

API Funciones detalladas

Definición de cada función, parámetros, con el valor de defecto entre corchetes, la acción o retorno y un ejemplo para Visual Basic.

Si usa Visual Basic Script elimine el nombre de la rutina Public Sub, su línea de End Sub y las líneas de definición de variables Dim.

AddCert.....	3
AddSpace	4
BeginRedo	5
Calculate	6
CheckProject.....	7
Close.....	8
CloseWindow	9
Command	10
Copy.....	11
CreateSpace	12
CreateVariable	13
Delete.....	15
DeleteRecord	16
Duplicate	17
EndRedo	18
EvalNum	19
EvalStr	20
Exit.....	21
ExecuteOption	22
ExportFIEBDC	23
ExportRPT	24
FindEqual.....	26
FindFirst.....	27
FindGreat.....	28
FindGreatEqual.....	29
FindLast	30
FindLessEqual	31
FindNext.....	32
FindPos.....	33
FindPrev.....	34
GetElement	35
GetField	36
GetFieldBinary	37
GetFieldNum.....	38
GetFieldStr.....	39
GetPos	40
GetSelection	41
GetText.....	42

ImportCad	43
ImportFIEBDC.....	44
InfoCode	45
InitRecord.....	46
InitVar	47
InsertAttachment.....	48
InsertAttachmentFromPath	49
InsertAttachmentFromUrl	50
InsertRecord	51
LogBegin.....	52
LogEnd.....	53
LogMsg	54
New.....	55
Open	56
PrintRPT	57
Project.....	57
ProjectLanguage	58
Quit	59
ReadOnly	60
Recalculate	61
Rename	62
SelCode	63
SetCharacter.....	64
SetConceptVarValue	65
SetCurrentInvPeriod.....	66
SetElement	67
SetField.....	68
SetFieldBinary.....	69
SetModal.....	70
SetPriceVarValue	71
SetRelationVarValue	72
SetTakeoffInsertMode	74
SetTakeoffVarValue	77
SetText.....	80
SetUniqueVarValue	81
SetUpdateScreen.....	82
SetVar	83
ShowWindow	84
SQLServerDelete.....	85
SQLServerExport.....	86
SQLServerImport	87
SubVersionNum	89
UpdateRecord.....	90
UpdateScreen	91
VersionNum	92
VersionStr	93

Función	AddCert
Sintaxis	Sub AddCert (certDescription As String, certColor As Integer)
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 <i>0xFF000000</i> , 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)
Parámetros	<i>certDescription</i> [""], resumen de la certificación en la tabla Agenda. <i>certColor</i> [0xFF000000], color para la certificación en la tabla Agenda
Retorno	El número de la certificación agregada o 0 en caso de error

Ejemplo

```
Public Sub poner_Str ()
    Dim po As PrestoLib.PrestoApplication
    po = CreateObject ("Presto.App.18")
    po.AddCert ("Existente", 16711680) 'Azul
    po.AddCert ("Fase 1")
    po.AddCert ("Fase 2", 65280) 'Verde
    po.AddCert ("Fase 3")
    po.AddCert ("Análisis", 255) 'Roja
End Sub
```

Función	AddSpace
Sintaxis	Function AddSpace (spcCode As String, spcDescription As String, spcColor As Integer, bimLevel as Double) As Integer
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	<i>spcCode</i> [""], código del concepto de tipo espacio a crear. <i>spcDescription</i> [""], resumen para el concepto de tipo espacio. <i>spcColor</i> [0xFF000000], color para el concepto de tipo espacio <i>bimLevel</i> [0], nivel BIM para el concepto de tipo espacio
Retorno	El número del espacio añadido si se pudo crear. 0 en caso contrario

Ejemplo

Dim po As Object

po = CreateObject ("Presto.App.18")

'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")

Función	BeginRedo
Sintaxis	Function BeginRedo () As Integer
Descripción	Inicio de acciones que se pueden deshacer
Parámetros	Ninguno
Retorno	0: correcto, otro valor: error

Ejemplo

Procedimiento que arranca una instancia de Presto y activa la opción deshacer. Finalmente se indica al usuario si se inició correctamente.

```
Public Sub Comienza_Deshacer ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    If po.BeginRedo = 0 Then  
        MsgBox ("Deshacer iniciado correctamente")  
    Else  
        MsgBox ("No se inició correctamente deshacer")  
    End If  
End Sub
```

Función	Calculate
Sintaxis	<i>Sub Calculate (calcMode As Integer)</i>
Descripción	Lanza en Presto uno de los cálculos del panel “Inicio: Calcular”. 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	<p><i>calcMode</i></p> <ul style="list-style-type: none"> 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

Ejemplo

Procedimiento que comprueba la obra abierta

```
Public Sub Comprueba_Obra ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.CheckProject ()  
End Sub
```

Función Close

Sintaxis Sub Close ()

Descripción Cierra una obra de Presto

Ejemplo

Procedimiento que cierra la obra activa en Presto. Primero creamos y abrimos una obra de ejemplo en la ruta "C:\TEMP\EJEMPLO.Presto" y posteriormente si se ha creado correctamente la cerramos mostrando un mensaje.

```
Public Sub Cierra_Obra ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    If po.New ("C:\TEMP\EJEMPLO.Presto") = 0 Then
        MsgBox ("Obra creada correctamente")
        po.Close ()
        MsgBox ("Se ha cerrado la obra")
    End If
End Sub
```


Función	CloseWindow
Sintaxis	Sub CloseWindow (Ventana As Integer)
Descripción	Cierra la ventana principal indicada
Parámetros	0: Presupuesto 1: Árbol 2: Mediciones 3: Mediciones temporales 4: Espacios
Retorno	Ninguno

Ejemplo

Procedimiento que añade un espacio a una obra de Presto, abre la ventana principal "Espacios", la posiciona como activa y posteriormente la cierra.

Dim po As Object

```
po = GetObject ("", "Presto.App.18")
po.SetUpdateScreen (0)
po.BeginRedo ()
```

' Espacio con número, código, descripción, color (rojo) y cota (10.77).

' Como el espacio es > 1, añadirá los espacios intermedios por defecto

```
po.CreateSpace (4, "ESP_COTA_10_77", "Espacio creado desde un complemento de
usuario", 255, 10.77)
```

' Añadimos un espacio a continuación del último, con su código, resumen, color (verde) y cota (0.55)

```
po.AddSpace ("ESP_COTA_0_55", "Espacio creado desde un complemento de
usuario", 65280, 0.55)
```

' Mostramos la ventana de espacios

```
po.ShowWindow (4)
po.EndRedo ()
po.SetUpdateScreen (1)
po.UpdateScreen ()
```

' Cerramos la ventana de espacios

```
po.CloseWindow (4)
po.UpdateScreen ()
```

```
MsgBox ("FIN inserta espacios !")
```

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	opcion es el nombre del menú de Presto que deseamos ejecutar
Retorno	Ninguno

Ejemplos

Ejecuta el menú de ver el árbol de un presupuesto

```
Public Sub Ver_Arbol ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.Command ("Presupuesto|Árbol")
End Sub
```

Ejecuta la opción de exportación a Excel

```
Public Sub Exportar_Excel ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.Command ("Exportar|Excel")
End Sub
```

Función	Copy
Sintaxis	Sub Copy (origen As String, destino As String, [dimension1 As Integer], [dimension2 As Integer])
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

Copia para todos los registros el valor de Conceptos.Código en Conceptos.Código2

```
Public Sub Copiar ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        po.Copy ("Conceptos.Código", "Conceptos.Código2")
        po.UpdateRecord ("Conceptos")
    End While
End Sub
```

Función	CreateSpace
Sintaxis	Sub CreateSpace (spcNumber As Integer, spcCode As String, spcDescription As String, spcColor As Integer, bimLevel as Double)
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 <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
Parámetros	<i>spcNumber</i> , número de espacio a crear, >=1 y <= 9999. <i>spcCode</i> [""], código del concepto de tipo espacio a crear. <i>spcDescription</i> [""], resumen para el concepto de tipo espacio <i>spcColor</i> [0xFF000000], color para el concepto de tipo espacio <i>bimLevel</i> [0], nivel BIM 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	Function 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) As String
Descripción	Crea / Actualiza un registro de la tabla de "Variables"
Parámetros	Variables, tomada de la tabla siguiente.
Retorno	Devuelve el <i>guid</i> de la variable creada o actualizada. En caso de error devuelve un <i>guid</i> vacío.

Variable	Significado
<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
<i>varName</i>	Nombre de la variable (puede estar repetido).
<i>varDescription</i> [""]	Descripción de la variable.
<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.
<i>varType</i> [1]	Indica el tipo de valor que puede recibir. 4: Integer 5: Real 6: String 7: Date 9: Bool 20: 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.
<i>varCalcType</i> [0]	Indica si Presto calcula valores de abajo a arriba para la variable. 0: No 1: Como precio de presupuesto 2: Suma.

Variable	Significado
<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 $\leq (\text{máximo}-\text{mínimo})$.
<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> .
<i>varComboList</i> [“”]	Si la variable es <i>combo</i> permite especificar, separados por <i>pipes</i> (“ ”), los distintos valores que puede tomar la variable.
<i>varStatus</i> [0]	Afecta al color de texto de la descripción. 0: Negro 1: Rojo 2: Gris 3: Verde.
<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.
<i>revitparam</i> <*> [0] ó [“”]	Datos provenientes de Revit.
<i>varDec</i> [0]	Redondeo para variables de tipo <i>Real</i> .
<i>varExp</i> [“”]	Expresión de cálculo.
<i>varCond</i> [“”]	Condición para el cálculo de la expresión.

Ejemplo

Ver ejemplo en asignación de valores.

Función Delete

Sintaxis Function Delete (obra As String) As Integer

Descripción Borra una obra

Parámetros Obra es la ruta completa de la obra a borrar

Retorno 0: correcto, otro valor: error

Ejemplo

Borra la obra "C:\TEMP\EJEMPLO.Presto"

```
Public Sub Borrar_Obra ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.Delete ("C:\TEMP\EJEMPLO.Presto")  
End Sub
```

Función	DeleteRecord
Sintaxis	Function DeleteRecord (tabla As String, [todo As Integer]) As Integer
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

Elimina todos los registros cuya naturaleza sea "Maquinaria"

```
Public Sub Borrar_Registro ()
    Dim s As String
    Dim po As Object, arrElementos () As String, i As Integer
    po = CreateObject ("Presto.App.18")
    po.Open ("C:\Temp\CENZANO Presupuesto y control de costes.Presto")
    po.SetElement (1, "Conceptos",,, "Conceptos.Nat == 7")
    s = ""
    While po.GetElement (1) = 0
        s = s + po.GetFieldStr ("Conceptos.Código") + ";"
    End While
    arrElementos = Split (s, ";")
    For i = LBound (arrElementos) To UBound (arrElementos)
        If po.FindEqual ("Conceptos", "Conceptos.Código", arrElementos (i)) = 0
    Then
                po.DeleteRecord ("Conceptos")
        End If
    Next
End Sub
```


Función	Duplicate
Sintaxis	Function Duplicate (codigo As String, nuevo As String) As Integer
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

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
    po = CreateObject ("Presto.App.18")
    po.Duplicate ("C01", "C01B")
End Sub
```

Función EndRedo

Sintaxis Sub EndRedo ()

Descripción Fin de acciones que se pueden deshacer

Ejemplo

Se 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
    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")
    End If
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
Retorno	Numérico resultante de la evaluación de la expresión

Ejemplo

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"
    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 evalúa el generador de expresiones.
Retorno	Cadena resultante de la evaluación de la expresión

Ejemplo

Se declara una variable que recibe como parámetro

```
Public Sub Declara_Cadena (variable As String)
    Dim po As Object
    'variable = "DivISO[1]"
    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)

Ejemplo

Se cierra una de las dos instancias Presto que se crean.

```
Public Sub Cierra_Instance ()  
    Dim po As Object  
    Dim OtroPresto As Object  
    po = CreateObject ("Presto.App.18")  
    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 ExecuteOption

Sintaxis Sub ExecuteOption (DialogOptionsCode,texto en formato JSon)

Ejemplo

Se hace un login a un PrestoServer, se abre una obra y se ejecuta la opción "Herramientas: Operar" con los parámetros del Json.

```
Sub ejecutaOption ()
    'ejecutaOption *****
    Dim strMethodName As String = MethodBase.GetCurrentMethod ().Name
    'Utils.PrintMethodFormatted (strMethodName)
    Dim User As PrestoClientApiUser = New PrestoClientApiUser (URL)
    Dim responseValue As ResponseValue = User.Login (strUser, strPassword)
    'Utils.PrintJsonResponse ("Login (" & m_Options.User & ")", User.m_strJsonResponse)
    If responseValue = responseValue.Ok Then
        Dim Project As PrestoClientApiProject = New PrestoClientApiProject (User)
        responseValue = Project.Open ("ESP\Modelo Educativo de Revit.Presto")
        'Utils.PrintJsonResponse ("OpenProject (" & m_Options.Project & ")",
        Project.m_strJsonResponse)
        If responseValue = responseValue.Ok Then
            Dim Com As PrestoClientApiCom = New PrestoClientApiCom (Project)
            Dim retValue As Integer = Com.OpenObj ()
            'Utils.PrintJsonResponse ("OpenCom", Com.m_strJsonResponse)
            If retValue = 0 Then
                With Com
                    Dim jobject As JObject = New JObject From {
                        {"concept", "0"},
                        {"checkedLabor", 1},
                        {"checkedPlant", 1},
                        {"checkedMaterials", 1},
                        {"checkedOther", 1},
                        {"percentageLabor", 90},
                        {"percentagePlant", 70},
                        {"percentageMaterials", 60},
                        {"percentageOther", 20}
                    }
                    .ExecuteOption (DialogOptionsCode.DLG_MULTIPLICAR_PRECIOS, jobject)
                    'Utils.PrintJsonResponse ("ExecuteOption", Com.m_strJsonResponse)
                    .Close ()
                End With
            End If
        End If
        MsgBox ("FIN")
    End If
    User.Logout ()
End Sub
```

Función	ExportFIEBDC
Sintaxis	Function ExportFIEBDC (archivo As String) As Integer
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

Se abre una obra y la exporta a FIEBDC en la ruta.

```
Public Sub Exporta_FIEBDC (obra As String)
    Dim po As Object
    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 Integer, formato As Integer, rellenar_blanco As Integer, imprimir_cabecera As Integer, separador_campos As Integer, imprimir As Integer, pagina_inicial As Integer, pagina_final As Integer, columnas As Integer, lineas As Integer, anadir As Integer) As Integer
Descripción	Exporta un informe
Parámetros	Indicados en la tabla siguiente
Retorno	0 correcto, otro valor: error

Parámetro	Significado
<i>informe</i>	Nombre del archivo que contiene el informe con path y con extensión
<i>archivo</i>	Nombre del archivo destino con path y con extensión
<i>parametros</i>	0 para tomar los parámetros por defecto y 1 para preguntarlos
<i>formato</i>	0 para formato ASCII, 1 para formato RTF y 2 para formato PDF
<i>rellenar_blanco</i>	0 para no rellenarlos y 1 para rellenarlos
<i>imprimir_cabecera</i>	0, nunca; 1, primera página y 2, siempre
<i>separador_campos</i>	0, ninguno; 1, espacio en blanco; 2, tabulador; 3, coma; 4, punto y coma
<i>imprimir</i>	0 para imprimir todo el documento y 1 para imprimir un rango de páginas
<i>pagina_inicial</i>	Número de página de comienzo de la exportación
<i>pagina_final</i>	Número de página de fin de la exportación
<i>columnas</i>	Número de caracteres de cada línea. En formato PDF es la anchura (mm)
<i>lineas</i>	Número de líneas por hoja. En formato PDF es la altura (mm)
<i>añadir</i>	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

Ejemplo

Se exporta el informe "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 exportan las páginas 1 a 5.

```
Public Sub Exportar ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.ExportRPT ("c:\informes\Presupuesto y mediciones.PrestoReport",  
"c:\informes\Presupuesto y mediciones.rft", 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 Integer
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

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
    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 Integer
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

Procedimiento que nos posiciona en el primer registro de la tabla Conceptos.

```
Public Sub posicionar_Primer ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    If po.FindFirst ("Conceptos") = 0 Then
        MsgBox ("Situado en el primer registro de Conceptos")
    End If
End Sub
```

Función	¡Error! La autoreferencia al marcador no es válida.
Sintaxis	Function FindGreat (tabla As String, clave As String, valor) As Integer
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

Se posiciona en el primer registro de la tabla Conceptos.

```
Public Sub posicionar_Mayor (valor As String)
    Dim po As Object
    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 Integer
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

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
    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 Integer
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

Se posiciona en el último registro de la tabla Conceptos.

```
Public Sub posicionar_Ultimo ()
    Dim po As Object
    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 Integer
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

Se posiciona en el registro menor o igual de la tabla Conceptos

```
Public Sub posicionar_MenorIgual (valor As String)
    Dim po As Object
    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 Integer
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

Ejem

plo

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
    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 Integer) As Integer
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 Procedimiento que primero obtiene la posición del primer registro de tipo "Partida", posteriormente continúa 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 Integer
    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
    End While
    po.FindPos ("Conceptos", "Conceptos.Código", Num_Registro)
End Sub
```

Función	FindPrev
Sintaxis	Function FindPrev (tabla As String, [clave As String]) As Integer
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

Se 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
    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 Integer, [anida As Integer]) As Integer
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

Se 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.

```
Public Sub Obtener_Elementos ()
    Dim po As Object
    Dim Num_Registros As Integer
    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
    End While
    MsgBox ("Número de registros de tipo capítulo: " & CStr (Num_Registros))
End Sub
```

Función	GetField
Sintaxis	Function GetField (campo As String, [dimension As Integer])
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

Se muestran al usuario todos los Conceptos.Pres

```
Public Sub Leer_Pres ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.SetElement (1, "Conceptos")  
    While po.GetElement (1) = 0  
        MsgBox (CStr (po.GetField ("Conceptos.Pres")))  
    End While  
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

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\"
    Dim po As Object
    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 ()
    End While
End Sub
```

Función	GetFieldNum
Sintaxis	Function GetFieldNum (campo As String, [dimension As Integer]) 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

Se muestra al usuario todos los Conceptos.Pres

```
Public Sub Leer_Num ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        MsgBox (CStr (po.GetFieldNum ("Conceptos.Pres")))
    End While
End Sub
```

Función	GetFieldStr
Sintaxis	Function GetFieldStr (campo As String, [dimension As Integer]) As String
Descripción	Devuelve el valor de un campo de tipo cadena. Si el campo es de tipo "fecha" devuelve una cadena en formato "AAAAMMDD"
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

Procedimiento que muestra al usuario todos los códigos de Conceptos

```
Public Sub Leer_Str ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        MsgBox (po.GetFieldStr ("Conceptos.Código"))
    End While
End Sub
```

Función	GetPos
Sintaxis	Function GetPos (tabla As String) As Integer
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

Se indica al usuario la posición del primer registro de tipo capítulo.

```
Public Sub posicion_Registro ()
    Dim po As Object
    Dim Num_Registro As Integer
    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 Integer, tabla As String) As Integer
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 recorreremos 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

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.

```
Public Sub Ver_Seleccion ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    Dim first As Integer
    first = 1
    While po.GetSelection (first, "Conceptos") = 0
        first = 0
        MsgBox ("Código: " & po.GetFieldStr ("Conceptos.Código") & vbCrLf &
"Resumen: " & po.GetFieldStr ("Conceptos.Resumen"))
    End While
End Sub
```

Función	GetText
Sintaxis	Function GetText (campo As String, [tipo As Integer = 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

Se obtiene la variable Conceptos.Texto de todos los registros.

```
Public Sub Leer_Memo ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        MsgBox (po.GetText ("Conceptos.Texto"))
    End While
End Sub
```

Función	ImportCad
Sintaxis	Function ImportCad (proyecto As String, archivo As String, tipo As Integer) As Integer
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

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
    po = CreateObject ("Presto.App.18")
    po.ImportCad (obra, archivo, 0)
End Sub
```

Función	ImportFIEBDC
Sintaxis	Function ImportFIEBDC (archivo As String) As Integer
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

Procedimiento que importa una obra recibida como parámetro en formato FIEBDC

```
Public Sub Importa_FIEBDC (archivo As String)
    Dim po As Object
    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 Integer
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

Se obtiene la información 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
    po = CreateObject ("Presto.App.18")
    po.InfoCode (obra, codigo, unidad, resumen, precio)
End Sub
```

Función	InitRecord
Sintaxis	Function InitRecord (tabla As String) As Integer
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`

Ejemplo

Se suprimen las variables declaradas con anterioridad

```
Public Sub Inicializa_Variables ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.InitVar ()  
End Sub
```

Función	InsertAttachment
Sintaxis	Function InsertAttachment (tableName As String, code As String, attachDescription As String, attachExtension As String, attachBuffer As Variant, attachType As Integer, resetSourcePath As Integer) As String
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><i>tableName</i>, nombre de la tabla a la que pertenece el código al cual queremos asociar el archivo: "Conceptos", "Pedidos", "Entregas", "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

Se inserta un archivo asociado a un código determinado

```
Public Sub Inserta_Asociado (codigo, buffer)
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.InsertAttachment ("Conceptos", codigo, "Prueba", "TXT", buffer)
End Sub
```


Función	<code>InsertAttachmentFromPath</code>
Sintaxis	Function <code>InsertAttachmentFromPath</code> (<code>tableName</code> As String, <code>code</code> As String, <code>attachFilePath</code> As String, <code>resetSourcePath</code> As Integer) As String
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><i>tableName</i>, nombre de la tabla a la que pertenece el código al cual queremos asociar el archivo: "Conceptos", "Pedidos", "Entregas", "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>
Retorno	Devuelve el <i>guid</i> del archivo asociado creado o cadena vacía en caso de error

Ejemplo

Procedimiento que inserta un archivo asociado a un código

```
Public Sub Inserta_Asociado (codigo As String, ruta As String)
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.InsertAttachmentFromPath ("Conceptos", codigo, ruta)
End Sub
```

Función	<code>InsertAttachmentFromUrl</code>
Sintaxis	<code>Function InsertAttachmentFromUrl (strTableName As String, strCode As String, strFile As String, strExtension as String, strUrl As String, nType as integer=0, bResetSourcePath as Boolean =False) As String</code>
Descripción	Inserta un archivo en Presto a partir de una Url
Parámetros	
Retorno	

Ejemplo

Procedimiento que inserta un archivo asociado a un código

```
Public Sub Inserta_Url ()  
    .InsertAttachmentFromUrl ("Conceptos", "000", "C:\Temp\1.rtf", "rtf",  
"www.api.com", 0, False)  
End Sub
```

Función	InsertRecord
Sintaxis	Function InsertRecord (tabla As String) As Integer
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

Se inserta un registro en la tabla que se recibe como parámetro

```
Public Sub Inserta_Registro (archivo As String)
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.InsertRecord (archivo)
End Sub
```

Atributo	IULanguage
Sintaxis	Property IULanguage As Integer
Descripción	Idioma del interfaz de Presto en formato número
Retorno	0: castellano 1: catalán 2: inglés 3: francés 4: gallego 5: portugués 6: alemán

Ejemplo

En este ejemplo se muestra el idioma del interfaz del programa y de la obra abierta

```
po = GetObject ("", "Presto.App.18")
MsgBox ("FIN P20_idioma.vbs" & Chr (13) & "Idioma IU: " & po.IULanguage & Chr (13) & "Idioma obra: " & po.ProjectLanguage)
```

Función	LogBegin
Sintaxis	Function LogBegin (logHead As String) As Integer
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, 1: error

Ejemplo

Ver ejemplo en función LogMsg

Función	LogEnd
Sintaxis	Function LogEnd () As Integer
Descripción	Cierra el proceso de log en Presto y muestra la ventana con todos los mensajes agregados
Parámetros	
Retorno	0: correcto, 1: error

Ejemplo

Ver ejemplo en función LogMsg

Función	LogMsg
Sintaxis	Function LogMsg (logMsg As String, mode As Integer) As Integer
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

```

Sub msj ()
    Dim po As Object
    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 += 1
        End While
        po.LogEnd ()
    End If
End Sub

```

Función	New
Sintaxis	Function New (obra As String) As Integer
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

Se arranca una instancia de Presto y crea una nueva obra en el path que se recibe como parámetro. Se indica si se creó correctamente.

Agregue la referencia de entre los tipos COM, "Presto 18.0 Type Library" en su compilador.

```
Public Sub Nueva_Obra (path As String)
    Dim po As PrestoLib.PrestoApplication
    po = CreateObject ("Presto.App.18")
    If po.New (path) = 0 Then
        MsgBox ("Obra creada correctamente")
    Else
        MsgBox ("No se creó correctamente la obra")
    End If
End Sub
```

Función	Open
Sintaxis	Function Open (obra As String, [solo_lectura As Integer]) As Integer
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 / 1 abre la obra en modo solo lectura / 0 abre la obra en modo escritura
Retorno	0: correcto, Otro valor: error

Ejemplo

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 Integer], [copias As Integer], [impresora As Integer]) As Integer
Descripción	Imprime un informe
Parámetros	informe es el nombre del informe con path y con extensión parámetros: 0 para tomar los de defecto y 1 para preguntarlos, copias: número a imprimir impresora: 0 para tomar la impresora por defecto y 1 para pedirla
Retorno	0 correcto, Otro valor: error

Ejemplo

Imprime el informe de la ruta "c:\informes\ejemplo.PrestoReport", con los parámetros por defecto, preguntando la impresora y realizando 3 copias

```
Public Sub Informe ()
    Dim po As PrestoLib.PrestoApplication
    po = CreateObject ("Presto.App.18")
    po.PrintRPT ("C:\informes\ejemplo.PrestoReport", 0, 3, 1)
End Sub
```

Atributo	Project
Sintaxis	Property Project As String
Descripción	Devuelve el nombre de path de la obra de Presto abierta

Ejemplo

Procedimiento que visualiza el valor de Project

```
Public Sub Ver_Project ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    MsgBox (po.Project)
End Sub
```

Atributo	ProjectLanguage
Sintaxis	Property ProjectLanguage As Integer
Descripción	Idioma de la obra de Presto en formato número
Retorno	0: castellano 1: catalán 2: inglés 3: francés 4: gallego 5: portugués 6: alemán
Ejemplo	Ver ejemplo en atributo IULanguage

Función	Quit
Sintaxis	Sub Quit ()
Descripción	Cierra la aplicación de Presto
Parámetros	Ninguno
Retorno	Ninguno

Ejemplo

Procedimiento que cierra la aplicación Presto.

```
Public Sub Cierra_Presto ()  
    Dim po As Object  
    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 Integer`

Descripción `Devuelve si la obra está en modo "solo lectura"`

Ejemplo

Visualiza si la obra está en "sólo lectura"

```
Public Sub Ver_SoloLectura ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.Open ("C:\Temp\Obra.Presto")  
    If (po.ReadOnly) Then  
        MsgBox ("Solo lectura")  
    Else  
        MsgBox ("Lectura y escritura")  
    End If  
End Sub
```

Función	Recalculate
Sintaxis	Sub Recalculate (activar As Integer)
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, llame a la función de nuevo para hacerlos todos de una vez.
Parámetros	0: Desactiva, 1: Activa de nuevo y recalcula
Retorno	Ninguno

Ejemplo

Procedimiento que bloquea los recálculos de Presto

```
Public Sub Bloquea_Calculos ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.Recalculate (0)
End Sub
```

Función	Rename
Sintaxis	Function Rename (campo As String, valor As String) As Integer
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

Procedimiento que renombra todos los registros añadiéndoles una X al final de su código

```
Public Sub Renombrar ()
    Dim po As PrestoLib.PrestoApplication
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        po.Rename ("Conceptos.Código", Trim (po.GetFieldStr ("Conceptos.Código")) & "X")
    End While
End Sub
```

Función	SelCode
Sintaxis	Function SelCode (HWND As Integer, obra As String, relacion As String) As Integer
Descripción	Elige un concepto de Presto
Parámetros	<p>HWND es la ventana que se tomará como padre del diálogo a mostrar</p> <p>obra contiene la ruta de la obra sobre la que se buscará el concepto</p> <p>relacion indica la relación donde se situará el diálogo</p>
Retorno	<p>1: se seleccionó un código válido, el código devuelto está en blanco o se canceló</p> <p>-1: si no hay una obra abierta o no se puede abrir la obra pasada a la función</p> <p>Otros: error</p>

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
    po = CreateObject ("Presto.App.18")
    po.SelCode (0, obra, relacion)
End Sub
```

Función	SetCharacter
Sintaxis	Sub SetCharacter (modo As Integer)
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 o 1 y es el juego de caracteres a utilizar
Retorno	Ninguno

Ejemplo

Procedimiento que pone como juego de caracteres el ANSI

```
Public Sub Caracter_ANSI ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.SetCharacter (0)  
End Sub
```


Función	SetConceptVarValue
Sintaxis	Function SetConceptVarValue (varGuid As String, conceptCode As String, varValue As Variant, valType As Integer, valUserTxt As String, valUserNum As Double, revitparamElementId As String, onlyNew As Integer) As Integer
Descripción	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".
Parámetros	<p><i>varGuid</i>, identificador único global de la variable.</p> <p><i>conceptCode</i>, código del concepto donde recibe valor la variable.</p> <p><i>varValue</i>, valor que se asociará a la variable.</p> <p><i>valType</i> [0], tipo del valor (no está relacionado 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).</p> <p><i>valUserTxt</i> [""], texto de usuario (nota) asociado al valor.</p> <p><i>valUserNum</i> [0], valor numérico de usuario asociado al valor.</p> <p><i>revitparamElementId</i> [""], dato proveniente de Revit.</p> <p><i>onlyNew</i> [0], con valor distinto de 0 si el valor ya existía no se actualizará.</p>
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 Integer
Descripción	Establece como certificación actual aprobada la fecha recibida.
Parámetros	Fecha de la certificación en formato yyymmdd. Tiene que existir previamente.
Retorno	0: correcto, otro valor: error

Ejemplo

Procedimiento que abre una obra y pone como fecha de certificación actual el 4 de mayo de 2021 y recalcula la obra.

```
Public Sub ponCertificacion ()
    Dim po As New PrestoLib.PrestoApplication
    Dim ret As New Integer
    po.Open ("C:\Temp\Presupuesto y mediciones.Presto")
    ret = po.SetCurrentInvPeriod ("20210504")
    If (ret = 0) Then po.Recalculate (1)
End Sub
```

Función	SetElement
Sintaxis	Function SetElement (elemento As Integer, tabla As String, [clave As String], [mascara As String], [seleccion As String], [ordenar_por As String], [orden As Integer]) As Integer
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. <i>tabla</i> : 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. <i>clave</i> establece el orden de lectura y se elige entre las posibilidades que aparecen en el campo "Clave" de la caja de diálogo anterior. <i>máscara</i> es una expresión de selección que se aplica al campo clave <i>seleccion</i> es una expresión que se calcula para filtrar elementos de una tabla <i>ordenar_por</i> es una expresión que sirve para ordenar la tabla <i>orden</i> indica 0 ascendente, 1 descendente
Retorno	0: correcto, otro valor: error

Ejemplo

Procedimiento que arranca una instancia de Presto y abre la tabla de "Conceptos", obteniendo solo los registros de tipo "Capítulo".

```
Public Sub Abrir_Conceptos ()
    Dim po As Object
    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 Integer])
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

Procedimiento que inicializa el valor del Resumen con el valor de Conceptos. Código correspondiente al registro

```
Public Sub poner_Str ()
    Dim po As Object
    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")
    End While
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

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")
    End While
End Sub
```

Función	SetModal
Sintaxis	Sub SetModal (activar As Integer)
Descripción	Activa y desactiva 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

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	SetPriceVarValue
Sintaxis	Sub SetPricetVarValue (varGuid As String, priceSupCode As String, priceInfCode 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 <i>guid</i> en la relación entre el concepto y la entidad de asignación de precios con los códigos indicados. Tanto la relación 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	<p><i>varGuid</i>, identificador único global de la variable.</p> <p><i>priceSupCode</i>, código del concepto de la relación donde recibe valor la variable.</p> <p><i>priceInfCode</i>, código de la entidad de asignación de precios de la relación donde recibe valor la variable.</p> <p><i>varValue</i>, valor que se asociará a la variable.</p> <p><i>valType</i> [0], tipo del valor (no está relacionado 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).</p> <p><i>valUserTxt</i> [""], texto de usuario (nota) asociado al valor.</p> <p><i>valUserNum</i> [0], valor numérico de usuario asociado al valor.</p> <p><i>revitparamElementId</i> [""], dato proveniente de Revit.</p>
Retorno	0: correcto, otro valor: error

Ejemplo

Ver ejemplo en SetRelationVarValue

Función	SetRelationVarValue
Sintaxis	Sub SetRelationVarValue (varGuid As String, relSupCode As String, relInfCode 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 relación entre conceptos con los códigos indicados. Tanto la relación 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	<p><i>varGuid</i>, identificador único global de la variable.</p> <p><i>relSupCode</i>, código del concepto superior de la relación donde recibe valor la variable.</p> <p><i>relInfCode</i>, código del concepto inferior de la relación donde recibe valor la variable.</p> <p><i>varValue</i>, valor que se asociará a la variable.</p> <p><i>valType</i> [0], tipo del valor (no está relacionado 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).</p> <p><i>valUserTxt</i> [""], texto de usuario (nota) asociado al valor.</p> <p><i>valUserNum</i> [0], valor numérico de usuario asociado al valor.</p> <p><i>revitparamElementId</i> [""], dato proveniente de Revit</p>
Retorno	0: correcto, otro valor: error

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")
' Precios
po.InitRecord ("Conceptos")
po.SetField ("Conceptos.Código", "PresUSD")
po.SetField ("Conceptos.Nat", 201)
po.SetField ("Conceptos.Resumen", "US dollar")
po.InsertRecord ("Conceptos")
po.InitRecord ("Precios")
po.SetField ("Precios.Código", "PARTIDA_VAR")
po.SetField ("Precios.Entidad", "PresUSD")
po.InsertRecord ("Precios")
' Variables y valores
Dim guid_var
' Variable de tipo cadena
guid_var = po.CreateVariable ("GUID_VAR_01", "VAR_01", "Variable tipo cadena creada desde VBS", 0, 6, 0, 0, 0, 0, 0, "", "", 0, 0)
po.SetRelationVarValue (guid_var, "CAPITULO_VAR", "PARTIDA_VAR", "Valor en relaciones", 0, "Texto de usuario 1", 1)
po.SetPriceVarValue (guid_var, "PARTIDA_VAR", "PresUSD", "Valor en precios", 0, "Texto de usuario 2", 2)
' Variable de tipo real
guid_var = po.CreateVariable ("GUID_VAR_02", "VAR_02", "Variable tipo real creada desde VBS", 0, 5, 0, 0, 5, 0, 2, 10, 7, 0, "", "", 0, 0, 0, "", "", "", "", "", "", "", 2)
po.SetRelationVarValue (guid_var, "CAPITULO_VAR", "PARTIDA_VAR", 5.5, 0, "Texto de usuario 3", 3)
po.SetPriceVarValue (guid_var, "PARTIDA_VAR", "PresUSD", 7.2, 0, "Texto de usuario 4", 4)
' Variable de tipo fecha
guid_var = po.CreateVariable ("GUID_VAR_03", "VAR_03", "Variable tipo fecha creada desde VBS", 0, 7, 0, 0, 0, 0, 0, "", "", 0, 0)
po.SetRelationVarValue (guid_var, "CAPITULO_VAR", "PARTIDA_VAR", "20200112", 0, "Texto de usuario 5", 5)
po.SetPriceVarValue (guid_var, "PARTIDA_VAR", "PresUSD", "20200112", 0, "Texto de usuario 6", 6)
po.EndRedo
po.SetUpdateScreen (1)
po.UpdateScreen
MsgBox "FIN inserta_variables_valores !"

```

Función	SetTakeoffInsertMode
Sintaxis	Sub SetTakeoffInsertMode (modo As Integer)
Descripción	Permite alternar entre la inserción estándar de registros en mediciones mediante "InsertRecord" y la inserción en mediciones temporales, si falla la inserción en mediciones.
Parámetros	<p>0: cuando se llame a "InsertRecord" sobre "Mediciones" se hará una inserción estándar en mediciones.</p> <p>1: cuando se llame a "InsertRecord" sobre "Mediciones" se intentará primero una inserción en mediciones y, si ésta fallase, se intentaría insertar la línea en mediciones temporales.</p> <p>2: cuando se llame a "InsertRecord" sobre "MedicionesTmp" se hará una inserción estándar en mediciones temporales.</p> <p>3: cuando se llame a "InsertRecord" sobre "MedicionesTmp" se hará una inserción en mediciones temporales con borrado previo de la tabla.</p>
Retorno	Ninguno

Ejemplo

```
Set po = GetObject ("", "Presto.App.18")
```

```
po.SetUpdateScreen (0)
po.BeginRedo
po.SetTakeoffInsertMode (1)
```

' Relaciones y conceptos

```
po.InitRecord ("Conceptos")
po.SetField ("Conceptos.Código", "CAPITULO_MED")
po.SetField ("Conceptos.Nat", 4)
po.InsertRecord ("Conceptos")
```

```
po.InitRecord ( ("Relaciones")
po.SetField ("Relaciones.CodSup", "0" )
po.SetField ("Relaciones.CodInf", "CAPITULO_MED")
po.InsertRecord ("Relaciones")
```

```
po.InitRecord ("Conceptos")
po.SetField ("Conceptos.Código", "PARTIDA_MED" )
po.SetField ("Conceptos.Nat", 5 )
po.InsertRecord ("Conceptos")
```

```
po.InitRecord ("Relaciones")
po.SetField ("Relaciones.CodSup", "CAPITULO_MED" )
po.SetField ("Relaciones.CodInf", "PARTIDA_MED") po.InsertRecord ("Relaciones")
```

' Mediciones

' Cantidad alzada

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_01")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.Cantidad", 100)
po.InsertRecord ("Mediciones")
```

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_01")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.Cantidad", 99)
po.InsertRecord ("Mediciones")
```

' Dimensiones

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_02")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 100)
po.InsertRecord ("Mediciones")
```

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_02")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 99)
po.InsertRecord ("Mediciones")
```

' Fórmula matemática

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_03")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 2)
po.SetField ("Mediciones.Longitud", 2)
po.SetField ("Mediciones.Altura", 2)
po.SetField ("Mediciones.Anchura", 2)
po.SetField ("Mediciones.Fórmula", "(A+B)*(C+D)")
po.InsertRecord ("Mediciones")
```

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_03")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 2)
```

```
po.SetField ("Mediciones.Longitud", 3)
po.SetField ("Mediciones.Altura", 2)
po.SetField ("Mediciones.Anchura", 2)
po.SetField ("Mediciones.Fórmula", "(A+B)*(C+D)")
po.InsertRecord ("Mediciones")
```

' Expresión de Presto

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_04")
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 5)
po.SetField ("Mediciones.Volumen", 2)
formula_med = "$[Mediciones[Guid= ""GUID_MED_03""].Longitud]$"
po.SetField ("Mediciones.Fórmula", formula_med)
po.InsertRecord ("Mediciones")
```

```
po.InitRecord ("Mediciones")
po.SetField ("Mediciones.GuidElem", "GUID_MED_04" )
po.SetField ("Mediciones.CodSup", "CAPITULO_MED")
po.SetField ("Mediciones.CodInf", "PARTIDA_MED")
po.SetField ("Mediciones.N", 6)
po.SetField ("Mediciones.Volumen", 2)
formula_med = "$[Mediciones[Guid= ""GUID_MED_03""].Longitud]$"
po.SetField ("Mediciones.Fórmula", formula_med)
po.InsertRecord ("Mediciones")
```

```
po.Calculate (3)
po.SetTakeoffInsertMode (0)
po.EndRedo
po.SetUpdateScreen (1)
po.UpdateScreen
MsgBox "FIN insertar_mediciones_existentes !"
```

Función `SetTakeoffVarValue`

Sintaxis `Function 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, onlyNew As Integer) As Integer`

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 *varGuid*, identificador único global de la variable.
takeoffGuidElement, *takeoffGuidAux*, identificador único global de la línea de medición concatenando ambos *guids*.
varValue, valor que se asociará a la variable.
valType [0], tipo del valor (no está relacionado 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.
onlyNew [0], Si es distinto de 0 si el valor ya existía no se actualizará.

Retorno 0 en caso correcto, otro valor 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")
```

' Medición

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 del que queremos modificar el texto (Conceptos.Texto, Facturas.Texto, etc.) valor es el contenido a insertar
Retorno	Ninguno

Ejemplo

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
    po = CreateObject ("Presto.App.18")
    po.SetElement (1, "Conceptos")
    While po.GetElement (1) = 0
        po.SetText ("Conceptos.Texto",
            po.GetFieldStr ("Conceptos.Código"))
    End While
End Sub
```


Función	SetUniqueVarValue
Sintaxis	Function SetUniqueVarValue (varGuid as String, varValue As Variant) As Integer
Descripción	Asigna un valor de tipo de asignación "Única" a una variable. La variable debe existir previamente. °
Parámetros	<i>varGuid</i> , identificador único global de la variable. <i>varValue</i> , valor que se asociará a la variable.
Retorno	0 en caso correcto, otro valor en caso contrario

Ejemplo

Procedimiento que asigna un valor a cuatro variables comunes de distinto tipo.

Dim po As Object

```
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_valores_unicos!")
```

Función	SetUpUpdateScreen
Sintaxis	Sub SetUpUpdateScreen (activar As Integer)
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 SetUpUpdateScreen con el parámetro 1 para reflejar todos los cambios de una sola vez.
Parámetros	0: desactiva , 1: activa
Retorno	Ninguno

Ejemplo

Procedimiento que bloquea los cambios en las ventanas de Presto

```
Public Sub Bloquea_VentanaPresto ()  
    Dim po As Object  
    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

Procedimiento que modifica una variable de Presto

```
Public Sub Set_Variable (var As String, valor As String)
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SetVar (var, valor)
End Sub
```

Función	ShowWindow
Sintaxis	Sub ShowWindow (Ventana As Integer)
Descripción	Abre y sitúa como activa la ventana principal indicada
Parámetros	0: Presupuesto 1: Árbol 2: Mediciones 3: Mediciones temporales 4: Espacios
Retorno	Ninguno

Ejemplo

Procedimiento que añade un espacio a una obra de Presto, abre la ventana principal "Espacios" y la posiciona como activa.

Dim po As Object

```
po = GetObject ("", "Presto.App.18")
po.SetUpdateScreen (0)
po.BeginRedo ()
```

' Espacio con número, código, descripción, color (rojo) y cota (10.77).

' Como el espacio es > 1, añadirá los espacios intermedios por defecto

```
po.CreateSpace (4, "ESP_COTA_10_77", "Espacio creado desde un complemento de
usuario", 255, 10.77)
```

' Añadimos un espacio a continuación del último, con su código, resumen, color (verde) y cota (0.55)

```
po.AddSpace ("ESP_COTA_0_55", "Espacio creado desde un complemento de usuario",
65280, 0.55)
```

' Mostramos la ventana de espacios

```
po.ShowWindow (4)
po.EndRedo ()
po.SetUpdateScreen (1)
po.UpdateScreen ()
```

```
MsgBox ("FIN inserta espacios !")
```

Función	SQLServerDelete
Sintaxis	Function SQLServerDelete (bstrServer As String, bstrUser As String, bstrPassword As String, nPort As Integer, nTimeout As Integer, bstrDataBase As String, bstrGuid As String) As Integer
Descripción	Borra la obra en SQL Server
Parámetros	<p><i>bstrServer</i>, identificador del servidor donde está alojado el servidor SQL. Mismo valor que "Nombre del servidor" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrUser</i>, usuario que se utilizará para acceder al servidor SQL. Mismo valor que "Usuario" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrPassword</i>, contraseña del usuario para acceder al servidor SQL. Mismo valor que "Contraseña" en Presto al configurar la conexión con SQL Server</p> <p><i>nPort</i>, puerto por el que se va a acceder al servidor SQL. Mismo valor que "Puerto" en Presto al configurar la conexión con SQL Server</p> <p><i>nTimeout</i>, timeout de la conexión con SQL Server</p> <p><i>bstrDataBase</i>, nombre de la base de datos sobre la que se va a borrar una obra bstrGuid,</p> <p><i>guid</i> de la obra de Presto que se va a eliminar</p>
Retorno	0: correcto, otro valor: error

Ejemplo

```
Public Sub DeleteSQL ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SQLServerDelete ("SRVPRESTO", "", "", -1, 10000, "Presto",
        "AGL7yh6UE1P34oz1SU5M_4PY")
End Sub
```

Función	SQLServerExport
Sintaxis	Function SQLServerExport (bstrServer As String, bstrUser As String, bstrPassword As String, nPort As Integer, nTimeout As Integer, bstrDataBase As String, bOverwrite As Integer, bCalcFields As Integer, bstrSelTables As String) As Integer
Descripción	Exporta la obra a SQL Server
Parámetros	<p><i>bstrServer</i>, identificador del servidor donde está alojado el servidor SQL. Mismo valor que "Nombre del servidor" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrUser</i>, usuario que se utilizará para acceder al servidor SQL. Mismo valor que "Usuario" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrPassword</i>, contraseña del usuario para acceder al servidor SQL. Mismo valor que "Contraseña" en Presto al configurar la conexión con SQL Server</p> <p><i>nPort</i>, puerto por el que se va a acceder al servidor SQL. Mismo valor que "Puerto" en Presto al configurar la conexión con SQL Server</p> <p><i>nTimeout</i>, timeout de la conexión con SQL Server</p> <p><i>bstrDataBase</i>, nombre de la base de datos sobre la que se va a exportar</p> <p><i>bOverwrite</i>, si se sobrescribirá la obra en caso de que exista. (1 Verdadero, 0 Falso)</p> <p><i>bCalcFields</i>, Si se exportarán campos calculados. (1 Verdadero, 0 Falso)</p> <p><i>bstrSelTables</i>, tablas de Presto que se van a exportar. Las tablas irán separadas con ";", en el caso de querer exportar todas las tablas, habría que dejar este campo vacío ("")</p>
Retorno	0: correcto, otro valor: error

Ejemplo

```
Public Sub ExportSQL ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SQLServerExport ("SRVPRESTO", "", "", -1, 10000, "Presto", 1, 0,
"Conceptos;Relaciones;")
End Sub
```

Función	SQLServerImport
Sintaxis	Function SQLServerImport (bstrServer As String, bstrUser As String, bstrPassword As String, nPort As Integer, nTimeout As Integer, bstrDataBase As String, bstrGuid As String, bstrSelTables As String) As Integer
Descripción	Importa una obra de SQL Server
Parámetros	<p><i>bstrServer</i>, identificador del servidor donde está alojado el servidor SQL. Mismo valor que "Nombre del servidor" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrUser</i>, usuario que se utilizará para acceder al servidor SQL. Mismo valor que "Usuario" en Presto al configurar la conexión con SQL Server</p> <p><i>bstrPassword</i>, contraseña del usuario para acceder al servidor SQL. Mismo valor que "Contraseña" en Presto al configurar la conexión con SQL Server</p> <p><i>nPort</i>, puerto por el que se va a acceder al servidor SQL. Mismo valor que "Puerto" en Presto al configurar la conexión con SQL Server</p> <p><i>nTimeout</i>, timeout de la conexión con SQL Server</p> <p><i>bstrDataBase</i>, nombre de la base de datos sobre la que se va a importar</p> <p><i>bstrGuid</i>, guid de la obra de Presto que se va a importar.</p> <p><i>bstrSelTables</i>, tablas de Presto que se van a importar. Las tablas irán separadas con ";", en el caso de querer exportar todas las tablas, habría que dejar este campo vacío ("")</p>
Retorno	0: correcto, otro valor: error

Ejemplo

```
Public Sub ImportSQL ()
    Dim po As Object
    po = CreateObject ("Presto.App.18")
    po.SQLServerImport ("SRVPRESTO", "", "", -1, 10000, "Presto",
"AGL7yh6UE1P34oz1SU5M_4PY", "Conceptos;Relaciones;")
End Sub
```

Atributo	Status
Sintaxis	Property Status As Integer
Descripción	Después de una llamada a una función del gestor de archivos devuelve 0 todo correcto

Ejemplo

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 Integer
Descripción	Sub-versión de Presto en formato número
Parámetros	
Retorno	

Ejemplo

Dim po As Object

po = GetObject ("", "Presto.App.18")

Dim strMensaje

strMensaje = po.VersionStr & ", " & po.Version.Num & po.SubVersionNum

MsgBox (strMensaje)

Función	UpdateRecord
Sintaxis	Function UpdateRecord (tabla As String) As Integer
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	<i>tabla</i> es el nombre de la tabla según informes
Retorno	0 correcto, otro valor: error

Ejemplo

Procedimiento que modifica el Resumen de todos los registros de naturaleza "Partida"

```
Public Sub Modifica_Registro ()
    Dim po As Object
    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")
    End While
End Sub
```

Función	UpdateScreen
Sintaxis	Sub UpdateScreen ()
Descripción	Actualiza las ventanas
Parámetros	Ninguno
Retorno	Ninguno

Ejemplo

Se actualizan las ventanas de Presto

```
Public Sub Actualiza_Ventanas ()  
    Dim po As Object  
    po = CreateObject ("Presto.App.18")  
    po.UpdateScreen ()  
End Sub
```

Atributo	VersionNum
Sintaxis	Property VersionNum As Integer
Descripción	Versión de Presto en formato número
Ejemplo	Ver ejemplo en atributo SubVersionNum

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