

API Funciones detalladas

Definición de las funciones disponibles en el API y en el WebAPI de Presto con un ejemplo para cada función.

- Las instrucciones generales se describen en el documento "Desarrollo de complementos API y WebAPI".
- Se requiere el conocimiento de Presto y del lenguaje de programación utilizados adecuados al nivel y complejidad del complemento que se quiera desarrollar.

La sintaxis de las funciones corresponde a Visual Basic, VB, y los ejemplos son para VB y Visual Basic Scripting, VBS.

Funciones

4

AddCert	Añade una fase de certificación	4
AddSpace	Añade un espacio por código.....	5
BeginRedo	Inicia un bloque de acciones para deshacer.....	6
Calculate	Lanza los cálculos del menú "Calcular"	7
CheckProject	Comprueba la integridad de la obra.....	7
Close	Cierra una obra	7
Command	Activa cualquier opción de menú de Presto.....	8
Copy	Copia un campo en otro.....	8
CreateSpace	Crea un espacio a partir del número.....	9
CreateVariable	Crea o actualiza una variable.....	10
Delete	Borra una obra	10
DeleteRecord	Borra un registro	11
Duplicate	Duplica un registro de la tabla de conceptos	12
EndRedo	Fin de un bloque de acciones que se pueden deshacer	12
EvalNum	Evalúa una expresión de Presto y devuelve un número	12
EvalStr	Evalúa una expresión de Presto y devuelve un texto	13
ExecuteOption	Ejecuta opciones con cajas de diálogo	14
Exit	Cierra una obra y la ventana de Presto	14
ExportFIEBDC	Exporta la obra en formato FIEBDC ("BC3")	15
ExportRPT	Exporta un informe ASCII, RTF o PDF	16
FindEqual	Ir a un registro mediante una clave.....	17
FindFirst	Ir al primer registro	18
FindGreat	Ir al registro cuya clave es mayor a un valor	18
FindGreatEqual	Ir al registro cuya clave es mayor o igual a un valor.....	19
FindLast	Ir al último registro.....	19
FindLessEqual	Ir al registro cuya clave es menor o igual a un valor.....	20
FindNext	Ir al registro siguiente.....	20
FindPos	Ir a un registro por su posición.....	21
FindPrev	Ir al registro anterior	21
GetElement	Ir al 1º registro de una tabla abierta con "SetElement"	22
GetField	Valor de un campo de Presto	22
GetFieldBinary	Valor de un campo binario de Presto	23
GetPos	Obtiene la posición de un registro.....	24

GetSelection	Recorre los elementos de una selección	25
GetText	Texto de los campos ".Texto"	26
ImportCad	Importa las mediciones de una obra	26
ImportFIEBDC	Importa un archivo en formato FIEBDC (".BC3").....	27
InfoCode	Información básica de un concepto en una obra	27
InitRecord	Pone a cero un registro	28
InitVar	Suprime variables usadas en "EvalStr"o "EvalNum"	28
InsertAttachment	Asocia un archivo a partir de una variable	29
InsertAttachmentFromPath	Asocia un archivo	30
InsertAttachmentFromUrl	Asocia un archivo a partir de una URL	30
InsertRecord	Inserta un registro	31
LogBegin	Inicia la ventana de mensajes	32
LogEnd	Muestra los mensajes	33
LogMsg	Agrega un mensaje a la ventana de mensajes	33
MsgClear	Limpia la ventana de mensajes.....	33
MsgSetTabs	Establece tabuladores en la ventana de mensajes	33
MsgText	Envía texto a la ventana de mensajes	34
MsgWrapText	Ajuste de línea de la ventana de mensajes	34
New	Crea y abre una obra	34
Open	Abre una obra existente	35
PrintRPT	Imprime un informe	36
Quit	Cierra la aplicación de Presto	36
Recalculate	Bloquea o desbloquea el recálculo	37
Rename	Permite modificar el campo "Conceptos.Código"	37
SelCode	Elige una unidad de obra en un árbol	38
SetCharacter	Selecciona ANSI (Windows) o OEM (MS-DOS).	39
SetConceptVarValue	Asigna valor a una variable para un concepto	39
SetCurrentInvPeriod	Establecer la fecha de certificación actual	41
SetElement	Accede a una tabla por clave, máscara y filtro	42
SetField	Asigna valor a un campo.....	43
SetFieldBinary	Asigna valor a un campo binario.....	43
SetModal	Bloquea y desbloquea el uso de Presto	44
SetPriceVarValue	Asigna variable a un concepto y una entidad de precios ...	45
SetRelationVarValue	Asigna variable a una relación entre conceptos	45
SetTakeoffInsertMode	Modo de inserción de las líneas de medición	47
SetTakeoffVarValue	Asigna variable y valor a una línea de medición.....	49
SetText	Rellena un campo de texto	51
SetUniqueVarValue	Asigna un valor de asignación única a una variable.....	51
SetUpdateScreen	Desactiva la actualización de las ventanas	52
SetVar	Modifica una variable de asignación única	52
ShowWindow	Abre una ventana.....	53
SQLServerDelete	Borra una obra en SQL Server	54
SQLServerExport	Exporta la obra a SQL Server	55
SQLServerImport	Importa una obra de SQL Server	55
UpdateRecord	Actualiza los cambios en un registro	56
UpdateScreen	Actualiza la información de las ventanas	56

Atributos (Properties) 58

Cajas de diálogo (WebAPI) 59

9901	DLG_REDUCIR_NIVELES	60
13351	DLG_MULTIPLICAR_PRECIOS	60
11901	DLG_AJUSTAR_PRECIO	61
13721	DLG_FUSIONAR_PARTIDAS	61
7051	DLG_COPIAR_CERT	61
10751	DLG_DESGLOSAR_PARTIDAS_DES	62
9801	DLG_ELIMINAR_DESCOMPOSICION_OBJ	62
7251	DLG_ELIMINAR_DESCOMPOSICION_OBJ	62
10801	DLG_GENERAR_OBJ	63
8151	DLG_GENERAR_TERMINOS	63
8851	DLG_CREAR_FECHAS.....	64
11951	DLG_OPERAR_GRAFICOS.....	64
15271	DLG_CONVERTIR_PRES_PLAN	64
8561	DLG_REESTRUCTURAR_LINEAS_MED.....	65
11701	DLG_RELLENAR_PLANIFICACION	65
7101	DLG_CREAR_FACTURA_MEDICION	66

Funciones

AddCert **Añade una fase de certificación**

Function **AddCert ([Resumen As String], [ColorRGB As Integer])**

ColorRGB Por defecto se crea un color aleatorio

Retorno El número de la fase agregada o 0 en caso de error

Crea una nueva fase de certificación a continuación de la última. Si no hubiera ninguna crea la primera.

El número de la fase y la fecha se calculan internamente en base al criterio de defecto (una certificación por mes).

Ejemplo VB

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

Ejemplo VBS

```
Obra.AddCert "Existente", 16711680 ' Azul
Obra.AddCert "Fase 1" ' Color elegido por Presto
Obra.AddCert "Fase 2"
Obra.AddCert "Fase 3"
Fase = Obra.AddCert ("Análisis", 0) ' Para recuperar la respuesta
Obra.LogMsg Fase, 0
```

AddSpace

Añade un espacio por código

Function	AddSpace ([Código As String], [Resumen As String], [ColorRGB As Integer], [Cota as Double]) As Integer
Código	Se recomienda usar la altura de la planta respecto del suelo en cm de la forma "0825" o "Spc0825"
ColorRGB	El valor 0 crea un color aleatorio
Cota	Altura de la planta respecto del suelo en m
Retorno	Número del espacio añadido o 0 si no se pudo crear

Agrega un concepto de tipo espacio con los parámetros indicados y le asigna el siguiente número de espacio válido, en el orden establecido por Presto

Ejemplo VB

Dim Obra As Object

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

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

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

Obra.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

Obra.CreateSpace (6) ' Espacios incorrectos

Obra.CreateSpace (6)

Obra.CreateSpace (7, "ESP_04_USR")

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

Obra.AddSpace ("ESP_ADD_USR", "Espacio creado desde un complemento de usuario", 65280)

' Añadimos otro espacio sin código ni resumen, pero con color (azul)

Obra.AddSpace ("", "", 16711680) ' Espacio añadido incorrectamente

Obra.AddSpace ("ESP_ADD_USR")

Ejemplo VBS

Obra.AddSpace "Spc0000", "Planta baja",255,0

Espacio = Obra.**AddSpace** ("Spc0340", "Planta primera", , 340.0)

If Espacio = 0 Then

 Obra.LogMsg "No se pudo crear el espacio", 2

Else

 Obra.LogMsg Espacio, 0

End If

BeginRedo

Inicia un bloque de acciones para deshacer

Function **BeginRedo () As Integer**

Retorno 0: correcto, otro valor: error

Las instrucciones "BeginRedo " y "EndRedo" deben ejecutarse en pareja, es decir, o las dos o ninguna. Ver "SetModal".

Para impedir que si el programa se interrumpe se ejecute solo la primera, se puede situar justo antes de modificar la obra y comprobar después antes de ejecutar la segunda, como en el ejemplo VBS.

Ejemplo VB

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

Ejemplo VBS

```
Set Obra = GetObject (, "Presto.App.18")
PrimeraVez = True ' Para lanzar más adelante el inicio de Deshacer
Obra.SetElement 1, "Conceptos","Conceptos.Código"
While Obra.GetElement(1) = 0
    Resumen = Obra.GetField ("Conceptos.Resumen")
    If PrimeraVez = True Then ' Lanza el inicio de Deshacer cuando es necesario
        Obra.BeginRedo
        PrimeraVez = False
    End If
    Obra.SetField "Conceptos.Resumen", UCase(Resumen) ' Cambia a mayúsculas
    Obra.UpdateRecord ("Conceptos")
Wend
If PrimeraVez = False Then ' Lanza el fin del deshacer si se lanzó el inicio
    Obra.EndRedo
    Obra.UpdateScreen
End If
```

Calculate Lanza los cálculos del menú “Calcular”

Sub Calculate (Modo As Integer)

Modo 1: Recalcular, 2: Precios, 3: Mediciones, 4: Tiempos, 5: Documentos, 6: Costes reales

Los cálculos dependen del estado del cálculo automático y la modulación accesible para el usuario.

Ejemplo VBS

Obra.Calculate 1

CheckProject Comprueba la integridad de la obra

Sub CheckProject ()

No disponible en WebAPI

Ejemplo VB

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

End Sub

Ejemplo VBS

Obra.CheckProject

Close Cierra una obra

Sub Close ()

Ver los ejemplos en “New”

Command Activa cualquier opción de menú de Presto

Sub Command (Opción As String)

Sólo funciona en opciones que requieren el posicionamiento, la ventana activa o una selección, y no actúa en WebAPI

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 ";".

Ejemplo VB

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

End Sub

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

End Sub

Ejemplo VBS

```
Obra.Command ("Tablas|Ajustar anchura")  
Obra.Command ("Presupuesto|Árbol;Conceptos") ' Abre o cierra las ventanas"  
Obra.Command ("Excel") ' Opción "Importar: Excel"  
Obra.Command ("Exportar|Excel") ' Opción "Exportar: Excel" con opciones de defecto
```

Copy Copia un campo en otro

Sub Copy (Origen As String, Destino As String, [DimensionOrigen As Integer], [DimensionDestino As Integer])

Dimension Índice en campos que permiten acceder a varias estructuras de precios.
Ver "GetField"

Ejemplo VB

```
Public Sub Copiar ()  
    Dim Obra As Object  
    Obra = CreateObject ("Presto.App.18")
```



```

Obra.SetElement (1, "Conceptos")
While Obra.GetElement (1) = 0
    Obra.Copy ("Conceptos.Pres", "Conceptos.Pres",0, 2)
    Obra.UpdateRecord ("Conceptos")
End While
End Sub

```

CreateSpace Crea un espacio a partir del número

Function CreateSpace (Número As Integer, [Código As String], [Resumen As String], [ColorRGB As Integer], [Cota as Double])

Número Número de espacio a crear entre 1 y 9999.
Se crean todos los espacios desde el número anterior para que no queden huecos

Retorno 0: se ha creado el espacio

Se diferencia de "AddSpace" en que "CreateSpace" se basa en el número y crea espacios para los números intermedios, si existen.

Ver más ejemplos en AddSpace.

Ejemplo VBS

```

Sub CreateSpace ' Crea el siguiente número de espacio
For i = 1 to 100
    Codigo = "Spc_" & Right("000" & i, 3)
                ' Numero, Código, Resumen, Color, Nivel
    Creado = Obra.CreateSpace (i, Codigo, "Planta baja",0)
    If Creado = 0 Then Exit for
Next
Obra.UpdateScreen
Obra.LogMsg "Número de espacio: "& i, 2
End Sub

```


Ejemplo VB

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

Ejemplo VBS

```
Sub Delete
    Error = Obra.Delete ("C:\TEMP\EJEMPLO.Presto")
    If Error = 0 Then Obra.LogMsg "Obra borrada", 0
End sub
```

DeleteRecord Borra un registro

Function DeleteRecord (Tabla As String, [Todo As Integer]) As Integer

Todo 1: Para borrar todos los registros relacionados

Retorno 0: correcto, otro valor: error

Mantiene la integridad relacional de la obra.

No se puede borrar el registro activo. Requiere estar posicionado en el registro como se ve en el ejemplo "FindEqual".

Ejemplo VB

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

Duplicate Duplica un registro de la tabla de conceptos

Function Duplicate (Código As String, CódigoNuevo As String) As Integer

Retorno 0: correcto, otro valor: error

Actúa 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 e insertarlo de nuevo con otra clave.

Ejemplo VB

```
Public Sub Duplicar ()  
    Dim Obra As Object  
    Obra = CreateObject ("Presto.App.18")  
    Obra.Duplicate ("C01", "C01B")  
End Sub
```

Ejemplo VBS

```
For i = 1 to 9  
    Error = Obra.Duplicate ("Maq0000", "Maq000"&i)  
    If Error = 0 Then Obra.LogMsg Error & " Duplicado concepto: Maq000"&i, 0  
Next
```

EndRedo Fin de un bloque de acciones que se pueden deshacer

Sub EndRedo ()

Ver "BeginRedo"

EvalNum Evalúa una expresión de Presto y devuelve un número

Function EvalNum (Expresión As String) As Double

Retorno Número resultante de la evaluación de la expresión

La expresión debe enviarse entre comillas. Las comillas dentro de la expresión deben sustituirse por dobles comillas o chr(34).

Los valores numéricos deben enviarse con punto decimal.

Antes de utilizar una expresión, pruébela creando un campo de usuario en Presto.

En campos que pueden estar anulados, como "Relaciones.CanPres", "EvalNum" devuelve 0, mientras "GetField" devuelve el valor guardado.

En los campos de dimensiones de las líneas de medición "EvalNum" devuelve el número guardado y hay que leer también el campo "Mediciones.Nul*" que corresponda.

Ejemplo VB

```
Public Sub Declara_Numérico (variable As String)
    Dim Obra As Object
    'variable = "DecCantRend"
    Obra = CreateObject ("Presto.App.18")
    MsgBox (Obra.EvalNum ("workd (" & Chr (34) & variable & Chr (34) & "))")
End Sub
```

Ejemplo VBS

```
Variable = "DecCantRend"
Respuesta = Obra.EvalNum ("Obra.DecCantRend")
MsgBox Variable & ": " & Respuesta
MsgBox "Máximo: " & Obra.EvalNum ("max(1,2,3,4,5,6,7,8,9)")
```

EvalStr Evalúa una expresión de Presto y devuelve un texto

Function EvalStr (expresion As String) As String

Expresión Expresión que evalúa el generador de expresiones.

Retorno Cadena resultante de la evaluación de la expresión

Ver "EvalNum"

Ejemplo VB

```
Public Sub Declara_Cadena (variable As String)
    Dim Obra As Object
    'variable = "DivISO[1]"
    Obra = CreateObject ("Presto.App.18")
    MsgBox (Obra.EvalStr ("work (" & Chr (34) & variable & Chr (34) & "))")
End Sub
```

Ejemplo VB

```
Obra.SetElement 1, "Conceptos", "Conceptos.Nat", "Conceptos.Nat == 4"
While Obra.GetElement(1) = 0
    Pres = Obra.GetField ("Conceptos.Pres")
    If Pres Then
        Valor = Replace (Pres, ",", ".")
```

```

Expresion = "words("& Valor &", ""m"" ) + iif(frc( "& Valor &"), "" con "" + words(frc("&
Valor &") * 100, ""m""), """"")
Respuesta = Obra.EvalStr (Expresion)
MsgBox Valor &": "& Respuesta
End If
Wend

```

ExecuteOption Ejecuta opciones con cajas de diálogo

Sub ExecuteOption (DialogOptionsCode, texto en formato JSON)

Ejecuta opciones que tienen cajas de diálogo insertando las respuestas en un archivo ".JSON".

Ver las cajas de diálogo y la sintaxis detallada en el apartado "Cajas de diálogo".

Sólo actúa en WebAPI

Exit Cierra una obra y la ventana de Presto

Sub Exit ()

No actúa en WebAPI

Ejemplo VB

```

Public Sub Cierra_Instanceia ()
    Dim Obra As Object
    Dim OtroPresto As Object
    Obra = CreateObject ("Presto.App.18")
    OtroPresto = CreateObject ("Presto.App.18")
    MsgBox ("Se han abierto dos instancias de Presto")
    Obra.Exit ()
    MsgBox ("Se ha cerrado una instancia Presto")

```

End Sub

Ejemplo VB

```

Obra.Exit

```

ExportFIEBDC Exporta la obra en formato FIEBDC (".BC3")

Function ExportFIEBDC (Archivo As String) As Integer

Archivo Archivo de destino con camino, nombre y extensión

Retorno 0 correcto, otro valor: error

Ejemplo VB

```
Public Sub Exporta_FIEBDC (obra As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.Open (obra)
    Obra.ExportFIEBDC ("C:\TEMP\EJEMPLO.BC3")
End Sub
```

Ejemplo VBS

```
Sub ExportarFIEBDC
    Titulo = "Exportar a formato FIEBDC (.BC3)"
    Set objShell = CreateObject("Shell.Application")
    Set Directorio = objShell.BrowseForFolder (0, "Selecciona un directorio:", 0, "C:\") ' Cambia
    "C:\" al directorio inicial deseado
    If Not Directorio Is Nothing Then
        Nombre = Obra.EvalStr("_Name") ' Nombre de archivo de la obra
        Nombre = left(Nombre, Instr (Nombre, ".Presto") - 1)
        Camino = Obra.EvalStr("_Path")
        Camino = left(Camino, Instr (Camino, Nombre) - 1)
        Path = Directorio.Path
        Respuesta = MsgBox (Titulo & " la obra:" & vbCrLf & Nombre & vbCrLf & vbCrLf &
        "situada en:" & vbCrLf & Camino & vbCrLf & vbCrLf & "al directorio:" & vbCrLf & Path
        ,vbOKCancel ,Titulo)
        If Respuesta = vbOK Then
            Error = Obra.ExportFIEBDC (Path & "\" & Nombre & ".BC3")
            If Error = 0 Then
                MsgBox "Obra exportada a " & vbCrLf & Path & "\" & Nombre & ".BC3", 0
            Else
                MsgBox "Error en la exportación"
            End If
        End If
    End If
End Sub
```

ExportRPT**Exporta un informe ASCII, RTF o PDF**

Function **ExportRPT (Informe As String, Destino As String, Preguntar As Integer, Formato As Integer, Blancos As Integer, Cabecera As Integer, Separador As Integer, Rango As Integer, PagInicial As Integer, PagFinal As Integer, Columnas As Integer, Líneas As Integer, Añadir As Integer) As Integer**

Informe	Camino, nombre y extensión del informe
Destino	Archivo de destino con camino y extensión
Preguntar	0: Usa las respuestas por defecto, 1: Realiza las preguntas del informe al usuario
Formato	0: ASCII, 1: RTF, 2: PDF
Blancos	0: Rellena blancos, 1: No rellena
Cabecera	0: No la imprime, 1: Primera página, 2: Siempre
Separador	0: Ninguno, 1: Espacio, 2: Tabulador, 3: Coma, 4: Punto y coma
Rango	0: Todo el documento, 1: Rango de páginas
PagInicial	Número de la primera página que se exporta
PagFinal	Número de la última página que se exporta
Columnas	Número de caracteres de cada línea. En formato PDF es la anchura (mm)
Lineas	Número de líneas por hoja. En formato PDF es la altura (mm)
Añadir	0: Añadir el informe al contenido del archivo destino, 1: Sobrescribir En formato PDF es el tamaño de página: 0: Sin definir, 1: A0, 2: A1, 3: A2, 4: A3, 5: A4; 6: Todo el documento, 7: Usuario
Retorno	0: Correcto, Otro valor: Error

Ejemplo VB

```
Public Sub Exportar ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.ExportRPT ("c:\informes\Presupuesto y mediciones.PrestoReport",
"c:\informes\Presupuesto y mediciones.rtf", 0, 1, 0, 1, 0, 1, 1, 5, 120, 60, 0)
End Sub
```


Ejemplo VBS

Sub **ExportarRPT**

```
Informe = "C:\Program Files\Presto 2024.01\Report\01 Presupuestos y mediciones\Presupuesto y mediciones.PrestoReport"
```

```
Destino = "c:\temp\Presupuesto y mediciones.RTF"
```

```
Preguntar = 0
```

```
Formato = 1 ' ASCII/RTF/PDF
```

```
Blancos = 1
```

```
Cabecera = 1
```

```
Separador = 1
```

```
Columnas = 120
```

```
Lineas = 60
```

```
Anadir = 0
```

```
Rango = 1
```

```
PagIni = 1
```

```
PagFin = 5
```

```
Error = Obra.ExportRPT (Informe, Destino, Preguntar, Formato, Blancos, Cabecera, Separador, Rango, PagIni, PagFin, Columnas, Lineas, Anadir)
```

```
If Error = 0 then MsgBox "Obra exportada correctamente"
```

```
End Sub
```

FindEqual Ir a un registro mediante una clave

Function **FindEqual** (Tabla As String, Clave As String, Valor) As Integer

Clave	Valor de la clave buscada
-------	---------------------------

Retorno	0: correcto, otro valor: error
---------	--------------------------------

Ver conjuntamente todas las posibilidades de posicionamiento en un registro, "Find"*

Ejemplo VB

```
Public Sub posicionar_Igual (valor As String)
```

```
Dim Obra As Object
```

```
Obra = CreateObject ("Presto.App.18")
```

```
If Obra.FindEqual ("Conceptos", "Conceptos.Código", valor) = 0 Then
```

```
MsgBox ("Situado en registro con Conceptos.Código " & valor)
```

```
End If
```

```
End Sub
```

Ejemplo VBS

```
' Borrar conceptos de naturaleza "Maquinaria"
```

```
Obra.SetElement 1, "Conceptos", "Conceptos.Nat", ""7""
```

```
Dim Codigos ()
```

```
i = 0
```

```
While Obra.GetElement (1) = 0
```

```
i = i + 1
```

```

ReDim Preserve Codigos (i)
Codigos(i) = Obra.GetField ("Conceptos.Código")
Wend
If i Then
  For i = 0 to UBound(Codigos)
    If Obra.FindEqual ("Conceptos", "Conceptos.Código", Codigos(i)) = 0 Then
      Error = Obra.DeleteRecord ("Conceptos")
      If Error = 0 then Obra.LogMsg "Borrado el concepto: " & Codigos (i), 0
    End If
  Next
End If

```

FindFirst Ir al primer registro

Function FindFirst (Tabla As String, Clave As String) As Integer

Clave Clave de ordenación

Retorno 0: correcto, otro valor: error

Ejemplo VB

```

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

```

Ejemplo VBS

```

Set Obra = GetObject ("", "Presto.App.18")
If Obra.FindFirst ("Conceptos", "Nat") = 0 Then
  MsgBox ("Primer registro de conceptos por orden de naturalezas")
  MsgBox (Obra.GetField ("Conceptos.Código")
Else
  MsgBox ("Error")
End If

```

FindGreat Ir al registro cuya clave es mayor a un valor

Function FindGreat (Tabla As String, Clave As String, Valor) As Integer

Clave Valor de la clave buscada

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_Mayor (valor As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindGreat ("Conceptos", "Conceptos.Código", valor) = 0 Then
        MsgBox("Cambia el resumen del siguiente registro a " & valor)
        Obra.SetField("Conceptos.Resumen", "Nuevo resumen")
        Obra.UpdateRecord("Conceptos")
    End If
End Sub
```

FindGreatEqual Ir al registro cuya clave es mayor o igual a un valor

Function FindGreatEqual (Tabla As String, Clave As String, Valor) As Integer

Clave Valor de la clave buscada

Retorno 0: correcto, otro valor: error

Ver *“FindEqual”, “FindFirst”, “FindGreat”, “FindGreatEqual”*

Ejemplo VB

Se posiciona en el primer registro de la tabla Conceptos que cumple la clave.

```
Public Sub posicionar_MayorIgual (valor As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindGreatEqual ("Conceptos", "Conceptos.Código", valor) = 0 Then
        MsgBox ("Cambia el resumen del siguiente registro a " & valor)
        Obra.SetField("Conceptos.Resumen", "Nuevo resumen")
        Obra.UpdateRecord("Conceptos")
    End If
End Sub
```

FindLast Ir al último registro

Function FindLast (Tabla As String, Clave As String) As Integer

Clave Clave de ordenación

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_Ultimo ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindLast ("Conceptos") = 0 Then
        MsgBox ("Situado en el último registro de Conceptos")
    End If
End Sub
```

FindLessEqual Ir al registro cuya clave es menor o igual a un valor

Function FindLessEqual (Tabla As String, Clave As String, Valor As Integer)

Clave Valor de la clave buscada

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_MenorIgual (valor As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindLessEqual ("Conceptos", "Conceptos.Código", valor) = 0 Then
        MsgBox ("Situado en el anterior registro a " & valor)
    End If
End Sub
```

FindNext Ir al registro siguiente

Function FindNext (Tabla As String, Clave As String) As Integer

Clave Clave de ordenación

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_Siguiente (valor As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindNext ("Conceptos", "Conceptos.Código") = 0 Then
        MsgBox ("Situado en registro con Conceptos.Código siguiente")
    End If
End Sub
```

FindPos Ir a un registro por su posición

Function FindPos (Tabla As String, Clave As String, Posición As Integer) As Integer

Clave Clave de ordenación a la que se refiere la posición

Posición Posición buscada, obtenida previamente mediante "GetPos".

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_Registro ()
    Dim Obra As Object
    Dim Num_Registro As Integer
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos", "Conceptos.Nat", """"5""")
    If Obra.GetElement (1) = 0 Then
        Num_Registro = Obra.Getpos ("Conceptos")
    End If
    Obra.SetElement (1, "Conceptos")
    While Obra.GetElement (1) = 0
        ' Avanzamos de registro
    End While
    Obra.FindPos ("Conceptos", "Conceptos.Código", Num_Registro)
End Sub
```

FindPrev Ir al registro anterior

Function FindPrev (Tabla As String, Clave As String) As Integer

Retorno 0: correcto, otro valor: error

Ejemplo VB

```
Public Sub posicionar_Siguiente (valor As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    If Obra.FindPrev ("Conceptos", " Conceptos.Código") = 0 Then
        MsgBox ("Situado en registro con Conceptos.Código anterior")
    End If
End Sub
```

GetElement Ir al 1º registro de una tabla abierta con "SetElement"

Function **GetElement (Elemento As Integer, [Anida As Integer]) As Integer**

Elemento Número indicado por el "SetElement" con el que se abrió

Anida Indica si queremos que se posicione en los registros relacionados, de la misma manera que en los informes

Retorno 0: Quedan más elementos

Ejemplo VB

```
Public Sub Obtener_Elementos ()
    Dim Obra As Object
    Dim Num_Registros As Integer
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos", "Conceptos.Nat", """"4""")
    Num_Registros = 0
    While Obra.GetElement (1) = 0
        Num_Registros = Num_Registros + 1
    End While
    MsgBox ("Número de registros de tipo capítulo: " & CStr (Num_Registros))
End Sub
```

Ejemplo VBS

```
Obra.SetElement 1, "Conceptos", "Conceptos.Código"
While Obra.GetElement(1) = 0
    Resumen = Obra.GetField ("Conceptos.Resumen")
    Obra.SetField "Conceptos.Resumen", UCase(Resumen) ' Cambia a mayúsculas
    Obra.UpdateRecord ("Conceptos")
Wend
```

GetField Valor de un campo de Presto

Function **GetField (Campo As String, [Dimension As Integer])**

Dimensión Índice en campos que se refieren a varios estructuras de precios, como "Conceptos.Coste" o "Relaciones.CanInI". Las dimensiones en el API van de 0 a 4, mientras que en Presto van de 1 a 5
Ver "Manual: Gestión del proyecto: Seguimiento del coste: Otros valores desglosados por estados"

Retorno Devuelve el valor del campo indicado como parámetro

Existen también las funciones “GetFieldNum” y “GetFieldStr” que devuelven los campos como número y como texto, respectivamente.

Para campos que pueden estar anulados o contienen valores calculados no guardados vea “EvalNum”

Ejemplo VB

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

Ejemplo VBS

```
Resumen = Obra.GetField ("Conceptos.Coste", 1)
```

GetFieldBinary Valor de un campo binario de Presto

Function GetFieldBinary (Campo As String)

Retorno Valor del campo

Ejemplo VB

```
Public Sub Leer_Archivos ()
    Dim ArchivoByte () As Byte = Nothing
    Dim bFileStream As FileStream
    Dim ArchivosNombre, ArchivosExtensión As String
    Dim Ruta As String = "C:\Temp\"
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Archivos")
    While Obra.GetElement (1) = 0
        ArchivoByte = Obra.GetFieldBinary ("Archivos.Archivo")
        ArchivosNombre = Obra.GetFieldStr ("Archivos.Nombre")
        ArchivosExtensión = Obra.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
```

Ejemplo VBS

```
' Copia a un directorio todos los archivos asociados
```

```

Ruta = "C:\TEMP\"
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetElement 1, "Archivos" ' No guardará archivos con el mismo nombre
While Obra.GetElement (1) = 0
    Buffer = Obra.GetFieldBinary ("Archivos.Archivo")
    Nombre = Obra.GetFieldStr ("Archivos.Nombre")
    Nombre = ReemplazarCaracteresInvalidos(Nombre)
    Extension = Obra.GetFieldStr ("Archivos.Extensión")
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set stream = CreateObject("ADODB.Stream")
    stream.Type = 1 'adTypeBinary
    stream.Open
    stream.Write(Buffer)
    stream.SaveToFile (Ruta & Nombre & "." & Extension, 2) ' Sobreescribe si existe
    stream.Close
Wend
Function ReemplazarCaracteresInvalidos(nombreArchivo)
    caracteresInvalidos = Array("\", "/", ":", "*", "?", "!", " ", "<", ">", "|")
    For Each caracter In caracteresInvalidos
        nombreArchivo = Replace(nombreArchivo, caracter, "_")
    Next
    ReemplazarCaracteresInvalidos = nombreArchivo
End Function

```

GetPos

Obtiene la posición de un registro

Function GetPos (Tabla As String) As Integer

Retorno Posición del registro activo en la tabla

La posición es un número asociado por el gestor de archivos a cada registro que sirve posteriormente para posicionarse de nuevo en el mismo registro.

Ejemplo VB

```

Public Sub posicion_Registro ()
    Dim Obra As Object
    Dim Num_Registro As Integer
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos", "Conceptos.Nat", """"4""")
    If Obra.GetElement (1) = 0 Then
        Num_Registro = Obra.GetPos ("Conceptos")
        MsgBox ("posición del primer registro de tipo capítulo: " & CStr
        (Num_Registro))
    End If
End Sub

```


Ejemplo VBS

```
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetElement 1, "Conceptos", "Conceptos.Nat", """"4""""
If Obra.GetElement (1) = 0 Then
    Num_Registro = Obra.GetPos ("Conceptos")
    MsgBox "Posición del primer registro de tipo 'Capítulo': " & CStr (Num_Registro)
End If
```

GetSelection Recorre los elementos de una selección

Function GetSelection (First As Integer, Tabla As String) As Integer

First 0: Se recorren todos, 1: Primer elemento

Retorno 0: Quedan elementos en la selección

Ejemplo VB

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

Ejemplo VBS

```
' Lee y guarda los códigos de una selección
Dim Codigos ()
First = 1 ' Comprueba si hay selección y lee
NumTextos = -1
Do While Obra.GetSelection(First, "Conceptos") = 0
    First = 0
    NumTextos = NumTextos + 1
    Redim Preserve Codigos(NumTextos)
    Codigos(NumTextos) = Obra.GetField ("Conceptos.Código")
Loop
```

GetText Texto de los campos ".Texto"

Function GetText (Campo As String, [Tipo As Integer = 1]) As String

Tipo 1: Texto ASCII, 2: Texto RTF

Retorno Texto

Ejemplo VB

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

Ejemplo VBS

```
Mensaje = "1: Convierte a texto plano TXT" & vbCrLf
Mensaje = mensaje & "2: Mantiene como texto enriquecido RTF" & vbCrLf
TXTorRTF = InputBox (Mensaje, descripcion, 1 )
While Obra.GetElement(1) = 0
    Texto = Obra.GetText ("Conceptos.Texto", TXTorRTF)
    .....
Wend
```

ImportCad Importa las mediciones de una obra

Function ImportCad (Obra As String, Archivo As String, Tipo As Integer) As Integer

Obra No se usa

Archivo Ruta y nombre del archivo a importar

Tipo No se usa

Retorno 0: correcto, otro valor: error

Llama a la opción "Importar" de la ventana "Mediciones temporales". No actúa en WebAPI

Ejemplo VB

```
Public Sub Importa_CAD (Obra As String, Archivo As String)
    Dim Obra As Object
```

```
Obra = CreateObject ("Presto.App.18")
Obra.ImportCad (Obra, Archivo, 0)
```

End Sub

ImportFIEBDC Importa un archivo en formato FIEBDC (".BC3")

ImportFIEBDC (Archivo As String) As Integer

Archivo	Ruta y nombre del archivo
Retorno	0 correcto, otro valor: error

Ejemplo VB

```
Public Sub Importa_FIEBDC (Archivo As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.ImportFIEBDC (Archivo)
```

End Sub

InfoCode Información básica de un concepto en una obra

Function InfoCode (Proyecto As String, Código As String, Unidad As String, Resumen As String, Precio As String) As Integer

Proyecto	Ruta y nombre de la obra en la que se busca el concepto
Código	Código del concepto que se busca
Unidad	Devuelven la información correspondiente
Resumen	
Precio	
Retorno	0 correcto, otro valor: error

No se puede usar en VBS porque "Unidad", "Resumen" y "Precio" deben ser variables de tipo string y en VBS no es posible especificar un tipo concreto

No actúa en WebAPI

Ejemplo VB

```
Public Sub Get_Concepto (Obra As String, Codigo As String)
    Dim Unidad, Resumen, Precio As String
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.InfoCode (Obra, Codigo, Unidad, Resumen, Precio)
```

End Sub

InitRecord Pone a cero un registro

Function InitRecord (Tabla As String) As Integer

Retorno 0: correcto, otro valor: error

Si no se usa "InitRecord", al insertar un registro con "InsertRecord" se toman los datos del registro anterior en los campos que no se rellenen en el registro actual.

Ejemplo VB

```
Public Sub Inicializa_Registro (Tabla As String)
    Obra.InitRecord ("Conceptos")
    Obra.SetField ("Conceptos.Nat", 4)
    Obra.SetField ("Conceptos.Código", "E")
    Obra.SetField ("Conceptos.Resumen", "EDIFICACIÓN")
    Obra.InsertRecord ("Conceptos")
```

End Sub

InitVar Suprime variables usadas en "EvalStr"o "EvalNum"

Sub InitVar ()

Suprime variables declaradas con float o con char dentro de una expresión usada en "EvalStr" o "EvalNum"

Ejemplo

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

End Sub

Ejemplo VBS

```
Set Obra = GetObject("", "Presto.App.18")
Obra.EvalNum ("float a; a=12; return(a);")
MsgBox "Valor variable 'a': " & Obra.EvalNum ("a")
Obra.InitVar()
' La siguiente sentencia devuelve error de sintaxis porque la variable 'a' ya no está
declarada
MsgBox "Valor variable 'a': " & Obra.EvalNum ("a")
```

InsertAttachment Asocia un archivo a partir de una variable

Function	InsertAttachment (Tabla As String, Código As String, Descripción As String, Extensión As String, Buffer As Variant, [Tipo As Integer]) As String
Descripción	Valores para esos campos del archivo asociado que se va a crear
Extensión	
Buffer	Contenido del archivo
Tipo	0: Indeterminado Ver los demás tipos en la ayuda del campo "Archivos.Tipo"
Retorno	<i>Guid</i> del archivo asociado creado o cadena vacía en caso de error

Ejemplo VB

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

Ejemplo VBS

```
Sub InsertarArchivo ' Convierte los textos en archivos asociados
    Set Obra = GetObject("", "Presto.App.18")
    Obra.SetElement 1, "Conceptos", "Conceptos.Código", "*" & "", "Len(Conceptos.Texto)"
    While Obra.GetElement(1) = 0
        Codigo = Obra.GetField ("Conceptos.Código")
        Texto = Obra.GetText ("Conceptos.Texto", 2)
        Obra.InsertAttachment "Conceptos", Codigo, Codigo, "RTF", Texto, 0
    Wend
End Sub
```

InsertAttachmentFromPath Asocia un archivo

Function	InsertAttachmentFromPath (Tabla As String, Código As String, Archivo As String, Reset As Integer) As String
Archivo	Camino y ruta del archivo
Reset	Un valor distinto de 0 indica que tras insertar el archivo se borre el campo "Archivos.Camino" y se rellene la fecha actual en "Archivos.Fecha". Es útil cuando la ubicación del archivo de origen es temporal
Retorno	Devuelve el <i>guid</i> del archivo asociado o cadena vacía en caso de error

Ver "*InsertAttachment*"

Ejemplo VB

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

InsertAttachmentFromUrl Asocia un archivo a partir de una URL

Function **InsertAttachmentFromUrl (Tabla As String, Código As String, Archivo As String, Extensión As String, URL As String, Tipo as integer = 0, Reset As Boolean = False) As String**

Retorno

Ver "*InsertAttachment*" y "*InsertAttachmentFromPath*"
Sólo actúa en WebAPI

Ejemplo VB

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

InsertRecord Inserta un registro

Function InsertRecord (Tabla As String) As Integer

Retorno 0: correcto, otro valor: error

Utiliza la clave y los demás campos de la tabla que se hayan definido con "SetField". Si la clave ya existe y la tabla no admite duplicados devolverá un error

Ejemplo VB

```
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetUpdateScreen (0)
Obra.BeginRedo
' Relaciones y conceptos
Obra.InitRecord ("Conceptos")
Obra.SetField ("Conceptos.Código", "CAP_01")
Obra.SetField ("Conceptos.Nat", 4)
Obra.InsertRecord ("Conceptos")
Obra.InitRecord ("Relaciones")
Obra.SetField ("Relaciones.CodSup", "0")
Obra.SetField ("Relaciones.CodInf", "CAP_01")
Obra.InsertRecord ("Relaciones")
Obra.InitRecord ("Conceptos")
Obra.SetField ("Conceptos.Código", "PAR_0101")
Obra.SetField ("Conceptos.Nat", 5)
Obra.InsertRecord ("Conceptos")
Obra.InitRecord ("Relaciones")
Obra.SetField ("Relaciones.CodSup", "CAP_01")
Obra.SetField ("Relaciones.CodInf", "PAR_0101")
Obra.InsertRecord ("Relaciones")
' Precios
Obra.InitRecord ("Conceptos")
Obra.SetField ("Conceptos.Código", "PresUSD")
Obra.SetField ("Conceptos.Nat", 201)
Obra.SetField ("Conceptos.Resumen", "US dollar")
Obra.InsertRecord ("Conceptos")
Obra.InitRecord ("Precios")
Obra.SetField ("Precios.Código", "PAR_0101")
Obra.SetField ("Precios.Entidad", "PresUSD")
Obra.InsertRecord ("Precios")
Obra.EndRedo
Obra.SetUpdateScreen (1)
Obra.UpdateScreen
MsgBox "FIN de inserción de Conceptos y relaciones"
```

Ejemplo VBS

Sub **InsertaRegistros**

```
Obra.BeginRedo
Obra.SetUpdateScreen 0
Set Obra = GetObject ("", "Presto.App.18")
CodSup = "CAP_01"
InsertaConcepto CodSup, 4, ""
Raiz = Obra.EvalStr ("Conceptos[Nat=0].Código")
InsertaRelacion Raiz, CodSup
CodInf= "PAR_0101"
InsertaConcepto CodInf, 5, ""
InsertaRelacion CodSup, CodInf
InsertaConcepto "PresUSD", 201, "US dollar"
Obra.SetField "Precios.Código", "PAR_0101" ' Precios
Obra.SetField "Precios.Entidad", "PresUSD"
Obra.SetField "Precios.Pres", 45 ' Requiere un valor para crear el registro
Obra.InsertRecord ("Precios")
Obra.EndRedo
Obra.SetUpdateScreen 1
Obra.UpdateScreen
```

End Sub

Function **InsertaConcepto** (Codigo, Nat, Resumen)

```
Obra.InitRecord "Conceptos"
Obra.SetField "Conceptos.Código", Codigo
Obra.SetField "Conceptos.Nat", Nat
Obra.SetField "Conceptos.Resumen", Resumen
InsertaConcepto = Obra.InsertRecord ("Conceptos")
If InsertaConcepto <> 0 Then Obra.LogMsg "Concepto: "& Codigo & " No insertado", 2
```

End Function

Function **InsertaRelacion** (CodSup, CodInf)

```
Obra.InitRecord "Relaciones"
Obra.SetField "Relaciones.CodSup", CodSup
Obra.SetField "Relaciones.CodInf", CodInf
InsertaRelacion = Obra.InsertRecord ("Relaciones")
If InsertaRelacion <> 0 Then Obra.LogMsg "Relación " & CodSup & "/" & CodInf & " No insertada", 2
```

End Function

LogBegin

Inicia la ventana de mensajes

Function **LogBegin** (Mensaje As String) As Integer

Mensaje Aparece en la primera línea de la ventana junto a la hora y la fecha. En VBS aparece el nombre del complemento.

Retorno 0: correcto, 1: error

Obsoleta desde Presto 24.01. Ver "LogMsg"

LogEnd Muestra los mensajes

Function LogEnd () As Integer

Retorno 0: correcto, 1: error

Obsoleta desde Presto 24.01. Ver "LogMsg"

LogMsg Agrega un mensaje a la ventana de mensajes

Function LogMsg (Mensaje As String, Color As Integer) As Integer

Color 0: Negro 1: Azul 2: Rojo

Retorno 0: correcto, otro valor: error

Obsoleta desde Presto 24.01. Ver "Msg.Clear", "Msg.Text", "Msg.SetTabs" y "Msg.WrapText".

Los mensajes sólo se muestran en la ventana cuando se activa "LogEnd" y el proceso completo del complemento ha terminado.

Hay un límite de 2000 mensajes y no se repite un mismo mensaje.

La instrucción "LogBegin" y "LogEnd" deben ejecutarse en pareja, o las dos o ninguna. Para impedir que se ejecute la primera, pero no la segunda, se puede utilizar como en el ejemplo. Ver "SetModal".

MsgClear Limpia la ventana de mensajes

Function MsgClear

Sólo desde Presto 24.01

MsgSetTabs Establece tabuladores en la ventana de mensajes

Function MsgSetTabs (Tabuladores As String)

Tabuladores Tabuladores en cm, en orden creciente, separados por ","

Sólo desde Presto 24.01

MsgText Envía texto a la ventana de mensajes

Function MsgText (Texto As String, Color As Integer)

Color 0: Negro 1: Azul 2: Rojo

Sólo desde Presto 24.01

A diferencia de la función "LogMsg", "MsgText" se envía a la ventana de mensajes tan pronto como se ejecuta, no requiere inicialización ni finalización de la ventana, es sensible a los tabuladores establecidos con la función "MsgSetTabs" y no incluye un retorno de carro, por lo que se pueden crear mensajes progresivamente o separados por tabuladores.

Ejemplo VBS

```
Obra.MsgClear  
Obra.MsgWrapText 0  
Obra.MsgSetTabs "3;7"  
Obra.MsgText Codigo & vbTab & Resumen & vbNewLine, 0
```

MsgWrapText Ajuste de línea de la ventana de mensajes

Function MsgWrapText (Modo As Integer)

Modo 0: No ajusta 1: Ajusta el texto al ancho de la ventana

Sólo desde Presto 24.01

New Crea y abre una obra

Function New (Obra As String) As Integer

Obra Ruta y nombre de la obra

Retorno 0: correcto, otro valor: error

Ejemplo VB

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

```

        Obra.Close ()
        MsgBox ("Se ha cerrado la obra")
    End If
End Sub

```

Ejemplo VBS

```

Set Obra = GetObject("", "Presto.App.18")
If Obra.New ("C:\TEMP\EJEMPLO.Presto") = 0 Then
    MsgBox "Obra creada correctamente"
    Obra.Close
    MsgBox "Se ha cerrado la obra"
End If

```

Open Abre una obra existente

Function **Open (Obra As String, [ReadOnly As Integer]) As Integer**

Obra	Camino y nombre de la obra
ReadOnly	1: abre la obra en modo sólo lectura 0: abre la obra en modo escritura 2: convierte la obra a formato ".Presto"
Retorno	0: correcto, otro valor: error

Ejemplo VB

```

Public Sub Abrir_Obra (path As String)
Dim Obra As Object
Set Obra = CreateObject ("Presto.App.18")
If Obra.Open (path,1)= 0 Then
    MsgBox "Obra abierta correctamente"
Else
    MsgBox "No se abrió correctamente la obra"
EndIf
End Sub

```

Ejemplo VBS

```

Set Obra = GetObject("", "Presto.App.18")
If Obra.Open ("C:\TEMP\Presto.Presto") = 0 Then
    MsgBox "Obra abierta correctamente"
Else
    MsgBox "No se ha abierto la obra"
End If

```

PrintRPT Imprime un informe

Function PrintRPT (Informe As String, [Preguntar As Integer], [Copias As Integer], [Impresora As Integer]) As Integer

Informe Camino y extensión del informe

Preguntar 0: Toma las respuestas de defecto, 1: Abre la caja de diálogo

Copias Número de copias

Impresora 0: Impresora por defecto, 1: Preguntar impresora

Retorno 0: correcto, otro valor: error

No actúa en WebAPI

Ejemplo VB

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

Ejemplo VBS

```
Set Obra = GetObject("", "Presto.App.18")  
Camino = "C:\Archivos de programa\Presto 2024.01\Report\01 Presupuestos y mediciones"  
Nombre = "Presupuesto y mediciones"  
Informe = Camino & "\ & Nombre & ".PrestoReport"  
Preguntar = 1 ' Defecto / Preguntar  
Copias = 1  
Impresora = 1 ' Defecto / Preguntar  
Respuesta = MsgBox (Informe, vbYesNo, "Imprimir")  
If Respuesta = vbYes Then  
    Error = Obra.PrintRPT (Informe, Preguntar, Copias, Impresora)  
    If Error = 0 Then  
        MsgBox "Informe imprimido"  
    Else  
        MsgBox "Informe no imprimido"  
    End If  
End If
```

Quit Cierra la aplicación de Presto

Sub Quit ()

No actúa en WebAPI

Ejemplo VB

```
Public Sub Cierra_Presto ()  
    Dim Obra As Object  
    Obra = CreateObject ("Presto.App.18")  
    MsgBox ("Se ha abierto Presto")  
    Obra.Quit ()  
    MsgBox ("Se ha cerrado Presto")  
End Sub
```

Recalculate Bloquea o desbloquea el recálculo

Sub Recalculate (Activar As Integer)

Activar 0: Desactiva, 1: Activa de nuevo y recalcula

Al terminar el proceso es conveniente activar la función para recalcular.

No actúa en WebAPI

Ejemplo

Procedimiento que bloquea los recálculos de Presto

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

Rename Permite modificar el campo "Conceptos.Código"

Function Rename (Clave As String, Valor As String) As Integer

Clave "Conceptos.Código"

Retorno 0: correcto, Otro valor: error

A diferencia de "SetField", "Rename" cambia el código del concepto manteniendo la integridad relacional de los demás registros vinculados a este elemento.

No se debe modificar el código cuando es el criterio de ordenación que se está utilizando para recorrer la tabla.

Ejemplo VB

```
Public Sub Renombrar ()
```

```

Dim Obra As PrestoLib.PrestoApplication
Obra = CreateObject ("Presto.App.18")
Obra.SetElement (1, "Conceptos")
While Obra.GetElement (1) = 0
Obra.Rename ("Conceptos.Código",Obra.GetField ("Conceptos.Código") & "_x")
End While

```

End Sub

Ejemplo VBS

```

Set Obra = GetObject ("Presto.App.18")
Obra.SetElement 1, "Conceptos", "Conceptos.Nat", "4"
PrimeraVez = True
While Obra.GetElement (1) = 0
  If PrimeraVez = True Then Obra.BeginRedo
  PrimeraVez = False
  Codigo = Obra.GetField ("Conceptos.Código")
  Error = Obra.Rename ("Conceptos.Código", Codigo & "_x")
  If Error Then Obra.LogMsg Codigo & " No renombrado", 2
Wend
If PrimeraVez = False then Obra.EndRedo

```

SelCode Elige una unidad de obra en un árbol

Function SelCode (Ventana As Integer, Obra As String, Relación As String) As Integer

Ventana Instancia abierta de Presto en la que se abre la caja de diálogo, por defecto 0

Obra Camino y nombre de la obra que se abre para elegir la unidad de obra.
"" representa la obra activa.

Relación Relación entre concepto superior e inferior seleccionada en la caja de diálogo
"" representa el concepto raíz

Retorno 0: Código en blanco
1: Se ha seleccionado un código
-1: Cancelado
Otros: error

El parámetro "Relación" devuelve la relación seleccionada por el usuario. Puesto que debe ser de tipo string, la función no puede utilizarse en VBS.

No actúa en WebAPI

Ejemplo VB

```
Public Sub Sel_Code (obra As String, relacion As String)
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SelCode (0, obra, relacion)
End Sub
```

SetCharacter Selecciona ANSI (Windows) o OEM (MS-DOS).

Sub SetCharacter (modo As Integer)

Modo 0: ANSI, Western Windows-1252
 1: OEM, MS-DOS CP 437

Se puede usar antes de leer un texto de un registro o de evaluar una expresión

Ejemplo VB

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

SetConceptVarValue Asigna valor a una variable para un concepto

Function SetConceptVarValue (Guid As String, Código As String, Valor As Variant, TipoCálculo As Integer, Nota As String, UsrNum As Double, ElementId As String, onlyNew As Integer) As Integer

Guid	Identificador único global de la variable
Código	Código del concepto al que se asigna el valor de la variable
TipoCálculo	Valores del campo con el mismo nombre de la tabla "Variables".
Nota	
Valor	Valores del campo con el mismo nombre de la tabla
UsrNum	"ValoresVar" que se asocian al concepto para esa variable
ElementID	
OnlyNew	0: Se actualizan los valores existentes, 1: No se modifican valores existentes
Retorno	0: Correcto, 1: No se pudo asociar el valor al concepto

Tanto el concepto como la variable deben existir antes de asignarles valor. El valor debe ser de un tipo compatible con el tipo de la variable

Ejemplo

Ver ejemplo en "*SetTakeoffVarValue*"

SetCurrentInvPeriod Establecer la fecha de certificación actual

Function SetCurrentInvPeriod (as String) As Integer

Fecha Se introduce en formato "yyyymmdd" y el día tiene que estar dado de alta previamente.

Retorno 0: Correcto, Otro valor: Error

Ejemplo VB

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

Ejemplo VBS

```
Set Obra = GetObject ("","Presto.App.18")
Error = Obra.SetCurrentInvPeriod ("20240223")
If Error Then
    MsgBox "No fijada"
Else
    MsgBox "Fijada la fecha"
End If
```

SetElement Accede a una tabla por clave, máscara y filtro

Function	SetElement (Elemento As Integer, Tabla As String, [Clave As String], [Máscara As String], [Selección As String], [OrdenPor As String], [Orden As Integer]) As Integer
Elemento	Número entre 1 y 25 que permite acceder a los elementos de las tablas de Presto y enlazarlas como en el diseño de informes
Clave	Afecta al orden de lectura y a la máscara.
Máscara	Se aplica al campo clave y debe llegar a Presto entre comillas. Filtrar por máscara es mucho más eficiente que por selección, ya que sólo se leen los registros que la cumplen.
Selección	Cualquier expresión evaluable por Presto que permite filtrar los elementos de la tabla. Ver "EvalNum" y "EvalStr".
OrdenPor	Expresión para ordenar la tabla
Orden	0: Ascendente, 1: Descendente
Retorno	0: Correcto, Otro valor: Error

Los parámetros funcionan igual que los campos de la caja de diálogo de propiedades de un elemento de informes.

Se accede a un elemento concreto con la función "GetElement".

Ejemplo VB

```
Public Sub Abrir_Conceptos ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos", "Conceptos.Nat", """"4""")
End Sub
```

Ejemplo VBS

```
' Conceptos con texto
Obra.SetElement 1, "Conceptos", "Conceptos.Código", """"*""", "len(Conceptos.Texto)"
' Conceptos de tipo Partida
Obra.SetElement 1, "Conceptos", "Conceptos.Código", chr(34)&"*"&chr(34), "Conceptos.Nat
== 5"
' Relaciones con conceptos de tipo partida o tarea que sean calculadas
Obra.SetElement 1, "Relaciones", "Relaciones.Relación", , "(Conceptos.Nat==5 ||
Conceptos.Nat==218) && Relaciones.TipoActividadPlan == 2"
' Conceptos con el campo "EDT" y de tipo precio
Obra.SetElement 1, "Conceptos", "Conceptos.Código", """"*""", "Conceptos.EDT != """" &&
Conceptos.Clase == 0" ' Limita a los de clase Precio
```

SetField Asigna valor a un campo

Sub SetField (Campo As String, Valor, [Dimensión As Integer])

Dimensión Índice del campo. Ver "GetField" y "Manual: Gestión del proyecto: Seguimiento del coste: Otros valores desglosados por estados".

Para guardar el cambio en la obra hay que usar "InsertRecord" o "UpdateRecord", o "Rename" en el caso de "Conceptos.Código".

Ejemplo VB

```
Public Sub poner_Str ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos")
    While Obra.GetElement (1) = 0
        Obra.SetField ("Conceptos.Resumen", Obra.GetField ("Conceptos.Código"))
        Obra.UpdateRecord ("Conceptos")
    End While
End Sub
```

Ejemplo VBS

```
Resumen = Obra.GetField ("Conceptos.Resumen")
Obra.SetField "Conceptos.Resumen", UCase(Resumen) ' Cambia a mayúsculas
Obra.UpdateRecord ("Conceptos")
```

SetFieldBinary Asigna valor a un campo binario

Sub SetFieldBinary (campo As String, valor)

Ejemplo

```
Public Sub poner_Archivo (valor)
    Dim Obra As Object
    Set Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Archivos")
    While Obra.GetElement (1) = 0
        Obra.SetFieldBinary ("Archivos.Archivo", valor)
        Obra.UpdateRecord ("Archivos")
    End While
End Sub
```

Sub SetModal (Modal As Integer)**Modal** 0: Bloquea, 1: Desbloquea

Conviene bloquear el uso de Presto para que no se altere mientras se ejecuta el complemento y realizarlo de manera que sólo se desbloquee si se ha bloqueado previamente, como se ve en el ejemplo.

No actúa en WebAPI.

Ejemplo VB

Procedimiento que bloquea la ventana de Presto.

```
Public Sub Bloquea_Presto ()  
    Dim Obra As Object  
    Set Obra = CreateObject ("Presto.App.18") Obra.SetModal ( 0)  
End Sub
```

Ejemplo VBS

' Al principio del complemento

PrimeraVez = True

' Antes de modificar la obra

If PrimeraVez = True then

Obra.LogBegin ""

Obra.SetModal 0

Obra.BeginRedo

Obra.SetUpdateScreen 0

PrimeraVez = false

End if

' Al finalizar el complemento

If PrimeraVez = False then

Obra.EndRedo

Obra.SetModal 1

Obra.LogEnd

Obra.SetUpdateScreen 1

Obra.UpdateScreen

End If

SetPriceVarValue Asigna variable a un concepto y una entidad de precios

Function SetPricetVarValue (Guid As String, Concepto Sup As String, Entidad As String, Valor As Variant, [TipoCálculo As Integer], [Nota As String], [UserNum As Double], [ElementId As String])

Guid Identificador único global de la variable.

Concepto Código del concepto donde recibe valor la variable

Entidad Código de la entidad de asignación de precios donde recibe valor la variable

Retorno 0: Correcto, Otro valor: Error

Asigna un valor a una variable en una relación entre un concepto y una entidad de asignación de precios. Además se pueden actualizar los campos, "ElementID", "Nota" y "UsrNum".

Tanto relación entre el concepto y la entidad como la variable deben existir previamente.

Ver "SetConceptVarValue" y "SetRelationVarValue"

SetRelationVarValue Asigna variable a una relación entre conceptos

Function SetRelationVarValue (Guid As String, CodSup As String, CodInf As String, Valor As Variant, [TipoCálculo] As Integer, [Nota As String], [UsrNum As Double], [ElementId As String])

Guid Identificador único global de la variable.

CodSup Código del concepto superior de la relación

CodInf Código del concepto inferior de la relación

Retorno 0: correcto, Otro valor: Error

Asigna un valor a una variable en una relación entre dos conceptos. Además se pueden actualizar los campos, "ElementID", "Nota" y "UsrNum".

Los conceptos, la relación entre ellos y la variable deben existir previamente.

Ver "SetConceptVarValue" y "SetPriceVarValue"

Los ejemplos corresponden a la obra creada en el ejemplo de "InsertRecord"

SetTakeoffInsertMode Modo de inserción de las líneas de medición

Sub SetTakeoffInsertMode (Modo As Integer)

Modo Comportamiento cuando se utiliza "InsertRecord" para insertar una línea de medición:

0: Inserción normal en "Mediciones".

1: Si la inserción en "Mediciones" falla se intenta insertar la línea en "MedicionesTmp" (mediciones temporales).

2: Inserción normal en "MedicionesTmp"

3: Inserción en "MedicionesTmp" con borrado previo de la tabla.

Ejemplo VB

' Actúa sobre la obra creada en InsertRecord

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

```
Obra.SetUpdateScreen (0)
```

```
Obra.BeginRedo
```

```
Obra.SetTakeoffInsertMode (1)
```

' Cantidad

```
Obra.InitRecord ("Mediciones")
```

```
Obra.SetField ("Mediciones.GuidElem", "GUID_MED_01")
```

```
Obra.SetField ("Mediciones.CodSup", "CAP_01")
```

```
Obra.SetField ("Mediciones.CodInf", "PAR_0101")
```

```
Obra.SetField ("Mediciones.Cantidad", 100)
```

```
Obra.InsertRecord ("Mediciones")
```

```
Obra.InitRecord ("Mediciones")
```

```
Obra.SetField ("Mediciones.GuidElem", "GUID_MED_01")
```

```
Obra.SetField ("Mediciones.CodSup", "CAP_01")
```

```
Obra.SetField ("Mediciones.CodInf", "PAR_0101")
```

```
Obra.SetField ("Mediciones.Cantidad", 99)
```

```
Obra.InsertRecord ("Mediciones")
```

' Dimensiones

```
Obra.InitRecord ("Mediciones")
```

```
Obra.SetField ("Mediciones.GuidElem", "GUID_MED_02")
```

```
Obra.SetField ("Mediciones.CodSup", "CAP_01")
```

```
Obra.SetField ("Mediciones.CodInf", "PAR_0101")
```

```
Obra.SetField ("Mediciones.N", 100)
```

```
Obra.InsertRecord ("Mediciones")
```

' Fórmula FIEBDC

```
Obra.InitRecord ("Mediciones")
```

```
Obra.SetField ("Mediciones.GuidElem", "GUID_MED_03")
```

```
Obra.SetField ("Mediciones.CodSup", "CAP_01")
```

```
Obra.SetField ("Mediciones.CodInf", "PAR_0101")
```

```
Obra.SetField ("Mediciones.N", 2)
```

```

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

```

' Expresión de Presto

```

Obra.InitRecord ("Mediciones")
Obra.SetField ("Mediciones.GuidElem", "GUID_MED_04")
Obra.SetField ("Mediciones.CodSup", "CAP_01")
Obra.SetField ("Mediciones.CodInf", "PAR_0101")
Obra.SetField ("Mediciones.N", 5)
Obra.SetField ("Mediciones.Volumen", 2)
formula_med = "$[Mediciones[Guid= ""GUID_MED_03""].Longitud]$"
Obra.SetField ("Mediciones.Fórmula", formula_med)
Obra.InsertRecord ("Mediciones")
Obra.Calculate (3)
Obra.SetTakeoffInsertMode (0)
Obra.EndRedo
Obra.SetUpdateScreen (1)
Obra.UpdateScreen
MsgBox "FIN insertar_mediciones"

```

Ejemplo VBS

```

' Para la obra creada en InsertRecord
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetUpdateScreen 0
Obra.BeginRedo
Obra.SetTakeoffInsertMode 0 ' Inserción normal
' Cantidad
Obra.InitRecord "Mediciones"
Obra.SetField "Mediciones.Cantidad", 99
GuidElem = InsertaMedicion ("CAP_01", "PAR_0101")
' Dimensiones
Obra.InitRecord "Mediciones"
Obra.SetField "Mediciones.N", 1
Obra.SetField "Mediciones.Longitud", 5.35
Obra.SetField "Mediciones.Anchura", 0.80
Obra.SetField "Mediciones.Altura", 1.10
GuidElem = InsertaMedicion ("CAP_01", "PAR_0101")
' Fórmula matemática repitiendo dimensiones anteriores
Obra.SetField "Mediciones.Fórmula", "(a+b)*(c+d)"
Obra.SetField "Mediciones.FórmulaTipo", 1
GuidElem = InsertaMedicion ("CAP_01", "PAR_0101")
'' Expresión de Presto
Obra.InitRecord "Mediciones"
Obra.SetField "Mediciones.N", 5
Obra.SetField "Mediciones.BIMVol", 2
GUID = InsertaMedicion ("CAP_01", "PAR_0101")
Expresion = "$[Mediciones[Guid= "" & GUID & ""].BIMVol]$"
Obra.SetField "Mediciones.Fórmula", Expresion

```



```

Error = Obra.UpdateRecord ("Mediciones")
If Error <> 0 Then Obra.LogMsg "Línea " & CodSup & "/" & CodInf & " No insertada", 2
Obra.Calculate 3
Obra.EndRedo
Obra.SetUpdateScreen 1
Obra.UpdateScreen

```

SetTakeoffVarValue Asigna variable y valor a una línea de medición

Function	SetTakeoffVarValue (Guid As String, GuidElement As String, GuidAux As String, Valor As Variant, [ValTipoCálculo As Integer], [ValNota As String], [ValUsrNum As Double], [ValElementId As String], [ValNuevos As Integer]) As Integer
Guid	Identificador único de la variable
GuidElement GuidAux	El identificador único de la línea de medición es la concatenación de ambos valores
Valor ValTipoCálculo ValNota ValUsrNum ValElementId	Campos asociados al valor en la tabla "Valores", no a la variable
ValNuevos	0: Los valores existentes se actualizan
Retorno	0: Correcto, Otro valor: Error

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 previamente.

Ejemplo VB

```

' Actúa sobre la obra creada en InsertRecord
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetUpdateScreen (0)
Obra.BeginRedo ()

Dim guid_var
' Variable de tipo cadena
guid_var = Obra.CreateVariable ("GUID_VAR_01", "VAR_01", "Variable tipo String ", 0, 6)
Obra.SetConceptVarValue (guid_var, "PAR_0101", "Valor en concepto")
Obra.SetTakeoffVarValue (guid_var, guid_elem, guid_aux, "Valor en medición")
' Variable de tipo cadena y combo
guid_var = Obra.CreateVariable ("GUID_VAR_02", "VAR_02", "Variable tipo String y combo", 0, 6, , , , 1, "A|B|C|D")
Obra.SetConceptVarValue (guid_var, "PAR_0101", "A")

```

```
Obra.SetTakeoffVarValue (guid_var, guid_elem, guid_aux, "C")
```

```
' Variable de tipo real (con mínimo, máximo, redondeo a 10 decimales)
```

```
guid_var = Obra.CreateVariable ("GUID_VAR_03", "VAR_03", "Variable tipo Real", 0, 5, 0, 0.5, 1, 10.7, , , , , , , , 10)
```

```
Obra.SetConceptVarValue (guid_var, "PAR_0101", 5.5)
```

```
Obra.SetTakeoffVarValue (guid_var, guid_elem, guid_aux, 7.2)
```

```
' Variable de tipo fecha
```

```
guid_var = Obra.CreateVariable ("GUID_VAR_04", "VAR_04", "Variable tipo fecha", 0, 7)
```

```
Obra.SetConceptVarValue (guid_var, "PAR_0101", "20180506")
```

```
Obra.SetTakeoffVarValue (guid_var, guid_elem, guid_aux, 14006)
```

```
Obra.EndRedo ()
```

```
Obra.SetUpdateScreen (1)
```

```
Obra.UpdateScreen ()
```

Ejemplo VBS

Sub InsertarVariables

```
' Actúa sobre la obra creada en InsertRecord y InsertarMediciones
```

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

```
Obra.SetUpdateScreen 0
```

```
Obra.BeginRedo
```

```
Obra.SetField "Mediciones.Cantidad", 99
```

```
GuidElem = InsertaMedicion ("CAP_01", "PAR_0101")
```

```
' Variable de tipo cadena
```

```
Guid = Obra.CreateVariable ("GUID_01", "VAR_01", "Variable string", 0, 6) ' Usuario, Cadena
```

```
Obra.SetConceptVarValue Guid, "PAR_0101", "Valor en concepto"
```

```
Obra.SetTakeoffVarValue Guid, GuidElem, "", "Valor en medición"
```

```
' Variable de tipo cadena y combo
```

```
Guid = Obra.CreateVariable ("GUID_02", "VAR_02", "Variable combo ", 0, 6, , , , , 1, "A|B|C|D")
```

```
Obra.SetConceptVarValue Guid, "PAR_0101", "A"
```

```
Obra.SetTakeoffVarValue Guid, GuidElem, "", "C"
```

```
' Variable de tipo real (con mínimo, máximo, redondeo a 10 decimales)
```

```
Guid = Obra.CreateVariable ("GUID_03", "VAR_03", "Variable real ", 0, 5, 0, 0.5, 1, 10.7, , , , , , , , , , , 10)
```

```
Obra.SetConceptVarValue Guid, "PAR_0101", 5.5
```

```
Obra.SetTakeoffVarValue Guid, GuidElem, "", 7.2
```

```
' Variable de tipo fecha
```

```
Guid = Obra.CreateVariable ("GUID_04", "VAR_04", "Variable de fecha ", 0, 7)
```

```
Obra.SetConceptVarValue Guid, "PAR_0101", "20240506"
```

```
Obra.SetTakeoffVarValue Guid, GuidElem, "", "20240506"
```

```
Obra.EndRedo
```

```
Obra.SetUpdateScreen 1
```

```
Obra.UpdateScreen
```

```
End Sub
```

SetText Rellena un campo de texto

Sub SetText (Campo As String, Valor As String)

Ejemplo VB

```
Public Sub poner_Texto ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos")
    While Obra.GetElement (1) = 0
        Obra.SetText ("Conceptos.Texto",
            Obra.GetFieldStr ("Conceptos.Código"))
    End While
End Sub
```

Ejemplo VBS

```
Obra.SetText "Facturas.Texto", Texto
```

SetUniqueVarValue Asigna un valor de asignación única a una variable

Function SetUniqueVarValue (Guid as String, Valor As Variant) As Integer

Retorno 0: Correcto, Otro valor en caso contrario

La variable debe existir previamente

Ejemplo VB

```
Dim Obra As Object
    Obra = GetObject ("", "Presto.App.18")
    Obra.SetUpdateScreen (0)
    Obra.BeginRedo ()
'Real CalcDurLab = 9.5
    Obra.SetUniqueVarValue ("PRESTOHEADVAR000-0020063", 9.5)
'Bool CalcNoRedCalculos = 1 (Sí)
    Obra.SetUniqueVarValue ("PRESTOHEADVAR000-0020060", 1)
'Date DivFecha[1] = 28/02/2018
    Obra.SetUniqueVarValue ("PRESTOHEADVAR000-0020086", "20240228")
    Obra.EndRedo ()
    Obra.SetUpdateScreen (1)
    Obra.UpdateScreen ()
```

```
MsgBox ("FIN actualiza_valores_unicos!")
```

Ejemplo VBS

```
Sub ActualizaVariableUnica
```

```
' Guid fijo de la tabla de variables
```

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

```
CalcDurLab = 8 ' Real
```

```
Obra.SetUniqueVarValue "PRESTOHEADVAR000-0020063", CalcDurLab
```

```
CalcNoRedCalculos = False
```

```
Obra.SetUniqueVarValue "PRESTOHEADVAR000-0020060", CalcNoRedCalculos
```

```
DivFecha1 = "20240228" ' Fecha
```

```
Obra.SetUniqueVarValue "PRESTOHEADVAR000-0020086", DivFecha1
```

```
End Sub
```

SetUpdateScreen Desactiva la actualización de las ventanas

Sub SetUpdateScreen (Activar As Integer)

Activar 0: Desactiva, 1: Activa

Si está desactivada, los valores de las ventanas visibles no se actualizan, agilizando el proceso cuando hay operaciones que afectan a muchos registros.

Al terminar hay que usar la función de nuevo con el parámetro 1 y ejecutar "UpdateScreen" para ver los cambios.

No actúa en WebAPI.

Ejemplo VB

```
Public Sub Bloquea_VentanaPresto ()
```

```
Dim Obra As Object
```

```
Obra = CreateObject ("Presto.App.18")
```

```
Obra.SetUpdateScreen (0)
```

```
End Sub
```

Ejemplo VBS

```
Obra.SetUpdateScreen 0
```

```
...
```

```
Obra.SetUpdateScreen 1
```

```
Obra.UpdateScreen
```

SetVar Modifica una variable de asignación única

Sub SetVar (Variable As String, Valor As String)

Variable GUID o nombre de la variable

Ejemplo VB

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

Ejemplo VB

```
Set Obra = GetObject ("", "Presto.App.18")
Obra.SetVar "DecCantMed", 3
```

ShowWindow Abre una ventana

Sub ShowWindow (Ventana As Integer)

Ventana 0: Árbol
1: Presupuesto
2: Mediciones
3: Mediciones temporales
4: Espacios

No actúa en WebAPI

Ejemplo VB

```
Dim Obra As Object
    Obra = GetObject ("", "Presto.App.18")
    Obra.SetUpdateScreen (0)
    Obra.BeginRedo ()
' Espacio con número, código, , color (rojo) y cota (10.77).
' Como el espacio es > 1, añadirá los espacios intermedios por defecto
    Obra.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)
    Obra.AddSpace ("ESP_COTA_0_55", "Espacio creado desde un complemento de usuario",
65280, 0.55)
' Mostramos la ventana de espacios
    Obra.ShowWindow (4)
    Obra.EndRedo ()
    Obra.SetUpdateScreen (1)
    Obra.UpdateScreen ()
MsgBox ("FIN inserta espacios !")
```

Ejemplo VBS

```
Set Obra = GetObject ("", "Presto.App.18")  
Obra.ShowWindow 0  
Obra.ShowWindow 1  
Obra.ShowWindow 2  
Obra.ShowWindow 3  
Obra.ShowWindow 4
```

SQLServerDelete Borra una obra en SQL Server

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
-----------------	---

Parámetros	Ver "SQLServerExport"
-------------------	-----------------------

bstrGuid	Guid de la obra que se va a eliminar
-----------------	--------------------------------------

Retorno	0: Correcto, otro valor: error
----------------	--------------------------------

Ejemplo

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

SQLServerExport Exporta la obra a SQL Server

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
<i>bstrServer</i>	Nombre del servidor donde está alojado el servidor SQL
<i>bstrUser</i>	Usuario que se utilizará para acceder al servidor SQL
<i>bstrPassword</i>	Contraseña para acceder al servidor SQL
<i>nPort</i>	Puerto por el que se va a acceder al servidor SQL
<i>nTimeout</i>	Timeout de la conexión con SQL Server
<i>bstrDataBase</i>	Nombre de la base de datos
<i>bOverwrite</i>	0: No sobrescribir la obra si existe
<i>bCalcFields</i>	0: No exportar campos calculados (grises)
<i>bstrSelTables</i>	Tablas que se van a exportar, separadas con ";". Para exportar todas las tablas: ""
Retorno	0: Correcto, otro valor: error

Ejemplo VB

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

SQLServerImport Importa una obra de SQL Server

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
Parámetros	Ver "SQLServerExport"
<i>bstrGuid</i>	Guid de la obra que se va a importar
Retorno	0: correcto, otro valor: error

Ejemplo VB

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

UpdateRecord Actualiza los cambios en un registro

Function UpdateRecord (Tabla As String) As Integer

Retorno 0 correcto, otro valor: error

*Actualiza en la obra un registro cuyos campos han cambiado.
Si se modifica un campo clave hay que usar "Rename" para no perder la integridad relacional.*

Ejemplo VB

```
Public Sub Modifica_Registro ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.SetElement (1, "Conceptos", "Conceptos.Nat", """"5""")
    While Obra.GetElement (1) = 0
        Obra.SetField ("Conceptos.Resumen", "Esto es una partida")
        Obra.UpdateRecord ("Conceptos")
    End While
End Sub
```

Ejemplo VBS

```
Obra.SetField "Conceptos.Resumen", UCase(Resumen) ' Cambia a mayúsculas
Obra.UpdateRecord ("Conceptos")
```

UpdateScreen Actualiza la información de las ventanas

Sub UpdateScreen ()

*Necesaria cuando se ha desactivado y luego activado la actualización con "SetUpdateScreen".
No actúa en WebAPI.*

Ejemplo VB

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

Ejemplo VBS

```
Obra.SetUpdateScreen 0  
...  
Obra.SetUpdateScreen 1  
Obra.UpdateScreen
```

Atributos (Properties)

Property	Property	Devuelve
IULanguage	Integer	Idioma del interfaz del programa
Project	String	Path de la obra de Presto abierta
ProjectLanguage	Integer	Idioma de la obra
ReadOnly	Integer	La obra está en modo "solo lectura"
Status	Integer	0 si todo es correcto después de una llamada a una función del gestor de archivos
SubVersionNum	Integer	Subversión de Presto en formato número
VersionNum	Integer	Versión de Presto en formato número
VersionStr	String	Nombre completo de la versión de Presto

Ejemplos

```
Public Sub Ver_Project ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    MsgBox (Obra.Project)
    MsgBox CStr (Obra.Status)
    Dim strMensaje
    strMensaje = p Obra o.VersionStr & ", " & Obra.Version.Num & Obra.SubVersionNum
    MsgBox (strMensaje)
End Sub
Public Sub Ver_SoloLectura ()
    Dim Obra As Object
    Obra = CreateObject ("Presto.App.18")
    Obra.Open ("C:\Temp\Obra.Presto")
    If (Obra.ReadOnly) Then
        MsgBox ("Solo lectura")
    Else
        MsgBox ("Lectura y escritura")
    End If
End Sub
```

Cajas de diálogo (WebAPI)

En cada ejemplo figura el número de opción y la constante para "DialogOptionsCode".

Todas las variables en las que no se indique lo contrario pueden tener los valores:

- 0 No activado | 1 Activado.
- "mask": Igual que en Presto
- "concept": Código del concepto sobre el que se ejecutará la opción
- "selection":
- "tables": Las tablas sobre las que se ejecutará la opción con filas seleccionadas.

Vector de posiciones: Con el identificador de la tabla a afectar, se indicarán en una lista las diferentes filas sobre las que se ejecutará la opción.

Los archivos JSON no admiten comentarios.

Ejemplo VB.NET

JSON base

```
{
  "objGuid": "{{objGuid}}",
  "opCode": "ComExecuteOption",
  "option": {{Nº de opción}},
  "parameters": {{Parámetros de la opción}},
  "sessionGuid": "{{sessionGuid}}"
}
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)
    If DataGridView1.SelectedRows.Count > 0 Then
      Dim iNumeroDeFilaSeleccionada = DataGridView1.CurrentRow.Index
      MsgBox(DataGridView1.Rows(iNumeroDeFilaSeleccionada).Cells(0).Value)
    Else
      'MessageBox.Show("Selecciona una fila")
    End If
    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 retVal As Integer = Com.OpenObj()
      'Utils.PrintJsonResponse("OpenCom", Com.m_strJsonResponse)
      If retVal = 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

```

9901

DLG_REDUCIR_NIVELES

```

{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "mode": 1
}

```

"mode": 0 Capítulos | 1 Subcapítulos | 2 Partidas | 3 Auxiliares

13351

DLG_MULTIPLICAR_PRECIOS

```

{
  "concept": "0",
  "checkedLabor": 1,
  "checkedMaterials": 1,
  "checkedOther": 1,
  "checkedPlant": 1,
  "percentageLabor": 80,
  "percentageMaterials": 70,
  "percentageOther": 10,
  "percentagePlant": 90
}

```

"percentage*": Porcentaje de las diferentes naturalezas a operar.

11901

DLG_AJUSTAR_PRECIO

```
{  
  "concept": "0",  
  "newPrice": 338551.74,  
  "onlyAffect": 0,  
  "priceChapters": 1,  
  "priceLabor": 1,  
  "priceMaterials": 1,  
  "priceOther": 1,  
  "pricePlant": 1,  
  "priceWorkitem": 1,  
  "quantityLabor": 0,  
  "quantityOther": 0,  
  "quantityPlant": 0  
}
```

"newPrice": El precio sobre el que se ajustará el presupuesto.

"onlyAffect": Se refiere a no alterar los precios de conceptos que afectan al resto del presupuesto.

13721

DLG_FUSIONAR_PARTIDAS

```
{  
  "mask": "*",  
  "selection": {  
    "Relaciones": [  
      1529,  
      1530  
    ],  
    "tables": [  
      "Relaciones"  
    ]  
  },  
  "checkField": 1,  
  "field": "Conceptos.CategoríaBIM",  
  "checkVar": "1",  
  "var": "120Volt | 802338"  
}
```

"field": Campo a comprobar

"var": Variable a comprobar

7051

DLG_COPIAR_CERT

```
{
```

```

"mask": "*",
"selection": {
  "Relaciones": [
    1529,
    1530
  ],
  "tables": [
    "Relaciones"
  ]
},
"source": 0,
"destination": 1,
}

```

"source": 0 Certificación | 1 Ejecución | 2 Planificación | 3 Imputación

"destination": 0 Certificación | 1 Ejecución | 2 Planificación

10751 DLG_DESGLOSAR_PARTIDAS_DES

```

{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  }
}

```

9801 DLG_ELIMINAR_DESCOMPOSICION_OBJ

```

{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "laborOption": 2,
  "plantOption": 2,
  "materialOption": 1,
  "otherOption": 1,
  "groupConsolidated": 0
}

```

"*Option": 0 No seleccionado | 1 Elimina | 2 Refunde

7251 DLG_ELIMINAR_DESCOMPOSICION_OBJ

```

{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "decomposition": 2,
  "reduceLevels": 1,
}

```

"decomposition": 0 Sin descomponer | 1 Mismas unidades de obra

"reduceLevels": Solo funciona si "decomposition" tiene valor 1

10801 DLG_GENERAR_OBJ

```

{
  "concept": "0",
  "action": 0,
  "checkedLabor": 1,
  "checkedPlant": 1,
  "checkedMaterials": 1,
  "checkedOther": 1,
  "percentageLabor": 80,
  "percentagePlant": 90,
  "percentageMaterials": 70,
  "percentageOther": 10,
  "notGenerate": 1
}

```

"action": 0 Generar objetivo | 1 Generar presupuesto

"percentage*": Porcentaje a aplicar sobre los precios, de 0 a 100.

"notGenerate": No generar cantidades de partidas

8151 DLG_GENERAR_TERMINOS

```

{
  "includeWordsLongerThan": 1,
  "charactersLongerThan": 8,
  "replaceSpecialChars": 1,
  "ignoreWordsWithNumbers": 0,
  "includeOnlyFirstWords": 1,
  "numberWordsOnlyFirst": 5,
  "deletePreviousGeneratedKeywords": 0
}

```

"charactersLongerThan": Número de caracteres mínimo para incluir palabras. Debe estar activado "includeWordsLongerThan".

"numberWordsOnlyFirst": Número de palabras que se incluirán del texto. Debe estar activado "includeOnlyFirstWords".

8851**DLG_CREAR_FECHAS**

```
{
  "initialDay": "20200430",
  "certificationDay": 31,
  "months": 9,
  "addAllMonthDays": 0,
}
```

"initialDay": Día inicial en formato Presto. Para convertir una fecha en este formato utilizar "Utils.GetDateInPrestoFormat"

"certificationDay": Día en el que se certificará

"months": Número de meses

11951**DLG_OPERAR_GRAFICOS**

```
{
  "changeFormat": 1,
  "newFormat": 2,
  "compression": 50,
  "changeFormatIfSizeIsSmaller": 1,
  "createImageFromVectorDrawing": 0,
  "longestSidePixels": 600,
  "whiteBackgroundInResultingImage": 0,
  "convertDXFtoDWG": 0,
}
```

"createImageFromVectorDrawing" y "convertDXFtoDWG" son excluyentes

"newFormat": 0 BMP | 1 EPS | 2 JPEG | 3 PCX | 4 PNG | 5 TIFF | 6 GIF

"compression": Entre 0 y 100 con valores incrementados de 10 en 10

"longestSidePixels": Lado mayor en píxeles entre 1 y 9999

15271**DLG_CONVERTIR_PRES_PLAN**

```
{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "createTasksByWorkItemBreakdown": 0,
  "createTasksByWorkItemGroups": 0,
  "createTasksByDivisionOfWorkItems": 1,
  "divideBy": "Mediciones.Espacio",
}
```

"divideBy": Campo de mediciones por el que se va a dividir. Tiene que estar activado "createTasksByDivisionOfWorkItems".

8561

DLG_REESTRUCTURAR_LINEAS_MED

```
{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "replaceReferences": 0,
  "convertIntoTakeoffLines": 0,
  "sort": 1,
  "orderField1": "Espacio",
  "orderField2": "Archivo",
  "orderField3": "",
  "createSubTotals": 0,
  "consolidate": 1,
  "consolidateType": 2
}
```

"orderField1", "orderField2", "orderField3": Valores por los que se ordenará

"consolidateType": 0 Iguales en todos los campos, excepto "N" | 1 Con una dimensión diferente, además de "N" | 2 Iguales, en fases distintas

11701

DLG_RELLENAR_PLANIFICACION

```
{
  "mode": 0,
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "target": 0.0,
  "certificationsWithAssignedWeight": 0,
  "weightSum": 1.0,
  "planDates": 0,
}
```

"mode": 0 Aplicando los datos del diagrama de barras | 1 Aplicando los pesos de planificación de cada certificación | 2 Aplicando a las unidades de obra la planificación de las tareas

"mask", "selection": Sólo modo 1

"target": Objetivo en formato double, sólo modo 2

"certificationsWithAssignedWeight": Certificaciones con peso asignado en formato entero, sólo modo 2

"weightSum": Suma de los pesos en formato double, sólo modo 2

"planDates": 0 Fechas ASAP | 1 Fechas ALAP, sólo modo 3

7101

DLG_CREAR_FACTURA_MEDICION

```
{
  "mask": "*",
  "selection": {
    "Relaciones": [
      1529,
      1530
    ],
    "tables": [
      "Relaciones"
    ]
  },
  "incomePlanning": 1,
  "incomeCertifications": 1,
  "expensesPlanning": 1,
  "expensesContract": 1,
  "supplierCertification": 0,
  "associatedSuppliers": 0,
  "priceWithAmountMode": 0,
}
```

Hay dos modos de operación, para la fase actual y para todas las fases. El modo se indica activando al menos un parámetro del modo deseado.

"incomePlanning", "incomeCertifications", "expensesPlanning", "expensesContract": Todas las fases

"supplierCertification", "associatedSuppliers": Fase actual.

"priceWithAmountMode": 0 Certificación | 1 Ejecución. Fase actual. Para utilizar estos modos "associatedSuppliers" debe estar activado