso de carga de registros para mostrar solamente valores únicos si existe un gran número de duplicados.

2.4 Ctsql

Posibilidad de utilizar en la instrucción insert las nuevas funciones: ABS, INITCAP, SUBSTR, CONCAT, LOWER, UPPER, LTRIM, RTRIM, DECODE, NVL, SIGN, TO_DATE y TO_CHAR.

2.5 APIS

2.5.1 BMAPSDLL

Optimización en el consumo de recursos y en la velocidad de carga de mapas.

2.5.2 TTCHARDLL

Se ha optimizado el repintado de los gráficos, ya que en algunas ocasiones era muy lento. Esta optimización es útil en las instalaciones de Cosmos sobre dispositivos tablet.

3. Variables de entorno

FETCHLASTBEFOREEDITNEW

Si su valor es TRUE, al añadir un nuevo registro con el método EditNew de la clase Formtable en un mantenimiento, el cursor se posicionará automáticamente en el último registro de la tabla.

Esta nueva variable de entorno debe utilizarse solo en caso de que el valor de la variable de entorno QUERYBUFFERING sea TRUE.

Si el valor de la variable de entorno QUERYBUFFERING es FALSE, cuando se ejecuta el método EditNew el cursor ya se posiciona automáticamente en la última línea porque la lista está cargada completamente.

Los valores posibles son TRUE y FALSE.

El valor de esta variable se podrá cambiar en ejecución y en la sección Environment global del fichero de configuración del proyecto y de Cosmos.

EDITDISABLEDCTRTEXTCOLOR

Permite cambiar el color del texto de los controles Edit Field, Drop List y Drop Edit deshabilitados (propiedad Disabled=TRUE).

El valor de esta variable se puede asignar en los ficheros de configuración del proyecto o de Cosmos en el environment global, con el método SetOption de la clase Module o Form o con el método PutEnv de la clase Module.

Si esta variable se define en el fichero de configuración del proyecto o en el fichero de configuración de Cosmos, su valor debe ser un número entero resultado de la combinación de los colores rojo, verde y azul (Ver método Rgb de la clase Module).

Sintaxis:

```
EDITDISABLEDCTRTEXTCOLOR=65280  
EDITDISABLEDCTRTEXTCOLOR=RGB (0 , 255 , 0)
```

HIDECOLUMNIFNODETAIL

Esta variable de entorno permite indicar si se mostrarán u ocultarán las columnas que no pertenezcan a una ruptura de grupo o a un cálculo de agregados cuando no se visualicen las líneas de detalle.

Los valores posibles son: TRUE o YES y FALSE o NO.

Su valor por defecto es FALSE.

Esta variable se define en el fichero de Cosmos o del proyecto en la sección "Environment Global". También se puede modificar su valor con el método PutEnv de la clase Module.

GWSENDLABELINDESCRIBE

Retorna el alias, si se ha indicado alguno, cuando se ejecuta el método SqlDescribe de la clase SqlServer de Cosmos contra el motor CTSQL en las columnas de tipo expresión.

Cuando se ejecuta ese método contra el gateway de Oracle o Informix, aunque exista alias retorna la cadena "(expression)". Si se desea que retorne el alias como en CTSQL será necesario declarar esta variable en el fichero de configuración del gateway (gworacle.env) con el valor YES, o exportarla con el método Putenv de la clase SqlServer.

MINIMIZEFORMCASCADE

Esta variable de entorno indica al runtime de Cosmos si al minimizar una ventana se minimizará también su ventana padre.

Esta variable solo se podrá definir en el fichero de configuración de Cosmos o en el del proyecto.

DBCONNBLOCKSIZE

Permite cambiar el tamaño del búffer en una conexión de Cosmos contra una base de datos CTSQL, Gworacle o Gwinformix.

4. Funciones de la API TTCHARTDLL

- **ChartShowElementLabelAsTooltip.** Permite mostrar las label de los elementos como tooltip.

Sintaxis:

```
ChartShowElementLabelAsTooltip(hWindows as integer, showAsTooltip as boolean)  
return boolean
```

Parámetros:

hWindows Identificador de la ventana del control de usuario.
showAsTooltip Booleano que indica si se mostrará la label como etiqueta o como tooltip.

Retorna: TRUE si la operación se ha realizado correctamente.

El texto del tooltip se corresponde con el texto indicado en el parámetro elementLabel de la función NewChartElement. El título del tooltip se corresponderá con el valor de la etiqueta del eje de abscisas.

Si se desea cambiar el texto del título del tooltip se indicará el texto del cuerpo y del título del tooltip separados por un carácter *pipe* ("|") en el parámetro elementLabel de la función NewChartElement.

- **ChartSetXAxisColorLabel.** Permite definir el color del texto de la etiqueta del eje X.

Sintaxis:

```
ChartSetXAxisColorLabel(hWindows as integer, textColor as integer) return boolean
```

Parámetros:

hWindows Identificador de la ventana del control de usuario.
textColor Color que se asignará al texto.

Retorna: TRUE si la operación se ha realizado correctamente.

- **ChartSetXAxisColorLine.** Permite definir el color de las líneas del eje X.

Sintaxis:

```
ChartSetXAxisColorLine(hWindows as integer, lineColor as integer) return boolean
```

Parámetros:

hWindows Identificador de la ventana del control de usuario.
lineColor Color que se asignará a las líneas del eje X.

Retorna: TRUE si la operación se ha realizado correctamente.

- **ChartSetYAxisColorLine.** Permite definir el color de las líneas del eje Y.

Sintaxis:

```
ChartSetYAxisColorLine(hWindows as integer, lineColor as integer) return boolean
```

Parámetros:

hWindows	Identificador de la ventana del control de usuario.
lineColor	Color que se asignará a las líneas del eje Y.

Retorna: TRUE si la operación se ha realizado correctamente.

- **ChartSetViewDataRange.** Permite establecer los valores mínimo y máximo de visualización de la escala de valores en el eje de ordenadas.

Sintaxis:

```
ChartSetViewDataRange (hWindows as integer, mainYAxis as boolean, enable as boolean, minValue as decimal, maxValue as decimal) return boolean
```

Parámetros:

hWindows	Identificador de la ventana del control de usuario.
mainYAxis	Indica si la escala de valores se mostrará en el eje de ordenadas principal o secundario. Si su valor es TRUE los valores se mostrarán en el eje principal, mientras que si es FALSE lo harán en el eje secundario.
minValue	Valor mínimo.
maxValue	Valor máximo.

Retorna: TRUE si la operación se ha realizado correctamente.

Esta función tiene una serie de limitaciones que se detallan a continuación:

- Los valores inferior y superior del eje de ordenadas no pueden ser negativos.
 - El valor inferior tiene que ser menor que el valor máximo.
 - Si alguno de los valores del gráfico es negativo, Cosmos dibujará el gráfico sin tener en cuenta los valores indicados como máximo y mínimo.
- **ChartSetYAxisLabelScale.** Permite indicar la escala (número de decimales) en los valores de las etiquetas del eje de ordenadas. La escala por defecto es de dos decimales.

Sintaxis:

```
ChartSetYAxisLabelPrecision (hWindows as integer, scale as integer, mainYAxis as boolean) return boolean
```

Parámetros:

hWindows	Identificador de la ventana del control de usuario.
scale	Número entero que indica la escala.
mainYAxis	Indica si la escala de valores se mostrará en el eje de ordenadas principal o secundario. Si su valor es TRUE los valores se mostrarán en el eje principal, mientras que si es FALSE lo harán en el eje secundario.

Retorna: TRUE si la operación se ha realizado correctamente.

Esta función no se puede utilizar si se desea mostrar las etiquetas del eje de ordenadas de manera abreviada (ChartSetShortYAxisLabel).

5. Métodos de la clase SimpleControl

- **SetListColorByLevel.** Permite indicar el color de fondo de los distintos niveles en las listas de agrupados para facilitar su visualización.

Sintaxis:

```
SetListColorByLevel(levelNumber as Smallint ,levelBackColor as Integer)
```

Parámetros:

levelNumber	Nivel de ruptura al que se le asignará el color de fondo, siendo 1 el primer nivel.
levelBackColor	Número entero en formato RGB que indica el color de fondo del nivel indicado en el primer parámetro.

NOTA: La lista no se visualizará con los colores definidos con este método hasta que no se ejecute el método EnableListColorByLevel.

- **EnableListColorByLevel.** Este método indica al runtime si en los controles List Box se mostrarán los colores definidos con el método SetListColorByLevel para los niveles de ruptura.

Sintaxis:

```
EnableListColorByLevel (setColor as Boolean)
```

Parámetros:

setColor	Booleano que indica si se activa o se desactiva la visualización de colores de fondo por nivel de agrupado.
----------	---

Si el valor de este parámetro es TRUE, las etiquetas de las funciones de agregados indicadas en los métodos ShowMultiColumnGroupDlg y ShowListAsMultiColumnGroup se mostrarán en la columna del grupo por el que se produce la ruptura y no en la columna del agregado. Además, añadirá una línea en blanco al finalizar los agregados, facilitando de esta manera la localización de las rupturas de grupo.

Detalle	Compacto	ColorByLevel	Etiqueta de la función de agregado	Línea en blanco entre grupos
Sí	No	No	No	No
Sí	No	Si	Sí	Sí
No	No	No	No	No
No	No	Si	Sí	Sí
No	Sí	No	No	No
No	Sí	Sí	Sí	Sí

NOTA: la llamada a este método se debe realizar antes de la llamada a los métodos ShowMultiColumnGroupDlg y ShowListAsMultiColumnGroup.

- **SetGridColumnDisabled.** Este método permite modificar la propiedad disabled de los control Grid.

Sintaxis:

```
SetGridColumnDisabled (index as integer, disabled as boolean)
```

Parámetros:

index	Identificador de la columna.
disabled	Booleano que indica si se deshabilita o habilita la propiedad.

Los valores posibles son: TRUE y FALSE:

TRUE. Deshabilita la columna.

FALSE. Habilita la columna.

- **SetTextDireccion.** Permite indicar la orientación del texto dentro de un control TEXT.

Sintaxis:

```
SetTextDireccion(align as Smallint)
```

Parámetros:

align	Valor entero que indica la dirección del texto. Sus posibles valores son: <ol style="list-style-type: none">1. Orientación horizontal (valor por defecto).2. Orientación vertical comenzando por la parte inferior del control.3. Orientación vertical comenzando por la parte superior del control.
-------	--

- **SetTextHorizontalAlign.** Permite indicar la alineación (izquierda, centrado y derecha) del texto en los controles tipo TEXT.

Sintaxis:

```
SetTextHorizontalAlign(align as Smallint)
```

Parámetros:

align	Valor entero que indica la alineación del texto. Sus posibles valores son: <ol style="list-style-type: none">1. Texto alineado a la izquierda.2. Texto centrado.3. Texto alineado a la derecha.
-------	---

- **SetTextVerticalAlign.** Permite indicar la alineación vertical del texto en los controles tipo TEXT, pero solo para los que tienen asignada la propiedad "multiline".

Sintaxis:

```
SetTextVerticalAlign(align as Smallint)
```

Parámetros:

align	Valor entero que indica la alineación. Sus posibles valores son: <ol style="list-style-type: none">1. La posición del texto dentro del control estará en la parte superior del control (alineación vertical superior).2. El texto ocupará todo el espacio vertical disponible (alineación vertical centrada). Ésta es la alineación por defecto.
-------	---

6. Métodos de la clase DATE

- **ToChar.** Retorna una cadena de caracteres a partir de un objeto de tipo date y un formato que recibe como parámetro.

Sintaxis:

```
ToChar(format as Char ,NLS as Char default NULL) return Char
```

Parámetros:

format Formato que se utiliza para convertir el valor a una cadena (ver [Anexo I](#)).

NLS Parámetro opcional que indica el idioma de la conversión (ver [Anexo II](#)).

Retorna: Cadena de caracteres con el formato y la fecha indicados.

Por ejemplo:

```
lDtFecha = '28/02/2014';  
lDtFecha.ToChar('YYYY-DD-MONTH');
```

Retorna: 2014-28-Febrero.

- **FromChar.** Asigna al objeto Date una fecha creada a partir de una expresión alfanumérica y un formato.

Sintaxis:

```
FromChar(date as Char ,format as Char ,NLS as Char default NULL) return Date
```

Parámetros:

date Cadena de caracteres con formato fecha.

format Formato de fecha del primer parámetro (ver [Anexo III](#)).

NLS Parámetro que indica el idioma en que están el día y el mes del primer parámetro (ver [Anexo II](#)).

Retorna: Objeto date con el valor modificado.

Por ejemplo:

```
DtFecha.FromChar('2009-28-FEBRERO', 'YYYY-DD-MONTH')
```

Asignará a la variable DtFecha el valor 28 de febrero de 2009.

Retorna: 28-02-2009.

7. Funciones CTSQL

7.1 DECODE

Esta función compara el parámetro *expr* con cada uno de valores de búsqueda uno a uno. Si *expr* es igual a un valor de búsqueda la base de datos devuelve el resultado correspondiente.

Funciona como un “if xxx then xxxx” o un switch case dentro de la sentencia select.

Sintaxis:

```
DECODE(expression, search, result [, search, result]... [, default ])
```

Parámetros:

expression	Expresión a comparar.
search	Valor que se compara con el primer parámetro.
result	Valor que se devuelve si los valores del primer y segundo parámetros son iguales.

Por ejemplo:

```
SELECT cliente,  
DECODE(provincia, 1, 'Álava',  
        8, 'Barcelona',  
        28, 'Madrid',  
        'Otra provincia') resultado  
FROM clientes;
```

En esta consulta, por cada registro retornado compara el valor del campo provincia con 1, 8 y 28. Si coincide con alguno de ellos retorna el valor correspondiente (1: Álava, 8: Barcelona y 28: Madrid). En caso de que no se corresponda con ninguno de estos tres retornará “Otra provincia”.

7.2 NVL

Compara *expr1* con NULL. Si *expr2* es NULL retornará *expr2*.

Funciona como un “if *expr1* == NULL then *expr2*, else *expr1*”.

Sintaxis:

```
NVL(expr1, expr2)
```

Parámetros:

expr1	Valor a comparar con el valor NULL.
expr1	Valor que mostrará si <i>expr1</i> es nulo.

Por ejemplo:

```
SELECT cliente,  
NVL(descripcion, "cliente sin nombre")  
FROM clientes
```


7.4.2 Números

Trunca el número indicado en el primer parámetro al número de decimales indicado en el segundo parámetro.

TRUNC (numero, decimales)

Parámetros:

número	Valor a truncar.
decimales	Número que indica a cuántos decimales queremos truncar el valor del primer parámetro.

Si el segundo parámetro es un número negativo, truncará hacia la izquierda del punto decimal poniendo ceros en su lugar (-1 unidades, -2 decenas, -3 centenas, ...).

Por ejemplo:

TRUNC(125.815, 0)	125
TRUNC(125.815, 1)	125.8
TRUNC(125.815, 2)	125.81
TRUNC(125.815, 3)	125.815
TRUNC(-125.815, 2)	-125.81
TRUNC(125.815, -1)	120
TRUNC(125.815, -2)	100
TRUNC(125.815, -3)	0
TRUNC(45633,-3)	45000
TRUNC(45633,-2)	45600

7.5 SIGN

Esta función retorna -1 si el número que recibe como parámetro es negativo, 1 si es positivo y 0 si es 0.

Sintaxis:

SIGN (num)

Parámetros:

Num	Número a evaluar.
-----	-------------------

```
SELECT
decode(sign( det.diario - tope.diario ), -1, det.diario, 0, det.diario, to-
pe.diario ) AS diario
decode( sign( det.semanal - tope.semanal ), -1, det.semanal, 0, det.semanal,
tope.semanal ) AS semanal
decode( sign( det.mensual - tope.mensual ), -1, det.mensual, 0, det.mensual,
tope.mensual ) AS mensual
FROM detalle det, tope;
```

7.6 TO_DATE

Esta función convierte una cadena de caracteres en un valor de tipo DATE a partir de un formato de entrada.

Sintaxis:

```
TO_date(cadena[, formato [, nlsparams ]])
```

Parámetros:

cadena	Cadena de caracteres que representa una fecha.
formato	Formato en el que está la cadena de caracteres que se le ha pasado como primer parámetro (ver Anexo III).
nlsparams	Parámetro opcional que permite introducir modificaciones en la llamada a la función, tales como el idioma en el que se especificarán las constantes de nombre del día de la semana o del mes (ver Anexo II).

```
Select TO_DATE('31/10/2009','DD/MM/YYYY') FROM  
RESULTADO: 31/10/2009  
Select TO_DATE('2009-31-ENE','YYYY-DD-MON') FROM  
RESULTADO: 31/01/2009
```

7.7 TO_CHAR

Convierte valores numéricos y de fecha a un valor de cadena de caracteres.

7.7.1 Fechas

Convierte un valor de tipo DATE en una cadena con el formato especificado en el segundo parámetro.

Sintaxis:

```
TO_CHAR(fecha, [formato [, nlsparams] ])
```

Parámetros:

fecha	Campo de tipo fecha que se va a convertir.
formato	Formato al que se desea convertir el tipo de dato (ver Anexo I).
nlsparams	Parámetro opcional que permite introducir modificaciones en la llamada a la función, tales como el idioma en el que se especificarán las constantes de nombre del día de la semana o del mes (ver Anexo II).

```
SELECT  
TO_CHAR(date(today), "dd-fmMonth-yyyy", "NLS_DATE_LANGUAGE=spanish") FROM ...  
Si today es 20/01/2013, retornará la cadena "20-Enero-2013".  
SELECT  
TO_CHAR(date(today), "dd-FMmm-yyyy") FROM ...  
Si today es 20/01/2013, retornará la cadena "20-1-2013".  
Retorna 1 en lugar de 01 porque se ha indicado el operador FM.
```

7.7.2 Números (smallint, integer, serial, money, decimal)

Convierte un valor de tipo numérico en una cadena de caracteres con el formato especificado en el segundo parámetro.

Sintaxis:

```
TO_CHAR(numero, [formato [, nlsparams] ])
```

Parámetros:

numero	Valor de tipo numérico que se desea formatear.
formato	Indica el formato que se le va a dar al valor numérico (ver Anexo V).
nlsparams	Parámetro opcional que permite introducir modificaciones en la llamada a la función, tales como el separador decimal, el separador de miles y el identificador de moneda.

Los valores que admite son NLS_NUMERIC_CHARACTERS, NLS_CURRENCY, NLS_ISO_CURRENCY. Se podrá indicar una o varias variables a la vez en la misma sentencia. Los valores que se asignan a estas variables podrán ir entrecomillados, con dos comillas simples o sin comillas (ver [Anexo VI](#)).

Por ejemplo:

```
SELECT TO_CHAR(-10000, 'L99G999G999G999D99PR',
'NLS_NUMERIC_CHARACTERS='',.', '
NLS_CURRENCY = 'AusDollars'
')...
```

Retornará: <AusDollars10,000.00>.

El valor se muestra entre los signos de mayor menor porque es un valor negativo y se ha indicado el operador PR.

“AusDollars” porque se ha indicado el operador L y la variable NLS_CURRENCY tiene ese valor.

```
SELECT TO_CHAR(-10000, 'C99G999G999G999D99pr',
'NLS_NUMERIC_CHARACTERS = ',.', '
NLS_ISO_CURRENCY = 'united kingdom' ')
```

Retornará: <GBP10.000,00>.

La cadena “GBP” es el identificador internacional de moneda correspondiente al país indicado en NLS_ISO_CURRENCY (“United Kingdom”), que es la variable usada al indicar el operador “C” en el formato.

El número de caracteres de la cadena de retorno se corresponderá con el número de caracteres indicados en el parámetro “formato”.

Si el número de dígitos de la parte entera del número es mayor que el número de dígitos de la parte entera del formato, retornará una cadena de caracteres con el carácter “#”.

Si el número de dígitos de la parte entera del número es menor que el número de dígitos de la parte entera del formato, retornará el número convertido a una cadena de caracteres y rellenada por la izquierda con tantos espacios en blanco como caracteres faltan hasta completar la longitud del formato.

Si el número de dígitos de la parte decimal del número es mayor que el número de dígitos de la parte decimal del formato, retornará una cadena con el número redondeado al número de decimales indicado en el formato.

Si el número de dígitos de la parte decimal del número es menor que el número de dígitos de la parte decimal del formato, retornará una cadena con el número de decimales del número a convertir.

```
SELECT  
TO_CHAR(1234.567, "999g999d999") FROM ...
```

Si la variable DBMONEY no está definida o está definida como “,” (coma), retornará la cadena:

“ 1.234,567”
[espacio en blanco][espacio en blanco]1.234,567

Si la variable DBMONEY está definida como “.” (punto), retornará la cadena:

“ 1,234.567”
[espacio en blanco][espacio en blanco]1,234.567.

Si se indica la variable NLS_NUMERIC_CHARACTERS como tercer parámetro, los separadores de decimales y de miles serán los indicados en la misma, ya que NLS_NUMERIC_CHARACTERS tiene precedencia sobre DBMONEY.

8. Correcciones

8.1 Runtime

- Cosrun. Si se editaba la última columna de una lista string, al efectuar Scroll horizontal el ancho del campo de edición era distinto al de la columna.
- Cosrun. En determinados casos, al ejecutar el método SelectWindow y pulsar sobre la cabecera de una columna para reordenarla, el resultado no era correcto. Este problema ocurría cuando la lectura se encontraba almacenada en el búffer.
- Cosrun. Al intentar filtrar una lista SQL con el método ShowListFilterBar, si en la sentencia SQL se había incluido una constante en la lista de columnas, el runtime no construía correctamente la frase SQL y no se mostraba ningún registro.
- Cosrun. Lista agrupada con el método ShowMultiColumnGroupDlg. El rendimiento del control tipo lista se veía afectado si almacenaba muchas filas y el cursor se encontraba situado al final de la lista.
- Cosrun. Al agrupar y ordenar una lista, si ésta tenía muchos registros y el proceso tardaba mucho, el puntero del ratón desaparecía y no se sabía si el proceso habría terminado o no. Ahora se muestra como puntero el reloj de arena mientras dura el proceso.
- Cosrun. Se permiten valores negativos en listas editables.
- Cosrun. Listas editables. Cuando la celda tenía un control de tipo drop edit, tras aceptar o cancelar no invocaba a los eventos ListAcceptEdit y ListCancelEdit.
- Cosrun. Si el texto descriptivo de la etiqueta del método GroupListColumns de la Clase SimpleControl era muy largo se producía un error de protección y se abortaba la ejecución del programa.
- Cosrun. El método Using de la clase Date con el parámetro 5 no retornaba correctamente el formato para el mes en Windows 8.1.
- Cosrun. En las listas editables con controles drop edit, cuando éste se desplegaba y posteriormente posicionaba el cursor del ratón en otra celda el foco permanecía en él y no pasaba a la nueva celda.
- Cosrun. Error en la exportación a Excel en las listas agrupadas con el método ShowListAsMultiColumnGroup, sin líneas de detalle y con el método AlternateBackColor.
- Cosrun. Al exportar a Excel un listado desde el método Preview de Cosmos, en modo solo datos y con muchas páginas, cuando se intentaba abrir la hoja Excel daba un error de estilo y tabla.
- Cosrun. En algunos casos, el método TotalizeColumn provocaba un error de ejecución en Cosmos si la sintaxis del tercer parámetro (condiciones) no era correcta.
- Cosrun. Se ha corregido un error de protección general que se producía al cerrar una ventana con una lista si ésta no había terminado de cargarse.
- Cosrun. No exportaba a Excel correctamente las lista de tipo "by column".
- Cosrun. La exportación a Excel de listas de tipo string y árbol no se realizaba correctamente si existían columnas calculadas y columnas decimales.

A partir de esta versión de Cosmos no será necesario utilizar el método NumericColumnsFormattedToChar para la exportación a Excel en la lista con columnas calculadas.

- Cosrun. En los controles TEXT multilínea vertical right o vertical left con más de una línea en su texto, éste no se mostraba correctamente.
- Cosrun. Método AllowColumnHeaderFilter. En los controles List Box de tipo Sql y con la variable de entorno "QUERYBUFFERING = TRUE", al desplegar la lista de la cabecera no se mostraban todos los registros.

8.2 Gestor base de datos

- CTSQL. Al concatenar con la función Concat, si el primer campo era de tipo decimal o Money no eliminaba los espacios en blanco.
- CTSQL. No se ordenaban correctamente los tipos de datos datetime. Este error se producía en las versiones del motor desde la 3.6 0.27 a la 3.6 0.33, ambas inclusive. Afecta a las versiones de Cosmos 5.0, 5.1, 5.2, 5.3 y 5.4.
- CTSQL. Si existía un índice por los campos A,D y otro índice por los campos A,B,C , en el "order by A,B,C" el motor seleccionaba el primer índice, siendo el segundo más óptimo. Se ha corregido para que en caso de haber más de un índice susceptible de ser elegido para la ordenación, elija el que tenga más campos implicados en la misma.
- TTTOOLS. En la aplicación TTTOOLS las views de la base de datos se mostraban como elementos a reparar o chequear. Al intentar repararla, mostraba el siguiente mensaje de error: "Error en nombre de DB y tabla".
- TTTOOLS. En algunas ocasiones se vaciaba la lista de tablas cuando se pulsaba el botón de "update statistics", siendo necesario volver a abrir la base de datos para realizar operaciones de chequeo o reparación de tablas.

8.3 ODBC

- ODBC. Al conectarse desde OpenOffice o LibreOffice al CTSQL y realizar una consulta no retornaba todos los registros.
- ODBC. Al conectarse desde LibreOffice y OpenOffice a una tabla en la que uno de los campos de la clave primaria era de tipo fecha, no permitía hacer scroll correctamente. Además, daba error al intentar modificar un dato desde LibreOffice Base.

8.4 Gateway Oracle

- GWORACLE. No traducía correctamente las sentencias outer con campos constantes en la cláusula where.
- GWORACLE. No traducía correctamente las sentencias outer cuando se utilizaban alias en el nombre de tablas.
- GWORACLE. Al ejecutar el método SqlDescribe de Cosmos sobre una columna de tipo expresión contra CTSQL retornaba el alias, en caso de tenerlo. Sin embargo, al ejecutar la misma frase contra el gateway de Oracle o Informix retornaba la cadena "(expression)". Para que los gateways se comporten como CTSQL se debe definir la variable de entorno GWSENDLABELINDESCRIBE=TRUE O GWSENDLABELINDESCRIBE=YES.

8.5 API TTCHARTDLL

- TTChartDll. Si en la función CharSetTitle se incluía como título la letra Ç o el símbolo del euro se producía un error.

ANEXO I

Posibles valores del parámetro format del método ToChar de la clase Date y la función TO_CHAR de CTSQL:

AD	Indicador AD sin signo de puntuación.
BC	Indicador BC sin signo de puntuación.
A.D.	Indicador AD con signo puntuación.
B.C.	Indicador BC con signo puntuación.
SCC	Siglo.
CC	Siglo.
DDD	Día ordinal del año.
DD	Día del mes.
DAY	Nombre del día de la semana en una cadena de 9 caracteres de longitud, rellenando con espacios en blanco por la derecha. El idioma en el que se mostrará por defecto el día de la semana será el indicado en la configuración regional ("locale") del sistema operativo, salvo que se indique otro idioma en el tercer parámetro de la función con la variable NLS_DATE_LANGUAGE.
DL	Fecha en formato largo. Por ejemplo: "25/12/2013" retornaría "Miércoles 25 de diciembre de 2013".
DS	Fecha en format corto. Mostrará la fecha en el formato indicado en la variable de entorno DBDATE, por defecto: "dd/mm/yyyy".
DY	Nombre del día se la semana abreviado a tres caracteres.
IW	Semana del año (1-52 o 1-53) basado en el estándar ISO ¹ .
IYYY	Año en 4 dígitos basado en el estándar ISO ¹ .
IYY	Últimos 3 dígitos del año basados en el estándar ISO ¹ .
IY	Últimos 2 dígitos del año basados en el estándar ISO ¹ .
I	Último dígito del año basado en el estándar ISO ¹ .
MM	Mes (01-12; Enero = 01).
MONTH	Nombre del mes en una cadena de 10 caracteres de longitud, rellenando con espacios en blanco por la derecha. El idioma en el que se mostrará por defecto el día de la semana será el indicado en la configuración regional ("locale") del sistema operativo, salvo que se indique otro idioma en el tercer parámetro de la función con la variable NLS_DATE_LANGUAGE.
MON	Nombre del mes abreviado a 3 caracteres. El idioma en el que se mostrará por defecto el día de la semana será el indicado en la configuración regional ("locale") del sistema operativo, salvo que se indique otro idioma en el tercer parámetro de la función con la variable NLS_DATE_LANGUAGE.
Q	Trimestre del año (1, 2, 3, 4; Enero - Marzo = 1).
RM	Mes en números romanos (I-XII; Enero = I).

¹ Según la norma ISO 8601, las semanas siempre comienzan en lunes y terminan en domingo. La primera semana del año será aquella que contenga el primer jueves del año. De esta manera, se puede dar el caso de que existan años con 52 semanas y años con 53 semanas, que los días 29, 30 y 31 de diciembre de un año pertenezcan a la primera semana del año siguiente, o que los días 1, 2 y 3 de enero pertenezcan a la última semana del año anterior.

WW	Semana del año (1-53), donde la semana 1 comienza en el primer día del año y continúa hasta el séptimo día del año.
W	Semana del mes (1-5), donde la semana 1 comienza en el primer día del mes y finaliza en el séptimo día del mes.
SYYYY	Año en 4 dígitos; La S indica que los años anteriores al año 1 llevarán como prefijo un signo menos ("-").
YYYY	Año en 4 dígitos.
Y,YYY	Año en 4 dígitos con la coma en la tercera posición.
YYY	Últimos 3 dígitos del año.
YY	Últimos 2 dígitos del año.
YEAR	Año escrito (en español).
Y	Último dígito del año.
D	Día de la semana (1-7).
FM	Retorna el valor del operador situado a su derecha formateado sin ceros a la izquierda. Válido en los operadores: MM, DD, IW, DDD, WW, YYY, IYY ² , YY, IY, RR.
FX	Requiere correspondencia exacta entre el carácter y el formato. CTSQL acepta el valor pero no lo trata.
RRRR	Año redondeado. Acepta entrada de 2 o 4 dígitos. Si se introducen 2 dígitos, retorna lo mismo que RR. Si se introducen 4 dígitos, retorna lo mismo que YYYY.
RR	Últimos 2 dígitos del año. Si es < 50, pertenece al siglo XXI. Si es igual o mayor a 50, pertenece al siglo XX.
SP	<i>Spelled number</i> . Sufijo que, colocado a la derecha de un operador, retorna su valor como número escrito. Es posible colocarlo como sufijo en los operadores: DD, MM, YYYY, IYYY, SYYYY, CC, SCC, DDD, WW, W, IW, IYY ¹ , YYY, YY, IY ² , Q, RR, Y,YYY.

Los operadores se pueden declarar tanto en minúsculas como en mayúsculas. Si se declaran en minúsculas, en el caso de mostrar un texto (nombre del mes, nombre del día de la semana, numeral del día, etc.) éste aparecerá en minúsculas. Si la primera letra del operador se declara en mayúsculas, el texto aparecerá en mayúsculas.

² *Ibid.* nota 1.

Anexo II

NLS_DATE_LANGUAGE	Esta variable especifica el idioma de los nombres de los días de la semana y de los meses. El idioma se indicará en inglés, pudiendo hacerlo entre comillas o no. Por ejemplo: NLS_DATE_LANGUAGE=spanish NLS_DATE_LANGUAGE="spanish"
-------------------	---

Anexo III

Tabla con los valores del formato para las fechas en el método FromChar y función TO_DATE del CTSQL

D	Día ordinal de la semana. El motor lo lee pero no lo trata.
DD	Día del mes.
DDD	Día del año.
DAY	Día de la semana en formato largo. Lo leemos pero no lo tratamos ni comprobamos que sea correcto.
DY	Día de la semana en formato corto (3 caracteres). Lo leemos pero no lo tratamos ni comprobamos que sea correcto.
W	Semana ordinal del mes (1 carácter). Lo leemos pero no lo tratamos ni comprobamos que sea correcto.
WW	Semana ordinal del año (2 caracteres). Lo leemos pero no lo tratamos ni comprobamos que sea correcto.
MM	Número del mes (2 caracteres).
MON	Nombre del mes en formato corto (3 caracteres).
MONTH	Nombre del mes en formato largo (n caracteres).
RM	Mes en números romanos.
Q	Trimestre del año (1 carácter). Lo leemos, pero no lo tratamos.
YYY	Últimos 3 dígitos del año.
YYYY	Últimos 4 dígitos del año.
YY	Últimos 2 dígitos del año.
RR	Últimos 2 dígitos del año. Si es < 50, pertenece al siglo XXI. Si es igual o mayor a 50, pertenece al siglo XX.
RRRR	Muestra el año en 4 dígitos . Admite 4 dígitos o 2 dígitos. Si son 2 dígitos, el resultado será el mismo que el formato RR (Si es < 50, pertenece al siglo XXI. Si es igual o mayor a 50, pertenece al siglo XX). Si son 4 dígitos los que se indican retornará el mismo valor que el formato YYYY.
Y	Último dígito del año.

Anexo IV

Formatos de fecha que acepta el parámetro format del la función TRUNC del CTSQL

Formato	Valor
YYYY, YYYY, YEAR, SYEAR, YYY, YY, Y	Retorna 1 de enero del año de la fecha indicada.
IYYY, IY, I	Retorna al primer día del año ISO ³ de la fecha indicada. Será el lunes anterior al 4 de enero del año indicado.
MONTH, MON, MM, RM	Retorna el primer día del mes (día 1).
WW	Redondea al mismo día de la semana que el primer día del año. Por ejemplo, si el primer día del año fue un miércoles, retornará el miércoles anterior a la fecha indicada.
W	Devuelve el día de la semana que coincida con el primer día del mes. Por ejemplo: Si el primer día de mes fue un martes y la fecha que se le pasa como parámetro es un jueves, devuelve la fecha correspondiente al martes de esa semana. Si la fecha que se pasa como parámetro es un lunes, la fecha que muestra será la del martes de la semana anterior.
IW	Semana ISO ³ . Devuelve la fecha del día de la semana igual al primer día de la semana del año ISO ³ , es decir, al lunes anterior de la semana a la que pertenece la fecha indicada.
DDD, DD	Retorna el mismo día.
DAY, DY, D	Retorna al lunes de la semana a la que pertenece la fecha indicada. Igual que IW.

³ *Ibid.* nota 1.

Anexo V

Posibles valores del parámetro format de la función TO_CHAR para valores numéricos de CTSQL:

Formato	Ejemplo	Descripción
"," (coma)	9,999	Retorna una coma en la posición especificada. Se corresponderá con el separador de miles. Se pueden especificar múltiples comas en un modelo de formato de número. Restricciones: Una coma no puede aparecer a la derecha de un carácter de separador de decimales en un modelo de formato de número.
"." (punto)	99.99	Retorna un punto en la posición especificada. Se corresponderá con el separador decimal. Restricciones: Un punto decimal no puede aparecer a la derecha de otro punto decimal.
\$	\$9999	Retorna el signo del dólar en el lugar especificado.
0 (cero)	0999 9990	Rellena con ceros por la izquierda (parte entera) o por la derecha (parte decimal) hasta la posición donde se encuentre el carácter "0" en el caso de que no existan dígitos en esa posición.
9	9999	Retorna el valor con el número especificado de dígitos con un espacio inicial si es positivo o con un signo menos delante si es negativo. Los ceros iniciales estarán en blanco, excepto para el valor "0" del formato.
B	B9999 B.9999	Si el número es 0, retorna una cadena de blancos. Si la parte entera es 0, retorna una cadena de blancos en su lugar.
C	C999	Retorna el nombre ISO de moneda en la posición especificada. Este nombre será el correspondiente a la configuración regional ("locale") del sistema operativo. El símbolo de la moneda podrá cambiarse asignando un valor a la variable NLS_ISO_CURRENCY.
D	99D99	Retorna el carácter decimal en la posición especificada. Si no está definida la variable de entorno DBMONEY, se corresponderá con el signo "," (coma). Si está definida la variable de entorno DBMONEY, se corresponderá con el valor indicado en la misma. Si se define la variable NLS_NUMERIC_CHARACTERS en el tercer parámetro, se corresponderá con el indicado en la misma, prevaleciendo esta variable sobre la DBMONEY.
FM	FM90.9	Retorna un valor sin blancos a la izquierda.
G	9G999	Retorna el carácter separador de grupo (miles) en la posición especificada. Si no está definida la variable de entorno DBMONEY, se corresponderá con el signo "." (punto). Si está definida la variable de entorno DBMONEY, se corresponderá con el valor contrario al indicado en la misma: Si DBMONEY=, (coma), entonces G será "." (punto). Si DBMONEY=. (punto) entonces G será "," (coma). Si se define la variable NLS_NUMERIC_CHARACTER en el tercer parámetro, se corresponderá con el indicado en la misma, prevaleciendo esta variable sobre la DBMONEY. Un separador de grupo no puede aparecer a la derecha de un separador de decimales.

Formato	Ejemplo	Descripción
L	L999	Retorna el símbolo local de moneda en la posición especificada. El valor por defecto viene determinado por la configuración regional ("locale") del sistema operativo. Se puede modificar su valor indicándolo en la variable NLS_CURRENCY en el tercer parámetro de la función.
MI	9999MI	Si el número es negativo, añade un signo "-" al final del mismo. Si el número es positivo, añade un espacio en blanco al final del mismo. El formato MI solo puede aparecer en la última posición del modelo de formato de número.
PR	9999PR	Si el número es negativo, retorna el número entre signos de menor que (<) y mayor que (>) Si el número es positivo, retorna el número con un blanco al comienzo y otro al final. El indicador de formato PR solo puede aparecer en la última posición del modelo de formato de número.
RN	RN	Retorna el número en caracteres romanos. El valor puede ser un entero entre 1 y 3999.
S	S9999 9999S	Si el número es negativo y se indica "S" al principio del formato, mostrará un signo "-" al comienzo del mismo. Si el número es positivo y se indica "S" al principio del formato, mostrará un signo "+" al comienzo del mismo. Si el número es negativo y se indica "S" al final del formato, mostrará un signo "-" al final del mismo. Si el número es positivo y se indica "S" al final del formato, mostrará un signo "+" al final del mismo.
X	XXXX xxxx	Retorna el valor hexadecimal de la parte entera del número. Solo acepta valores positivos o 0.

Anexo VI

Posibles valores del parámetro `nlsparams` (Native Language Support) de la función `TO_CHAR` para el formateo de valores numéricos:

NLS_NUMERIC_CHARACTERS	Esta variable especifica el carácter decimal y el de separador de miles que se emplearán en los operadores “G” y “D” de la cadena de formato del segundo parámetro. Es una cadena de dos caracteres donde el primero indica el separador decimal (para “D”) y el segundo el separador de miles (para “G”). Si no se especifica esta variable, se emplearán los caracteres indicados en la variable <code>DBMONEY</code> , es decir, esta variable tiene precedencia sobre la <code>DBMONEY</code> .
NLS_CURRENCY	Esta variable indica la cadena de caracteres (texto libre) que se empleará como símbolo local de moneda cuando se utilice el operador “L” en la cadena de formato del segundo parámetro. Si no se especifica esta variable, cuando se asigne el valor “L” al formato se empleará el símbolo local de moneda correspondiente a la configuración regional (“locale”) del sistema operativo.
NLS_ISO_CURRENCY	Al utilizar el operador “C” en la cadena de formato, éste se sustituirá por el nombre internacional de la moneda ⁴ correspondiente con la configuración regional (“locale”) del sistema operativo. Si se desea emplear un nombre de moneda diferente a la configuración regional, se indicará en la variable <code>NLS_ISO_CURRENCY</code> el nombre del país (en inglés) cuya moneda se desea utilizar.

⁴ El símbolo internacional de moneda es un estándar internacional, denominado ISO 4217, cuyo objetivo es definir códigos de tres letras para las monedas existentes en todos los países.

Como regla general, las dos primeras letras del código corresponden al código del país de la moneda según el estándar ISO 3166-1, y la tercera es la inicial de la moneda en sí. Esto no es así en el caso, por ejemplo, del euro, cuyo código ISO 3166-1 es EUR.