

API y WebAPI Desarrollo de complementos

API y WebAPI	2
Desarrollo de complementos	2
Complementos sencillos en Visual Basic.....	3
Creación de complementos con el API de Presto	5
Puesta en marcha WebAPI	7
Comprobación de PrestoServer.....	7
Activar Internet Information Services	7
Instalar ASP.NET Runtime	8
Configurar IP y puerto	8
Agregar el sitio web en IIS	9
Modificar el archivo "hosts"	10
Ejemplo de PrestoCloudApiClient en VB.NET	11
Diferencias entre complementos API y WebAPI	13
Diferencias entre API y WebAPI.....	13
Transformar un complemento API en WebAPI.....	13
Ejemplos	14
Ejemplos para ejecutar diálogos u opciones	15
Notas comunes	15
Ejemplo de código VB.NET.....	15
Reducir niveles	16
Operar.....	16
Ajustar.....	17
Fusionar	17
Copiar cantidades	17
Desglosar partidas	18
Reestructurar descomposición.....	18
Fusionar partidas	18
Generar objetivo	19
Generar términos	19
Crear fechas	20
Operar con gráficos.....	20
Crear tareas	20
Reestructurar mediciones	21
Rellenar la planificación económica.....	21
Crear facturas	22
Lista de funciones	23
Ejemplo: Recorrer el árbol de Presto por niveles	29
C#	29
VB.NET	30

API y WebAPI

Los complementos o "plugins" son programas independientes de Presto que realizan tareas específicas sobre las obras.

Los complementos suministrados con Presto o realizados por los usuarios aparecen en el menú dinámico "Complementos" y se pueden utilizar por cualquier usuario con acceso a ellos.

Los complementos entregados con Presto 22 se describen en los apartados del manual o las notas técnicas correspondientes. Es posible entregar su código fuente mediante acuerdo con RIB Spain.

Desarrollo de complementos

Los complementos se escriben utilizando el API, Application Programming Interface de Presto. Las funciones del API permiten interactuar con una obra y realizar todo tipo de operaciones, creación, modificación y borrado de los registros, incluyendo el acceso directo a todas las opciones del programa.

Para escribir un complemento es necesario conocer algún lenguaje de programación capaz de interactuar con el servidor de automatización de Presto, como VBScript.

API

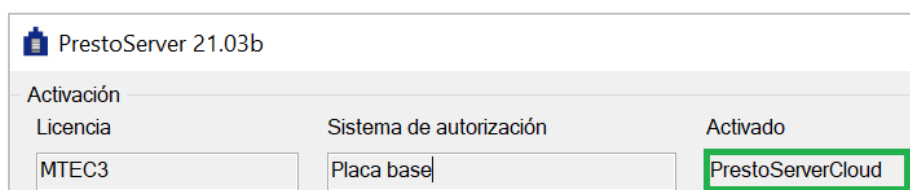
El API permite desarrollar complementos para interactuar con una obra accesible localmente y pueden utilizarlo todos los usuarios.

WebAPI

El servicio WebAPI, disponible desde Presto 2021, permite interactuar también con las obras accesibles en Presto Server a través de peticiones POST.

- Incluye API para encapsular llamadas REST y realizar complementos tradicionales de Presto o utilizar los existentes, que son totalmente compatibles.
- Permite crear páginas web sobre las obras publicadas en PrestoServer, con autenticación mediante login y password.
- Puede ejecutar opciones de menús de Presto mediante archivos JSON con los parámetros.

Para utilizar el WebAPI es necesaria la licencia de Presto ServerCloud.



Complementos sencillos en Visual Basic

Puede utilizar un editor de texto, como el bloc de notas, pero es recomendable un editor especializado en código, como Code Writer o Microsoft Windows Based Script Host.

- Las tablas y los nombres de las variables de Presto aparecen en las cabeceras de cada columna y en la opción "Ver: Lista de campos" del diseño de informes.
- La lista de funciones se encuentra en la tabla más adelante y la sintaxis y ejemplos de cada una en el documento "API y WebAPI Funciones detalladas".

Guarde el archivo como texto ANSI, con el nombre que desee ver en el menú y la extensión ".VBS", en el directorio indicado en "Archivo: Entorno de trabajo: Directorios: Complementos [Usuario]".

Si el formato no es ANSI puede obtenerse un error si existen caracteres acentuados.

El complemento aparecerá en el menú dinámico "Complementos" de Presto, siempre que haya una única obra abierta.

Marcar en estado o color rojo conceptos por sus iniciales

'Crea el objeto que representa a Presto

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

'Filtra la tabla de conceptos por aquellos cuyo código comienza por "E04". Chr(34) representa las comillas entre las que hay que encerrar el propio texto "E04".

```
PrestoEjemplo.SetElement 1, "Conceptos", "Conceptos.Código", Chr(34) & "E04*" & Chr(34)
```

'Recorre los conceptos filtrados

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

```
    'Asigna "Rojo" como estado de los conceptos seleccionados
```

```
    PrestoEjemplo.SetField "Conceptos.Estado", 1
```

```
    'Guarda el cambio realizado en el concepto
```

```
    PrestoEjemplo.UpdateRecord ("Conceptos")
```

```
Wend
```

Exportar facturas a archivo de texto

Este complemento abre la obra especificada, exporta los datos de las facturas a un archivo txt, separados por punto y coma, y cierra la obra, creando un grupo de acciones que se pueden deshacer.

'Definición de variables

```
Dim po
```

```
Dim ruta
```

```
Dim fso,MyFile
```

```
Dim FD,FF,FB,FI
```

'Refiere a la obra

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

'Abre la obra indicada

```
po.Open "C:\temp\Producción y control de costes a partir de partidas.Presto"
```

'Empieza las acciones que se pueden deshacer

```
po.BeginRedo
```

```
ruta = "C:\Temp\Facturas.txt"
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
Set MyFile = fso.CreateTextFile(ruta)
```

```
po.SetElement 1, "Facturas", "Facturas.Documento", Chr(34) & "*" & Chr(34)
```

'Bucle que recorre la tabla "Facturas"

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

```
    FD=po.getfield("Facturas.Documento")
```

```
    FF=po.getfield("Facturas.Fecha")
```

```
    FB=po.getfield("Facturas.BaseFac")
```

```
    FI=po.getfield("Facturas.IVA")
```

```
    MyFile.WriteLine(FD&";"&FF&";"&FB&";"&FI)
```

```
Wend
```

'Fin de las acciones que se pueden deshacer

```
po.EndRedo
```

'Cierra la obra

```
po.close
```

'Cierra el archivo de texto

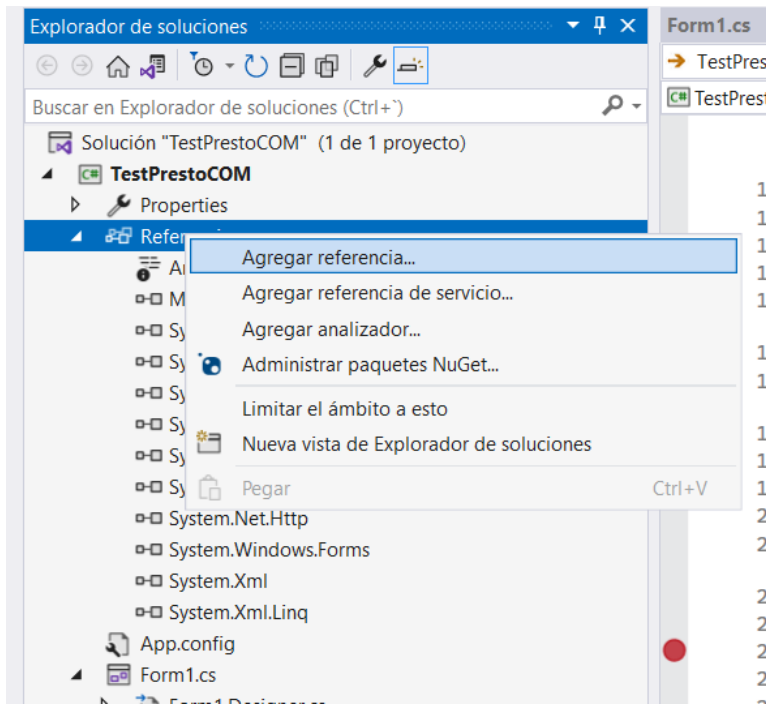
```
MyFile.close
```

Creación de complementos con el API de Presto

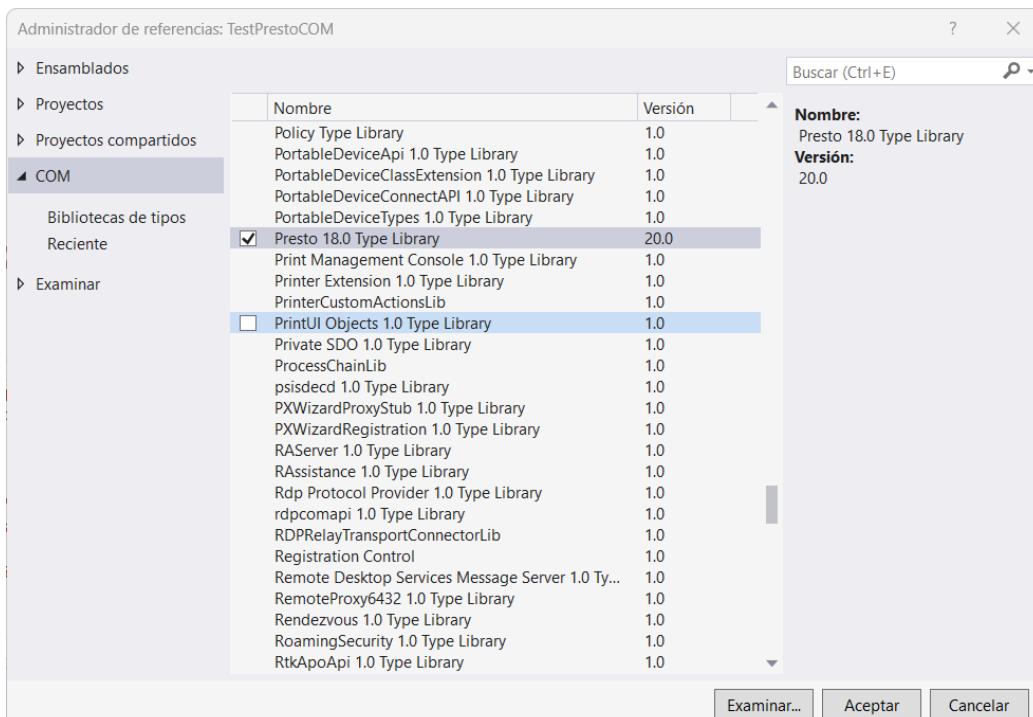
Para comenzar a desarrollar un complemento con el API de Presto hace falta haber ejecutado Presto como administrador, al menos una vez, en el ordenador en el que se va a programar.

Se recomienda el uso de Visual Studio para este propósito, y es el IDE (entorno de desarrollo integrado) con el que se explicará el proceso.

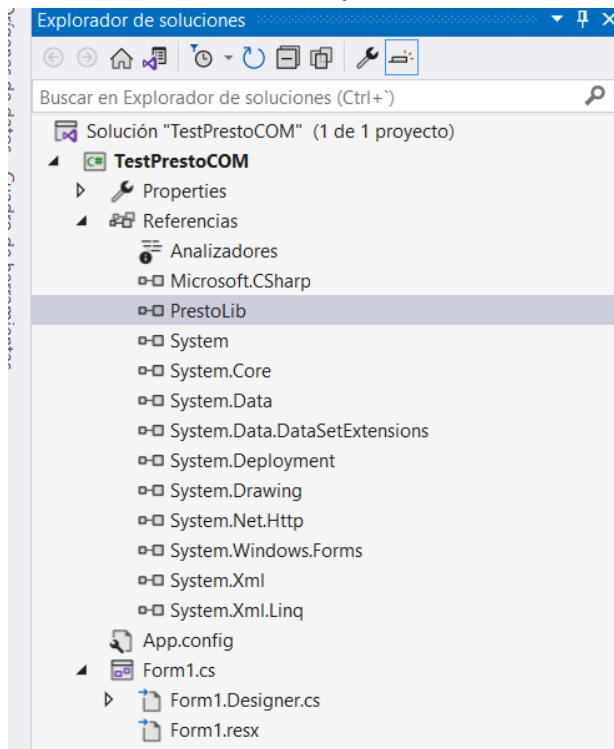
- Con un proyecto creado, agregar referencia como se indica en la imagen



- Buscar en COM la librería Presto 18.0 Type Library y seleccionarla



- De esta manera, ya estaría referenciada la librería de Presto para utilizarla en el desarrollo de complementos.



Puesta en marcha WebAPI

Para acceder al servidor de automatización hace falta:

- Disponer de la licencia de Presto ServerCloud.
- Haber ejecutado Presto como administrador al menos una vez en el ordenador en el que se va a programar.

Para su utilización por el usuario se deben incluir en el mismo directorio donde se encuentren ubicados los complementos los archivos "PrestoCloudJsonModel.dll", "PrestoCloudApiClient.dll" y "Newtonsoft.Json.dll".

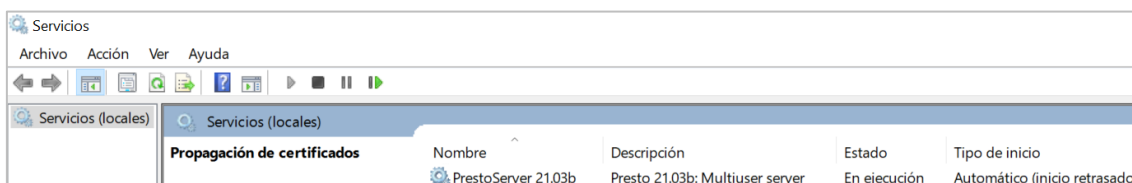
Comprobación de PrestoServer

Requiere tener instalado PrestoServer en un equipo al que tenga acceso el equipo usado para el desarrollo.

Entorno de trabajo						
Generales	Act	Alias	Dirección IP	Puerto	Usuario	Contraseña
	1	<input checked="" type="checkbox"/>	EJEMPLO	192.168.50.43	5200	Administrador *****

Entorno de trabajo: PrestoServer

Para comprobar que está funcionando el servicio PrestoServer acceda a los servicios de Windows.

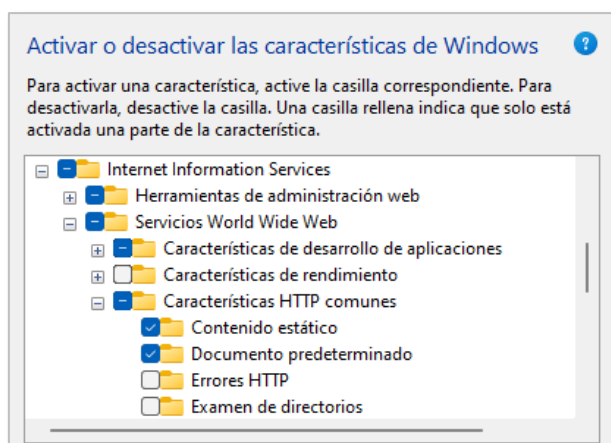


Nombre	Descripción	Estado	Tipo de inicio
PrestoServer 21.03b	Presto 21.03b: Multiuser server	En ejecución	Automático (inicio retrasado)

Servicio PrestoServer

Activar Internet Information Services

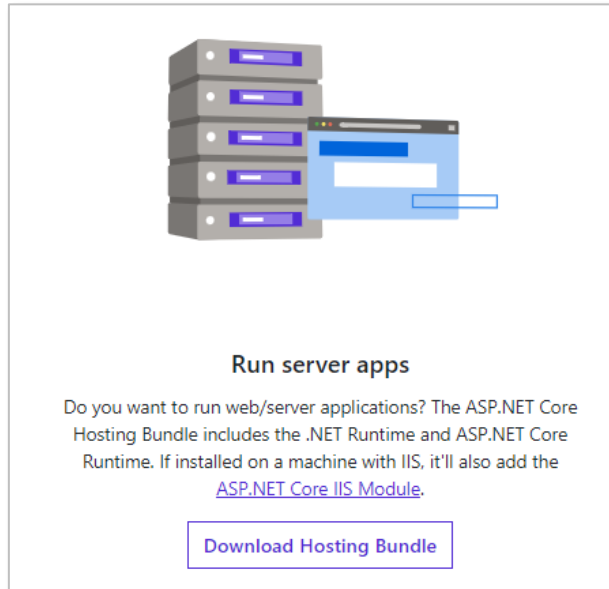
Acceda a "Activar o desactivar las características de Windows".



Instalar ASP.NET Runtime

Descargar e instalar la última versión de "ASP.NET Core Hosting Bundle".

<https://dotnet.microsoft.com>



Para su uso con Presto 22 debe estar instalada la versión 3. Puede comprobar la versión instalada ejecutando el comando `dotnet --list-runtimes` desde Windows PowerShell

Configurar IP y puerto

Copie a `C:\` la carpeta `PrestoCloudWebApi` de ese mismo directorio.

Abra con cualquier editor de textos el archivo:

`C:\PrestoCloudWebApi\appsettings.json`

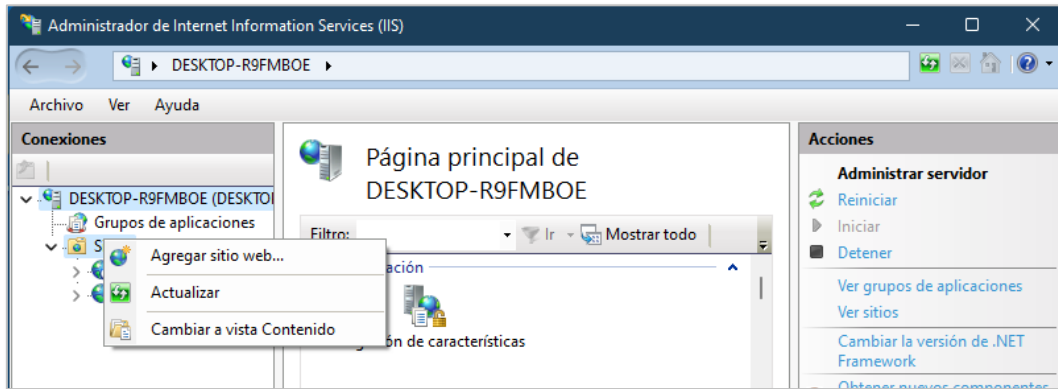
Introduzca la "Ip" y el "Puerto" donde esté instalado PrestoServer, como figuran en "Entorno de trabajo: PrestoServer":

```
appsettings.json
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft": "Warning",
6        "Microsoft.Hosting.Lifetime": "Information"
7      }
8    },
9    "PrestoServer": {
10     "Ip": "192.168.50.43",
11     "Puerto": 5200,
12     "Timeout": 60
13   },
14   "AllowedHosts": "*"
15 }
```

Configuración de "appsettings.json"

Agregar el sitio web en IIS

Acceda al administrador de Internet Information Services (IIS) y sobre "Sitios" active "Agregar sitio web..."



Rellene los valores solicitados.

Agregar sitio web

Nombre del sitio: Grupo de aplicaciones:

Directorio de contenido

Ruta de acceso física:

Autenticación de paso a través

Enlace

Tipo: Dirección IP: Puerto:

Nombre de host:

Valores para añadir el sitio web

Compruebe con "Probar configuración" que los valores son válidos.

Conexión de prueba

Resultados:

Prueba	Configuración
✓ Autenticación	Nombre de usuario (RIBSPAIN\Jorge)
✓ Autorización	La ruta de acceso es accesible (C:\PrestoCloudWebApi).

Conexión de prueba

Si lo desea puede seleccionar que el acceso sea a través de un enlace "https".

Enlace

Tipo: Dirección IP: Puerto:

Nombre de host:

Requerir indicación del nombre de servidor
 Deshabilitar TLS 1.3 a través de TCP Deshabilitar QUIC
 Deshabilitar TLS heredada Deshabilitar HTTP/2
 Deshabilitar asociación de OCSP

Certificado SSL: **Seleccione uno**

Configuración de enlace "https"

Seleccione un certificado SSL válido.

Pruebe la configuración para comprobar que las credenciales son válidas.

En caso de error cierre la ventana "Conexión de prueba" y pulse en "Conectar como". Seleccione "Usuario específico" y añada el usuario, con su dominio, y la contraseña con la que se ha accedido al equipo.

Compruebe que el sitio se ha añadido correctamente.

Nombre	Id.	Estado	Enlace	Ruta de acceso
Default Web Site	1	Iniciado (http)	*:80 (http)	%SystemDrive%\inetpub\wwwroot
webapi.presto.es	2	Iniciado (http)	webapi.presto.es on 192.168.50.43:80 (http)	C:\PrestoCloudWebApi
webapi2.presto.es	3	Iniciado (http)	webapi.presto.es on 192.168.50.43:443 (https)	C:\PrestoCloudWebApi

Sitios web agregados en IIS

Modificar el archivo "hosts"

Abra el archivo "C:\windows\system32\drivers\etc\hosts" con un editor de textos como administrador y añada una línea con la dirección IP de la máquina y el nombre del host "webapi.presto.es", separados por un tabulador.

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost
192.168.50.43 webapi.presto.es
```

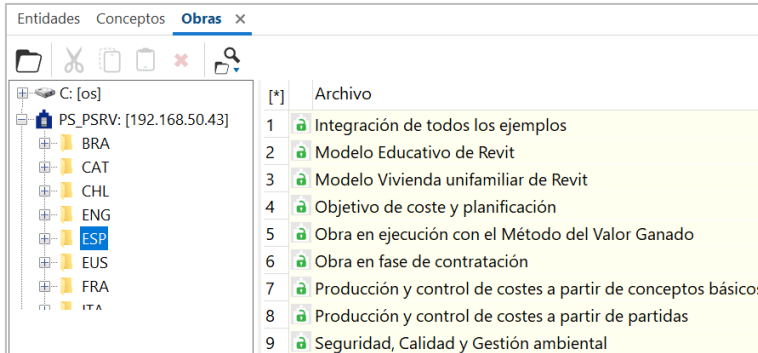
Dirección IP y nombre del host

Opcionalmente puede introducir como nombre del host "iplocal".

Para comprobar la configuración de WebAPI ejecute la aplicación [PrestoCloudTestWebApi](#), como se ha indicado anteriormente.

Ejemplo de PrestoCloudApiClient en VB.NET

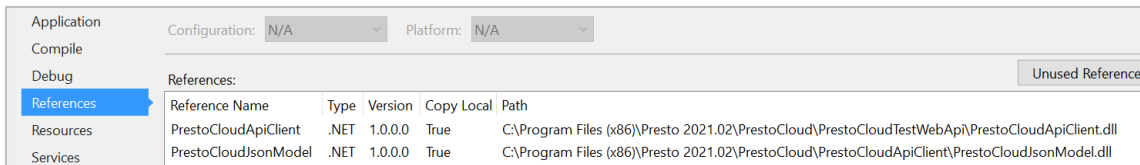
Ejemplo de un complemento que lista todas las obras accesibles en un PrestoServer.



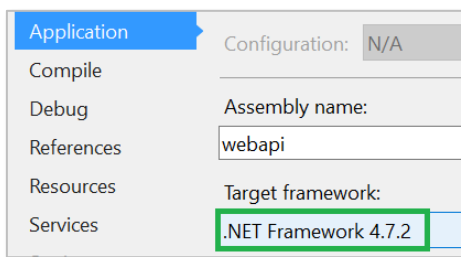
Obras de PrestoServer

Iniciación

- Crear un proyecto VB.NET (.NET Framework Versión 4.7.2 superior o .NET Core 2.0 o superior)
- Añadir referencia a las DLL "PrestoCloudJsonModel" y "PrestoCloudWebApi"



Configuración del compilador



Pasos del código

1. Crear un objeto de tipo PrestoClientApiUser, pasando la dirección del WebAPI.
2. Realizar un "Login" con un usuario válido.
3. Crear un objeto de tipo "PrestoClientApiFiles".
4. Realizar una llamada para obtener el listado de carpetas "ListFiles".
5. Realizar un "Logout".

Código

```
Imports Presto.CloudApiClient
Imports PrestoCloudJsonModel
Public Class Form1
    Private m_prestoUser As PrestoClientApiUser = Nothing
    Private m_prestoCom As PrestoClientApiCom = Nothing
    Dim DirectorioPS As String = "ESP"
    Dim strUser As String = "Administrador"
    Dim strPassword As String = "admin"
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Puebla_DatagridView_Obras_PS()
    End Sub
    Sub Puebla_DatagridView_Obras_PS()
        Dim User As New PrestoClientApiUser("http://webapi.presto.es/webapi/")

        Dim responseValue As ResponseValue = User.Login(strUser, strPassword)

        If (responseValue = responseValue.Ok) Then
            Dim Files As New PrestoClientApiFiles(User)
            Dim listFiles As New List(Of PrestoCloudJsonModel.FileItemInfo)

            responseValue = Files.ListFiles(listFiles, DirectorioPS, False)

            For f = 0 To listFiles.Count - 1
                DataGridView1.Rows.Add(listFiles.Item(f).name)
            Next

            End If

            CreatePrestoApiInstance()

            DataGridView1.AllowUserToAddRows = False
            DataGridView1.AutoSizeColumns()
            User.Logout()
        End Sub
        Private Sub CreatePrestoApiInstance()
            If m_prestoUser Is Nothing Then
                m_prestoUser = New PrestoClientApiUser("http://webapi.presto.es/webapi/")
                Dim returnValue As ResponseValue = m_prestoUser.Login(strUser, strPassword)

                If returnValue = ResponseValue.Ok Then
                    m_prestoCom = New PrestoClientApiCom(m_prestoUser)
                    For f = 0 To DataGridView1.Rows.Count - 2
                        '// Ruta + proyecto dentro de PrestoServer
                        Dim retValue As Integer = m_prestoCom.Open(DirectorioPS & "\" &
DataGridview1.Rows(f).Cells(0).Value)
                        If retValue <> 0 Then m_prestoCom = Nothing
                        With m_prestoCom

                            If .FindEqual("Conceptos", "Conceptos.Nat", 0) = 0 Then
                                DataGridView1.Rows(f).Cells(1).Value = .EvalNum("Conceptos.Pres")
                                DataGridView1.Rows(f).Cells(2).Value = .EvalNum("Conceptos.Obj")
                            End If

                        End With
                    Next
                End If
            End Sub
        End Class
```

Diferencias entre complementos API y WebAPI

Diferencias entre API y WebAPI

Pasa de entero a booleana: ReadOnly

Pasan de dynamic a string: GetField, GetFieldBinary

Algunos métodos pueden haber variado su retorno.

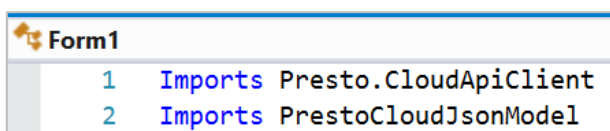
Todos los métodos que antes devolvían "void" ahora devuelven "int", 0 si la operación se ha ejecutado correctamente y otro valor en caso contrario.

Métodos que no hacen nada porque dependen de la interfaz de usuario:

- Command
- CheckProject
- Exit
- ImportCad
- InfoCode
- PrintRPT
- Quit
- Recalculate
- SelCode
- SetModal
- SetUpdateScreen
- ShowWindow
- UpdateScreen

Transformar un complemento API en WebAPI

Añadir referencia a las DLL "PrestoCloudJsonModel" y "PrestoCloudApiClient"



```
Form1
1 Imports Presto.CloudApiClient
2 Imports PrestoCloudJsonModel
```

Comentar la creación del objeto COM antiguo.

Cualquier referencia a:

```
Dim po As PrestoLib.PrestoApplication
po = CreateObject("Presto.App.18")
```

debe sustituirse por

```
m_prestoCom.
```

COLUMNA	CONTENIDO
C# Código similar	Type oPrestoType = Type.GetTypeFromProgID("Presto.App.18") PrestoApplication prestoApplication = (PrestoApplication)Activator.CreateInstance(oPrestoType);
VB.NET ""	Set po = GetObject("", "Presto.App.18")

Crear un método, que simplificará las actualizaciones, donde se crearán los objetos para la comunicación.

Ejemplos

VB.NET

```
Public Class Form1
    Private m_prestoUser As PrestoClientApiUser = Nothing
    Private m_prestoCom As PrestoClientApiCom = Nothing
    Private Sub CreatePrestoApiInstance()
        If m_prestoUser Is Nothing Then

            m_prestoUser = New PrestoClientApiUser("URL del servidor")
            Dim returnValue As ResponseValue = m_prestoUser.Login("User", "Password")

            If returnValue = ResponseValue.Ok Then

                m_prestoCom = New PrestoClientApiCom(m_prestoUser)
                Dim retValue As Integer = m_prestoCom.Open("PrestoEmpty.Presto")
                '// Ruta + proyecto dentro de PrestoServer
                If retValue <> 0 Then m_prestoCom = Nothing

            End If
        End If
    End Sub
End Class
```

C#

```
private static PrestoClientApiUser m_prestoUser = null;
private static PrestoClientApiCom m_prestoCom = null;
private static void CreatePrestoApiInstance()
{
    if (m_prestoUser == null)
    {
        m_prestoUser = new PrestoClientApiUser("URL del servidor");
        ResponseValue returnValue = m_prestoUser.Login("User", "Password");

        if (returnValue == ResponseValue.Ok)
        {
            m_prestoCom = new PrestoClientApiCom(m_prestoUser);
            int retValue = m_prestoCom.Open("PrestoEmpty.Presto");
            // Ruta + proyecto dentro de PrestoServer
            if (retValue != 0) m_prestoCom = null;
        }
    }
}
```

Ejemplos para ejecutar diálogos u opciones

Notas comunes

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 de código 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 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

```

Reducir niveles

9901 DLG_REDUCIR_NIVELES

```

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

```

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

Operar

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.

Ajustar

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.

Fusionar

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

Copiar cantidades

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

Desglosar partidas

10751 DLG_DESGLOSAR_PARTIDAS_DES

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

Reestructurar descomposición

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

Fusionar partidas

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

Generar objetivo

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

Generar términos

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".

Crear fechas

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

Operar con gráficos

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

Crear tareas

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".

Reestructurar mediciones

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

Rellenar la planificación económica

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

Crear facturas

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.

Lista de funciones

Los parámetros, la definición de la acción o retorno y un ejemplo de uso figuran en el documento "API Funciones detalladas".

* Funciones accesibles sólo en WebAPI

FUNCIÓN	DESCRIPCIÓN
AddCert	Crea una nueva certificación a continuación de la última existente. Si no hay certificaciones crea la primera. El número y la fecha de la certificación se calculan con el criterio de defecto (una certificación por mes). Si en el color se recibe <code>0xFF000000</code> se genera un color aleatorio.
AddSpace	Agrega un espacio con el código, descripción y color indicados y el siguiente número de espacio válido (en el orden establecido por Presto)
BeginRedo	Inicio de acciones que se pueden deshacer
Calculate	Lanza en Presto uno de los cálculos del 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.)
CheckProject	Comprueba la obra abierta
Close	Cierra una obra de Presto
Command	Activa cualquier opción de menú de Presto. Puede llamarse sólo con el nombre de la opción o precediéndola con el nombre del menú separado por pipe " ". Es posible agrupar varios "Command" separándolos por punto y coma ";"
Copy	Copia un campo en otro
CreateSpace	Crea un espacio con el número, código, resumen y color indicados. Si no se indican código o resumen se generan los de defecto. Puede asignar color para controlar las alternativas de presupuesto. Si en el color se recibe <code>0xFF000000</code> se genera un color aleatorio. Si el número de espacio o el código ya existe devolverá error. Se crean todos los espacios necesarios para que no haya huecos.
CreateVariable	Crea o actualiza un registro de la tabla "Variables"
Delete	Borra una obra
DeleteRecord	Borra un registro manteniendo la integridad relacional

FUNCIÓN	DESCRIPCIÓN
Duplicate	Duplica un registro de la tabla de conceptos de la misma forma que duplicar de Presto, es decir, copiando la descomposición y demás informaciones asociadas. Para copiar simplemente un registro de esta o de otra tabla, basta con leerlo, cambiar la clave e insertarlo de nuevo
EndRedo	Fin de acciones que se pueden deshacer
EvalNum	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor numérico
EvalStr	Admite cualquier expresión soportada por el generador de expresiones de Presto y devuelve un valor de cadena
ExecuteOption *	Ejecuta una opción de Presto
Exit	Cierra una instancia de Presto (sólo la ventana principal a la que nos referimos, no toda la aplicación)
ExportFIEBDC	Exporta la obra en formato FIEBDC
ExportRPT	Exporta un informe
FindEqual	Se posiciona en una tabla dado un valor de una clave
FindFirst	Se posiciona en el primer registro de una tabla según una clave
FindGreat	Se posiciona en el registro mayor a un valor dado de una tabla según una clave
FindGreatEqual	Se posiciona en el registro mayor o igual a un valor dado de una tabla según una clave
FindLast	Se posiciona en el último registro de una tabla según una clave
FindLessEqual	Se posiciona en el registro menor o igual a un valor dado de una tabla según una clave
FindNext	Se posiciona en el registro siguiente de una tabla
FindPos	Se posiciona en un registro de una tabla ordenada por una determinada clave, conociendo su posición
FindPrev	Se posiciona en el registro anterior de una tabla
GetElement	Se posiciona en el primer registro de una tabla
GetField	Devuelve el valor de un campo de Presto
GetFieldBinary	Devuelve el valor de un campo binario de Presto
GetFieldNum	Devuelve el valor de un campo numérico

FUNCIÓN	DESCRIPCIÓN
GetFieldStr	Devuelve el valor de un campo de tipo cadena
GetPos	Obtiene la posición de un registro. La posición es un número asociado por el gestor de archivos a cada registro
GetSelection	Recorre los elementos seleccionados de una tabla
GetText	Devuelve el valor del texto con los mismos parámetros que otras funciones "Get"
ImportCad	Importa un archivo CAD (AllPlan, ArchiCad, FIEBDC) sobre una obra de Presto
ImportFIEBDC	Importa la obra en formato FIEBDC
InfoCode	Información de un concepto de Presto
InitRecord	Pone a cero un registro del tipo indicado en la tabla. Este registro se puede llenar luego con valores y grabar con las demás funciones
InitVar	Suprime variables declaradas con float o con char dentro de una expresión usada en EvalStr o EvalNum
InsertAttachment	Inserta un archivo en Presto a partir de un buffer y lo asocia con un registro de una tabla por medio del código
InsertAttachmentFrom Path	Inserta un archivo en Presto a partir de un <i>path</i> y lo asocia con un registro de una tabla por medio del código
InsertAttachmentFrom Url *	Inserta un archivo a partir de una URL
InsertRecord	Inserta en la tabla un registro con la clave que tenga en ese momento. Si la clave ya existe y la tabla no admite duplicados, devolverá un error
LogBegin	Inicia un proceso de log en Presto
LogEnd	Cierra el proceso de log en Presto y muestra la ventana con todos los mensajes agregados. Incluye un mensaje con la fecha y hora de finalización
LogMsg	Agrega un mensaje al sistema de log
New	Crea y abre una obra de Presto
Open	Abre una obra de Presto
PrintRPT	Imprime un informe
Property IULanguage As Integer	Devuelve el idioma del interfaz del programa

FUNCIÓN	DESCRIPCIÓN
Property Project As String	Devuelve el path de la obra de Presto abierta
Property ProjectLanguage As Integer	Devuelve el idioma de la obra
Property ReadOnly As Integer	Devuelve si la obra está en modo "solo lectura"
Property Status As Integer	Después de una llamada a una función del gestor de archivos devuelve 0 todo correcto
Property SubVersionNum As Integer	Devuelve la sub-versión de Presto en formato número
Property VersionNum As Integer	Devuelve la versión de Presto en formato número
Property VersionStr As String	Devuelve el nombre completo de la versión de Presto
Quit	Cierra la aplicación de Presto
Recalculate	Bloquea o desbloquea los recálculos intermedios para evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamarla de nuevo para hacerlos todos de una vez
Rename	Cambia (renombrar) la clave de un concepto, estando posicionados en el concepto que se quiere renombrar
SelCode	Elige un concepto de Presto
SetCharacter	Asigna valor a la variable identificada por su <i>guid</i> en el concepto con el código indicado. El concepto y la variable deben existir antes de asignar valor. El tipo del valor debe ser compatible con el de la variable
SetConceptVarValue	Establece como certificación actual aprobada la fecha recibida
SetCurrentInvPeriod	Selecciona el juego ANSI (Windows) o OEM (MS-DOS). Se usa antes de leer un texto de un registro o de evaluar una expresión
SetElement	Define un elemento. Un elemento es una forma de acceder a una tabla de una obra de Presto, similar a la utilizada en informes. Se accede a un elemento concreto con GetElement Elemento indica un número entre 1 y 25, ya que se pueden definir varios SetElement al tiempo

FUNCIÓN	DESCRIPCIÓN
SetField	Asigna un valor a un campo (en la tabla solo se graba con InsertRecord o UpdateRecord)
SetFieldBinary	Asigna un valor a un campo binario de Presto
SetModal	Activa y desactiva el modo modal de Presto (mientras está puesto, la ventana de Presto está bloqueada): Bloquea o Desbloquea el uso de Presto La velocidad de ejecución es mucho más rápida
SetPriceVarValue	Asigna valor a la variable identificada por su <i>guid</i> en la relación entre el concepto y la entidad de asignación de precios indicados. Tanto el concepto como la variable deben existir antes de asignar valor. El tipo del valor debe ser compatible con el de la variable
SetRelationVarValue	Asigna valor a la variable identificada por su <i>guid</i> en la relación indicada. Tanto la relación como la variable deben existir antes de asignar valor. El tipo del valor debe ser compatible con el de la variable
SetTakeoffInsertMode	Permite alternar entre la inserción estándar de registros en mediciones mediante "InsertRecord" y la inserción en mediciones temporales, si falla la inserción en mediciones
SetTakeoffVarValue	Asigna valor a la variable identificada por su <i>guid</i> en la línea de medición identificada por su <i>guidElement</i> y su <i>guidAux</i> . Tanto la línea de medición como la variable deben existir antes de asignar valor. El valor debe ser de un tipo compatible con el tipo de la variable
SetText	Rellena un campo de texto
SetUniqueVarValue	Asigna un valor de tipo de asignación "Única" a una variable. La variable debe existir previamente
SetUpdateScreen	No actualiza los cambios que deberían reflejarse en las ventanas visibles, incluyendo su recorrido, para evitar demoras cuando hay operaciones que afectan a muchos registros. Al terminar conviene llamar a SetUpdateScreen con el parámetro 1 para reflejar todos los cambios de una sola vez
SetVar	Modifica una variable de Presto
ShowWindow	Abre y sitúa como activa la ventana principal indicada
SQLServerDelete	Borra en SQL Server
SQLServerExport	Exporta a SQL Server
SQLServerImport	Importa de SQL Server

FUNCIÓN	DESCRIPCIÓN
UpdateRecord	Actualiza en la tabla un registro cuyos campos han cambiado. Si el campo clave hubiera sido modificado, cambiará también en la tabla, pero sin mantener la integridad relacional de las tablas asociadas. Para cambiar los campos clave, usar la función Rename
UpdateScreen	Actualiza las ventanas

Ejemplo: Recorrer el árbol de Presto por niveles

C#

```
Private void SurroundingSub ()
{
    Object po;
    String codigo;
    String nombre;
    String superior;
    var CodigoInf;
    po = Interaction.GetObject ("", "Presto.App.18");
    superior = "###";
    po.SetElement (1, "Relaciones", "Relaciones.CodSup", Strings.Chr (34) + superior +
Strings.Chr (34), "Conceptos.Nat!=219");
    po.SetModal(0)
    While (po.GetElement (1) == 0)
    {
        superior = po.GetFieldStr ("Relaciones.CodInf");
        codigo = po.GetFieldStr ("Conceptos.Código");
        nombre = po.GetFieldStr ("Conceptos.Resumen");
        po.SetElement (2, "Relaciones", "Relaciones.CodSup", Strings.Chr (34) + superior +
Strings.Chr (34), "Conceptos.Nat!=219");
        While (po.GetElement (2) == 0)
        {
            superior = po.GetFieldStr ("Relaciones.CodInf");
            codigo = po.GetFieldStr ("Conceptos.Código");
            nombre = po.GetFieldStr ("Conceptos.Resumen");
            po.SetElement (3, "Relaciones", "Relaciones.CodSup", Strings.Chr (34) + superior +
Strings.Chr (34), "Conceptos.Nat!=219");
            While (po.GetElement (3) == 0)
            {
                superior = po.GetFieldStr ("Relaciones.CodInf");
                codigo = po.GetFieldStr ("Conceptos.Código");
                nombre = po.GetFieldStr ("Conceptos.Resumen");
                po.SetElement (4, "Relaciones", "Relaciones.CodSup", Strings.Chr (34) + superior +
Strings.Chr (34), "Conceptos.Nat!=219");
                While (po.GetElement (4) == 0)
                {
                    superior = po.GetFieldStr ("Relaciones.CodInf");
                    codigo = po.GetFieldStr ("Conceptos.Código");
                    nombre = po.GetFieldStr ("Conceptos.Resumen");
                }
            }
        }
    }
    po.SetModal(1)
}
```

VB.NET

```
Private Sub SurroundingSub ()
    Dim po As Object
    Dim code As String
    Dim name As String
    Dim superior As String

    po = GetObject ("", "Presto.App.18")
    superior = "###"
    po.SetElement (1, "Relaciones", "Relaciones.CodSup", Chr (34) & superior & Chr (34),
"Conceptos.Nat!=219")
    While po.GetElement (1) = 0
        superior = po.GetFieldStr ("Relaciones.CodInf")
        code = po.GetFieldStr ("Conceptos.Código")
        name = po.GetFieldStr ("Conceptos.Resumen")

        po.SetElement (2, "Relaciones", "Relaciones.CodSup", Chr (34) & superior & Chr (34),
"Conceptos.Nat!=219")
        While po.GetElement (2) = 0
            superior = po.GetFieldStr ("Relaciones.CodInf")
            code = po.GetFieldStr ("Conceptos.Código")
            name = po.GetFieldStr ("Conceptos.Resumen")

            po.SetElement (3, "Relaciones", "Relaciones.CodSup", Chr (34) & superior & Chr
(34), "Conceptos.Nat!=219")
            While po.GetElement (3) = 0
                superior = po.GetFieldStr ("Relaciones.CodInf")
                code = po.GetFieldStr ("Conceptos.Código")
                name = po.GetFieldStr ("Conceptos.Resumen")

                po.SetElement (4, "Relaciones", "Relaciones.CodSup", Chr (34) & superior & Chr
(34), "Conceptos.Nat!=219")
                While po.GetElement (4) = 0
                    superior = po.GetFieldStr ("Relaciones.CodInf")
                    code = po.GetFieldStr ("Conceptos.Código")
                    name = po.GetFieldStr ("Conceptos.Resumen")

                    End While
                End While
            End While
        End While
    End Sub
```