

# Tabla de Contenido

Contenido	Pag.
PERMISOS PERSONALIZADOS.....	7
IMPRESIÓN DE FORMATOS PERSONALIZADOS .....	7
ADICION DE TABLAS Y CAMPOS PERSONALIZADOS A LA BASE DE DATOS ..	7
MENUS PERSONALIZADOS.....	8
FUNCIONES Y PROCEDIMIENTOS UTILIZADOS PARA LA PROGRAMACIÓN DE UN EVENTO PERSONALIZADO.....	9
FUNCIONES DE USO GENERAL .....	9
FUNCIONES PARA MANEJO DE PUERTOS .....	19
FUNCIONES PARA MANEJO DE ARCHIVOS PLANOS .....	20
FUNCIONES DE IMPRESIÓN.....	21
FUNCIONES DE BASE DE DATOS .....	23
CONEXION CON OTRA GDB .....	26
CONEXIÓN CON ODBC.....	27
CONEXIÓN CON ARCHIVOS DBF .....	27
ENVIIO DE ARCHIVOS VIA MAIL .....	28
FUNCIONES IMPRESORA FISCAL BEMATECH.....	29
FUNCIONES IMPRESORA FISCAL BIXOLON.....	30
FUNCIONES DE WEB SERVER .....	31
TABLA VARIOS .....	31

## MANUAL DE PROGRAMACIÓN DE EVENTOS PERSONALIZADOS

Un evento personalizado es un segmento de código que se ejecuta en los módulos de Visual TNS después de un evento determinado tal como asentar, imprimir o editar documentos. El evento personalizado se programa desde el módulo de administración en la opción de Log y Configuración – Evento Personalizado. Este código del evento personalizado se almacena en la tabla “Scripts” de la base de datos de la empresa donde se guarda el nombre del evento y su descripción.

Existen algunos eventos los cuales se deben grabar con un nombre predefinido por Visual TNS para que el programa los ejecute en el momento requerido. Los eventos se pueden ejecutar al momento Asentar o editar un Documento.

Los nombres definidos con los que se deben grabar los eventos para que se ejecuten en el momento indicado se enuncian a continuación.

**EDIT\_FACVTA:** Se ejecuta en Ventas y en Ventas Mostrador en dos momentos: Antes de guardar el encabezado de la factura o cuando se hace clic en el botón de datos adicionales. (Ver Figura 1.)

**Guardar encabezado**

Cód. Artículo	Artículo	Bod	Und	Cantidad	Vr. Neto	Vr. Parcial	IVA
00.03.000604	LIC.EXT.TNS-FACT INVENT,CXC,CXP.	03	UN	1.000	327,120.00	327,120.00	16.00
00.00.000035	ASESORIA Y CAPACITACION SECTOR PRIV	03	HR	1.000	21,999.99	21,999.99	16.00

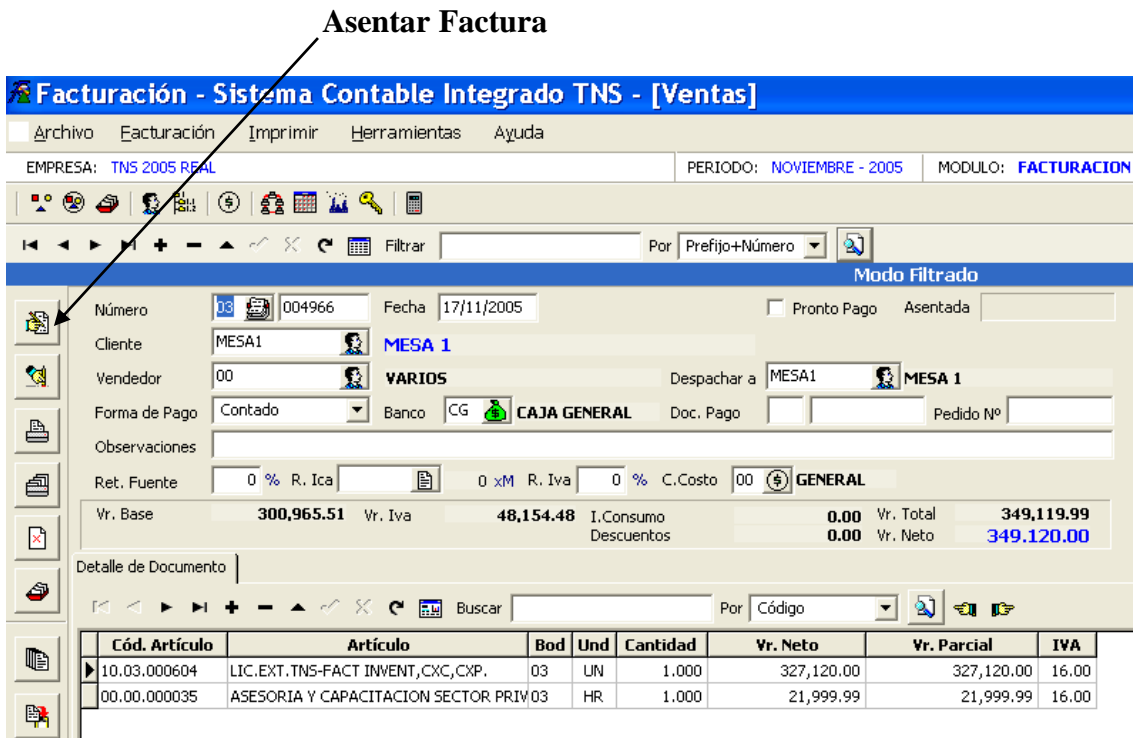
**Datos Adicionales**

**Figura 1. Ventas**

Este evento es útil cuando se quieren pedir datos adicionales en la factura de venta que sirvan para complementar la información ya existente o hacer algún proceso con ellos.

**EDIT\_DEFACVTA:** Se ejecuta en Ventas y Venta Mostrador al momento de insertar o editar un artículo del detalle de la factura de venta.

**ASEN\_FACVTA:** Se ejecuta en Ventas y en Ventas Mostrador antes de llevar a cabo el proceso de asentar la factura. (Ver Figura 2.)



**Figura 2. Ventas**

Este evento es útil para realizar validaciones previas de la información y evitar asentar un documento que no cumpla con los criterios establecidos por la empresa.

**REVE\_FACVTA:** Se ejecuta en Ventas y en Ventas Mostrador antes de llevar a cabo el proceso de Reversar la factura.

**ASEN\_FACVTADES:** Se ejecuta en Ventas y en Ventas Mostrador Después de llevar a cabo el proceso de Asentar la factura.

**REVE\_FACVTADES:** Se ejecuta en Ventas y en Ventas Mostrador Después de llevar a cabo el proceso de Reversar la factura.

**ANUL\_FACVTA:** Se ejecuta en Ventas y en Ventas Mostrador Antes de llevar a cabo el proceso de Anular la factura.

**ANUL\_FACVTADES:** Se ejecuta en Ventas y en Ventas Mostrador Después de llevar a cabo el proceso de Anular la factura.

**TNS\_GE\_ImprimirFact(vkardexid:string)**, Imprime una Factura desde un evento personalizado ejecutado desde el modulo de facturación. Vkardexid el id de la factura.

**TNS\_GE\_ASENTARDOCUMENTO** (vdocid:String):Boolean Asienta Facturas, Compras, Notas de inventario y consumos.

**TNS\_GE\_ImprimirFact(vkardexid:string)** función para imprimir factura desde evento personalizado

**TNS\_GE\_ImpComanda(vtipoimp,vkardexid:string)** función para imprimir comanda desde evento personalizado. Vtipoimp si es R es reimprimir comanda.

De Igual manera funcionan estos eventos para los demás documentos de los demás módulos así:

### **DEVOLUCIONES EN VENTAS**

ASEN\_DEVVTA, REVE\_DEVVTA, ASEN\_DEVVTADES, REVE\_DEVVTADES.

#### **REMISIONES DE SALIDA**

EDIT\_REMSAL, ASEN\_REMSAL, REVE\_REMSAL, ASEN\_REMSALDES, REVE\_REMSALDES.

#### **PEDIDOS DE VENTA**

EDIT\_PEDVTA, EDIT\_DEPEDVTA, ASEN\_PEDVTA, REVE\_PEDVTA, ASEN\_PEDVTADES, REVE\_PEDVTADES.

#### **COTIZACIONES**

EDIT\_FACCOT, EDIT\_DEFACCOT, ASEN\_FACCOT, REVE\_FACCOT, ASEN\_FACCOTDES, REVE\_FACCOTDES.

#### **COMPRAS**

EDIT\_FACCOM, EDIT\_DEFACCOM, ASEN\_FACCOM, REVE\_FACCOM, ASEN\_FACCOMDES, REVE\_FACCOMDES.

### **DEVOLUCIONES EN COMPRAS**

ASEN\_DEVCOM, REVE\_DEVCOM, ASEN\_DEVCOMDES, REVE\_DEVCOMDES.

#### **REMISIONES DE ENTRADA**

ASEN\_REMENT, ASEN\_REMENTDES.

#### **PEDIDOS DE ENTRADA**

EDIT\_PEDC, ASEN\_PEDC, REVE\_PEDC, ASEN\_PEDCDES, REVE\_PEDCDES.

#### **NOTAS DE INVENTARIO**

EDIT\_NINV, ASEN\_NOTAINV, ASEN\_NOTAINVDES, REVE\_NOTAINV, REVE\_NOTAINVDES

## **TRASLADOS**

EDIT\_TRASLA, IMP\_TRALA, ASEN\_TRASLA, ASEN\_TRASLADES,  
REVE\_TRASLA, REVE\_TRASLADES

## **CARTERA**

EDIT\_DOC, ASEN\_DOC, REVE\_DOC, ASEN\_DOCDES, REVE\_DOCDES,  
ASEN\_RECCAJA, REVE\_RECCAJA, ASEN\_RECCAJADES, REVE\_RECCAJADES

## **TESORERIA**

EDIT\_DOCTES, EDIT\_DEDOCTES, EDIT\_RECTES, EDIT\_DERECTES,  
EDIT\_DERECTESFP.

## **CARTERA/TESORERIA**

**TNS\_GE\_ASENTARDOCCAR** (vdocid:String;vmodo:Boolean):Boolean Asienta o reversa un Documento de Cartera o de Tesorería desde un evento personalizado

**TNS\_GE\_ASENTARDOCUMENTOCAR** (vdocid:String;vmodo:Boolean):Boolean Asienta o reversa un Recibo de Cartera o desde Tesorería desde un evento personalizado

**TNS\_GE\_ASENTARDOCCAR** (vdocid: String;vmodo:Boolean) Asienta o reversa un Documento de Cartera o desde Tesorería desde un evento personalizado

**TNS\_GE\_ASENTARDOCUTESORERIA** (vdocid:String;vmodo:Boolean):Boolean. Asienta un documento de Tesorería desde un evento personalizado

## **CARTERA FINANCIERA**

EDIT\_DOCFIN, ASEN\_DOCFIN, REVE\_DOCFIN, ASEN\_DOCFINDES,  
REVE\_DOCFINDES, ASEN\_RECCARFIN, REVE\_RECCARFIN,  
ASEN\_RECCARFINDES, REVE\_RECCARFINDES.

## **CONTABILIDAD**

EDIT\_DECOM

**TNS\_GE\_COMPRES** (movid,prefijo,numero:String) Resume un Comprobante de contabilidad.

**TNS\_GE\_GENCON** (vmodulo:string;vmodo:string;) se asigna integración o no a un modulo con contabilidad Ejemplo TNS\_GE\_GenCon('FACT',False) deja de integrar Facturación con Contabilidad

## **HOSPITAL**

EDIT\_FACTHOS, ASEN\_FACTHOS, ASEN\_FACTHOSDES, REVE\_FACTHOS,  
REVE\_FACTHOSDES, ANUL\_FACTHOS, ANUL\_FACTHOSDES, SCRIPT\_FACTHOS  
(Datos Adicionales de Factura Hospitalaria)

Cuando las Facturas son Capitadas:

ASEN\_FACTHOSCAP, ASEN\_FACTHOSCAPDES, SCRIPT\_FACTHOSCAP

SCRIPT\_HISTOCLINICA (Datos Adicionales de Historias Clínicas).

SCRIPT\_PROFESIONAL (Datos Adicionales para la tabla Profesionales).

SCRIPT\_SERVICIO (Datos Adicionales para la tabla Servicios).

SCRIPT\_USUAHOSP (Datos Adicionales para la tabla Usuarios).

SCRIPT\_HOSPITAL (Se dispara al iniciar el módulo de Hospital)

SCRIPT\_CONTRATO (Datos Adicionales para la tabla Contratos).

DISPARASER (Al insertar o editar un servicio)

DISPARACONT (Al insertar o editar un contrato)

DISPARAUSUA (Se dispara al insertar o editar un usuario o después de ejecutar la opción Asignar Contrato con click derecho sobre el usuario).

DISPARAPROF (Al insertar o editar un profesional)

GENSUPERSALUD (Se dispara en el menú Herramientas en la opción “Generar Archivo SuperIntendencia de Salud).

ASIGNAR\_SERCITA (Datos adicionales al asignar una cita en la agenda)

HOS\_REPHISTOCLI (Evento que se ejecuta antes de imprimir la Historia Clínica.)

En todos los eventos que se derivan de los documentos siempre se puede saber el ID del documento actual mediante el parámetro general 1 que se puede obtener con la función LeeParam(1).

**EDIT\_TERADIC:** Se Ejecuta sobre el botón de Datos Adicionales en la pantalla de Terceros. Sirve para elaborar pantallas que pregunten datos complementarios de un tercero.

**EDIT\_CLASIFADIC:** Se Ejecuta sobre el botón de Datos Adicionales en la pantalla de Clasificación de Terceros.

**EDIT\_ARTADIC:** Se Ejecuta sobre el botón de Datos Adicionales en la pantalla de Artículos.

**EDIT\_CRMLLAMA:** Para el Módulo de CRM se ejecuta sobre el botón datos adicionales de la pantalla de llamadas.

Eventos que se ejecutan al iniciar cada modulo:

**INI\_FAC:** Se ejecuta al iniciar el módulo de facturación

**INI\_CAR:** Se ejecuta al iniciar el módulo de Cartera

**INI\_CARFIN:** Se ejecuta al iniciar el módulo de Cartera Financiera

## **PERMISOS PERSONALIZADOS**

Para definir permisos personalizados se deben configurar desde el módulo de Administración en la opción de permisos de usuario en la pestaña Específicos. Aquí se encuentra un botón Personalizar que ejecuta el evento EDIT\_PERMISO que es donde se codificará toda la programación de permisos especiales que se quieran asignar a tablas o procesos personalizados. El evento personalizado recibe en el Parámetro 1 el ID del usuario sobre el que se está asignando el permiso.

Para hacer uso de estos permisos se usa la función **TNS\_GE\_PERMISO (vpermiso)**. Esta función devuelve Verdadero o Falso indicando si el usuario actual tiene permiso para llevar a cabo la operación indicada en el parámetro *vpermiso*.

## **IMPRESIÓN DE FORMATOS PERSONALIZADOS**

Cuando se quiere imprimir un documento en un formato personalizado de acuerdo a los requerimientos de la empresa se debe programar el evento personalizado de impresión ya sea en modo texto o gráfico y se guarda con un nombre libre ejm: IMP\_FACVTA. Después desde la aplicación requerida se entra por Archivo-Configuración –Generales y se sobre el documento que se desee se configura formato “Personalizado” indicando el nombre del evento personalizado que ejecutará la impresión del reporte.

## **ADICION DE TABLAS Y CAMPOS PERSONALIZADOS A LA BASE DE DATOS**

Al elaborar un evento personalizado muchas veces se requiere pedir datos adicionales para almacenarlos en la base de datos. Estos campos o tablas se pueden crear por la consola de edición de la base de datos como el IBOconsole mediante comandos SQL y después desde el evento personalizado pedir los datos y almacenarlos.

Pero ocurre que si se recibe una actualización y se optimiza la base de datos esta información se perdería. Para solucionar este inconveniente se creó una estructura de archivos que se deben crear para que el optimizador de Visual TNS considere estos cambios y los incluya dentro de su proceso de optimización.

Para llevar a cabo este proceso se deben crear 2 archivos en la Ruta Visual TNS\Específicos. Estos archivos son:

**SCRIPTNS.txt:** Aquí se escriben los comandos SQL para crear las nuevas tablas, campos, relaciones etc. que se desean adicionar a la base de datos.

**TABLASTNS.txt.** Aquí se deben especificar los nombres de las nuevas tablas y sus campos auto numérico correspondiente que se crearon en el scriptns.txt. Se deben escribir los nombres de las tablas en orden el de optimización deben quedar y el nombre de la tabla que precederán

### **Ejemplo práctico**

Se desea crear una nueva tabla de barrios para pedirla como dato adicional en la tabla terceros.

El archivo SCRIPTNS.txt quedaría así:

```
CREATE TABLE BARRIOS (  
  IDBARRIO INTEGER,  
  CODBARRIO VARCHAR (3) NOT NULL,  
  DESBARRIO VARCHAR (40),  
  PRIMARY KEY (IDBARRIO)  
);
```

```
ALTER TABLE TERCEROS ADD CODBARRIO VARCHAR (3);
```

El Archivo TABLASTNS.txt quedaría así:

```
TERCEROS: BARRIOS, IDBARRIO
```

Esto significa que la tabla de BARRIOS tiene un campo auto numérico llamado IDBARRIO y que se debe optimizar antes que la de terceros ya que terceros necesita la información de la tabla barrios.

#### **Nota:**

Si se crean Generadores para incrementar un campo auto numérico mediante un TRIGGER este generador debe llevar el nombre del ID de la tabla seguido de \_GEN. Ejm: En la tabla BARRIO que tiene un campo auto numérico llamado IDBARRIO el generador debe llamarse IDBARRIO\_GEN.

## **MENUS PERSONALIZADOS**

Se pueden crear menús personalizados en cualquier módulo adicionando opciones al menú principal o creando nuevos ítems con submenús. Para esto se debe crear un archivo plano en la carpeta específicos llamado MENU.TXT y la sintaxis de cada ítem que se desee adicionar al menú es la siguiente:

```
MODULO:RUTA:SUB:NUEVOITEM:EVENTO
```

MODULO: Indica el módulo sobre el cual se adicionará el ítem: Contabilidad, Facturacion, Inventario, Cartera, Tesoreria, CarFinan, Carga, Produccion, CostosABC, Hospital.



RUTA: Es la ruta del menú sobre el cual se adicionará el nuevo ítem: Ejm: *Archivo,Configuracion,Generales*.

SUB: Indica si el nuevo ítem a adicionar va después de la ruta ó dentro de la ruta especificada. Ejm:

*Facturacion:Archivo,Configuracion,Generales::SUB:Pantallas:*. Indica que dentro del menú Configuración va el ítem pantallas. ó

*Facturacion:Archivo,Configuracion,Generales::Pantallas:*. Indica que el ítem pantallas va después del ítem generales.

NUEVOITEM: Es el Texto del Nuevo Item a adicionar.

EVENTO: Es el nombre del evento personalizado que se ejecutará al momento de seleccionar la opción.

A continuación se muestra un ejemplo completo:

*Facturacion:Facturación::Carbón : //Después de Facturación adiciona el ítem Carbón*  
*Facturacion:Carbón:SUB:Minas:MANMINAS //Dentro del ítem carbón adiciona Minas*  
*Facturacion:Carbón,Minas::Pacios:MANPATIOS //Después del ítem Minas adicionas Pacios*  
*Facturacion:Imprimir,Resumenes de Ventas,Listado de Ventas entre fechas::Nuevo Informe:NUEVOINFORME //Después del ítem Listado de ventas entre fechas adiciona el ítem Nuevo Informe*

## **FUNCIONES Y PROCEDIMIENTOS UTILIZADOS PARA LA PROGRAMACIÓN DE UN EVENTO PERSONALIZADO**

### **FUNCIONES DE USO GENERAL**

- **TNS\_GE\_Ceros\_i( X: Integer, Y: Integer ):string**

Completa un número entero con ceros a la izquierda retornando el valor como cadena. Ejm: TNS\_GE\_Ceros\_i(5,3) retorna '005'

- **TNS\_GE\_Input(vtexto: String):string**

Muestra una ventana con una caja de texto para pedir un dato específico y retorna dicho valor. Ejm: TNS\_GE\_input('Digite un número') pide por consola un número y retorna el valor digitado.

- **TNS\_GE\_LeeParam(vvar: Integer):string**

Lee el valor almacenado en una de las 5 variable globales que tiene el compilador para uso general. Ejem: TNS\_GE\_LeeParam(1) Lee el contenido de la variable 1. El parámetro **vvar** indica el número de la variable que deseo acceder (1..5).

- **TNS\_GE\_GrabaParam(vvar: Integer;valor:String)**

Graba un valor en una de las 5 variables globales destinadas para uso general. Ejm : TNS\_GE\_GrabaParam(3,'hola'). Guarda la cadena Hola en la variable global número 3

- **TNS\_GE\_retorna(valor: String)**

Sirve para que el Evento personalizado retorne un valor que puede ser usado desde otro módulo

- **TNS\_GE\_FechaLetras(Vfec: String):String;**

Retorna una fecha en formato mmmm dd DEL yyyy. Ejm: TNS\_GE\_FechaLetras('01/01/2005'). Retorna "ENERO 01 de 2005"

- **TNS\_GE\_Fecha\_mda(vfec: String):String**

Convierte una fecha en formato mm/dd/yyyy . Ejm: TNS\_GE\_fecha\_mda('31/12/2005'). Retorna "12/31/2005"

- **TNS\_GE\_Fecha\_dd(valor:String):String**

Extrae los días de una fecha. Ejm: TNS\_GE\_fecha\_dd('31/12/2005'). Retorna "31"

- **TNS\_GE\_Fecha\_mm(valor:String):String**

Extrae el mes de una fecha. Ejm: TNS\_GE\_fecha\_mm('31/12/2005'). Retorna "12"

- **TNS\_GE\_Fecha\_aa(valor:String):String**

Extrae el año de una fecha. Ejm: TNS\_GE\_fecha\_aa('31/12/2005'). Retorna "2005"

- **TNS\_GE\_Izq(vcadena: String, vlon: Integer)**

Extrae una cantidad caracteres (*vlon*) de la parte Izquierda una cadena (*vcad*). Ejm: tns\_ge\_izq('Hola mundo',4) retorna 'Hola'

- **TNS\_GE\_DER(vcad:String,vlon:Integer):String**

Extrae una cantidad caracteres (*vlon*) de la parte derecha una cadena (*vcad*). Ejm: tns\_ge\_der('Hola mundo',4) retorna 'undo'

- **TNS\_GE\_EjecutarEvento(varchivo: String)**

Ejecuta un evento personalizado definido en el módulo de administración

- **TNS\_GE\_Seleccionar(vtabla,vcaption,vcampo1,vcampo2,vcampo3,vnombre1,vnombre2,vnombre3:String):String**

Muestra una ventana con una grid que contiene información de una tabla especificada y permite retornar el campo ubicado en la primera columna del grid. Ejm:

TNS\_GE\_Seleccionar('plancuentas', 'Plan de cuentas', 'codigo', 'nombre', '', 'CUENTA', 'NOMBRE', ''). Muestra una grid con el plan de cuentas que contiene el código y el nombre y permite seleccionar cualquier registro retornando el código de la cuenta seleccionada.

Los parámetros de esta función son los siguiente:

vtabla: Nombre de la tabla de la Base de datos que se quiere mostrar

vcaption: Título de la ventana

vcampo1,vcampo2,vcampo3: Nombre de los campo de la tabla a mostrar

vnombre1, vnombre2, vnombre3: Título de los campos a mostrar

- **TNS\_GE\_FECHA\_DD(vfecha:string):String**

Devuelve el día de una fecha determinada

- **TNS\_GE\_FECHA\_MM(vfecha:string) :String**

Devuelve el mes de una fecha determinada

- **TNS\_GE\_FECHA\_AA(vfecha:string) :String**

Devuelve el año de una fecha determinada

- **TNS\_GE\_MONTO\_ESC(valor:Double) :String**

Convierte un valor a Letras y lo muestra con el nombre de la moneda local.

- **TNS\_GE\_MONTO\_ESC2(valor:Double,vmoneda:string) :String**

Convierte un valor a Letras y lo muestra con el nombre de la moneda que se especifica en el parámetro *vmoneda*.

- **TNS\_GE\_EjecutarExe(vnombre,parametro:String)**

Ejecuta un programa (.EXE), desde cualquier directorio. Ejm:  
tns\_ge\_ejecutarexe('notepad.exe','c:\tempo\tempo.txt') ejecuta el Bloc de Notas y abre el archivo tempo.txt.

- **TNS\_GE\_EjecutarExeBack(vnombre,parametro:String)**

Función similar a tns\_ge\_ejecutarexe pero se ejecuta en segundo plano (Background)

- **TNS\_DE\_StringReplace(vcadena,vcadIni,vcadFin):String**

Reemplaza en la cadena *vcadena*, las ocurrencias de una subcadena especificada en *vcadIni* por la subcadena especificada en *vcadFin* y retorna la cadena modificada. Ejm: *tns\_ge\_stringreplace('Lola mira las olas', 'ola', 'xxx')* retorna *'Lxxx mira las xxxs'*.

- **TNS\_GE\_PADR (vtexto:String;vcarac:String;x:Integer):String**

Completa la cadena *vtexto* hacia la derecha con el carácter *vcarac* hasta completar la longitud *x*. Y retorna la cadena modificada. Ejm: *tns\_ge\_padr('Hola','x',7)* retorna *'Holaxxx'*

- **TNS\_GE\_PADL (vtexto:String;vcarac:String;x:Integer):String**

Completa la cadena *vtexto* hacia la izquierda con el carácter *vcarac* hasta completar la longitud *x*. Y retorna la cadena modificada. Ejm: *tns\_ge\_padl('Hola','x',7)* retorna *'xxxHola'*

- **TNS\_GE\_LEEVAR (vtexto:String):String**

Lee una Variable de Visual TNS almacenada en la tabla VARIOS. Y retorna el valor que contiene.

- **TNS\_GE\_GRABAVAR (vtexto,vvalor:String):String**

Graba una Variable la tabla VARIOS de Visual TNS. En *vtexto* se especifica el nombre de la variable a almacenar y en *vvalor* se especifica el valor a almacenar.

- **TNS\_GE\_MESLETRAS (valor:Integer):String**

Retorna el nombre del mes especificado en la variable *valor*. Ejm: *tns\_ge\_mesletras(5)* retorna *'MAYO'*.

- **TNS\_GE\_FECPRIDIA(vmes,vano:Integer):String**

Retorna la fecha del primer día del mes del año especificado. Ejm: *tns\_ge\_fecpridia(6,2006)* retorna *'01/06/2006'*

- **TNS\_GE\_FECULTDIA(vmes,vano:Integer):String**

Retorna la fecha del último día del mes del año especificado. Ejm: *tns\_ge\_fecultdia(6,2006)* retorna *'30/06/2006'*

- **TNS\_GE\_FormatoTxt(vformato:String;vvalor:Double):String**

Convierte un valor numérico al formato especificado en la variable *vformato*. Ejm: `tns_ge_formatotxt('###,###,##0.00',125250)` retorna ' 125,500.00'.

- **TNS\_GE\_Consecutivo(vconse:String):String**

Retorna el siguiente consecutivo de un valor establecido en *vconse*. Ejm: `tns_ge_consecutivo('A001')` retorna 'A002'.

- **TNS\_GE\_CopyFileTo(vorigen,vdestino:String):Boolean**

Copia un archivo definido en *vorigen* a otro archivo definido en *vdestino* . Ej: `tns_ge_copyfileto("C:\hola.txt","C:\tempo\hola2.txt")`.

- **TNS\_GE\_Sleep(vtiempo:Integer)**

Espera una pausa en el programa por un tiempo determinado en *vtiempo*

- **TNS\_GE\_EditarTabla**

**(vtabla,vcaption,vcampo1,vcampo2,vcampo3,vnombre1,vnombre2,vnombre3,vnomento:String):String.**

Permite Crear el mantenimiento completo de una tabla definida en la base de datos actual. Esta función muestra el contenido de la tabla definidos en los parámetros:

*vtabla*: Nombre de la tabla a editar.

*vcaption*: Título de la ventana a Mostrar.

*Vcampo1,vcampo2,vcampo3*: Nombre de los campos de la tabla a mostrar.

*Vnombre1,vnombre2,vnombre3*: Titulo de los campos a mostrar.

*Vnomevento*: Nombre del evento personalizado externo que hará el manejo del mantenimiento de la tabla. Este evento ejecutará la inserción, actualización o borrado del registro actual. Este Evento personalizado recibe 2 parámetros globales que se pueden obtener ejecutando la función `Leeparam(1)` y `LeeParam(2)`. El parámetro 1 indica la acción a Realizar (INSERT, EDIT, DELETE) y el parámetro 2 indica el primer campo del registro actual.

- **TNS\_GE\_EditarQTabla**

**(vsql,vcaption,vcampo1,vcampo2,vcampo3,vnombre1,vnombre2,vnombre3,vnomevento:String):String.**

Es una función similar a `TNS_GE_EDITARTABLA` solo que el primer parámetro no es el nombre de una tabla sino un SQL que puede vincular varias tablas.

- **TNS\_GE\_Consec(vcodcomp,vcodpref:String):String.**

Devuelve el consecutivo siguiente para los documentos de tipo *vcodcomp* y prefijo *vcodpref*.  
Ejm: TNS\_GE\_Consec('FV','00') devuelve el siguiente consecutivo para las facturas de venta con prefijo 00.

- **TNS\_GE\_CalValores(vkdxid)**

Recalcula los totales de una factura, es decir calcula la Base, IVA, Retenciones y Neto de una factura y graba estos datos. El Parámetro recibe el ID de la factura a actualizar.

- **TNS\_GE\_Redondea(Valor: Real; Decimales:String):Real**

Redondea un valor con los decimales especificados en el parámetro.

- **TNS\_GE\_asentarDocumento(vdocid: String;vmodo:Boolean):Boolean**

Asienta un documento recibiendo como parámetros el id del kardex y el modo que puede ser true si va asentar y false si va reversar , se pueden asentar las que son tipo venta ,factura compra , nota de inventario, consumo, pedido venta, pedido compra, remisión de entrada, devolución de compra, remisión de salida

- **TNS\_GE\_AnularDocumento(vdocid: String):Boolean**

Anula un documento recibiendo como parámetro el id del kardex , se anulan los que son de tipo factura venta , pedido venta y pedido compra

- **TNS\_GE\_AnularDocumento(vdocid: String;vmodo,Anular:Boolean):Boolean**

Anula un documento recibiendo como parámetro el id del kardex y un modo que puede ser true si es asentar o false si es reversar , se anulan los que son de tipo factura venta , pedido venta y pedido compra

- **TNS\_GE\_asentarMovi(vdocid:String;vmodo:Boolean):Boolean**

Asienta solo comprobantes de contabilidad , y el modo es true si va asentar y false si va reversar

- **TNS\_GE\_ASENTARMOVI2(vcodcomp,vcodprefijo,vnumero:String;vmodo:Boolean):Boolean**

Asienta comprobantes de contabilidad, enviando como parámetros el codcomp , codprefijo, número del movimiento y un boolean que es true si va asentar y false si va a reversar

- **TNS\_GE\_ASENTARDOCUMENTO2(vcodcomp,vcodprefijo,vnumero:String;vmodo:Boolean):Boolean**

Asienta el documento en kardex, los que el codcomp sea tipo venta ,factura compra , nota de inventario, consumo, pedido venta, pedido compra, remisión de entrada, devolución de compra, remisión de salida

- **TNS\_GE\_ImprimirFact(vkardexid:string)**

Imprime una factura venta o venta mostrador , recibiendo como parametro el id del kardex

- **TNS\_GE\_ImprimirFactMos(vkardexid:string)**

Imprime una factura venta mostrador, recibiendo como parámetro el id del kardex

- **TNS\_GE\_ImpComanda(vtipoimp,vkardexid:string)**

Imprime una comanda de venta mostrador en donde se envía como parámetros:

*Vtipoimp*: que puede ser N: si es primera vez de impresión de la comanda

R: si es reimpresión de la comanda

*Vkardexid*: el kardex id

- **TNS\_GE\_asentarDocumentoCar(vdocid: String;vmodo:Boolean):Boolean**

Asienta un documento en cartera cuando es de tipo recibo caja o comprobante de egreso, el parámetro que recibe es el id del recibo y el modo es true si va asentar y false y va a reversar

- **TNS\_GE\_asentarDocCar(vdocid: String;vmodo:Boolean):Boolean**

Recibe como parámetro el id del documento y asienta el documento si es de tipo ingreso o egreso

- **TNS\_GE\_GenCon(vmodulo:string;vmodo:string)**

Activa o Desactiva la Integración con el módulo de contabilidad para evitar que se genere comprobantes al momento de asentar un documento. *Vmodulo* indica el módulo que se quiere Activar la Integración o desintegración con Contabilidad, puede ser CAR (cartera), TES (Tesorería), FACT (Facturación) o INV (Inventario). *Vmodo* Indica si se activa o desactiva la Integración T:Activar, F: Desactivar.

- **TNS\_GE\_GenCar(vmodulo:string;vmodo:string)**

Activa o Desactiva la Integración con el módulo de cartera donde *vmodulo* puede ser es HOS (hospital) y *vmodo* Indica si se activa o desactiva la Integración T: Activar, F: Desactivar.

- **TNS\_GE\_GenInv(vmodulo:string;vmodo:string;)**

Activa o Desactiva la Integración con el módulo de inventario donde vmodulo puede ser es HOS (hospital) y vmodo Indica si se activa o desactiva la Integración T: Activar, F: Desactivar.

- **TNS\_GE\_asentarDocuTesoreria(vdocid:String;vmodo:Boolean):Boolean**

Asienta un documento en tesorería donde recibe como parámetros el id del documento y vmodo es true si va asentar y false si va a reversar

- **TNS\_GE\_EditarQTabla6(vsql, vcaption, vcampo1, vcampo2, vcampo3, vcampo4, vcampo5, vcampo6, vnombre1, vnombre2, vnombre3, vnombre1, vnombre2, vnombre3, vnomevento:String):String.**

Es una función similar a TNS\_GE\_EDITARQTABLA solo que tiene 3 campos adicionales de consulta, es decir, seis en total.

- **TNS\_GE\_EDITARQTABLAAREA(area:Integer;vcaption,vcampo1,vcampo2,vcampo3,vcampo4,vcampo5,vcampo6,vnombre1,vnombre2,vnombre3,vnombre4,vnombre5,vnombre6,vnomevento:String):String**

Permite crear el mantenimiento completo de una tabla definida en la base de datos actual. Los parámetros que se reciben son:

*Área:* Las áreas que se pueden utilizar son de la 6 a la 12

*vcaption:* Título de la ventana a Mostrar.

*Vcampo1,vcampo2,vcampo3,vcampo4,vcampo5,vcampo6:* Nombre de los campos de la tabla a mostrar.

*Vnombre1,vnombre2,vnombre3, vnombre4,vnombre5,vnombre6:* Titulo de los campos a mostrar.

*Vnomevento:* Nombre del evento personalizado externo que hará el manejo del mantenimiento de la tabla. Este evento ejecutará la inserción, actualización o borrado del registro actual. Este Evento personalizado recibe 2 parámetros globales que se pueden obtener ejecutando la función Leeparam(1) y LeeParam(2). El parámetro 1 indica la acción a Realizar (INSERT, EDIT, DELETE) y el parámetro 2 indica el primer campo del registro actual.

- **TNS\_GE\_QSeleccionar(vsql,vcaption,vcampo1,vcampo2,vcampo3,vcampo4,vcampo5,vcampo6,vcampo7,vnombre1,vnombre2,vnombre3,vnombre4,vnombre5,vnombre6,vnombre7:String):String**

Permite Mostrar una consulta en una ventana de selección de datos y retornar el primer campo de dicha consulta.



- **TNS\_GE\_QSeleccionar2(vsqli,vcaption,vcampo1,vcampo2,vcampo3,vcampo4,vcampo5,vcampo6,vcampo7,vnombre1,vnombre2,vnombre3,vnombre4,vnombre5,vnombre6,vnombre7:String;vlog1,vlog2,vlog3,vlog4,vlog5,vlog6,vlog7:Integer):String**

Permite Mostrar una consulta en una ventana de selección de datos y retornar el primer campo de dicha consulta.

- **TNS\_GE\_QSeleccionarAnt(vsqli,vcaption,vcampo1,vcampo2,vcampo3,vcampo4,vcampo5,vcampo6,vcampo7,vnombre1,vnombre2,vnombre3,vnombre4,vnombre5,vnombre6,vnombre7:String):String**

Permite Mostrar una consulta en una ventana de selección de datos de otra base de datos y retornar el primer campo de dicha consulta.

- **TNS\_GE\_ACTDKDXCOMBO(vmatid,vdekardeid:String).**

Permite Insertar en la factura el detalle de insumos que forman parte del combo de un articulo. *Vmatid*: ID del articulo que es combo. *Vdekardeid*: ID del detalle de la factura sobre el cual se quiere insertar el detalle de insumos del articulo combo.

- **TNS\_GE\_GUSUARIO():String.**

Devuelve el ID del usuario que tiene la sesión del programa abierta.

- **TNS\_GE\_NOMUSUARIO():String.**

Devuelve el nombre del usuario que tiene la sesión del programa abierta.

- **TNS\_GE\_SELSTART(vcontrol:TEdit;vvalor:Integer).**

Ubica el cursor en un punto determinado de una caja de texto. *vvalor*: Es la posición dentro del control *vcontrol* donde se ubicará el cursor para empezar a editar.

- **TNS\_GE\_COPYCLIP(vtexto:String).**

Copia una cadena de texto en el portapapeles. Es similar al Ctrl+C.

- **TNS\_GE\_LEERCLIP():String.**

Devuelve la cadena de texto que está almacenada en el portapapeles. Es similar al Ctrl+V.

- **TNS\_GE\_ANOFIS():String**

Obtiene el año fiscal de la empresa

- **TNS\_GE\_PERMISO(voperacion:string):Boolean**

Verificar si el usuario actual tiene permiso para realizar la operación

- **TNS\_GE\_VALIDAUSU(vmensforma,vmensimp,vmensaje,vusuario:string):Boolean**

Validar usuario, los parámetros a enviar son:

*Vmensforma*: mensaje que despliega en la forma

*Vmensimp*: mensaje que despliega el groupbox

*Vmensaje*: mensaje sobre lo que no tiene permiso

*Vusuario*: usuario a validar

- **TNS\_GE\_COMPRES(movid,prefijo,numero:String)**

Agrupar las cuentas los parámetros a enviar son:

*Movid*: id del movimiento

*Prefijo*: opcional

*Numero*: opcional

- **TNS\_GE\_ExpoExcelSQLT(varea:Integer;vsql,Titulo:String)**

Ejecuta la consulta *vsql* en el área *varea* y la exporta a Excel con el título *Titulo* que desee, los parámetros a enviar son:

*Varea*: área

*Vsql*: consulta sql

*Titulo*: título que se desea colocar

- **TNS\_GE\_ExpoExcelT(varea:Integer;Titulo:String)**

Devuelve los datos del área en un libro de Excel con el título que desee

- **TNS\_GE\_AddLog(vlog:String)**

Añade al Log de Auditoría desde un evento personalizado.

- **TNS\_PR\_ExportarPDF(vFile:String):Boolean**

- **TNS\_PR\_REPORTEPDF(varchivo,vFile:String).**

Función para exportar a PDF sin vista previa. *Varchivo* es el reporte y *vfile* es la dirección donde se va a guardar el .pdf generado

- **TNS\_GE\_VERIFICLAVE(vusuario,vclave): Boolean**

Función para validar usuario y contraseña, si son reales devuelve Verdadero, de lo contrario devuelve Falso.

- **TNS\_GE\_ValorLetras(vvalor:Double;vmoneda:String='')**

Función para retornar el valor en letras de un número, con el segundo parámetro puede agregar el nombre de una moneda

- **TNS\_GE\_GNOMEMP():string**

Se obtiene el Nombre de la empresa

- **TNS\_GE\_GANOFIS:String**

Se obtiene el Año Actual de la Empresa

- **TNS\_GE\_GNITEMP()**

Obtiene el Nit de la Empresa actual

- **TNS\_SD\_DTF():doublé**

Abstrae el DTF de la fecha actual

## **FUNCIONES PARA MANEJO DE PUERTOS**

- **TNS\_GE\_AbrirCom(vport:integer,vbaudios,vparidad,vbitdatos:String):Boolean**

Esta función abre el puerto serial COM. La configuración del protocolo esta determinada por los parametros como el numero del puerto, la tasa en baudios, la paridad que utiliza y la cantidad de bits de datos que utiliza. A continuación esta los valores que se pueden enviar como parametros.

<b>Baudios</b>	<b>Paridad</b>	<b>Bits de Datos</b>
110 = 0	Ninguno = 0	4 bits = 0
330 = 1	Impar = 1	5 bits = 1
600 = 2	Par = 2	6 bits = 2
1200 = 3	Marca = 3	7 bits = 3
2400 = 4	Espacio = 4	8 bits = 4
4800 = 5		
9600 = 6		
14400 = 7		
19200 = 8		
38400 = 9		
56000 = 10		
57600 = 11		
115200 = 12		
128000 = 13		
256000 = 14		

Ejemplo: `TNS_GE_AbrirCom(1,6,0,3);`

- **TNS\_GE\_LeerCom : String**

Retorna un String con los datos que se encuentren en el puerto serial.

- **TNS\_GE\_EscribirCom(vtexto:String) : Boolean**

Escribe una cadena String en el puerto, el cual ya se ha abierto con anterioridad.

- **TNS\_GE\_LimpiarCom** Limpia el buffer de datos del puerto COM.
- **TNS\_GE\_CerrarCom** Cierra el puerto COM.
- **TNS\_GE\_ActivoCom():Boolean** Devuelve True o False si el Puerto está activo o no.

## **FUNCIONES PARA MANEJO DE ARCHIVOS PLANOS**

- **TNS\_GE\_PlanoCrear(vnombre:String)**

Creación de un archivo plano en la carpeta donde se está ejecutando la aplicación, el nombre del plano debe llevar la extensión, si existe éste se sobrescribe. Ej: Entrada.txt.

- **TNS\_GE\_PlanoCrear2(vnombre:String)**

Función similar a PlanoCrear pero sirve para crear otro archivo simultáneamente (*Archivo2*).

- **TNS\_GE\_PlanoAbrir(vnombre:String)**

Abre un archivo plano de la carpeta en donde se está ejecutando la aplicación, si no existe se genera un mensaje de error de archivo no encontrado (File No Found).

- **TNS\_GE\_PlanoAbrir2(vnombre:String)**

Función similar a PlanoAbrir y Abre el *Archivo2*.

- **TNS\_GE\_PlanoLeer():String**

Retorna una cadena tipo string, cada cadena es una línea del archivo plano.

- **TNS\_GE\_PlanoLeer2():String**

Función similar a PlanoLeer y Lee del *Archivo2*.

- **TNS\_GE\_PlanoEscribir(vcadena:String)**

Escribe una cadena de texto en el archivo plano, donde vcadena es el texto a escribir.

- **TNS\_GE\_PlanoEscribir2(vcadena:String)**

Función similar a PlanoEscribir y Escribe en el *Archivo2*.

- **TNS\_GE\_PlanoCerrar()** Cierra un archivo plano abierto con anterioridad.

- **TNS\_GE\_PlanoCerrar2()** Cierra el *Archivo2*.
- **TNS\_GE\_PlanoEof():boolean**  
Indica el fin de archivo de un plano y retorna true o false si ha llegado al final del archivo.
- **TNS\_GE\_PlanoEof2():boolean**  
Indica el fin de *archivo2* y retorna true o false si ha llegado al final.
- **TNS\_GE\_CampoPlano(cadena:string;numcampo:integer;separador:string):string.**

Devuelve el valor de un campo contenido dentro de una línea de caracteres de un archivo plano separado por algún carácter especial. *Cadena* Es la cadena de caracteres de donde se quiere extraer el valor de una columna. *Numcampo* Es el número de columna dentro de la cadena que se desea extraer. *Separador* es el carácter especial que separa cada campo de la cadena. Ejm: `tns_ge_CampoPlano("codigo,nombre,dirección,telefono",3,",")` devuelve "dirección".

- **TNS\_GE\_ExpoExcelSQL(varea:Integer;vsql:String)**

Ejecuta una consulta sobre la base de datos actual utilizando como recepción de la información el área **varea**. Devolviendo el resultado en un libro de excel.

- **TNS\_GE\_ExpoExcel(varea:Integer)**

Devuelve los datos del área en un libro de excel.

## **FUNCIONES DE IMPRESIÓN**

- **TNS\_PR\_IniciarImp():Bolean;**

Inicializa los archivos para impresión solo texto en impresoras de matriz de punto. Los parámetros de puerto de impresión y ruta de archivos temporales son los mismos definidos para el usuario ADMIN. desde los menús de Archivo-Configuración-Parámetros de usuario en cualquiera de los módulos de Visual TNS. Si no puede crear los archivos temporales retorna False.

- **TNS\_PR\_Enfa()**

Activa el modo negrilla para impresión solo texto.

- **TNS\_PR\_noenfa()**

Desactiva el modo negrilla para impresión solo texto.

- **TNS\_PR\_conden()**

Activa el modo condensado para impresión solo texto.

- **TNS\_PR\_normal()**

Desactiva el modo condensado para impresión solo texto.

- **TNS\_PR\_Extend()**

Activa el modo extendido para impresión solo texto.

- **TNS\_PR\_NoExtend()**

Desactiva el modo extendido para impresión solo texto.

- **TNS\_PR\_N(vfil,vcol:Integer;vtexto:String)**

Imprime una cadena de caracteres en la fila y columna especificada.

- **TNS\_PR\_FinImp()**

Cierra los archivos de impresión y ejecuta la orden de imprimir directamente a la ruta del puerto configurado para usuario ADMIN..

- **TNS\_PR\_SQLReporte(vselect:string);**

Ejecuta una consulta SQL sobre la base de datos actual y utiliza como parámetro la instrucción SQL a utilizar en el reporte.

Ejemplo: TNS\_PR\_SQLReporte('Select \* from material');

- **TNS\_PR\_Reporte(varchivo:String);**

Ejecuta un reporte (.FR3) y utiliza como parámetro el nombre del reporte el cual debe estar guardado en la Carpeta Visual TNS\Especificos\ o Visual TNS.

Ejemplo: TNS\_PR\_Reporte('Prueba.fr3');

Donde Prueba.fr3 es el reporte que se ha diseñado y que utiliza campos de la consulta realizada en TNS\_PR\_SQLReporte().

- **TNS\_PR\_ReporteVP(varchivo:String ;vistaprevia:Boolean);**

Ejecuta un reporte similar a la función TNS\_PR\_REPORTTE pero recibe como parámetro adicional si se quiere mostrar vista previa del reporte (true) o mandarlo directamente a impresora (false).

- **TNS\_PR\_PARREP(vnumpar:Integer;vvalor:String)**

Asigna un valor a un parámetro que se puede usar en los reportes gráficos diseñado en Fast Report 3. Existen disponibles 50 variables para ser utilizadas en el diseñador de reportes. Para usar la variable en el diseñador de reportes se coloca la etiqueta [**vparRep*n***] donde *n* es el número del parámetro a usar (1..50). Ejm: Para asignar un valor a la variable vparRep15 se escribe TNS\_PR\_ParRep(15,"hola").

- **TNS\_PR\_SQLReporteDBF(vselect,vruta:string)**

Ejecuta una consulta SQL sobre una tabla DBF ubicada en una dirección especificada en el parámetro *vruta* y la usa como base para imprimir un reporte en formato FR3.

Ejemplo: TNS\_PR\_SQLReporteDBF("Select \* from material","C:\TNS\ABC2007");

- **TNS\_PR\_ReporteDBF(varchivo:String);**

Realiza la misma función de tns\_pr\_reporte pero sobre la consulta hecha en la instrucción tns\_pr\_SqlReporteDbf.

- **TNS\_PR\_ReporteTabla(varchivo:String ; vistaprevia :Boolean);**

Muestra un reporte basado en una consulta hecha en el área 6 de ejecución de consultas de eventos personalizados. El parámetro *vistaprevia* indica si el reporte muestra vista previa o se imprime directamente sobre la impresora predeterminada.

Uno de los usos que se puede dar a esta función es ir armando información a partir de una consulta vacía, insertando, borrando o actualizando registros según las necesidades de los datos a mostrar. Se usa el área 6 ya que sobre esta área se pueden hacer insert, post, delete, etc. Es decir, se puede tratar como una tabla abierta para armar consultas complicadas que necesiten análisis y condiciones especiales al momento de armar la información que se necesita mostrar. Ejm:

Tns\_bd\_sql(6,'Select \* from plancuentas');

Tns\_pr\_reporte("prueba.fr3",False). Estas instrucciones imprimen directamente a impresora el plan de cuentas de la empresa.

## **FUNCIONES DE BASE DE DATOS**

- **TNS\_BD\_LOCKUP((valor,vtabla,vcampo,vretorna:String):String)**

Retorna el valor de un campo especificado en una tabla de la base de datos. Ejm:

TNS\_BD\_Lockup('001','terceros','nit','nombre'). Buscar en la tabla **terceros** el **nit "001"** y Retorna el **nombre**. Si no encuentra coincidencia retorna "".

- **TNS\_BD\_SQL(varea:Integer;vsq:String)**

Ejecuta una consulta sobre la base de datos actual utilizando como recepción de la información al área **varea**. Existen 10 áreas predefinidas para este fin, las áreas del 1 a 4 se usan para ejecutar consultas rápidas o que solo se necesiten recorrer para obtener información ejm Update, Delete, Select. Las áreas del 6 al 10 se usan para cargar datos y poderlas tratar como una tablas, es decir, se pueden adicionar índices, Insertar nuevos registros, borrar, se puede recorrer hacia delante o atrás, etc.

Ejm: TNS\_BD\_sql(1,'select \* from kardex where fecha="01/01/2005" order by numero'). Consulta el movimiento de kardex del día 01/01/2005 ordenado por número y almacena los registros retornados en el área 1.

- **TNS\_BD\_SQLEXECUTEBLOCK(varea:Integer;vsql,vparametros:String)**

Ejecuta un execute block, los parámetros a enviar son:

*Varea*: un área de la 1 a la 12

*Vsql*: sentencia **EXECUTE BLOCK**, Esta sentencia no es realmente una sentencia de PSQL aunque hace que un 'PSQL dinámico' esté disponible en una sentencia SELECT. Tiene el efecto de que un bloque de código de PSQL se pueda ejecutar en una sentencia de DSQL como si fuera un procedimiento almacenado.

*Vparametros*: valor opcional

- **TNS\_BD\_Eof(varea:Integer):Boolean**

Indica el fin de archivo de un área de trabajo y retorna True o False si está o no ubicado en el fin del archivo

- **TNS\_BD\_AREASELECCIONAR(varea:Integer;vcaption,vcampo1,vcampo2,vcampo3,vnombre1,vnombre2,vnombre3:String):String**

Selecciona un área

Los parámetros a ingresar son:

*Varea*: de la 6 a la 12

*Vcaption*: título de la ventana a mostrar

*vcampo1, vcampo2, vcampo3*: nombres de los campos de la tabla

*vnombre1,vnombre2,vnombre3*: titulo de los campos

- **TNS\_BD\_Next(varea:Integer)**

Ubica el puntero en el siguiente registro en el área de trabajo.

- **TNS\_BD\_Prior(varea:Integer)**

Ubica el puntero en el anterior registro en el área de trabajo.

- **TNS\_BD\_First(varea:Integer)**

Ubica el puntero en el primer registro en el área de trabajo.



- **TNS\_BD\_Last(varea:Integer)**

Ubica el puntero en el último registro en el área de trabajo.

- **TNS\_BD\_Insert(varea:Integer)**

Inserta un registro en la tabla cargada en el área *varea*. Esta área de estar especificada entre 6 y 9.

- **TNS\_BD\_Append(varea:Integer)**

Inserta un registro al final de la tabla cargada en el área *varea*. Esta área de estar especificada entre 6 y 9.

- **TNS\_BD\_Edit(varea:Integer)**

Edita un registro de la tabla cargada en el área *varea*. Esta área de estar especificada entre 6 y 9.

- **TNS\_BD\_Post(varea:Integer)**

Girada los cambios efectuados a un registro de la tabla cargada en el área *varea*. Esta área de estar especificada entre 6 y 9.

- **TNS\_BD\_Campo(varea:Integer;vcampo:String):String;**

Retorna el valor de un campo especificado en el área de trabajo y lo retorna como tipo Texto.

- **TNS\_BD\_CampoFloat(varea:Integer;vcampo:String):String;**

Retorna el valor de un campo especificado en el área de trabajo lo retorna como tipo Float.

- **TNS\_BD\_Locate(varea:Integer;vindice,vval1:String):Boolean;**

Ubica un registro dentro de la consulta cargada en el area *varea* que coincida con el valor *vval1* buscado por el índice *vindice*.

- **TNS\_BD\_Locate2(varea:Integer;vindice,vval1,vval2:String):Boolean;**

Similar a la función Locate, pero se puede ubicar el registro buscando por un índice compuesto de 2 campos Ejm: `tns_bd_Locate2(1,'departamento;ciudad','DEP01','CIU01')`.

En este ejemplo ubica un registro que tenga el departamento = 'DEP01' y ciudad ='CIU01'

- **TNS\_BD\_Locate3(varea:Integer;vindice,vval1,vval2,vval3:String):Boolean;**

Similar a la función Locate, pero se puede ubicar el registro buscando por un índice compuesto de 3 campos Ejm: `tns_bd_Locate3(1, 'departamento;ciudad;barrio', 'DEP01', 'CIU01','BAR01')`.

En este ejemplo ubica un registro que tenga el departamento = 'DEP01', ciudad = 'CIU01' y BARRIO = 'BAR01'

- **TNS\_BD\_Locate4(varea:Integer;vindice,vval1,vval2,vval3,vval4:String):Boolean**

Ubica un registro dentro de la consulta cargada en el area *varea* buscando por un índice compuesto de 4 campos

- **TNS\_BD\_Locate5(varea:Integer;vindice,vval1,vval2,vval3,vval4,vval5:String):Boolean**

Ubica un registro dentro de la consulta cargada en el area *varea* buscando por un índice compuesto de 5 campos

- **TNS\_BD\_AsignaCampo(varea:Integer;vcampo,vvalor:String);**

Asigna un valor tipo String a un campo de la tabla cargada en al área *varea*. Ejm:  
`Tns_bd_asignaCampo(1,"CODIGO","150721")` Asigna al campo *codigo* del área 1 la cadena "150721"

- **TNS\_BD\_IniciarTransaccion()**

Inicia una transacción en la base de datos

- **TNS\_BD\_CommitTransaccion()**

Termina la transacción y la hace efectiva

- **TNS\_BD\_RollbackTransaccion()**

Hubo un error durante la transacción y se devuelve al inicio de la transacción

- **TNS\_BD\_RecordCount(varea:Integer):Integer**

## CONEXION CON OTRA GDB

Estas funciones permiten conectarse a otra Base de Datos Firebird diferente a la actual. Los nombres de la funciones son similares a los de conexión con la Base de datos actual agregando el número 2.

- **TNS\_BD\_Conectar2(ruta:String):Boolean**

Permite conectar una base de datos con una ruta especificada

- **TNS\_BD\_sql2(varea:Integer;vsql:String)**
- **TNS\_BD\_eof2(varea:Integer):Boolean**
- **TNS\_BD\_Next2(varea:Integer)**
- **TNS\_BD\_Prior2(varea:Integer)**
- **TNS\_BD\_First2(varea:Integer)**
- **TNS\_BD\_Last2(varea:Integer)**
- **TNS\_BD\_Campo2(varea:Integer;vcampo:String):String**
- **TNS\_BD\_CampoFloat2(varea:Integer;vcampo:String):double**
- **TNS\_BD\_AsignaCampo2(varea:Integer;vcampo,vvalor:String)**
- **TNS\_BD\_Locate12(varea:Integer;vindice,vval1:String):Boolean**
- **TNS\_BD\_Locate22(varea:Integer;vindice,vval1,vval2:String):Boolean**
- **TNS\_BD\_Locate32(varea:Integer;vindice,vval1,vval2,vval3:String):Boolean**
- **TNS\_BD\_Append2(varea:Integer)**
- **TNS\_BD\_Insert2(varea:Integer)**
- **TNS\_BD\_Edit2(varea:Integer)**
- **TNS\_BD\_Post2(varea:Integer)**

## CONEXIÓN CON ODBC

Estas funciones permiten conectarse con cualquier tipo de base de datos a través de un Origen de Datos ODBC y hacer consultas, recorrer, borrar y actualizar los registros. Los nombres de las funciones y su operatividad son muy similares a los de conexión general de Base de datos descritos anteriormente y son:

- **TNS\_BD\_ConectarODBC(alias,cuenta,password:String):Boolean**
- **TNS\_BD\_CERRARODBC():Boolean**
- **TNS\_BD\_SQLODBC(varea:Integer,vselect:String)**
- **TNS\_BD\_EofODBC(varea:Integer):Boolean**
- **TNS\_BD\_NextODBC(varea:Integer)**
- **TNS\_BD\_PriorODBC(varea:Integer)**
- **TNS\_BD\_FirstODBC(varea:Integer)**
- **TNS\_BD\_LastODBC(varea:Integer)**
- **TNS\_BD\_CampoODBC(varea:Integer,vcampo:String):String**
- **TNS\_BD\_CampoFloatODBC(varea:Integer;vcampo:String):double**

## CONEXIÓN CON ARCHIVOS DBF

Estas funciones son similares a la vista anteriormente de conexión y son las siguientes:

- **TNS\_DBF\_Tabla(vnombre,vruta:String);**

Abre una tabla DBF con el nombre *vnombre* y el parámetro *vruta* es la ruta donde se encuentra la DBF.

Ejemplo: `TNS_DBF_TABLA('Material.DBF','C:\Visual\varios\');`

Las demás funciones son similares.

`TNS_DBF_Insert():Boolean`  
`TNS_DBF_Eof():Boolean`  
`TNS_DBF_Next;`  
`TNS_DBF_Post;`  
`TNS_BD_First;`  
`TNS_DBF_AsignaCampo(vnombre,vvalor:String);`  
`TNS_DBF_Cerrar;`

Para Realizar las mismas funciones de conexión y recorrido sobre archivos DBF pero sobre una consulta hecha a una DBF se usan las siguientes funciones:

`TNS_BD_SQLDBF(vselect,vruta:string)`  
`TNS_BD_eofDBF():Boolean`  
`TNS_BD_NextDBF()`  
`TNS_BD_PriorDBF()`  
`TNS_BD_FirstDBF()`  
`TNS_BD_LastDBF()`  
`TNS_BD_CampoDBF(vcampo:String):String`

## **ENVIIO DE ARCHIVOS VIA MAIL**

- **`TNS_ES_SendMail(vHost,vfromname,vfromaddress,vtoaddress,vattachments,vsubject,vbody:String):Boolean`**
- **`TNS_ES_SendMailSSL(vHost,vfromaddress,vsubject,vbody,vattachments,varvSuccess:String;vPort:Integer;vlogin:Boolean):Boolean`**

Permite enviar un correo, los parámetros a enviar son:

*vHost*: cuenta de correo de gmail que envía el mensaje

*vfromaddress*: correo a donde se envía el mensaje

*vsubject*: asunto

*vbody*: cuerpo del mensaje

*vattachments*: datos adjuntos

*var vSuccess*: opcional

*vPort*: Puerto que por lo general es el 587

*vlogin*: si deseo logear o no

- **`TNS_PR_ExportarPDF(vFile:String):Boolean`**

- **TNS\_ES\_VISUALREPORT(vGuia,vGenerado:String;vPrevi:Boolean):Boolean**
- **TNS\_VR\_AsignarVariable(const Parameters: array of variant)**
- **TNS\_VR\_MasterDetalle(const Parameters: array of variant)**

## **FUNCIONES IMPRESORA FISCAL BEMATECH**

TNS\_PR\_Bematech\_FI\_CambiaSimboloMoneda( SimboloMoneda: String ): Integer

TNS\_PR\_Bematech\_FI\_ProgramaAlicuota( Aliquota: String ; ICMS\_ISS: Integer ): Integer

TNS\_PR\_Bematech\_FI\_EspacioEntreLineas( Dots: Integer ): Integer

TNS\_PR\_Bematech\_FI\_AbreComprobanteDeVenta(RIF: String; Nombre: String) : Integer

TNS\_PR\_Bematech\_FI\_AbreComprobanteDeVentaEx(RIF: String; Nombre: String; Direccion: String) : Integer

TNS\_PR\_Bematech\_FI\_DevolucionArticulo(cCodigo: String; cDescripcion:String; cAlicuota:String; cTipoCantidad:String; cCantidad:String; iCasasDecimales : Integer; cValorUnit: String; cTipoDescuento:String; cValorDesc: String): Integer

TNS\_PR\_Bematech\_FI\_AbreNotaDeCredito(cNombre: String; cNumeroSerie: String; cRIF: String; cDia: String;cMes: String;cAno: String; cHora: String; cMinuto: String; cSecundo: String; cCOO: String): Integer

TNS\_PR\_Bematech\_FI\_VendeArticulo(Codigo: String ; Descripcion: String ; Alicuota: String ; TipoCantidad: String ; Cantidad: String ; CasasDecimales: Integer; ValorUnitario: String ; TipoDescuento: String ; Descuento: String ): Integer

TNS\_PR\_Bematech\_FI\_AnulaArticuloAnterior: Integer

TNS\_PR\_Bematech\_FI\_AnulaArticuloGenerico( NumeroItem: String ): Integer

TNS\_PR\_Bematech\_FI\_AnulaCupon: Integer

TNS\_PR\_Bematech\_FI\_CierraCuponReducido( FormaPago: String ; Mensaje: String ): Integer

TNS\_PR\_Bematech\_FI\_CierraCupon( FormaPago: String ; IncrementoDescuento: String ; TipoIncrementoDescuento: String ; ValorIncrementoDescuento: String ; ValorPago: String ; Mensaje: String ): Integer

TNS\_PR\_Bematech\_FI\_ResetaImpresora: Integer

TNS\_PR\_Bematech\_FI\_IniciaCierreCupon( IncrementoDescuento: String ;  
TipoincrementoDescuento: String ; ValorIncrementoDescuento: String ): Integer

TNS\_PR\_Bematech\_FI\_EfectuaFormaPago( FormaPago: String ; ValorFormaPago: String  
): Integer

TNS\_PR\_Bematech\_FI\_FinalizarCierreCupon( Mensaje: String ): Integer

TNS\_PR\_Bematech\_FI\_UsaUnidadMedida( UnidadMedida: String ): Integer

TNS\_PR\_Bematech\_FI\_ExtenderDescripcionArticulo( Descripcion: String ): Integer

TNS\_PR\_Bematech\_FI\_LecturaX: Integer

TNS\_PR\_Bematech\_FI\_ReducccionZ( Fecha: String ; Hora: String ): Integer

TNS\_PR\_Bematech\_FI\_NumeroSerie( NumeroSerie: String ): Integer

TNS\_PR\_Bematech\_FI\_SubTotal( SubTotal: String ): Integer

TNS\_PR\_Bematech\_FI\_NumeroCupon( NumeroCupon: String ): Integer

TNS\_PR\_Bematech\_FI\_GranTotal( GranTotal: String ): Integer

TNS\_PR\_Bematech\_FI\_ValorFormaPago( FormaPago: String ; Valor: String ): Integer

TNS\_PR\_Bematech\_FI\_AbrePuertaSerial: Integer

TNS\_PR\_Bematech\_FI\_CierraPuertaSerial: Integer

TNS\_PR\_Bematech\_FI\_VerificaImpresoraPrendida: Integer

**TNS\_GE\_verificlave(vusua,vclave:String):Boolean**

Verifica que la clave corresponda al usuario del sistema

**TNS\_GE\_ValorLetras(vvalor:Double;vmoneda:String='):String**

Devuelve el valor ingresado en letras, donde valor indica el valor a convertir en letra y vmoneda es un parámetro opcional

## **FUNCIONES IMPRESORA FISCAL BIXOLON**

TNS\_PR\_OPENFPCTRL(lpcstr: string):Boolean

TNS\_PR\_CLOSEFPCTRL(): boolean

TNS\_PR\_CHECKFPFPPRINTER(): boolean

TNS\_PR\_SENDCMD(cmd:string): boolean

## **FUNCIONES DE WEB SERVER**

**TNS\_WEB\_Returna(valor:String)** Esta función retorna la cadena de texto que mesclará con el archivo HTML de donde fue llamado el evento personalizado.

**TNS\_WEB\_ReturnaError(valor:String)**

**TNS\_WEB\_Parametro(vNombreParametro:String)** lee el valor del parámetro Web que se coloque en el archivo HTML para capturar variables.

**TNS\_WEB\_Root()** Retorna la ruta del servidor donde se ésta ejecutando el webserver.

**TNS\_WEB\_ReadFile(vArchivo:String)** Retorna el archivo leído en la variable vararchivo. Si es un .pdf la página debe ser extensión pdf.

**TNS\_ES\_VISUALREPORT (vGuia,vGenerado:String;vPrevi:Boolean);** Ejecuta Visual Report desde una página web.

**TNS\_GE\_VALIDAUSU (vmensforma,vmensimp,vmensaje,vusuario:string)**

Valida permisos y si no tiene pide autorización

vmensforma: El mensaje que muestra el formulario

vmensimp: El mensaje que muestra el groupbox

vmensaje: Mensaje de Autorización

vusuario: Para que Usuario: A= solo permite usuario ADMIN y pide autorización para cualquier otro; Con T: valida si el usuario tiene permiso

## **TABLA VARIOS**

**GNOMEMP:** Nombre de la Empresa como aparece registrada en Administración.

**GNITEMP:** Nit de la Empresa como aparece registrada en Administración.

**GREPRESEMP:** Nombre del Representante Legal de la Empresa como aparece registrada en Administración.