RIB Spain SA
Rodríguez San Pedro 10
28015 Madrid
+34 914 483 800
info@rib-software.es
www.rib-software.es

Lista de funciones de Presto

Este documento contiene la lista de funciones de Presto accesibles desde cualquier programa y lenguaje de programación que soporte el sistema COM.

Estas funciones permiten interactuar con una obra de Presto y realizar todo tipo de operaciones, creación, modificación y borrado de los registros, incluyendo el acceso directo a todas las opciones del programa.

A continuación les mostramos la lista de funciones, y más adelante la lista detallada.

Lista de funciones

Función	Descripción
AddCert	Crea una nueva certificación a continuación de la última existente. Si no hubiera certificaciones en la obra crea la primera. El número de la certificación y la fecha asociada a la misma las calcula internamente Presto en base al criterio de defecto (una certificación por mes). Si en el color se recibe *0xFF000000*, color inválido, se genera para la certificación un color aleatorio (el mismo que en Presto se generaría si pulsamos [*Supr]* sobre el color)
AddSpace	Agrega un espacio con el código, descripción y color inidicados y el siguiente número de espacio válido (en el orden establecido por Presto)
BeginRedo	Inicio de acciones que se pueden deshacer
Calculate	Lanza en Presto uno de los cálculos del menú “CÁLCULOS”. Sólo funciona si hay una obra abierta en modo lectura/escritura. Crea el cancelar correspondiente. Lo que pueda o no hacer el cálculo dependerá de la configuración de Presto (modulación, cálculo automático, etc)
CheckProject	Comprueba la obra abierta
Close	Cierra una obra de Presto
Command	Activa cualquier opción de menú de Presto. Puede llamarse sólo con el nombre de la opción o precediéndola con el nombre del menú separado por pipe "|". Es posible agrupar varios “Command” separándolos por punto y coma “;”
Copy	Copia un campo en otro

CreateSpace	Crea un espacio con el número, código, resumen y color indicados. Si no se indican código o resumen se generan los de defecto. Se incluye la posibilidad de asignar color para controlar las alternativas de presupuesto. Si en el color se recibe *0xFF000000*, color inválido, se genera para el espacio un color aleatorio (el mismo que en Presto se generaría si pulsamos *[Supr]* sobre el color). Si el número de espacio o el código ya existe devolverá error. Se crean todos los espacios necesarios para que no haya huecos
CreateVariable	Crea / Actualiza un registro de la tabla de "Variables".
Delete	Borra una obra
DeleteRecord	Borra un registro y mantiene la integridad relacional
Duplicate	Duplica un registro de la tabla de conceptos de la misma forma que duplicar de Presto, es decir, copiando la descomposición y demás informaciones asociadas. Para copiar simplemente un registro de esta o de otra tabla, basta con leerlo, cambiar la clave e insertarlo de nuevo
EndRedo	Fin de acciones que se pueden deshacer
EvalNum	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor numérico
EvalStr	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor de cadena
Exit	Cierra una instancia de Presto (sólo la ventana principal a la que nos referimos, no toda la aplicación)
ExportFIEBDC	Exporta la obra en formato FIEBDC
ExportRPT	Exporta un informe
FindEqual	Se posiciona en una tabla dado un valor de una clave
FindFirst	Se posiciona en el primer registro de una tabla según una clave
FindGreat	Se posiciona en el registro mayor a un valor dado de una tabla según una clave
FindGreatEqual	Se posiciona en el registro mayor o igual a un valor dado de una tabla según una clave
FindLast	Se posiciona en el último registro de una tabla según una clave
FindLessEqual	Se posiciona en el registro menor o igual a un valor dado de una tabla según una clave
FindNext	Se posiciona en el registro siguiente de una tabla
Findpos	Se posiciona en un registro de una tabla ordenada por una determinada clave, conociendo su posición
FindPrev	Se posiciona en el registro anterior de una tabla
GetElement	Se posiciona en el primer registro de una tabla

GetField	Devuelve el valor de un campo de Presto
GetFieldBinary	Devuelve el valor de un campo binario de Presto
GetFieldNum	Devuelve el valor de un campo numérico
GetFieldStr	Devuelve el valor de un campo de tipo cadena
Getpos	Obtiene la posición de un registro. La posición es un número asociado por el gestor de archivos a cada registro
GetSelection	Recorre los elementos seleccionados de una tabla
GetText	Devuelve el valor del texto con los mismos parámetros que otras funciones "Get".
ImportCad	Importa un archivo de CAD (AllPlan, ArchiCad, FIEBDC) sobre una obra de Presto
ImportFIEBDC	Importa la obra en formato FIEBDC
InfoCode	Información de un concepto de Presto
InitRecord	pone a cero un registro del tipo indicado en la tabla. Este registro se puede llenar luego con valores y grabar con las demás funciones
InitVar	Suprime variables declaradas con float o con char dentro de una expresión usada en EvalStr o EvalNum
InsertAttachment	Inserta un archivo en Presto a partir de un buffer y lo asocia con un registro de una tabla por medio del código
InsertAttachmentFromPath	Inserta un archivo en Presto a partir de un *path* y lo asocia con un registro de una tabla por medio del código
InsertRecord	Inserta en la tabla un registro con la clave que tenga en ese momento. Si la clave ya existe y la tabla no admite duplicados, devolverá un error
LogBegin	Inicia un proceso de log en Presto
LogEnd	Cierra el proceso de log en Presto y muestra la ventana con todos los mensajes agregados. Incluye un mensaje con la fecha y hora de finalización
LogMsg	Agrega un mensaje al sistema de log
New	Crea y abre una obra de Presto
Open	Abre una obra de Presto
PrintRPT	Imprime un informe
Quit	Cierra la aplicación de Presto
Recalculate	Bloquea o Desbloquea los recálculos. Si está bloqueada no se hacen recálculos intermedios, a fin de evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamarla de nuevo para hacerlos todos de golpe

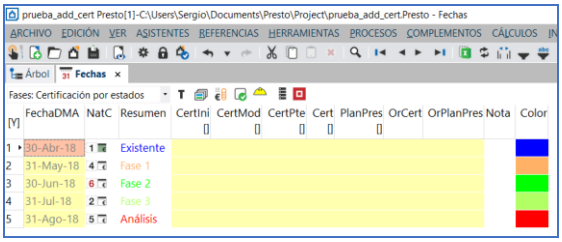
Rename	Cambia (renombra) la clave de un concepto, estando posicionados en el concepto que se quiere renombrar
SelCode	Elige un concepto de Presto
SetCharacter	Selecciona el juego ANSI (Windows) o OEM (MS-DOS). Se usa antes de leer un texto de un registro o de evaluar una expresión
SetConceptVarValue	Asigna valor a la variable identificada por su *guid* en el concepto con el código indicado. Tanto el concepto como la variable deben existir antes de asignar valor. El valor debe ser de un tipo compatible con el tipo de la variable
SetCurrentInvPeriod	Establece como certificación actual aprobada la fecha recibida
SetElement	Define un elemento. Un elemento es una forma de acceder a una tabla de una obra de Presto, similar a la utilizada en informes. Se accede a un elemento concreto con GetElement Elemento indica un número entre 1 y 25, ya que se pueden definir varios SetElement al tiempo
SetField	Asigna un valor a un campo (en la tabla solo se graba con InsertRecord o UpdateRecord)
SetFieldBinary	Asigna un valor a un campo binario de Presto
SetModal	pone y quita el modo modal de Presto (mientras está puesto, la ventana de Presto está bloqueada): Bloquea o Desbloquea el uso de Presto
SetTakeoffVarValue	Asigna valor a la variable identificada por su *guid* en la línea de medición identificada por su *guidElement* y su *guidAux*. Tanto la línea de medición como la variable deben existir antes de asignar valor. El valor debe ser de un tipo compatible con el tipo de la variable
SetText	Rellena un campo de texto.
SetUniqueVarValue	Asigna un valor de tipo de asignación "Única" a una variable. La variable debe existir previamente.
SetUpdateScreen	Si está desactivada, los cambios que deberían reflejarse en las ventanas visibles no se actualizan, incluyendo su recorrido, a fin de evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamar a SetUpdateScreen con el parámetro 1 para reflejar todos los cambios de una sola vez
SetVar	Modifica una variable de Presto
UpdateRecord	Actualiza en la tabla un registro cuyos campos han cambiado. Si el campo clave hubiera sido modificado, cambiará también en la tabla, pero sin mantener la integridad relacional de las tablas asociadas. Para cambiar los campos clave, usar la función Rename
UpdateScreen	Actualiza las ventanas

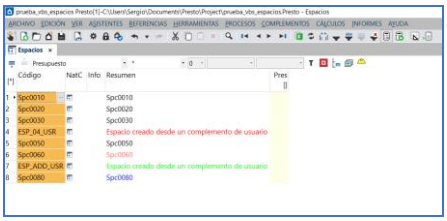
Atributos	Descripción
Property Status As Long	Después de una llamada a una función del gestor de archivos devuelve 0 todo correcto
Property ReadOnly As Long	Devuelve si la obra está en modo "solo lectura"
Property Project As String	Devuelve el nombre de path de la obra de Presto abierta
Property VersionStr As String	Devuelve el nombre completo de la versión de Presto
Property VersionNum As Long	Devuelve la versión de Presto en formato número
Property SubVersionNum As Long	Devuelve la sub-versión de Presto en formato número

A continuación, se incluye una lista detallada que contiene los parámetros, la definición de la acción o retorno y un ejemplo de uso.

Los ejemplos están hechos para un compilador de Visual Basic. Si usa VBS (Visual Basic Script), elimine el nombre de la rutina Public Sub (y su línea de fin End Sub) y las líneas de definición de variables Dim.

Lista detallada de funciones

Función	AddCert
Sintaxis	Sub AddCert(certDescription As String, certColor As Long)
Descripción	Crea una nueva certificación a continuación de la última existente. Si no hubiera certificaciones en la obra crea la primera. El número de la certificación y la fecha asociada a la misma las calcula internamente Presto en base al criterio de defecto (una certificación por mes). Si en el color se recibe *0xFF000000*, color inválido, se genera para la certificación un color aleatorio (el mismo que en Presto se generaría si pulsamos [*Supr]* sobre el color)
Parámetros	Entre corchetes se indica el valor por defecto *certDescription* [""], resumen de la certificación en la tabla Agenda. *certColor* [0xFF000000], color para la certificación en la tabla Agenda
Retorno	El número de la certificación agregada o 0 en caso de error
Ejemplo	Set po = GetObject("", "Presto.App.18") po.SetUpdateScreen(0) po.BeginRedo ' Azul po.AddCert "Existente", 16711680 po.AddCert "Fase 1" ' Verde po.AddCert "Fase 2", 65280 po.AddCert "Fase 3" ' Roja po.AddCert "Análisis", 255 po.EndRedo po.SetUpdateScreen(1) po.UpdateScreen 

Función	**AddSpace**
Sintaxis	Sub AddSpace(spcCode As String, spcDescription As String, spcColor As Long)
Descripción	Agrega un espacio con el código, descripción y color indicados y el siguiente número de espacio válido (en el orden establecido por Presto)
Parámetros	Entre corchetes se indica el valor por defecto *spcCode* [""], código del concepto de tipo espacio a crear. *spcDescription* [""], resumen para el concepto de tipo espacio. *spcColor* [0xFF000000], color para el concepto de tipo espacio
Retorno	El número del espacio añadido si se pudo crear. 0 en caso contrario
Ejemplo	Set po = GetObject("", "Presto.App.18") po.SetUpdateScreen(0) po.BeginRedo ' Espacio con número, código, descripción y color (rojo). ' Como el espacio es > 1, añadirá los espacios intermedios por defecto po.CreateSpace 4,"ESP_04_USR","Espacio creado desde un complemento de usuario",255 ' Espacio con número (el resto de valores por defecto). ' Añadirá los espacios intermedios desde el último existente po.CreateSpace 6 ' Espacios incorrectos po.CreateSpace 6 po.CreateSpace 7,"ESP_04_USR" ' Añadimos un espacio a continuación del último, con su código, resumen y color (verde) po.AddSpace "ESP_ADD_USR","Espacio creado desde un complemento de usuario",65280 ' Añadimos otro espacio sin código ni resumen, pero con color (azul) po.AddSpace "","",16711680 ' Espacio añadido incorrectamente po.AddSpace "ESP_ADD_USR" po.EndRedo po.SetUpdateScreen(1) po.UpdateScreen 

Función	**BeginRedo**
Sintaxis	Function BeginRedo() As Long
Descripción	Inicio de acciones que se pueden deshacer
Parámetros	Ninguno
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y activa la opción de deshacer. Finalmente se indica al usuario si se inició correctamente.
	
	Public Sub Comienza_Deshacer()
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.BeginRedo** = 0 then
	MsgBox "Deshacer iniciado correctamente"
	Else
	MsgBox "No se inició correctamente deshacer"
	EndIf
	End Sub

Función	**Calculate**
Sintaxis	*Sub Calculate(calcMode As Long)*
Descripción	Lanza en Presto uno de los cálculos del menú "CÁLCULOS". Sólo funciona si hay una obra abierta en modo lectura/escritura. Crea el cancelar correspondiente. Lo que pueda o no hacer el cálculo dependerá de la configuración de Presto (modulación, cálculo automático, etc)
Parámetros	*calcMode* 1: Calcular todo 2: Calcular precios 3: Calcular mediciones 4: Calcular tiempos 5: Calcular documentos 6: Calcular costes reales
Retorno	No devuelve nada

Función	**CheckProject**
Sintaxis	Sub CheckProject()
Descripción	Comprueba la obra abierta
Parámetros	Ninguno
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y comprueba la obra abierta Public Sub Comprueba_Obra() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.CheckProject** End Sub

Función	**Close**
Sintaxis	Sub Close()
Descripción	Cierra una obra de Presto
Parámetros	Ninguno
Retorno	Ninguno

Ejemplo	En este ejemplo se muestra un procedimiento que cierra la obra activa en Presto. Primero creamos y abrimos una obra de ejemplo en la ruta "C:\TEMP\EJEMPLO.PRESTOOBRA" y posteriormente si se ha creado correctamente la cerramos mostrando un mensaje.
	
	Public Sub Cierra_Obra()
	Dim po As Object
	Set po = CreateObject ("Presto.App.18")
	If po.New("C:\TEMP\EJEMPLO.PrestoObra") = 0 Then
	MsgBox "Obra creada correctamente"
	po.Close
	MsgBox "Se ha cerrado la obra"
	End If
	End Sub

Función	Command
Sintaxis	Sub Command(opcion As String)
Descripción	Activa cualquier opción de menú de Presto. Puede llamarse sólo con el nombre de la opción o precediéndola con el nombre del menú separado por pipe "|". Es posible agrupar varios commands separándolos por punto y coma ";"
Parámetros	opción es el nombre del menú de Presto que deseamos ejecutar
Retorno	Ninguno
Ejemplos	En este ejemplo se muestra un procedimiento que ejecuta el menú de ver el árbol de un presupuesto Public Sub Ver_Arbol() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.Command** ("VER|Árbol") End Sub En este ejemplo se muestra un procedimiento que ejecuta la opción de exportación a Excel Public Sub Exportar_Excel() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.Command** ("Exportar|Excel") End Sub

Función	**Copy**
Sintaxis	Sub Copy(origen As String, destino As String, [dimension1 As Long], [dimension2 As Long])
Descripción	Copia un campo en otro
Parámetros	origen es el campo a copiar destino es el campo donde queremos copiar dimension1 es la dimensión del campo origen dimension2 es la dimensión del campo destino
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que copia para todos los registros el valor de Conceptos.Código en Conceptos.Código2 Public Sub Copiar () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 **po.Copy** "Conceptos.Código", "Conceptos.Código2" po.UpdateRecord "Conceptos" Wend End Sub

Función	**CreateSpace**
Sintaxis	Sub CreateSpace(spcNumber As Integer, spcCode As String, spcDescription As String, spcColor As Long)
Descripción	Crea un espacio con el número, código, resumen y color indicados. Si no se indican código o resumen se generan los de defecto. Se incluye la posibilidad de asignar color para controlar las alternativas de presupuesto. Si en el color se recibe *0xFF000000*, color inválido, se genera para el espacio un color aleatorio (el mismo que en Presto se generaría si pulsamos *[Supr]* sobre el color). Si el número de espacio o el código ya existe devolverá error. Se crean todos los espacios necesarios para que no haya huecos
Parámetros	(entre corchetes se indica el valor por defecto) *spcNumber*, número de espacio a crear, >=1 y <= 9999. *spcCode* [""], código del concepto de tipo espacio a crear. *spcDescription* [""], resumen para el concepto de tipo espacio. *spcColor* [0xFF000000], color para el concepto de tipo espacio
Retorno	0 si pudo crear correctamente el espacio. Distinto de 0 en caso contrario
Ejemplo	Ver ejemplo en *AddSpace*

Función	**CreateVariable**
Sintaxis	Sub CreateVariable(varGuid As String, varName As String, varDescription As String, varSource As Integer, varType As Integer, varCalcType As Integer, varMinVal As Double, varStepVal As Double, varMaxVal As Double, varIsCombo As Integer, varComboList As String, varStatus As Integer, varIsReadOnly As Integer, revitparamShared As Integer, revitparamUnitType As String, revitparamDisplayUnitType As String, revitparamStorageType As String, revitparamType As String, revitparamGroup As String, revitparamSharedGuid As String), varDec As Integer, varExp As String, varCond As String)
Descripción	Crea / Actualiza un registro de la tabla de "Variables"
Parámetros	Entre corchetes se indica el valor por defecto *varGuid*, identificador único global para la variable. Si no se especifica Presto generará uno (y será siempre una creación). Si ya existe una variable con igual *guid* la actualiza con la información indicada. *varName*, nombre de la variable (puede estar repetido). *varDescription* [""], descripción de la variable. *varSource* [0], origen de la variable (de dónde proviene). 0: Usuario 1: Auxiliar 2: Interna 3: Informe 4: Plantilla Word 5: Plantilla Excel 6: Asistente 7: Contabilidad 8: Revit 10: Complemento. *varType* [1], indica el tipo de valor que puede recibir. 1: String 2: Bool 3: Integer 4: Real 5: Date 6: Keyword. El 0 indica tipo inválido. Los valores asociados a variables de tipo *Keyword* representan ocurrencias del término, pero no especifican valor para la variable. *varCalcType* [0], indica si Presto calcula valores de abajo a arriba para la variable. 0: No calcular 1: Normal. *varMinVal*, *varStepVal*, *varMaxVal* [0], en variables de tipo *Integer* o *Real* indican el valor mínimo y máximo (ambos incluídos) que pueden tomar las variables, así como el tamaño estándar del incremento del valor desde el mínimo hasta el máximo (que se utilizará en el sugerir del valor). Si máximo y mínimo son iguales e iguales a 0.0 se considera que la variable no especifica mínimo ni máximo. El incremento sólo se utiliza para el sugerir cuando su valor es mayor que 0.0 y debe cumplir <= *(máximo-mínimo)*. *varIsCombo* [0], indica con valor distinto de 0 que la variable sólo puede tomar valores de un conjunto reducido de valores posibles. Sólo es compatible con variables de tipo *Integer*, *Real* o *Cadena*. *varComboList* [""], si la variable es *combo* permite especificar, separados por *pipes* ("|"), los distintos valores que puede tomar la variable. *varStatus* [0], afecta al color de texto de la descripción. 0: Negro 1: Rojo 2: Gris 3: Verde. *varIsReadOnly* [0], con valor distinto de 0 indica que el usuario no podrá editar (en el interfaz de Presto) los valores asignados a la variable. *revitparam<*>* [0] ó [""], datos provenientes de Revit.

	varDec [0], redondeo para variables de tipo *Real*. *varExp* [""], expresión de cálculo. *varCond* [""], condición para el cálculo de la expresión.
Retorno	Devuelve el *guid* de la variable creada o actualizada. En caso de error devuelve un *guid* vacío.
Ejemplo	Ver ejemplo en asignación de valores.

Función	**Delete**
Sintaxis	Function Delete(obra As String) As Long
Descripción	Borra una obra
Parámetros	obra es la ruta completa de la obra a borrar
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que borra la obra "C:\TEMP\EJEMPLO.PRESTOOBRA" Public Sub Borrar_Obra () Dim po as Object Set po = CreateObject ("Presto.App.18") po.Delete("C:\TEMP\EJEMPLO.PrestoObra") End Sub

Función	**DeleteRecord**
Sintaxis	Function DeleteRecord(tabla As String, [todo As Long]) As Long
Descripción	Borra un registro y mantiene la integridad relacional
Parámetros	tabla es el nombre de la tabla según informes todo debe valer 1 si queremos borrar todos los registros relacionados
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que elimina todos los registros cuya naturaleza sea "Maquinaria"
	
	Public Sub Borrar_Registro ()
	Dim po As Object, arrElementos() As String, i As Integer
	Set po = CreateObject("Presto.App.18")
	po.Open "K:\Temp\CENZANO Presupuesto y control de costes.PrestoObra"
	po.SetElement 1, "Conceptos", , , "Conceptos.Nat == 7"
	s = ""
	While po.GetElement(1) = 0
	s = s + po.GetFieldStr("Conceptos.Código") + ";"
	Wend
	arrElementos = Split(s, ";")
	For i = LBound(arrElementos) To UBound(arrElementos)
	If po.FindEqual("Conceptos", "Conceptos.Código", Trim(arrElementos(i))) = 0 Then
	po.DeleteRecord ("Conceptos")
	End If
	Next
	End Sub

Función	**Duplicate**
Sintaxis	Function Duplicate(codigo As String, nuevo As String) As Long
Descripción	Duplica un registro de la tabla de conceptos de la misma forma que la función duplicar de Presto, es decir, copiando la descomposición y demás informaciones asociadas. Para copiar simplemente un registro de esta o de otra tabla, basta con leerlo, cambiar la clave e insertarlo de nuevo
Parámetros	codigo es el código del registro origen de la duplicación nuevo es el código del registro destino de la duplicación
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que duplica el registro con código "C01" dando lugar a otro registro exactamente igual con código "C01B" Public Sub Duplicar () Dim po as Object Set po = CreateObject ("Presto.App.18") **po.Duplicate** "C01", "C01B" End Sub

Función	**EndRedo**
Sintaxis	Sub EndRedo()
Descripción	Fin de acciones que se pueden deshacer
Parámetros	Ninguno
Retorno	Ninguno

Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y activa la opción de deshacer. Se le indica al usuario si se activó correctamente, y si es así se desactiva.
	
	Public Sub Finaliza_Deshacer()
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If po.BeginRedo = 0 then
	MsgBox "Deshacer iniciado correctamente"
	po.EndRedo
	MsgBox "Deshacer finalizado"
	Else
	MsgBox "No se inició correctamente deshacer"
	EndIf
	End Sub

Función	**EvalNum**
Sintaxis	Function EvalNum(expresion As String) As Double
Descripción	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor numérico
Parámetros	Expresión es la expresión que evaluará el generador de expresiones de Presto
Retorno	Numérico resultante de la evaluación de la expresión
Ejemplo	En este ejemplo se muestra un procedimiento que devuelve el valor de la variable numérica que se le pasa como parámetro Public Sub Declara_Numérico(variable as String) Dim po as Object 'variable = "DecCantRend" Set po = CreateObject ("Presto.App.18") MsgBox(**po.EvalNum** ("workd(" & Chr(34) & variable & Chr(34) & ")")) End Sub

Función	**EvalStr**
Sintaxis	Function EvalStr(expresion As String) As String
Descripción	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor de cadena
Parámetros	Expresión es la expresión que evaluará el generador de expresiones de Presto
Retorno	Cadena resultante de la evaluación de la expresión
Ejemplo	En este ejemplo se muestra un procedimiento que declara una variable que recibe como parámetro Public Sub Declara_Cadena(variable as String) Dim po as Object 'variable = "DivISO[1]" Set po = CreateObject ("Presto.App.18") MsgBox(**po.EvalStr** ("work(" & Chr(34) & variable & Chr(34) & ")")) End Sub

Función	**Exit**
Sintaxis	Sub Exit()
Descripción	Cierra una instancia de Presto (sólo la ventana principal a la que nos referimos, no toda la aplicación)
Parámetros	Ninguno
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que cierra una de las dos instancias Presto que se crean. Public Sub Cierra_Instancia() Dim po As Object Dim OtroPresto As Object Set po = CreateObject ("Presto.App.18") Set OtroPresto = CreateObject ("Presto.App.18") MsgBox "Se han abierto dos instancias de Presto" **po.Exit** MsgBox "Se ha cerrado una instancia Presto" End Sub

Función	**ExportFIEBDC**
Sintaxis	Function ExportFIEBDC(archivo As String) As Long
Descripción	Exporta la obra en formato FIEBDC
Parámetros	archivo es el nombre del archivo de destino con path y con extensión.
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que abre una obra recibida como parámetro y la exporta a FIEBDC en la ruta "C:\TEMP\EJEMPLO.BC3" Public Sub Exporta_FIEBDC (obra As String) Dim po as Object Set po = CreateObject ("Presto.App.18") po.Open obra **po.ExportFIEBDC** "C:\TEMP\EJEMPLO.BC3" End Sub

Función	**ExportRPT**
Sintaxis	Function ExportRPT(informe As String, archivo As String, parametros As Long, formato As Long, rellenar_blancos As Long, imprimir_cabecera As Long, separador_campos As Long, imprimir As Long, pagina_inicial As Long, pagina_final As Long, columnas As Long, lineas As Long, anadir As Long) As Long
Descripción	Exporta un informe
Parámetros	informe es el nombre del archivo que contiene el informe con path y con extensión archivo es el nombre del archivo destino con path y con extensión parametros valdrá 0 para tomar los parámetros por defecto y 1 para preguntarlos formato valdrá 0 para formato ASCII, 1 para formato RTF y 2 para formato PDF rellenar_blancos valdrá 0 para no rellenarlos y 1 para rellenarlos imprimir_cabecera valdrá: 0, nunca; 1, primera página y 2, siempre separador_campos valdrá: 0, ninguno; 1, espacio en blanco; 2, tabulador; 3, coma; 4, punto y coma imprimir valdrá 0 para imprimir todo el documento y 1 para imprimir un rango de páginas pagina_inicial contendrá el número de página de comienzo de la exportación pagina_final contendrá el número de página de fin de la exportación columnas indica el número de caracteres de cada línea. En formato PDF es la anchura (mm) lineas indica el número de líneas por hoja. En formato PDF es la altura (mm) añadir valdrá 0 para añadir el informe al contenido actual del archivo destino y 1 para sobrescribirlo. En formato PDF es el tamaño de página: 0, Sin definir; 1, A0; 2, A1; 3, A2; 4; A3; 5, A4; 6, Todo el documento; 7, Usuario
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se muestra la línea necesaria para exportar el informe contenido en "c:\informes\Presupuesto y mediciones.PrestoReport" al archivo "c:\informes\Presupuesto y mediciones.rft", tomando los parámetros por defecto, formato RTF, sin rellenar blancos, imprimiendo la cabecera para la primera página y añadiendo al contenido anterior. Sólo se exportarán las páginas de la 1 a la 5. Public Sub Exportar () Dim po as Object Set po = CreateObject ("Presto.App.18") **po.ExportRPT** "c:\informes\Presupuesto y mediciones.PrestoReport", "c:\informes\Presupuesto y mediciones.rtf", 0, 1, 0, 1, 0, 1, 1, 5, 120, 60, 0”

	End Sub

Función	**FindEqual**
Sintaxis	Function FindEqual(tabla As String, clave As String, valor) As Long
Descripción	se posiciona en una tabla dado un valor de una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes valor es el valor de la clave que se busca
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el registro con Conceptos.Código de la tabla Conceptos igual a un valor que recibe como parámetro.
	
	Public Sub posicionar_Igual(valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindEqual** ("Conceptos", "Conceptos.Código", valor) = 0 Then
	MsgBox "Situado en registro con Conceptos.Código " & valor
	End If
	End Sub

Función	**FindFirst**
Sintaxis	Function FindFirst(tabla As String, [clave As String]) As Long
Descripción	se posiciona en el primer registro de una tabla según una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el primer registro de la tabla Conceptos.
	Public Sub posicionar_Primero()
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindFirst** ("Conceptos") = 0 Then
	MsgBox "Situado en el primer registro de Conceptos"
	End If
	End Sub

Función	**FindGreat**
Sintaxis	Function FindGreat(tabla As String, clave As String, valor) As Long
Descripción	se posiciona en el registro mayor a un valor dado de una tabla según una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes valor es el valor de la clave que se busca
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el primer registro de la tabla Conceptos.
	
	Public Sub posicionar_Mayor(valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindGreat** ("Conceptos", "Conceptos.Código", valor) = 0 Then
	MsgBox "Situado en el siguiente registro a " & valor
	End If
	End Sub

Función	**FindGreatEqual**
Sintaxis	Function FindGreatEqual(tabla As String, clave As String, valor) As Long
Descripción	se posiciona en el registro mayor o igual a un valor dado de una tabla según una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes valor es el valor de la clave que se busca
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el registro mayor o igual de la tabla Conceptos.
	
	Public Sub posicionar_MayorIgual(valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindGreatEqual** ("Conceptos", "Conceptos.Código", valor) = 0 Then
	MsgBox "Situado en el siguiente registro a " & valor
	End If
	End Sub

Función	**FindLast**
Sintaxis	Function FindLast(tabla As String, [clave As String]) As Long
Descripción	se posiciona en el último registro de una tabla según una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el último registro de la tabla Conceptos.
	
	Public Sub posicionar_Ultimo()
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindLast** ("Conceptos") = 0 Then
	MsgBox "Situado en el último registro de Conceptos"
	End If
	End Sub

Función	**FindLessEqual**
Sintaxis	Function FindLessEqual(tabla As String, clave As String, valor) As Long
Descripción	se posiciona en el registro menor o igual a un valor dado de una tabla según una clave
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el registro menor o igual de la tabla Conceptos.
	
	Public Sub posicionar_MenorIgual(valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po. FindLessEqual** ("Conceptos", "Conceptos.Código", valor) = 0 Then
	MsgBox "Situado en el anterior registro a " & valor
	End If
	End Sub

Función	**FindNext**
Sintaxis	Function FindNext(tabla As String, [clave As String]) As Long
Descripción	se posiciona en el registro siguiente de una tabla
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el registro con Conceptos.Código de la tabla Conceptos siguiente al que se encuentra.
	
	Public Sub posicionar_Siguiente (valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindNext** ("Conceptos", "Conceptos.Código") = 0 Then
	MsgBox "Situado en registro con Conceptos.Código siguiente"
	End If
	End Sub

Función	**Findpos**
Sintaxis	Function Findpos(tabla As String, clave As String, pos As Long) As Long
Descripción	Se posiciona en un registro de una tabla ordenada por una determinada clave, conociendo su posición
Parámetros	tabla es el nombre de la tabla según informes clave es la clave de ordenación de la tabla a la cual se refiere la posición pos es la posición del registro en el que nos queremos colocar
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que primeramente obtiene la posición del primer registro de tipo "Partida". posteriormente continua recorriendo la tabla "Conceptos" y finalmente coloca como registro activo el primer registro de tipo "Partida"
	
	Public Sub posicionar_Registro()
	Dim po as Object
	Dim Num_Registro as Long
	Set po = CreateObject ("Presto.App.18")
	po.SetElement 1, "Conceptos",,, "Conceptos.Nat == 5 "
	If po.GetElement (1) = 0 Then
	Num_Registro = po.Getpos ("Conceptos")
	End If
	po.SetElement 1, "Conceptos"
	While po.GetElement(1) = 0
	' Avanzamos de registro
	Wend
	po.Findpos "Conceptos", "Conceptos.Código", Num_Registro
	End Sub

Función	**FindPrev**
Sintaxis	Function FindPrev(tabla As String, [clave As String]) As Long
Descripción	se posiciona en el registro anterior de una tabla
Parámetros	tabla es el nombre de la tabla según informes clave es clave según informes
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que nos posiciona en el registro con Conceptos.Código de la tabla Conceptos anterior al que se encuentra.
	
	Public Sub posicionar_Siguiente (valor As String)
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If **po.FindPrev**("Conceptos", " Conceptos.Código") = 0 Then
	MsgBox "Situado en registro con Conceptos.Código anterior"
	End If
	End Sub

Función	**GetElement**
Sintaxis	Function GetElement(elemento As Long, [anida As Long]) As Long
Descripción	Se posiciona en el primer registro de una tabla
Parámetros	elemento es el número de elemento asociado al SetElement con el que abrimos la tabla anida indica si queremos que se posicione en los registros relacionados al igual que elemento del módulo de Informes
Retorno	Recupera un elemento de una tabla abierta con SetElement y devuelve distinto de cero cuando ya no quedan Elemento es el número indicado por el SetElement con el que se abrió la tabla
Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y abre la tabla de "Conceptos", obteniendo solamente los registros de tipo "Capítulo". posteriormente recorremos la tabla con la función GetElement y contamos el número de registros obtenido. Public Sub Obtener_Elementos() Dim po as Object Dim Num_Registros as Long Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos",,, "Conceptos.Nat == 4" Num_Registros = 0 While **po.GetElement** (1) = 0 Num_Registros = Num_Registros + 1 Wend MsgBox "Número de registros de tipo capítulo: " & CStr(Num_Registros) End Sub

Función	**GetField**
Sintaxis	Function GetField(campo As String, [dimension As Long])
Descripción	Devuelve el valor de un campo de Presto
Parámetros	campo es el nombre del campo que se quiere obtener dimension es la dimensión del campo solicitado. por defecto la 0
Retorno	Devuelve el valor del campo indicado como parámetro
Ejemplo	En este ejemplo se muestra un procedimiento que muestra al usuario todos los Conceptos.Pres Public Sub Leer_Pres () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 MsgBox CStr (**po.GetField** ("Conceptos.Pres")) Wend End Sub

Función	**GetFieldBinary**
Sintaxis	Function GetFieldBinary(campo As String)
Descripción	Devuelve el valor de un campo binario de Presto
Parámetros	campo es el nombre del campo que se quiere obtener
Retorno	Devuelve el valor del campo indicado como parámetro
Ejemplo	En este ejemplo se muestra un procedimiento que recorre, devuelve y copia en un directorio los archivos de una obra. Public Sub Leer_Archivos () Dim ArchivoByte() As Byte = Nothing Dim bFileStream As FileStream Dim ArchivosNombre As String Dim ArchivosExtensión As String Dim Ruta As String = "C:\Temp\" Set po = CreateObject ("Presto.App.18") po.SetElement(1, "Archivos") While po.GetElement(1) = 0 ArchivoByte= **po.GetFieldBinary**("Archivos.Archivo") ArchivosNombre = Trim(po.GetFieldStr("Archivos.Nombre")) ArchivosExtensión = Trim(po.GetFieldStr("Archivos.Extensión")) bFileStream = New FileStream(Ruta & ArchivosNombre & "." & ArchivosExtensión, FileMode.CreateNew, FileAccess.Write) bFileStream.Write(ArchivoByte, 0, ArchivoByte.Length) bFileStream.Close() Wend End Sub

Función	**GetFieldNum**
Sintaxis	Function GetFieldNum(campo As String, [dimension As Long]) As Double
Descripción	Devuelve el valor de un campo numérico
Parámetros	campo es el nombre del campo que se quiere obtener dimension es la dimensión del campo solicitado. por defecto la 0
Retorno	Devuelve el valor del campo indicado como parámetro
Ejemplo	En este ejemplo se muestra un procedimiento que muestra al usuario todos los Conceptos.Pres Public Sub Leer_Num () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 MsgBox CStr(**po.GetFieldNum** ("Conceptos.Pres")) Wend End Sub

Función	**GetFieldStr**
Sintaxis	Function GetFieldStr(campo As String, [dimension As Long]) As String
Descripción	Devuelve el valor de un campo de tipo cadena
Parámetros	campo es el nombre del campo que se quiere obtener dimension es la dimensión del campo solicitado. por defecto la 0
Retorno	Devuelve el valor del campo indicado como parámetro
Ejemplo	En este ejemplo se muestra un procedimiento que muestra al usuario todos los códigos de Conceptos Public Sub Leer_Str () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 MsgBox **po.GetFieldStr** ("Conceptos.Código") Wend End Sub

Función	**Getpos**
Sintaxis	Function Getpos(tabla As String) As Long
Descripción	Obtiene la posición de un registro. La posición es un número asociado por el gestor de archivos a cada registro que sirve posteriormente si es necesario posicionarse de nuevo en el mismo registro
Parámetros	tabla es el nombre de la tabla según informes
Retorno	posición del registro activo en la tabla indicada

Ejemplo	En este ejemplo se muestra un procedimiento que indica al usuario la posición del primer registro de tipo capítulo.
	
	Public Sub posicion_Registro()
	Dim po as Object
	Dim Num_Registro as Long
	Set po = CreateObject ("Presto.App.18")
	po.SetElement 1, "Conceptos",,, "Conceptos.Nat == 4 "
	If po.GetElement (1) = 0 Then
	Num_Registro = **po.Getpos** ("Conceptos")
	MsgBox "posición del primer registro de tipo capítulo: " & CStr(Num_Registro)
	End If
	End Sub

Función	**GetSelection**
Sintaxis	Function GetSelection(first As Long, tabla As String) As Long
Descripción	Recorre los elementos seleccionados de una tabla
Parámetros	first indica si queremos obtener el primer elemento seleccionado o recorrerlos. Si first = 1 nos colocamos en el primero, si first = 0 los recorremos tabla indica el esquema cuyos elementos seleccionados queremos recorrer
Retorno	Recupera un elemento seleccionado de una tabla indicada y devuelve distinto de cero cuando ya no quedan
Ejemplo	En este ejemplo se muestra un procedimiento que recorre tos los elementos seleccionados de la tabla Conceptos y muestra su Código y Resumen. Suponemos que se tiene creada una instancia de Presto y se han marcado los elementos deseados. Public Sub Ver_Seleccion() Dim first As Long first = 1 While **po.GetSelection** (first, "Conceptos") = 0 first = 0 MsgBox ("Código: " & po.GetFieldStr ("Conceptos.Código") & vbCrLf & "Resumen: " & po.GetFieldStr ("Conceptos.Resumen")) Wend End Sub

Función	**GetText**
Sintaxis	Function GetText(campo As String, [tipo As Long = 1]) As String
Descripción	Devuelve el valor del texto con los mismos parámetros que arriba.
Parámetros	campo indica el campo del que queremos obtener el texto (Conceptos.Texto, Facturas.Texto, etc) tipo es 1 si valor es un texto ASCII y 2 si valor está en rtf
Retorno	Valor del campo texto
Ejemplo	En este ejemplo se muestra un procedimiento que obtiene la variable Conceptos.Texto de todos los registros. Public Sub Leer_Memo () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 MsgBox **po.GetText** ("Conceptos.Texto") Wend End Sub

Función	**ImportCad**
Sintaxis	Function ImportCad(proyecto As String, archivo As String, tipo As Long) As Long
Descripción	Importa un archivo de cad (AllPlan, ArchiCad, FIEBDC) sobre una obra de Presto
Parámetros	proyecto contiene la ruta de la obra sobre la que se realizará la importación archivo es la ruta completa del archivo a importar tipo no se usa
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que importa un archivo de CAD recibido como parámetro sobre una obra Public Sub Importa_CAD (obra As String, archivo As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.ImportCad** obra, archivo, 0 End Sub

Función	**ImportFIEBDC**
Sintaxis	Function ImportFIEBDC(archivo As String) As Long
Descripción	Importa la obra en formato FIEBDC
Parámetros	archivo es el nombre del archivo de origen con path y con extensión
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que importa una obra recibida como parámetro en formato FIEBDC Public Sub Importa_FIEBDC (archivo As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.ImportFIEBDC** archivo End Sub

Función	**InfoCode**
Sintaxis	Function InfoCode(proyecto As String, codigo As String, unidad As String, resumen As String, precio As String) As Long
Descripción	Información de un concepto de Presto
Parámetros	proyecto contiene la ruta de la obra sobre la que se buscará el concepto codigo es el código del concepto a buscar unidad, resumen y precio devuelven la información correspondiente si se encuentra
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se obtiene la informacion de un concepto recibido como parámetro. Public Sub Get_Concepto (obra As String ,codigo As String) Dim unidad, resumen, precio As String Dim po as Object Set po = CreateObject ("Presto.App.18") **po.InfoCode** obra, codigo, unidad, resumen, precio End Sub

Función	**InitRecord**
Sintaxis	Function InitRecord(tabla As String) As Long
Descripción	pone a cero un registro del tipo indicado en la tabla. Este registro se puede llenar luego con valores y grabar con las demás funciones
Parámetros	tabla es el nombre de la tabla según informes
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se pone a cero un registro de la tabla recibida como parámetro, posteriormente este registro se podrá rellenar con los valores adecuados Public Sub Inicializa_Registro(archivo As String) **po.InitRecord** archivo End Sub

Función	**InitVar**
Sintaxis	Sub InitVar()
Descripción	Suprime variables declaradas con float o con char dentro de una expresión usada en la función EvalStr o EvalNum
Parámetros	Ninguno
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que suprime las variables declaradas con anterioridad Public Sub Inicializa_Variables() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.InitVar** End Sub

Función	**InsertAttachment**
Sintaxis	Sub InsertAttachment(tableName As String, code As String, attachDescription As String, attachExtension As String, attachBuffer As Variant, attachType As Long, resetSourcePath As Bool)
Descripción	Inserta un archivo en Presto a partir de un buffer y lo asocia con un registro de una tabla por medio del código
Parámetros	Entre corchetes se indica el valor por defecto *tableName,* nombre de la tabla a la que pertenece el código al cual queremos asociar el archivo. "Conceptos" "Pedidos" "Entregas" "Facturas" *code,* código que identificar el registro al cual queremos asociar el archivo. *attachDescription,* descripción del archivo asociado. *attachExtension,* extensión para el archivo asociado. *attachBuffer,* buffer con el contenido del archivo asociado. *attachType* [0], tipo del archivo asociado 0: Indeterminado 1: RTF Especificación 2: RTF Características técnicas 3: RTF Condiciones previas 4: RTF Ejecución 5: RTF Medición 6: RTF Control 7: RTF Seguridad y salud 8: RTF Gestión ambiental 9: RTF Normas de aplicación 10: RTF Mantenimiento 11: RTF Varios 19: RTF Texto sin faceta 20: Paramétrico 30: Gráfico 31: Modelo o familia Revit. *resetSourcePath* [0], indica con valor distinto de 0 que tras insertar el archivo asociado hay que borrar el campo *Archivos.Camino* de Presto e inicializar *Archivos.Fecha* con la fecha actual. Es útil cuando el archivo a insertar está en disco de forma temporal
Retorno	Devuelve el *guid* del archivo asociado creado o cadena vacía en caso de error
Ejemplo	En este ejemplo se muestra un procedimiento que inserta un archivo asociado a un código determinado Public Sub Inserta_Asociado(codigo As String, buffer) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.InsertAttachment** "Conceptos", codigo, "Prueba", "TXT", buffer End Sub

Función	**InsertAttachmentFromPath**
Sintaxis	Sub InsertAttachment(tableName As String, code As String, attachFilePath As String, resetSourcePath As Bool)
Descripción	Inserta un archivo en Presto a partir de un buffer y lo asocia con un registro de una tabla por medio del código
Parámetros	(entre corchetes se indica el valor por defecto) *tableName,* nombre de la tabla a la que pertenece el código al cual queremos asociar el archivo. "Conceptos" "Pedidos" "Entregas" "Facturas" *code,* código que identificar el registro al cual queremos asociar el archivo. *attachFilePath, path* del archivo. *resetSourcePath* [0], indica con valor distinto de 0 que tras insertar el archivo asociado hay que borrar el campo *Archivos.Camino* de Presto e inicializar *Archivos.Fecha* con la fecha actual. Es útil cuando el archivo a insertar está en disco de forma temporal
Retorno	Devuelve el *guid* del archivo asociado creado o cadena vacía en caso de error
Ejemplo	En este ejemplo se muestra un procedimiento que inserta un archivo asociado a un código determinado Public Sub Inserta_Asociado(codigo As String, ruta As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.InsertAttachmentFromPath** "Conceptos", codigo, ruta End Sub

Función	**InsertRecord**
Sintaxis	Function InsertRecord(tabla As String) As Long
Descripción	Inserta en la tabla un registro con la clave que tenga en ese momento. Si la clave ya existe y la tabla no admite duplicados devolverá un error
Parámetros	tabla es el nombre de la tabla según informes
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se inserta un registro en la tabla que se recibe como parámetro Public Sub Inserta_Registro(archivo As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.InsertRecord** archivo End Sub

Función	**LogBegin**
Sintaxis	Function LogBegin(logHead As String)
Descripción	Inicia un proceso de log en Presto
Parámetros	logHead es el texto del mensaje que aparecerá en la primera línea del log junto a la hora y la fecha
Retorno	0: correcto (se pudo iniciar correctamente el proceso de log) 1: error
Ejemplo	Ver ejemplo en función LogMsg

Función	**LogEnd**
Sintaxis	Function LogEnd()
Descripción	Cierra el proceso de log en Presto y muestra la ventana con todos los mensajes agregados
Parámetros	Ninguno
Retorno	0: correcto (se pudo cerrar correctamente el proceso de log) 1: error
Ejemplo	Ver ejemplo en función LogMsg

Función	LogMsg
Sintaxis	Function LogMsg(logMsg As String, mode As Long)
Descripción	Agrega un mensaje al sistema de log
Parámetros	logMsg es el texto del mensaje mode es el tipo de mensaje, 0: información 1: advertencia 2: error
Retorno	0: correcto Otro valor: error

Ejemplo	Set po = GetObject("", "Presto.App.18")
	Dim nMensajes
	Dim strMensaje
	If **po.LogBegin**("Comienzo del LOG") = 0 Then
	nMensajes = 1
	While nMensajes < 10
	strMensaje = "Mensaje de información nº " & nMensajes
	po.LogMsg strMensaje, 0
	nMensajes = nMensajes + 1
	Wend
	po.LogEnd
	End If

Función	New
Sintaxis	Function New(obra As String) As Long
Descripción	Crea y abre una obra de Presto
Parámetros	obra es la ruta de la obra a crear
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y crea una nueva obra en el path que se recibe como parámetro. Finalmente se indica al usuario si se creó correctamente.
	Agregue la referencia de entre los tipos COM, "Presto 10.0 Type Library" en su compilador.
	
	Public Sub Nueva_Obra(*path As String*)
	Dim po As PrestoLib.PrestoApplication
	Set po = CreateObject ("Presto.App.18")
	If **po.New**(path)= 0 Then
	MsgBox "Obra creada correctamente"
	Else
	MsgBox "No se creó correctamente la obra"
	EndIf
	End Sub

Función	Open
Sintaxis	Function Open(obra As String, [solo_lectura As Long]) As Long
Descripción	Abre una obra de Presto
Parámetros	obra es la ruta de la obra a abrir solo lectura = 2, convierte la obra a formato .Presto / solo_lectura = 1, abre la obra en modo solo lectura / solo_lectura = 0, abre la obra en modo escritura
Retorno	0: correcto Otro valor: error

Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y abre una obra situada en el path que se recibe como parámetro en modo "solo lectura". Finalmente se indica al usuario si se abrió correctamente.
	
	Public Sub Abrir_Obra (*path As String*)
	Dim po As Object
	Set po = CreateObject ("Presto.App.18")
	If **po.Open**(path,1)= 0 Then
	MsgBox "Obra abierta correctamente"
	Else
	MsgBox "No se abrió correctamente la obra"
	EndIf
	End Sub

Función	**PrintRPT**
Sintaxis	Function PrintRPT(informe As String, [parametros As Long], [copias As Long], [impresora As Long]) As Long
Descripción	Imprime un informe
Parámetros	informe es el nombre del archivo que contiene el informe con path y con extensión parametros valdrá 0 para tomar los parámetros por defecto y 1 para preguntarlos copias es el número de copias que se desea imprimir impresora valdrá 0 para tomar la impresora por defecto y 1 para pedirla
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se muestra la línea necesaria para imprimir el informe que se encuentra en la ruta "c:\informes\ejemplo.rpt", tomando los parámetros por defecto, preguntando la impresora y realizando 3 copias Public Sub Informe () Dim po as Object Set po = CreateObject ("Presto.App.18") **po.PrintRPT** "C:\informes\ejemplo.rpt", 0, 3, 1 End Sub

Atributo	**Project**
Sintaxis	Property Project As String
Descripción	Devuelve el nombre de path de la obra de presto abierta
Parámetros	---
Retorno	---
Ejemplo	En este ejemplo se muestra un procedimiento que visualiza el valor de Project Public Sub Ver_Project() Dim po as Object Set po = CreateObject ("Presto.App.18") MsgBox **po.Project** End Sub

Función	**Quit**
Sintaxis	Sub Quit()
Descripción	Cierra la aplicación de Presto
Parámetros	Ninguno
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que cierra la aplicación Presto. Public Sub Cierra_Presto() Dim po As Object Set po = CreateObject ("Presto.App.18") MsgBox "Se ha abierto Presto" po.Quit MsgBox "Se ha cerrado Presto" End Sub

Atributo	**ReadOnly**
Sintaxis	Property ReadOnly As Long
Descripción	Devuelve si la obra está en modo "solo lectura"
Parámetros	---
Retorno	---

Ejemplo	En este ejemplo se muestra un procedimiento que visualiza si la obra está en "sólo lectura"
	
	Public Sub Ver_SoloLectura()
	Dim po as Object
	Set po = CreateObject ("Presto.App.18")
	If (**po.ReadOnly**)
	MsgBox "Solo lectura"
	
	Else
	MsgBox "No solo lectura"
	EndIf
	End Sub

Función	**Recalculate**
Sintaxis	Sub Recalculate(activar As Long)
Descripción	Bloquea o Desbloquea los recálculos Si está bloqueada no se hacen recálculos intermedios, a fin de evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamar a la función de nuevo para hacerlos todos de golpe.
Parámetros	0: Desactiva 1: Activa de nuevo y recalcula
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que bloquea los recálculos de Presto Public Sub Bloquea_Calculos() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.Recalculate** (0) End Sub

Función	Rename
Sintaxis	Function Rename(campo As String, valor As String) As Long
Descripción	Cambia (renombra) la clave de un concepto, estando posicionados en el concepto que se quiere renombrar.
Parámetros	campo es el campo clave que queremos renombrar “Conceptos.Código” valor es el nuevo valor que le vamos a dar al campo clave
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que renombra todos los registros añadiéndoles una X al final de su código Public Sub Renombrar () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 **po.Rename** " Conceptos.Código ", trim(po.GetFieldStr ("Conceptos.Código")) & "X" Wend End Sub

Función	**SelCode**
Sintaxis	Function SelCode(HWND As Long, proyecto As String, relacion As String) As Long
Descripción	Elige un concepto de Presto
Parámetros	HWND es la ventana que se tomará como padre del diálogo a mostrar proyecto contiene la ruta de la obra sobre la que se buscará el concepto relacion indica la relación donde se situará el diálogo
Retorno	Devuelve 0: Código en blanco 1: Código seleccionado -1: Cancelar Otros: error
Ejemplo	En este ejemplo se obtiene la información de un concepto recibido como parámetro. Public Sub Sel_Code (obra As String ,relacion As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SelCode** 0, obra, relacion End Sub

Función	**SetCharacter**
Sintaxis	Sub SetCharacter(modo As Long)
Descripción	Selecciona el juego ANSI (Windows) o OEM (MS-DOS). Se usa antes de leer un texto de un registro o de evaluar una expresión
Parámetros	modo puede valer 0 ó 1 y nos indicará el juego de caracteres a utilizar
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que pone como juego de caracteres el juego ANSI Public Sub Caracter_ANSI() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SetCharacter** (0) End Sub

Función	**SetConceptVarValue**
Sintaxis	Sub SetConceptVarValue(varGuid As String, conceptCode As String, varValue As Variant, valType As Integer, valUserTxt As String, valUserNum As Double, revitparamElementId As String)
Descripción	Asigna valor a la variable identificada por su *guid* en el concepto con el código indicado. Tanto el concepto como la variable deben existir antes de asignar valor. El valor debe ser de un tipo compatible con el tipo de la variable. Además del valor se actualizan los campos, "ElementID", "Nota" y "UsrNum".
Parámetros	Entre corchetes se indica el valor por defecto *varGuid*, identificador único global de la variable. *conceptCode*, código del concepto donde recibe valor la variable. *varValue*, valor que se asociará a la variable. *valType* [0], tipo del valor (nada que ver con el tipo de la variable). 0: Normal 1: Calculado por Presto 2: Término introducido por el usuario (manual) 3: Término introducido por Presto (automático). *valUserTxt* [""], texto de usuario (nota) asociado al valor. *valUserNum* [0], valor numérico de usuario asociado al valor. *revitparamElementId* [""], dato proveniente de Revit.
Retorno	0 si pudo asignar correctamente el valor a la variable. Distinto de 0 en caso contrario
Ejemplo	ver ejemplo en *SetTakeoffVarValue*

Función	**SetCurrentInvPeriod**
Sintaxis	Function SetCertificacion(fecha as String) As Long
Descripción	Establece como certificación actual aprobada la fecha recibida.
Parámetros	Fecha de la certificación en formato yyyymmdd. Tiene que existir previamente.
Retorno	0: correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto abre una obra y pone como fecha de certificación actual el 30 de Abril de 2009, finalmente recalcula la obra. Public Sub ponCertificacion() Dim po As New PrestoLib.PrestoApplication Dim ret As New Integer po.Open("C:\Temp\Presupuesto y mediciones.PrestoObra") ret = po. SetCurrentInvPeriod ("20090430")

```
        If (ret = 0) Then
            po.Recalculate(1)
        End If
    End Sub
```

Función	**SetElement**
Sintaxis	Function SetElement(elemento As Long, tabla As String, [clave As String], [mascara As String], [seleccion As String], [ordenar_por As String], [orden As Long]) As Long
Descripción	Define un elemento. Un elemento es una forma de acceder a una tabla de una obra de Presto, similar a la utilizada en informes. Se accede a un elemento concreto con la función GetElement. Elemento indica un número entre 1 y 25, ya que se pueden definir varios SetElement al tiempo.
Parámetros	Los siguientes parámetros funcionan igual que los campos de igual nombre de la caja de diálogo de propiedades un elemento de informes. tabla: nombre de la tabla a la que se accede según figura en el campo "Tabla" de la caja de diálogo de propiedades de un elemento. clave establece el orden de lectura y se elige entre las posibilidades que aparecen en el campo "Clave" de la caja de diálogo anterior. máscara es una expresión de selección que se aplica al campo clave seleccion es una expresión que se calcula para filtrar elementos de una tabla ordenar_por es una expresión que sirve para ordenar la tabla orden indica 0 ascendente, 1 descendente
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que arranca una instancia de Presto y abre la tabla de "Conceptos", obteniendo solamente los registros de tipo "Capítulo". Public Sub Abrir_Conceptos() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SetElement** 1, "Conceptos",,, "Conceptos.Nat == 4" End Sub

Función	**SetField**
Sintaxis	Sub SetField(campo As String, valor, [dimension As Long])
Descripción	Asigna un valor a un campo (en la tabla solo se graba con InsertRecord o UpdateRecord)
Parámetros	campo es el nombre del campo valor es el contenido dimension es la dimensión del campo
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que inicializa el valor del Resumen con el valor de Conceptos.Código correspondiente al registro Public Sub poner_Str () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 **po. SetField** "Conceptos.Resumen", po.GetField("Conceptos.Código") po.UpdateRecord "Conceptos" Wend End Sub

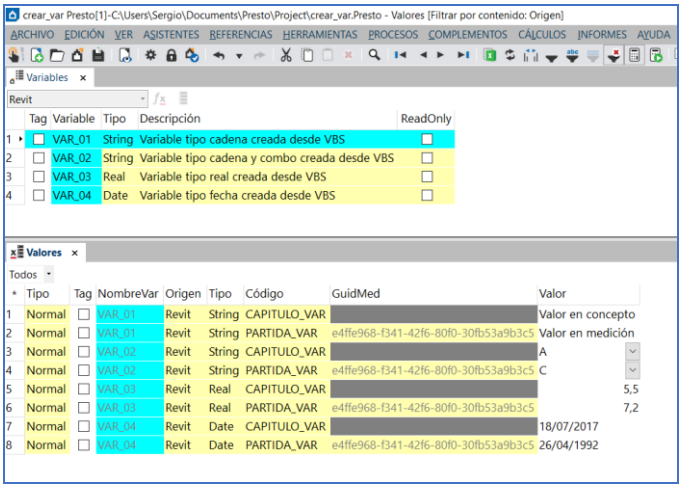
Función	**SetFieldBinary**
Sintaxis	Sub SetFieldBinary(campo As String, valor)
Descripción	Asigna un valor a un campo binario de Presto
Parámetros	campo es el nombre del campo valor es el contenido
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que modifica un archivo Public Sub poner_Archivo (valor) Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, " Archivos " While po.GetElement (1) = 0 **po. SetFieldBinary** "Archivos.Archivo", valor po.UpdateRecord " Archivos " Wend End Sub

Función	**SetModal**
Sintaxis	Sub SetModal(activar As Long)
Descripción	pone y quita el modo modal de Presto (mientras está puesto, la ventana de Presto está bloqueada): Bloquea o Desbloquea el uso de Presto
Parámetros	0: desactiva 1: activa
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que bloquea la ventana de Presto Public Sub Bloquea_Presto() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SetModal** (0) End Sub

Función	**SetTakeoffVarValue**
Sintaxis	Sub SetTakeoffVarValue(varGuid As String, takeoffGuidElement As String, takeoffGuidAux As String, varValue As Variant, valType As Integer, valUserTxt As String, valUserNum As Double, revitparamElementId As String)
Descripción	Asigna valor a la variable identificada por su *guid* en la línea de medición identificada por su *guidElement* y su *guidAux*. Tanto la línea de medición como la variable deben existir antes de asignar valor. El valor debe ser de un tipo compatible con el tipo de la variable. Además del valor se actualizan los campos, "ElementID", "Nota" y "UsrNum".
Parámetros	Entre corchetes se indica el valor por defecto) *varGuid*, identificador único global de la variable. *takeoffGuidElement*, *takeoffGuidAux*, identificador único global de la línea de medición obtenido como concatenación de ambos *guids*. *varValue*, valor que se asociará a la variable. *valType* [0], tipo del valor (nada que ver con el tipo de la variable). 0: Normal 1: Calculado por Presto 2: Término introducido por el usuario (manual) 3: Término introducido por Presto (automático). *valUserTxt* [""], texto de usuario (nota) asociado al valor. *valUserNum* [0], valor numérico de usuario asociado al valor. *revitparamElementId* [""], dato proveniente de Revit.
Retorno	0 si pudo asignar correctamente el valor a la variable. Distinto de 0 en caso contrario.
Ejemplo	Set po = GetObject("", "Presto.App.18") po.SetUpdateScreen(0) po.BeginRedo() ' Relaciones y conceptos po.InitRecord("Conceptos") po.SetField("Conceptos.Código", "CAPITULO_VAR") po.SetField("Conceptos.Nat", 4) po.InsertRecord("Conceptos") po.InitRecord("Relaciones") po.SetField("Relaciones.CodSup", "0") po.SetField("Relaciones.CodInf", "CAPITULO_VAR") po.InsertRecord("Relaciones") po.InitRecord("Conceptos") po.SetField("Conceptos.Código", "PARTIDA_VAR") po.SetField("Conceptos.Nat", 5)

```
po.InsertRecord("Conceptos")

po.InitRecord("Relaciones")
po.SetField("Relaciones.CodSup", "CAPITULO_VAR")
po.SetField("Relaciones.CodInf", "PARTIDA_VAR")
po.InsertRecord("Relaciones")

' Medicion

Dim guid_elem
Dim guid_aux

po.InitRecord("Mediciones")
po.SetField("Mediciones.CodSup", "CAPITULO_VAR")
po.SetField("Mediciones.CodInf", "PARTIDA_VAR")
po.SetField("Mediciones.Cantidad", 100)
po.InsertRecord("Mediciones")

guid_elem = po.GetFieldStr("Mediciones.GuidElem", 0)
guid_aux = po.GetFieldStr("Mediciones.GuidAux", 0)

' Variables y valores

Dim guid_var

' Variable de tipo cadena

guid_var = po.CreateVariable("GUID_VAR_01", "VAR_01", "Variable tipo String
creada desde VB", 8, 6)

po.SetConceptVarValue(guid_var, "PARTIDA_VAR", "Valor en concepto")
po.SetTakeoffVarValue(guid_var, guid_elem, guid_aux, "Valor en medición")

' Variable de tipo cadena y combo

guid_var = po.CreateVariable("GUID_VAR_02", "VAR_02", "Variable tipo String
y combo creada desde VB", 8, 6, , , , , 1, "A|B|C|D")

po.SetConceptVarValue(guid_var, "PARTIDA_VAR", "A")
po.SetTakeoffVarValue(guid_var, guid_elem, guid_aux, "C")
```

```
' Variable de tipo real (con mínimo, máximo, redondeo a 10 decimales)
guid_var = po.CreateVariable("GUID_VAR_03", "VAR_03", "Variable tipo Real creada desde VB", 8, 5, 0, 0.5, 1, 10.7, , , , , , , , , , , 10)

po.SetConceptVarValue(guid_var, "PARTIDA_VAR", 5.5)
po.SetTakeoffVarValue(guid_var, guid_elem, guid_aux, 7.2)

' Variable de tipo fecha

guid_var = po.CreateVariable("GUID_VAR_04", "VAR_04", "Variable tipo fecha creada desde VB", 8, 7)

po.SetConceptVarValue(guid_var, "PARTIDA_VAR", "20180506")
po.SetTakeoffVarValue(guid_var, guid_elem, guid_aux, 14006)

po.EndRedo()
po.SetUpdateScreen(1)

po.UpdateScreen()
```



Función	**SetText**
Sintaxis	Sub SetText(campo As String, valor As String)
Descripción	Rellena un campo de texto.
Parámetros	campo indica el campo del que queremos modificar el texto (Conceptos.Texto, Facturas.Texto, etc) valor es el contenido a insertar
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que coloca en la variable Conceptos.Texto de todos los registros el valor de Conceptos.Código Public Sub poner_Texto () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 **po.SetText** "Conceptos.Texto", po.GetFieldStr ("Conceptos.Código") Wend End Sub

Función	**SetUniqueVarValue**
Sintaxis	Sub SetUniqueVarValue(varGuid as String, varValue As Variant)
Descripción	Asigna un valor de tipo de asignación "Única" a una variable. La variable debe existir previamente.
Parámetros	*varGuid*, identificador único global de la variable. *varValue*, valor que se asociará a la variable.
Retorno	0 si pudo asignar correctamente el valor a la variable. Distinto de 0 en caso contrario.
Ejemplo	En este ejemplo se muestra un procedimiento que asigna un valor a cuatro variables comunes de distinto tipo Set po = GetObject("", "Presto.App.18") po.SetUpdateScreen(0) po.BeginRedo **po.SetUniqueVarValue** "PRESTOHEADVAR000-0020063",9.5 'Real CalcDurLab = 9.5 **po.SetUniqueVarValue** "PRESTOHEADVAR000-0020060",1 'Bool CalcNoRedCalculos = 1 (Sí) **po.SetUniqueVarValue** "PRESTOHEADVAR000-0020075",22526 'Integer Combo CalcSubAutoCampo1 = 22526 (Zona) **po.SetUniqueVarValue** "PRESTOHEADVAR000-0020086","20180228" 'Date DivFecha[1] = 28/02/2018 po.EndRedo po.SetUpdateScreen(1) po.UpdateScreen MsgBox "FIN actualiza_valor_unico !"

Función	**SetUpdateScreen**
Sintaxis	Sub SetUpdateScreen(activar As Long)
Descripción	Si está desactivada, los cambios que deberían reflejarse en las ventanas visibles no se actualizan, incluyendo su recorrido, a fin de evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamar a la función SetUpdateScreen con el parámetro 1 para reflejar todos los cambios de una sola vez.
Parámetros	0: desactiva 1: activa
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que bloquea los cambios en las ventanas de Presto Public Sub Bloquea_VentanaPresto() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SetUpdateScreen** (0) End Sub

Función	**SetVar**
Sintaxis	Sub SetVar(variable As String, valor As String)
Descripción	Modifica una variable de Presto
Parámetros	0: desactiva 1: activa
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que modifica una variable de Presto Public Sub Set_Variable(var As String, valor As String) Dim po as Object Set po = CreateObject ("Presto.App.18") **po.SetVar** (var, valor) End Sub

Atributo	**Status**
Sintaxis	Property Status As Long
Descripción	Después de una llamada a una función del gestor de archivos devuelve 0 todo correcto
Parámetros	---
Retorno	---
Ejemplo	En este ejemplo se muestra un procedimiento que visualiza el valor de Status después de realizar cualquier operación Public Sub Ver_Status() Dim po as Object Set po = CreateObject ("Presto.App.18") MsgBox CStr (**po.Status**) End Sub

Atributo	**SubVersionNum**
Sintaxis	Property SubVersionNum As Long
Descripción	Sub-versión de Presto en formato número
Parámetros	---
Retorno	---
Ejemplo	Set po = GetObject("", "Presto.App.18") Dim strMensaje strMensaje = po.VersionStr & ", " & po.VersionNum & ", " & **po.SubVersionNum** MsgBox strMensaje

Función	**UpdateRecord**
Sintaxis	Function UpdateRecord(tabla As String) As Long
Descripción	Actualiza en la tabla un registro cuyos campos han cambiado. Si el campo clave hubiera sido modificado, cambiará también en la tabla, pero sin mantener la integridad relacional de las tablas asociadas. Para cambiar los campos clave, usar la función Rename
Parámetros	tabla es el nombre de la tabla según informes
Retorno	0 correcto Otro valor: error
Ejemplo	En este ejemplo se muestra un procedimiento que modifica el Resumen de todos los registros cuya naturaleza sea "Partida" Public Sub Modifica_Registro () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos",,, "Conceptos.Nat==5" While po.GetElement(1) = 0 po.SetField "Conceptos.Resumen", "Esto es una partida" **po.UpdateRecord** "Conceptos" Wend End Sub

Función	**UpdateScreen**
Sintaxis	Sub UpdateScreen()
Descripción	Actualiza las ventanas
Parámetros	Ninguno
Retorno	Ninguno
Ejemplo	En este ejemplo se muestra un procedimiento que actualiza las ventanas de Presto Public Sub Actualiza_Ventanas() Dim po as Object Set po = CreateObject ("Presto.App.18") **po.UpdateScreen** End Sub

Atributo	**VersionNum**
Sintaxis	Property VersionNum As Long
Descripción	Versión de Presto en formato número
Parámetros	---
Retorno	---
Ejemplo	Ver ejemplo en atributo SubVersionNum

Atributo	**VersionStr**
Sintaxis	Property VersionStr As String
Descripción	Nombre completo de la versión de Presto
Parámetros	---
Retorno	---
Ejemplo	Ver ejemplo en atributo SubVersionNum