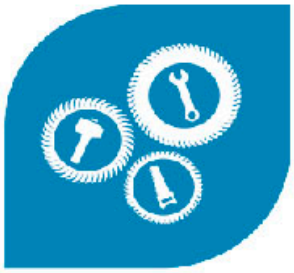

SDK

HERRAMIENTAS DESARROLLO DE APLICACIONES

Sistemas comerciales

SDK de los sistemas Comerciales

Aviso de derechos del propietario

Guía de información

Sistemas CONTPAQi®

Para que los métodos de este manual de referencia funcionen correctamente es necesario tener instaladas las siguientes versiones:

CONTPAQi® Comercial:	2.0.0 (o posterior)
Componentes:	1.1.8 (o posterior)

Información del documento

Elaborado por:	Ingenia L.I. Juan Raymundo Briseño Rubio I.S.C. Fabián Orlando Cebreros Cota
Fecha actual:	04 de noviembre de 2015

FCDCNT771150410

Contenido

Visión general

Introducción

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 **CONTPAQi® Comercial**, 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 **CONTPAQi® Comercial**.

Objetivo

El objetivo de este manual de referencia es que el lector tenga el conocimiento del listado de métodos que contiene la librería MGWServicios.dll

Cómo funciona el SDK

Los métodos disponibles en el SDK se comunican con **CONTPAQi® Comercial** a través de métodos de clases, éstas a su vez hacen llamados a las clases "base" de **CONTPAQi® Comercial**, es decir, a las clases usadas dentro de **CONTPAQi® Comercial**.

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 **CONTPAQi® Comercial**).

Protege las bases de datos, sus relaciones y sigue las reglas de negocio de **CONTPAQi® Comercial**.

Requerimientos para trabajar con el SDK

Ambiente

- **CONTPAQi® Comercial** instalado (monousuario o como estación).
- En caso de no tener instalado **CONTPAQi® Comercial** es necesario contar con los archivos que conforman el SDK de **CONTPAQi® Comercial** en la misma carpeta donde reside la aplicación en desarrollo.
- **Microsoft Excel ®** (cualquier versión) - Para desarrollo de modelos u Hojas electrónicas.
- Entorno de programación. Editor/Compilador del lenguaje elegido

(VB / C / Plataforma .Net, etc.).

Archivos utilizados por el SDK

Archivo: MGWSERVICIOS.dll

Descripción: Es la interface del SDK con **CONTPAQi® Comercial**.

Librería de encadenado, aquí se encuentran las funciones del **SDK**.

Ubicación: C:\Archivos de programa\Compac**CONTPAQi® Comercial**

*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 **CONTPAQi® Comercial** con la que va a interactuar. Es decir, no se recomienda desarrollar una aplicación con el SDK de **CONTPAQi® Comercial 2.0.0** para interactuar con un **CONTPAQi® Comercial 1.1.3.***

Tips y Conceptos Básicos

• Antes de hacer accesos mediante el SDK, se debe asegurar que **CONTPAQi®** Comercial funciona correctamente y que la información que se está generando es correcta.

• Estar familiarizado con la estructura de la Base de Datos de **CONTPAQi®** Comercial.

• 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 **CONTPAQi®** Comercial).

• Modularidad en el código: Esto es crear diversos módulos para separar funcionalidad global y local. (Si el entorno de programación lo permite).

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 MGWSERVICIOS.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 **CONTPAQi®** Comercial.

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 **CONTPAQi®** Comercial.

• Validar desde la aplicación que se desarrolla que los datos que se envían sean consistentes y tengan 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. fSetNombrePaq.
Esta función se llama una sola vez al iniciar la sesión sobre una empresa y posteriormente crear registros y posteriormente se terminara la sesión con la función fTerminaSDK.

Ejemplo: En 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

Ejemplo de varias empresas: Se inicia el SDK (solo una ocasión), se abre la empresa A se hacen todas las operaciones requeridas, se cierra la empresa A, se abre la empresa B se hacen todas las operaciones requeridas, se cierra la empresa B y luego se termina el SDK

Con esto nos damos cuenta que son una solo sesión de SDK podemos interactuar con distintas empresas.

• 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.

• 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. Ésta función se llama una sola vez al finalizar un proceso o acción completa, por ejemplo al final de día si es que nuestro proceso o aplicación esta en funcionamiento durante todo este tiempo.

Establecer el directorio del MGWSERVICIOS
Inicializar SDK (fSetNombrePaq)
Abrir Empresa
Tu función o proceso completo
Cerrar Empresa
Terminar SDK

Trabajando con Documentos

En el SDK de **CONTPAQi®** Comercial 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 de CONTPAQi® Comercial.

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

Establecer el directorio del MGWSERVICIOS

Inicializar SDK
Abrir Empresa
Alta de documento Cargo/Abono
Cerrar Empresa
Terminar SDK

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

Establecer el directorio del MGWSERVICIOS

Inicializar SDK
Abrir Empresa
Alta de documento
Alta de movimientos
Alta del movimiento con series o pedimentos
Calcula los movimientos con series o pedimentos
Cerrar Empresa
Terminar SDK

Funciones Generales

Inicialización / Terminación

fSetNombrePAQ ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetNombrePAQ(*aSistema*)

Parámetros

Nombre	Tipo	Uso	Descripción
aSistema	Cadena	Por referencia	Nombre del sistema al que se conectará el SDK Descripción del error. Para establecer una conexión a **CONTPAQ i® Comercial** este parámetro deberá ser igual a "**CONTPAQ I Comercial**"

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 es este caso es

Ejemplo El siguiente código crea una conexión al SDK de **CONTPAQ I Comercial**.

```
lResultado = fSetNombrePAQ("CONTPAQ I Comercial");
        if (lResultado != 0)
        {
           fError(lResultado, sMensaje, 512);
           MessageBox.Show("Error: " + sMensaje);
        }
```

fTerminaSDK ()

Disponibilidad	**CONTPAQi® Comercial 2.0.0.**
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 **CONTPAQi®** Comercial.

```
fTerminaSDK()
```

Manejo de errores

fError ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aNumError	Entero	Por valor	Número del error.
aMensaje	Cadena	Por referencia	Descripción del error.
aLen	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 fSetNombrePAQ(), 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 512 caracteres.

En C# cuando se manejan Cadenas por Referencia es necesario utilizar StringBuilder para Retornar el valor.

```
lError = fSetNombrePAQ("CONTPAQ I COMERCIAL");
  if(lError ¡= 0)
  {
    rError(lError);
   }
 // Función para el manejo de errores en SDK
public static void rError(int aNumError)
 {   StringBuilder aMensaje = new StringBuilder(512);
   if (iError != 0)
   { fError(aNumError, aMensaje, 512);
     MessageBox.show("Error: " + aMensaje);
   }
}
```

Funciones de empresas

fPosPrimerEmpresa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aIdEmpresa	Entero	Por referencia	Identificador de la empresa.
aNombreEmpresa	Cadena	Por referencia	Nombre de la empresa.
aDirectorioEmpresa	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.

aIdEmpresa: 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 **CONTPAQi®** Comercial, 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 **CONTPAQi®** Comercial.

```
fPosPrimerEmpresa(IIdEmpresa, INombreEmpresa, IDirectorioEmpresa)
```

Navegación

fPosSiguienteEmpresa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aIdEmpresa	Entero	Por referencia	Identificador de la empresa.
aNombreEmpresa	Cadena	Por referencia	Nombre de la empresa.
aDirectorioEmpresa	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.

aIdEmpresa: 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 **CONTPAQi®** Comercial; 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 **CONTPAQi®** Comercial.

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

Apertura/Cierre

fAbreEmpresa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAbreEmpresa (*aDirectorioEmpresa*)

Parámetros

Nombre	Tipo	Uso	Descripción
aDirectorioEmpresa	Cadena	Por valor	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:\Compac\Empresas\EmpresaEjemplo*.

```
lDirectorioEmpresa = "C:\Compac\Empresas\EmpresaEjemplo"
fAbreEmpresa (lDirectorioEmpresa)
```

fCierraEmpresa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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()
```

Funciones de documentos

Bajo Nivel – Lectura/Escritura

fEditarDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

```
lError = fBuscaDocumento(lLlaveDocto )
If lError <> 0 Then
    MensajeError lError
Else
    fEditarDocumento ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fGuardaDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 () o fEditarDocumento() y se graban los valores en los campos correspondientes.

```
fGuardaDocumento ()
```

fCancelarModificacionDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fBorraDocumento ()

Disponibilidad **CONTPAQi®** Comercial 2.0.0.

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

```
lError = fBuscaDocumento(lLlaveDocto )
If lError <> 0 Then
    MensajeError lError
Else
    fBorraDocumento ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fBorrarAsociacion_Param ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodConcepto_Pagar	Cadena	Por valor	Código del concepto del documento pagado.
aSerie_Pagar	Cadena	Por valor	Serie del documento pagado.
aFolio_Pagar	Doble	Por valor	Folio del documento pagado.
aCodConcepto_Pago	Cadena	Por valor	Código del concepto del documento que pagó.
aSerie_Pago	Cadena	Por valor	Serie del documento que pagó.
aFolio_Pago	Doble	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 elimina 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.

```
lError = fBorrarAsociacion (lCodConcepto_Pagar, lSerie_Pagar, lFolio_Pagar
lCodConcepto_Pago, lSerie_Pago, lFolio_Pago)
If lError <> 0 Then
    MensajeError lError
Else
```

fSetDatoDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoDocumento (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fLeeDatoDocumento (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	Cadena	Por referencia	Valor de escritura
	aLen	Entero	Por valor	Tamaño del campo.

Retorna Valores enteros:

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

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

aCampo. Corresponde al Nombre del campo de la Base de Datos.

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

aLen. Corresponde a la Longitud del campo en la Base de Datos.

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.

Para más información de los tamaños de los campos verificar *Estructura de las tablas de la BDD línea Comercial.*

```
lError = fBuscaDocumento(lLlaveDocto )
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoDocumento("cFecha", lFecha, 8)
End If
```

fSiguienteFolio ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoConcepto	Cadena	Por valor	Código del concepto del documento.
aSerie	Cadena	Por referencia	Serie del documento
aFolio	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.

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 el siguiente folio disponible para un determinado concepto.

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

fSetFiltroDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros	Nombre	Tipo	Uso	Descripción
	aFechaInicio	Cadena	Por valor	Fecha inicial del rango.
	aFechaFin	Cadena	Por valor	Fecha final del rango.
	aCodigoConcepto	Cadena	Por valor	Código del concepto a filtrar.
	aCodigoCteProv	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 utilizando formato tablas.

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.

```
lFechaInicio = "01/01/2006"
lFechaFin = "06/30/2006"
lCodConcepto = "4"
lCodCteProv = "CTE002"
lError = fSetFiltroDocumento (lFechaInicio, lFechaFin, lCodConcepto, lCodCteProv )
```

fCancelaFiltroDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último filtro activo de documentos.

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

```
fCancelaFiltroDocumento ()
```

fDocumentoImpreso ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fDocumentoImpreso (*almpreso*)

Parámetros

Nombre	Tipo	Uso	Descripción
almpreso	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 realizar la búsqueda 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)
```

Bajo Nivel - Búsqueda/Navegación

fBuscarDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscarDocumento (*aCodConcepto, aSerie, aFolio*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCodConcepto	Cadena	Por valor	Código del concepto del documento.
	aSerie	Cadena	Por valor	Serie del documento.
	aFolio	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.

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

fBuscarIdDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscarIdDocumento (*aIdDocumento*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdDocumento	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 (lIdDocumento)
```

fPosPrimerDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerDocumento ()
```

Bajo Nivel - Búsqueda/Navegación, *continúa...*

fPosUltimoDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerDocumento ()
```

fPosSiguienteDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteDocumento ()
```

fPosAnteriorDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorDocumento ()
```

fPosBOF ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fPosEOF ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

Alto Nivel – Lectura/Escritura

fAltaDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaDocumento (*aIdDocumento, aDocumento*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aIdDocumento	Entero largo	Por referencia	Identificador del documento.
	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.

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

Descripción Esta función da de alta documentos de tipo factura

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

Es necesario declarar la estructura de Documento y llenar los campos necesarios.

```
lError = fAltaDocumento (llIdDocto, lDocumento)
If lError <> 0 Then
    MensajeError lError
Else
```

fAltaDocumentoCargoAbono ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

```
lError = fAltaDocumentoCargoAbono (lDocumento)
If lError <> 0 Then
    MensajeError lError
Else
```

fSaldarDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSaldarDocumento (*aDoctoaPagar*, aDoctoPago, aImporte, aIdMoneda, aFecha)

Parámetros

Nombre	Tipo	Uso	Descripción
aDoctoaPagar	tLlaveDocto	Por valor	Tipo de dato abstracto.
aDoctoPago	tLlaveDocto	Por valor	Tipo de dato abstracto.
aImporte	Doble	Por valor	Importe del pago.
aIdMoneda	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 salde un documento con la información enviada, en caso de presentarse algún error manda el mensaje correspondiente.

```
lError = fSaldarDocumento (lDoctoaPagar, lDoctoPago, lImporte, lIdMoneda,
lFecha)
If lError <> 0 Then
    MensajeError lError
Else
```

Lectura/Escritura

fRegresaIVACargo ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
astDocto	tLlaveDocto	Por valor	Tipo de dato abstracto.
aNetoTasa15	Doble	Por referencia	Base de la tasa de 15%
aNetoTasa10	Doble	Por referencia	Base de la tasa de 10%
aNetoTasaCero	Doble	Por referencia	Base de la tasa cero
aNetoTasaExcenta	Doble	Por referencia	Base de productos exentos
aNetoOtrasTasas	Doble	Por referencia	Base de otras tasas
aIVATasa15	Doble	Por referencia	IVA de la tasa de 15%
aIVATasa10	Doble	Por referencia	IVA de la tasa de 10%
aIVAOtrasTasas	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 *astDocto* y que regrese los valores correspondientes a los parámetros por valor especificados.

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

fRegresaIVAPago ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

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

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

```
lError = fRegresaIVAPago (lLlaveDocto, lNetoTasa15, lNetoTasa10, lNetoTasaCero,
lNetoTasaExcenta, lNetoOtrasTasas, lIVATasa15, lIVATasa10, lIVAOtrasTasas)
```

Lectura/Escritura, *continúa...*

fGetTamSelloDigitalYCadena ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis *fGetTamSelloDigitalYCadena (atPtrPassword, aEspSelloDig, aEspCadOrig)*

Parámetros	Nombre	Tipo	Uso	Descripción
	atPtrPassword	Cadena	Por Referencia	Contraseña del certificado.
	aEspSelloDig	Entero	Por Referencia	Tamaño del sello Digital.
	aEspCadOrig	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 **lPassword** e indica a la aplicación que obtenga el tamaño del sello digital y cadena original del documento, recibiéndolos en las variables **lEspSello** e **lEspSello**.

```
lError = fGetTamSelloDigitalYCadena (lPassword, lEspSello,
lEspCadOrig)
```

Lectura/Escritura, *continúa...*

fGetSelloDigitalYCadena ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis *fGetSelloDigitalYCadena (atPtrPassword, aEspSelloDig, aEspCadOrig)*

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	atPtrPassword	Cadena	Por Referencia	Contraseña del certificado.
	aEspSelloDig	Entero	Por Referencia	Tamaño del sello Digital.
	aEspCadOrig	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 **lPassword** e indica a la aplicación que obtenga el sello digital y cadena original del documento, recibiéndolos en las variables **lEspSello** e **lEspCadOrig**.

```
lError = fGetSelloDigitalYCadena (lPassword, lEspSello, lEspCadOrig)
```

Lectura/Escritura, *continúa...*

fEmitirDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodConcepto	Cadena	Por referencia	Código del concepto
aSerie	Cadena	Por Referencia	Serie del Documento.
aFolio	Doble	Por Valor	Folio del documento.
aPassword	Cadena	Por Referencia	Contraseña del certificado de sello digital
aArchivoAdicional	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

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

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("100","CFDI",154,"12345678a","divisas.xml")
```

fDocumentoUUID()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodConcepto	Cadena	Por referencia	Código del concepto
aSerie	Cadena	Por Referencia	Serie del Documento.
aFolio	Doble	Por Valor	Folio del documento.
atPtrCFDIUUID	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 lUUID.

```
lError = fDocumentoUUID(aCodConcepto, aSerie, aFolio,atPtrCFDIUUID)
```

fActivarPrecioCompra ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis

fActivarPrecioCompra (aActivar)

Parámetros	Nombre	Tipo	Uso	Descripción
	aActivar	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.

fEntregEnDiscoXML ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodConcepto	Cadena	Por Referencia	Código del concepto
aSerie	Cadena	Por Referencia	Serie del documento
aFolio	Doble	Por Valor	Folio del documento
aFormato	Entero	Por Valor	Formato de entrega(0 = XML, 1 = PDF)

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:\Compac\Empresas\Reportes**CONTPAQi®** Comercial \Plantilla_Factura_cfdi_1.html"

```
lError = fEntregEnDiscoXML ("4", "B1", 45, 1,
"C:\Compac\Empresas\Reportes\CONTPAQi® Comercial \Plantilla_Factura_cfdi_1.html")
```

Alto Nivel – Busqueda/Navegación

fBuscaDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscaDocumento (*aLlaveDocto*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aLlaveDocto	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*.

```
lError = fBuscaDocumento (aLlaveDocto)
```

Funciones de movimientos

Bajo Nivel – Lectura/Escritura

EditarMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

```
lError = fBuscarIdMovimiento (lIdMovto)
If lError <> 0 Then
    MensajeError lError
Else
    fEditarMovimiento ()
End If
```

fGuardaMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 fEditarMovimiento()

```
fGuardaMovimiento ()
```

fCancelaCambiosMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fCancelaCambiosMovimiento ()
```

fAltaMovimientoCaracteristicas_Param ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimientoCaracteristicas_Param (*aIdMovimiento, aIdMovtoCaracteristicas, aUnidades, aValorCaracteristica1, aValorCaracteristica2, aValorCaracteristica3*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Cadena	Por valor	Identificador del movimiento.
aIdMovtoCaracteristicas	Cadena	Por valor	Identificador del movimiento con características.
aUnidades	Cadena	Por valor	Unidades.
aValorCaracteristica1	Cadena	Por valor	Valor de la característica 1.
aValorCaracteristica2	Cadena	Por valor	Valor de la característica 2.
aValorCaracteristica3	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 de un producto 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.

```
lError = fAltaMovimientoCaracteristicas_Param ("lIdMovimiento,
lIdMovtoCaracteristicas,
        lUnidades, lValorCaracteristica1, lValorCaracteristica2,   lValorCaracteristica3)
If lError <> 0 Then
    MensajeError lError
Else
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fAltaMovtoCaracteristicasUnidades_Param ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovtoCaracteristicasUnidades_Param (*aIdMovimiento, aIdMovtoCaracteristicas,* aUnidad, aUnidades, aUnidadesNC, aValorCaracteristica1, aValorCaracteristica2, *aValorCaracteristica3*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Cadena	Por valor	Identificador del movimiento.
aIdMovtoCaracteristicas	Cadena	Por valor	Identificador del movimiento con características.
aUnidad	Cadena	Por valor	Abreviatura de la unidad de compra venta
aUnidades	Cadena	Por valor	Las unidades del movimiento de características.
aUnidadesNC	Cadena	Por valor	Abreviatura de la unidad de compra venta no convertible.
aValorCaracteristica1	Cadena	Por valor	Valor de la característica 1.
aValorCaracteristica2	Cadena	Por valor	Valor de la característica 2.
aValorCaracteristica3	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 ejemplo da de alta movimiento de características con unidades de compra venta.

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

fAltaMovimientoSeriesCapas_Param ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimientoSeriesCapas _Param (*aIdMovimiento, aUnidades, aTipoCambio, aSeries, aPedimento, aAgencia, aFechaPedimento, aNumeroLote, aFechaFabricacion, aFechaCaducidad*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Cadena	Por valor	Identificador del movimiento.
aUnidades	Cadena	Por valor	Unidad de peso y medida.
aTipoCambio	Cadena	Por valor	Tipo de cambio.
aSeries	Cadena	Por valor	Series.
aPedimento	Cadena	Por valor	Referencia del pedimento.
aAgencia	Cadena	Por valor	Referencia de la agencia.
aFechaPedimento	Cadena	Por valor	Fecha del pedimento.
aNumeroLote	Cadena	Por valor	Número de lote.
aFechaFabricacion	Cadena	Por valor	Fecha de fabricación.
aFechaCaducidad	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 número 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 número de serie, lote y/o pedimento asociado.

```
lError = fAltaMovimientoSeriesCapas _Param (lIdMovimiento, lUnidades,
lTipoCambio, lSeries, lPedimento,  lAgencia, lFechaPedimento, lNumeroLote,
lFechaFabricacion, lFechaCaducidad)
```

fObtieneUnidadesPendientes ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aConceptoDocto	Cadena	Por valor	Código del concepto del documento a buscar.
aCodigoProducto	Cadena	Por valor	Código del producto a buscar sus unidades pendientes.
aCodigoAlmacen	Cadena	Por valor	Código del almacén a buscar si es igual a 0 (cero) busca en todos los almacenes.
aUnidades	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 *lCodigoProducto* del almacen *lCodigoAlmacen*, del documento **lConceptoDocto**.

```
lError = fObtieneUnidadesPendientes (lConceptoDocto, lCodigoProducto,
lCodigoAlmacen,
        lUnidades)
```

fObtieneUnidadesPendientesCarac ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aConceptoDocto	Cadena	Por valor	Código del concepto del documento a buscar.
aCodigoProducto	Cadena	Por valor	Código del producto a buscar sus unidades pendientes.
aCodigoAlmacen	Cadena	Por valor	Código del almacén a buscar si es igual a 0 (cero) busca en todos los almacenes.
aValorCaracteristica1	Cadena	Por valor	Valor característica 1
aValorCaracteristica2	Cadena	Por valor	Valor característica 2
aValorCaracteristica3	Cadena	Por valor	Valor característica 3
aUnidades	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 lCodigoProducto del almacén lCodigoAlmacen, del documento lConceptoDocto.

```
lError = fObtieneUnidadesPendientesCarac (aConceptoDocto, aCodigoProducto,
aCodigoAlmacen,
                                          aValorCaracteristica1,
aValorCaracteristica2,
                                          aValorCaracteristica3, aUnidades)
```

fModificaCostoEntrada ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fModificaCostoEntrada (*aIdMovimiento, aCostoEntrada*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Cadena	Por valor	Identificador del movimiento a modificar.
aCostoEntrada	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 lIdMovimiento.

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

fSetDatoMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetDatoMovimiento (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 **lDescuento** en el campo **cDescuen01** de la tabla de movimientos; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscarIdMovimiento (lIdMovto)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoMovimiento ("cDescuen01 ", lDescuento)
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino.
	aValor	Cadena	Por referencia	Valor de lectura.
	aLen	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 extrae el contenido del campo **cDescuen01** en la variable **lDescuento** con una longitud de **8** caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscarIdMovimiento (lIdMovto)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoMovimiento ("cDescuen01 ", lDescuento, 8)
End If
```

Bajo Nivel – Búsqueda/Navegación

fSetFiltroMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetFiltroMovimiento(aIdDocumento)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aIdDocumento	**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 *lIdDocto*.

```
lError = fSetFiltroMovimiento (lIdDocto)
```

fCancelaFiltroMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

fBuscarIdMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscarIdMovimiento (*aIdMovimiento*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	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)
```

fPosPrimerMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosPrimerMovimiento ()
```

fPosUltimoMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 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 Movimientos.

```
lError = fPosUltimoMovimiento ()
```

fPosSiguienteMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 posiciona 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.

```
lError = fPosSiguienteMovimiento ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosAnteriorMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosAnteriorMovimiento ()
```

fPosMovimientoBOF ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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**.

```
lInicioTablaMovto = fPosMovimientoBOF ()
```

fPosMovimientoBOF ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fPosMovimientoBOF ()

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 **fPosMovimientoBOF**.

```
lFinTablaMovto = fPosMovimientoBOF ()
```

Alto Nivel – Lectura/Escritura

fAltaMovimiento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimiento (*aIdDocumento, aIdMovimiento, astMovimiento*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdDocumento	Entero largo	Por valor	Identificador del movimiento.
aIdMovimiento	Entero largo	Por referencia	Identificador del documento.
astMovimiento	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.

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

Descripción 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)
```

fAltaMovimientoCDesct ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimientoCDesct (*aldDocumento, aldMovimiento, astMovimiento*)

Parámetros

Nombre	Tipo	Uso	Descripción
aldDocumento	Entero largo	Por valor	Identificador del documento.
aldMovimiento	Entero largo	Por Referencia	Identificador del movimiento
astMovimiento	tMovmientoDesc	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.

fAltaMovimientoCaracteristicas ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimientoCaracteristicas (*aIdMovimiento, aIdMovtoCaracteristicas, aCaracteristicas*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Entero largo	Por valor	Identificador del movimiento.
aIdMovtoCaracteristicas	Entero largo	Por referencia	Contendrá Identificador movto.
aCaracteristicas	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.

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

Descripción Esta función da de alta un movimiento con características.

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

```
fAltaMovimientoCaracteristicas (aIdMovimiento, aIdMovtoCaracteristicas, aCaracteristicas)
```

fAltaMovtoCaracteristicasUnidades ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovtoCaracteristicasUnidades (*aIdMovimiento, aIdMovtoCaracteristicas, aCaracteristicasUnidades*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Entero largo	Por valor	Identificador del movimiento.
aIdMovtoCaracteristicas	Entero largo	Por referencia	Contendra Identificador movto.
aCaracteristicasUnidades	*tCaracteristicasUnidades*	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.

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

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

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

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

fAltaMovimientoSeriesCapas ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fAltaMovimientoSeriesCapas (*aIdMovimiento, aSeriesCapas*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdMovimiento	Entero Largo	Por valor	Identificador del movimiento.
aSeriesCapas	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 número 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 número de serie, lote y/o pedimento asociado.

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

Funciones de Clientes / Proveedores

Bajo Nivel – Lectura/Escritura

fInsertaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fEditaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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
```

fGuardaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

fBorraCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

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

fCancelarModificacionCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fCancelarModificacionCteProv ()
```

fEliminarCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fEliminarCteProv (*aCodigoCteProv*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCodigoCteProv	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 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.

```
lError = fEliminarCteProv (aCodigoCteProv)
If lError <> 0 Then
    MensajeError lError
End If
```

fSetDatoCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetDatoCteProv (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 lRFC en el campo cRFC de la tabla de Cliente / Proveedor; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaCteProv (lCodCteProv)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoCteProv ("cRFC ", lRFC)
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino.
	aValor	Cadena	Por referencia	Valor de lectura.
	aLen	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 lee el contenido del campo cRFC y lo guarda en la variable lRFC con una longitud de 20 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscarIdMovimiento (lIdMovto)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoCteProv ("cRFC ", lRFC, 20)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdCteProv (*aCodCteProv*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCodCteProv	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)
```

fBuscaIdCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdCteProv (*aIdCteProv*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aIdCteProv	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)
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosPrimerCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerCteProv ()
```

fPosUltimoCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Clientes / Proveedores.

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

```
lError = fPosUltimoCteProv ()
```

fPosSiguienteCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteCteProv ()
```

fPosAnteriorCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorCteProv ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosBOFCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 Documentos

Ejemplo El siguiente código indica a la aplicación que asigne a la variable lInicioTablaCteProv el resultado de la función fPosBOFCteProv.

```
lInicioTablaCteProv= fPosBOFCteProv ()
```

fPosEOFCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 lFinTablaCteProvel resultado de la función fPosEOFCteProv.

```
lFinTablaCteProv = fPosEOFCteProv ()
```

Alto Nivel – Lectura/Escritura

fAltaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaCteProv (*aIdCteProv, astCteProv*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdCteProv	Entero largo	Por referencia	Identificador del Cliente / Proveedor.
astCteProv	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.

aIdCteProv: Al finalizar la función este parámetro contiene el identificador del nuevo Cliente / Proveedor.

Descripción Esta función da de alta un nuevo Cliente / Proveedor.

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

```
fAltaCteProv (IdCteProv, ICteProv)
```

fActualizaCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fActualizaCteProv (*aCodigoCteProv, astCteProv*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoCteProv	Cadena	Por referencia	Identificador del Cliente / Proveedor.
astCteProv	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)
```

fLlenaRegistroCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fLlenaRegistroCteProv (*astCteProv, aEsAlta)*

Parámetros

Nombre	Tipo	Uso	Descripción
astCteProv	tCteProv	Por valor	Tipo de dato abstracto.
aEsAlta	Entero	Por valor	1 = Nuevo Cliente / Proveedor. 2 = Actualización 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 asigna al registro de un nuevo Cliente / Proveedor en la tabla de Clientes / Proveedores.

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

Funciones de Productos

Bajo Nivel – Lectura/Escritura

fInsertaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
fInsertaProducto ()
```

fEditaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

```
lError = fBuscaProducto (aCodProducto)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaProducto ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fGuardaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fBorraProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 producto por su código, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

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

fCancelarModificacionProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fEliminarProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoProducto (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 lFechaAlta en el campo cFechaAl01 de la tabla de Cliente / Proveedor; 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
```

fLeeDatoProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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 lee el contenido del 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
```

fRecuperaTipoProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fRecuperaTipoProducto(lUnidades, lSerie, lLote, lPedimento,
lCaracteristicas)


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

Bajo Nivel – Lectura/Escritura, *continúa...*

fRegresaPrecioVenta ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoConcepto	Cadena	Por valor	Código del concepto.
aCodigoCliente	Cadena	Por valor	Código del cliente.
aCodigoProducto	Cadena	Por valor	Código del producto.
aPrecioVenta	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.

aPrecioVenta: Al finalizar la función este parámetro contiene el precio de venta del producto solicitado.

Descripción 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.

```
lError = fRegresaPrecioVenta (lCodigoConcepto, lCodigoCliente, lCodigoProducto, aPrecioVenta)
If lError <> 0 Then
    MensajeError lError
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaProducto (*aCodProducto*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCodProducto	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 (lCodProducto)
```

fBuscaIdProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdProducto (*aIdProducto*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aIdProducto	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 (lIdProducto)
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosPrimerProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerProducto ()
```

fPosUltimoProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Productos.

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

```
lError = fPosUltimoProducto ()
```

fPosSiguienteProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteProducto ()
```

fPosAnteriorProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorProducto ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosBOFProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 lInicioTablaProductos el resultado de la función fPosBOFProducto.

```
lInicioTablaProductos = fPosBOFProducto ()
```

fPosEOFProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 lFinTablaProductos el resultado de la función fPosEOFProducto.

```
lFinTablaProductos = fPosEOFProducto ()
```

Alto Nivel – Lectura/Escritura

fAltaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaProducto (*aIdProducto, astProducto*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdProducto	Entero	Por referencia	Identificador del producto.
astProducto	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.

aIdProducto: Al finalizar la función este parámetro contiene el identificador del nuevo producto.

Descripción Esta función da de alta un nuevo Producto.

Ejemplo El siguiente código da de alta un nuevo producto utilizando la estructura de producto.

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

fActualizaProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fActualizaProducto (*aCodigoProducto, astCteProv*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Entero largo	Por referencia	Código del producto.
astProducto	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)
```

fLlenaRegistroProducto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fLlenaRegistroCteProv (astProducto, aEsAlta)

Parámetros

Nombre	Tipo	Uso	Descripción
astProducto	tProducto	Por valor	Tipo de dato abstracto.
aEsAlta	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)
```

Funciones de Addenda

Bajo Nivel – Lectura/Escritura

fInsertaDatoCompEducativo ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fInsertaDatoCompEducativo(int aIdServicio, int aNumCampo, char *aDato)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdServicio	Entero	Por valor	Identificador del servicio.
aNumCampo	Entero	Por valor	Número de campo
aDato	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.

```
fInsertaDatoCompEducativo (aIdServicio, aNumCampo, aDato)
```

fInsertaDatoAddendaDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fInsertaDatoAddendaDocto(aIdAddenda, aIdCatalogo, aNumCampo, aDato)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdAddenda	Entero	Por valor	Iedntificador de la Addenda
aIdCatalogo	Entero	Por valor	Identificador del documento
aNumCampo	Entero	Por Valor	Número del documento
aDato	Cadena	Por referencia	

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.

```
fInsertaDatoAddendaDocto(aIdAddenda, aIdCatalogo, aNumCampo, aDato)
```

Funciones de Direcciones

Bajo Nivel – Lectura/Escritura

fInsertaDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fInsertaDireccion ()
```

fEditaDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

```
lError = fBuscaDireccionEmpresa ()
If lError <> 0 Then
    MensajeError lError
Else
    fEditaDireccion ()
End If
```

fGuardaDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

fCancelarModificacionDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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
```

fLeeDatoDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoDireccion (*aCampo*, *aValor*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino
aValor	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 Direcciones.

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
```

Bajo Nivel – Búsqueda/Navegación

fBuscaDireccionEmpresa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fBuscaDireccionEmpresa ()
```

fBuscaDireccionCteProv ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fBuscaDireccionCteProv (lCodCteProv, lTipoDireccion)
If lError <> 0 Then
    MensajeError lError
End If
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fBuscaDireccionDocumento ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaDireccionDocumento (*aIdDocumento, aTipoDireccion*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdDocumento	Entero largo	Por valor	Identificador del documento.
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 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.

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

fPosPrimerDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerDireccion ()
```

fPosUltimaDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Direcciones.

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

```
lError = fPosUltimaDireccion ()
```

fPosSiguienteDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteDireccion ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosAnteriorDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorDireccion ()
```

fPosBOFDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lInicioTablaDirs = fPosBOFDireccion ()
```

fPosEOFDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lFinTablaDirs = fPosEOFDireccion ()
```

Alto Nivel – Lectura/Escritura

fAltaDireccion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaDireccion (*aIdDireccion, astDireccion*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdDireccion	Entero	Por referencia	Identificador de la dirección.
astDireccion	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.

aIdDireccion: Al finalizar la función este parámetro contiene el identificador del nuevo producto.

Descripción Esta función da de alta una nueva dirección.

Ejemplo El siguiente código da de alta una nueva dirección.

```
fAltaDireccion (llIdDireccion, ltDireccion)
```

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 **CONTPAQi® Comercial 2.0.0**

Sintaxis fActualizaDireccion (*astDireccion*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	astDireccion	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.

```
fActualizaDireccion (ltDireccion)
```

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 **CONTPAQi® Comercial 2.0.0**

Sintaxis fLlenaRegistroDireccion (*astDireccion, aEsAlta)*

Parámetros

Nombre	Tipo	Uso	Descripción
astDireccion	tDireccion	Por valor	Tipo de dato abstracto.
aEsAlta	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)
```

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 **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Cadena	Por valor	Código del producto.
aCodigoAlmacen	Cadena	Por valor	Código del almacén.
aAnio	Cadena	Por valor	Año.
aMes	Cadena	Por valor	Mes.
aDia	Cadena	Por valor	Día.
aExistencia	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.

```
lError = fRegresaExistencia (lCodigoProducto, lCodigoAlmacen, lAnio, lMes, lDia, lExistencia)
If lError <> 0 Then
    MensajeError lError
End If
```

fRegresaExistenciaCaracteristicas ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Cadena	Por valor	Código del producto.
aCodigoAlmacen	Cadena	Por valor	Código del almacén.
aAnio	Cadena	Por valor	Año.
aMes	Cadena	Por valor	Mes.
aDia	Cadena	Por valor	Día.
aValorCaracteristica1	Cadena	Por valor	Valor característica 1.
aValorCaracteristica2	Cadena	Por valor	Valor característica 2.
aValorCaracteristica3	Cadena	Por valor	Valor característica 3.
aExistencia	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 en una fecha determinada.

Ejemplo

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

```
lError = fRegresaExistenciaCaracteristicas (lCodigoProducto, lCodigoAlmacen, lAnio, lMes, lDia, lValorCaracteristica1, lValorCaracteristica2, lValorCaracteristica3, lExistencia)

If lError <> 0 Then

    MensajeError lError

End If
```

Funciones de Costo Histórico

Bajo Nivel – Lectura/Escritura

fRegresaCostoPromedio ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Cadena	Por valor	Código del producto.
aCodigoAlmacen	Cadena	Por valor	Código del almacén. 0 (cero) – Todos los almacenes.
aAnio	Cadena	Por valor	Año.
aMes	Cadena	Por valor	Mes.
aDia	Cadena	Por valor	Día.
aCostoPromedio	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.

```
lError = fRegresaCostoPromedio (lCodigoProducto, lCodigoAlmacen, lAnio, lMes, lDia, lCostoP)
If lError <> 0 Then
    MensajeError lError
End If
```

fRegresaUltimoCosto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Cadena	Por valor	Código del producto.
aCodigoAlmacen	Cadena	Por valor	Código del almacén. 0 (cero) – Todos los almacenes.
aAnio	Cadena	Por valor	Año.
aMes	Cadena	Por valor	Mes.
aDia	Cadena	Por valor	Día.
aUltimoCosto	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.

```
lError = fRegresaUltimoCosto (lCodigoProducto, lCodigoAlmacen, lAnio, lMes, lDia, lUltimoCosto)
If lError <> 0 Then
    MensajeError lError
End If
```

Bajo Nivel – Lectura/Escritura, *continúa…*

fRegresaCostoEstandar ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fRegresaCostoEstandar (*aCodigoProducto, aCostoEstandar*)

Parámetros

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

```
lError = fRegresaCostoEstandar (lCodigoProducto,  lCostoEstandar)
If lError <> 0 Then
    MensajeError lError
End If
```

fRegresaCostoCapa ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoProducto	Cadena	Por valor	Código del producto.
aCodigoAlmacen	Cadena	Por valor	Código del almacén.
aUnidades	Doble	Por valor	Unidades a costear.
aImporteCosto	Cadena	Por referencia	Importe del costo de las unidades recibidas.

Retorna Valores enteros:

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

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

aImporteCosto: 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.

```
lError = fRegresaCostoCapa (lCodigoProducto, lCodigoAlmacen, lUnidades,
lImporteCosto)
If lError <> 0 Then
    MensajeError lError
End If
```

Funciones de Conceptos de Documentos

Bajo Nivel – Lectura/Escritura

fLeeDatoConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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
```

fEditaConceptoDocto()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 lSerie.
- Asigna al campo CIDDIRSUCU el contenido de la variable lDireccion.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
lerror = fBuscaConceptoDocto(lConcepto)
lerror = fEditaConceptoDocto
lerror = fSetDatoConceptoDocto("CSERIEPO01", lSerie)
lerror = fSetDatoConceptoDocto("CIDDIRSUCU", lDireccion)
lerror = fGuardaConceptoDocto()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros	Nombre	Tipo	Uso	Descripción
	aCampo	Cadena	Por valor	Campo destino.
	aValor	Cadena	Por referencia	Valor de lectura.

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 lSerie.
- Asigna al campo CIDDIRSUCU el contenido de la variable lDireccion.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
lerror = fBuscaConceptoDocto(lConcepto)
lerror = fEditaConceptoDocto
lerror = fSetDatoConceptoDocto("CSERIEPO01", lSerie)
lerror = fSetDatoConceptoDocto("CIDDIRSUCU", lDireccion)
lerror = fGuardaConceptoDocto()
```

fGuardaConceptoDocto()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 lSerie.
- Asigna al campo CIDDIRSUCU el contenido de la variable lDireccion.
- Guarda los cambios al concepto con la función fGuardaConceptoDocto.

```
lerror = fBuscaConceptoDocto(lConcepto)
lerror = fEditaConceptoDocto
lerror = fSetDatoConceptoDocto("CSERIEPO01", lSerie)
lerror = fSetDatoConceptoDocto("CIDDIRSUCU", lDireccion)
lerror = fGuardaConceptoDocto()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fRegresPorcentajeImpuesto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fRegresPorcentajeImpuesto (*aIdConceptoDocumento, aIdClienteProveedor, aIdProducto, aPorcentajeImpuesto*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdConceptoDocumento	Entero	Por valor	Identificador del concepto del documento.
aIdClienteProveedor	Entero	Por valor	Identificador del cliente o proveedor.
aIdProducto	Entero	Por valor	Identificador del producto.
aPorcentajeImpuesto	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.

aPorcentajeImpuesto: Al finalizar la función este parámetro contiene el porcentaje del impuesto requerido.

Descripción 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 = fRegresPorcentajeImpuesto (lIdConceptoDocumento, lIdClienteProveedor,
lIdProducto,   lPorcentajeImpuesto)
```

Bajo Nivel – Búsqueda/Navegación

fBuscaConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaConceptoDocto (*aCodConcepto*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCodConcepto	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)
```

fBuscaIdConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdConceptoDocto (*aIdConcepto*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdConcepto	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.

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosPrimerConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerConceptoDocto ()
```

fPosUltimaConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Conceptos.

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

```
lError = fPosUltimaConceptoDocto ()
```

fPosSiguienteConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteConceptoDocto ()
```

fPosAnteriorConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorConceptoDocto ()
```

fPosBOFConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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**.

```
lInicioTablaConcepto = fPosBOFConceptoDocto ()
```

fPosEOFConceptoDocto ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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**.

```
lFinTablaConcepto = fPosEOFConceptoDocto ()
```

Funciones del Catálogo de Clasificaciones

Bajo Nivel – Lectura/Escritura

fEditaClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fBuscaIdClasificacion (lIdClasificacion)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaClasificacion ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fGuardaClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 fEditaClasificacion () y se graban los valores en los campos correspondientes.

```
fGuardaClasificacion ()
```

fCancelarModificacionClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fActualizaClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fActualizaClasificacion (*aClasificacionDe, aNumClasificacion, aNombreClasificacion*)

Parámetros

Nombre	Tipo	Uso	Descripción
aClasificacionDe	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacen 5 – Producto.
aNumClasificacion	Entero	Por valor	Numero de la clasificacion (1-6)
aNombreClasificacion	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 una clasificación

Ejemplo El siguiente código actualiza el valor de la clasificación

```
fActualizaClasificacion (aClasificacionDe, aNumClasificacion, aNombreClasificacion)
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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 clasificación, en caso de que la encuentre, guarda el contenido del campo **cNombreC01** en la variable la variable **lClasif** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdClasificacion ()
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoClasificacion ("cNombreC01", lClasif, 60)
End If
```

fSetDatoClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoClasificacion (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 clasificación, en caso de que la encuentre escribe el contenido de la variable **lClasif** en el campo **cNombreC01** de la tabla de clasificaciones; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdClasificacion ()
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoClasificacion ("cNombreC01 ", lClasif)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaClasificacion (*aClasificacionDe, aNumClasificacion*)

Parámetros

Nombre	Tipo	Uso	Descripción
aClasificacionDe	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacen 5 – Producto.
aNumClasificacion	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)
```

fBuscaIdClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdClasificacion (*aIdClasificacion*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdClasificacion	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 una clasificación por su Identificador.

Ejemplo El siguiente código busca una clasificación por su identificador.

```
fBuscaIdClasificacion (lIdClasificacion)
```

fPosPrimerClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosUltimoClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

fPosSiguienteClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 ()
```

fPosAnteriorClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosAnteriorClasificacion ()
```

fPosBOFClasificacion ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 lInicioTablaVC el resultado de la función fPosBOFClasificacion.

```
lInicioTablaC = fPosBOFClasificacion ()
```

fPosEOFClasificacion ()

Disponibilidad	**CONTPAQi® Comercial 2.0.0**
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 lFinTablaVC el resultado De la función fPosEOFClasificacion.

```
lFinTablaC = fPosEOFClasificacion()
```

Funciones del Catálogo de Valores de Clasificaciones

Bajo Nivel – Lectura/Escritura

fInsertaValorClasif ()

Disponibilidad	**CONTPAQi® Comercial 2.0.0.**
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.

```
fInsertaValorClasif ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fEditaValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 una clasificacion por su Identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
lError = fBuscaIdValorClasif (lIdValorClasif)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaValorClasif ()
End If
```

fGuardaValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fBorraValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 valor de clasificación por su identificador, si lo encuentra lo borra, en caso contrario envía el mensaje de error correspondiente

```
lError = fBuscaIdValorClasif (lIdValorClasif)
If lError <> 0 Then
    MensajeError lError
Else
    fBorraValorClasif ()
End If
```

fCancelarModificacionValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fEliminarValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fEliminarValorClasif (*aClasificacionDe, aNumClasificacion, aCodValorClasif*)

Parámetros

Nombre	Tipo	Uso	Descripción
aClasificacionDe	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacen 5 – Producto.
aNumClasificacion	Entero	Por valor	Numero de la clasificacion (1-6)
aCodValorClasif	Cadena	Por valor	Código del Valor Clasificacion 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

```
lError = fEliminarValorClasif (lClasificacionDe, lNumClasificacion, lCodValorClasif)

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

fSetDatoValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoValorClasif (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 **lCasif** en el campo **cIdClasi01** de la tabla de Valores de Clasificación; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdValorClasif (lIdValorClasif)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoValorClasif ("cIdClasi01", lCasif)
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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 **cIdClasi01** en la variable la variable **lCasif** con una longitud de 11 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscarIdMovimiento (lIdMovto)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoValorClasif ("cIdClasi01", lCasif, 11)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaClasificacion (*aClasificacionDe, aNumClasificacion, aCodValorClasif*)

Parámetros

Nombre	Tipo	Uso	Descripción
aClasificacionDe	Entero	Por valor	Clasificación de 1 – Agente 2 – Cliente 3 – Proveedor 4 – Almacen 5 – Producto.
aNumClasificacion	Entero	Por valor	Numero de la clasificacion (1-6)
aCodValorClasif	Cadena	Por valor	Código del Valor Clasificacion 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 clasificación de acuerdo a los parámetros recibidos y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca una clasificación.

```
fBuscaClasificacion (lClasificacionDe, lNumClasificacion)
```

fBuscaIdValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdValorClasif (*aIdValorClasif*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdValorClasif	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)
```

fPosPrimerValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerValorClasif ()
```

fPosUltimoValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Valores de Clasificación.

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

```
fPosUltimoValorClasif()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosSiguienteValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteValorClasif ()
```

fPosAnteriorValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorValorClasif ()
```

fPosBOFValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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**.

```
lInicioTablaVC = fPosBOFValorClasif ()
```

fPosEOFValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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**.

```
lFinTablaVC = fPosEOFValorClasif ()
```

Alto Nivel – Lectura/Escritura

fAltaValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaValorClasif (*aIdValorClasif, astValorClasif*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdValorClasif	Entero	Por referencia	Identificador de la dirección.
astValorClasif	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.

aIdValorClasif: 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.

```
fAltaValorClasif (lIdValorClasif, lstValorClasif)
```

fActualizaValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fActualizaProducto (*aCodigoValorClasif, astValorClasif*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoValorClasif	Cadena	Por valor	Código del valor de clasificación.
astValorClasif	*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 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)
```

fLlenaRegistroValorClasif ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fLlenaRegistroValorClasif (*astValorClasif)*

Parámetros

Nombre	Tipo	Uso	Descripción
astValorClasif	*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 Esta función asigna al registro de la base de datos los valores de la estructura e datos del valor de clasificación.

Ejemplo El siguiente código asigna al registro de la base de datos los valores de la estructura e datos del valor de clasificación.

```
fLlenaRegistroValorClasif (lstValorClasif)
```

Funciones Catálogo de Unidades de Medida y Peso

Bajo Nivel – Lectura/Escritura

fInsertaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fEditaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

```
lError = fBuscaIdUnidad(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaUnidad ()
End If
```

fGuardaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 **fInsertaUnidad ()** o **fEditaUnidad()** y se graban los valores en los campos correspondientes.

```
fGuardaUnidad ()
```

fBorraUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 una unidad por su identificador, si la encuentra la borra, en caso contrario envía el mensaje de error correspondiente

```
lError = fBuscaIdUnidad(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    fBorraUnidad ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fCancelarModificacionUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 ()
```

fEliminarUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fSetDatoUnidad (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 **lNombreU** en el campo **cNombreU01**de la tabla de Medida y Peso; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdUnidad(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoUnidad ("cNombreU01", lNombreU)
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino.
	aValor	Cadena	Por referencia	Valor de lectura.
	aLen	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 **lNombreU** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdUnidad(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoUnidad ("cNombreU01", lNombreU, 60)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaUnidad (*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 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)
```

fBuscaIdUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fBuscaIdUnidad (*aIdUnidad*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdValorClasif	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 (aIdUnidad)
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosPrimerUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosPrimerUnidad ()
```

fPosUltimoUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 último registro de la tabla de Unidades de Medida y Peso.

Ejemplo El siguiente código indica a la aplicación que se posicione en el último registro de la tabla de Unidades de Medida y Peso.

```
fPosUltimoUnidad ()
```

fPosSiguienteUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosSiguienteUnidad ()
```

fPosAnteriorUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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.

```
lError = fPosAnteriorUnidad ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosBOFUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 **lInicioTablaUnidades** el resultado de la función **fPosBOFUnidad**.

```
lInicioTablaUnidades = fPosBOFUnidad ()
```

fPosEOFUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

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 **lFinTablaUnidades** el resultado

De la función **fPosEOFUnidad**.

```
lFinTablaUnidades = fPosEOFUnidad ()
```

Alto Nivel – Lectura/Escritura

fAltaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fAltaUnidad (*aIdUnidad,LPFREGUNIDAD astUnidad*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdUnidad	Entero	Por referencia	Identificador de la unidad.
astUnidad	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.

aIdUnidad: Al finalizar la función este parámetro contiene el identificador del nuevo valor de clasificación.

Descripción Esta función da de alta una nueva unidad de medida y peso.

Ejemplo El siguiente código da de alta una nueva unidad de medida y peso.

```
fAltaUnidad (lIdValorClasif, lstValorClasif)
```

fActualizaUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fActualizaUnidad (*aNombreUnidad, astUnidad*)

Parámetros

Nombre	Tipo	Uso	Descripción
aNombreUnidad	Cadena	Por valor	Nombre de la unidad.
astUnidad	*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.

Descripción Esta función actualiza la unidad de medida y peso del registro especificado por el parámetro *aCodigoValorClasif*.

Ejemplo El siguiente código actualiza la unidad de medida y peso del registro especificado por el parámetro *aCodigoValorClasif*.

```
fActualizaUnidad (lNombreUnidad, lstValorClasif)
```

fLlenaRegistroUnidad ()

Disponibilidad **CONTPAQi® Comercial 2.0.0.**

Sintaxis fLlenaRegistroUnidad (*astUnidad)*

Parámetros

Nombre	Tipo	Uso	Descripción
astUnidad	*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.

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 de la unidad de medida peso.

```
fLlenaRegistroUnidad (lstValorClasif)
```

Funciones Catálogo de Agentes

Bajo Nivel – Lectura/Escritura

fInsertaAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fInsertaAgente ()
```

fEditaAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 el agente por su identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
lError = fBuscaAgente (lCodigoAgente)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaAgente ()
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fGuardaAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fGuardaAgente ()
```

fCancelarModificacionAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fCancelarModificacionAgente ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetDatoAgente (*aCampo*, *aValor*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino
aValor	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 **lNombreA** en el campo **cNombreA01** del catálogo de Agentes; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaAgente (lCodigoAgente)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoAgente ("cNombreA01", lNombreA)
End If
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fLeeDatoAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros	Nombre	Tipo	Uso	Descripción
	aCampo	Cadena	Por valor	Campo destino.
	aValor	Cadena	Por referencia	Valor de lectura.
	aLen	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 **lNombreA** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdAgente(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoAgente ("cNombreA01", lNombreA, 60)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscaAgente (*aCodigoAgente*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCodigoAgente	Cadena	Por valor	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)
```

fBuscaIdAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscaIdAgente (*aIdAgente*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aIdAgente	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 un agente por su Identificador y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca un agente por su identificador.

```
fBuscaIdAgente (lIdAgente)
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosPrimerAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosPrimerAgente ()
```

fPosUltimoAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 último registro de la tabla de Agentes.

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

```
fPosUltimoAgente ()
```

fPosSiguienteAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosSiguienteAgente ()
```

fPosAnteriorAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosAnteriorAgente ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosBOFAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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**.

```
lInicioTablaAgentes = fPosBOFAgente ()
```

fPosEOFAgente ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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**.

```
lFinTablaAgentes = fPosEOFAgente ()
```

Funciones Catálogo de Almacenes

Bajo Nivel – Lectura/Escritura

fInsertaAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

```
fInsertaAlmacen ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fEditaAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 un almacen por su identificador, si lo encuentra lo activa en modo edición, en caso de no encontrarlo envía el mensaje de error correspondiente

```
lError = fBuscaAlmacen (lCodigoAlmacen)
If lError <> 0 Then
    MensajeError lError
Else
    fEditaAlmacen ()
End If
```

fGuardaAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fGuardaAlmacen ()
```

---

fCancelarModificacionAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
fCancelarModificacionAlmacen ()
```

Bajo Nivel – Lectura/Escritura, *continúa...*

fSetDatoAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fSetDatoAlmacen (*aCampo*, *aValor*)

Parámetros	**Nombre**	**Tipo**	**Uso**	**Descripción**
	aCampo	Cadena	Por valor	Campo destino
	aValor	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 **lNombreA** en el campo **cNombreA01** del catálogo de Almacenes; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaAlmacen (lCodigoAgente)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fSetDatoAlmacen ("cNombreA01", lNombreA)
End If
```

fLeeDatoAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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

Parámetros

Nombre	Tipo	Uso	Descripción
aCampo	Cadena	Por valor	Campo destino.
aValor	Cadena	Por referencia	Valor de lectura.
aLen	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 **lNombreA** con una longitud de 60 caracteres; en caso contrario muestra el mensaje de error correspondiente.

```
lError = fBuscaIdUnidad(lIdUnidad)
If lError <> 0 Then
    MensajeError lError
Else
    lError = fLeeDatoAlmacen ("cNombreA01", lNombreA, 60)
End If
```

Bajo Nivel – Búsqueda/Navegación

fBuscaAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscaAlmacen (*aCodigoAlmacen*)

Parámetros

Nombre	Tipo	Uso	Descripción
aCodigoAlmacen	Cadena	Por valor	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 almacen por su código y se posiciona en el registro correspondiente.

Ejemplo El siguiente código busca un Almacén.

```
fBuscaAlmacen (lCodigoAlmacen)
```

fBuscaIdAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

Sintaxis fBuscaIdAlmacen (*aIdAgente*)

Parámetros

Nombre	Tipo	Uso	Descripción
aIdAgente	Entero	Por valor	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.

```
fBuscaIdAlmacen (lCodigoAlmacen)
```

fPosPrimerAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosPrimerAlmacen ()
```

fPosUltimoAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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 último registro de la tabla de Almacenes.

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

```
fPosUltimoAlmacen ()
```

Bajo Nivel – Búsqueda/Navegación, *continúa...*

fPosSiguienteAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosSiguienteAlmacen ()
```

fPosAnteriorAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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.

```
lError = fPosAnteriorAlmacen ()
```

fPosBOFAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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**.

```
lInicioTablaAlmacenes = fPosBOFAlmacen ()
```

fPosEOFAlmacen ()

Disponibilidad **CONTPAQi® Comercial 2.0.0**

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**.

```
lFinTablaAlmacenes = fPosEOFAlmacen ()
```

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
kLonSerieCertSAT	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.
kLongCadOrigCompISAT	500	Longitud máxima de caracteres para la cadena original.

Nombre	Longitud	Descripción
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.
kLogitudLugarExpedicion	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.
aImporte	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	Doble	NA	Importes de gastos para las compras
aGasto2	Doble	NA	Importes de gastos para las compras
aGasto3	Doble	NA	Importes de gastos para las compras

Llave del Documento – RegMovimiento – tRegLlaveMov

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	Códogo 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 clasificacuión

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	Códogo 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 clasificacuión

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).
aPorcDescto1	Doble	NA	Porcentaje del Descuento 1
aImporteDescto1	Doble	NA	Importe del Descuento 1
aPorcDescto2	Doble	NA	Porcentaje del Descuento 2
aImporteDescto2	Doble	NA	Importe del Descuento 2
aPorcDescto3	Doble	NA	Porcentaje del Descuento 3
aImporteDescto3	Doble	NA	Importe del Descuento 3
aPorcDescto4	Doble	NA	Porcentaje del Descuento 4
aImporteDescto4	Doble	NA	Importe del Descuento 4
aPorcDescto5	Doble	NA	Porcentaje del Descuento 5
aImporteDescto5	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 clasificacuión

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 Caracteristicas – Caracteristicas – tCaracteristicas

Campo	Tipo	Longitud	Descripción
aUnidades	Doble	NA	Unidades del movimiento.
aValorCaracteristica1	Cadena	kLongDescripcion + 1	Valor de la xaracteristica 1 del movimiento.
aValorCaracteristica2	Cadena	kLongDescripcion + 1	Valor de la xaracteristica 2 del movimiento.
aValorCaracteristica3	Cadena	kLongDescripcion + 1	Valor de la xaracteristica 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 exixtencia.
cMetodoCosteo	Entero	NA	1. Costo Promedio Base a Entradas, Costo Promedio Base a Entradas Almacen Último costo, 2. UEPS, 3. PEPS, 4. Costo específico, 5. 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.

Campo	Tipo	Longitud	Descripción
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 caracteristica 1.
cNombreCaracteristica2	Cadena	kLongAbreviatura + 1	Nombre de la caracteristica 2.
cNombreCaracteristica3	Cadena	kLongAbreviatura + 1	Nombre de la caracteristica 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.

Campo	Tipo	Longitud	Descripción
cFechaBaja	Cadena	kLongFecha + 1	Fecha de baja.
cFechaUltimaRevision	Cadena	kLongFecha + 1	Fecha de última revisión.
cLimiteCreditoCliente	Doble	NA	Limite 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	Interes moratorio.
cDiaPago	Entero	NA	Día de pago.
cDiasRevision	Entero	NA	Días de revisión.
cMensajeria	Cadena	kLongDesCorta + 1	Mensajeria.
cCuentaMensajeria	Cadena	kLongDescripcion + 1	Cuenta de mensajeria.
cDiasEmbarqueCliente	Entero	NA	Dias 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.

Cliente/Proveedor – RegCteProv – tCteProv

Campo	Tipo	Longitud	Descripción
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	Limite de credito del proveedor.
cDiasCreditoProveedor	Entero	NA	Días de credito 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 interes 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.

Campo	Tipo	Longitud	Descripción
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 Clasificacion – RegValorClasificacion – tValorClasificacion

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.

Como declarar constantes C# y VB.Net

Al trabajar con las estructuras es necesario tener declaradas nuestras constantes de longitud.

Ejemplo:

C#

```
public const int kLongFecha = 24;
public const int kLongSerie = 12;
public const int kLongCodigo = 31;
public const int kLongNombre = 61;
public const int kLongReferencia = 21;
public const int kLongDescripcion = 61;
public const int kLongCuenta = 101;
public const int kLongMensaje = 3001;
public const int kLongNombreProducto = 256;
public const int kLongAbreviatura = 4;
public const int kLongCodValorClasif = 4;
```

VB.Net

```
Public Const kLongCodigo As Integer = 30 + 1
Public Const kLongNombre As Integer = 60 + 1
Public Const kLongNombreProducto As Integer = 255 + 1
Public Const kLongFecha As Integer = 23 + 1
Public Const kLongAbreviatura As Integer = 3 + 1
Public Const kLongCodValorClasif As Integer = 3 + 1
Public Const kLongTextoExtra As Integer = 50 + 1
Public Const kLongNumSerie As Integer = 11 + 1
Public Const kLongReferencia As Integer = 20 + 1
```

Como declarar estructuras C# y VB.Net

Sera necesario declarar todos los campos en la estructura, independiente mente de cuantos campos se utilizaran para cargar información en el proyecto.

C#

```
[StructLayout(LayoutKind.Sequential, CharSet = CharSet.Ansi, Pack = 4)]
    public struct tDocumento
    {

        public Double aFolio;
        public int aNumMoneda;
        public Double aTipoCambio;
        public Double aImporte;
        public Double aDescuentoDoc1;
        public Double aDescuentoDoc2;
        public int aSistemaOrigen;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongCodigo)]
        public String aCodConcepto;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongSerie)]
        public String aSerie;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongFecha)]
        public String aFecha;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongCodigo)]
        public String aCodigoCteProv;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongCodigo)]
        public String aCodigoAgente;
        [MarshalAs(UnmanagedType.ByValTStr, SizeConst = constantes.kLongReferencia)]
        public String aReferencia;
        public int aAfecta;
        public int aGasto1;
        public int aGasto2;
        public int aGasto3;

    }
```

VB.Net

```
<StructLayout(LayoutKind.Sequential, CharSet:=CharSet.Ansi, Pack:=4)>
  Public Structure tDocumento
    Public aFolio As Double
    Public aNumMoneda As Integer
    Public aTipoCambio As Double
    Public aImporte As Double
    Public aDescuentoDoc1 As Double
    Public aDescuentoDoc2 As Double
    Public aSistemaOrigen As Integer
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongCodigo)> Public aCodConcepto As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongNumSerie)> Public aSerie As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongFecha)> Public aFecha As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongCodigo)> Public aCodigoCteProv As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongCodigo)> Public aCodigoAgente As String
    <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=kLongReferencia)> Public aReferencia As String
    Public aAfecta As Integer
    Public aGasto1 As Double
    Public aGasto2 As Double
    Public aGasto3 As Double
 End Structure
```

Ejemplos de SDK

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;
using System.Runtime.InteropServices; // using necesario para DLLImport
using Microsoft.Win32; //SetCurrentDiectory

namespace SDK_Documento
{
   class Program
   {
      //Declaraciones Funciones en C#

      [DllImport("MGWSERVICIOS.DLL")]
      public static extern int fSetNombrePAQ(string aNombrePAQ);
      [DllImport("MGWSERVICIOS.DLL")]
      public static extern int fAbreEmpresa(string Directorio);
      [DllImport("MGWSERVICIOS.DLL")]
      public static extern void fTerminaSDK();
      [DllImport("MGWSERVICIOS.DLL")]
      public static extern void fCierraEmpresa();
      [DllImport("MGWSERVICIOS.DLL")]
      public static extern void fError(int NumeroError, StringBuilder Mensaje, int Longitud);
      [DllImport("KERNEL32")]
      public static extern int SetCurrentDirectory(string pPtrDirActual);

      static void Main(string[] args)
      {
         //Tomar resultado Función Manejo Errores
         int IResultado;
         //asignar la Ruta de DLL MGWServicios
         string szRegKeySistema = @"SOFTWARE\\Computación en Acción, SA CV\\CONTPAQ I COMERCIAL ";

         //Establece la ruta donde se encuentar el archivo MGWSERVICIOS.DLL
         RegistryKey keySistema = Registry.LocalMachine.OpenSubKey(szRegKeySistema);
         object IEntrada = keySistema.GetValue("DirectorioBase");
         SetCurrentDirectory(IEntrada.ToString());

         //Sistemas : CONTPAQi® Comercial
        IResultado = fSetNombrePAQ("CONTPAQ I COMERCIAL ");
        if (IResultado != 0)
        {
           rError(IResultado);
        }
        else
        {
```

```
            //Se abre empresa
            fAbreEmpresa("C:\\Compac\\Empresas\\comTrebol");
              if (lResultado != 0)
              {
                 rError(lResultado);
              }
              else
              {
                 Console.WriteLine("Se Abrio Empresa Correctamente...");
              }
            //Se verifica Opción
              Console.WriteLine("¿Deseas Cerrar Empresa y Terminar SDK ? 1 = Si || 2 = No");
                if (Console.ReadLine() == "1")
                {
                   fCierraEmpresa(); // se Cierra Empresa
                   fTerminaSDK();    //Se termina SDK
                }

        }//fin else fSetNombrePAQ


    }//fin Main


    // Función para el manejo de errores en SDK
    public static void rError(int iError)
    {
       StringBuilder sMensaje = new StringBuilder(512);

       if (iError != 0)
       {
          fError(iError, sMensaje, 512);
          Console.WriteLine("Error: " + sMensaje);
          Console.ReadKey();
       }
    }


}//fin clase principal

}//fin Namespace
```

VB.Net

```
Imports System.Runtime.InteropServices 'Import Necesario para DLLImport
Imports System.Text

Module Module1

    'Declaraciones de las Funciones VB.net
    Public Declare Function fSetNombrePAQ Lib "MGWSERVICIOS.DLL" (ByVal aNombrePAQ As String) As Integer
    Public Declare Function fAbreEmpresa Lib "MGWSERVICIOS.DLL" (ByVal aEmpresa As String) As Integer
    Public Declare Sub fCierraEmpresa Lib "MGWSERVICIOS.DLL" ()
    Public Declare Sub fTerminaSDK Lib "MGWSERVICIOS.DLL" ()
    Public Declare Sub fError Lib "MGWSERVICIOS.DLL" (ByVal aNumError As Integer, aMensaje As StringBuilder, ByVal aLen As
Integer)
    Public Declare Function SetCurrentDirectory Lib "KERNEL32" Alias "SetCurrentDirectoryA" (ByVal pPtrDirActual As String) As Integer

    Sub Main()
        'Variable para tomar resultado de la función
        Dim lResultado As Integer
        Dim sLlaveSis = "HKEY_LOCAL_MACHINE\SOFTWARE\Computación en Acción, SA CV\CONPAQ I COMERCIAL"

        'Establece la ruta donde se encuentar el archivo MGWSERVICIOS.DLL
        Dim lRutaBinarios = My.Computer.Registry.GetValue(sLlaveSis, "DIRECTORIOBASE", Nothing)
        SetCurrentDirectory(lRutaBinarios)

        'Sistemas : CONTPAQi® Comercial
        lResultado = fSetNombrePAQ("CONTPAQ I COMERCIAL")
        If lResultado <> 0 Then
            Console.WriteLine(rError(lResultado))
            Exit Sub
        Else
            'Se abre empresa
            fAbreEmpresa("C:\\Compac\\Empresas\\comTrebol")
            If lResultado <> 0 Then
                Console.WriteLine(rError(lResultado))
            Else
                Console.WriteLine("Se Abrio Empresa Correctamente...")
            End If
        End If
        'Se verifica Opción
        Console.WriteLine("¿Deseas Cerrar Empresa y Terminar SDK ? 1 = Si || 2 = No")
```

```
        If (Console.ReadLine() = "1") Then
            fCierraEmpresa()
            fTerminaSDK()
        End If


    End Sub

    ' Función para el manejo de errores en el SDK
    Public Function rError(ByRef aError As Integer) As String
        Dim aMensaje As StringBuilder = New StringBuilder(512)

        ' Recupera el mensaje de error del SDK
        fError(aError, aMensaje, 350)
        Return aMensaje.ToString()
    End Function


End Module
```

Ejemplo Agentes CONTPAQi® Comercial

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel;
using System.Runtime.InteropServices; // using necesario para DLLImport
using Microsoft.Win32; //SetCurrentDiectory

namespace SDK_Documento
{
    class Program
    {
        //Declaraciones Funciones en C#

        [DllImport("MGWSERVICIOS.DLL")]
        public static extern int fSetNombrePAQ(string aNombrePAQ);
        [DllImport("MGWSERVICIOS.DLL")]
        public static extern int fAbreEmpresa(string Directorio);
        [DllImport("MGWSERVICIOS.DLL")]
        public static extern void fTerminaSDK();
        [DllImport("MGWSERVICIOS.DLL")]
        public static extern void fCierraEmpresa();
        [DllImport("MGWSERVICIOS.DLL")]
        public static extern void fError(int NumeroError, StringBuilder Mensaje, int Longitud);
        [DllImport("KERNEL32")]
        public static extern int SetCurrentDirectory(string pPtrDirActual);

        //funciones de Agente
        [DllImport("MGWServicios.dll")]
        public static extern int fGuardaAgente();
        [DllImport("MGWServicios.dll")]
        public static extern void fSetDatoAgente(string aCampo, string aValor);
        [DllImport("MGWSERVICIOS.dll")]
        public static extern int fBuscaAgente(string aCodigoAgente);
        [DllImport("MGWSERVICIOS.dll")]
        public static extern int fEditaAgente();
```

```
static void Main(string[] args)
{
   //Tomar resultado Función Manejo Errores
   int lResultado;
   //asignar la Ruta del Sistema
   string szRegKeySistema = @"SOFTWARE\\Computación en Acción, SA CV\\CONTPAQ I COMERCIAL ";

   //Establece la ruta donde se encuentar el archivo MGWSERVICIOS.DLL
   RegistryKey keySistema = Registry.LocalMachine.OpenSubKey(szRegKeySistema);
   object lEntrada = keySistema.GetValue("DirectorioBase");
   SetCurrentDirectory(lEntrada.ToString());

   //Sistema : CONTPAQi® Comercial
  lResultado = fSetNombrePAQ("CONTPAQ I COMERCIAL ");
  if (lResultado != 0)
  {
     rError(lResultado);
  }
  else
  {
     //Se abre empresa
     fAbreEmpresa("C:\\Compac\\Empresas\\comTrebol");
        if (lResultado != 0)
        {
           rError(lResultado);
        }
        else
        {
           //Aqui va todo Proceso a Realizar con las Funciones SDK

           Console.WriteLine("Se Abrio Empresa Correctamente...");
           Console.ReadKey();
           Console.Clear();
           Console.WriteLine("\t\t=== Editar Agente ===");

           Console.Write("Codigo del Agente: ");
           lResultado = fBuscaAgente(Console.ReadLine());
              if (lResultado != 0)
              {
                 rError(lResultado);
              }
              else
              {
```

```
                    //Se pone en modo de Edición
                    fEditaAgente();

                    //Set al Campo de la BD
                    Console.Write("\tCodigo del Agente: ");
                    fSetDatoAgente("CCODIGOA01", Console.ReadLine());
                    Console.Write("\tNombre Agente: ");
                    fSetDatoAgente("CNOMBREA01", Console.ReadLine());
                    Console.Write("\tFecha Alta Agente: ");

                    //Se guarda la información
                    lResultado = fGuardaAgente();
                    if (lResultado != 0)
                    {
                        rError(lResultado);
                    }
                    else
                    {
                        Console.Clear();
                        Console.WriteLine("\t\t== Agente Modificado ==");
                        Console.ReadKey();

                    }//fin else Guarda Agente

                }//fin else Busqueda Agente

            } //fin else Abrio Empresa

             //Se verifica Opción
            Console.WriteLine("¿Deseas Cerrar Empresa y Terminar SDK ? 1 = Si || 2 = No");
               if (Console.ReadLine() == "1")
               {
                  fCierraEmpresa(); // se Cierra Empresa
                  fTerminaSDK();    //Se termina SDK
               }

        }//fin else fSetNombrePAQ
    }//fin Main


     // Función para el manejo de errores en SDK
     public static void rError(int iError)
     {
        StringBuilder sMensaje = new StringBuilder(512);
```

```
        if (iError != 0)
        {
          fError(iError, sMensaje, 512);
          Console.WriteLine("Error: " + sMensaje);
          Console.ReadKey();
        }
      }//fin Función Error


    }//fin clase principal

}//fin namespace
```