

## 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
<b>AddCert</b>	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 <code>0xFF000000</code> , color inválido, se genera para la certificación un color aleatorio (el mismo que en Presto se generaría si pulsamos [ <i>Supr</i> ] sobre el color)
<b>AddSpace</b>	Agrega un espacio con el código, descripción y color indicados y el siguiente número de espacio válido (en el orden establecido por Presto)
<b>BeginRedo</b>	Inicio de acciones que se pueden deshacer
<b>Calculate</b>	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)
<b>CheckProject</b>	Comprueba la obra abierta
<b>Close</b>	Cierra una obra de Presto
<b>Command</b>	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 ";"
<b>Copy</b>	Copia un campo en otro

<b>CreateSpace</b>	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 <i>0xFF000000</i> , color inválido, se genera para el espacio un color aleatorio (el mismo que en Presto se generaría si pulsamos [ <i>Supr</i> ] 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
<b>CreateVariable</b>	Crea / Actualiza un registro de la tabla de "Variables".
<b>Delete</b>	Borra una obra
<b>DeleteRecord</b>	Borra un registro y mantiene la integridad relacional
<b>Duplicate</b>	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
<b>EndRedo</b>	Fin de acciones que se pueden deshacer
<b>EvalNum</b>	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor numérico
<b>EvalStr</b>	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor de cadena
<b>Exit</b>	Cierra una instancia de Presto (sólo la ventana principal a la que nos referimos, no toda la aplicación)
<b>ExportFIEBDC</b>	Exporta la obra en formato FIEBDC
<b>ExportRPT</b>	Exporta un informe
<b>FindEqual</b>	Se posiciona en una tabla dado un valor de una clave
<b>FindFirst</b>	Se posiciona en el primer registro de una tabla según una clave
<b>FindGreat</b>	Se posiciona en el registro mayor a un valor dado de una tabla según una clave
<b>FindGreatEqual</b>	Se posiciona en el registro mayor o igual a un valor dado de una tabla según una clave
<b>FindLast</b>	Se posiciona en el último registro de una tabla según una clave
<b>FindLessEqual</b>	Se posiciona en el registro menor o igual a un valor dado de una tabla según una clave
<b>FindNext</b>	Se posiciona en el registro siguiente de una tabla
<b>Findpos</b>	Se posiciona en un registro de una tabla ordenada por una determinada clave, conociendo su posición
<b>FindPrev</b>	Se posiciona en el registro anterior de una tabla
<b>GetElement</b>	Se posiciona en el primer registro de una tabla

<b>GetField</b>	Devuelve el valor de un campo de Presto
<b>GetFieldBinary</b>	Devuelve el valor de un campo binario de Presto
<b>GetFieldNum</b>	Devuelve el valor de un campo numérico
<b>GetFieldStr</b>	Devuelve el valor de un campo de tipo cadena
<b>Getpos</b>	Obtiene la posición de un registro. La posición es un número asociado por el gestor de archivos a cada registro
<b>GetSelection</b>	Recorre los elementos seleccionados de una tabla
<b>GetText</b>	Devuelve el valor del texto con los mismos parámetros que otras funciones "Get".
<b>ImportCad</b>	Importa un archivo de CAD (AllPlan, ArchiCad, FIEBDC) sobre una obra de Presto
<b>ImportFIEBDC</b>	Importa la obra en formato FIEBDC
<b>InfoCode</b>	Información de un concepto de Presto
<b>InitRecord</b>	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
<b>InitVar</b>	Suprime variables declaradas con float o con char dentro de una expresión usada en EvalStr o EvalNum
<b>InsertAttachment</b>	Inserta un archivo en Presto a partir de un buffer y lo asocia con un registro de una tabla por medio del código
<b>InsertAttachmentFrom Path</b>	Inserta un archivo en Presto a partir de un <i>path</i> y lo asocia con un registro de una tabla por medio del código
<b>InsertRecord</b>	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
<b>LogBegin</b>	Inicia un proceso de log en Presto
<b>LogEnd</b>	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
<b>LogMsg</b>	Agrega un mensaje al sistema de log
<b>New</b>	Crea y abre una obra de Presto
<b>Open</b>	Abre una obra de Presto
<b>PrintRPT</b>	Imprime un informe
<b>Quit</b>	Cierra la aplicación de Presto
<b>Recalculate</b>	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

<b>Rename</b>	Cambia (renombrar) la clave de un concepto, estando posicionados en el concepto que se quiere renombrar
<b>SelCode</b>	Elige un concepto de Presto
<b>SetCharacter</b>	Selecciona el juego ANSI (Windows) o OEM (MS-DOS). Se usa antes de leer un texto de un registro o de evaluar una expresión
<b>SetConceptVarValue</b>	Asigna valor a la variable identificada por su <i>guid</i> 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
<b>SetCurrentInvPeriod</b>	Establece como certificación actual aprobada la fecha recibida
<b>SetElement</b>	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
<b>SetField</b>	Asigna un valor a un campo (en la tabla solo se graba con InsertRecord o UpdateRecord)
<b>SetFieldBinary</b>	Asigna un valor a un campo binario de Presto
<b>SetModal</b>	pone y quita el modo modal de Presto (mientras está puesto, la ventana de Presto está bloqueada): Bloquea o Desbloquea el uso de Presto
<b>SetTakeoffVarValue</b>	Asigna valor a la variable identificada por su <i>guid</i> en la línea de medición identificada por su <i>guidElement</i> y su <i>guidAux</i> . 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
<b>SetText</b>	Rellena un campo de texto.
<b>SetUniqueVarValue</b>	Asigna un valor de tipo de asignación "Única" a una variable. La variable debe existir previamente.
<b>SetUpdateScreen</b>	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
<b>SetVar</b>	Modifica una variable de Presto
<b>UpdateRecord</b>	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
<b>UpdateScreen</b>	Actualiza las ventanas

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

[#]	FechaDMA	NatC	Resumen	CertIni	CertMod	CertPte	Cert	PlanPres	OrCert	OrPlanPres	Nota	Color
1	30-Abr-18	1	Existente									Blue
2	31-May-18	4	Fase 1									Orange
3	30-Jun-18	6	Fase 2									Green
4	31-Jul-18	2	Fase 3									Light Green
5	31-Ago-18	5	Análisis									Red

---

**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)

---

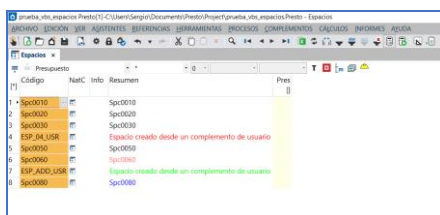
**Parámetros**    Entre corchetes se indica el valor por defecto  
*spcCode* [""], código del concepto de tipo espacio a crear.  
*spcDescription* [""], resumen para el concepto de tipo espacio.  
*spcColor* [0xFF000000], color para el concepto de tipo espacio

---

**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



<b>Función</b>	<b>BeginRedo</b>
<b>Sintaxis</b>	Function BeginRedo() As Long
<b>Descripción</b>	Inicio de acciones que se pueden deshacer
<b>Parámetros</b>	Ninguno
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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.</p> <pre> Public Sub Comienza_Deshacer() Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.BeginRedo</b> = 0 then MsgBox "Deshacer iniciado correctamente" Else MsgBox "No se inició correctamente deshacer" EndIf End Sub </pre>

<b>Función</b>	<b>Calculate</b>
<b>Sintaxis</b>	<i>Sub Calculate(calcMode As Long)</i>
<b>Descripción</b>	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)
<b>Parámetros</b>	<i>calcMode</i> 1: Calcular todo 2: Calcular precios 3: Calcular mediciones 4: Calcular tiempos 5: Calcular documentos 6: Calcular costes reales
<b>Retorno</b>	No devuelve nada



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

<b>Función</b>	<b>Close</b>
<b>Sintaxis</b>	Sub Close()
<b>Descripción</b>	Cierra una obra de Presto
<b>Parámetros</b>	Ninguno
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>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.</p> <pre> 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" <b>po.Close</b> MsgBox "Se ha cerrado la obra" End If End Sub </pre>

<b>Función</b>	<b>Command</b>
<b>Sintaxis</b>	Sub Command(opcion As String)
<b>Descripción</b>	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 ";"
<b>Parámetros</b>	opción es el nombre del menú de Presto que deseamos ejecutar
<b>Retorno</b>	Ninguno
<b>Ejemplos</b>	<p>En este ejemplo se muestra un procedimiento que ejecuta el menú de ver el árbol de un presupuesto</p> <pre>Public Sub Ver_Arbol() Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.Command</b> ("VER Árbol") End Sub</pre> <p>En este ejemplo se muestra un procedimiento que ejecuta la opción de exportación a Excel</p> <pre>Public Sub Exportar_Excel() Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.Command</b> ("Exportar Excel") End Sub</pre>

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

<b>Función</b>	<b>CreateSpace</b>
<b>Sintaxis</b>	Sub CreateSpace(spcNumber As Integer, spcCode As String, spcDescription As String, spcColor As Long)
<b>Descripción</b>	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
<b>Parámetros</b>	(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
<b>Retorno</b>	0 si pudo crear correctamente el espacio. Distinto de 0 en caso contrario
<b>Ejemplo</b>	Ver ejemplo en <i>AddSpace</i>

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	<p>Entre corchetes se indica el valor por defecto</p> <p><i>varGuid</i>, 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 <i>guid</i> la actualiza con la información indicada.</p> <p><i>varName</i>, nombre de la variable (puede estar repetido).</p> <p><i>varDescription</i> [""], descripción de la variable.</p> <p><i>varSource</i> [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.</p> <p><i>varType</i> [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 <i>Keyword</i> representan ocurrencias del término, pero no especifican valor para la variable.</p> <p><i>varCalcType</i> [0], indica si Presto calcula valores de abajo a arriba para la variable. 0: No calcular 1: Normal.</p> <p><i>varMinVal</i>, <i>varStepVal</i>, <i>varMaxVal</i> [0], en variables de tipo <i>Integer</i> o <i>Real</i> indican el valor mínimo y máximo (ambos incluidos) 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 <math>\leq (\text{máximo} - \text{mínimo})</math>.</p> <p><i>varIsCombo</i> [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 <i>Integer</i>, <i>Real</i> o <i>Cadena</i>.</p> <p><i>varComboList</i> [""], si la variable es <i>combo</i> permite especificar, separados por pipes (" "), los distintos valores que puede tomar la variable.</p> <p><i>varStatus</i> [0], afecta al color de texto de la descripción. 0: Negro 1: Rojo 2: Gris 3: Verde.</p> <p><i>varIsReadOnly</i> [0], con valor distinto de 0 indica que el usuario no podrá editar (en el interfaz de Presto) los valores asignados a la variable.</p> <p><i>revitparam&lt;*&gt;</i> [0] ó [""], datos provenientes de Revit.</p>

---

*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
```

---

<b>Función</b>	<b>DeleteRecord</b>
<b>Sintaxis</b>	Function DeleteRecord(tabla As String, [todo As Long]) As Long
<b>Descripción</b>	Borra un registro y mantiene la integridad relacional
<b>Parámetros</b>	tabla es el nombre de la tabla según informes todo debe valer 1 si queremos borrar todos los registros relacionados
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que elimina todos los registros cuya naturaleza sea "Maquinaria"</p> <pre> 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 </pre>

Función	<b>Duplicate</b>
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	<p>En este ejemplo se muestra un procedimiento que duplica el registro con código "C01" dando lugar a otro registro exactamente igual con código "C01B"</p> <pre> Public Sub Duplicar () Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.Duplicate</b> "C01", "C01B" End Sub </pre>

<b>Función</b>	<b>EndRedo</b>
<b>Sintaxis</b>	Sub EndRedo()
<b>Descripción</b>	Fin de acciones que se pueden deshacer
<b>Parámetros</b>	Ninguno
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>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.</p> <pre> Public Sub Finaliza_Deshacer() Dim po as Object Set po = CreateObject ("Presto.App.18") If po.BeginRedo = 0 then MsgBox "Deshacer iniciado correctamente" <b>po.EndRedo</b> MsgBox "Deshacer finalizado" Else MsgBox "No se inició correctamente deshacer" EndIf End Sub </pre>

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



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

<b>Función</b>	<b>Exit</b>
<b>Sintaxis</b>	Sub Exit()
<b>Descripción</b>	Cierra una instancia de Presto (sólo la ventana principal a la que nos referimos, no toda la aplicación)
<b>Parámetros</b>	Ninguno
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que cierra una de las dos instancias Presto que se crean.</p> <pre> Public Sub Cierra_Instancea() 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" <b>po.Exit</b> MsgBox "Se ha cerrado una instancia Presto" End Sub </pre>

<b>Función</b>	<b>ExportFIEBDC</b>
<b>Sintaxis</b>	Function ExportFIEBDC(archivo As String) As Long
<b>Descripción</b>	Exporta la obra en formato FIEBDC
<b>Parámetros</b>	archivo es el nombre del archivo de destino con path y con extensión.
<b>Retorno</b>	0 correcto Otro valor: error
<b>Ejemplo</b>	<p>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"</p> <pre> Public Sub Exporta_FIEBDC ( obra As String) Dim po as Object Set po = CreateObject ("Presto.App.18") po.Open obra <b>po.ExportFIEBDC</b> "C:\TEMP\EJEMPLO.BC3" End Sub </pre>

Función	ExportRPT
Sintaxis	Function ExportRPT(informe As String, archivo As String, parametros As Long, formato As Long, rellenar_blanco 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	<p>informe es el nombre del archivo que contiene el informe con path y con extensión</p> <p>archivo es el nombre del archivo destino con path y con extensión</p> <p>parametros valdrá 0 para tomar los parámetros por defecto y 1 para preguntarlos</p> <p>formato valdrá 0 para formato ASCII, 1 para formato RTF y 2 para formato PDF</p> <p>rellenar_blanco valdrá 0 para no rellenarlos y 1 para rellenarlos</p> <p>imprimir_cabecera valdrá: 0, nunca; 1, primera página y 2, siempre</p> <p>separador_campos valdrá: 0, ninguno; 1, espacio en blanco; 2, tabulador; 3, coma; 4, punto y coma</p> <p>imprimir valdrá 0 para imprimir todo el documento y 1 para imprimir un rango de páginas</p> <p>pagina_inicial contendrá el número de página de comienzo de la exportación</p> <p>pagina_final contendrá el número de página de fin de la exportación</p> <p>columnas indica el número de caracteres de cada línea. En formato PDF es la anchura (mm)</p> <p>lineas indica el número de líneas por hoja. En formato PDF es la altura (mm)</p> <p>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</p>
Retorno	<p>0 correcto</p> <p>Otro valor: error</p>
Ejemplo	<p>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.</p> <pre>Public Sub Exportar () Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.ExportRPT</b> "c:\informes\Presupuesto y mediciones.PrestoReport", "c:\informes\Presupuesto y mediciones.rft", 0, 1, 0, 1, 0, 1, 1, 5, 120, 60, 0"</pre>

---

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
```

---

<b>Función</b>	<b>FindFirst</b>
<b>Sintaxis</b>	Function FindFirst(tabla As String, [clave As String]) As Long
<b>Descripción</b>	se posiciona en el primer registro de una tabla según una clave
<b>Parámetros</b>	tabla es el nombre de la tabla según informes clave es clave según informes
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que nos posiciona en el primer registro de la tabla Conceptos.</p> <pre>Public Sub posicionar_Primer() Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.FindFirst</b> ("Conceptos") = 0 Then MsgBox "Situado en el primer registro de Conceptos" End If End Sub</pre>

<b>Función</b>	<b>FindGreat</b>
<b>Sintaxis</b>	Function FindGreat(tabla As String, clave As String, valor) As Long
<b>Descripción</b>	se posiciona en el registro mayor a un valor dado de una tabla según una clave
<b>Parámetros</b>	tabla es el nombre de la tabla según informes clave es clave según informes valor es el valor de la clave que se busca
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que nos posiciona en el primer registro de la tabla Conceptos.</p> <pre>Public Sub posicionar_Mayor(valor As String) Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.FindGreat</b> ("Conceptos", "Conceptos.Código", valor) = 0 Then MsgBox "Situado en el siguiente registro a " &amp; valor End If End Sub</pre>

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

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

<b>Función</b>	<b>FindLessEqual</b>
<b>Sintaxis</b>	Function FindLessEqual(tabla As String, clave As String, valor) As Long
<b>Descripción</b>	se posiciona en el registro menor o igual a un valor dado de una tabla según una clave
<b>Parámetros</b>	tabla es el nombre de la tabla según informes clave es clave según informes
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que nos posiciona en el registro menor o igual de la tabla Conceptos.</p> <pre> Public Sub posicionar_MenorIgual(valor As String) Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.FindLessEqual</b> ("Conceptos", "Conceptos.Código", valor) = 0 Then MsgBox "Situado en el anterior registro a " &amp; valor End If End Sub </pre>

<b>Función</b>	<b>FindNext</b>
<b>Sintaxis</b>	Function FindNext(tabla As String, [clave As String]) As Long
<b>Descripción</b>	se posiciona en el registro siguiente de una tabla
<b>Parámetros</b>	tabla es el nombre de la tabla según informes clave es clave según informes
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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.</p> <pre> Public Sub posicionar_Siguiente (valor As String) Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.FindNext</b> ("Conceptos", "Conceptos.Código") = 0 Then MsgBox "Situado en registro con Conceptos.Código siguiente" End If End Sub </pre>

<b>Función</b>	<b>Findpos</b>
<b>Sintaxis</b>	Function Findpos(tabla As String, clave As String, pos As Long) As Long
<b>Descripción</b>	Se posiciona en un registro de una tabla ordenada por una determinada clave, conociendo su posición
<b>Parámetros</b>	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
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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"</p> <pre> 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 <b>po.Findpos</b> "Conceptos", "Conceptos.Código", Num_Registro End Sub </pre>



<b>Función</b>	<b>FindPrev</b>
<b>Sintaxis</b>	Function FindPrev(tabla As String, [clave As String]) As Long
<b>Descripción</b>	se posiciona en el registro anterior de una tabla
<b>Parámetros</b>	tabla es el nombre de la tabla según informes clave es clave según informes
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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.</p> <pre> Public Sub posicionar_Siguiente (valor As String) Dim po as Object Set po = CreateObject ("Presto.App.18") If <b>po.FindPrev</b>( "Conceptos", " Conceptos.Código") = 0 Then MsgBox "Situado en registro con Conceptos.Código anterior" End If End Sub </pre>

<b>Función</b>	<b>GetElement</b>
<b>Sintaxis</b>	Function GetElement(elemento As Long, [anida As Long]) As Long
<b>Descripción</b>	Se posiciona en el primer registro de una tabla
<b>Parámetros</b>	<p>elemento es el número de elemento asociado al SetElement con el que abrimos la tabla</p> <p>anida indica si queremos que se posicione en los registros relacionados al igual que elemento del módulo de Informes</p>
<b>Retorno</b>	<p>Recupera un elemento de una tabla abierta con SetElement y devuelve distinto de cero cuando ya no quedan</p> <p>Elemento es el número indicado por el SetElement con el que se abrió la tabla</p>
<b>Ejemplo</b>	<p>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 recorreremos la tabla con la función GetElement y contamos el número de registros obtenido.</p> <pre> 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 <b>po.GetElement</b> (1) = 0 Num_Registros = Num_Registros + 1 Wend MsgBox "Número de registros de tipo capítulo: " &amp; CStr(Num_Registros) End Sub </pre>

---

<b>Función</b>	<b>GetField</b>
<b>Sintaxis</b>	Function GetField(campo As String, [dimension As Long])
<b>Descripción</b>	Devuelve el valor de un campo de Presto
<b>Parámetros</b>	campo es el nombre del campo que se quiere obtener dimension es la dimensión del campo solicitado. por defecto la 0
<b>Retorno</b>	Devuelve el valor del campo indicado como parámetro
<b>Ejemplo</b>	En este ejemplo se muestra un procedimiento que muestra al usuario todos los Conceptos.Pres  <pre>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 ( <b>po.GetField</b> ( "Conceptos.Pres")) Wend End Sub</pre>

---

<b>Función</b>	<b>GetFieldBinary</b>
<b>Sintaxis</b>	Function GetFieldBinary(campo As String)
<b>Descripción</b>	Devuelve el valor de un campo binario de Presto
<b>Parámetros</b>	campo es el nombre del campo que se quiere obtener
<b>Retorno</b>	Devuelve el valor del campo indicado como parámetro
<b>Ejemplo</b>	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

```

<b>Función</b>	<b>GetFieldNum</b>
<b>Sintaxis</b>	Function GetFieldNum(campo As String, [dimension As Long]) As Double
<b>Descripción</b>	Devuelve el valor de un campo numérico
<b>Parámetros</b>	campo es el nombre del campo que se quiere obtener dimension es la dimensión del campo solicitado. por defecto la 0
<b>Retorno</b>	Devuelve el valor del campo indicado como parámetro
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que muestra al usuario todos los Conceptos.Pres</p> <pre> 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( <b>po.GetFieldNum</b> ("Conceptos.Pres")) Wend End Sub </pre>

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

<b>Función</b>	<b>Getpos</b>
<b>Sintaxis</b>	Function Getpos(tabla As String) As Long
<b>Descripción</b>	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
<b>Parámetros</b>	tabla es el nombre de la tabla según informes
<b>Retorno</b>	posición del registro activo en la tabla indicada
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que indica al usuario la posición del primer registro de tipo capítulo.</p> <pre> 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 = <b>po.Getpos</b> ( "Conceptos" )     MsgBox "posición del primer registro de tipo capítulo: " &amp; CStr(Num_Registro) End If End Sub </pre>

<b>Función</b>	<b>GetSelection</b>
<b>Sintaxis</b>	Function GetSelection(first As Long, tabla As String) As Long
<b>Descripción</b>	Recorre los elementos seleccionados de una tabla
<b>Parámetros</b>	<p>first indica si queremos obtener el primer elemento seleccionado o recorrerlos. Si first = 1 nos colocamos en el primero, si first = 0 los recorreremos</p> <p>tabla indica el esquema cuyos elementos seleccionados queremos recorrer</p>
<b>Retorno</b>	Recupera un elemento seleccionado de una tabla indicada y devuelve distinto de cero cuando ya no quedan
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que recorre todos 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.</p> <pre> Public Sub Ver_Seleccion()     Dim first As Long     first = 1     While <b>po.GetSelection</b> (first, "Conceptos") = 0         first = 0         MsgBox ( "Código: " &amp; po.GetFieldStr ( "Conceptos.Código" ) &amp; vbCrLf &amp; "Resumen: " &amp; po.GetFieldStr ( "Conceptos.Resumen" ))     Wend End Sub </pre>

<b>Función</b>	<b>GetText</b>
<b>Sintaxis</b>	Function GetText(campo As String, [tipo As Long = 1]) As String
<b>Descripción</b>	Devuelve el valor del texto con los mismos parámetros que arriba.
<b>Parámetros</b>	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
<b>Retorno</b>	Valor del campo texto
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que obtiene la variable Conceptos.Texto de todos los registros.</p> <pre> Public Sub Leer_Memo () Dim po as Object Set po = CreateObject ("Presto.App.18") po.SetElement 1, "Conceptos" While po.GetElement (1) = 0 MsgBox <b>po.GetText</b> ("Conceptos.Texto") Wend End Sub </pre>

<b>Función</b>	<b>ImportCad</b>
<b>Sintaxis</b>	Function ImportCad(proyecto As String, archivo As String, tipo As Long) As Long
<b>Descripción</b>	Importa un archivo de cad (AllPlan, ArchiCad, FIEBDC) sobre una obra de Presto
<b>Parámetros</b>	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
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que importa un archivo de CAD recibido como parámetro sobre una obra</p> <pre> Public Sub Importa_CAD (obra As String, archivo As String) Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.ImportCad</b> obra, archivo, 0 End Sub </pre>



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

<b>Función</b>	<b>InfoCode</b>
<b>Sintaxis</b>	Function InfoCode(proyecto As String, codigo As String, unidad As String, resumen As String, precio As String) As Long
<b>Descripción</b>	Información de un concepto de Presto
<b>Parámetros</b>	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
<b>Retorno</b>	0 correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se obtiene la informacion de un concepto recibido como parámetro.</p> <pre> 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") <b>po.InfoCode</b> obra, codigo, unidad, resumen, precio End Sub </pre>

<b>Función</b>	<b>InitRecord</b>
<b>Sintaxis</b>	Function InitRecord(tabla As String) As Long
<b>Descripción</b>	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
<b>Parámetros</b>	tabla es el nombre de la tabla según informes
<b>Retorno</b>	0 correcto Otro valor: error
<b>Ejemplo</b>	<p>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</p> <pre>Public Sub Inicializa_Registro(archivo As String)     <b>po.InitRecord</b> archivo End Sub</pre>

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

Función	<b>InsertAttachment</b>
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	<p>Entre corchetes se indica el valor por defecto</p> <p><i>tableName</i>, nombre de la tabla a la que pertenece el código al cual queremos asociar el archivo.</p> <p>"Conceptos"</p> <p>"Pedidos"</p> <p>"Entregas"</p> <p>"Facturas"</p> <p><i>code</i>, código que identificar el registro al cual queremos asociar el archivo.</p> <p><i>attachDescription</i>, descripción del archivo asociado.</p> <p><i>attachExtension</i>, extensión para el archivo asociado.</p> <p><i>attachBuffer</i>, buffer con el contenido del archivo asociado.</p> <p><i>attachType</i> [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.</p> <p><i>resetSourcePath</i> [0], indica con valor distinto de 0 que tras insertar el archivo asociado hay que borrar el campo <i>Archivos.Camino</i> de Presto e inicializar <i>Archivos.Fecha</i> con la fecha actual. Es útil cuando el archivo a insertar está en disco de forma temporal</p>
Retorno	Devuelve el <i>guid</i> del archivo asociado creado o cadena vacía en caso de error
Ejemplo	<p>En este ejemplo se muestra un procedimiento que inserta un archivo asociado a un código determinado</p> <pre>Public Sub Inserta_Asociado(codigo As String, buffer) Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.InsertAttachment</b> "Conceptos", codigo, "Prueba", "TXT", buffer End Sub</pre>

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

<b>Función</b>	<b>InsertRecord</b>
<b>Sintaxis</b>	Function InsertRecord(tabla As String) As Long
<b>Descripción</b>	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
<b>Parámetros</b>	tabla es el nombre de la tabla según informes
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se inserta un registro en la tabla que se recibe como parámetro</p> <pre> Public Sub Inserta_Registro(archivo As String) Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.InsertRecord</b> archivo End Sub </pre>

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

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

<b>Función</b>	<b>LogMsg</b>
<b>Sintaxis</b>	Function LogMsg(logMsg As String, mode As Long)
<b>Descripción</b>	Agrega un mensaje al sistema de log
<b>Parámetros</b>	logMsg es el texto del mensaje mode es el tipo de mensaje, 0: información 1: advertencia 2: error
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<pre> Set po = GetObject("", "Presto.App.18") Dim nMensajes Dim strMensaje If <b>po.LogBegin</b>("Comienzo del LOG") = 0 Then     nMensajes = 1     While nMensajes &lt; 10         strMensaje = "Mensaje de información nº " &amp; nMensajes         <b>po.LogMsg</b> strMensaje, 0         nMensajes = nMensajes + 1     Wend     <b>po.LogEnd</b> End If </pre>

<b>Función</b>	<b>New</b>
<b>Sintaxis</b>	Function New(obra As String) As Long
<b>Descripción</b>	Crea y abre una obra de Presto
<b>Parámetros</b>	obra es la ruta de la obra a crear
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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.</p> <p>Agregue la referencia de entre los tipos COM, "Presto 10.0 Type Library" en su compilador.</p> <pre> Public Sub Nueva_Obra(<i>path As String</i>) Dim po As PrestoLib.PrestoApplication Set po = CreateObject ("Presto.App.18") If <b>po.New</b>( path)= 0 Then MsgBox "Obra creada correctamente" Else MsgBox "No se creó correctamente la obra" EndIf End Sub </pre>

<b>Función</b>	<b>Open</b>
<b>Sintaxis</b>	Function Open(obra As String, [solo_lectura As Long]) As Long
<b>Descripción</b>	Abre una obra de Presto
<b>Parámetros</b>	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
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	<p>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.</p> <pre> Public Sub Abrir_Obra (<i>path As String</i>) Dim po As Object Set po = CreateObject ("Presto.App.18") If <b>po.Open</b>( path,1)= 0 Then MsgBox "Obra abierta correctamente" Else MsgBox "No se abrió correctamente la obra" EndIf End Sub </pre>



<b>Función</b>	<b>PrintRPT</b>
<b>Sintaxis</b>	Function PrintRPT(informe As String, [parametros As Long], [copias As Long], [impresora As Long]) As Long
<b>Descripción</b>	Imprime un informe
<b>Parámetros</b>	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
<b>Retorno</b>	0 correcto Otro valor: error
<b>Ejemplo</b>	<p>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</p> <pre> Public Sub Informe () Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.PrintRPT</b> "C:\informes\ejemplo.rpt", 0, 3, 1 End Sub </pre>

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

<b>Función</b>	<b>Quit</b>
<b>Sintaxis</b>	Sub Quit()
<b>Descripción</b>	Cierra la aplicación de Presto
<b>Parámetros</b>	Ninguno
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que cierra la aplicación Presto.</p> <pre> 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 </pre>

<b>Atributo</b>	<b>ReadOnly</b>
<b>Sintaxis</b>	Property ReadOnly As Long
<b>Descripción</b>	Devuelve si la obra está en modo "solo lectura"
<b>Parámetros</b>	---
<b>Retorno</b>	---
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que visualiza si la obra está en "sólo lectura"</p> <pre> Public Sub Ver_SoloLectura() Dim po as Object Set po = CreateObject ("Presto.App.18") po.Open("C:\Temp\Obra.PrestoObra") If (<b>po.ReadOnly</b>) MsgBox "Solo lectura"  Else MsgBox "No solo lectura" EndIf End Sub </pre>

<b>Función</b>	<b>Recalculate</b>
<b>Sintaxis</b>	Sub Recalculate(activar As Long)
<b>Descripción</b>	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.
<b>Parámetros</b>	0: Desactiva 1: Activa de nuevo y recalcula
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que bloquea los recálculos de Presto</p> <pre> Public Sub Bloquea_Calculos() Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.Recalculate</b> (0) End Sub </pre>

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

<b>Función</b>	<b>SelCode</b>
<b>Sintaxis</b>	Function SelCode(HWND As Long, proyecto As String, relacion As String) As Long
<b>Descripción</b>	Elige un concepto de Presto
<b>Parámetros</b>	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
<b>Retorno</b>	Devuelve 0: Código en blanco 1: Código seleccionado -1: Cancelar Otros: error
<b>Ejemplo</b>	<p>En este ejemplo se obtiene la información de un concepto recibido como parámetro.</p> <pre>Public Sub Sel_Code (obra As String ,relacion As String)   Dim po as Object   Set po = CreateObject ("Presto.App.18")   <b>po.SelCode</b> 0, obra, relacion End Sub</pre>

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

<b>Función</b>	<b>SetConceptVarValue</b>
<b>Sintaxis</b>	Sub SetConceptVarValue(varGuid As String, conceptCode As String, varValue As Variant, valType As Integer, valUserTxt As String, valUserNum As Double, revitparamElementId As String)
<b>Descripción</b>	Asigna valor a la variable identificada por su <i>guid</i> 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".
<b>Parámetros</b>	Entre corchetes se indica el valor por defecto <i>varGuid</i> , identificador único global de la variable. <i>conceptCode</i> , código del concepto donde recibe valor la variable. <i>varValue</i> , valor que se asociará a la variable. <i>valType</i> [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). <i>valUserTxt</i> [""], texto de usuario (nota) asociado al valor. <i>valUserNum</i> [0], valor numérico de usuario asociado al valor. <i>revitparamElementId</i> [""], dato proveniente de Revit.
<b>Retorno</b>	0 si pudo asignar correctamente el valor a la variable. Distinto de 0 en caso contrario
<b>Ejemplo</b>	ver ejemplo en <i>SetTakeoffVarValue</i>

<b>Función</b>	<b>SetCurrentInvPeriod</b>
<b>Sintaxis</b>	Function SetCertificacion(fecha as String) As Long
<b>Descripción</b>	Establece como certificación actual aprobada la fecha recibida.
<b>Parámetros</b>	Fecha de la certificación en formato yyyyymmdd. Tiene que existir previamente.
<b>Retorno</b>	0: correcto Otro valor: error
<b>Ejemplo</b>	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.  <pre>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")</pre>

---

```
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
```

---

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



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

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

<b>Función</b>	<b>SetTakeoffVarValue</b>
<b>Sintaxis</b>	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)
<b>Descripción</b>	Asigna valor a la variable identificada por su <i>guid</i> en la línea de medición identificada por su <i>guidElement</i> y su <i>guidAux</i> . 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".
<b>Parámetros</b>	Entre corchetes se indica el valor por defecto) <i>varGuid</i> , identificador único global de la variable. <i>takeoffGuidElement</i> , <i>takeoffGuidAux</i> , identificador único global de la línea de medición obtenido como concatenación de ambos <i>guids</i> . <i>varValue</i> , valor que se asociará a la variable. <i>valType</i> [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). <i>valUserTxt</i> [""], texto de usuario (nota) asociado al valor. <i>valUserNum</i> [0], valor numérico de usuario asociado al valor. <i>revitparamElementId</i> [""], dato proveniente de Revit.
<b>Retorno</b>	0 si pudo asignar correctamente el valor a la variable. Distinto de 0 en caso contrario.
<b>Ejemplo</b>	<pre> 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) </pre>

---

```
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)
```

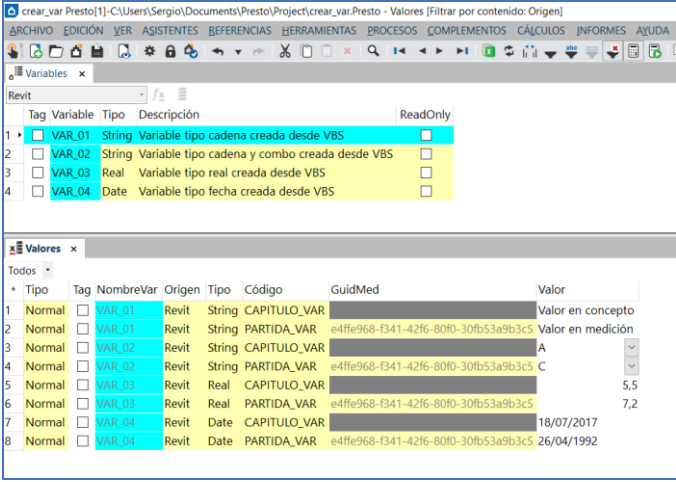
```
' Variable de tipo fecha
```

```
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()
```



Tipo	Tag	NombreVar	Origen	Tipo	Código	GuidMed	Valor
1 Normal	<input type="checkbox"/>	VAR_01	Revit	String	CAPITULO_VAR		Valor en concepto
2 Normal	<input type="checkbox"/>	VAR_01	Revit	String	PARTIDA_VAR	e4ffe968-f341-42f6-80f0-30fb53a9b3c5	Valor en medición
3 Normal	<input type="checkbox"/>	VAR_02	Revit	String	CAPITULO_VAR		A
4 Normal	<input type="checkbox"/>	VAR_02	Revit	String	PARTIDA_VAR	e4ffe968-f341-42f6-80f0-30fb53a9b3c5	C
5 Normal	<input type="checkbox"/>	VAR_03	Revit	Real	CAPITULO_VAR		5,5
6 Normal	<input type="checkbox"/>	VAR_03	Revit	Real	PARTIDA_VAR	e4ffe968-f341-42f6-80f0-30fb53a9b3c5	7,2
7 Normal	<input type="checkbox"/>	VAR_04	Revit	Date	CAPITULO_VAR		18/07/2017
8 Normal	<input type="checkbox"/>	VAR_04	Revit	Date	PARTIDA_VAR	e4ffe968-f341-42f6-80f0-30fb53a9b3c5	26/04/1992

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

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

<b>Función</b>	<b>SetUpUpdateScreen</b>
<b>Sintaxis</b>	Sub SetUpUpdateScreen(activar As Long)
<b>Descripción</b>	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 SetUpUpdateScreen con el parámetro 1 para reflejar todos los cambios de una sola vez.
<b>Parámetros</b>	0: desactiva 1: activa
<b>Retorno</b>	Ninguno
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que bloquea los cambios en las ventanas de Presto</p> <pre> Public Sub Bloquea_VentanaPresto() Dim po as Object Set po = CreateObject ("Presto.App.18") <b>po.SetUpUpdateScreen ( 0)</b> End Sub </pre>

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

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	<p>En este ejemplo se muestra un procedimiento que visualiza el valor de Status después de realizar cualquier operación</p> <pre> Public Sub Ver_Status() Dim po as Object Set po = CreateObject ("Presto.App.18") MsgBox CStr ( <b>po.Status</b>) End Sub </pre>

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



<b>Función</b>	<b>UpdateRecord</b>
<b>Sintaxis</b>	Function UpdateRecord(tabla As String) As Long
<b>Descripción</b>	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
<b>Parámetros</b>	tabla es el nombre de la tabla según informes
<b>Retorno</b>	0 correcto Otro valor: error
<b>Ejemplo</b>	<p>En este ejemplo se muestra un procedimiento que modifica el Resumen de todos los registros cuya naturaleza sea "Partida"</p> <pre> 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" <b>po.UpdateRecord</b> "Conceptos" Wend End Sub </pre>

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

---

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

---

---

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

---