

SAIT Software Development Kit (SAIT SDK)

Oct 2003

Este documento contiene toda la información que usted como desarrollador necesita para realizar adecuaciones y funciones para mejorar la adaptabilidad de SAIT en su empresa o con su cliente

SAIT esta desarrollado usando la plataforma Visual Fox Pro 6.3

Por lo que usted debera contar con esta herramienta para poder desarrollar aplicaciones y adecuaciones a SAIT

FoxPro es un interprete por lo que las adecuaciones son facilmente realizables.

SAIT AutoRun

SAIT Automáticamente ejecuta todos los archivos .APP que se encuentren en el directorio de la empresa
Usted deberá hacer una aplicación en VFP6.3 para agregar menús o redefinir opciones del menú actual

Otra opción de añadir funcionalidad especial a SAIT es creando el programa AutoRun.Prg y compilando en la versión VFP 6.3. Sait automáticamente ejecutara el archivo AutoRun.Fxp (En las versiones 3 y 4, es necesario agrega el archivo DoAutoRun.App)

Configurando las ventas de compras y ventas

Las siguientes propiedades de _Screen pueden ser usadas en los modulos especiales para añadir funcionalidad a las ventanas de compras.

Propiedad	Descripción
ActTotCompras	Se manda llamar cada vez que se va a actualizar el total que se muestra en la pantalla
F7Compras	Se manda llamar al presionar F7
PreValidaCompras	Se manda llamar despues de dar clie en <Procesar> y antes de grabar el documento, debe dejar _Screen.IValidaCompra en .t. para que pase
SaveDocCompras	Se manda llamar despues de haber grabado la compra

Las siguientes propiedades de _Screen pueden ser usadas en los modulos especiales para anadir funcionalidad a las ventanas de ventas.

Propiedad	Se manda llamar cuando:
VentasInit	se crea la ventana de Ventas
VentasIntiDoc	cada vez que el usuario cambia de tipo de documento a elaborar
VentasNumCliValid	cada vez que el usuario cambia de cliente al elaborar un documento
VentasNumArtValid	cada vez que el usuario escribe un articulo distinto
VentasPrevalida	Actualizar _Screen.IValidaVenta en .t.
VentasSaveDoc	
ActTotVentas	Se manda llamar cada vez que se va a actualizar el total que se muestra en la pantalla
F7Ventas	Se manda llamar al presionar F7
PreValidaVentas	Se manda llamar despues de dar clie en <Procesar> y antes de grabar el documento, debe dejar _Screen.IValidaVenta en .t. para que pase
SaveDocVentas	Se manda llamar despues de haber grabado la venta

En VentasActTot, si se desea actualizar los valores de los importes totales, es necesario actualizar las siguientes variables:

```
m.Importe
m.Descuento
m.Impuesto1
m.Impuesto2
m.Retencion1
m.Retencion2
```

Esto debido a que antes de grabar esos campos en la tabla de Docum, se manda llamar el metodo para determinar los valores correspondientes según las partidas.

En los procedimientos de VentasActTot tienes que tener acceso a los diferentes controles de la forma de ventas

Control	Comentario
Total	
Subtotal	
Impuesto1	
Impuesto2	

Las propiedades son de tipo caracter, y contienen el comando a ejecutar.

Ejemplo:

NumMotor.Prg

```
Procedure NumMotor
  With _Screen
    .AddProperty('F7Compras'      , 'Do PideSer in NumMotor')
    .AddProperty('SaveDocCompras' , 'Do SaveSer in NumMotor')
  EndWith
Return
```

Para crear una clase especial e incluirla en la ventana de ventas

La clase debe de ser un contenedor

Y contener los siguientes metodos:

```
Consultar    Llamado al consultar un documento, para cargar los datos almacenados
Limpiar      Llamado cuando se termina una factura, para limpiar la clase y prepararla para la sig factura
Modificar    Llamado cuando se quiere modificar una cotizacion o pedido
Grabar       Llamado al grabar el documento (igual y se usa un procedimiento en VentasSaveDoc)
ModificarCliente Llamado cuando el usuario cambia al numero de cliente
ModificarTipoDoc Llamado cuando el usuario cambia el tipo de documento, pasa de factura a cotizacion
Enlazar      Llamado cuando el usuario quiere enlazar una factura
Validar      Llamada antes de grabar un documento, en el proceso de validacion
```

El contenedor debera cargarse en en el procedimiento VentasInit

Y llamarse OtrosDatos

Ademas cambiar la propiedad lOtrosDatos ← .f. para que sait sepa que debe mandar llamar los metodos de otros datos

Y se debe de cargar

```
Procedure AutoRun
  _Screen.VentasInit = 'Do VentasInit in AutoRun with thisform '
Return
```

Procedure VentasInit

LParameter oForma

* Agregar el contenedor a la forma

* El contenedor se debe de llamar: OtrosDatos
* Inicializar la variable: lOtrosDatos con .t. para que sait sepa que la debe de mandar llamar.
oForma.NewObject('OtrosDatos', <nombre de la clase>, <nombre de la librería en donde se almaceno>)
oForma.lOtrosDatos = .t.

* Si deseas que aparezca arriba del grid
oForma.OtrosDatos.TabIndex = oForma.Grid.TabIndex - 1
oForma.BajarControles()

Return

Para mandar llamar la ventana de agregar poliza

DoForm('AddPol', nModo, cTipoPol, cFolio, dFecha, cConc, aPoliza)

Parametros

nModo: 2 = Mostrar ventana precargando una poliza almacenada en aPoliza

cTipoPol: Tipo de poliza a agregar

cFolioPol: Folio o numero de poliza

dFecha: Fecha de la poliza

cConc: Concepto general de la poliza

aPoliza: Matriz conteniendo la poliza a precargar. Cada renglon debe tener el sig formato: (cCuenta, cCA, nImporte, cConcepto)

cCuenta: Numero de cuenta

cCA: 'C' si es cargo, 'A' si es Abono

nImpore: Importe del movimiento

cConcepto: Concepto del movimiento

Como agregar un menu

Codigo ejemplo para agregar un menu como un submenu de ventas

Procedure Crm

```

If .f.
    * Si se desea que el menu se agregue al menu principal
    Define Pad PadCRM of _MSysMenu Prompt "CR\<M" Key ALT-M
    On Pad PadCRM of _MSysMenu Activate PopUp CRM
Else
    * Si se desea que el menu se agregue como un submenu de ventas
    Define Bar 20 of Ventas Prompt 'CR\<M'
    On Bar 20 of Ventas Activate PopUp CRM
Endif

Define PopUp CRM Margin Relative Shadow

AddBar('CRM','A\<gendar actividad','Do DoForm with "CrmAddtarea" in Crm ','Ctrl-G')
AddBar('CRM','Consultar actividades pendientes', 'Do DoForm with "TareasConsulta" in Crm')
AddBar('CRM','\-',")
AddBar('CRM','Agregar \<actividad realizada','Do DoForm with "CrmAddAccion" in Crm','CTRL-A')
AddBar('CRM','Consultar actividades realizadas','Do DoForm with "AccionesConsulta" in Crm')
AddBar('CRM','\-',")
AddBar('CRM','\<Contactos','Do DoForm with "CatContactos" in crm')
AddBar('CRM','Configurar categorías','Do DoForm with "CRMConfCat" in crm')
AddBar('CRM','\<Reportes','Activate Popup CrmReportes')

DefineMenuReportes()
Set Procedure to Crm Additive

```

Return

Procedure DefineMenuReportes

```

Define Popup CrmReportes MARGIN RELATIVE SHADOW COLOR SCHEME 4
Use Reportes
Scan
    If Allt(TIPO)='CRM'
        AddBar('CrmReportes',Allt(Reportes.DESC), 'Do DoForm with "CRMReporte" in Crm')
    Endif
EndScan
Use
AddBar('CrmReportes','\-',")
AddBar('CrmReportes','Configurar reportes de CRM', 'Do Form ConfRep with "CRM" ')

```

Return

Si se desea agregar un menu al menu principal escribir las sig instrucciones

```

DEFINE PAD reGistro OF _MSYSMENU PROMPT "Yonque" KEY ALT+Y, "Yonque" ;
COLOR SCHEME 3
ON PAD reGistro OF _MSYSMENU ACTIVATE POPUP meNu1

```

Clases para desarrollo de formas en módulos especiales

Las siguientes clases ayudan al desarrollo de aplicaciones, se encuentran en la librería de clase: MslLib60.Vcx

Clase	
Busqueda	Permite hacer una ventana de búsqueda
LeeCve	Permite pedir una clave de catálogo, con opción a búsqueda
FormaMsl	Permite hacer una forma con sesión privada, cerrar con Escape y cerrar las tablas al salir
TextoFecha	Permite pedir una fecha en un TextBox

Texto DoDefault() en init

TextoFecha DoDefault() en init

CajaVerif1 (Cuando se use init y lostfocus, darle dodefault())

Para empezar a usar las clases es necesario, configurar algunos detalles en VFP

Agregar la librería de clases, para tenerla disponible en las formas

- En el menú de VFP ir a Tools / Options
- Aparece la ventana de Options
- Seleccionar la pestaña: Controls
- Activar la opción: Visual Class Libraries
- Haga clic en <Add>
- Seleccionar \Sistemas\MslLib60\MslLib60.Vcx
- Haga clic en <Set As Default>
- Haga clic en <OK>

Agregar el path de las librerías:

- En el menú de VFP ir a Tools / Options
- Aparece la ventana de Options
- Seleccionar la pestaña: Controls
- Activar la opción: Visual Class Libraries
- Haga clic en <Add>
- Seleccionar \Sistemas\MslLib60\MslLib60.Vcx
- Haga clic en <Set As Default>
- Haga clic en <OK>

Cuando creas una forma, usar el comando:

Create Form <nombre> as FormaMsl From MslLib60

Busqueda

Permite hacer una ventana de busqueda, en forma rapida y sencilla

Para usarla:

Create Form BuscaCli As Busqueda From MslLib60

Modifique las propiedades de la tabla

La ventana es de tipo modal y tiene su propia sesion de datos

Propiedades	Significado
cExprs	Expresiones a desplegar en cada columna de la lista. Ejemplo: NumCli,NomCli,Telefono
cAnchos	Ancho en pixeles, de las columnas Ejemplo: 100,200,50
cRetVal	Nombre del campo a regresar como resultado de la busqueda Ejemplo: NUMCLI
cSql	SQL a realizar para la busqueda Ejemplo: Select Clientes.NUMCLI,Clientes.NOMCLI,Clientes.TELEFONO From Clientes Where Upper(m.Val1) \$ Upper(NOMCLI+TELEFONO) Order By NOMCLI La variable m.Val1 contiene el texto que el usuario desea buscar Es necesario considerar mayusculas y minusculas
Caption	Titulo de la ventana
Label1.Caption	Titulo del texto de busqueda

CatMsl

Permite hacer una ventana de catalogo en forma sencilla.

Muestra un grid con los registros actuales de la base de datos y presenta botones para Agregar/ Modificar y Borrar registros

Ademas permite definir nuevas vistas o subqueries para la misma base de datos. Definiendo las columnas a desplegar y las restricciones de los registros a incluir.

Por default debe existir una vista de todos los registros de la base de datos asi como las propiedades de esa vista. Eso se especifica en las propiedades abajo.

Para usarla:

Create Form CatCli As CatMsl From CatMsl

Modifique las propiedades que se indican abajo

Propiedades	Significado
cAlias	Alias de la base de datos principal del catalogo
cAnchos	Anchos de las columnas a desplegar en la vista principal. Separados con
cCatName	Nombre del catalogo. Usado para diferenciarlo de otros catalogos. En la tabla de Cats.DBF se guardan todas las vistas de los diferentes catalogos.
cExprs	Campos a desplegar en la vista principal, separados con
cJoins	Joins o relaciones con otras tablas, se le agrega al SQL cuando se manda llamar el query
cKey0	Llave primaria de la base de datos. Sirve para poder enlazar las vistas de los subqueries con la base de datos principal
cTitulos	Titulos a desplegar en las columnas de la vista principal.
FormaBusqueda	Nombre de la forma a usar para realizar una búsqueda. Debe ser modal y regresar la clave
FormaCatalogo	Nombre de la forma a usar para hacer cambios en el catalogo. Se manda llamar cuando el usuario presiona los botones de: Agregar, Modificar y Eliminar Se manda 2 parametros: Accion (1=Alta, 2=Modificar, 3=Borrar) y # de registro a modificar o a borrar
FormaReporte	Nombre de la forma a usar para mandar imprimir reportes

Es necesario que en el metodo INIT de la forma, abras la base de datos del catalogo ordenada en base a la llave primaria. Por ejemplo:

```
If not OpenDbf('Clientes','NumCli')
    Return .f.
Endif
```

Se debera hacer UNA SOLA forma para pedir los datos.

LeeCve

Permite pedir una clave de catalogo con opcion a busqueda

Para usarla: agregue la clase a un formulario

Propiedades	Significado
cAlias	Alias donde se buscara la clave. Debe de estar activo el orden Ejemplo: Clientes
cExprNombre	Expresion que me devuelve el nombre a desplegar Ejemplo: Clientes.NOMCLI
cFormaBusqueda	Nombre de la forma que realizara la busqueda por nombre. Debe ser modal y regresar la clave seleccionada por el usuario Ejemplo: ListClie
nAnchoClave	Numero de caracteres que tiene de ancho la clave. Por default las claves se alinean a la derecha, rellendo con espacios
Clave.Value	Permite accesar el valor de la clave
Nombre.Value	Permite accesar el valor del nombre
SetValue (cValor)	Permite fijar el valor de la clase, y automaticamente hace la busqueda y llena el valor del nombre
DioEnter	Se activa cuando, el usuario presiona [Enter] en la clase. Ahi puede meter codigo de validacion

Clase TextoFecha

Permite leer una fecha en un text box

Para usarla:

Agregue el control a cualquier formaulario

Propiedades	Significado
dValue	Contiene el valor en formato fecha leído
Value	Contiene el valor en formato string de la fecha. Ejemplo: 15-Sep-2001
SetValue	Metodo que permite fijar el valor de la fecha. Ejemplo: ThisForm.FechaCompra.SetValue(Date())

Funciones internas de SAIT

Hay algunas funciones que hemos desarrollado y que puedes usar dentro de tus formas y modulos especiales, las hemos clasificado en las siguientes categorias

Grupo	Funciones
Interfase de usuario	GetYN, GetNY, InitPje, ActPje, ClosePje, AddBar
Manejo de fechas	StoD, MexDate, MexDate2, UsDate, SigMes, NomMes, cNomMes
Impresion de Reportes	SendRep, ModIFto, DoCmd, InitVar, Letra, CampoArea
Impresion en modo texto	DoblAnch, Elite,Pica,Octavos,Sextos, Bold,NoBold, Under,NoUnder, Cond, NoCond
Funciones varias	SubLin, SigDoc, ValProp
Almacenar variables	GetMsl, PutMsl, UserVal
Manejo de DBFs	OpenDbf, OpenExcl, DelAlias
Compresion de Archivos	MakeZip2, Unzip2, DynaZip
Enlace con otros programas	OpenExcel

Funciones Varias

SubLin (cLista, nItem [,cSeparador])

Permite obtener el elemento nItem, de una lista tipo string. Opcionalmente usted puede indicar el caracter usado como separador, Por default el separador es: ^

Permite el manejo de lista en forma agil y sencilla

Ejemplo:

```
? SubLin('Juan^Perez',2)
Perez
? SubLin('Ignacio,Elizabeth,Nicole', 3 , ',')
Nicole
For i=1 to 3
    ? SubLin('X,Y,Z', i)
Next
```

SigDoc (cFolioActual)

Regresa en formato string, el siguiente folio para ser usado en algun documento dependiendo del valor que se manda

Usado para incrementar el campo que representa un folio

Ejemplo:

```
? SigDoc('A105')
'A106'
? SigDoc('1054')
'1055'
```

ValProp (cMemo, cPropiedad)

Regresa el valor de una propiedad almacenada en un campo memo.

Por ejemplo en el campo Clientes.OTROSDATOS se pueden almacenar otra informacion adicional que la empresa requiera.

De tal forma que se guarda en un campo memo con el siguiente formato:

<VarName> = <Valor>

<VarName> = <Valor>

Por ejemplo el campo Clientes.OTROSDATOS puede contener:

```
COD=SI
Mensaje=Avisar al gerente cuando se presente el cliente. Quedo de pagar una factura muy atrasada.
Para imprimir y usar el valor de esas variables use la funcion ValPro
? ValProp( Clientes.OTROSDATOS, 'COD')
'SI'
? ValProp(Clientes.OTROSDATOS, "Mensaje")
'Avisar al gerente cuando se presente el cliente. Quedo de pagar una factura muy atrasada.'
```

Funciones de Impresion de Reportes

SendRep (cFormato [, cDest [, cScope]])

Presenta la ventana para mandar el formato a: Pantalla, Impresora, Disco, Email

Ademas permite modificar el formato a imprimir

Opcionalmente permite especificar el destino a donde se desea mandar el reporte

Asi como el rango de registros a imprimir, segun los indiados en fox: Rest, Next nRec, While <|Cond>, etc.

Si se trata de un formato modo texto, debe contar con el archivo Imprime.bat en directorio actual.

Ejemplos:

```
SendRep('Nota','LPT1')
SendRep('RepVentas')
SendRep('Factura','', 'Rest')
```

ModiFto (cFormato)

Activa el disenador de reportes de fox, editando el formato que se manda como parametro

DoCmd (cComando)

Permite ejecutar cualquier comando dentro de un formato, util cuando quieres abrir otras tablas, o hacer un proceso especial.

Siempre regresa: ''

Ejemplo:

```
DoCmd('Use Arts in 0 Order NumArt')
```

InitVar (cVarName, eExpresion)

Inicializa una variable como publica con el valor que se pasa como parametro

Letra (nImporte, cMoneda)

Regresa en forma de string, la cantidad con letra correspondiente al importe que se pasa como parametro. cMoneda: 'P' Pesos 'D' Dolares

Ejemplo:

```
Letra(Cxc.IMPORTE, Cxc.DIVISA)
```

LetBill(nImpore)

Regresa en forma de texto, la cantidad con letra correspondiente al importe que se pasa como parametro. No incluye los centavos. No incluye el nombre de la moneda. Solo la cantidad con letra. Puede servir para hacer funciones para incluir el nombre de otras monedas.

CampoArea (eKey, cAlias, cNomCampo)

Busca en un area cierta llave y regresa el campo que se pasa como parametro

Ejemplo:

```
CampoArea(Docum.NUMCLI, 'Clientes', 'NOMCLI')
```

Busca en Clientes la expresion Docum.NUMCLI y regresa el nombre del cliente

Funciones para almacenar variables

Estas funciones permite almacenar variables en 2 archivos:

Config.MSL que se guarda en el directorio de la empresa

Apolo.INI que se guarda en el directorio de windows de la estacion

Los parametros especificos a la estacion se guardan en Apolo.INI y las variables comunes para todas las PCs se guardan en Config.MSL

GetMsl (cVarName)

Regresa en formato string, el valor actual de una variable almacenada en el archivo Config.Msl

Ejemplo:

```
? GetMsl('Decimales_en_Precio')
```

PutMsl (cVarName, cValue)

Almacena una variable en el archivo Config.Msl

Regresa .t. si pudo grabarlo o .f. en caso contrario

Ejemplo:

```
PutMsl('Decimales_en_precio', '2')
```

UserVal (cVarName [,cValue])

Tiene 2 funciones:

- 1) Grabar una variable en C:\Windows\Apolo.Ini
- 2) Obtener el valor de una variable de C:\Windows\Apolo.Ini

Si no se pasa el parametro cValue, entonces la funcion regresa en formato tipo string, el valor de una variable almacenada en Apolo.INI

Si se pasan los 2 parametros, entonces la funcion, graba la variable que se pasa, en el archivo Apolo.INI. El valor almacenado es el del 2do parametro

Estas variables son utiles, cuando se trata de parametros especificos a la computadora del usuario.

Ejemplos:

```
? UserVal('BitMap')
UserVal('BitMap', 'C:\SAIT\FONDO.BMP')
```

Funciones para el manejo de DBFs

OpenDbf (cDbf [,cTag [,cAliasUsar]])

Permite abrir un archivo en modo compartido, con un índice predeterminado por cTag, y adicionalmente indicar bajo que Alias se desea identificar ese DBF

Regresa .t. si el DBF se pudo abrir, .f. si hubo algun problema

Si no se pasa el 3er parametro, entonces el nombre del archivo se considerara como Alias

Ejemplo:

```
If not OpenDbf('Cleintes','NumCli')
    Aviso('No se pudo abrir')
Endif
```

OpenExcl (cListaDbfs)

Permite abrir en modo exclusivo todos los archivos que se pasan como parametro

Regresa .t. si todos los archivos se pudieron abrir, .f. si hubo algun problema

Ejemplo:

```
If OpenExcl('Clientes,Arts, Vend')
    Do Process...
Endif
```

DelAlias (cAlias)

Borra el archivo asociado con un alias actual.

CUIDADO esta funcion debe usarse con mucho cuidado

BORRA el archivo DBF especificado en el alias que se pasa como parametro

Sirve para borrar DBFs temporales que se usan en alguna consulta.

NUNCA usarla sobre archivos de datos. DelAlias('Clientes') borrara completamente del disco el archivo de clientes

Ejemplo:

```
DelAlias('Temp')
```

Funciones de compresion de archivos

MakeZip2 (cZipFileToCreate, cFilesToInclude)

Crea un archivo zip, que contiene los archivos especificados

No muestra avances, solo genera el archivo

IMPORTANTE: Se debe incluir toda la ruta de todos los archivos

Ejemplo

```
MakeZip2( 'A:\Envio.Zip', 'C:\SAIT\CIA001\Docum.DBF C:\SAIT\CIA001\Docum.FPT C:\SAIT\CIA001\*.FR?')
```

DynaZip (cZipFileToCreate, cFilesToInclude)

Crea un archivo zip, que contiene los archivos especificados

Muestra el porcentaje de avance

IMPORTANTE: Se debe incluir toda la ruta de todos los archivos

Ejemplo

```
MakeZip2( 'A:\Envio.Zip', 'C:\SAIT\CIA001\Docum.DBF C:\SAIT\CIA001\Docum.FPT C:\SAIT\CIA001\*.FR?')
```

UnZip2 (cZipFileName)

Descomprime el archivo indicado en el directorio actual

IMPORTANTE: Debe especificar toda la ruta del archivo

Ejemplo:

```
Unzip2('A:\ENVIO.ZIP')
```


Funciones para la interfase de usuario

Aviso (cMensaje)

Presenta un mensaje al usuario con el boton de Aceptar/Ok con el signo de Aviso

Ejemplo:

```
Aviso('Proceso ha sido terminado')
```

Alerta (cMensaje)

Presenta un mensaje al usuario con el boton de Aceptar/Ok con el signo de Error

Ejemplo:

```
Alerta('Proceso ha sido terminado')
```

GetYN (cMensaje)

Presenta una ventana con el mensaje indicado, preguntando SI o NO, regresa .t. si el usuario presion SI, falso si el usuario presiono NO

El boton predeterminado es <SI>

Ejemplo:

```
If not GetYN("Desea continuar?")
```

```
Return
```

```
Endif
```

GetNY (cMensaje)

Presenta una ventana con el mensaje indicado, preguntando SI o NO, regresa .t. si el usuario presion SI, falso si el usuario presiono NO

El boton predeterminado es <NO>

Se debe usar para procesos delicados

Ejemplo:

```
If not GetNY("Desea borrar la informacion?")
```

```
Return
```

```
Endif
```

InitPje ([cTitulo])

Activa una ventana en donde se mostrara el porcentaje de avance de un proceso

ActPje (nPorcentaje)

Actualiza el porcentaje de avance en la barra

nPorcentaje debe ser un numero entre 0 y 1. Es decir que 0.25 mostrara un avance de 25% 1.00 mostrara un avance del 100%

ClosePje ()

Cierra la ventana donde se muestra el porcentaje de avance

Ejemplo:

```
Use clientes
```

```
InitPje('Procesando')
```

```
Scan
```

```
ActPje(Recno() / Reccount() )
```

```
Do Proceso
```

```
EndScan
```

```
ClosePje()
```

AddBar (cPopUpName, cPrompt, cComando, cKey)

Agrega una barra a un menu

Ejemplo:

```
AddBar('PuntoDeVen','Reloj checador','Do Form Reloj','CTRL-R')
```

```
AddBar('Inventario','Busqueda especial de articulos','Do BusMax in NumMotor')
```

Funciones para el manejo de fechas

StoD (cFecha)

Convierte una fecha en formato string aaaammdd a formato tipo fecha

Ejemplo:

? Stod('19701203')

03/12/1970

MexDate (dFecha)

Regresa la fecha pasada en formato string en el estilo: Lun.13/Sep/2003

MexDate2 (dFecha)

Regresa la fecha pasada en formato string en el estilo: 03/Dic/1970

MexDate3 (dFecha)

Regresa la fecha pasada en el formato string en el estilo: 03/Dic/70

UsDate (dFecha)

Regresa la fecha pasada en el formato string en el estilo: 03-Aug-1985

SigMes (cPeriodo)

Regresa el siguiente mes del periodo que se pasa

Ejemplo:

SigMes('200212') = '200301'

SigMes('200201') = '200202'

NomMes (dFecha)

Regresa el nombre del mes de la fecha pasada

Ejemplo:

? NomMes ({^2003/05/31 })

Mayo

cNomMes (nMes)

Regresa el nombre del numero de mes que se pasa

Ejemplo:

cNomMes(1) = 'Enero'

Funciones para impresión en modo texto

Estas funciones se usan en los formatos de modo texto, y son las secuencias de caracteres ESC que las impresoras Epson o compatibles, reconocen para cambiar el tipo de letra en modo texto.

En formatos modo grafico, NO se usan

Elite()

Regresa la cadena ESC para indicarle a la impresora que imprima en 12 ccp (caracteres por pulgada) cuando no esta condensado y 16 cpp cuando esta activado el condensado

Pica()

Regresa la cadena ESC para indicarle a la impresora que imprima a 10 ccp en modo normal o 17 cpp en modo condensado. Es el default en las impresoras.

Sextos()

Regresa la cadena ESC para indicarle a la impresora que imprima en 6 lineas por pulgada (6lpp) que es el default = 66 lineas en una hoja tamaño carta

Octavos()

Regresa la cadena ESC para indicarle a la impresora que imprima en 8 lpp = 88 lineas en una hoja tamaño carta

SetPgLen(nLineas)

Regresa la cadena ESC para indicarle a la impresora que se estara usando una hoja diferente que el tamaño carta. nLineas es el numero de lineas que mide tu hoja.

Ejemplo:

```
SetPgLen(33)
```

Le indica a la impresora que estas usando una hoja media carta, de esta forma no va a saltar tanto cuando termine de imprimir el reporte.

Cond()

Regresa la cadena ESC para indicarle a la impresora que active el tipo de letra condensado.

NoCond()

Regresa la cadena ESC para indicarle a la impresora que cancele el tipo de letra condensado y regrese al normal.

Under()

Regresa la cadena ESC para indicarla a la impresora que active el tipo de letra subrayado.

NoUnder()

Regresa la cadena ESC para indicarle a la impresora que cancele el tipo de letra subrayado.

Bold()

Regresa la cadena ESC para indicarle a la impresora que active el tipo de letra Bold

NoBold()

Regresa la cadena ESC para indicarle a la impresora que cancele el tipo de letra Bold

Al momento de grabar una compra Temp contiene las siguientes columnas

Campo	Descripción
Cant	Cantidad de articulos
Series	Contiene las series que el usuario capturo en forma de lista: '129292,129293,129294'

Funciones de enlace con otros programas

OpenExcel (cTit1, cTit2, cTitulos, cMascaras)

El area activa la pasa a excel. Todos los campos

cTit1 titulo 1 generalmente el nombre de la empresa

cTit2. Titulo 2, generalmente el nombre del reporte

cTitulos Es una lista con los titulos a colocar en cada columna. Usando el separador ^

cMascaras Es una lista con la mascara que debe usar excel. Usar el separador ^.

Ejemplos de mascarar:

9,999,999.99

C (Para campos tipo carácter)

D (Para campos tipo fecha)

Entrada

Procedure Entrada

Debes tener abiertos las tablas de:

Entradas
Minv
MultiAlm
Arts

cNumArt	Clave del Articulo
nCantEnt	Cantidad que entra
dFechaEnt,	Fecha de entrada
cTipoDoc	Tipo de documento que genera la entrada
cNumDoc	Folio o numero del documento que genera la entrada
cNumpar	Numero de la partida, del documento
cNumCli	Clave del cliente asociado con la entrada
cNumProv	Clave del proveedor asociado con la entrada
nPrecioMN	Precio unitario en moneda nacional del articulo
nCostoMN,	Costo unitario en moneda nacional del articulo
nCostoDls,	Costo unitario en dólares del articulo
nCostoProMN,	Costo unitario promedio en moneda nacional
cPedimento,	Numero de pedimento (si existe)
cFechaPed,	Fecha del pedimento (si existe)
cAduana,	Aduana del pedimento (si existe)
CLote	Lote de la articulo
DCaduc,	Fecha de caducidad
CSeries	Series de los articulos que entra

Procedure Salida

Antes de mandarla llamar, colocarte en el registro del articulo en arts.dbf

Si usas almacenes, se actualiza MultiAlm.EXISTENCIA (la existencia del articulo en ese almacen)

Afecta las capas de entrada

Parametro	Significado
cNumArt	
nCantSal	
dFechaSal	
cTipoDoc	
cNumDoc	
cNUmPar	
cNumCli	
cNumProv	
nPrecioMN	Precio unitario en MN, sin IVA
nUltCost	Ultimo costo que tiene el articulo: Arts.ULTCOSTO Unitario, en MN y sin IVA
cCapaIni	Se usa cuando se va a cancelar una entrada y se desea que la capa especificada, sea la afectada al momento de realizar la salida. Generalmente al cancelar compras.
cSeries	