

SDK de los sistemas

AdminPAQ[®]

CONTPAQⁱ
FACTURA ELECTRÓNICA

73320140115

INTRODUCCIÓN

Qué es un SDK

Software Development Kit (SDK) o kit de desarrollo de software. Es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema bastante concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, ordenadores, videoconsolas, sistemas operativos, etcétera.

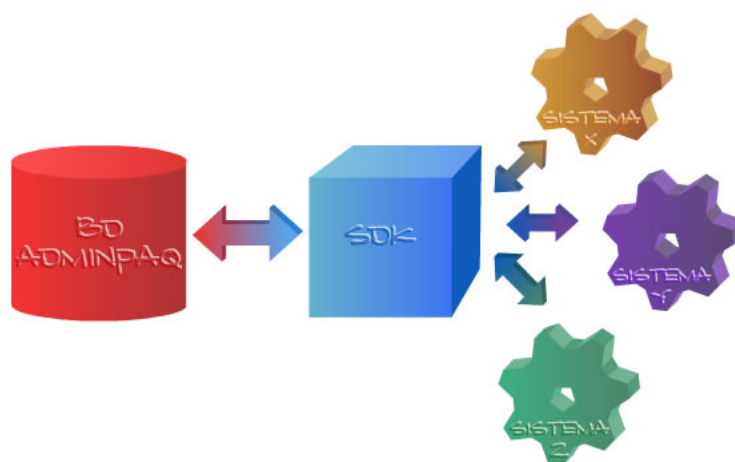
En el caso de **AdminPAQ** y **CONTPAQ i® FACTURA ELECTRÓNICA**, el **SDK** es un conjunto de archivos que contienen funciones publicadas, las cuales pueden ser usadas por desarrolladores externos para manipular (consultar o modificar) información de la base de datos de estos sistemas.

Cómo funciona

Las funciones disponibles en el **SDK** se comunican con **AdminPAQ** o **CONTPAQ i® FACTURA ELECTRÓNICA** a través de métodos de clases, estas a su vez hacen llamados a las clases "base" de **AdminPAQ** o **CONTPAQ i® FACTURA ELECTRÓNICA**, es decir, a las clases usadas dentro de **AdminPAQ** o **CONTPAQ i® FACTURA ELECTRÓNICA**.

El **SDK** controla la concurrencia en un ambiente multiusuario, es decir las funciones dan el soporte para los bloqueos y protegen los accesos. (Permite operar como si se tratara de una estación de **AdminPAQ** o **CONTPAQ i® FACTURA ELECTRÓNICA**).

Protege las bases de datos, sus relaciones y sigue las reglas de negocio de **AdminPAQ** o **CONTPAQ i® FACTURA ELECTRÓNICA**



Requerimientos para trabajar con el SDK

Ambiente

- **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA** instalado (monousuario o como estación).
- Entorno de programación. Editor/Compilador del lenguaje elegido (VB / Delphi / C / Plataforma .net, etc).
- Verifica contar con la licencia requerida por las funciones. Algunas funciones, como las de timbrado requieren licencias de un número de usuarios específico.
- Si estás programando en VBA (Excel) el SDK solo funciona en Office de 32 bits.

Archivos usados por el SDK.

Todos estos archivos son utilizados por el SDK:

Archivo	Descripción	Ubicación
MGW_SDK.dll	Es la interfase del SDK con AdminPAQ . Librería de encadenado, aquí se encuentran las funciones del SDK.	C:\Archivos de programa\Compacw\AdminPAQ
MGW_SDK.h	Encabezado que contiene la declaración de las funciones, constates y estructuras definidas en C++ Builder.	C:\Archivos de programa\Compacw\AdminPAQ\SDK
MGW_SDK.lib	Librería que contiene las funciones del SDK de AdminPAQ , generada con C++ Builder.	C:\Archivos de programa\Compacw\AdminPAQ\SDK
CAC000.dll CAC100.dll CAC200.dll CAC300.dll CAC400.dll CAC500.dll CAC600.dll CAC700.dll CAC800.dll	Modelo de componentes compuestos	C:\Archivos de programa\Compacw\AdminPAQ
CACc4dll.dll	Motor BDD	C:\Archivos de programa\Compacw\AdminPAQ
caccodbase.dll	Motor BDD	C:\Archivos de programa\Compacw\AdminPAQ
CACEsp.dll	Textos	C:\Archivos de programa\Compacw\AdminPAQ
Estandar.mgw	Textos/Dialogos/Mensajes	C:\Archivos de programa\Compacw\AdminPAQ
MGW000.dll MGW001.dll MGW002.dll	Reglas de negocio AdminPAQ	C:\Archivos de programa\Compacw\AdminPAQ
MgwEsp.dll	Textos	C:\Archivos de programa\Compacw\AdminPAQ
DBFWIN32.DLL BDEWIN.DLL	Funciones genéricas de lectura utilizada por los sistemas de Computación en Acción. En el caso de AdminPAQ se hace uso del DBFWIN32.DLL.	C:\Archivos de programa\Compacw\AdminPAQ

CONTPAQ i® FACTURA ELECTRÓNICA: Para el caso de este sistema los archivos se encuentran en la carpeta **Facturacion** en vez de la carpeta **AdminPAQ**.

Importante: Se debe tener especial cuidado con el control de versiones con el **SDK** en la que se desarrolla una aplicación y la versión de **AdminPAQ** con la que va a interactuar. Es decir, no se recomienda desarrollar una aplicación con el **SDK** de **AdminPAQ 6.0.0** para interactuar con un **AdminPAQ 7.0.0**.

RECOMENDACIONES Y CONSIDERACIONES IMPORTANTES

Tips y Conceptos Básicos

- Siempre ten en cuenta que las funciones del SDK están en **C++**, el objetivo al declarar las funciones en tu lenguaje es pasar los tipos de datos que C++ pueda recibir. Busca el tipo de datos en tu lenguaje que coincida mejor con el tipo de C++.
- En C++ todas las cadenas son de tipo **Char***, por lo que si en tu lenguaje de programación utilizas el tipo **String** estos siempre se deberán pasar **Por Valor**.
- Antes de hacer accesos mediante el **SDK**, asegurarse que **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA** funciona correctamente y que la información que está generando es correcta.
- Estar familiarizado con la estructura de la Base de Datos de **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA**.
- Tener claro y bien conceptualizado el fin y el alcance de la aplicación a desarrollar.
- Ir por "partes", es decir: Primero crear la conexión a la base de datos, inicializar el **SDK** y generar un documento desde la aplicación; posteriormente verificar que funciona correctamente (que se crea sin problemas el documento en **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA**).
- Modularizar el código (Si el entorno de programación lo permite). Esto es crear diversos módulos para separar funcionalidad global y local.

Ejemplo: Usar un módulo en el cual se realice la declaración de constantes, variables globales, estructuras de datos y enlace a las funciones del archivo **MGW_SDK.DLL**; y usar otro modulo para las funciones creadas por el desarrollador y que modificaran la información que se recibe y envía de la Base de Datos de **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA**.

Esto facilitará la portabilidad y la reutilización de código, así como el mantenimiento y actualización de la funcionalidad.

- Revisar que los documentos y sus movimientos se graban/actualizan de manera correcta en **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA**.
- Validar desde la aplicación que se desarrolla que los datos que se envían sea consistente y que tenga el formato correcto.
- Probar continuamente la aplicación con todas las posibles combinaciones que permita.

Funciones Obligatorias

Son las funciones que forzosamente deben incluirse en cualquier aplicación que use el SDK:

El método, a grandes rasgos, se compone de:

- Inicializar el SDK al inicio de cada proceso. **fInicializaSDK**.
Esta función se llama una sola vez al iniciar un proceso o acción completa.

Ejemplo: El alta de un documento y todos sus movimientos. Se inicia el SDK, se hace el llamado a todas las funciones requeridas y luego se termina el SDK.

- Funciones para abrir y cerrar empresa:
Se usan para indicar las bases de datos de la empresa a la cual afectará la aplicación que hace uso del SDK. (**fAbreEmpresa** / **fCierraEmpresa**)

Solo se puede trabajar en una empresa a la vez (a menos que se corran la misma aplicación dos veces).

- Incluir la función **fError** del SDK para recuperar la descripción de los posibles errores. La mayoría de las funciones regresan un código de error, donde 0 indica que no se presentaron errores y un número diferente de 0 cuando ocurrió algún error.

Se utiliza la función **fError** para recuperar la descripción de dicho error.

- Usar siempre la función **fTerminaSDK** para liberar todos los recursos solicitados por el SDK, al final de cada proceso completo. Esta función se llama una sola vez al finalizar un proceso o acción completa.

Estructura general de una aplicación desarrollada con el **SDK**.

Establecer el directorio del MGW_SDK

Inicializar SDK

Abrir Empresa

Tu función o proceso completo

Cerrar Empresa

Terminar SDK

Trabajando con documentos

Cuando se trabaje con documentos siempre se deben afectar.

Al crear documentos La existencia y los costos se afectan, sin embargo los acumulados del sistema no, por lo que es necesario después de crear documentos con sus movimientos correspondientes es necesario afectarlos.

En el SDK existen dos tipos de afectación, una para los documentos de cargo y abono y otra para los demás tipos de documento.

Estructura general de una aplicación que da de alta documentos y sus movimientos con el **SDK**.

Establecer el directorio del MGW_SDK

Inicializar SDK

Abrir Empresa

Alta de documento

Alta de movimientos

Afectar documento

Cerrar Empresa

Terminar SDK

Estructura general de una aplicación que da de alta documentos de Cargo y Abono con el **SDK**.

Establecer el directorio del MGW_SDK

Inicializar SDK

Abrir Empresa

Alta de documento Cargo/Abono

Afectar documento

Cerrar Empresa

Terminar SDK

Las funciones de afectación de documentos son:

fAfectaDocto_Param () y **fAfectaDocto ()**, bajo y alto nivel respectivamente.

Estructura general de un documento que maneja series y/o pedimentos

Establecer el directorio del MGW_SDK

Inicializar SDK

Abrir Empresa

Alta de documento

Alta de movimientos

Alta del movimiento con series o pedimentos

Calcula los movimientos con series o pedimentos

Afectar documento

Cerrar Empresa

Terminar SDK

Trabajando con productos o clientes

Estructura general para dar de alta productos o clientes.

Establecer el directorio del MGW_SDK
Inicializar SDK
 Abrir Empresa
 Alta de producto o cliente
 Cerrar Empresa
Terminar SDK

Estructura general para editar productos.

Establecer el directorio del MGW_SDK
Inicializar SDK
 Abrir Empresa
 Busca el producto o cliente
 Edita el producto o cliente
 Modifica el campo del producto o el cliente
 Guarda el producto o cliente
 Cerrar Empresa
Terminar SDK

Timbrar documentos

Estructura general para crear un documento y timbrarlo.

Establecer el directorio del MGW_SDK
Inicializar SDK
 Abrir Empresa
 Alta de documento
 Alta de movimientos
 Afectar documento

 Inicializa información de la licencia
 Emite/Timbra el documento
 Entrega el documento
 Cerrar Empresa
Terminar SDK

Estructura general para timbrar un XML creado por un tercero.

Establecer el directorio del MGW_SDK
Inicializar SDK
 Abrir Empresa
 Inicializa información de la licencia
 Timbra el XML
 Cerrar Empresa
Terminar SDK

Cuándo usar funciones de Alto Nivel y cuando de Bajo Nivel

En términos generales se recomienda usar las funciones de alto nivel debido a que estas realizan todo los procesos necesarios para mantener las reglas de negocio y la base de datos estable.

Cualquier lenguaje de programación que soporte estructuras de datos podrá hacer uso de las funciones de alto nivel, la razón es que como generalidad las funciones de alto nivel efectúan operaciones con registros completos.

Las funciones de bajo nivel permiten más flexibilidad en cuanto que datos se graban en la base de datos, pero implican más trabajo, por realizar escritura campo por campo, y complejidad pues se tienen que validar diversos puntos para no romper las reglas de negocio, por lo que para su uso se requiere mas precisión al desarrollar el proceso.

Estas funciones se pueden usar en cualquier lenguaje de programación, más son de carácter obligatorio en aquellos que no manejen estructuras de datos. Por ejemplo **Visual FoxPro**.

Ejemplo: Dar de alta de datos extras del catálogo sólo se puede efectuar con las funciones de "bajo nivel"

Algunos lenguajes como **Visual FoxPro** no soportan el uso de estructuras de datos, por lo que forzosamente se deben usar las funciones de bajo nivel.

Restricciones al usar funciones de Bajo Nivel

Las funciones de bajo nivel permiten la escritura campo a campo en la BD de **AdminPAQ**, sin embargo existen campos que no pueden ser modificadas por dichas funciones pues son valores que calcula o modifica **AdminPAQ o CONTPAQ i® FACTURA ELECTRÓNICA**.

Campo	Razón
cIdDocumento	Es un dato autogenerado.
cIdDocumentoDe	Depende de la plantilla del documento.
cIdConcepto	Es un dato autogenerado.
cIdCteProv	Es un dato autogenerado.
cIdAgente	Es un dato autogenerado.
cIdConcepto	Es un dato autogenerado.
cNeto	Es un campo calculado.
cTotal	Es un campo calculado.
cAfectado	Es un campo protegido.
cNaturaleza	Es un dato autogenerado.
cDocumentoOrigen	Es un dato autogenerado.
cPlantillaCUsaProveedor	Es un campo calculado.
cUsaCliente	Es un dato autogenerado.
cNetoCTotalUnidades	Es un campo calculado.
cBanObservaciones	Es un dato autogenerado.
cBanDatosEnvio	Es un dato autogenerado.
cBanCondCredito	Es un dato autogenerado.
CUnidadesPendientes	Es un campo calculado.
cTimeStamp	Es un dato autogenerado.

Recomendaciones para el manejo de cadenas

La forma en que cada lenguaje de programación define los tipos de datos cadena es varía entre lenguajes (en cuanto a su tamaño en bytes). Por esta razón los tipos de datos manejados por distintos lenguajes pueden presentar problemas al pasar información al SDK. En C++ Builder y Delphi este inconveniente no se presenta.

Al usar el SDK en Visual Basic. Para llenar los campos cadena que forman parte de la estructura, es necesario llenar con espacios en blanco las variables tipo cadena hasta alcanzar la longitud requerida por el SDK, por la diferencia que existe con este lenguaje al manejar los tipos de datos.

El error que se produce cuando no se llenan adecuadamente las estructuras es "código no existe". Para contrarrestar este error se usan dos funciones de manipulación de cadenas.

La función para llenar espacios en Visual Basic es la siguiente:

```
' Función f_RellenaConBlancos
Public Sub f_RellenaConBlancos(ByRef aCadena As String, aTamanio As Integer)
    Dim lEspacios As String
    Dim lTamanio As Long

    lEspacios = Space(aTamanio)
    lTamanio = aTamanio - Len(Trim(aCadena)) - 1
    aCadena = Trim(aCadena) & Left(lEspacios, lTamanio) & Chr(0)
End Sub
```

Para realizar comparaciones dentro de VB es necesario quitar el carácter nulo.

```
' Funcion Quita Caracteres nulos
Public Function f_QuitaCaracterNulo(aCadena As String) As String
    Dim lCadenaTemp As String

    lCadenaTemp = Replace(aCadena, Chr(0), "")
    if_QuitaCaracterNulo = Trim(lCadenaTemp)
End Function
```

FUNCIONES GENERALES

Inicialización / Terminación

fnicializaSDK ()

Disponibilidad AdminPAQ 2002.

Sintaxis fnicializaSDK()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Inicializa el SDK de AdminPAQ. Se requiere llamar esta función al inicio de cualquier aplicación que utilice el SDK.

Establece la conexión entre la aplicación desarrollada y la Base de datos de AdminPAQ. Su uso es obligatorio.

Ejemplo El siguiente código inicializa el SDK de AdminPAQ y asigna el resultado a una variable entera que se evalúa posteriormente; si su valor es distinto de 0 (cero) la aplicación se detiene.

```

IError = fnicializaSDK()

If IError <> 0 Then
    MensajeError IError
End
End If
    
```

Ver también [fTerminaSDK \(\)](#), [fError\(\)](#)

fTerminaSDK ()

Disponibilidad AdminPAQ 2002.

Sintaxis fTerminaSDK ()

Parámetros No usa.

Retorna No tiene valor de retorno.

Descripción Libera todos los recursos solicitados por el SDK, se requiere llamar al terminar de utilizar el SDK.

Ejemplo El siguiente código termina el SDK de AdminPAQ.

```
fTerminaSDK()
```

Ver también [fnicializaSDK \(\)](#), [fError\(\)](#)

fSetNombrePAQ ()

Disponibilidad AdminPAQ 2.2.0. CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0

Sintaxis fSetNombrePAQ(*aSistema*)

Parámetros	Nombre <i>aSistema</i>	Tipo Cadena	Uso Por referencia	Descripción
				Nombre del sistema al que se conectará el SDK. Para establecer una conexión a CONTPAQ i® FACTURA ELECTRÓNICA este parámetro deberá ser igual a " CONTPAQ I Facturacion ".
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función define el sistema al que se conectará el SDK. Sino se usa esta función la conexión por omisión será al sistema AdminPAQ . Si se desea establecer una conexión a CONTPAQ i® FACTURA ELECTRÓNICA el parámetro <i>aSistema</i> deberá ser CONTPAQ I Facturacion y se deberá utilizar en vez de la función <code>fInicializaSDK()</code> .			
Ejemplo	El siguiente código crea una conexión al SDK de CONTPAQ i® FACTURA ELECTRÓNICA .			

```
Public Const kFACTURACION_I = "CONTPAQ I Facturacion"
Call fSetNombrePAQ(kFACTURACION_I)
```

Ver también [fInicializaSDK \(\)](#), [fTerminaSDK \(\)](#)

Manejo de errores

fError ()

Disponibilidad AdminPAQ 2002.

Sintaxis fError(*aNumError*, *aMensaje*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aNumError</i>	Entero	Por valor	Número del error.
	<i>aMensaje</i>	Cadena	Por referencia	Descripción del error.
	<i>aLen</i>	Entero	Por valor	Longitud del mensaje de error.

Retorna *aMensaje*: Al finalizar la función este parámetro contiene el mensaje de error correspondiente al número de error especificado en *aNumError*.

Descripción Esta función recupera el mensaje de error del SDK.

Ejemplo El siguiente código asigna a la variable **lError** el resultado de la función **finicializaSDK()**, en caso de que suceda algún error (valor distinto de 0), la función **fError** se ejecuta obteniendo el mensaje correspondiente al número de error enviado, mostrando una longitud de mensaje de 350 caracteres.

```
lError = finicializaSDK()

If lError <> 0 Then
    fError lError, lMensaje, 350
End
End If
```

Ver también [finicializaSDK \(\)](#), [fTerminaSDK \(\)](#)

FUNCIONES DE EMPRESAS

Navegación

fPosPrimerEmpresa ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosPrimerEmpresa(*aldEmpresa*, *aNombreEmpresa*, *aDirectorioEmpresa*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldEmpresa</i>	Entero	Por referencia	Identificador de la empresa.
	<i>aNombreEmpresa</i>	Cadena	Por referencia	Nombre de la empresa.
	<i>aDirectorioEmpresa</i>	Cadena	Por referencia	Directorio de la empresa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aldEmpresa: Al finalizar la función este parámetro contiene el identificador de la primera empresa registrada en la Base de Datos.

aNombreEmpresa: Al finalizar la función este parámetro contiene el nombre de la primera empresa registrada en la Base de Datos.

aDirectorioEmpresa: Al finalizar la función este parámetro contiene el directorio de la primera empresa registrada en la base de datos.

Descripción Esta función se posiciona en el primer registro de la base de datos de empresas de AdminPAQ, modifica los parámetros *aNombreEmpresa* y *aDirectorioEmpresa*, en los cuales guarda el nombre de la primera empresa y su ruta, correspondientemente.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de empresas de la base de datos de AdminPAQ.

```
fPosPrimerEmpresa(lIdEmpresa, lNombreEmpresa, lDirectorioEmpresa)
```

Ver también [fPosSiguienteEmpresa \(\)](#), [fAbreEmpresa \(\)](#), [fCierraEmpresa \(\)](#)

fPosSiguienteEmpresa ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosSiguienteEmpresa (*aldEmpresa*, *aNombreEmpresa*, *aDirectorioEmpresa*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldEmpresa</i>	Entero	Por referencia	Identificador de la empresa.
	<i>aNombreEmpresa</i>	Cadena	Por referencia	Nombre de la empresa.
	<i>aDirectorioEmpresa</i>	Cadena	Por referencia	Directorio de la empresa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aldEmpresa: Al finalizar la función este parámetro contiene el identificador de la siguiente empresa registrada en la Base de Datos.

aNombreEmpresa: Al finalizar la función este parámetro contiene el nombre de la siguiente empresa registrada en la base de datos.

aDirectorioEmpresa: Al finalizar la función este parámetro contiene el directorio de la siguiente empresa registrada en la base de datos.

Descripción Esta función avanza al siguiente registro en la tabla de Empresas de AdminPAQ; en caso de que no exista un siguiente registro, la función retorna un valor distinto de 0 (cero).

Ejemplo El siguiente código termina el SDK de AdminPAQ.

```
fPosSiguienteEmpresa (lIdEmpresa, lNombreEmpresa, lDirectorioEmpresa )
```

Ver también [fPosPrimerEmpresa \(\)](#), [fAbreEmpresa \(\)](#), [fCierraEmpresa \(\)](#)

Apertura / Cierre

fAbreEmpresa ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAbreEmpresa (*aDirectorioEmpresa*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aDirectorioEmpresa</i>	Cadena	Por Referencia	Directorio de la empresa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función abre la empresa que corresponde a la ruta especificada en el parámetro *aDirectorioEmpresa*.

Ejemplo El siguiente código indica a la aplicación que abra la empresa ubicada en la ruta C:\Compacw\Empresas\EmpresaEjemplo.

```
IDirectorioEmpresa = "C:\Compacw\Empresas\EmpresaEjemplo"
fAbreEmpresa (IDirectorioEmpresa)
```

Ver también [fPosPrimerEmpresa \(\)](#), [fPosSiguienteEmpresa \(\)](#), [fCierraEmpresa \(\)](#), [fInicializaSDK \(\)](#), [fTerminaSDK \(\)](#)

fCierraEmpresa ()

Disponibilidad AdminPAQ 2002.

Sintaxis fCierraEmpresa ()

Parámetros No usa.

Retorna No tiene valor de retorno.

Descripción Cierra la conexión con la empresa activa en la aplicación que usa el SDK.

Ejemplo El siguiente código cierra la empresa activa.

```
fCierraEmpresa()
```

Ver también [fPosPrimerEmpresa \(\)](#), [fPosSiguienteEmpresa \(\)](#), [fAbreEmpresa \(\)](#), [fInicializaSDK \(\)](#), [fTerminaSDK \(\)](#)

FUNCIONES DE DOCUMENTOS

Bajo Nivel – Lectura/Escritura

fnInsertarDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fnInsertarDocumento ()

Parámetros No usa.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Adiciona un nuevo registro en la tabla de Documentos en modo de inserción.

Ejemplo El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Documentos.

```
fnInsertarDocumento ()
```

Ver también [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fAfectaDocto_Param \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fBuscaDocumento \(\)](#)

fEditarDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fEditarDocumento ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Activa el modo de Edición de un registro en la tabla de Documentos.

Ejemplo El siguiente código busca un documento por su llave, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
IError = fBuscaDocumento(ILlaveDocto )
If IError <> 0 Then
    MensajeError IError
Else
    fEditarDocumento ()
End If
```

Ver también [fInsertarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#),
[fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fSetDatoDocumento \(\)](#),
[fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#),
[fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAfectaDocto \(\)](#),
[fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#),
[fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fGuardaDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fGuardaDocumento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Guarda los cambios realizados a un documento.

Ejemplo El siguiente código indica a la aplicación que guarde los cambios al documento activo. Esta función se llama después de que se utiliza la función fInsertarDocumento() o fEditarDocumento y se graban los valores en los campos correspondientes.

```
fGuardaDocumento ()
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fBorraDocumento\(\)](#), [fCancelaDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto_Param \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fCancelarModificacionDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fCancelarModificacionDocumento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela las modificaciones al registro actual de documentos. El registro debe estar en modo de edición o inserción.

Ejemplo El siguiente código indica a la aplicación que cancele la modificación a un registro de documentos que estaba en modo de inserción o edición.

```
fCancelarModificacionDocumento ()
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fBuscarDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fBorraDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBorraDocumento ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Borra un registro en la tabla de Documentos.

Ejemplo El siguiente código busca un documento por su llave, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```
IError = fBuscaDocumento(ILlaveDocto )  
If IError <> 0 Then  
    MensajeError IError  
Else  
    fBorraDocumento ()  
End If
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#),
[fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fSetDatoDocumento \(\)](#),
[fLeeDatoDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscaDocumento \(\)](#)

fCancelaDocumento ()

Disponibilidad AdminPAQ 2006 v. 3.2.2.

Sintaxis fCancelaDocumento ()

Parámetros No usa.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela documentos de AdminPAQ.

Ejemplo El siguiente código indica a la aplicación que cancele la modificación a un registro de documentos que estaba en modo de inserción o edición.

```
IError = fBuscaDocumento(ILlaveDocto )  
If IError <> 0 Then  
    MensajeError IError  
Else  
    fCancelaDocumento ()  
End If
```

Ver también [fCancelaDocumento \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fBorraDocumento_CW ()

Disponibilidad AdminPAQ 2005.

Sintaxis fBorraDocumento_CW ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Borra un documento de AdminPAQ y si este estuviera contabilización también borra la póliza correspondiente en ContPAQ.

Ejemplo El siguiente código busca un documento por su llave, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente.

```
IError = fBuscaDocumento(ILlaveDocto )  
If IError <> 0 Then  
    MensajeError IError  
Else  
    fBorraDocumento_CW ()  
End If
```

Ver también [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fCancelaDocumento_CW ()

Disponibilidad AdminPAQ 2006 v. 3.2.2.

Sintaxis fCancelaDocumento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela un documento de AdminPAQ y borra la poliza correspondiente en ContPAQ.

Ejemplo El siguiente código busca un documento por su llave, si lo encuentra lo cancela, en caso contrario envía el mensaje de error correspondiente.

```
IError = fBuscaDocumento(ILlaveDocto )
If IError <> 0 Then
    MensajeError IError
Else
    fCancelaDocumento ()
End If
```

Ver también [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fAfectaDocto_Param ()

Disponibilidad AdminPAQ 2005.

Sintaxis fAfectaDocto (aCodConcepto, aSerie, aFolio, aAfecta)

Parámetros	Nombre	Tipo	Uso	Descripción
	aCodConcepto	Cadena	Por valor	Código del concepto del documento.
	aSerie	Cadena	Por valor	Serie del documento
	aFolio	Doble	Por valor	Folio del documento
	aAfecta	Lógico (Bool)	Por valor	Verdadero o falso. Afectar o desafectar.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función utiliza aCodConcepto, aSerie, y aFolio como llave del documento y aAfecta para afectar o desafectarlo.

Ejemplo El siguiente código indica a la aplicación que de afecte un documento, en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fAfectaDocto (ICodConcepto, ISerie, IFolio, IAfecta)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fCancelaDocumento \(\)](#), [fCancelaDocumento_CW \(\)](#), [fBuscarDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fSaldarDocumento_Param ()

Disponibilidad AdminPAQ 2005.

Sintaxis fSaldarDocumento_Param (*aCodConcepto_Pagar, aSerie_Pagar, aFolio_Pagar, aCodConcepto_Pago, aSerie_Pago, aFolio_Pago, alImporte, aldMoneda, aFecha*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto_Pagar</i>	Cadena	Por valor	Código del concepto del documento a pagar.
	<i>aSerie_Pagar</i>	Cadena	Por valor	Serie del documento a pagar.
	<i>aFolio_Pagar</i>	Doble	Por valor	Folio del documento a pagar.
	<i>aCodConcepto_Pago</i>	Cadena	Por valor	Código del concepto del documento que paga.
	<i>aSerie_Pago</i>	Cadena	Por valor	Serie del documento que paga.
	<i>aFolio_Pago</i>	Cadena	Por valor	Folio del documento que paga.
	<i>alImporte</i>	Doble	Por valor	Importe del pago.
	<i>aldMoneda</i>	Entero	Por valor	Moneda del pago.
	<i>aFecha</i>	Cadena	Por valor	Fecha del pago.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asocia documentos y salda sus importes.

Ejemplo El siguiente código indica a la aplicación que de salde un documento con la información enviada, en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fSaldarDocumento_Param (ICodConcepto_Pagar, ISerie_Pagar, IFolio_Pagar
ICodConcepto_Pago, ISerie_Pago, IFolio_Pago, IImporte, IldMoneda, IFecha)
If IError <> 0 Then
    MensajeError IError
Else

```

Ver también [fBorrarAsociacion_Param \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fBorrarAsociacion_Param ()

Disponibilidad AdminPAQ 2005.

Sintaxis fBorrarAsociacion (*aCodConcepto_Pagar, aSerie_Pagar, aFolio_Pagar*
aCodConcepto_Pago, aSerie_Pago, aFolio_Pago)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto_Pagar</i>	Cadena	Por valor	Código del concepto del documento pagado.
	<i>aSerie_Pagar</i>	Cadena	Por valor	Serie del documento pagado.
	<i>aFolio_Pagar</i>	Double	Por valor	Folio del documento pagado.
	<i>aCodConcepto_Pago</i>	Cadena	Por valor	Código del concepto del documento que pagó.
	<i>aSerie_Pago</i>	Cadena	Por valor	Serie del documento que pagó..
	<i>aFolio_Pago</i>	Double	Por valor	Folio del documento que pagó.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función la asociación de documentos.

Ejemplo El siguiente código indica a la aplicación que elimine la asociación entre el documento pagado y el que pagó ; en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fBorrarAsociacion (ICodConcepto_Pagar, ISerie_Pagar, IFolio_Pagar
ICodConcepto_Pago, ISerie_Pago, IFolio_Pago)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fBuscarDocumento \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fSetDatoDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetDatoDocumento (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por Valor	Campo destino
	<i>aValor</i>	Cadena	Por Valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que busque un documento por su llave, en caso de que lo encuentre escribe el contenido de la variable lFecha en el campo cFecha de la tabla de documentos; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaDocumento(lLlaveDocto )
If lError <> 0 Then
  MensajeError lError
Else
  lError = fSetDatoDocumento("cFecha", lFecha)
End If
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fBuscaDocumento \(\)](#)

fLeeDatoDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fLeeDatoDocumento (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por referencia	Valor de escritura
	<i>alen</i>	Entero	Por valor	Longitud del dato de lectura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que busque un documento por su llave, en caso de que lo encuentre lee el contenido del campo cFecha de la tabla de documentos y lo asigna a la variable lFecha; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaDocumento(lLlaveDocto )
If lError <> 0 Then
  MensajeError lError
Else
  lError = fLeeDatoDocumento("cFecha", lFecha, lLen)
End If
```

Ver también [fSetDatoDocumento \(\)](#), [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBuscaDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fSaldarDocumento_Param \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#)

fSiguienteFolio ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSiguienteFolio(*aCodigoConcepto*, *aSerie*, *aFolio*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoConcepto</i>	Cadena	Por valor	Código del concepto del documento.
	<i>aSerie</i>	Cadena	Por referencia	Serie del documento
	<i>aFolio</i>	Doble	Por referencia	Folio del documento

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aSerie: Al finalizar la función este parámetro contiene el valor de la serie del documento especificado.

aFolio: Al finalizar la función este parámetro contiene el siguiente folio del documento especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que obtenga

```
lError = fSiguienteFolio(lCodigoConcepto, lSerie, lFolio )
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fBuscaDocumento \(\)](#)

fSetFiltroDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetFiltroDocumento(*aFechaInicio*, *aFechaFin*, *aCodigoConcepto*, *aCodigoCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aFechaInicio</i>	Cadena	Por valor	Fecha inicial del rango.
	<i>aFechaFin</i>	Cadena	Por valor	Fecha final del rango.
	<i>aCodigoConcepto</i>	Cadena	Por valor	Código del concepto a filtrar.
	<i>aCodigoCteProv</i>	Cadena	Por valor	Código del Cliente/Proveedor a filtrar.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función aplica un filtro a los documentos de acuerdo a su código y al código del cliente/proveedor en un rango de fechas especificados.

Ejemplo El siguiente código indica a la aplicación que realice un filtro de documentos para el primer semestre del año 2006, para el código de concepto 4 y el código de Cliente/Proveedor CTE002.

```
IFechaInicio = "01/01/2006"
IFechaFin = "06/30/2006"
ICodConcepto = "4"
ICodCteProv = "CTE002"

IError = fSetFiltroDocumento (aFechaInicio, aFechaFin, aCodigoConcepto,
aCodigoCteProv)
```

Ver también [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fCancelaFiltroDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fCancelaFiltroDocumento ()

Parámetros No usa.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela el ultimo filtro activo de documentos.

Ejemplo El siguiente código cancela el ultimo filtro activo de documentos.

```
fCancelaFiltroDocumento ()
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fSetFiltroDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fBuscaDocumento \(\)](#)

fDocumentoImpreso ()

Disponibilidad AdminPAQ 2003.

Sintaxis fDocumentoImpreso (*alImpreso*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>alImpreso</i>	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cambia la bandera de documento impreso.
Es necesario estar en el registro del documento que se quiere actualizar la bandera.

Ejemplo El siguiente código indica a la aplicación que cambie el estado de impresión del documento a Falso.

```
lImpreso = False  
lError = fDocumentoImpreso (lImpreso)
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fBuscaDocumento \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#)

Bajo Nivel – Búsqueda/Navegación

fBuscarDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBuscarDocumento (*aCodConcepto*, *aSerie*, *aFolio*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto</i>	Cadena	Por valor	Código del concepto del documento.
	<i>aSerie</i>	Cadena	Por valor	Serie del documento.
	<i>aFolio</i>	Cadena	Por valor	Folio del documento.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un documento por su llave, si lo encuentra se posiciona en el registro correspondiente.

Ejemplo El siguiente código indica a la aplicación que busque un documento por su código de concepto, serie y folio.

```
Error = fBuscarDocumento (aCodConcepto, aSerie, aFolio)
```

Ver también [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBuscarIdDocumento \(\)](#), [fBuscaDocumento \(\)](#)

fBuscarIdDocumento ()

Disponibilidad AdminPAQ 2005.

Sintaxis fBuscarIdDocumento (*aldDocumento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDocumento</i>	Entero	Por valor	Identificador del documento.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un documento por su identificador.

Ejemplo El siguiente código busca un documento por su identificador.

```
fBuscarIdDocumento (!IdDocumento)
```

Ver también [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fBuscarDocumento \(\)](#), [fAfectaDocto \(\)](#), [fBuscaDocumento \(\)](#)

fPosPrimerDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosPrimerDocumento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Documentos.

```
!Error = fPosPrimerDocumento ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fPosUltimoDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosUltimoDocumento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el último registro de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el último registro de la tabla de Documentos.

```
!Error = fPosUltimoDocumento ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fPosSiguienteDocumento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosSiguienteDocumento ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de documentos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de documentos.

```
IError = fPosSiguienteDocumento ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fPosAnteriorDocumento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosAnteriorDocumento ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de documentos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de documentos.

```
IError = fPosAnteriorDocumento ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fPosBOF ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosBOF ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Documentos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaDocto el resultado de la función fPosBOF .

```
lInicioTablaDocto = fPosBOF ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

fPosEOF ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosEOF ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Documentos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaDocto el resultado de la función fPosEOF .

```
lFinTablaDocto = fPosEOF ()
```

Ver también [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fBuscaDocumento \(\)](#)

Alto Nivel – Lectura/Escritura

fAltaDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAltaDocumento (*aldDocumento*, *aDocumento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDocumento</i>	Entero largo	Por referencia	Identificador del documento.
	<i>aDocumento</i>	tDocumento	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aldDocumento: Al finalizar la función este parámetro contiene el identificador del nuevo documento.

Descripción Esta función da de alta documentos de cargo o abono.

Ejemplo El siguiente código indica a la aplicación que de alta un documento, en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fAltaDocumento (IldDocto, IDocumento)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fBuscaDocumento \(\)](#)

fAltaDocumentoCargoAbono ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAltaDocumentoCargoAbono (aDocumento)

Parámetros	Nombre	Tipo	Uso	Descripción
	aDocumento	tDocumento	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función da de alta documentos de cargo o abono.

Ejemplo El siguiente código indica a la aplicación que de alta un documento de cargo/abono, en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fAltaDocumentoCargoAbono (IDocumento)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fAfectaDocto Param \(\)](#), [fSaldarDocumento Param \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fAfectaDocto ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAfectaDocto (aLlaveDocto, aAfecta)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aLlaveDocto</i>	tLlaveDocto	Por valor	Tipo de dato abstracto.
	<i>aAfecta</i>	Lógico (Bool)	Por valor	Verdadero o falso. Afectar o desafectar.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función utiliza *aLlaveDocto* como llave del documento y *aAfecta* para afectar o desafectarlo.

Ejemplo El siguiente código indica a la aplicación que de afecte un documento, en caso de presentarse algún error manda el mensaje correspondiente.

```
IError = fAfectaDocto (IDocumento, True)
If IError <> 0 Then
  MensajeError IError
Else
```

Ver también [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fBuscaDocumento \(\)](#)

fSaldarDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSaldarDocumento (aDoctoaPagar, aDoctoPago, alImporte, aldMoneda, aFecha)

Parámetros	Nombre	Tipo	Uso	Descripción
	aDoctoaPagar	tLlaveDocto	Por valor	Tipo de dato abstracto.
	aDoctoPago	tLlaveDocto	Por valor	Tipo de dato abstracto.
	alImporte	Doble	Por valor	Importe del pago.
	aldMoneda	Entero	Por valor	Moneda del pago.
	aFecha	Cadena	Por valor	Fecha del pago.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asocia documentos y salda sus importes.

Ejemplo El siguiente código indica a la aplicación que de salde un documento con la información enviada, en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fSaldarDocumento (IDoctoaPagar, IDoctoPago, IImporte, IldMoneda, IFecha)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fSaldarDocumentoCheqPAQ ()

Disponibilidad AdminPAQ 2004.

Sintaxis fSaldarDocumentoCheqPAQ (aDoctoPagara, aDoctoPago, alImporte, aldMoneda, aFecha, aTipoCambioCheqPAQ)

Parámetros	Nombre	Tipo	Uso	Descripción
	aDoctoPagara	tLlaveDocto	Por valor	Tipo de dato abstracto.
	aDoctoPago	tLlaveDocto	Por valor	Tipo de dato abstracto.
	alImporte	Doble	Por valor	Importe del pago.
	aldMoneda	Entero	Por valor	Moneda del pago.
	aFecha	Cadena	Por valor	Fecha del pago.
	aTipoCambioCheqPAQ	Doble	Por valor	Tipo de cambio de la moneda extranjera que interviene en el cargo-abono del día de pago.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asocia documentos y salda sus importes.

Ejemplo El siguiente código indica a la aplicación que de salde un documento con el documento correspondiente en CheqPAQ usando el tipo de cambio del día de pago; en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fSaldarDocumentoCheqPAQ (IDoctoPagara, IDoctoPago, IImporte, IldMoneda, IFecha)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguieteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguieteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fBorrarAsociacion ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBorrarAsociacion (aDoctoPagara, aDoctoPago)

Parámetros	Nombre	Tipo	Uso	Descripción
	aDoctoPagara	tLlaveDocto	Por valor	Tipo de dato abstracto.
	aDoctoPago	tLlaveDocto	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función la asociación de documentos.

Ejemplo El siguiente código indica a la aplicación que elimine la asociación entre *IDoctoPagara* y *IDoctoPago* ; en caso de presentarse algún error manda el mensaje correspondiente.

```

IError = fBorrarAsociacion (IDoctoPagara, IDoctoPago)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoChegPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fRegresaIVACargo ()

Disponibilidad AdminPAQ 2002.

Sintaxis fRegresaIVACargo (*aLlaveDocto*, *aNetoTasa15*, *aNetoTasa10*, *aNetoTasaCero*, *aNetoTasaExcenta*, *aNetoOtrasTasas*, *aIVATasa15*, *aIVATasa10*, *aIVAOtrasTasas*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aLlaveDocto</i>	<i>tLlaveDocto</i>	Por valor	Tipo de dato abstracto.
	<i>aNetoTasa15</i>	Doble	Por referencia	Base de la tasa de 15%
	<i>aNetoTasa10</i>	Doble	Por referencia	Base de la tasa de 10%
	<i>aNetoTasaCero</i>	Doble	Por referencia	Base de la tasa cero
	<i>aNetoTasaExcenta</i>	Doble	Por referencia	Base de productos exentos
	<i>aNetoOtrasTasas</i>	Doble	Por referencia	Base de otras tasas
	<i>aIVATasa15</i>	Doble	Por referencia	IVA de la tasa de 15%
	<i>aIVATasa10</i>	Doble	Por referencia	IVA de la tasa de 10%
	<i>aIVAOtrasTasas</i>	Doble	Por referencia	IVA de otras tasas

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función regresa el desglose de IVA de un documento.

Ejemplo El siguiente código indica a la aplicación que obtenga el desglose del IVA del documento especificado en el parámetro *lLlaveDocto* y que regrese los valores correspondientes a los parámetros por valor especificados.

```
Error = fRegresaIVACargo (lLlaveDocto, lNetoTasa15, lNetoTasa10, lNetoTasaCero,
lNetoTasaExcenta, lNetoOtrasTasas, lIVATasa15, lIVATasa10, lIVAOtrasTasas)
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguieteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguieteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

fGetTamSelloDigitalYCadena ()

Disponibilidad AdminPAQ 2007. Versión 4.1.4

Sintaxis *fGetTamSelloDigitalYCadena (atPtrPassword, aEspSelloDig, aEspCadOrig)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>atPtrPassword</i>	Cadena	Por referencia	Contraseña del certificado.
	<i>aEspSelloDig</i>	Entero	Por referencia	Tamaño del Sello digital.
	<i>aEspCadOrig</i>	Entero	Por referencia	Tamaño de la Cadena original.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Con esta función se obtiene el tamaño de la cadena original y el sello digital, mismas que se guardarán en las variables ***aEspSelloDig*** y ***aEspCadOrig***.

Ejemplo En el siguiente código se pasa como parámetro la contraseña del certificado almacenada en la variable **IPassword** e indica a la aplicación que obtenga el tamaño del sello digital y cadena original del documento, recibiendo en las variables **IEspSello** y **IEspSello**.

```
IEspSello = fGetTamSelloDigitalYCadena (IPassword, IEspSello, IEspCadOrig)
```

Ver también [fGetSelloDigitalYCadena\(\)](#)

fGetSelloDigitalYCadena ()

Disponibilidad AdminPAQ 2007. Versión 4.1.4

Sintaxis *fGetSelloDigitalYCadena (char *atPtrPassword, char* atPtrSelloDigital, char* atPtrCadenaOriginal)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>atPtrPassword</i>	Cadena	Por referencia	Contraseña del certificado.
	<i>atPtrSelloDigital</i>	Cadena	Por referencia	Sello digital.
	<i>atPtrCadenaOriginal</i>	Cadena	Por referencia	Cadena original.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Con esta función se obtiene el sello digital y la cadena original de un CFD.

Ejemplo En el siguiente código se pasa como parámetro la contraseña del certificado almacenada en la variable **IPassword** e indica a la aplicación que obtenga el tamaño del sello digital y cadena original del documento, recibiendo en las variables **IEspSello** y **IEspSello**.

```
IEspSello = fGetSelloDigitalYCadena (IPassword, IespSello, IespCadOrig)
```

Ver también [fGetTamSelloDigitalYCadena\(\)](#)

fnicializaLicenseInfo()

Disponibilidad AdminPAQ 2011. Versión 6.2.1

Sintaxis *fnicializaLicenseInfo (aSistema)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aSistema</i>	Unsigned char	Por valor	Sistema: 0 = AdminPAQ 1 = CONTPAQ i® Factura Electrónica
Retorna	Valores enteros: kSIN_ERRORES = • 0 (cero) que significa que se pudo conectar y obtener información del Servidor de Licencias, aunque la verificación del número de usuarios se hace hasta el uso de la función fEmitirDocumento. !kSIN_ERRORES = • -1 que significa que hubo un error al intentar obtener información del Servidor de Licencias del sistema especificado.			

Descripción Esta función verifica que el sistema esté activado y tenga una licencia válida.

Ejemplo En el siguiente código pasa como parámetro un 0 indicando que se trata del sistema AdminPAQ para saber si el sistema está activado.

```
!Error = fnicializaLicenseInfo (0)
```

Ver también [fEmitirDocumento\(\)](#)

fEmitirDocumento ()

Disponibilidad AdminPAQ 2011. Versión 6.2.1

Sintaxis *fEmitirDocumento (aCodConcepto, aSerie, aFolio, aPassword, aArchivoAdicional)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto</i>	Cadena	Por referencia	Código del concepto
	<i>aSerie</i>	Cadena	Por referencia	Serie del documento
	<i>aFolio</i>	Doble	Por valor	Folio del documento
	<i>aPassword</i>	Cadena	Por referencia	Contraseña del certificado de sello digital
	<i>aArchivoAdicional</i>	Cadena	Por referencia	Nombre del archivo con el complemento, este archivo ya debe existir en la carpeta "Adicionales" dentro de la empresa.

Retorna Valores enteros:

- KSIN_ERRORES = • 0 (cero) si no hubo error.
- !KSIN_ERRORES = • -1 • -1 que significa que hubo un error con la Licencia (la licencia es para menos de 10 usuarios, es temporal, de evaluación, no está activada, etc.)
- !KSIN_ERRORES = • Un número de error positivo del que se puede obtener la descripción con la función fError.

Descripción Para poder utilizar la función fEmitirDocumento, se deberá ejecutar primero la función fnicializaLicenseInfo.

La función fEmitirDocumento recibe como parámetro el nombre de un archivo el cual contiene un complemento que se insertará al CFD/CFDI al emitirse.

Esta función requiere una licencia monousuario. Si cuentas con un licenciamiento anual además se requiere que la licencia sea multiempresa.

Esta función solo soporta los divisas, EstadoDeCuentaBancario, EstadoDeCuentaCombustible, PrestadoresDeServiciosDeCFD y la combinacion de estos.

Ejemplo La siguiente función emite un documento, pasando como parámetros el código = 1001", serie=CFDI, folio = 154 del concepto, contraseña del certificado a0123456789, y nombre del archivo complemento "divisas.xml".

```
lError = fEmitirDocumento("1001", "CFDI", 154, "a0123456789", "divisas.xml")
```

Ver también [fnicializaLicenseInfo\(\)](#)

fDocumentoUUID()

Disponibilidad AdminPAQ 2011. Versión 6.2.3

Sintaxis *fDocumentoUUID (aCodigoConcepto, aSerie, aFolio, atPtrCFDIUUID)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto</i>	Cadena	Por referencia	Código del concepto
	<i>aSerie</i>	Cadena	Por referencia	Serie del documento
	<i>aFolio</i>	Doble	Por valor	Folio del documento
	<i>atPtrCFDIUUID</i>	Cadena	Por referencia	Cadena para colocar el valor de UUID

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función despliega el UUID de un documento.

Ejemplo En el siguiente código se pasan el código, serie y folio del concepto para recibir el UUID en la variable IUUID.

```
IError = fDocumentoUUID (IConcepto, ISerie, IFolio, IUUID)
```

fGetSerieCertificado ()

Disponibilidad AdminPAQ 2007. Versión 4.3.1

Sintaxis *fGetSerieCertificado (atPtrPassword, aPtrSerieCertificado)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>atPtrPassword</i>	Cadena	Por referencia	Contraseña del certificado
	<i>aPtrSerieCertificado</i>	Cadena	Por referencia	Serie del certificado

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función regresa la serie de un Certificado utilizado por una factura electrónica.

fActivarPrecioCompra ()

Disponibilidad AdminPAQ 2007. Versión 4.3.2

Sintaxis *fActivarPrecioCompra (aActivar)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aActivar</i>	Entero	Por valor	0 = No busca el precio 1 = Valor asumido (busca el precio)

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función determina si al momento de registrar una compra vía SDK se ejecutará la función que busca el último precio de compra registrado en caso de que el precio sea igual a cero.

fDocumentoDevuelto ()

Disponibilidad AdminPAQ 2007. Versión 4.3.2

Sintaxis *fDocumentoDevuelto (aDevuelto)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aDevuelto</i>	Entero	Por valor	0 = No devuelto 1 = Devuelto
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función ajusta el estado de un documento en devuelto o no devuelto.			

fEntregEnDiscoXML ()

Disponibilidad AdminPAQ 2011. Versión 7.0.0

Sintaxis *fEntregEnDiscoXML (aCodConcepto, aSerie, aFolio, aFormato, aFormatoAmig)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto</i>	Cadena	Por referencia	Código del concepto
	<i>aSerie</i>	Cadena	Por referencia	Serie del documento
	<i>aFolio</i>	Doble	Por valor	Folio del documento
	<i>aFormato</i>	Entero	Por valor	Formato de entrega (0 = XML, 1 = PDF)
				Nota: Al seleccionar la opción de entrega 1= PDF, por disposición fiscal también se generará el XML.
	<i>aFormatoAmig</i>	Cadena	Por referencia	Plantilla de impresión
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función entrega el XML en un archivo.			
Ejemplo	En el siguiente ejemplo, se entrega el XML del concepto 4 (factura), Serie B1, folio 45, en formato PDF, en la ruta "C:\Compacw\Empresas\Reportes\AdminPAQ\Plantilla_Factura_cfdi_1.html"			
	<pre>lError = fEntregEnDiscoXML ("4", "B1", 45, 1, "C:\Compacw\Empresas\Reportes\AdminPAQ\Plantilla_Factura_cfdi_1.html")</pre>			

fObtieneDatosCFDI ()

Disponibilidad AdminPAQ 2012. Versión 7.2.0

Sintaxis *fObtieneDatosCFDI (atPtrPassword)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>atPtrPassword</i>	Cadena	Por referencia	Contraseña del certificado

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción La función **fObtieneDatosCFDI** obtiene los datos del CFDI del documento previamente definido con la función **fBuscarDocumento**. Esta función almacena en variables globales los datos del CFDI dentro del mismo SDK para posteriormente ser leídos con la función **fLeeDatoCFDI**.

Ejemplo El siguiente ejemplo obtiene los datos del CFDI previamente ubicado con la función **fBuscarDocumento** pasando como parámetro la contraseña del certificado en la variable IPassword.

```
lError = fObtieneDatosCFDI(IPassword)
```

Ver también [fBuscarDocumento\(\)](#), [fLeeDatoCFDI\(\)](#)

fLeeDatoCFDI ()

Disponibilidad AdminPAQ 2012. Versión 7.2.0

Sintaxis *fLeeDatoCFDI (aValor, aDato)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aValor</i>	Cadena	Por referencia	Cadena donde se regresará el dato requerido
	<i>aDato</i>	Entero	Por valor	1 = Serie del Certificado del Emisor 2 = Folio Fiscal (UUID) 3 = Número de Serie del Certificado del SAT 4 = Fecha y Hora de Certificación 5 = Sello Digital del CFDI 6 = Sello SAT 7 = Cadena Original del Complemento de Certificación Digital del SAT 8 = Método de Pago 9 = Lugar de expedición 10 = Régimen Fiscal 11 = Folio Fiscal de origen* 12 = Serie del Folio Fiscal de origen* 13 = Fecha del Folio Fiscal de origen* 14 = Monto del Folio Fiscal de origen* * Para documentación de Deuda o Pago en Parcialidades

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción La función fLeeDatoCFDI lee los datos previamente accedidos con la función fObtieneDatosCFDI.

La función recibe como parámetros, la cadena donde copiará el dato requerido y un entero donde se indica qué dato se desea y regresará un número de error en caso de existir alguno.

Ejemplo El siguiente ejemplo el UUID de un documento y lo almacena en la variable IUUID.

```
IError = fLeeDatoCFDI (IUUID, 2)
```

Ver también [fBuscarDocumento\(\)](#), [fObtieneDatosCFDI\(\)](#)

Alto Nivel – Búsqueda/Navegación

fBuscaDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBuscaDocumento (*aLlaveDocto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aLlaveDocto</i>	tLlaveDocto	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un documento por su llave, si lo encuentra se posiciona en el registro correspondiente.

Ejemplo El siguiente código indica a la aplicación que busque el documento cuya llave es el contenido del parámetro *lLlaveDocto*.

```
!Error = fBuscaDocumento (lLlaveDocto )
```

Ver también [fInsertarDocumento \(\)](#), [fEditarDocumento \(\)](#), [fGuardaDocumento \(\)](#), [fBorraDocumento \(\)](#), [fCancelarModificacionDocumento \(\)](#), [fCancelaDocumento \(\)](#), [fBorraDocumento_CW \(\)](#), [fCancelaDocumento_CW \(\)](#), [fAfectaDocto_Param \(\)](#), [fSaldarDocumento_Param \(\)](#), [fBorrarAsociacion_Param \(\)](#), [fSetDatoDocumento \(\)](#), [fLeeDatoDocumento \(\)](#), [fSiguienteFolio \(\)](#), [fSetFiltroDocumento \(\)](#), [fCancelaFiltroDocumento \(\)](#), [fDocumentoImpreso \(\)](#), [fBuscarDocumento \(\)](#), [fBuscarIdDocumento \(\)](#), [fPosPrimerDocumento \(\)](#), [fPosUltimoDocumento \(\)](#), [fPosSiguienteDocumento \(\)](#), [fPosAnteriorDocumento \(\)](#), [fPosBOF \(\)](#), [fPosEOF \(\)](#), [fAltaDocumento \(\)](#), [fAltaDocumentoCargoAbono \(\)](#), [fAfectaDocto \(\)](#), [fSaldarDocumento \(\)](#), [fSaldarDocumentoCheqPAQ \(\)](#), [fBorrarAsociacion \(\)](#), [fRegresaIVACargo \(\)](#), [fRegresaIVAPago \(\)](#), [fBuscaDocumento \(\)](#)

Funciones de Movimientos

Bajo Nivel – Lectura/Escritura

fInsertarMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fInsertarMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Añade un nuevo registro en la tabla de Movimientos en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Movimientos.
	<pre>fInsertarMovimiento ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditarMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fEditarMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Activa el modo de Edición de un registro en la tabla de Movimientos.
Ejemplo	El siguiente código busca un movimiento por su Identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente
	<pre>lError = fBuscarIdMovimiento (lIdMovto) If lError <> 0 Then MensajeError lError Else fEditarMovimiento () End If</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fGuardaMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fGuardaMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un movimiento.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Documentos. Esta función se llama después de que se utiliza la función fInsertarMovimiento() o fEditarMovimiento() y se graban los valores en los campos correspondientes.
	<pre>fGuardaMovimiento ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelaCambiosMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fCancelaCambiosMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual de movimientos. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro de Movimientos que estaba en modo de inserción o edición.
	<pre>fCancelaCambiosMovimiento ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoCaracteristicas_Param ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaMovimientoCaracteristicas_Param (*aldMovimiento, aldMovtoCaracteristicas, aUnidades, aValorCaracteristica1, aValorCaracteristica2, aValorCaracteristica3*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Cadena	Por valor	Identificador del movimiento.
	<i>aldMovtoCaracteristicas</i>	Cadena	Por valor	Identificador del movimiento con características.
	<i>aUnidades</i>	Cadena	Por valor	Unidades.
	<i>aValorCaracteristica1</i>	Cadena	Por valor	Valor de la característica 1.
	<i>aValorCaracteristica2</i>	Cadena	Por valor	Valor de la característica 2.
	<i>aValorCaracteristica3</i>	Cadena	Por valor	Valor de la característica 3.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función inserta un movimiento con características.

Ejemplo El siguiente código indica a la aplicación que inserte un movimiento con características en la base de datos, en caso de que ocurra un error muestra el mensaje de error correspondiente.

```

IError = fAltaMovimientoCaracteristicas_Param ("IldMovimiento, IldMovtoCaracteristicas,
IUnidades, IValorCaracteristica1, IValorCaracteristica2, IValorCaracteristica3)
If IError <> 0 Then
  MensajeError IError
Else

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovtoCaracteristicasUnidades_Param ()

Disponibilidad AdminPAQ 2005.

Sintaxis fAltaMovtoCaracteristicasUnidades_Param (*aldMovimiento*, *aldMovtoCaracteristicas*, *aUnidad*, *aUnidades*, *aUnidadesNC*, *aValorCaracteristica1*, *aValorCaracteristica2*, *aValorCaracteristica3*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Cadena	Por valor	Identificador del movimiento.
	<i>aldMovtoCaracteristicas</i>	Cadena	Por valor	Identificador del movimiento con características.
	<i>aUnidad</i>	Cadena	Por valor	Abreviatura de la unidad de compra venta
	<i>aUnidades</i>	Cadena	Por valor	Las unidades del movimiento de características.
	<i>aUnidadesNC</i>	Cadena	Por valor	Abreviatura de la unidad de compra venta no convertible.
	<i>aValorCaracteristica1</i>	Cadena	Por valor	Valor de la característica 1.
	<i>aValorCaracteristica2</i>	Cadena	Por valor	Valor de la característica 2.
	<i>aValorCaracteristica3</i>	Cadena	Por valor	Valor de la característica 3.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función da de alta movimiento de características con unidades de compra venta.

Ejemplo El siguiente da de alta movimiento de características con unidades de compra venta.

```
fAltaMovtoCaracteristicasUnidades_Param (lIdDocumento, lIdMovimiento, lMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoSeriesCapas_Param ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaMovimientoSeriesCapas_Param (*aldMovimiento, aUnidades, aTipoCambio, aSeries, aPedimento, aAgencia, aFechaPedimento, aNumeroLote, aFechaFabricacion, aFechaCaducidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Cadena	Por valor	Identificador del movimiento.
	<i>aUnidades</i>	Cadena	Por valor	Unidad de peso y medida.
	<i>aTipoCambio</i>	Cadena	Por valor	Tipo de cambio.
	<i>aSeries</i>	Cadena	Por valor	Series.
	<i>aPedimento</i>	Cadena	Por valor	Referencia del pedimento.
	<i>aAgencia</i>	Cadena	Por valor	Referencia de la agencia.
	<i>aFechaPedimento</i>	Cadena	Por valor	Fecha del pedimento.
	<i>aNumeroLote</i>	Cadena	Por valor	Número de lote.
	<i>aFechaFabricacion</i>	Cadena	Por valor	Fecha de fabricación.
	<i>aFechaCaducidad</i>	Cadena	Por valor	Fecha de caducidad.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función agrega el movimiento de numero de serie, lote y/o pedimento asociados un movimiento cuyo producto maneje cualquiera de estas posibles configuraciones.

Ejemplo El siguiente código da de alta un movimiento para un producto con numero de serie, lote y/o pedimento asociado.

```
IError = fAltaMovimientoSeriesCapas_Param (IldMovimiento, IUnidades, ITipoCambio, ISeries,
IPedimento, IAgencia, IFechaPedimento, INumeroLote, IFechaFabricacion,
IFechaCaducidad)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCalculaMovtoSerieCapa ()

Disponibilidad AdminPAQ 2004.

Sintaxis fCalculaMovtoSerieCapa (*aldMovimiento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Entero largo	Por valor	Identificador del movimiento a recalcular.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función recalcula el movimiento cuando este pertenece a un producto con series, lotes o pedimentos.

Ejemplo El siguiente código recalcula el movimiento cuando este pertenece a un producto con series, lotes o pedimentos.

```
lError = fCalculaMovtoSerieCapa (lldMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fObtieneUnidadesPendientes ()

Disponibilidad AdminPAQ 2004.

Sintaxis fObtieneUnidadesPendientes (*aConceptoDocto*, *aCodigoProducto*, *aCodigoAlmacen*, *aUnidades*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aConceptoDocto</i>	Cadena	Por valor	Código del concepto del documento a buscar.
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto a buscar su unidades pendientes.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén a buscar si es igual a 0 (cero) busca en todos los almacenes.
	<i>aUnidades</i>	Cadena	Por referencia	Valor de retorno con las unidades pendientes.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aUnidades: Al finalizar la función este parámetro contiene las unidades pendientes.

Descripción Esta función obtiene la cantidad de unidades pendientes de cierto concepto de documento para un almacén/almacenes de un determinado producto en toda la historia del sistema.

Ejemplo El siguiente código indica a la aplicación que obtenga las unidades pendientes para el producto especificado en *ICodigoProducto* del almacén *ICodigoAlmacen*, del documento **IConceptoDocto**.

```
!Error = fObtieneUnidadesPendientes (IConceptoDocto, ICodigoProducto, ICodigoAlmacen, IUnidades)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fObtieneUnidadesPendientesCarac ()

Disponibilidad AdminPAQ 2004.

Sintaxis fObtieneUnidadesPendientesCarac (*aConceptoDocto*, *aCodigoProducto*, *aCodigoAlmacen*, *aValorCaracteristica1*, *aValorCaracteristica2*, *aValorCaracteristica3*, *aUnidades*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aConceptoDocto</i>	Cadena	Por valor	Código del concepto del documento a buscar.
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto a buscar su unidades pendientes.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén a buscar si es igual a 0 (cero) busca en todos los almacenes.
	<i>aValorCaracteristica1</i>	Cadena	Por valor	Valor característica 1
	<i>aValorCaracteristica2</i>	Cadena	Por valor	Valor característica 2
	<i>aValorCaracteristica3</i>	Cadena	Por valor	Valor característica 3
	<i>aUnidades</i>	Cadena	Por referencia	Valor de retorno con las unidades pendientes.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aUnidades: Al finalizar la función este parámetro contiene las unidades pendientes.

Descripción Esta función obtiene la cantidad de unidades pendientes de cierto concepto de documento para un almacén/almacenes de un determinado producto con características en toda la historia del sistema.

Ejemplo El siguiente código indica a la aplicación que obtenga las unidades pendientes para el producto especificado en ICodigoProducto del almacen ICodigoAlmacen, del documento IConceptoDocto.

```
IError = fObtieneUnidadesPendientes (IConceptoDocto, ICodigoProducto, ICodigoAlmacen,
IValorCaracteristica1, IValorCaracteristica2, IValorCaracteristica3,
IUnidades)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fModificaCostoEntrada ()

Disponibilidad AdminPAQ 2005.

Sintaxis fModificaCostoEntrada (*aldMovimiento*, *aCostoEntrada*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Cadena	Por valor	Identificador del movimiento a modificar.
	<i>aCostoEntrada</i>	Cadena	Por valor	Valor del costo a asignar al movimiento.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función modifica el costo de una entrada de inventario.

Ejemplo El siguiente código indica a la aplicación que modifique el costo de la entrada de inventario especificada en lldMovimiento.

```
lError = fModificaCostoEntrada (lldMovimiento, lCostoEntrada)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetDatoMovimiento (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que busque un movimiento por su identificador, en caso de que lo encuentre escribe el contenido de la variable IDescuento en el campo cDescuen01 de la tabla de movimientos; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscarIdMovimiento (IIdMovto)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoMovimiento ("cDescuen01 ", IDescuento)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fLeeDatoMovimiento (*aCampo*, *aValr*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que busque un movimiento por su identificador, en caso de que lo encuentre escribe el contenido del campo **cDescuen01** en la variable **IDescuento** con una longitud de 8 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscarIdMovimiento (IIdMovto)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoMovimiento ("cDescuen01 ", IDescuento, 8)
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fSetFiltroMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetFiltroMovimiento(aldDocumento)

Parámetros	Nombre	Tipo	Uso	Descripción
	aldDocumento	Long	Por valor	Identificador del documento.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función aplica un filtro de movimientos de acuerdo al documento indicado.

Ejemplo El siguiente código indica a la aplicación que realice un filtro de movimientos para el documento especificado en el parámetro *lldDocto*.

```
lError = fSetFiltroMovimiento (lldDocto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelaFiltroMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fCancelaFiltroMovimiento ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función aplica un filtro de movimientos de acuerdo al documento indicado.

Ejemplo El siguiente código indica a la aplicación que cancele el filtro de movimientos activo.

```
fCancelaFiltroMovimiento ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscarIdMovimiento ()

Disponibilidad AdminPAQ 2004.

Sintaxis fBuscarIdMovimiento (*aldMovimiento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Entero largo	Por valor	Identificador del movimiento.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un movimiento por su identificador. Si lo encuentra se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca un movimiento por su identificador.

```
fBuscarIdMovimiento (lIdMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosPrimerMovimiento ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de movimientos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Movimientos.

```
IError = fPosPrimerMovimiento ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimoMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosUltimoMovimiento ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de documentos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Movimientos.

```
IError = fPosUltimoMovimiento ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosSiguienteMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de documentos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Movimientos. <pre>IError = fPosSiguienteMovimiento ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorMovimiento ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosAnteriorMovimiento ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de documentos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Movimientos. <pre>IError = fPosAnteriorMovimiento ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosMovimientoBOF ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosMovimientoBOF ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Movimientos.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaMovto el resultado de la función fPosMovimientoBOF .
	<pre>lInicioTablaMovto = fPosMovimientoBOF ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosMovimientoEOF ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosMovimientoEOF ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Documentos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaMovto el resultado de la función fPosMovimientoEOF .
	<pre>lFinTablaMovto = fPosMovimientoEOF ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Alto Nivel – Lectura/Escritura

fAltaMovimiento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAltaMovimiento (*aldDocumento*, *aldMovimiento*, *astMovimiento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDocumento</i>	Entero largo	Por valor	Identificador del movimiento.
	<i>aldMovimiento</i>	Entero largo	Por referencia	Identificador del documento.
	<i>astMovimiento</i>	tMovimiento	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldMovimiento*: Al finalizar la función este parámetro contiene el identificador del nuevo movimiento.
 Esta función da de alta un nuevo registro en la tabla de Movimientos.

Ejemplo El siguiente código da de alta un nuevo movimiento.

```
fAltaMovimiento (lIdDocumento, lIdMovimiento, lMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoEx ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAltaMovimientoEx (*aldMovimiento*, *aTipoProducto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Entero largo	Por referencia	Identificador del documento.
	<i>aTipoProducto</i>	tTipoProducto	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función da de alta de un movimiento los datos adicionales de un producto con series, lotes, pedimientos o características.

Ejemplo El siguiente código da de alta un movimiento los datos adicionales de un producto con series, lotes, pedimientos o características.

```
fAltaMovimientoEx (aldMovimiento, aTipoProducto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoCDesct ()

Disponibilidad AdminPAQ 2007.

Sintaxis fAltaMovimientoCDesct (*aldDocumento*, *aldMovimiento*, *astMovimiento*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDocumento</i>	Entero largo	Por valor	Identificador del documento.
	<i>aldMovimiento</i>	Entero largo	Por Referencia	Identificador del movimiento
	<i>astMovimiento</i>	tMovimientoDesc	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función da de alta un nuevo registro en la tabla de Movimientos.
 Esta función incluye Importes y Porcentajes de Descuentos, a diferencia de la función fAltaMovimiento.

Ejemplo El siguiente código da de alta un movimiento los datos adicionales de un producto con series, lotes, pedimientos o características.

```
fAltaMovimientoCDesct (aldDocumento, aldMovimiento, astMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoCaracteristicas ()

Disponibilidad AdminPAQ 2004.

Sintaxis fAltaMovimientoCaracteristicas (*aldMovimiento*, *aldMovtoCaracteristicas*, *aCaracteristicas*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Entero largo	Por valor	Identificador del movimiento.
	<i>aldMovtoCaracteristicas</i>	Entero largo	Por referencia	Identificador del documento.
	<i>aCaracteristicas</i>	tCaracteristicas	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aldMovtoCaracteristicas: Al finalizar la función este parámetro contiene el identificador del nuevo movimiento.

Descripción Esta función inserta un movimiento con características.

Ejemplo El siguiente código da de alta un movimiento con características.

```
fAltaMovimientoCaracteristicas (lIdDocumento, lIdMovimiento, lMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovtoCaracteristicasUnidades ()

Disponibilidad AdminPAQ 2005.

Sintaxis fAltaMovtoCaracteristicasUnidades (*aldMovimiento*, *aldMovtoCaracteristicas*,
aCaracteristicasUnidades)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	Entero largo	Por valor	Identificador del movimiento.
	<i>aldMovtoCaracteristicas</i>	Entero largo	Por referencia	Identificador del documento.
	<i>aCaracteristicasUnidades</i>	<i>tCaracteristicasUnidades</i>	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldMovtoCaracteristicas*: Al finalizar la función este parámetro contiene el identificador del nuevo movimiento.

Esta función da de alta movimiento de características con unidades de compra venta.

Ejemplo El siguiente da de alta movimiento de características con unidades de compra venta.

```
fAltaMovtoCaracteristicasUnidades (lIdDocumento, lIdMovimiento, lMovimiento)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fAltaMovimientoSeriesCapas ()

Disponibilidad AdminPAQ 2002.

Sintaxis fAltaMovimientoSeriesCapas (*aldMovimiento*, *aSeriesCapas*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldMovimiento</i>	long	Por valor	Identificador del movimiento.
	<i>aSeriesCapas</i>	tSeriesCapas	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función agrega el movimiento de numero de serie, lote y/o pedimento asociados un movimiento cuyo producto maneje cualquiera de estas posibles configuraciones.

Ejemplo El siguiente da de alta un movimiento para un producto con numero de serie, lote y/o pedimento asociado.

```
lError = fAltaMovimientoSeriesCapas (lIdMovimiento, lSeriesCapas)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES DE CLIENTES / PROVEEDORES

Bajo Nivel – Lectura/Escritura

fInsertaCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fInsertaCteProv ()

Parámetros No usa.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Adiciona un nuevo registro en la tabla de Clientes / Proveedores en modo de inserción.

Ejemplo El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Clientes / Proveedores.

```
fInsertaCteProv ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fEditaCteProv ()

Parámetros No usa.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Activa el modo de Edición de un registro en la tabla de Clientes / Proveedores.

Ejemplo El siguiente código busca un cliente/proveedor por su Identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
lError = fBuscaIdCteProv (lIdCteProv)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaCteProv ()
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fGuardaCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fGuardaCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro de cliente/proveedor.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Clientes / Proveedores. Esta función se llama después de que se utiliza la función fInsertaCteProv() o fEditaCteProv() y se graban los valores en los campos correspondientes.

```
fGuardaCteProv ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fBorraCteProv ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fBorraCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Borra un registro en la tabla de Clientes / Proveedores.
Ejemplo	El siguiente código busca un documento por su identificador, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```

IError = fBuscaIdCteProv (IIdCteProv)
If IError <> 0 Then
    MensajeError IError
Else
    fBorraCteProv ()
End If

```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	---

fCancelarModificacionCteProv ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fCancelarModificacionCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual de Clientes / Proveedores. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro de Clientes / Proveedores que estaba en modo de inserción o edición.
	<pre>fCancelarModificacionCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEliminarCteProv ()

Disponibilidad	AdminPAQ 2003.			
Sintaxis	fEliminarCteProv (aCodigoCteProv)			
Parámetros	Nombre aCodigoCteProv	Tipo Cadena	Uso Por valor	Descripción Código del Cliente / Proveedor
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función elimina un Cliente / Proveedor usando su código.			
Ejemplo	El siguiente código elimina un Cliente / Proveedor, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente			
	<pre>IError = fEliminarCteProv (aCodigoCteProv) If IError <> 0 Then MensajeError IError End If</pre>			
Ver también	fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),			

fSetDatoCteProv ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetDatoCteProv (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de Cliente / Proveedor.

Ejemplo El siguiente código indica a la aplicación que busque un movimiento por su código, en caso de que lo encuentre escribe el contenido de la variable IRFC en el campo cRFC de la tabla de Cliente / Proveedor; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaCteProv (ICodCteProv)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoCteProv ("cRFC ", IRFC)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoCteProv ()

Disponibilidad AdminPAQ 2002.

Sintaxis fLeeDatoCteProv (*aCampo*, *aValr*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de Cliente / Proveedor.

Ejemplo El siguiente código indica a la aplicación que busque un movimiento por su identificador, en caso de que lo encuentre escribe el contenido de l campo cRFC en la variable la variable IRFC con una longitud de 20 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscarIdMovimiento (IIdMovto)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoCteProv ("cRFC ", IRFC, 20)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaCteProv ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBuscaCteProv (*aCodCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodCteProv</i>	Cadena	Por valor	Código del Cliente / Proveedor.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un Cliente / Proveedor por su código.

Ejemplo El siguiente código busca un Cliente / Proveedor por su código.

```
fBuscaIdCteProv (lCodCteProv)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaIdCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaIdCteProv (*aldCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldCteProv</i>	Entero	Por valor	Identificador del Cliente / Proveedor.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un Cliente / Proveedor por su Identificador.

Ejemplo El siguiente código busca un Cliente / Proveedor por su identificador.

```
fBuscaIdCteProv (lIdCteProv)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosPrimerCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el primer registro de la tabla de Clientes / Proveedores.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Clientes / Proveedores.
	<pre>IError = fPosPrimerCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimoCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosUltimoCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el ultimo registro de la tabla de Clientes / Proveedores.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Clientes / Proveedores.
	<pre>IError = fPosUltimoCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosSiguienteCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Clientes / Proveedores.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Clientes / Proveedores.
	<pre>IError = fPosSiguienteCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosAnteriorCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Clientes / Proveedores.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Clientes / Proveedores.
	<pre>IError = fPosAnteriorCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosBOFCteProv()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Documentos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable InicioTablaCteProv el resultado de la función fPosBOFCteProv. <pre>InicioTablaCteProv= fPosBOFCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFCteProv ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosEOFCteProv ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Documentos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable IFinTablaCteProv el resultado de la función fPosEOFCteProv. <pre>IFinTablaCteProv = fPosEOFCteProv ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Alto Nivel – Lectura/Escritura

fAltaCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaCteProv (*aldCteProv*, *astCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldCteProv</i>	Entero	Por referencia	Identificador del Cliente / Proveedor.
	<i>astCteProv</i>	tCteProv	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldCteProv*: Al finalizar la función este parámetro contiene el identificador del nuevo Cliente / Proveedor.
 Esta función da de alta un nuevo Cliente / Proveedor.

Ejemplo El siguiente código da de alta un nuevo Cliente / Proveedor.

```
fAltaCteProv (ldCteProv, lCteProv)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fActualizaCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fActualizaCteProv (*aCodigoCteProv*, *astCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoCteProv</i>	Cadena	Por referencia	Identificador del Cliente / Proveedor.
	<i>astCteProv</i>	tCteProv	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función actualiza un Cliente / Proveedor por medio su código.

Ejemplo El siguiente código actualiza un Cliente / Proveedor por medio su código.

```
fActualizaCteProv (lCodigoCteProv, lCteProv)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLlenaRegistroCteProv ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLlenaRegistroCteProv (*astCteProv*, *aEsAlta*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astCteProv</i>	tCteProv	Por valor	Tipo de dato abstracto.
	<i>aEsAlta</i>	Entero	Por valor	1 = Nuevo Cliente / Proveedor. 2 = Actualizacion Cliente / Proveedor.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asigna al registro de la tabla de Clientes / Proveedores los valores de la estructura de datos *astCteProv*.

Ejemplo El siguiente código da de alta un nuevo Cliente / Proveedor en la tabla de Clientes / Proveedores.

```
fLlenaRegistroCteProv (ltCteProv, 1)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES DE PRODUCTOS

Bajo Nivel – Lectura/Escritura

fInsertaProducto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fInsertaProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en la tabla de productos en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Productos.
	<pre>fInsertaProducto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaProducto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fEditaProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Activa el modo de Edición de un registro en la tabla de Productos.
Ejemplo	El siguiente código busca un producto por su código , si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente
	<pre>!Error = fBuscaProducto (aCodProducto) If !Error <> 0 Then MensajeError !Error Else fEditaProducto () End If</pre>
Ver también	fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),

fGuardaProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fGuardaProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro de productos.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en la tabla de productos. Esta función se llama después de que se utiliza la función fInsertaProducto() o fEditaProducto() y se graban los valores en los campos correspondientes.

```
fGuardaProducto ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fBorraProducto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fBorraProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Borra un registro en la tabla de productos.
Ejemplo	El siguiente código busca un productos por su código, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```

IError = fBuscaProducto (aCodProducto)
If IError <> 0 Then
    MensajeError IError
Else
    fBorraProducto ()
End If

```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	---

fCancelarModificacionProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fCancelarModificacionProducto ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela las modificaciones al registro actual de productos. El registro debe estar en modo de edición o inserción.

Ejemplo El siguiente código indica a la aplicación que cancele la modificación a un registro de productos que estaba en modo de inserción o edición.

```
fCancelarModificacionProducto ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEliminarProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fEliminarProducto (aCodigoProducto)

Parámetros	Nombre	Tipo	Uso	Descripción
	aCodigoProducto	Cadena	Por valor	Código del producto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función elimina un producto usando su código.

Ejemplo El siguiente código elimina un producto, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```
lError = fEliminarProducto (aCodigoProducto)

If lError <> 0 Then
    MensajeError lError
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fSetDatoProducto (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla Productos.

Ejemplo El siguiente código indica a la aplicación que busque un movimiento por su código, en caso de que lo encuentre escribe el contenido de la variable lFechaAlta en el campo cFechaAl01 de la tabla Productos; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaProducto (aCodProducto)
If lError <> 0 Then
  MensajeError lError
Else
  lError = fSetDatoProducto ("cFechaAl01 ", lFechaAlta)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoProducto (*aCampo*, *aValr*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de productos.

Ejemplo El siguiente código indica a la aplicación que busque un producto por su código, en caso de que lo encuentre escribe el contenido de l campo cFechaAl01 en la variable la variable lFechaAlta con una longitud de 20 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaProducto (aCodProducto)
If lError <> 0 Then
  MensajeError lError
Else
  lError = fLeeDatoProducto ("cFechaAl01", lFechaAlta, 8)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRecuperaTipoProducto ()

Disponibilidad AdminPAQ 2002.

Sintaxis fRecuperaTipoProducto(aUnidades, aSerie, aLote, aPedimento, aCaracteristicas)

Parámetros	Nombre	Tipo	Uso	Descripción
	aUnidades	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso. Maneja unidades o no.
	aSerie	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso. Maneja series o no.
	aLote	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso. Maneja lotes o no.
	aPedimento	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso. Maneja pedimentos o no.
	aCaracteristicas	Lógico (bool)	Por referencia	Valor lógico. Verdadero o Falso. Maneja caracterisricas o no.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aUnidades: Al finalizar la función este parámetro indica si el producto maneja unidades o no.

aSerie: Al finalizar la función este parámetro indica si el producto maneja series o no.

aLote: Al finalizar la función este parámetro indica si el producto maneja lotes o no.

aPedimento: Al finalizar la función este parámetro indica si el producto maneja pedimentos o no.

aCaracteristicas: Al finalizar la función este parámetro indica si el producto maneja características o no.

Descripción Esta función define el tipo de producto, indicando si maneja series, lotes, pedimentos, unidades y/o características.

Ejemplo El siguiente código indica a la aplicación que recupere las cualidades del producto especificado.

```

IError = fRecuperaTipoProducto(IUnidades, ISerie, ILote, IPedimento, ICaracteristicas)

If IError <> 0 Then
    MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRecosteoProducto ()

Disponibilidad AdminPAQ 2004.

Sintaxis fRecosteoProducto (*aCodigoProducto, aEjercicio, aPeriodo, aCodigoClasificacion1, aCodigoClasificacion2, aCodigoClasificacion3, aCodigoClasificacion4, aCodigoClasificacion5, aCodigoClasificacion6, aNombreBitacora, aSobreEscribirBitacora, aEsCalculoAritmetico*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aEjercicio</i>	Entero	Por valor	Ejercicio a partir del cual se recostea.
	<i>aPeriodo</i>	Entero	Por valor	Periodo a partir del cual se recostea.
	<i>aCodigoClasificacion1</i>	Cadena	Por valor	Código de la clasificación 1.
	<i>aCodigoClasificacion2</i>	Cadena	Por valor	Código de la clasificación 2.
	<i>aCodigoClasificacion3</i>	Cadena	Por valor	Código de la clasificación 3.
	<i>aCodigoClasificacion4</i>	Cadena	Por valor	Código de la clasificación 4.
	<i>aCodigoClasificacion5</i>	Cadena	Por valor	Código de la clasificación 5.
	<i>aCodigoClasificacion6</i>	Cadena	Por valor	Código de la clasificación 6.
	<i>aNombreBitacora</i>	Cadena	Por valor	Nombre de la bitácora a escribir.
	<i>aSobreEscribirBitacora</i>	Entero	Por valor	Sobrescribir la bitácora, si ésta existe. 0 - Sobrescribe la bitácora 1 - No Sobrescribe la bitácora
	<i>aEsCalculoAritmetico</i>	Entero	Por valor	Determina el tipo de calculo para el costo promedio. 0 = MegaPAQ. 1 = AdminPAQ.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función recostea un producto determinado a partir de un periodo y un ejercicio.

Ejemplo El siguiente código recostea un producto a partir de un periodo y un ejercicio; sobrescribe la bitácora, si esta existe y utiliza el costo promedio de AdminPAQ.

```

IError = fRecosteoProducto (ICodigoProducto, IEjercicio, IPeriodo, ICodigoClasificacion1,
ICodigoClasificacion2, ICodigoClasificacion3, ICodigoClasificacion4, ICodigoClasificacion5,
ICodigoClasificacion6, INombreBitacora, 0, 1)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresaPrecioVenta ()

Disponibilidad AdminPAQ 2005.

Sintaxis fRegresaPrecioVenta (*aCodigoConcepto*, *aCodigoCliente*, *aCodigoProducto*, *aPrecioVenta*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoConcepto</i>	Cadena	Por valor	Código del concepto.
	<i>aCodigoCliente</i>	Cadena	Por valor	Código del cliente.
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aPrecioVenta</i>	Cadena	Por referencia	Precio de venta.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aPrecioVenta*: Al finalizar la función este parámetro contiene el precio de venta del producto solicitado.
 Esta función obtiene el precio de venta de un producto de un determinado cliente para un concepto de documento en específico.

Ejemplo El siguiente código obtiene el precio de venta de un producto de un determinado cliente para un concepto de documento en específico.

```

IError = fRegresaPrecioVenta (ICodigoConcepto, ICodigoCliente, ICodigoProducto,
aPrecioVenta)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaProducto ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBuscaProducto (*aCodProducto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodProducto</i>	Cadena	Por valor	Código del producto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un producto por su código.

Ejemplo El siguiente código busca un producto por su código.

```
fBuscaProducto (ICodProducto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaIdProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaIdProducto (*aldProducto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldProducto</i>	Entero	Por valor	Identificador del producto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un producto por su Identificador.

Ejemplo El siguiente código busca un producto por su identificador.

```
fBuscaIdProducto (IIdProducto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosPrimerProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el primer registro de la tabla de Productos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Productos.

```
IError = fPosPrimerProducto ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fPosUltimoProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosUltimoProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el ultimo registro de la tabla de Productos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Productos.

```
IError = fPosUltimoProducto ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fPosSiguienteProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosSiguienteProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Productos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Productos. <pre>IError = fPosSiguienteProducto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosAnteriorProducto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Productos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Productos. <pre>IError = fPosAnteriorProducto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosBOFProducto ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Productos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable InicioTablaProductos el resultado de la función fPosBOFProducto. <pre>InicioTablaProductos = fPosBOFProducto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFProducto ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosEOFProducto ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Productos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable IFinTablaProductos el resultado de la función fPosEOFProducto. <pre>IFinTablaProductos = fPosEOFProducto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Alto Nivel – Lectura/Escritura

fAltaProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaProducto (*aldProducto*, *astProducto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldProducto</i>	Entero	Por referencia	Identificador del producto.
	<i>astProducto</i>	tProducto	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldProducto*: Al finalizar la función este parámetro contiene el identificador del nuevo producto.

Esta función da de alta un nuevo Producto.

Ejemplo El siguiente código da de alta un nuevo producto.

```
fAltaProducto (lIdProducto, ltProducto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fActualizaProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fActualizaProducto (*aCodigoProducto*, *astCteProv*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Entero largo	Por referencia	Código del producto.
	<i>astProducto</i>	tProducto	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función actualiza un producto.

Ejemplo El siguiente código actualiza un producto.

```
fActualizaProducto (lCodigoProducto, ltProducto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLlenaRegistroProducto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLlenaRegistroCteProv (*astProducto*, *aEsAlta*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astProducto</i>	tProducto	Por valor	Tipo de dato abstracto.
	<i>aEsAlta</i>	Entero	Por valor	1 = Nuevo Producto. 2 = Actualizacion Producto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asigna al registro de la tabla de productos los valores de la estructura de datos *astCteProv*.

Ejemplo El siguiente código da de alta un nuevo producto en la tabla de productos.

```
fLlenaRegistroProducto (astProducto, 1)
```

Ver también fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),

FUNCIONES DE ADDENDAS

Bajo Nivel – Lectura/Escritura

flnseraDatoCompEducativo ()

Disponibilidad AdminPAQ 7.2.1. CONTPAQ i® FACTURA ELECTRÓNICA 1.0.1

Sintaxis flnseraDatoCompEducativo(int aldServicio, int aNumCampo, char *aDato)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldServicio</i>	Entero	Por valor	Identificador del servicio
	<i>aNumCampo</i>	Entero	Por valor	Número de campo
	<i>aDato</i>	Cadena	Por referencia	Valor a insertar

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Esta función inserta un registro correspondiente a los datos adicionales para el complemento educativo del catálogo servicios.

Ejemplo El siguiente código indica a la aplicación que inserten los datos adicionales del complemento educativo.

```
flnseraDatoCompEducativo (aldServicio, aNumCampo, aDato)
```

Ver también flnseraDatoAddendaDocto(), fObtieneLicencia(), fObtienePassProxy().

flnseraDatoAddendaDocto ()

Disponibilidad AdminPAQ 7.3.2. CONTPAQ i® FACTURA ELECTRÓNICA 2.5.1

Sintaxis flnseraDatoAddendaDocto(aldAddenda, aldCatalogo, aNumCampo, aDato)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldAddenda</i>	Entero	Por valor	Identificador de la Addenda
	<i>aldCatalogo</i>	Entero	Por valor	Identificador del documento
	<i>aNumCampo</i>	Entero	Por valor	Número de campo
	<i>aDato</i>	Cadena	Por referencia	Valor a insertar

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Agrega los datos de la addenda para los documentos.

Ejemplo El siguiente código indica a la aplicación que inserte los datos de la addenda para los documentos.

```
flnseraDatoAddendaDocto(aldAddenda, aldCatalogo, aNumCampo, aDato)
```

Ver también flnseraDatoCompEducativo (), fObtieneLicencia(), fObtienePassProxy().

fObtieneLicencia ()

Disponibilidad AdminPAQ 7.3.0. CONTPAQ i® FACTURA ELECTRÓNICA 2.2.1

Sintaxis fObtieneLicencia (aCodActiva, aCodSitio, aSerie, aTagVersion)

Parámetros	Nombre	Tipo	Uso	Descripción
	aCodActiva	Cadena	Por referencia	Variable en la que regresa el código de activación del Sistema
	aCodSitio	Cadena	Por referencia	Variable en la que regresa el código de sitio del Sistema
	aSerie	Cadena	Por referencia	Variable en la que regresa el número de serie del Sistema
	aTagVersion	Cadena	Por referencia	Variable en la que regresa el versión del Sistema

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Esta función regresa la licencia del producto.

Nota: Antes de llamar la función fObtieneLicencia se deberá llamar la función fInicializaLicenseInfo.

Ejemplo El siguiente código regresa la licencia del producto.

```
fInicializaLicenseInfo(aSistema)
fObtieneLicencia (aCodActiva, aCodSitio, aSerie, aTagVersion)
```

Ver también fInicializaLicenseInfo(), fInsertaDatoCompEducativo (), fInsertaDatoAddendaDocto(), fObtienePassProxy().

fObtienePassProxy ()

Disponibilidad AdminPAQ 7.3.0. CONTPAQ i® FACTURA ELECTRÓNICA 2.2.1

Sintaxis fObtienePassProxy(aPassProxy)

Parámetros	Nombre	Tipo	Uso	Descripción
	aPassProxy	Cadena	Por referencia	Variable en la que regresa la contraseña del Proxy

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Regresa la contraseña del Proxy.

Ejemplo El siguiente código regresa la contraseña del Proxy.

```
fObtienePassProxy (aPassProxy)
```

Ver también fInsertaDatoCompEducativo (), fInsertaDatoAddendaDocto(), fObtieneLicencia().

fTimbraXML ()

Disponibilidad AdminPAQ 7.3.2

Sintaxis fTimbraXML(char *aRutaXML, char *aCodConcepto, char *aUUID, char *aRutaDDA, char *aRutaResultado, char *aPass, char *aRutaFormato);

Parámetros	Nombre	Tipo	Uso	Descripción
	aRutaXML	Cadena	Por referencia	Variable donde se especifica la ruta y archivo donde está ubicado el XML a timbrar.
	aCodConcepto	Cadena	Por referencia	Variable donde se pasa el código del concepto a utilizar para timbrar el XML. Este concepto deberá estar configurado como CFDI.
	aUUID	Cadena	Por referencia	Variable donde se regresa el UUID del XML timbrado.
	aRutaDDA	Cadena	Por referencia	Variable donde se especifica la ruta y archivo DDA que contiene información adicional del XML.
	aRutaResultado	Cadena	Por referencia	Variable donde se especifica la ruta donde se generará el XML, HTML y las imágenes para la entrega en formato amigable.
	aPass	Cadena	Por referencia	Variable donde se especifica la contraseña del certificado para timbrar el XML.
	aRutaFormato	Cadena	Por referencia	Variable con la ruta y archivo del formato de impresión.

Retorna Valores enteros:
kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Esta función timbra un XML creado con una aplicación de un tercero.

El XML deberá estar sin emitir, sin sello y sin certificado.

Esta función requiere una licencia de 2 o más usuarios. Si cuentas con un licenciamiento anual además se requiere que la licencia sea multiempresa.

Ejemplo El siguiente código timbra un XML.

```
fInicializaLicenseInfo(aSistema)

aRutaXML = 'C:\CompacW\Empresas\adPrueba\F001.XML'
aCodConcepto = 4
aRutaDDA = 'C:\CompacW\Empresas\adPrueba\F001.DDA'
aPass = '12345678a'
aRutaFormato = 'C:\CompacW\Empresas\adPrueba\Plantilla_Factura_CFDI.htm'

fTimbraXML(aRutaXML, aCodConcepto, aUUID, aRutaDDA, aRutaResultado, aPass,
aRutaFormato )
```

Ver también fInicializaLicenseInfo(), fInsertaDatoCompEducativo (),fInsertaDatoAddendaDocto(), fObtieneLicencia().

fTimbraNominaXML ()

Disponibilidad AdminPAQ 7.3.3

Sintaxis fTimbraNominaXML(char *aRutaXML, char *aCodConcepto, char *aUUID, char *aRutaDDA, char *aRutaResultado, char *aPass, char *aRutaFormato);

Parámetros	Nombre	Tipo	Uso	Descripción
	aRutaXML	Cadena	Por referencia	Variable donde se especifica la ruta y archivo donde está ubicado el XML a timbrar.
	aCodConcepto	Cadena	Por referencia	Variable donde se pasa el código del concepto a utilizar para timbrar el XML. Este concepto deberá estar configurado como CFDI.
	aUUID	Cadena	Por referencia	Variable donde se regresa el UUID del XML timbrado.
	aRutaDDA	Cadena	Por referencia	Variable donde se especifica la ruta y archivo DDA que contiene información adicional del XML.
	aRutaResultado	Cadena	Por referencia	Variable donde se especifica la ruta donde se generará el XML, HTML y las imágenes para la entrega en formato amigable.
	aPass	Cadena	Por referencia	Variable donde se especifica la contraseña del certificado para timbrar el XML.
	aRutaFormato	Cadena	Por referencia	Variable con la ruta y archivo del formato de impresión.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Esta función timbra un XML de una nómina creado con una aplicación de un tercero.

El XML deberá estar sin emitir, sin sello y sin certificado.
 Es obligatorio que el XML lleve el domicilio del emisor.
 Si deseas ver en la impresión del formato amigable algún dato del complemento de nómina se deberá insertar en el DDA.
 Esta función requiere una licencia de 5 o más usuarios. Si cuentas con un licenciamiento anual además se requiere que la licencia sea multiempresa.

Ejemplo El siguiente código timbra un XML.

```
fInicializaLicenseInfo(aSistema)

aRutaXML = 'C:\CompacW\Empresas\adPrueba\F001.XML'
aCodConcepto = 4
aRutaDDA = 'C:\CompacW\Empresas\adPrueba\F001.DDA'
aPass = '12345678a'
aRutaFormato = 'C:\CompacW\Empresas\adPrueba\Plantilla_Factura_CFDI.htm'

fTimbraNominaXML(aRutaXML, aCodConcepto, aUUID, aRutaDDA, aRutaResultado,
aPass, aRutaFormato )
```

Ver también fInicializaLicenseInfo(), fInsertaDatoCompEducativo (),fInsertaDatoAddendaDocto(), fObtieneLicencia().

FUNCIONES DE DIRECCIONES

Bajo Nivel – Lectura/Escritura

fInsertaDireccion ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fInsertaDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en la tabla de Direcciones en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Direcciones.
	<pre>fInsertaDireccion ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaDireccion ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fEditaDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Activa el modo de Edición de un registro en la tabla de Direcciones.
Ejemplo	El siguiente código busca la dirección de la empresa, si la encuentra activa el registro en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente
	<pre>!Error = fBuscaDireccionEmpresa () If !Error <> 0 Then MensajeError !Error Else fEditaDireccion () End If</pre>
Ver también	fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),

fGuardaDireccion ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fGuardaDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro de productos.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Direcciones. Esta función se llama después de que se utiliza la función fInsertaDireccion () o fEditaDireccion () y se graban los valores en los campos correspondientes.

```
fGuardaDireccion ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fCancelarModificacionDireccion ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fCancelarModificacionDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual de direcciones. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro de dirección que estaba en modo de inserción o edición.

```
fCancelarModificacionDireccion ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fLeeDatoDireccion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoDireccion (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de Direcciones.

Ejemplo El siguiente código busca la dirección de la empresa, en caso de que lo encuentre guarda el contenido del campo cColonia en la variable la variable lColonia con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaDireccionEmpresa ()
If lError <> 0 Then
  MensajeError lError
Else
  lError = fLeeDatoDireccion ("cColonia", lColonia, 60)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoDireccion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fSetDatoDireccion (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de Cliente / Proveedor.

Ejemplo El siguiente código busca la dirección de la empresa, en caso de que la encuentre escribe el contenido de la variable lColonia en el campo cColonia de la tabla de direcciones; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaDireccionEmpresa ()
If lError <> 0 Then
  MensajeError lError
Else
  lError = fSetDatoDireccion ("cColonia ", lColonia)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaDireccionEmpresa ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fBuscaDireccionEmpresa ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función busca la dirección de la empresa.
Ejemplo	El siguiente código indica a la aplicación que busque la dirección de la empresa.
	<pre>IError = fBuscaDireccionEmpresa ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaDireccionCteProv ()

Disponibilidad	AdminPAQ 2002.			
Sintaxis	fBuscaDireccionCteProv (aCodCteProv, aTipoDireccion)			
Parámetros	Nombre	Tipo	Uso	Descripción
	aCampo	Cadena	Por valor	Código del cliente/proveedor.
	aValor	Cadena	Por valor	Tipo de dirección 0 = Fiscal, 1 = Envío
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca una dirección de un cliente/proveedor.			
Ejemplo	El siguiente código busca una dirección de un cliente/proveedor, en caso de que la encuentre se posiciona en el registro encontrado; en caso contrario muestra el mensaje de error correspondiente.			
	<pre>IError = fBuscaDireccionCteProv (ICodCteProv, ITipoDireccion) If IError <> 0 Then MensajeError IError End If</pre>			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fBuscaDireccionDocumento ()

Disponibilidad AdminPAQ 2002.

Sintaxis fBuscaDireccionDocumento (*aldDocumento*, *aTipoDireccion*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDocumento</i>	Entero largo	Por valor	Identificador del documento.
	<i>aValor</i>	Cadena	Por valor	Tipo de dirección 0 = Fiscal, 1 = Envío

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca una dirección de un documento.

Ejemplo El siguiente código busca una dirección de un documento, en caso de que la encuentre se posiciona en el registro encontrado; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaDireccionDocumento (lIdDocumento, lTipoDireccion)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerDireccion ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosPrimerDireccion ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de Direcciones.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Direcciones.

```
IError = fPosPrimerDireccion ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimaDireccion ()

Disponibilidad AdminPAQ 2002.

Sintaxis fPosUltimaDireccion ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de Direcciones.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Direcciones.

```
IError = fPosUltimaDireccion ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteDireccion ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosSiguienteDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Direcciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Direcciones.
	<pre>!Error = fPosSiguienteDireccion ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorDireccion ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosAnteriorDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Direcciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Direcciones.
	<pre>!Error = fPosAnteriorDireccion ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFDireccion ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosBOFDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Direcciones.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaDirs el resultado de la función fPosBOFDireccion. <pre>lInicioTablaDirs = fPosBOFDireccion ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFDireccion ()

Disponibilidad	AdminPAQ 2002.
Sintaxis	fPosEOFDireccion ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Direcciones
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaDirs el resultado de la función fPosEOFDireccion. <pre>lFinTablaDirs = fPosEOFDireccion ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Alto Nivel – Lectura/Escritura

fAltaDireccion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaDireccion (*aldDireccion*, *astDireccion*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldDireccion</i>	Entero	Por referencia	Identificador de la dirección.
	<i>astDireccion</i>	tDireccion	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldDireccion*: Al finalizar la función este parámetro contiene el identificador del nuevo producto.
 Esta función da de alta una nueva dirección.

Ejemplo El siguiente código da de alta una nueva dirección.

```
fAltaDireccion (lIdDireccion, ltDireccion)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Importante: Al usar esta función de alto nivel es necesario asignar al campo **cTipoDireccion** alguno de los siguientes valores: 1 = Domicilio Fiscal, 2 = Domicilio Envío

fActualizaDireccion ()

Disponibilidad AdminPAQ 2004.

Sintaxis fActualizaProducto (*astDireccion*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astDireccion</i>	tDireccion	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función actualiza la dirección del registro de Cliente/Proveedor activo.

Ejemplo El siguiente código actualiza la dirección del registro de Cliente/Proveedor activo.

```
fActualizaProducto (ltDireccion)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Importante: Al usar esta función de alto nivel es necesario asignar al campo **cTipoDireccion** alguno de los siguientes valores: 1 = Domicilio Fiscal, 2 = Domicilio Envío

fLlenaRegistroDireccion ()

Disponibilidad AdminPAQ 2004.

Sintaxis fLlenaRegistroDireccion (*astDireccion*, *aEsAlta*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astDireccion</i>	tDireccion	Por valor	Tipo de dato abstracto.
	<i>aEsAlta</i>	Entero	Por valor	1 = Nueva dirección. 2 = Actualización.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asigna al registro de la base de datos los valores de la estructura de datos de la Dirección.

Ejemplo El siguiente código da de alta una nueva dirección.

```
fLlenaRegistroDireccion (ltDireccion, 1)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Importante: Al usar esta función de alto nivel es necesario asignar al campo **cTipoDireccion** alguno de los siguientes valores: 1 = Domicilio Fiscal, 2 = Domicilio Envío

FUNCIONES DE EXISTENCIAS

Bajo Nivel – Lectura/Escritura

fRegresaExistencia ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaExistencia (*aCodigoProducto*, *aCodigoAlmacen*, *aAnio*, *aMes*, *aDia*, *aExistencia*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén.
	<i>aAnio</i>	Cadena	Por valor	Año.
	<i>aMes</i>	Cadena	Por valor	Mes.
	<i>aDia</i>	Cadena	Por valor	Día.
	<i>aExistencia</i>	Doble	Por referencia	Existencia

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aExistencia: Al finalizar la función este parámetro contiene la existencia del producto requerido.

Descripción Esta función regresa la existencia de un producto en un almacén a una determinada fecha.

Ejemplo El siguiente código obtiene la existencia de determinado producto en cierto almacén, en la fecha especificada.

```

IError = fRegresaExistencia (ICodigoProducto, ICodigoAlmacen, IAnio, IMes, IDia, IExistencia)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresaExistenciaCaracteristicas ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaExistenciaCaracteristicas (*aCodigoProducto*, *aCodigoAlmacen*, *aAnio*, *aMes*, *aDia*, *aValorCaracteristica1*, *aValorCaracteristica2*, *aValorCaracteristica3*, *aExistencia*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén.
	<i>aAnio</i>	Cadena	Por valor	Año.
	<i>aMes</i>	Cadena	Por valor	Mes.
	<i>aDia</i>	Cadena	Por valor	Día.
	<i>aValorCaracteristica1</i>	Cadena	Por valor	Valor característica 1.
	<i>aValorCaracteristica2</i>	Cadena	Por valor	Valor característica 2.
	<i>aValorCaracteristica3</i>	Cadena	Por valor	Valor característica 3.
	<i>aExistencia</i>	Doble	Por referencia	Existencia

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aExistencia: Al finalizar la función este parámetro contiene la existencia del producto requerido.

Descripción Esta función regresa la existencia de un producto con características en un almacén a una determinada fecha.

Ejemplo El siguiente código obtiene la existencia de determinado producto con características en cierto almacén, en la fecha especificada.

```

IError = fRegresaExistenciaCaracteristicas (ICodigoProducto, ICodigoAlmacen, IAnio, IMes, IDia,
IValorCaracteristica1, IValorCaracteristica2, IValorCaracteristica3, IExistencia)
If IError <> 0 Then
  MensajeError IError
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES DE COSTO HISTÓRICO

Bajo Nivel – Lectura/Escritura

fRegresaCostoPromedio ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaCostoPromedio (*aCodigoProducto*, *aCodigoAlmacen*, *aAnio*, *aMes*, *aDia*, *aCostoPromedio*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén.
				0 (cero) – Todos los almacenes.
	<i>aAnio</i>	Cadena	Por valor	Año.
	<i>aMes</i>	Cadena	Por valor	Mes.
	<i>aDia</i>	Cadena	Por valor	Día.
	<i>aCostoPromedio</i>	Cadena	Por referencia	Costo promedio

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aCostoPromedio: Al finalizar la función este parámetro contiene el costo promedio del producto requerido.

Descripción Esta función se encarga de obtener el costo promedio de un producto en determinada fecha para todos los almacenes o para uno solo.

Ejemplo El siguiente código obtiene el costo promedio de un producto en una fecha para todos los almacenes o para uno solo.

```

IError = fRegresaCostoPromedio (ICodigoProducto, ICodigoAlmacen, IAnio, IMes, IDia, ICostoP)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresaUltimoCosto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaUltimoCosto (*aCodigoProducto*, *aCodigoAlmacen*, *aAnio*, *aMes*, *aDia*, *aUltimoCosto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén. 0 (cero) – Todos los almacenes.
	<i>aAnio</i>	Cadena	Por valor	Año.
	<i>aMes</i>	Cadena	Por valor	Mes.
	<i>aDia</i>	Cadena	Por valor	Día.
	<i>aUltimoCosto</i>	Cadena	Por referencia	Ultimo costo.

Retorna Valores enteros:

kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.

!kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aUltimoCosto: Al finalizar la función este parámetro contiene el ultimo costo del producto requerido.

Descripción Esta función se encarga de obtener el último costo de un producto en determinada fecha para todos los almacenes o para uno solo.

Ejemplo El siguiente código obtiene el último costo de un producto en una fecha para todos los almacenes o para uno solo.

```

IError = fRegresaCostoPromedio (ICodigoProducto, ICodigoAlmacen, IAnio, IMes, IDia, ICostoU)
If IError <> 0 Then
    MensajeError IError
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresaCostoEstandar ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaCostoEstandar (*aCodigoProducto*, *aCostoEstandar*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCostoEstandar</i>	Cadena	Por referencia	Costo estándar.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aCostoEstandar: Al finalizar la función este parámetro contiene el costo estándar del producto requerido.

Descripción Esta función se encarga de obtener el costo estándar de un producto.

Ejemplo El siguiente código obtiene el costo estándar de un producto.

```

IError = fRegresaCostoEstandar (ICodigoProducto, ICostoEstandar)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresaCostoCapa ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresaCostoCapa (*aCodigoProducto*, *aCodigoAlmacen*, *aUnidades*, *alImporteCosto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoProducto</i>	Cadena	Por valor	Código del producto.
	<i>aCodigoAlmacen</i>	Cadena	Por valor	Código del almacén.
	<i>aUnidades</i>	Doble	Por valor	Unidades a costear.
	<i>alImporteCosto</i>	Cadena	Por referencia	Importe del costo de la unidades recibidas.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

alImporteCosto: Al finalizar la función este parámetro contiene el costo UEPS o PEPS del producto requerido.

Descripción Esta función obtiene el costo UEPS o PEPS de un producto en un almacén en base a una cantidad de unidades proporcionadas.

Ejemplo El siguiente código obtiene el costo UEPS o PEPS de un producto en un almacén en base a una cantidad de unidades proporcionadas.

```

IError = fRegresaCostoCapa (ICodigoProducto, ICodigoAlmacen, IUnidades, IImporteCosto)
If IError <> 0 Then
  MensajeError IError
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES DE CONCEPTOS DE DOCUMENTOS

Bajo Nivel – Lectura/Escritura

fLeeDatoConceptoDocto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoConceptoDocto (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee un campo del registro actual de conceptos documentos.

Ejemplo El siguiente código busca un concepto por su código, en caso de que lo encuentre guarda el contenido del campo **cNombreC01** en la variable la variable **lNombreC** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaConceptoDocto (lCodConcepto)
If lError <> 0 Then
  MensajeError lError
Else
  lError = fLeeDatoConceptoDocto ("cNombreC01", lNombreC, 60)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fRegresPorcentajImpuesto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fRegresPorcentajImpuesto (*aldConceptoDocumento*, *aldClienteProveedor*, *aldProducto*, *aPorcentajImpuesto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldConceptoDocumento</i>	Entero	Por valor	Identificador del concepto del documento.
	<i>aldClienteProveedor</i>	Entero	Por valor	Identificador del cliente o proveedor.
	<i>aldProducto</i>	Entero	Por valor	Identificador del producto.
	<i>aPorcentajImpuesto</i>	Doble	Por referencia	Porcentaje de impuesto.

Retorna Valores enteros:
 KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aPorcentajImpuesto*: Al finalizar la función este parámetro contiene el porcentaje del impuesto requerido.
 Esta función regresa el porcentaje de impuesto de un concepto documento, del cual se obtiene su configuración y se busca el porcentaje de la tabla de Clientes/Proveedores, Productos o de Parámetros generales.

Ejemplo El siguiente código obtiene el porcentaje de impuesto para el concepto de un documento.

```
lError = fRegresPorcentajImpuesto (lIdConceptoDocumento, lIdClienteProveedor, lIdProducto, lPorcentajImpuesto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaConceptoDocto()

Disponibilidad AdminPAQ 2011. Versión 6.2.3

Sintaxis *fEditaConceptoDocto ()*

Parámetros No recibe parámetros.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función activa el modo de edición de un registro del catálogo Conceptos.

Ejemplo El siguiente código:

- Busca el concepto con la función fBuscaConceptoDocto.
- Abre la edición de ese concepto con la función fEditaConceptoDocto.
- Asigna al campo **CSERIEPO01** el contenido de la variable **ISerie**.
- Asigna al campo **CIDDIRSUCU** el contenido de la variable **IDireccion**.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
error = fBuscaConceptoDocto(lConcepto)
error = fEditaConceptoDocto
        error = fSetDatoConceptoDocto("CSERIEPO01", ISerie)
        error = fSetDatoConceptoDocto("CIDDIRSUCU", IDireccion)
error = fGuardaConceptoDocto()
```

Ver también [fBuscaConceptoDocto\(\)](#), [fSetDatoConceptoDocto\(\)](#), [fGuardaConceptoDocto\(\)](#)

fSetDatoConceptoDocto ()

Disponibilidad AdminPAQ 2011. Versión 6.2.3

Sintaxis *fSetDatoConceptoDocto (const char *aCampo, char *aValor)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por referencia	Nombre del campo
	<i>aValor</i>	Cadena	Por referencia	Valor del campo
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla Conceptos.			
Ejemplo	El siguiente código:			

- Busca el concepto con la función fBuscaConceptoDocto.
- Abre la edición de ese concepto con la función fEditaConceptoDocto.
- Asigna al campo **CSERIEPO01** el contenido de la variable **ISerie**.
- Asigna al campo **CIDDIRSUCU** el contenido de la variable **IDireccion**.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
error = fBuscaConceptoDocto(lConcepto)
error = fEditaConceptoDocto
    error = fSetDatoConceptoDocto("CSERIEPO01", ISerie)
    error = fSetDatoConceptoDocto("CIDDIRSUCU", IDireccion)
error = fGuardaConceptoDocto()
```

Ver también [fBuscaConceptoDocto\(\)](#), [fEditaConceptoDocto\(\)](#), [fGuardaConceptoDocto\(\)](#)

fGuardaConceptoDocto()

Disponibilidad AdminPAQ 2011. Versión 6.2.3

Sintaxis *fGuardaConceptoDocto()*

Parámetros No recibe parámetros.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función guarda los cambios efectuados al registro de la tabla Conceptos.

Ejemplo El siguiente código:

- Busca el concepto con la función fBuscaConceptoDocto.
- Abre la edición de ese concepto con la función fEditaConceptoDocto.
- Asigna al campo **CSERIEPO01** el contenido de la variable **ISerie**.
- Asigna al campo **CIDDIRSUCU** el contenido de la variable **IDireccion**.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
error = fBuscaConceptoDocto(lConcepto)
error = fEditaConceptoDocto
        error = fSetDatoConceptoDocto("CSERIEPO01", ISerie)
        error = fSetDatoConceptoDocto("CIDDIRSUCU", IDireccion)
error = fGuardaConceptoDocto()
```

Ver también [fBuscaConceptoDocto\(\)](#), [fEditaConceptoDocto\(\)](#), [fSetDatoConceptoDocto\(\)](#)

Bajo Nivel – Búsqueda/Navegación

fBuscaConceptoDocto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaConceptoDocto (*aCodConcepto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodConcepto</i>	Cadena	Por valor	Código del concepto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un concepto por su código.

Ejemplo El siguiente código busca un concepto por su código.

```
fBuscaConceptoDocto (lCodConcepto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaldConceptoDocto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaldConceptoDocto (*aldConcepto*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldConcepto</i>	Entero	Por valor	Identificador del concepto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un concepto por su Identificador.

Ejemplo El siguiente código busca un concepto por su identificador.

```
fBuscaldConceptoDocto (lIdConcepto)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerConceptoDocto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosPrimerConceptoDocto ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de Conceptos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Conceptos.

```
IError = fPosPrimerConceptoDocto ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimaConceptoDocto ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosUltimaConceptoDocto ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de Conceptos.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Conceptos.

```
IError = fPosUltimaConceptoDocto ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteConceptoDocto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosSiguienteConceptoDocto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Conceptos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Conceptos. <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> IError = fPosSiguienteConceptoDocto () </div>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorConceptoDocto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosAnteriorConceptoDocto ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Conceptos.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Conceptos. <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> IError = fPosAnteriorConceptoDocto () </div>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFConceptoDocto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosBOFConceptoDocto ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Conceptos.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaConcepto el resultado de la función fPosBOFConceptoDocto .
	<pre>lInicioTablaConcepto = fPosBOFConceptoDocto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFConceptoDocto ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosEOFConceptoDocto ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Conceptos
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaConcepto el resultado de la función fPosEOFConceptoDocto .
	<pre>lFinTablaConcepto = fPosEOFConceptoDocto ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Funciones de Parametros

Bajo Nivel – Lectura/Escritura

fLeeDatoParametros ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoParametros (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee un campo del registro actual de parametros.

Ejemplo El siguiente código busca un concepto por su código, en caso de que lo encuentre guarda el contenido del campo cRFCEmpr01 en la variable la variable IRFC con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
!Error = fLeeDatoParametros ("cRFCEmpr01", IRFC, 60)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaParametros ()

Disponibilidad AdminPAQ 2.2.0 CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0

Sintaxis fEditaParametros ()

Parámetros No recibe parámetros.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función activa el modo de edición de un registro de los Parámetros.

Ejemplo El siguiente código:

- Abre la edición de los parámetros con la función fEditaParametros.
- Asigna al campo **cRFCEmpr01** el contenido de la variable **IRFC**.
- Guarda los cambios a los parámetros con la función fGuardaParametros.

```
!error = fEditaParametros
!error = fSetDatoParametros ("cRFCEmpr01", IRFC)
!error = fGuardaParametros ()
```

Ver también fLeeDatoParametros(), fGuardaParametros(), fSetDatoParametros()

fSetDatoParametros ()

Disponibilidad AdminPAQ 2.4.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0

Sintaxis *fSetDatoParametros(aCampo, aValor)*

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por referencia	Nombre del campo
	<i>aValor</i>	Cadena	Por referencia	Valor del campo

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla Parámetros.

Ejemplo El siguiente código:

- Abre la edición de los parámetros con la función fEditaParametros.
- Asigna al campo **cRFCEmpr01** el contenido de la variable **IRFC**.
- Guarda los cambios a los parámetros con la función fGuardaParametros.

```
error = fEditaParametros
      error = fSetDatoParametros ("cRFCEmpr01", IRFC)
error = fGuardaParametros ()
```

Ver también fLeeDatoParametros(), fGuardaParametros(), fSetDatoParametros()

fGuardaParametros ()

Disponibilidad AdminPAQ 2.4.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0

Sintaxis *fGuardaParametros()*

Parámetros No recibe parámetros.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función guarda los cambios efectuados al registro de la tabla Parámetros.

Ejemplo El siguiente código:

- Abre la edición de los parámetros con la función fEditaParametros.
- Asigna al campo **cRFCEmpr01** el contenido de la variable **IRFC**.
- Guarda los cambios a los parámetros con la función fGuardaParametros.

```
error = fEditaParametros
      error = fSetDatoParametros ("cRFCEmpr01", IRFC)
error = fGuardaParametros ()
```

Ver también fLeeDatoParametros(), fGuardaParametros(), fSetDatoParametros()

FUNCIONES DEL CATÁLOGO DE CLASIFICACIONES

Bajo Nivel – Lectura/Escritura

fEditaClasificacion ()

Disponibilidad AdminPAQ 2004.

Sintaxis fEditaClasificacion ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error

Descripción Activa el modo de Edición de un registro en la tabla de Clasificaciones.

Ejemplo El siguiente código clasificación por su identificador, si la encuentra activa el registro en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```

IError = fBuscaIdClasificacion (IdClasificacion)
If IError <> 0 Then
    MensajeError IError
Else
    fEditaClasificacion ()
End If
    
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fGuardaClasificacion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fGuardaClasificacion ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Guarda los cambios realizados a un registro de clasificaciones.

Ejemplo El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Clasificaciones. Esta función se llama después de que se utiliza la función fInsertaDireccion () o fEditaDireccion () y se graban los valores en los campos correspondientes.

```

fGuardaClasificacion ()
    
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelarModificacionClasificacion ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fCancelarModificacionClasificacion ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual de clasificaciones. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro de clasificaciones que estaba en modo de inserción o edición.

```
fCancelarModificacionClasificacion()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fActualizaClasificacion ()

Disponibilidad	AdminPAQ 2004.			
Sintaxis	fActualizaClasificacion (aClasificacionDe, aNumClasificacion, aNombreClasificacion)			
Parámetros	Nombre <i>aClasificacionDe</i>	Tipo Entero	Uso Por valor	Descripción Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacén 5 – Producto.
	<i>aNumClasificacion</i>	Entero	Por valor	Numero de la clasificacion (1-6)
	<i>aNombreClasificacion</i>	Cadena	Por valor	Texto a actualizar en la clasificación.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función actualiza la dirección del registro de Cliente/Proveedor activo.			
Ejemplo	El siguiente código actualiza la dirección del registro de Cliente/Proveedor activo.			
	<pre>fActualizaClasificacion (ltDireccion)</pre>			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fLeeDatoClasificacion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoClasificacion (*aCampo*, *aValr*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de Clasificaciones.

Ejemplo El siguiente código busca la dirección de la empresa, en caso de que lo encuentre guarda el contenido del campo **cNombreC01** en la variable la variable **IClasif** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaIdClasificacion ()
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoClasificacion ("cNombreC01", IClasif, 60)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoClasificacion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fSetDatoClasificacion (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de Clasificaciones.

Ejemplo El siguiente código busca la dirección de la empresa, en caso de que la encuentre escribe el contenido de la variable **IClasif** en el campo **cNombreC01** de la tabla de direcciones; en caso contrario muestra el mensaje de error correspondiente.

```
IError = fBuscaDireccionEmpresa ()
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoClasificacion ("cNombreC01 ", IClasif)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaClasificacion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaClasificacion (*aClasificacionDe*, *aNumClasificacion*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aClasificacionDe</i>	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacén 5 – Producto.
	<i>aNumClasificacion</i>	Entero	Por valor	Numero de la clasificacion (1-6)

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca una clasificacion de acuerdo a los parámetros recibidos y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca una clasificación.

```
fBuscaClasificacion (lClasificacionDe, lNumClasificacion)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaIdClasificacion ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaIdClasificacion (*aldClasificacion*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldClasificacion</i>	Entero	Por valor	Identificador del concepto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca un concepto por su Identificador.

Ejemplo El siguiente código busca un concepto por su identificador.

```
fBuscaIdClasificacion (lIdClasificacion)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0
Sintaxis	fPosPrimerClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el primer registro de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla Clasificaciones.

```
lError = fPosPrimerClasificacion ()
```

Ver también fBuscaIdClasificacion(), fBuscaClasificacion()

fPosUltimoClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0.
Sintaxis	fPosUltimoClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el último registro de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el último registro de la tabla Clasificaciones.

```
fPosUltimoClasificacion ()
```

Ver también fBuscaIdClasificacion(), fBuscaClasificacion()

fPosSiguienteClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0.
Sintaxis	fPosSiguienteClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla Clasificaciones.

```
lError = fPosSiguienteClasificacion ()
```

Ver también fBuscaIdClasificacion(), fBuscaClasificacion()

fPosAnteriorClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0.
Sintaxis	fPosAnteriorClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla Clasificaciones.

```
IError = fPosAnteriorClasificacion ()
```

Ver también fBuscaldClasificacion(), fBuscaClasificacion()

fPosBOFClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0.
Sintaxis	fPosBOFClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable InicioTablaVC el resultado de la función fPosBOFClasificacion .

```
InicioTablaC = fPosBOFClasificacion ()
```

Ver también fBuscaldClasificacion(), fBuscaClasificacion()

fPosEOFClasificacion ()

Disponibilidad	AdminPAQ 2.2.0, CONTPAQ i® FACTURA ELECTRÓNICA 1.0.0..
Sintaxis	fPosEOFClasificacion()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla Clasificaciones.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable IFinTablaVC el resultado de la función fPosEOFClasificacion .

```
IFinTablaC = fPosEOFClasificacion()
```

Ver también fBuscaldClasificacion(), fBuscaClasificacion()

FUNCIONES DEL CATÁLOGO DE VALORES DE CLASIFICACIONES

Bajo Nivel – Lectura/Escritura

fInsertaValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fInsertaValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en la tabla de Valores de Clasificación en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Valores de Clasificación.
	<pre>fInsertaValorClasif ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fEditaValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Activa el modo de Edición de un registro en la tabla de Valores de Clasificación.
Ejemplo	El siguiente código busca un cliente/proveedor por su Identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente
	<pre>Error = fBuscaIdValorClasif (IdValorClasif) If !Error <> 0 Then MensajeError !Error Else fEditaValorClasif () End If</pre>
Ver también	fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),

fGuardaValorClasif ()

Disponibilidad AdminPAQ 2002.

Sintaxis fGuardaValorClasif ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Guarda los cambios realizados a un registro de Valores de Clasificación.

Ejemplo El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Valores de Clasificación. Esta función se llama después de que se utiliza la función **fInsertaValorClasif ()** o **fEditaValorClasif ()** y se graban los valores en los campos correspondientes.

```
fGuardaValorClasif ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBorraValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBorraValorClasif ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Borra un registro en la tabla de Valores de Clasificación.

Ejemplo El siguiente código busca un documento por su identificador, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```
IError = fBuscaIdValorClasif (IdValorClasif)
If IError <> 0 Then
    MensajeError IError
Else
    fBorraValorClasif ()
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelarModificacionValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fCancelarModificacionValorClasif ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela las modificaciones al registro actual de Valores de Clasificación. El registro debe estar en modo de edición o inserción.

Ejemplo El siguiente código indica a la aplicación que cancele la modificación a un registro de Valores de Clasificación que estaba en modo de inserción o edición.

```
fCancelarModificacionValorClasif ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEliminarValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fEliminarValorClasif (*aClasificacionDe*, *aNumClasificacion*, *aCodValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aClasificacionDe</i>	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacén 5 – Producto.
	<i>aNumClasificacion</i>	Entero	Por valor	Número de la clasificación (1-6)
	<i>aCodValorClasif</i>	Cadena	Por valor	Código del Valor Clasificación Producto

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función elimina un registro de la tabla Valores de Clasificación usando su código.

Ejemplo El siguiente código elimina un registro de la tabla Valores de Clasificación, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```

IError = fEliminarValorClasif (IClasificacionDe, INumClasificacion, ICodValorClasif)

If IError <> 0 Then
    MensajeError IError
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoValorClasif ()

Disponibilidad AdminPAQ 2002.

Sintaxis fSetDatoValorClasif (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de Valores de Clasificación.

Ejemplo El siguiente código indica a la aplicación que busque un valor de clasificación por su identificador, en caso de que lo encuentre escribe el contenido de la variable **ICasif** en el campo **cldClasi01** de la tabla de Valores de Clasificación; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaldValorClasif (lIdValorClasif)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoValorClasif ("cldClasi01", ICasif)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fLeeDatoValorClasif ()

Disponibilidad AdminPAQ 2002.

Sintaxis fLeeDatoValorClasif (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de Valores de Clasificación.

Ejemplo El siguiente código indica a la aplicación que busque un valor de clasificación por su identificador, en caso de que lo encuentre escribe el contenido de l campo **cldClasi01** en la variable la variable **ICasif** con una longitud de 11 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscarIdMovimiento (IIdMovto)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoValorClasif ("cldClasi01", ICasif, 11)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaClasificacion (*aClasificacionDe*, *aNumClasificacion*, *aCodValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aClasificacionDe</i>	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacén 5 – Producto.
	<i>aNumClasificacion</i>	Entero	Por valor	Numero de la clasificacion (1-6)
	<i>aCodValorClasif</i>	Cadena	Por valor	Código del Valor Clasificación Producto
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca una clasificacion de acuerdo a los parámetros recibidos y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca una clasificación.			

```
fBuscaClasificacion (lClasificacionDe, lNumClasificacion)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaIdValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaIdValorClasif (*aldValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldValorClasif</i>	Entero	Por valor	Identificador del valor de clasificación.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca un valor de clasificación por su Identificador y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca un valor de clasificación por su identificador.			

```
fBuscaIdValorClasif (lIdValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosPrimerValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el primer registro de la tabla de Valores de Clasificación.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Valores de Clasificación.

```
IError = fPosPrimerValorClasif ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fPosUltimoValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosUltimoValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !KSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el ultimo registro de la tabla de Valores de Clasificación.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Valores de Clasificación.

```
fPosUltimoValorClasif()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fPosSiguienteValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosSiguienteValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Valores de Clasificación.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Valores de Clasificación.
	<pre>IError = fPosSiguienteValorClasif ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosAnteriorValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Valores de Clasificación.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Valores de Clasificación.
	<pre>IError = fPosAnteriorValorClasif ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosBOFValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Valores de Clasificación.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaVC el resultado de la función fPosBOFValorClasif .
	<pre>lInicioTablaVC = fPosBOFValorClasif ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFValorClasif ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosEOFValorClasif ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Valores de Clasificación
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaVC el resultado de la función fPosEOFValorClasif .
	<pre>lFinTablaVC = fPosEOFValorClasif ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES DEL CATÁLOGO DE VALORES DE CLASIFICACIONES

Alto Nivel – Lectura/Escritura

fAltaValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaValorClasif (*aldValorClasif*, *astValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldValorClasif</i>	Entero	Por referencia	Identificador de la dirección.
	<i>astValorClasif</i>	tValorClasif	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción *aldValorClasif*: Al finalizar la función este parámetro contiene el identificador del nuevo valor de clasificación.
 Esta función da de alta un nuevo valor de clasificación.

Ejemplo El siguiente código da de alta un nuevo valor de clasificación.

```
fAltaValorClasif (lIdValorClasif, lstValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fActualizaValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fActualizaProducto (*aCodigoValorClasif*, *astValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCodigoValorClasif</i>	Cadena	Por valor	Código del valor de clasificación.
	<i>astValorClasif</i>	<i>tValorClasif</i>	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función actualiza el valor de clasificación del registro especificado por el parametro *aCodigoValorClasif*.

Ejemplo El siguiente código actualiza el valor de clasificación del registro especificado por el parametro *aCodigoValorClasif*.

```
fActualizaProducto (lCodigoValorClasif, lstValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLlenaRegistroValorClasif ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLlenaRegistroValorClasif (*astValorClasif*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astValorClasif</i>	<i>tValorClasif</i>	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asigna al registro de la base de datos los valores de la estructura e datos del valor de clasificacion.

Ejemplo El siguiente código asigna al registro de la base de datos los valores de la estructura e datos del valor de clasificacion.

```
fLlenaRegistroValorClasif (lstValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES CATÁLOGO DE UNIDADES DE MEDIDA Y PESO

Bajo Nivel – Lectura/Escritura

fInsertaUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fInsertaUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en la tabla de Unidades de Medida y Peso en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en la tabla de Unidades de Medida y Peso.

```
fInsertaUnidad ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fEditaUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fEditaUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Activa el modo de Edición de un registro en la tabla de Unidades de Medida y Peso.
Ejemplo	El siguiente código busca una unidad por su identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
!Error = fBuscaIdUnidad(IdUnidad)
If !Error <> 0 Then
    MensajeError !Error
Else
    fEditaUnidad ()
End If
```

Ver también	fInsertarDocumento(),fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(),fCancelarModificacionDocumento(),
--------------------	--

fGuardaUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fGuardaUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro de Unidades de Medida y Peso.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en la tabla de Unidades de Medida y Peso. Esta función se llama después de que se utiliza la función InsertaUnidad () o EditaUnidad() y se graban los valores en los campos correspondientes.
	<pre>fGuardaUnidad ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBorraUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fBorraUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Borra un registro en la tabla de Unidades de Medida y Peso.
Ejemplo	El siguiente código busca un documento por su identificador, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente
	<pre>IError = fBuscaIdUnidad(IIdUnidad) If IError <> 0 Then MensajeError IError Else fBorraUnidad () End If</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelarModificacionUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fCancelarModificacionUnidad ()

Parámetros No usa.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función cancela las modificaciones al registro actual de Unidades de Medida y Peso. El registro debe estar en modo de edición o inserción.

Ejemplo El siguiente código indica a la aplicación que cancele la modificación a un registro de Unidades de Medida y Peso que estaba en modo de inserción o edición.

```
fCancelarModificacionUnidad ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEliminarUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fEliminarUnidad (aNombreUnidad)

Parámetros	Nombre	Tipo	Uso	Descripción
	aNombreUnidad	Cadena	Por valor	Nombre de la unidad.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función elimina un registro de la tabla Medida y Peso usando su nombre.

Ejemplo El siguiente código elimina un registro de la tabla Medida y Peso; si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```
lError = fEliminarUnidad (lNombreUnidad)

If lError <> 0 Then
    MensajeError lError
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fSetDatoUnidad (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo de la tabla de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que busque una unidad por su identificador, en caso de que lo encuentre escribe el contenido de la variable **INombreU** en el campo **cNombreU01** de la tabla de Medida y Peso; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaIdUnidad(IIdUnidad)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoUnidad ("cNombreU01", INombreU)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLeeDatoUnidad (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que busque una unidad por su identificador, en caso de que lo encuentre escribe el contenido de l campo **cNombreU01** en la variable la variable **INombreU** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaIdUnidad(IIdUnidad)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoUnidad ("cNombreU01", INombreU, 60)
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaUnidad (*aNombreUnidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aNombreUnidad</i>	Cadena	Por valor	Nombre de la unidad.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca una Unidad de Medida y Peso de acuerdo a los parámetros recibidos y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca una clasificación.

```
fBuscaUnidad (lNombreUnidad)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fBuscaIdUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fBuscaIdUnidad (*aldUnidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldValorClasif</i>	Entero	Por valor	Identificador de la Unidad.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función busca una unidad de medida y peso por su Identificador y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca una unidad de medida y peso por su identificador.

```
fBuscaIdUnidad (aldUnidad)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosPrimerUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosPrimerUnidad ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Unidades de Medida y Peso.

```
IError = fPosPrimerUnidad ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimoUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosUltimoUnidad ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Unidades de Medida y Peso.

```
fPosUltimoUnidad ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosSiguienteUnidad ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el siguiente registro de la posición actual de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Unidades de Medida y Peso.

```
IError = fPosSiguienteUnidad ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fPosAnteriorUnidad ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el registro anterior de la posición actual de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Unidades de Medida y Peso.

```
IError = fPosAnteriorUnidad ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosBOFUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Unidades de Medida y Peso.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable InicioTablaUnidades el resultado de la función fPosBOFUnidad .

```
InicioTablaUnidades = fPosBOFUnidad ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

fPosEOFUnidad ()

Disponibilidad	AdminPAQ 2003.
Sintaxis	fPosEOFUnidad ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Unidades de Medida y Peso
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable IFinTablaUnidades el resultado de la función fPosEOFUnidad .

```
IFinTablaUnidades = fPosEOFUnidad ()
```

Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),
--------------------	--

Alto Nivel – Lectura/Escritura

fAltaUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fAltaUnidad (*aldUnidad*,
LPFREGUNIDAD *astUnidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aldUnidad</i>	Entero	Por referencia	Identificador de la unidad.
	<i>astUnidad</i>	tUnidad	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aldUnidad: Al finalizar la función este parámetro contiene el identificador del nuevo valor de clasificación.

Descripción Esta función da de alta un nuevo valor de clasificación.

Ejemplo El siguiente código da de alta un nuevo valor de clasificación.

```
fAltaUnidad (lIdValorClasif, lstValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Alto Nivel – Lectura/Escritura

fActualizaUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fActualizaUnidad (*aNombreUnidad*, *astUnidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aNombreUnidad</i>	Cadena	Por valor	Nombre de la unidad.
	<i>astUnidad</i>	<i>tUnidad</i>	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función actualiza la unidad de medida y peso del registro especificado por el parametro *aCodigoValorClasif*.

Ejemplo El siguiente código actualiza la unidad de medida y peso del registro especificado por el parametro *aCodigoValorClasif*.

```
fActualizaUnidad (lNombreUnidad, lValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLlenaRegistroUnidad ()

Disponibilidad AdminPAQ 2003.

Sintaxis fLlenaRegistroUnidad (*astUnidad*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>astUnidad</i>	<i>tUnidad</i>	Por valor	Tipo de dato abstracto.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función asigna al registro de la base de datos los valores de la estructura de datos de la unidad de medida peso.

Ejemplo El siguiente código asigna al registro de la base de datos los valores de la estructura de datos unidad de medida peso.

```
fLlenaRegistroUnidad (lValorClasif)
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

FUNCIONES CATÁLOGO DE AGENTES

Bajo Nivel – Lectura/Escritura

fInsertaAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fInsertaAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en el catálogo de Agentes en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en el catálogo de Agentes. <pre>fInsertaAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaAgente ()

Disponibilidad	AdminPAQ 2006 v3.2.2.
Sintaxis	fEditaAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Activa el modo de Edición de un registro en el catálogo de agentes.
Ejemplo	El siguiente código busca una unidad por su identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente <pre>!Error = fBuscaAgente (ICodigoAgente) If !Error <> 0 Then MensajeError !Error Else fEditaAgente () End If</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fGuardaAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fGuardaAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro en el catálogo de agentes.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en el catálogo de Agentes. Esta función se llama después de que se utiliza la función finsertaAgente () y se graban los valores en los campos correspondientes.
	<pre>fGuardaAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelarModificacionAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fCancelarModificacionAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual del catálogo de Agentes. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro del catálogo de Agentes que estaba en modo de inserción o edición.
	<pre>fCancelarModificacionAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoAgente ()

Disponibilidad AdminPAQ 2004.

Sintaxis fSetDatoAgente (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo del catálogo de Agentes.

Ejemplo El siguiente código indica a la aplicación que busque un agente por su código, en caso de que lo encuentre escribe el contenido de la variable **INombreA** en el campo **cNombreA01** del catálogo de Agentes; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaAgente (ICodigoAgente)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoAgente ("cNombreA01", INombreA)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoAgente ()

Disponibilidad AdminPAQ 2004.

Sintaxis fLeeDatoAgente (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla del catálogo de Agentes.

Ejemplo El siguiente código indica a la aplicación que busque un agente por su código, en caso de que lo encuentre escribe el contenido de l campo **cNombreA01** en la variable la variable **INombreA** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaIdUnidad(IIdUnidad)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoAgente ("cNombreA01", INombreA, 60)
End If

```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaAgente ()

Disponibilidad	AdminPAQ 2004.			
Sintaxis	fBuscaAgente (<i>aCodigoAgente</i>)			
Parámetros	Nombre <i>aCodigoAgente</i>	Tipo Cadena	Uso Por valor	Descripción Código del agente.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca un agente por su código y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca un agente. fBuscaAgente (lCodigoAgente)			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fBuscaldAgente ()

Disponibilidad	AdminPAQ 2004.			
Sintaxis	fBuscaldAgente (<i>aldAgente</i>)			
Parámetros	Nombre <i>aldAgente</i>	Tipo Entero	Uso Por valor	Descripción Identificador de la Unidad.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca un agente y peso por su Identificador y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca un agente por su identificador. fBuscaldAgente (lIdAgente)			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fPosPrimerAgente ()

Disponibilidad AdminPAQ 2004.

Sintaxis fPosPrimerAgente ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de Agentes.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Agentes.

```
IError = fPosPrimerAgente ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimoAgente ()

Disponibilidad AdminPAQ 2004.

Sintaxis fPosUltimoAgente ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de Agentes.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Agentes.

```
fPosUltimoAgente ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosSiguienteAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Agentes.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Agentes. <pre>IError = fPosSiguienteAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosAnteriorAgente ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Agentes.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Agentes. <pre>IError = fPosAnteriorAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosBOFAgente ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Agentes.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaAgentes el resultado de la función fPosBOFAgente .
	<pre>lInicioTablaAgentes = fPosBOFAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFAgente ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosEOFAgente ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Agentes
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaAgentes el resultado de la función fPosEOFAgente .
	<pre>lFinTablaAgentes = fPosEOFAgente ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Funciones Catálogo de Almacenes

Bajo Nivel – Lectura/Escritura

fInsertaAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fInsertaAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error
Descripción	Adiciona un nuevo registro en el catálogo de Almacenes en modo de inserción.
Ejemplo	El siguiente código indica a la aplicación que inserte un nuevo registro en el catálogo de Almacenes.
	<pre>fInsertaAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fEditaAlmacen ()

Disponibilidad	AdminPAQ 2006.
Sintaxis	fEditaAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Activa el modo de Edición de un registro en el catálogo de agentes.
Ejemplo	El siguiente código busca una unidad por su identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente
	<pre>IError = fBuscaAlmacen (ICodigoAlmacen) If IError <> 0 Then MensajeError IError Else fEditaAlmacen () End If</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fGuardaAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fGuardaAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Guarda los cambios realizados a un registro en el catálogo de Almacenes.
Ejemplo	El siguiente código indica a la aplicación que guarde cierto registro en el catálogo de Almacenes. Esta función se llama después de que se utiliza la función finsertaAgente () y se graban los valores en los campos correspondientes.
	<pre>fGuardaAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fCancelarModificacionAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fCancelarModificacionAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función cancela las modificaciones al registro actual del catálogo de Almacenes. El registro debe estar en modo de edición o inserción.
Ejemplo	El siguiente código indica a la aplicación que cancele la modificación a un registro del catálogo de Almacenes que estaba en modo de inserción o edición.
	<pre>fCancelarModificacionAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fSetDatoAlmacen ()

Disponibilidad AdminPAQ 2004.

Sintaxis fSetDatoAlmacen (*aCampo*, *aValor*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino
	<i>aValor</i>	Cadena	Por valor	Valor de escritura

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función escribe el valor indicado en el campo correspondiente en el registro activo del catálogo de Almacenes.

Ejemplo El siguiente código indica a la aplicación que busque un agente por su código, en caso de que lo encuentre escribe el contenido de la variable **INombreA** en el campo **cNombreA01** del catálogo de Almacenes; en caso contrario muestra el mensaje de error correspondiente.

```
IError = fBuscaAlmacen (ICodigoAgente)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fSetDatoAlmacen ("cNombreA01", INombreA)
End If
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fLeeDatoAlmacen ()

Disponibilidad AdminPAQ 2004.

Sintaxis fLeeDatoAlmacen (*aCampo*, *aValor*, *aLen*)

Parámetros	Nombre	Tipo	Uso	Descripción
	<i>aCampo</i>	Cadena	Por valor	Campo destino.
	<i>aValor</i>	Cadena	Por referencia	Valor de lectura.
	<i>aLen</i>	Entero	Por valor	Longitud del dato de lectura.

Retorna Valores enteros:
 kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
 !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.

aValor: Al finalizar la función este parámetro contiene el valor del campo especificado.

Descripción Esta función lee el valor indicado del campo correspondiente en el registro activo de la tabla del catálogo de Almacenes.

Ejemplo El siguiente código indica a la aplicación que busque un agente por su código, en caso de que lo encuentre escribe el contenido de l campo **cNombreA01** en la variable la variable **INombreA** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```

IError = fBuscaIdUnidad(IIdUnidad)
If IError <> 0 Then
  MensajeError IError
Else
  IError = fLeeDatoAlmacen ("cNombreA01", INombreA, 60)
End If
  
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

Bajo Nivel – Búsqueda/Navegación

fBuscaAlmacen ()

Disponibilidad	AdminPAQ 2004.			
Sintaxis	fBuscaAlmacen (<i>aCodigoAlmacen</i>)			
Parámetros	Nombre <i>aCodigoAlmacen</i>	Tipo Cadena	Uso Por valor	Descripción Código del almacén.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca un agente por su código y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca un Almacen.			
	fBuscaAlmacen (lCodigoAlmacen)			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fBuscaldAlmacen ()

Disponibilidad	AdminPAQ 2004.			
Sintaxis	fBuscaldAlmacen (<i>aldAgente</i>)			
Parámetros	Nombre <i>aldAgente</i>	Tipo Entero	Uso Por valor	Descripción Identificador del almacén.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.			
Descripción	Esta función busca un Almacen por su Identificador y se posiciona en el registro correspondiente.			
Ejemplo	El siguiente código busca un agente por su identificador.			
	fBuscaldAlmacen (lCodigoAlmacen)			
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),			

fPosPrimerAlmacen ()

Disponibilidad AdminPAQ 2004.

Sintaxis fPosPrimerAlmacen ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el primer registro de la tabla de Almacenes.

Ejemplo El siguiente código indica a la aplicación que se posicione en el primer registro de la tabla de Almacenes.

```
IError = fPosPrimerAlmacen ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosUltimoAlmacen ()

Disponibilidad AdminPAQ 2004.

Sintaxis fPosUltimoAlmacen ()

Parámetros No usa.

Retorna Valores enteros:
KSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito.
!KSIN_ERRORES = Diferente de 0 (cero) – Código del error.

Descripción Esta función se ubica en el ultimo registro de la tabla de Almacenes.

Ejemplo El siguiente código indica a la aplicación que se posicione en el ultimo registro de la tabla de Almacenes.

```
IError = fPosUltimoAlmacen ()
```

Ver también fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosSiguienteAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosSiguienteAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el siguiente registro de la posición actual de la tabla de Almacenes.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el siguiente registro de la tabla de Almacenes. <pre>!Error = fPosSiguienteAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosAnteriorAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosAnteriorAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: kSIN_ERRORES = 0 (cero) – La operación fue realizada con éxito. !kSIN_ERRORES = Diferente de 0 (cero) – Código del error.
Descripción	Esta función se ubica en el registro anterior de la posición actual de la tabla de Almacenes.
Ejemplo	El siguiente código indica a la aplicación que se posicione en el registro anterior de la tabla de Almacenes. <pre>!Error = fPosAnteriorAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosBOFAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosBOFAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el inicio de la tabla de Almacenes.
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lInicioTablaAlmacenes el resultado de la función fPosBOFAlmacen .
	<pre>lInicioTablaAlmacenes = fPosBOFAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

fPosEOFAlmacen ()

Disponibilidad	AdminPAQ 2004.
Sintaxis	fPosEOFAlmacen ()
Parámetros	No usa.
Retorna	Valores enteros: 1 (uno) – Verdadero. 0 (cero) – Falso.
Descripción	Informa si el registro activo se encuentra en el fin de la tabla de Almacenes
Ejemplo	El siguiente código indica a la aplicación que asigne a la variable lFinTablaAlmacenes el resultado de la función fPosEOFAlmacen .
	<pre>lFinTablaAlmacenes = fPosEOFAlmacen ()</pre>
Ver también	fInsertarDocumento(), fEditarDocumento(), fGuardaDocumento(), fBorraDocumento(), fBorraDocumento_CW(), fCancelarModificacionDocumento(),

CONSTANTES DEL SDK

Constantes de longitud

Nombre	Longitud	Descripción
kLongFecha	23	Longitud máxima de caracteres para los campos de fechas.
kLongSerie	11	Longitud máxima de caracteres para las series.
kLongCodigo	30	Longitud máxima de caracteres usada para los códigos.
kLongNombre	60	Longitud máxima de caracteres para los nombres.
kLongReferencia	20	Longitud máxima de caracteres para las referencias.
kLongDescripcion	60	Longitud máxima de caracteres para las descripciones.
kLongCuenta	100	Longitud máxima de caracteres para las cuentas.
kLongMensaje	3000	Longitud máxima de caracteres para los mensajes.
kLongNombreProducto	255	Longitud máxima de caracteres para los nombres de producto.
kLongAbreviatura	3	Longitud máxima de caracteres para las abreviaturas.
kLongCodValorClasif	3	Longitud máxima de caracteres para los valores de clasificación.
kLongDenComercial	50	Longitud máxima de caracteres para la denominación comercial.
kLongRepLegal	50	Longitud máxima de caracteres para el representante legal.
kLongTextoExtra	50	Longitud máxima de caracteres para los textos extra.
kLongRFC	20	Longitud máxima de caracteres para el RFC.
kLongCURP	20	Longitud máxima de caracteres para el CURP.
kLongDesCorta	20	Longitud máxima de caracteres para descripciones cortas.
kLongNumeroExtInt	6	Longitud máxima de caracteres para el número exterior/interior.
kLongNumeroExpandido	30	Longitud máxima de caracteres para el número expandido.
kLongCodigoPostal	6	Longitud máxima de caracteres para el código postal.
kLongTelefono	15	Longitud máxima de caracteres para números de teléfono.
kLongEmailWeb	50	Longitud máxima de caracteres para direcciones de correo electrónico.
kLongSelloSat	175	Longitud máxima de caracteres para el sello del SAT.
kLongSerieCertSAT	20	Longitud máxima de caracteres para la serie del certificado del SAT.
kLongFechaHora	35	Longitud máxima de caracteres para la fecha y hora.
kLongSelloCFDI	175	Longitud máxima de caracteres para el sello del CFDI.
kLongCadOrigComplSAT	500	Longitud máxima de caracteres para la cadena original.
kLongitudUUID	36	Longitud máxima de caracteres para el UUID.
kLongitudRegimen	100	Longitud máxima de caracteres para el régimen fiscal de la empresa.
kLongitudMoneda	60	Longitud máxima de caracteres para la moneda.
kLongitudFolio	16	Longitud máxima de caracteres para el folio.
kLongitudMonto	30	Longitud máxima de caracteres para el monto.
kLongitudLugarExpedicion	400	Longitud máxima de caracteres para el lugar de expedición.

TIPOS DE DATO ABSTRACTOS DEL SDK

Definición de las Estructuras de Datos

Documentos – RegDocumento – tDocumento

Campo	Tipo	Longitud	Descripción
aFolio	Doble	NA	Folio del documento.
aNumMoneda	Entero	NA	Moneda del documento. 1 = Pesos MN, 2 = Moneda extranjera.
aTipoCambio	Doble	NA	Tipo de cambio del documento.
alImporte	Doble	NA	Importe del documento. Sólo se usa en documentos de cargo/abono.
aDescuentoDoc1	Doble	NA	No tiene uso, valor por omisión = 0 (cero).
aDescuentoDoc2	Doble	NA	No tiene uso, valor por omisión = 0 (cero).
aSistemaOrigen	Entero	NA	Valor mayor a 5 que indica una aplicación diferente a los PAQ's.
aCodConcepto	Cadena	kLongCodigo + 1	Código del concepto del documento.
aSerie	Cadena	kLongSerie + 1	Serie del documento.
aFecha	Cadena	kLongFecha + 1	Fecha del documento. Formato mm/dd/aaaa Las "/" diagonales son parte del formato.
aCodigoCteProv	Cadena	kLongCodigo + 1	Código del Cliente/Proveedor.
aCodigoAgente	Cadena	kLongCodigo + 1	Código del Agente.
aReferencia	Cadena	kLongReferencia + 1	Referencia del Documento.
aAfecta	Entero	NA	No tiene uso, valor por omisión = 0 (cero).
aGasto1	Double	NA	Valor por omisión = 0 (cero).
aGasto2	Double	NA	Valor por omisión = 0 (cero).
aGasto3	Double	NA	Valor por omisión = 0 (cero).

Llave del Documento – RegLlaveDoc – tLlaveDoc*

Campo	Tipo	Longitud	Descripción
aConcepto	Cadena	kLongCodigo + 1	Código del concepto del documento.
aSerie	Cadena	kLongSerie + 1	Serie del documento.
aFolio	Doble	NA	Folio del documento.

Movimientos – RegMovimiento – tMovimiento

Campo	Tipo	Longitud	Descripción
aConsecutivo	Entero	NA	Consecutivo del movimiento.
aUnidades	Doble	NA	Unidades del movimiento.
aPrecio	Doble	NA	Precio del movimiento (para doctos. de venta).
aCosto	Doble	NA	Costo del movimiento (para doctos. de compra).
aCodProdSer	Cadena	kLongCodigo + 1	Código del producto o servicio.
aCodAlmacen	Cadena	kLongCodigo + 1	Código del Almacén.
aReferencia	Cadena	kLongReferencia + 1	Referencia del movimiento.
aCodClasificacion	Cadena	kLongCodigo + 1	Código de la clasificaciún

Movimientos – RegMovimiento – tMovimientoDesc

Campo	Tipo	Longitud	Descripción
aConsecutivo	Entero	NA	Consecutivo del movimiento.
aUnidades	Doble	NA	Unidades del movimiento.
aPrecio	Doble	NA	Precio del movimiento (para doctos. de venta).
aCosto	Doble	NA	Costo del movimiento (para doctos. de compra).
aPorcDesccto1	Doble	NA	Porcentaje del Descuento 1
aImporteDesccto1	Doble	NA	Importe del Descuento 1
aPorcDesccto2	Doble	NA	Porcentaje del Descuento 2
aImporteDesccto2	Doble	NA	Importe del Descuento 2
aPorcDesccto3	Doble	NA	Porcentaje del Descuento 3
aImporteDesccto3	Doble	NA	Importe del Descuento 3
aPorcDesccto4	Doble	NA	Porcentaje del Descuento 4
aImporteDesccto4	Doble	NA	Importe del Descuento 4
aPorcDesccto5	Doble	NA	Porcentaje del Descuento 5
aImporteDesccto5	Doble	NA	Importe del Descuento 5
aCodProdSer	Cadena	kLongCodigo + 1	Código del producto o servicio.
aCodAlmacen	Cadena	kLongCodigo + 1	Código del Almacén.
aReferencia	Cadena	kLongReferencia + 1	Referencia del movimiento.
aCodClasificacion	Cadena	kLongCodigo + 1	Código de la clasificaciún

Movimientos con Serie/Capas – SeriesCapas – tSeriesCapas

Campo	Tipo	Longitud	Descripción
aUnidades	Doble	NA	Unidades del movimiento.
aTipoCambio	Doble	NA	Tipo de cambio del movimiento.
aSeries	Cadena	kLongCodigo + 1	Series del movimiento.
aPedimento	Cadena	kLongDescripcion + 1	Pedimento del movimiento.
aAgencia	Cadena	kLongDescripcion + 1	Agencia aduanal del movimiento.
aFechaPedimento	Cadena	kLongFecha + 1	Fecha de pedimento del movimiento.
aNumeroLote	Cadena	kLongDescripcion + 1	Número de lote del movimiento.
aFechaFabricacion	Cadena	kLongFecha + 1	Fecha de fabricación del movimiento.
aFechaCaducidad	Cadena	kLongFecha + 1	Fecha de Caducidad del movimiento.

Movimientos con Características – Características – tCaracterísticas

Campo	Tipo	Longitud	Descripción
aUnidades	Doble	NA	Unidades del movimiento.
aValorCaracteristica1	Cadena	kLongDescripcion + 1	Valor de la característica 1 del movimiento.
aValorCaracteristica2	Cadena	kLongDescripcion + 1	Valor de la característica 2 del movimiento.
aValorCaracteristica3	Cadena	kLongDescripcion + 1	Valor de la característica 3 del movimiento.

Movimientos con datos adicionales – RegTipoProducto – tTipoProducto

Campo	Tipo	Longitud	Descripción
aSeriesCapas	aSeriesCapas	NA	Tipo de dato abstracto: tSeriesCapas.
aCaracteristicas	aCaracteristicas	NA	Tipo de dato abstracto: Caracteristicas.

Llave de aperturas – RegLlaveAper - tLlaveAper

Campo	Tipo	Longitud	Descripción
aCodCaja	Cadena	kLongCodigo + 1	Código de la caja.
aFechaApe	Cadena	kLongFecha + 1	Fecha de apertura.

Productos – RegProducto – tProducto

Campo	Tipo	Longitud	Descripción
cCodigoProducto	Cadena	kLongCodigo + 1	Código del producto.
cNombreProducto	Cadena	kLongNombre + 1	Nombre del producto.
cDescripcionProducto	Cadena	kLongNombreProducto + 1	Descripción del producto.
cTipoProducto	Entero	NA	1- Producto, 2 - Paquete, 3 - Servicio
cFechaAltaProducto	Cadena	kLongFecha + 1	Fecha de alta del producto.
cFechaBaja	Cadena	kLongFecha + 1	Fecha de baja del producto.
cStatusProducto	Entero	NA	0 - Baja Lógica, 1 - Alta
cControlExistencia	Entero	NA	Control de existencia.
cMetodoCosteo	Entero	NA	1 - Costo Promedio Base a Entradas, 2 - Costo Promedio Base a Entradas Almacen 3 - Último costo, 4 - UEPS, 5 - PEPS, 6 - Costo específico, 7 - Costo Estandar.
cCodigoUnidadBase	Cadena	kLongCodigo + 1	Código de la unidad base.
cCodigoUnidadNoConvertible	Cadena	kLongCodigo + 1	Código de la unidad no convertible.
cPrecio1	Doble	NA	Lista de precios 1.
cPrecio2	Doble	NA	Lista de precios 2.
cPrecio3	Doble	NA	Lista de precios 3.
cPrecio4	Doble	NA	Lista de precios 4.
cPrecio5	Doble	NA	Lista de precios 5.
cPrecio6	Doble	NA	Lista de precios 6.
cPrecio7	Doble	NA	Lista de precios 7.
cPrecio8	Doble	NA	Lista de precios 8.
cPrecio9	Doble	NA	Lista de precios 9.
cPrecio10	Doble	NA	Lista de precios 10.
cImpuesto1	Doble	NA	Impuesto 1.
cImpuesto2	Doble	NA	Impuesto 2.
cImpuesto3	Doble	NA	Impuesto 3.
cRetencion1	Doble	NA	Retención 1.
cRetencion2	Doble	NA	Retención 2.
cNombreCaracteristica1	Cadena	kLongAbreviatura + 1	Nombre de la característica 1.
cNombreCaracteristica2	Cadena	kLongAbreviatura + 1	Nombre de la característica 2.
cNombreCaracteristica3	Cadena	kLongAbreviatura + 1	Nombre de la característica 3.
cCodigoValorClasificacion1	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 1.
cCodigoValorClasificacion2	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 2.
cCodigoValorClasificacion3	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 3.
cCodigoValorClasificacion4	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 4.
cCodigoValorClasificacion5	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 5.
cCodigoValorClasificacion6	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación 6.
cTextoExtra1	Cadena	kLongTextoExtra + 1	Texto extra 1.
cTextoExtra2	Cadena	kLongTextoExtra + 1	Texto extra 2.
cTextoExtra3	Cadena	kLongTextoExtra + 1	Texto extra 3.
cFechaExtra	Cadena	kLongFecha + 1	Fecha extra
cImporteExtra1	Doble	NA	Importe Extra 1.
cImporteExtra2	Doble	NA	Importe Extra 2.
cImporteExtra3	Doble	NA	Importe Extra 3.
cImporteExtra4	Doble	NA	Importe Extra 4.

Cliente/Proveedor – RegCteProv – tCteProv

Campo	Tipo	Longitud	Descripción
cCodigoCliente	Cadena	kLongCodigo + 1	Código del Cliente / Proveedor.
cRazonSocial	Cadena	kLongNombre + 1	Razón social.
cFechaAlta	Cadena	kLongFecha + 1	Fecha de alta.
cRFC	Cadena	kLongRFC + 1	RFC.
cCURP	Cadena	kLongCURP + 1	CURP.
cDenComercial	Cadena	kLongDenComercial + 1	Denominación comercial.
cRepLegal	Cadena	kLongRepLegal + 1	Representante legal.
cNombreMoneda	Cadena	kLongNombre + 1	Nombre de la moneda.
cListaPreciosCliente	Entero	NA	Lista de precios.
cDescuentoMovto	Doble	NA	Descuento.
cBanVentaCredito	Entero	NA	Bandera de venta a crédito. 0 – No se permite, 1 – Se permite.
cCodigoValorClasificacionCliente1	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 1.
cCodigoValorClasificacionCliente2	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 2.
cCodigoValorClasificacionCliente3	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 3.
cCodigoValorClasificacionCliente4	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 4.
cCodigoValorClasificacionCliente5	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 5.
cCodigoValorClasificacionCliente6	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 6.
cTipoCliente	Entero	NA	1 – Cliente, 2 – Cliente/Proveedor, 3 – Proveedor.
cEstatus	Entero	NA	Estado: 0 – Inactivo, 1 – Activo.
cFechaBaja	Cadena	kLongFecha + 1	Fecha de baja.
cFechaUltimaRevision	Cadena	kLongFecha + 1	Fecha de última revisión.
cLimiteCreditoCliente	Doble	NA	Límite de crédito.
cDiasCreditoCliente	Entero	NA	Días de crédito del cliente.
cBanExcederCredito	Entero	NA	Bandera de exceder crédito. 0 – No se permite, 1 – Se permite.
cDescuentoProntoPago	Doble	NA	Descuento por pronto pago.
cDiasProntoPago	Entero	NA	Días para pronto pago.
cInteresMoratorio	Doble	NA	Interés moratorio.
cDiaPago	Entero	NA	Día de pago.
cDiasRevision	Entero	NA	Días de revisión.
cMensajería	Cadena	kLongDesCorta + 1	Mensajería.
cCuentaMensajería	Cadena	kLongDescripcion + 1	Cuenta de mensajería.
cDiasEmbarqueCliente	Entero	NA	Días de embarque del cliente.
cCodigoAlmacen	Cadena	kLongCodigo + 1	Código del almacén.
cCodigoAgenteVenta	Cadena	kLongCodigo + 1	Código del agente de venta.
cCodigoAgenteCobro	Cadena	kLongCodigo + 1	Código del agente de cobro.
cRestriccionAgente	Entero	NA	Restricción de agente.
cImpuesto1	Doble	NA	Impuesto 1.
cImpuesto2	Doble	NA	Impuesto 2.
cImpuesto3	Doble	NA	Impuesto 3.
cRetencionCliente1	Doble	NA	Retención al cliente 1.
cRetencionCliente2	Doble	NA	Retención al cliente 2.
...	<i>Continúa en la siguiente página.</i>

Cliente/Proveedor – RegCteProv – tCteProv

Campo	Tipo	Longitud	Descripción
...	<i>Continúa.</i>
cCodigoValorClasificacionProveedor1	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 1.
cCodigoValorClasificacionProveedor2	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 2.
cCodigoValorClasificacionProveedor3	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 3.
cCodigoValorClasificacionProveedor4	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 4.
cCodigoValorClasificacionProveedor5	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 5.
cCodigoValorClasificacionProveedor6	Cadena	kLongCodValorClasif + 1	Código del valor de clasificación 6.
cLimiteCreditoProveedor	Doble	NA	Límite de crédito del proveedor.
cDiasCreditoProveedor	Entero	NA	Días de crédito del proveedor.
cTiempoEntrega	Entero	NA	Tiempo de entrega.
cDiasEmbarqueProveedor	Entero	NA	Días de embarque.
cImpuestoProveedor1	Doble	NA	Impuesto proveedor 1.
cImpuestoProveedor2	Doble	NA	Impuesto proveedor 2.
cImpuestoProveedor3	Doble	NA	Impuesto proveedor 3.
cRetencionProveedor1	Doble	NA	Retención proveedor 1.
cRetencionProveedor2	Doble	NA	Retención proveedor 2.
cBanInteresMoratorio	Entero	NA	Bandera de cálculo de interés moratorio. 0 – No se calculan, 1 – Si se calculan.
cTextoExtra1	Cadena	kLongTextoExtra + 1	Texto extra 1.
cTextoExtra2	Cadena	kLongTextoExtra + 1	Texto extra 2.
cTextoExtra3	Cadena	kLongTextoExtra + 1	Texto extra 3.
cFechaExtra	Cadena	kLongFecha + 1	Fecha extra.
cImporteExtra1	Doble	NA	Importe extra 1.
cImporteExtra2	Doble	NA	Importe extra 2.
cImporteExtra3	Doble	NA	Importe extra 3.
cImporteExtra4	Doble	NA	Importe extra 4.

Valor de Clasificación – RegValorClasificación – tValorClasificación

Campo	Tipo	Longitud	Descripción
cClasificacionDe	Entero	NA	Clasificación.
cNumClasificacion	Entero	NA	Número de la clasificación.
cCodigoValorClasificacion	Cadena	kLongCodValorClasif + 1	Código del valor de la clasificación.
cValorClasificacion	Cadena	kLongDescripcion + 1	Valor de la clasificación.

Unidad – RegUnidad – tUnidad

Campo	Tipo	Longitud	Descripción
cNombreUnidad	Cadena	kLongNombre + 1	Nombre de la unidad.
cAbreviatura	Cadena	kLongAbreviatura + 1	Abreviatura.
cDespliegue	Cadena	kLongAbreviatura + 1	Valor de despliegue.

Direcciones – RegDireccion– tDireccion

Campo	Tipo	Longitud	Descripción
cCodCteProv	Cadena	kLongCodigo + 1	Código cliente / proveedor.
cTipoCatalogo	Entero	NA	Tipo de catálogo.
cTipoDireccion	Entero	NA	Tipo de dirección.
cNombreCalle	Cadena	kLongDescripcion + 1	Calle.
cNumeroExterior	Cadena	kLongNumeroExtInt + 1	Número exterior.
cNumeroInterior	Cadena	kLongNumeroExtInt + 1	Número interior.
cColonia	Cadena	kLongDescripcion + 1	Colonia.
cCodigoPostal	Cadena	kLongCodigoPostal + 1	Código postal.
cTelefono1	Cadena	kLongTelefono + 1	Telefono 1.
cTelefono2	Cadena	kLongTelefono + 1	Telefono 2.
cTelefono3	Cadena	kLongTelefono + 1	Telefono 3.
cTelefono4	Cadena	kLongTelefono + 1	Telefono 4.
cEmail	Cadena	kLongEmailWeb + 1	Correo electrónico.
cDireccionWeb	Cadena	kLongEmailWeb + 1	Página web.
cCiudad	Cadena	kLongDescripcion + 1	Ciudad.
cEstado	Cadena	kLongDescripcion + 1	Estado.
cPais	Cadena	kLongDescripcion + 1	País.
cTextoExtra	Cadena	kLongDescripcion + 1	Texto extra.