



**SDK Administrative v2.0**

**saint**



# Tabla de contenido

Parte I .....	7
Instalación .....	9
Requerimientos .....	9
Instalación de .NET Framework 4.0.....	9
Instalación de Internet Information Services (IIS).....	11
Instalación del SDK Administrative v2.0.....	13
Parte II .....	17
Configuración .....	19
Activación del SDK.....	20
Parte III .....	21
Región Infos.....	23
Info_Global() .....	23
Info_Quantity().....	24
Info_Quantity( .....	24
Info_Query() .....	24
Región InitProcess .....	26
Init_AdditionalFields() .....	26
Init_CorrelatesOperation().....	27
Init_CorrelatesFact().....	27
Región HeaderProcess .....	29
Header_Customer().....	29
Header_AddCustomer() .....	30
Header_AdditionalReturns() .....	31
Header_CustomerAgreements() .....	31
Header_Seller() .....	32
Header_Deposit() .....	33
Header_Suppliers().....	33
Header_Supplier() .....	34
Header_AddSupplier().....	35
Header_SaveSupplier().....	36

Header_DelSupplier()	36
Header_DelCustomer()	37
Header_SaveCustomer()	38
Región ItemsProcess	39
Item_FindAll()	39
Item_OffersAgreements()	40
Item_CalculatedTax()	40
Item_Product()	41
Item_Products()	42
Item_ProductsLoc()	43
Item_PartsProduct()	44
Item_Taxes()	45
Item_Service()	46
Item_ProdExistence()	46
Item_Lote()	47
Item_Serials()	48
Item_ProductRef()	48
Total_ValidateAgreements()	50
Total_SaveDocument()	51
Total_PaymentInstruments()	52
Total_WithHoldingTax	53
Total_WaitDocumentTypeFac()	53
Región ReportProcess	55
Report_SalesDetails()	56
Report_AnalyticalSuppliers()	57
Report_AnalyticalSupplierH()	58
Report_AnalyticalSupplierD()	58
Report_AnalyticalCustomer()	59
Report_AnalyticalCustomerH()	60
Report_AnalyticalCustomerD()	60
Report_AdvSupplier()	61
Report_AdvSupplierD()	62
Report_AdvCustomer()	62

Report_AdvCustomerD()	63
Report_EffectsReceivable()	64
Report_EffectsReceivableD()	64
Report_EffectsReceivableDO()	65
Report_GeneralSales()	66
Report_EffectsPayable()	66
Report_EffectsPayableD()	67
Report_TotEffectsPayable()	68
Report_TotEffectsReceivable()	68
Report_DeferredDebits()	69
Report_DeferredCredits()	70
Report_TotPaymentCxC()	70
Report_TotPaymentsCxC()	71
Report_Payments()	72
Report_PaymentsLastMonths()	72
Report_Sales()	73
Report_SalesLastMonths()	74
Report_SalesInvoiceCreditD()	74
Report_SalesInvoiceCashD()	75
Report_TotAcctPayable()	76
Report_TotAcctsPayable()	76
Report_ItemsInvoice()	77
Report_TotReleases()	78
Report_ServiceFees()	78
Report_ServiceFeesD()	79
Report_BillsPayableSupp()	80
Report_BillsReceivableCust()	80
Report_TotDistAcctsPayable()	81
Report_TotDistAcctsPayableD()	82
Report_TotDistAcctsReceivable()	82
Report_TotDistAcctsReceivableD()	83
Report_BalanceBanks()	84
Report_ExplorationBanks()	84

Report_CashBanks()	85
Report_BalanceDBanks()	86
Report_DebitCreditNotes()	86
Report_DebitNotesD()	87
Report_CreditNotesD()	88
Report_AcctsReceivable()	88
Report_TotAcctReceivable()	89
Report_TotAcctReceivable()	90
Report_PurchasesIVA()	90
Report_SalesIVA()	91
Report_WithholdingIVAReceivableD()	92
Report_WithholdingIVASalesD()	92
Report_WithholdingIVAPurchasesD()	93
Report_WithholdingIVA()	94
Report_InventoryFixtures()	94
Report_InventoryFixturesD()	95
Report_DischargeInventoryD()	96
Report_SalesProd()	96
Report_SalesServ()	97
Report_SalesProdD()	98
Report_PurchasesProdD()	98
Report_TotalSales()	99
Report_SalesDiscounts()	100
Report_CostsProdServD()	100
Report_FeesCosts()	101
Report_PurchaseLastMonths()	102
Report_SalesNumber()	102
Report_ComsnSellers()	103
Report_ComsnSellers()	104
Report_TotComsnSeller()	104

# Parte I

## Instalación

---

Esta sección contiene la descripción paso a paso para la instalación del SDK Administrative v2.0





# Instalación

A partir de este párrafo describiremos paso a paso el proceso de instalación del SDK Administrative v2.0.

## Requerimientos.

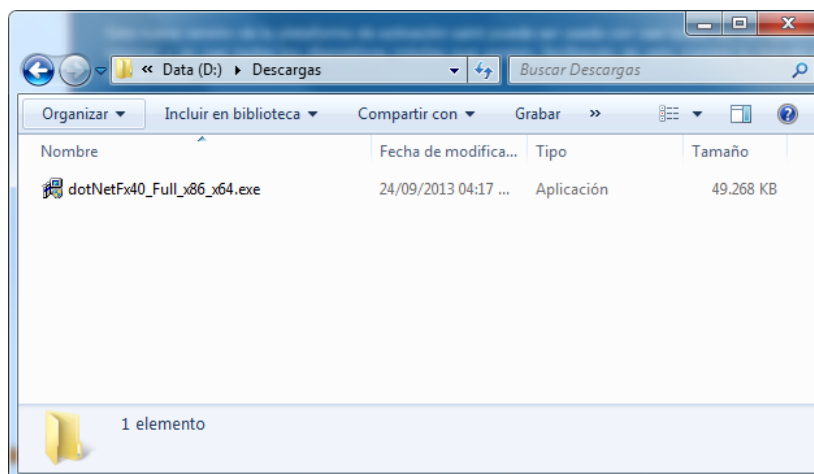
Antes de iniciar la instalación del SDK Administrative v2.0, es importante observar que para instalar y funcionar adecuadamente, son necesarios algunos requisitos tales como:

- ✓ Sistema Operativo Windows 7.x o superior.
- ✓ Internet Information Server (IIS) 7.x o superior, y estar integrado el .NET Framework 4.0.
- ✓ Espacio Libre en disco 5.0 MB
- ✓ Saint License Manager v2.x. Para más información sobre la instalación puedes visitar el siguiente enlace [www.saintnet.com](http://www.saintnet.com), en la sección Descargas -> Utilitarios.
- ✓ Saint Enterprise Administrativo. Para más información visita [www.saintnet.com](http://www.saintnet.com)

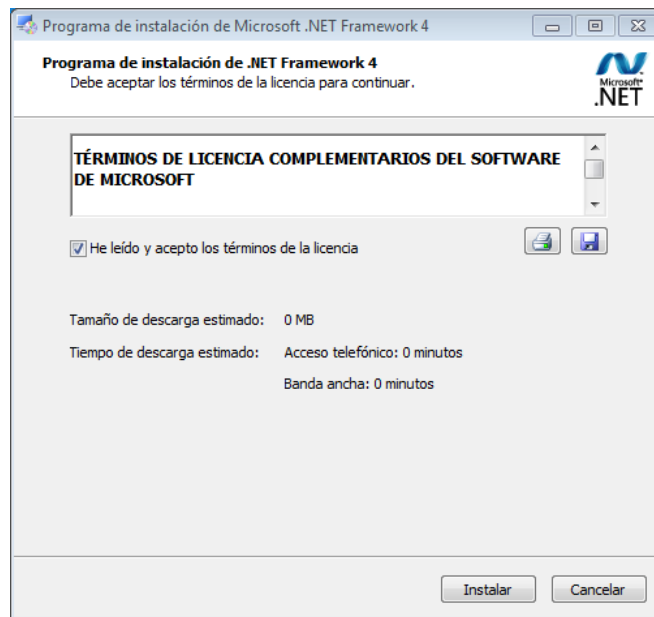
## Instalación de .NET Framework 4.0

.NET Framework es el modelo de programación completo y coherente de Microsoft para compilar aplicaciones que ofrezcan una sensacional experiencia visual del usuario, comunicación perfecta y segura, y la capacidad de modelar una amplia gama de procesos empresariales.

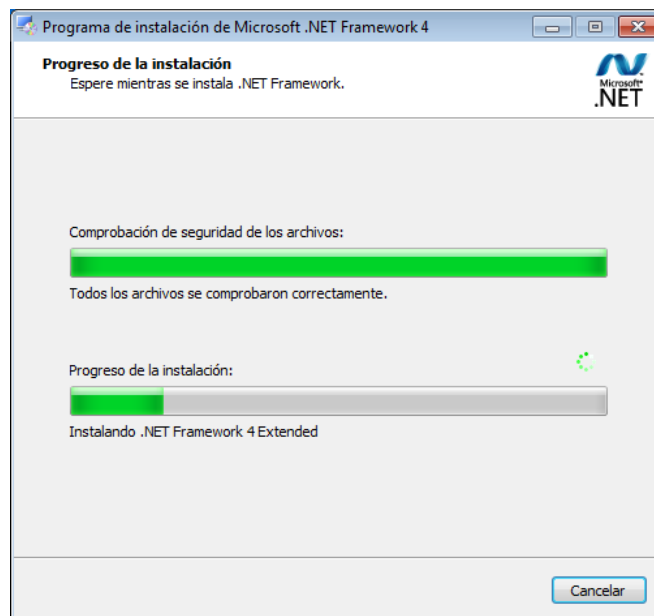
1. Realizar la descarga del instalador Microsoft .NET Framework 4.0 (independiente), mayor información en <http://www.microsoft.com/es-es/download/>, haga clic en descargar y guarde el archivo. Debe utilizarse la versión Full del .NET Framework 4.0, de lo contrario la instalación del SDK puede ocasionar errores.
2. En el explorador de Windows, busque la carpeta donde se realizó la descarga, presiona el botón derecho sobre el archivo dotNetFx40\_Full\_x86\_x64.exe, y luego ejecutar como administrador.



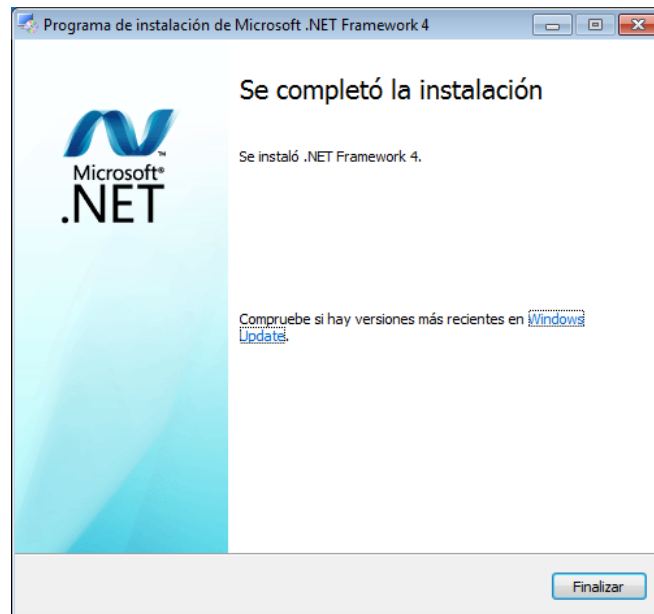
3. En la siguiente ventana, aceptamos los términos de la licencia, haga clic en instalar.



4. El programa comenzará con la instalación de los componentes del .NET Framework 4.0, mostrando una barra de progreso.



5. Por último, el programa de instalación muestra que se ha completado con éxito.



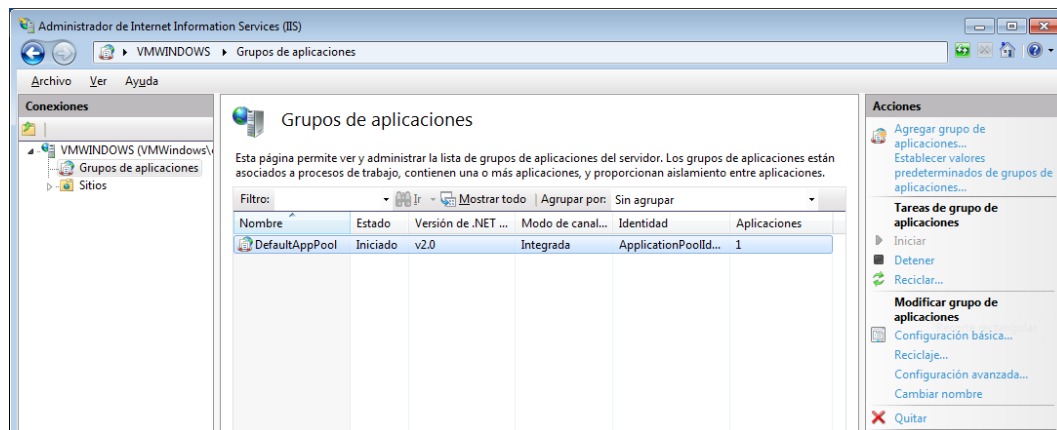
### Instalación de Internet Information Services (IIS).

Internet Information Services (IIS) 7.0 no se instala de forma predeterminada al instalar Windows. Para instalarlo con la configuración predeterminada, que únicamente suministra contenidos estáticos como archivos HTML y archivos de mapa de bits (.bmp), siga el procedimiento explicado a continuación:

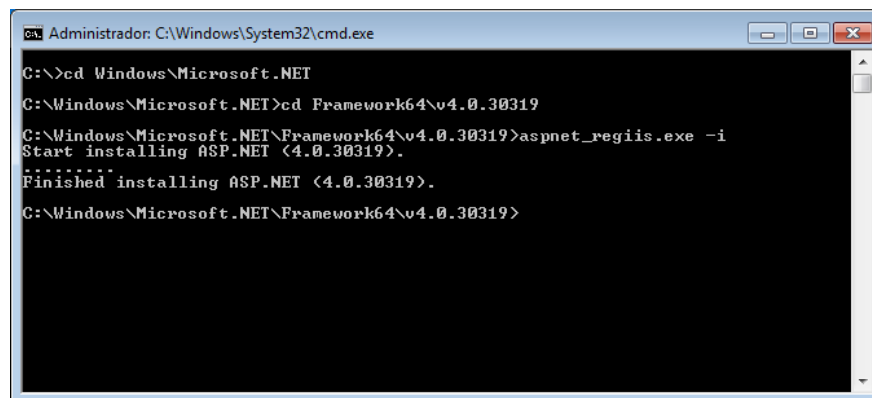
1. Haga clic en el botón Inicio, luego en Panel de control.
2. Haga clic en Programas, a continuación en Activar o desactivar las características de Windows. Se requiere permiso de administrador, si se le solicita una contraseña de administrador o una confirmación, escriba la contraseña o proporcione la confirmación.
3. En la lista de funciones de Windows marca la casilla: "Internet Information Services" y haga clic en Aceptar. Esta es la instalación predeterminada que incluye las características más comunes.



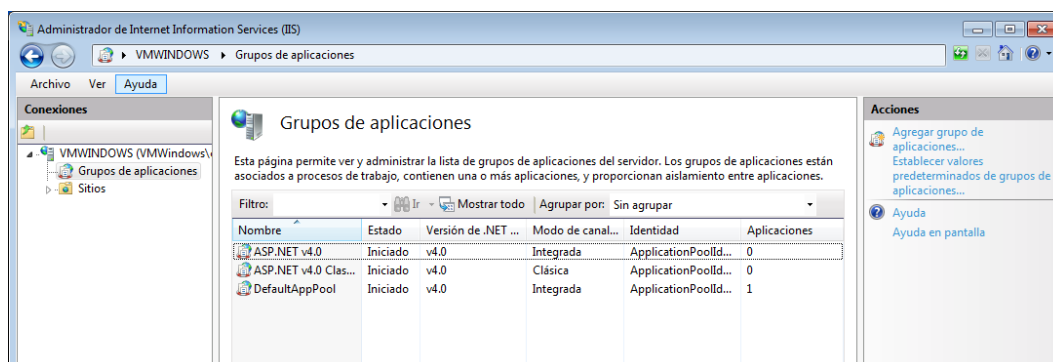
4. El sistema realizará los cambios necesarios. Esto puede tardar unos minutos. Seguidamente ejecute la consola de administración de IIS, haga clic en Grupo de aplicaciones del lado izquierdo de la pantalla, y verifique que versión de .NET está integrada al pool de aplicaciones por defecto (DefaultAppPool).



5. Si el pool de aplicaciones se encuentra apuntando a otra versión de .NET, debemos ejecutar los siguientes pasos:
  - a. Abrimos una ventana de consola en el modo administrador.
  - b. Ubicamos la siguiente dirección: C:\Windows\Microsoft.NET\Framework\v4.0xxx\ para versiones 32 bits, y C:\Windows\Microsoft.NET\Framework64\v4.0xxx\ para versiones de 64 bits, en este caso usaremos la de 64 bits.
  - c. Escribimos en la línea de comandos: **aspnet\_regiis.exe -i**. Luego que el sistema haga los cambios necesarios, cerramos la ventana.



6. Finalmente, ejecutamos nuevamente el IIS, y verificamos que efectivamente ahora el ASP.NET 4.0 se encuentra registrado dentro del IIS.

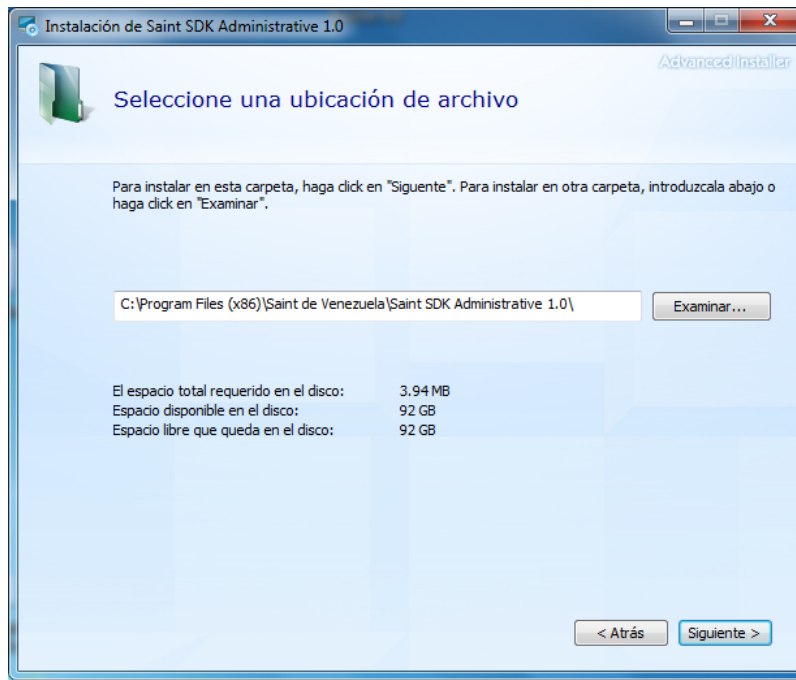


## Instalación del SDK Administrative v2.0.

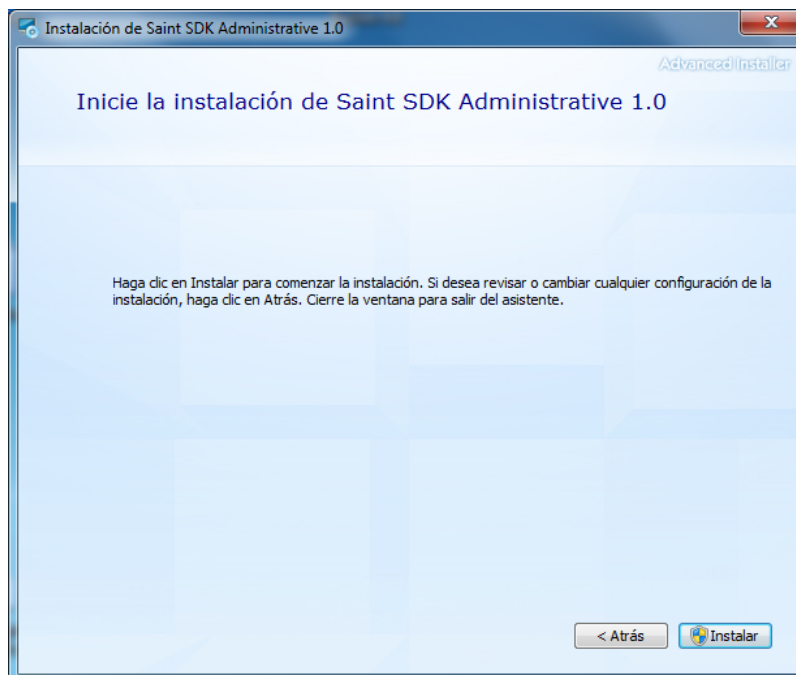
1. Para iniciar el proceso de instalación del SDK Administrative v2.0, haga doble clic sobre el archivo installSDK.exe.
2. En la ventana de bienvenida, haga clic en siguiente.



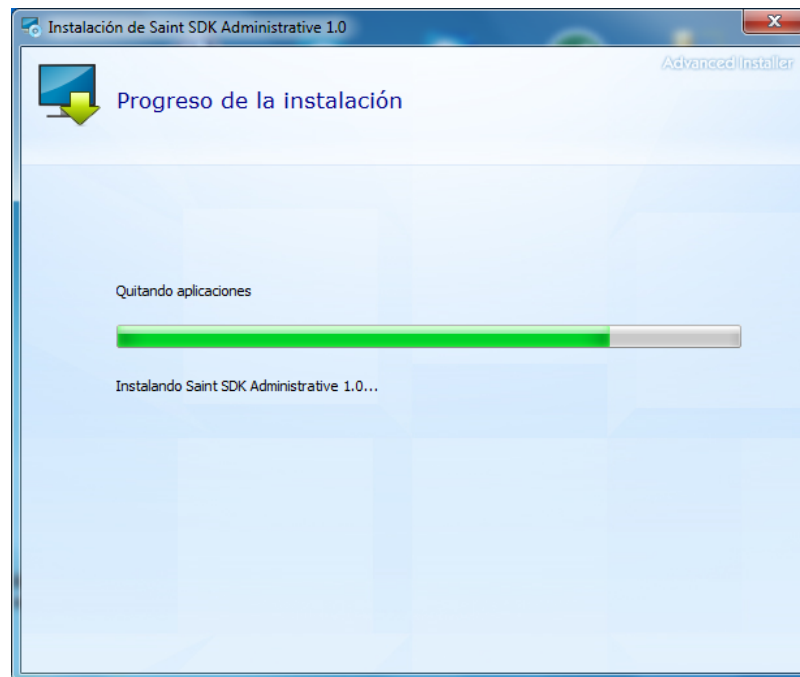
3. En la ventana selección de ubicación de archivo, escribe el directorio de destino, o haga clic en examinar y selecciona la carpeta donde deseas realizar la instalación. Se recomienda utilizar la ubicación por defecto.



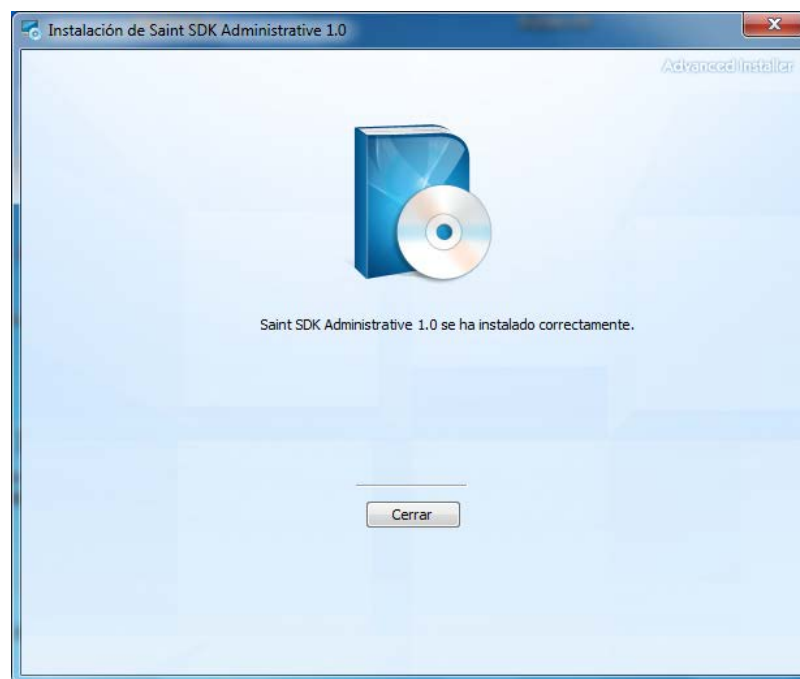
- Haga clic en instalar para iniciar el proceso de instalación del SDK Administrative v2.0. Se requiere permiso de administrador, si se le solicita una contraseña de administrador o una confirmación, escriba la contraseña o proporcione la confirmación. Si desea modificar algún parámetro de la configuración de instalación, haga clic en atrás.



5. La siguiente ventana permite acompañar el proceso de instalación de cada componente.



6. Instalación realizada con éxito. Haga clic en cerrar para terminar.







# Parte II

## Configuración

---

Esta sección describe la configuración del archivo `web.config`



# Configuración

El archivo web.config contiene la información relacionada con la conexión a la base de datos, dirección de la máquina y puerto que escucha el SLM, así como los datos de la licencias a utilizar.

En la sección `<applicationSettings>`, se realizarán las modificaciones necesarias para el buen funcionamiento del SDK. Es importante señalar que se realicen sólo los cambios que se mencionan a continuación, los cambios realizados en otras secciones del documento pueden generar errores inesperados en la ejecución del SDK.

Primero ubicamos el archivo web.config en la carpeta `c:\Archivos de Programa\Saint de Venezuela\ Saint SDK Administrative 1.0\`, haga clic con el botón derecho, y luego en abrir. Puede usar el bloc de notas, o el editor de su preferencia.

Escribimos la cadena de conexión que usará el SDK para conectarse a la base de datos.

```
<applicationSettings>
  <Saint.Web.Services.Properties.Settings>
    <setting name="DBServer" serializeAs="String">
      <value>Data Source=localhost;Initial Catalog=DBDEMO;Persist Security Info=True;User ID=ADMIN;Password=123</value>
    </setting>
  </Saint.Web.Services.Properties.Settings>
</applicationSettings>
```

Luego modificamos los parámetros para el funcionamiento del SLM, escribimos la dirección de la máquina donde se está ejecutando y el puerto que usa para escuchar.

```
<setting name="SLMServer" serializeAs="String">
  <value>localhost</value>
</setting>
<setting name="SLMPort" serializeAs="String">
  <value>10000</value>
</setting>
```

Por último, escribimos los valores de la licencia del producto. La licencia del SDK debe ser activada para su uso, consultar método `SLM_Activation`. El SDK a diferencia de otros productos saint, usa un Key adicional para la licencia master.

```
<setting name="LicenseSerial" serializeAs="String">
  <value>SDKA103171181811012</value>
</setting>
<setting name="LicenseKey" serializeAs="String">
  <value>1922132</value>
</setting>
<setting name="LicenseMaster" serializeAs="String">
  <value>0</value>
</setting>
<setting name="LicenseMasterKey" serializeAs="String">
  <value>0</value>
</setting>
```

# Activación del SDK

Luego de la instalación del SDK y configurar los parámetros en el archivo web.config, debemos realizar el siguiente paso para la activación del producto:

- Invocar el método SLM\_Activate(string **serial**, string **key**, string **product**, string **versión**)
  - **Serial**: Licencia válida del SDK. (Número de licencia configurado en el archivo web.config).
  - **Key**: Llave de activación (Número de key configurado en el archivo web.config).
  - **Product**: Nombre del producto, en este caso SDK en mayúsculas.
  - **Versión**: Número de versión del producto, 2.0.0.0

Este procedimiento debe ser realizado una sola vez.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.SLM_Activate("SDKA987654321012345", "1234567", "SDK", "2.0.0.0");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
```

# Parte III

## Métodos

---

Esta sección contiene la descripción detallada de los métodos, información de parámetros y valores devueltos, que integran al SDK Administrative v2.0, así como ejemplos ilustrados de la implementación.



## Región Infos

Esta región proporciona métodos para la construcción de consultas más específicas que se pueden realizar a la base de datos. Los métodos que conforman esta región son los siguientes:

- ✓ Info\_Global.
- ✓ Info\_Quantity.
- ✓ Info\_Querys.

### Info\_Global()

Info\_Global(*string* fields, *string* orders, *string* tables, *string* wheres, *int* qty, *int* start)

Construye una sentencia sql para realizar una consulta paginada a la base de datos, a partir de los parámetros enviados.

Parámetros	Tipo	Descripción
<i>fields</i>	<i>string</i>	Nombres de los campo(s).
<i>orders</i>	<i>string</i>	Establecer el orden de los resultados.
<i>tables</i>	<i>string</i>	Nombres de las tabla(s).
<i>wheres</i>	<i>string</i>	Sentencia(s) where aplicadas a la consulta.
<i>qty</i>	<i>int</i>	Cantidad de registro(s) que devuelve la consulta.
<i>start</i>	<i>int</i>	Número de fila que indica desde donde contar la cantidad de registro(s) a devolver.
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos:

En el siguiente ejemplo se muestra como llamar al método **Info\_Global**. Se realiza una consulta a la tabla de clientes, con los parámetros **qty** en cuatro (4) y **start** en uno (1). Es decir, el método devuelve la cantidad de cuatro registros, contados a partir del número 1.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Info_Global("CodClie", "Descrip", "Direc1", "CodClie", "SACLIE", "", 4, 1);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CodClie":"01434101","Descrip":"VISION 215427 C.A","Direc1":"DIRECCIÓN","ROW":1,"LASTROW":1},
 * {"CodClie":"04468968","Descrip":"JOSE PEDRO LOPEZ","Direc1":"DIRECCIÓN","ROW":2,"LASTROW":2},
 * {"CodClie":"05674253","Descrip":"MAURO CRESPO","Direc1":"LA VEGA, CALLE PPAL LAS CASITAS","ROW":3,"LASTROW":3},
 * {"CodClie":"07454148","Descrip":"ANGEL PEÑALOZA","Direc1":"DIRECCIÓN","ROW":4,"LASTROW":4}]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Info\_Quantity()

Info\_Quantity(string tables, string wheres)

Realiza una consulta, y devuelve el número de registros que cumpla con la cláusula where de la sentencia sql.

Parámetros	Tipo	Descripción
<i>tables</i>	string	Nombre(s) de la(s) tabla(s).
<i>wheres</i>	string	Sentencia where aplicada a la consulta.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: quantity

En el siguiente ejemplo se muestra cómo llamar al método **Info\_Quantity**.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Info_Quantity("SACLIE", " Descrip LIKE '%Cliente%'");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"QUANTITY":3}]
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Info\_Query()

Info\_Query(string fields, string tables, string wheres, string orders)

Realiza la construcción de una sentencia sql para realizar una consulta a la base de datos. Puede incluir diferentes campos de distintas tablas, la incorporación de cláusula where, y la ordenación de los registros.

Parámetros	Tipo	Descripción
<i>fields</i>	string	Nombres de los campos.
<i>tables</i>	string	Nombres de las tablas.
<i>wheres</i>	string	Sentencias “where” aplicadas.
<i>orders</i>	string	Establece el orden de los resultados.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: Todos los incluidos en <b>fields</b> .



En el siguiente ejemplo se muestra como llamar al método **Info\_Query**.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Info_Query("CodProd, Descrip, Precio1, Precio2, Precio3", "SAPROD", "CodProd = '31410204'", "");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CodProd":"31410204","Descrip":"TORN. HEX. M10 X 1.5 X 25MM CLASE 8.8","Precio1":8.16000,
 *  "Precio2":5.98000,"Precio3":5.98000}]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

# Región InitProcess

Contiene los métodos de inicialización de una operación (factura, devolución, nota de entrega, devolución nota de entrega, pedido, presupuesto), por ejemplo: determinar los campos adicionales necesarios en la operación, obtener el número correlativo de un documento. Los siguientes métodos están incluidos en esta región:

- ✓ Init\_AdditionalFields.
- ✓ Init\_CorrelatesOperation.
- ✓ Init\_CorrelatesFact.

## Init\_AdditionalFields()

Init\_AdditionalFields (int typeOperation)

Devuelve una lista de los campos adicionales relacionados para un tipo de operación. (factura, devolución, nota de entrega, devolución nota de entrega, pedido, presupuesto).

Parámetros	Tipo	Descripción
<i>typeOperation</i>	int	Valor que representa el tipo de operación. (Factura = 0, Devolución = 1, Nota de Entrega = 2, Devolución nota de entrega = 3, Pedido = 4, Presupuesto = 5).
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: tblname, codtbl, tipocpo, nrooper, numgrp, aliasgrp, nombcpo, aliascpo, longitud, requerido, colorder.

En el siguiente ejemplo se muestra el método **Init\_AdditionalFields**. En el código de ejemplo se pasa el número cuatro (4) como parámetro, y se retornan los campos adicionales para la operación de un pedido.

```
//Init_AdditionalFields()
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Init_AdditionalFields(4);
        _result.ToString();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"TBLNAME":"SACLIE","CODTBL":"01","TIPOPCPO":2,"NROOPER":"333","NUMGRP":1,"ALIASGRP":"GRUPO01",
 * "NOMBPCPO":"NUMEROADICIONAL","ALIASCPO":"NUMADIC","LONGITUD":25,"REQUERIDO":1,"COLORDER":0},]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Init\_CorrelatesOperation()

Init\_CorrelatesOperation(string codSuc, string fieldName)

Retorna el número correlativo de la operación a realizar.

Parámetros	Tipo	Descripción
<i>codSuc</i>	string	Número de la sucursal.
<i>fieldName</i>	string	Prefijo del documento que se desea consultar.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: prefijo, correlativo.

En el siguiente ejemplo se muestra como llamar al método **Init\_CorrelatesOperation**. El valor devuelto es el número de identificación para un pedido.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Init_CorrelatesOperation("00000", "PrxPedido");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"PREFIJO":"","CORRELATIVO":3802}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Init\_CorrelatesFact()

Init\_CorrelatesFact(string codSuc, string fieldName)

Retorna los números para los correlativos de facturación.

Parámetros	Tipo	Descripción
<i>codSuc</i>	string	Número de la sucursal.
<i>fieldName</i>	string	Prefijo del documento que se desea consultar.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: lenght.

En el siguiente ejemplo se muestra el método **Init\_CorrelatesFact**. El valor del resultado, se refiere al número correlativo de una operación de facturación.

```

using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Init_CorrelatesFact("00000", "PrxPedido");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"PREFIXO":"","CORRELATIVO":003802}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */

```

El parámetro *fieldName*, que recibe los métodos **Init\_CorrelatesOperation()** y **Init\_CorrelatesFact()**, se refiere al prefijo que usa en los tipos de documentos con el cual estamos trabajando. Por ejemplo:

NumPedido: Numero de pedido,  
 PrxCotiza: Cotización,  
 PrxFact: Factura,  
 PrxNotaCr: Nota de crédito,  
 PrxNotaDb: Nota de débito,  
 PrxOrdenC: Orden de compra,  
 PrxPedido: Pedido,  
 PrxODPago: Orden de pago, entre otros.

La tabla SACORRELSIS, contiene todos los prefijos que se pueden usar.

# Región HeaderProcess

Esta sección contiene los métodos necesarios para construir el encabezado de un documento, por ejemplo: la información del cliente, convenios asociados, datos del vendedor, entre otros. Los métodos incluidos son:

- ✓ Header\_Customer.
- ✓ Header\_AddCustomer.
- ✓ Header\_AdditionalReturns.
- ✓ Header\_CustomerAgreements.
- ✓ Header\_Seller.
- ✓ Header\_Deposit.
- ✓ Header\_Suppliers.
- ✓ Header\_Supplier.
- ✓ Header\_AddSupplier.
- ✓ Header\_SaveSupplier.
- ✓ Header\_DelSupplier.
- ✓ Header\_DelCustomer.
- ✓ Header\_SaveCustomer.

## Header\_Customer()

Header\_Customer([string](#) codClie)

Obtiene la información específica de un cliente.

Parámetros	Tipo	Descripción
<i>codClie</i>	<a href="#">string</a>	Código del cliente.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: tipocli, descrip, id3, activo, direc1, direc2, telef, codconv, esmoneda, codvend, tipopvp, diascred, limitecred, escredito, codalte, saldo.

El ejemplo muestra la forma de llamar al método **Header\_Customer**. Se envía como parámetro al método, el número de identificación de un cliente y este devuelve los datos del mismo.

```

using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_Customer("483355610-9");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"TIPOCLI":0,"DESCRIP":"PEDRO PEREZ","ID3":"V-483355610-9","ACTIVO":1,
 * "DIREC1":"LAGUNILLAS CAMPO 1 TERMINAL","DIREC2":null,"TELEF":"0416-6650037","CODCONV":null,
 * "ESMONEDA":0,"CODVEND":null,"TIPOPOPVP":3,"DIASCRED":0,"LIMITECRED":0.0000,"ESCREDITO":0,
 * "CODALTE":null,"SALDO":0.0000,"DESCRIPEXT":"PEDRO PEREZ"}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */

```

## Header\_AddCustomer()

Header\_AddCustomer([string](#) jsonClie)

Permite añadir un cliente en una operación de venta.

Parámetros	Tipo	Descripción
<i>jsonClie</i>	<a href="#">string</a>	Cadena de texto, que contiene los datos del cliente, en formato json.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: status.

En el siguiente ejemplo se muestra el método **Header\_AddCustomer**. En el código, la variable *\_jsonClie* es una cadena de texto con el formato Json que representa los datos del cliente que se desea insertar, este valor se envía como parámetro al método.

```

using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _jsonClie = "{ \"CodClie\": \"09998899\", \"Descrip\": \"INSERTAR CLIENTE NUEVO\", \"ID3\": \"J12345678\",
        \": \"CLIENTE\", \"CodConv\": \"30\", \"Represent\": \"CLIENTE NUEVO\", \"Direc1\": \"DIRECCIÓN PRINCIPAL\", \"Direc2\": \"DIRECCIÓN SECUNDARIA\", \"Pais\": \"58\", \"Estado\": \"12\", \"Ciudad\": \"2\", \"Municipio\": \"9\", \"ZipCode\": \"1040\", \"Telef\": \"0444-12345678\", \"Movil\": \"999-9999123\", \"Fax\": \"888-1234567\", \"Email\": \"cliente@tu-email.com\", \"FechaE\": \"2011-09-17T11:56:07\", \"Observa\": \"CLIENTE NUEVO\", \"TipoCli\": \"0\", \"Activo\": \"1\" }";
        string _result = string.Empty;

        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_AddCustomer(_jsonClie);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"status":"OK"}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */

```

## Header\_AdditionalReturns()

Header\_AdditionalReturns([string](#) codClie) (Método)

Retorna una lista de documentos pendientes por cobrar de un cliente.

Parámetros	Tipo	Descripción
<i>codClie</i>	<a href="#">string</a>	Código del cliente.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: saldo, monto, saldomex, baseimpo, mtotax, tipocxc, numerod, document, codvend, fecha, fechav, nrounico, comision.

El ejemplo muestra al método **Header\_AdditionalReturns**.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_AdditionalReturns("000341168");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"SALDO":3650.0000,"MONTO":3650.0000,"SALDOMEX":0.0000,"BASEIMPO":3258.9300,
 * "MTOTAX":391.0700,"TIPOCX":10,"NUMEROD":"00005677","DOCUMENT":"00005677",
 * "CODVEND":"01","FECHAE":"2011-09-10T11:56:06.967","FECHAV":"2011-09-17T11:56:07",
 * "NROUNICO":11680,"COMISION":0.0000}]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Header\_CustomerAgreements()

Header\_CustomerAgreements([string](#) codConv)

Devuelve una lista de ofertas y convenios asociados a un cliente.

Parámetros	Tipo	Descripción
<i>codConv</i>	<a href="#">string</a>	Código del convenio.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codconv, codsucu, descrip, autori, respon, fecha, fechav, frecuencia, fechauc, esfijo, tipocnv, esbase, activo.

En el siguiente ejemplo se muestra como llamar al método **Header\_CustomerAgreements**. Se pasa como parámetro el código de un convenio.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_CustomerAgreements("1");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODCONV": "1", "CODSUCU": "00000", "DESCRIP": "CONVENIO_01", "AUTORI": "AUTORIZADO", "RESPON": "RESPONSABLE",
 * "FECHAE": "2013-01-01T12:00:00", "FECHAV": "2013-03-31T00:00:00", "FRECUENCIA": 0, "FECHAUC": "2013-01-15T00:00:00",
 * "ESFIJO": 0, "TIPOCNV": 2, "ESBASE": 1, "ACTIVO": 1}]
 * Mensaje en caso de error: {"error": "Desconocido"}
 */
```

## Header\_Seller()

Header\_Seller(string codVend)

Este método devuelve la información específica de un vendedor.

Parámetros	Tipo	Descripción
<i>codVend</i>	string	Código del vendedor.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: descrip, escomipv, activo, codvend, tiporgo, nrounico, desde, hasta, monto.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_Seller("03");
        Console.WriteLine(_result);
        Console.Read();
    }
}

/* Este código genera el siguiente resultado:
 * [{"DESCRIP": "RODRIGUEZ QUIROZ EDWIN", "ESCOMIPV": 1, "ACTIVO": 1,
 * "CODVEND": "03", "TIPOARGO": 0, "NROUNICO": 0, "DESDE": 0.0000,
 * "HASTA": 0.0000, "MONTO": 1.0000}]
 * Mensaje en caso de error: {"error": "Desconocido"}
 */
```



## Header\_Deposit()

Header\_Deposit(string codUbic)

Devuelve la información de un depósito.

Parámetros	Tipo	Descripción
<i>codUbic</i>	string	Código del depósito.
<b>Valor devuelto</b>	string	Cadena de texto con formato Json. Campos: codubic, descrip, codsucu, clase, activo, represent, direc1, direc2, telef, pais, estado, ciudad, esvirtual, codalte, printer, municipio, zipcode.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Header_Deposit("01");
        Console.WriteLine(_result);
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"CODUBIC":"01","DESCRIP":"DETAL","CODSUCU":"00000","CLASE":null,
 * "ACTIVO":1,"REPRESENT":null,"DIREC1":null,"DIREC2":null,"TELEF":null,
 * "PAIS":0,"ESTADO":0,"CIUDAD":0,"ESVIRTUAL":0,"CODALTE":null,"PRINTER":0,
 * "MUNICIPIO":0,"ZIPCODE":null}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Header\_Suppliers()

Header\_Suppliers(int page, int pageSize)

Devuelve una lista de proveedores.

Parámetros	Tipo	Descripción
<i>page</i>	int	Número de página.
<i>pageSize</i>	int	Número de filas por página..
<b>Valor devuelto</b>	string	Cadena de texto con formato Json. Campos: fila, codprov, descrip, codsucu, tipopr, id3, clase, activo, represent, direc1, direc2, pais, estado, ciudad, telef, fax, nit, email, fecha, observa, esmoneda,

diascred, saldo, montomax, pagosa, prompago, reteniva, fechauc, montouc, numerouc, fechaup, montoup, numeroup, edokc, movil, esreten, tipoid3, descorder, retenislr, municipio, zipcode, tipoid.

### Código de ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Header_Suppliers(1, 2);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
/* Este código genera el siguiente resultado:
* [{"FILA":1,"CODPROV":"00000000","DESCRIP":"INCAVEL ONIBUS E PECAS LTDA","CODSUCU":"","TIPOPRV":1,"ID3":"00000000",
* "CLASE":"CHASIS","ACTIVO":1,"REPRESENT":"Olávio Viecke Dias","DIREC1":"RUA BELCZAK Nº ABC477",
* "DIREC2":"CRISTO REI, CIUDAD DE C, -PR, BRASIL","PAIS":0,"ESTADO":0,"CIUDAD":0,
* "TELEF":"55 4999625400 FAX 554133626016","FAX":"","NIT":"","EMAIL":"exportacion@email.com.br","FECHAE":"2011-03-18",
* "OBSERVA":"","ESMONEDA":0,"DIASCREd":0,"SALDO":0.000,"MONTOMAX":0.000,"PAGOSA":0.000,"PROMPAGO":0.000,"RETENIVA":0.000,
* "FECHAUC":"","MONTouc":0.000,"NUMERouc":"","FECHAUP":"","MONToup":0.000,"NUMERoup":"","EDOKC":"","MOVIL":"","ESREten":0,"TIPOID3":0,"DESCORDER":"","RETENISLR":0,"MUNICIPIO":0,"ZIPCODE":"80.050-570","TIPOID":0}, {"FILA":2,
* "CODPROV":"05137739","DESCRIP":"ANA MARIA","CODSUCU":"","TIPOPRV":0,"ID3":"05137739","CLASE":"","ACTIVO":1,
* "REPRESENT":"","DIREC1":"","DIREC2":"","PAIS":0,"ESTADO":0,"CIUDAD":0,"TELEF":"","FAX":"","NIT":"","EMAIL":"","
* "FECHAE":"2013-04-30","OBSERVA":"","ESMONEDA":0,"DIASCREd":0,"SALDO":0.000,"MONTOMAX":0.000,"PAGOSA":0.000,
* "PROMPAGO":0.000,"RETENIVA":0.000,"FECHAUC":"","MONTouc":0.000,"NUMERouc":"","FECHAUP":"","MONToup":0.000,
* "NUMERoup":"","EDOKC":"","MOVIL":"","ESREten":1,"TIPOID3":1,"DESCORDER":"13","RETENISLR":0,"MUNICIPIO":0,"ZIPCODE":"","
* "TIPOID":0}]
*/
/* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
}
```

## Header\_Supplier()

Header\_Supplier(string codProv)

Devuelve información de un proveedor.

Parámetros	Tipo	Descripción
<i>codProv</i>	string	Código del proveedor.
Valor devuelto	string	Cadena de texto con formato Json. Campos: codprov, descrip, codsucu, tipopr, id3, clase", activo, represent, direc1, direc2, pais, estado, ciudad, telef, fax, nit, email, fecha, observa, esmoneda, diascred, saldo, montomax, pagosa, prompago, reteniva, fechauc, montouc, numerouc, fechaup, montoup, numeroup, edokc, movil, esreten, tipoid3, descorder, retenislr, municipio, zipcode, tipoid.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Header_Supplier("01238899");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"CODPROV":"01238899","DESCRIP":"ANA MENDOZA","CODSUCU":"","TIPOPRV":0,"ID3":"05137739","CLASE":"","ACTIVO":1,
 * "REPRESENT":"","DIREC1":"","DIREC2":"","PAIS":0,"ESTADO":0,"CIUDAD":0,"TELEF":"","FAX":"","NIT":"","EMAIL":"","
 * "FECHAE":"2013-04-30","OBSERVA":"","ESMONEDA":0,"DIASCRE":0,"SALDO":0.000,"MONTOMAX":0.000,"PAGOSA":0.000,
 * "PROMPAGO":0.000,"RETENIVA":0.000,"FECHAUC":"","MONTOUN":0.000,"NUMEROUC":"","FECHAUP":"","MONTOUN":0.000,"NUMEROUP":"","
 * "EDOKC":"","MOVIL":"","ESRETE":1,"TIPOID3":1,"DESCORDER":"13","RETENISLR":0,"MUNICIPIO":0,"ZIPCODE":"","TIPOID":0}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Header\_AddSupplier()

Header\_AddSupplier(string jsonSupplier)

Inserta un nuevo proveedor al sistema.

Parámetros	Tipo	Descripción
<i>jsonSupplier</i>	string	Cadena de texto en formato Json (Proveedor).
Valor devuelto	string	Cadena de texto con formato Json. Campos: status.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        string proveedor = "{ \"CODPROV\": \"V099123456\", \"DESCRIP\": \"NUEVA\", \"CODSUCU\": \"\", \"TIPOPRV\": 1, "+
        "\"ID3\": \"V014550489\", \"CLASE\": \"1234567890\", \"ACTIVO\": 1, \"REPRESENT\": \"QWERTYUIOP\", \"DIREC1\": \"\", "+
        "\"DIREC2\": \"\", \"PAIS\": 58, \"ESTADO\": 20, \"CIUDAD\": 17, \"TELEF\": \"0273123456\", \"FAX\": \"0273123456\", "+
        "\"NIT\": \"0123456789\", \"EMAIL\": \"MIEMAIL@SERVER.COM\", \"FECHAE\": \"2013-01-01\", \"OBSERVA\": \"\", \"ESMONEDA\": 0, "+
        "\"DIASCRE\": 15, \"SALDO\": 1300, \"MONTOMAX\": 10000, \"PAGOSA\": 0, \"PROMPAGO\": 0, \"RETENIVA\": 0, \"FECHAUC\": \"\" "+
        "\"2013-01-01\", \"MONTOUN\": 0, \"NUMEROUC\": \"123\", \"FECHAUP\": \"2013-01-01\", \"MONTOUN\": 0, \"NUMEROUP\": \"098\", "+
        "\"EDOKC\": \"999\", \"MOVIL\": \"\", \"ESRETE\": 0, \"TIPOID3\": 1, \"DESCORDER\": \"\", \"RETENISLR\": 1, "+
        "\"MUNICIPIO\": 10, \"ZIPCODE\": 5201, \"TIPOID\": 5}";

        _result = _serviceClient.Header_AddSupplier(proveedor);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"status":"OK" }
 *
 * En caso de que existe el registro:
 *
 * {"status":"Ya existe un proveedor registrado con ese código."}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Header\_SaveSupplier()

Header\_SaveSupplier([string](#) jsonSupplier)

Permite modificar los datos de un proveedor existente en el sistema.

Parámetros	Tipo	Descripción
<i>jsonSupplier</i>	<a href="#">string</a>	Cadena de texto en formato Json (Proveedor).
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto con formato Json. Campos: status.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        string proveedor = "{ \"CODPROV\": \"V099123456\", \"DESCRIP\": \"MODIFICAR PROVEEDOR\", \"CODSUCU\": \"\", \"TIPOPRV\": 1, \"ID3\": \"V014550489\", \"CLASE\": \"1234567890\", \"ACTIVO\": 1, \"REPRESENT\": \"QWERTYUIOP\", \"DIREC1\": \"\", \"DIREC2\": \"\", \"PAIS\": 58, \"ESTADO\": 20, \"CIUDAD\": 17, \"TELEF\": \"0273123456\", \"FAX\": \"0273123456\", \"NIT\": \"0123456789\", \"EMAIL\": \"MIEMAIL@SERVER.COM\", \"FECHAE\": \"2013-01-01\", \"OBSERVA\": \"\", \"ESMONEDA\": 0, \"DIASCREDO\": 15, \"SALDO\": 1300, \"MONTOMAX\": 10000, \"PAGOSA\": 0, \"PROMPAGO\": 0, \"RETENIVA\": 0, \"FECHAUC\": \"\", \"2013-01-01\", \"MONTUOC\": 0, \"NUMEROUC\": \"123\", \"FECHAUP\": \"2013-01-01\", \"MONTUOP\": 0, \"NUMEROU\": \"098\", \"EDOKC\": \"999\", \"MOVIL\": \"\", \"ESRETEH\": 0, \"TIPOID3\": 1, \"DESCORDER\": \"\", \"RETENISLR\": 1, \"MUNICIPIO\": 10, \"ZIPCODE\": 5201, \"TIPOID\": 5 }";

        _result = _serviceClient.Header_SaveSupplier(proveedor);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * { "status": "OK" }
 */
/* Mensaje en caso de error: "{ \"error\": \"Desconocido\"}" */
```

## Header\_DelSupplier()

Header\_DelSupplier([string](#) codProv)

Permite eliminar a un proveedor existente en el sistema.

Parámetros	Tipo	Descripción
<i>codProv</i>	<a href="#">string</a>	Código del proveedor.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto con formato Json. Campos: status.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Header_DelSupplier("V099123456");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"status":"OK" }           Si el registro ha sido eliminado
    * {"status":"No se puede eliminar"} Si el registro no ha sido eliminado
    */
    /* Mensaje en caso de error: "{"error":"Desconocido"}" */
}
```

## Header\_DelCustomer()

Header\_DelCustomer(string codClie)

Permite eliminar a un cliente existente en el sistema.

Parámetros	Tipo	Descripción
<i>codClie</i>	string	Código del cliente.
<b>Valor devuelto</b>	string	Cadena de texto con formato Json. Campos: status.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Header_DelCustomer("99888938");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"status":"OK" }           Si el registro ha sido eliminado
    * {"status":"No se puede eliminar"} Si el registro no ha sido eliminado
    */
    /* Mensaje en caso de error: "{"error":"Desconocido"}" */
}
```

## Header\_SaveCustomer()

Header\_SaveCustomer(string jsonClie)

Permite modificar los datos de un cliente existente en el sistema.

Parámetros	Tipo	Descripción
<i>jsonClie</i>	string	Cadena de texto en formato Json (Cliente).
Valor devuelto	string	Cadena de texto con formato Json. Campos: status.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        string _jsonClie = "{ \"CodClie\": \"044615800\", \"Descrip\": \"MODIFICAR CLIENTE \", \"ID3\": \"J12345678\", \"Clase\": \"CLIENTE\", \"CodConv\": \"30\", \"Represent\": \"CLIENTE NUEVO\", \"Direc1\": \"DIRECCIÓN PRINCIPAL\", \"Direc2\": \"DIRECCIÓN SECUNDARIA\", \"Pais\": \"58\", \"Estado\": \"12\", \"Ciudad\": \"2\", \"Municipio\": \"9\", \"ZipCode\": \"1040\", \"Telef\": \"0444-12345678\", \"Movil\": \"999-9999123\", \"Fax\": \"888-1234567\", \"Email\": \"cliente@tu-email.com\", \"FechaE\": \"2011-09-17T11:56:07\", \"Observa\": \"CLIENTE NUEVO\", \"TipoCli\": \"0\", \"Activo\": \"1\" }";

        _result = _serviceClient.Header_SaveCustomer(_jsonClie);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"status":"OK" }           Si los datos son modificados
    * {"status":"No encontrado" } Si el cliente no existe
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Región ItemsProcess

Contiene los métodos aplicados a los ítems de un documento, por ejemplo: determinar los impuestos del ítem, buscar un producto o servicio. A continuación los métodos son los siguientes:

- ✓ Item\_FindAll.
- ✓ Item\_OffersAgreements.
- ✓ Item\_CalculatedTax.
- ✓ Item\_Product.
- ✓ Item\_Products.
- ✓ Item\_ProductsLoc.
- ✓ Item\_PartsProduct.
- ✓ Item\_Taxes.
- ✓ Item\_Service.
- ✓ Item\_ProdExistence.
- ✓ Item\_Lote.
- ✓ Item\_Serials.
- ✓ Item\_ProductRef.

### Item\_FindAll()

Item\_FindAll(**string** codFind)

Obtiene información de un producto o servicio. Realiza una búsqueda en la base de datos, donde el valor del parámetro enviado, coincida con cualquier campo en la tabla ya sea producto o servicio.

Parámetros	Tipo	Descripción
<i>codFind</i>	<b>string</b>	Valor de búsqueda que coincida con algún campo de la tabla de un producto o servicio.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: esserv, [codprod   codserv], descrip.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_FindAll("CAMIÓN");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/*
 * Este código genera el siguiente resultado:
 * [{"ESSERV":0,"CODPROD":"10018402","DESCRIP":"CAMIÓN FORD CARGO 4532"},
 * {"ESSERV":0,"CODPROD":"10018403","DESCRIP":"CAMIÓN FORD CARGO 1721 CHASIS LARGO 18CD"},
 * {"ESSERV":0,"CODPROD":"10018404","DESCRIP":"CAMIÓN FORD CARGO 2632"},
 * {"ESSERV":0,"CODPROD":"10018406","DESCRIP":"CAMIÓN FORD 350, 4 X 4, SINCRONICO 8 CIL"},
 * {"ESSERV":0,"CODPROD":"28C5","DESCRIP":"CAMIÓN CARGA CON CAJON TRASERO 3 COMPART"},
 * {"ESSERV":0,"CODPROD":"28C6","DESCRIP":"CAMIÓN CARGA COMPUERTA TRASERA ABATIBLE"},
 * {"ESSERV":0,"CODPROD":"28C7","DESCRIP":"CAMIÓN DE CARGA TRASLADO Y/O RESCATE"}]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_OffersAgreements()

Item\_OffersAgreements(string codInv, string codClie, string codConv, decimal montoBase, decimal cantidad, string fechaE)

Calcula y devuelve el valor final del producto o servicio, de acuerdo a la oferta y/o del convenio que se encuentre asociado.

Parámetros	Tipo	Descripción
<i>codInv</i>	string	Código de inventario del producto o servicio.
<i>codClie</i>	string	Código del cliente.
<i>codConv</i>	string	Código del convenio.
<i>montoBase</i>	decimal	Monto del ítem.
<i>Cantidad</i>	decimal	Cantidad del ítem.
<i>fechaE</i>	string	Fecha de emisión.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: precio.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_OffersAgreements("02615633", "0123456", "0056", 4526.96M, 1M, "");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"PRECIO":4105.26}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_CalculatedTax()

Item\_CalculatedTax(string codInv, decimal precio, decimal cantidad, int esServ, int esUnid)

Calcula y retorna el impuesto asociado a un ítem.

Parámetros	Tipo	Descripción
<i>codInv</i>	string	Código de inventario del producto o servicio.
<i>precio</i>	decimal	Precio del ítem.
<i>cantidad</i>	decimal	Cantidad del ítem.
<i>esServ</i>	int	Determina si el ítem es un servicio. (0 = no; 1 = sí)



<i>esUnid</i>	<b>Int</b>	Determina si el ítem es unidad. (0 = no; 1 = sí)
---------------	------------	--

<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: mtotax.
-----------------------	---------------	---

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient serviceSEA = new AdminServiceClient();
        var _result = serviceSEA.Item_CalculatedTax("24C12", 1536.98M, 2.3M, 0, 0);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"MTOTAX":636.30972}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_Product()

Item\_Product(**string** codProd)

Devuelve la información sobre un único producto.

Parámetros	Tipo	Descripción
<i>codProd</i>	<b>string</b>	Código del producto.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: codprod, codinst, descorrel, digitosc, refere, desseri, descomp, esexento, exdecimal, unidad, undempa, deslote, espesa, tara, desvence, descto, cantempa, peso, volumen, undvol, costpro, descrip, descrip2, descrip3, preciou, preciou2, preciou3, precio1, precio2, precio3, isadic, costo.

## Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_Product("000063");
        Console.WriteLine(_result);
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"CODPROD":"000063","CODINST":13,"DESCORREL":0,"DIGITOSC":0,"REFERE":"","DESSERTI":0,
 * "DESCOMP":0,"ESEXENTO":0,"EXDECIMAL":0,"UNIDAD":"PZA","UNDEMPAQ":"","DESLOTE":0,
 * "ESPESA":0,"TARA":0.0000,"DESVENTE":0,"DESCTO":0.0000,"CANTEMPAQ":0.0000,"PESO":0.0000,
 * "VOLUMEN":0.0000,"UNDEVOL":"","COSTPRO":28.6650,"DESCRIP":"COMPRESOR AIRE 250 LIBRAS M",
 * "DESCRIP2":"","DESCRIP3":"","PRECIU":0.00000,"PRECIU2":0.00000,"PRECIU3":0.00000,
 * "PRECI1":61.60710,"PRECI2":61.60710,"PRECI3":61.60710,"ISADIC":0,"COSTO":28.6650}"
 *
 * Mensaje en caso de error: {"error":"Desconocido"}"
 */
```

## Item\_Products()

Item\_Products(int isEnser, string order, int page, int pageSize)

Devuelve una lista de los productos existentes.

Parámetros	Tipo	Descripción
<i>isEnser</i>	int	Si el producto es enser
<i>order</i>	string	Condición de filtrado para los resultados (where)
<i>page</i>	int	Número de página
<i>pageSize</i>	int	Número de registros por página.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: fila, codprod, descrip, descrip2, descrip3, descripall, esfreep, precio1, precio2, precio3, preciou2, preciou3, refere, marca, unidad, activo, undempaq, cantempaq, preciou, costact, costpro, costant, peso, volumen, undvol, existen, exunidad, compro, pedido, minimo, maximo, desvence, espesa, tara, esimport, esexento, esenser, esempaque, exdecimal, desccto, descomp, desseri, deslote, descomi, descorrel, digitosc, codinst, countlote, countseri, countpartes.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_Products(0, "", 1, 2);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"FILA":1,"CODPROD":"10018402","DESCRIP":"CAMIÓN FORD CARGO 4532","DESCRIP2":"","DESCRIP3":"","DESCRIPALL":"CAMIÓN FORD CARGO 4532",
 * "ESFREEP":0,"PRECIO1":210600.00000,"PRECIO2":210600.21000,"PRECIO3":210600.21000,"PRECIU2":0.00000,"PRECIU3":0.00000,"REFERE":"","
 * "MARCA":"","UNIDAD":"PZA","ACTIVO":1,"UNDEMPAQ":"","CANTEMPAQ":0.000,"PRECIU":0.00000,"COSTACT":0.000,"COSTPRO":0.000,
 * "COSTANT":0.000,"PESO":0.000,"VOLUMEN":0.000,"UNVOL":"","EXISTEN":0.000,"EXUNIDAD":0.000,"COMPRO":0.000,"PEDIDO":0.000,
 * "MINIMO":0.000,"MAXIMO":0.000,"DESVENTE":0,"ESPESA":0,"TARA":0.000,"ESIMPORT":0,"ESEXENTO":0,"ESENSER":0,"ESEMPAQUE":0,"EXDECIMAL":0,
 * "DESCTO":0.000,"DESCOMP":0,"DESSERTI":0,"DESLOTE":0,"DESCOMI":0,"DESCORREL":0,"DIGITOSC":1,"CODINST":82,"COUNTLOTE":0,"COUNTSERI":0,
 * "COUNTPARTES":0},{{"FILA":2,"CODPROD":"10018403","DESCRIP":"CAMIÓN FORD CARGO 1721 CHASIS LARGO 18CD","DESCRIP2":"","DESCRIP3":"","
 * "DESCRIPALL":"CAMIÓN FORD CARGO 1721 CHASIS LARGO 18CD",
 * "ESFREEP":0,"PRECIO1":163577.98000,"PRECIO2":180000.00000,
 * "PRECIO3":160000.00000,"PRECIU2":0.00000,"PRECIU3":0.00000,"REFERE":"","MARCA":"","UNIDAD":"UND","ACTIVO":1,"UNDEMPAQ":"","
 * "CANTEMPAQ":0.000,"PRECIU":0.00000,"COSTACT":422321.430,"COSTPRO":422321.430,"COSTANT":202770.990,"PESO":0.000,"VOLUMEN":0.000,
 * "UNVOL":"","EXISTEN":0.000,"EXUNIDAD":0.000,"COMPRO":0.000,"PEDIDO":-1.000,"MINIMO":0.000,"MAXIMO":0.000,"DESVENTE":0,"ESPESA":0,
 * "TARA":0.000,"ESIMPORT":0,"ESEXENTO":0,"ESENSER":0,"ESEMPAQUE":0,"EXDECIMAL":0,"DESCTO":0.000,"DESCOMP":0,"DESSERTI":0,"DESLOTE":0,
 * "DESCOMI":0,"DESCORREL":0,"DIGITOSC":1,"CODINST":82,"COUNTLOTE":0,"COUNTSERI":0,"COUNTPARTES":0}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Item\_ProductsLoc()

Item\_ProductsLoc(int isEnser, string codUbic, string order, int page, int pageSize)

Devuelve una lista de productos en una ubicación y/o depósito específico.

Parámetros	Tipo	Descripción
<i>isEnser</i>	int	Si el producto es enser
<i>codUbic</i>	string	Ubicación del producto.
<i>order</i>	string	Condición de filtrado para los resultados (where)
<i>page</i>	int	Número de página
<i>pageSize</i>	int	Número de registros por página.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: fila, codprod, descrip, codinst, activo, descrip2, descrip3, refere, marca, unidad, undempa, cantempa, precio1, precio2, precio3, preciou, costact, costpro, costant, existen, exunidad, existencon, exunidadcon, compro, pedido, minimo, maximo, tara, descomp, descomi, desseri, deslote, desvence, edoabp, espublish, esimport, esexento, esenser, esoferta, espesa, esempaue, exdecimal, diasentr, fechauv, fechauc, preciou2, preciou3, peso, volumen, undvol, esreten, compro_e, pedido_e, codubic, existen_e, puestoi, cantcom, unidcom.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_ProductsLoc(0, "0001", "", 1, 1);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
*
* [{"FILA":1,"CODPROD":"20C05001","DESCRIP":"CHASIS CHEVROLET MOD NPR BUS","CODINST":29,"ACTIVO":1,"DESCRIP2":"","DESCRIP3":"","
* "REFERE":"","MARCA":"","UNIDAD":"UND","UNDEMPAQ":"","CANTEMPAQ":0.000,"PRECIO1":252196.43000,"PRECIO2":252196.43000,
* "PRECIO3":252196.43000,"PRECIU":0.00000,"COSTACT":299241.070,"COSTPRO":259307.196,"COSTANT":299241.070,"EXISTEN":20.000,
* "EXUNIDAD":0.000,"EXISTENCON":0.000,"EXUNIDADCON":0.000,"COMPRO":0.000,"PEDIDO":-18.000,"MINIMO":0.000,"MAXIMO":0.000,
* "TARA":0.000,"DESCOMP":0,"DESCOMI":0,"DESSERI":1,"DESLOTE":0,"DESVENCE":0,"EDOABP":null,"ESPUBLISH":0,"ESIMPORT":0,
* "ESEXENTO":0,"ESENSER":0,"ESOFERTA":0,"ESPESA":0,"ESEMPAQUE":0,"EXDECIMAL":0,"DIASENTR":0,"FECHAUV":"","
* "FECHAUC":"2013-05-31","PRECIU2":0.00000,"PRECIU3":0.00000,"PESO":0.000,"VOLUMEN":0.000,"UNDVOL":"","ESRETEN":0,
* "COMPRO_E":0.000,"PEDIDO_E":-18.000,"CODUBIC":"0001","EXISTEN_E":20.000,"PUESTOI":"","CANTCOM":0.000,"UNIDCOM":0.000}]
*/
/* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
```

## Item\_PartsProduct()

Item\_PartsProduct([string](#) codInv, [string](#) codUbic)

Devuelve una lista de las partes que componen a un producto, cuando éste es considerado un producto compuesto.

Parámetros	Tipo	Descripción
<i>codInv</i>	<a href="#">string</a>	Código de inventario.
<i>codUbic</i>	<a href="#">string</a>	Código del depósito.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codprod, codalte, cantidad, esunid, esserv, nrounico, codinst, descrip, descrip2, descrip3, esempaue, cantempa, desvence, costact, costpro, costant, esporcost, precio1,precio2, precio3, preciou1, preciou2, preciou3, desseri, refere, deslote, espesa, tara, usaserv, unidad, undempa, exdecimal, existant.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_PartsProduct("23C2", "0001");
        Console.WriteLine(_result);
        Console.Read();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODPROD":"23C2","CODALTE":"30220007","CANTIDAD":7.000,"ESUNID":0,"ESSERV":0,"NROUNICO":2,"CODINST":82,
 * "DESCRIP":"LAMINA DE YESO 1/2","DESCRIP2":"","DESCRIP3":"","EEMPAQUE":0,"CANTEMPAQ":0.000,"DESVENTE":0,
 * "COSTACT":78.560,"COSTPRO":78.560,"COSTANT":78.600,"ESPORCOST":0,"PRECIO1":78.60000,"PRECIO2":0.00000,
 * "PRECIO3":0.00000,"PRECIU1":0.00000,"PRECIU2":0.00000,"PRECIU3":0.00000,"DESSERI":0,"REFERE":"","
 * "DESLOTE":0, * "ESPESA":0,"TARA":0.000,"USASERV":0,"UNIDAD":"UND","UNDEMPAQ":"","EXDECIMAL":0,
 * "EXISTANT":0.000}]}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_Taxes()

Item\_Taxes(string codice, int type)

Devuelve una lista de los impuestos asociados al ítem, ya sea un producto o servicio.

Parámetros	Tipo	Descripción
<i>codice</i>	string	Código del producto o servicio.
<i>type</i>	int	Identificador del ítem. (producto = 0; servicio = 1).
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: [codprod   codserv], codtaxs, descrip, esporct, monto, eslibroi, esfijo, estaxventa, estaxcompra, esreten, codoper, sustraendo).

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_Taxes("32980046", 0);
        Console.WriteLine(_result);
        Console.Read();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODPROD":"20MB5001","CODTAXS":"IVA","DESCRIP":"I.V.A.", "ESPORCT":1,"MONTO":18.000,"ESLIBROI":1,
 * "ESFIO":1,"ESTAXVENTA":1,"ESTAXCOMPRA":1,"ESRETEN":0,"CODOPER":null,"SUSTRANDO":0.000}]}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_Service()

Item\_Service(string codServ)

Devuelve la información sobre un servicio.

Parámetros	Tipo	Descripción
<i>codServ</i>	string	Código del servicio.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codserv, descrip, descrip2, descrip3, descripall, esfreep, precio1, precio2, precio3, esimport, activo, unidad, clase, costo, esexento, codinst, descto, esporcost, usaserv, comision, esporcomi, fechauc, isadic.

Código de ejemplo:

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Item_Service("DE018");
        Console.WriteLine(_result);
    }
}

/*
 * Este código genera el siguiente resultado:
 *
 * {"CODSERV":"DE018","DESCRIP":"PULITURA CRISTALIZADO 3M RINES 4 PIEZAS","DESCRIP2":"","
 * "DESCRIP3":"","DESCRIPALL":"PULITURA CRISTALIZADO 3M RINES 4 PIEZAS  ", "ESFREEP":0,
 * "PRECIO1":440.00000,"PRECIO2":440.00000,"PRECIO3":440.00000,"ESIMPORT":0,"ACTIVO":1,
 * "UNIDAD":"","CLASE":"DE","COSTO":0.00000,"ESEXENTO":0,"CODINST":27,"DESCTO":0.00000,
 * "ESPORCOST":1,"USASERV":1,"COMISION":0.00000,"ESPORCOMI":0,"FECHAUC":null,"FECHAUC":null,
 * "ISADIC":0}
 *
 * Mensaje en caso de error: {"error":"Desconocido"}
 */
```

## Item\_ProdExistence()

Item\_ProdExistence(string codProd string codUbic)

Devuelve la existencia de un producto por depósito.

Parámetros	Tipo	Descripción
<i>codProd</i>	string	Código del producto.
<i>codUbic</i>	string	Código de ubicación.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codprod, codubic, puestoi, existen, exunidad, cantped, unidped, cantcom, unidcom.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_ProdExistence("20C05001", "0001");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"CODPROD": "20C05001", "CODUBIC": "0001", "PUESTOI": "", "EXISTEN": 20.000, "EXUNIDAD": 0.000, "CANTPED": 0.000, "UNIDPED": 0.000,
    * "CANTCOM": 0.000, "UNIDCOM": 0.000}
    */
    /* Mensaje en caso de error: {"error": "Desconocido"} */
}
```

## Item\_Lote()

Item\_Lote([string](#) codProd)

Devuelve lotes de un producto

Parámetros	Tipo	Descripción
<i>codProd</i>	<a href="#">string</a>	Código del producto.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codprod, nrounico, codubic, nrolote, costo, precio, preciou, cantidad, cantidadu, fecha, fechav, precio2, preciou2, precio3, preciou3, precio1, preciou1, puestoi.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_Lote("20C05001");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"CODPROD": "NROUNICO", "CODUBIC": "NROLOTE", "COSTO": "PRECIO", "PRECIU": "CANTIDAD", "CANTIDADU": "FECHAE",
    * "FECHAV": "PRECIO2", "PRECIO2": "PRECIO3", "PRECIO3": "PRECIO1", "PRECIO1": "PRECIU1", "PUESTOI": ""}
    */
    /* Mensaje en caso de error: {"error": "Desconocido"} */
}
```

## Item\_Serials()

Item\_Serials(string codProd)

Devuelve los serials que posee un producto

Parámetros	Tipo	Descripción
<i>codProd</i>	string	Código del producto.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codprod, codubic, nroserial.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_Serials("20C05001");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODPROD":"20C05001","CODUBIC":"0001","NROSERIAL":"8ZBFNP1V80G400282"}, {"CODPROD":"20C05001","CODUBIC":"0001",
 * "NROSERIAL":"8ZBFNP1Y0CG407158"}, {"CODPROD":"20C05001","CODUBIC":"0001","NROSERIAL":"8ZBFNP1Y1CG407072"},
 * {"CODPROD":"20C05001","CODUBIC":"0001","NROSERIAL":"8ZBFNP1Y1DG400284"}, {"CODPROD":"20C05001","CODUBIC":"0001",
 * "NROSERIAL":"8ZBFNP1Y1DG401225"}, {"CODPROD":"20C05001","CODUBIC":"0001","NROSERIAL":"8ZBFNP1Y3DG400285"}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Item\_ProductRef()

Item\_ProductRef(string refere)

Busca productos por su código de referencia.

Parámetros	Tipo	Descripción
<i>refere</i>	string	Código de referencia.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codprod, codinst, descorrel, digitosc, refere, desseri, descomp, esexento, exdecimal, unidad, undempa, deslote, espesa, tara, desvence, descto, cantempa, peso, volumen, undvol, costpro, descrip, descrip2, descrip3, preciou, preciou2, preciou3, precio1, precio2, precio3, isadic, costo.



## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Item_ProductRef("CATALOG");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODPROD":"P03","CODINST":98,"DESCORREL":0,"DIGITOSC":1,"REFERE":"CATALOGO 23","DESSERI":0,"DESCOMP":0,"ESEXENTO":0,
    * "EXDECIMAL":0,"UNIDAD":"","UNDEMPAQ":"","DESLOTE":0,"ESPESA":0,"TARA":0.000,"DESVENCE":0,"DESCTO":0.000,"CANTEMPAQ":0.000,
    * "PESO":0.000,"VOLUMEN":0.000,"UNDVOL":"","COSTPRO":64.470,"DESCRIP":"p03","DESCRIP2":"","DESCRIP3":"","PRECIU":0.00000,
    * "PRECIU2":0.00000,"PRECIU3":0.00000,"PRECI01":67.80000,"PRECI02":76.27000,"PRECI03":84.75000,"ISADIC":0,"COST0":64.470},
    * {"CODPROD":"10018407","CODINST":82,"DESCORREL":0,"DIGITOSC":1,"REFERE":"CATALOGO DX3L64","DESSERI":0,"DESCOMP":0,
    * "ESEXENTO":0,"EXDECIMAL":0,"UNIDAD":"","UNDEMPAQ":"","DESLOTE":0,"ESPESA":0,"TARA":0.000,"DESVENCE":0,"DESCTO":0.000,
    * "CANTEMPAQ":0.000,"PESO":0.000,"VOLUMEN":0.000,"UNDVOL":"","COSTPRO":0.000,"DESCRIP":"CAMION RAM 4000 4X2",
    * "DESCRIP2":"MOD.: RAM 4000 4X2.MOTOR 5.7 HEMY V8","DESCRIP3":"POTE.:330HP@4800RPM","PRECIU":0.00000,"PRECIU2":0.00000,
    * "PRECIU3":0.00000,"PRECI01":119000.00000,"PRECI02":119000.00000,"PRECI03":119000.00000,"ISADIC":0,"COST0":0.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Región TotalizerProcess

Esta sección contiene los métodos relacionados a la totalización de un tipo de documento. Los siguientes métodos integran esta sección:

- ✓ Total\_ValidateAgreements.
- ✓ Total\_SaveDocument.
- ✓ Total\_PaymentInstruments.
- ✓ Total\_SaleStatistics.
- ✓ Total\_WithHoldingTax.
- ✓ Total\_WaitDocumentTypeFac.

### Total\_ValidateAgreements()

Total\_ValidateAgreements([string](#) codClie, [decimal](#) montoBase, [decimal](#) cantidad, [string](#) fechaE)

Valida los convenios de un cliente.

Parámetros	Tipo	Descripción
<i>codClie</i>	<a href="#">string</a>	Código del cliente.
<i>montoBase</i>	<a href="#">decimal</a>	Monto base del ítem.
<i>Cantidad</i>	<a href="#">decimal</a>	Cantidad del ítem.
<i>fechaE</i>	<a href="#">string</a>	Fecha de emisión.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codclie, descrip, codsuc, id3, clase, represent, direc1, direc2, activo, pais, estado, ciudad, telef, movil, email, fax, nit, codzona, codvend, codconv, tipocli, tipopvp, intmora, escredito, limitecred, diascred, estoleran, diastole, fechaE, observa, saldo, saldoptos, montomax, mtomaxcred, pagosa, prompago, reteniva, descto, fechauv, montouv, numerouv, fechaup, montoup, numeroup, esmoneda, edokc, codalte, tipoid3, descorder, municipio, zipcode, descripext, tipoid, desctocv.

```

using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();
        _result = serviceSEA.Total_ValidateAgreements("08584744", 1523.6M, 3.8M, "2012-01-01 08:00:00.000");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"CODCLIE":"08584744","DESCRIP":"CARLOS AZUAJE","CODSUCU":"00000","ID3":"08584744","CLASE":"","REPRESENT":"","
 * "DIREC1":"","MCPO. JOSE RAFAEL REVENGA, URB. BOSQUE","DIREC2":"","LINDO, AV. PPAL. LAS ROSAS 149,
 * EL CONSEJO","ACTIVO":0,"PAIS":1,"ESTADO":20,"CIUDAD":0,"TELEF":"0416 - 4325509","MOVIL":"","( ) - ",
 * "EMAIL":"","FAX":"","NIT":"-----","CODZONA":"","CODVEND":"","CODCONV":"1","TIPOCLI":0,"TIPOPPV":0,
 * "INTMORA":0,"ESCRECREDITO":1,"LIMITECRED":0.000,"DIASCREC":0,"ESTOLERAN":0,"DIASSTOLE":0,
 * "FECHAE":"2008-05-29T08:10:58.337","OBSERVA":"","SALDO":0.000,"SALDOPTOS":0.000,"MONTOMAX":0.000,
 * "MTOMAXCRED":0.000,"PAGOSA":0.000,"PROMPAGO":-4.000,"RETENIVA":0.000,"DESCTO":0.000,"FECHAUV":null,
 * "MONTOUV":0.000,"NUMEROUV":null,"FECHAUP":"2008-07-21T00:00:00","MONTOUTP":327000.000,"NUMEROUTP":"000058",
 * "ESMONEDA":0,"EDOKC":null,"CODALTE":"","TipoID3":0,"DESCORDER":"","Municipio":0,"ZIPCODE":"","
 * "DESCRIPEXT":"CARLOS AZUAJE","TIPOID":0,"DESCTOCV":"001523.600"}
 *
 * Mensaje en caso de error: "{"error":"Desconocido"}"
 */

```

Luego implementamos el método `Total_SaveDocument()`, y pasamos como parámetro, la cadena string que devuelve el método `Documento()`, de la clase `MiClase`. Recuerde, esta clase es solo para mostrar la forma de implementar el método para grabar.

```
using System;
using testdynamic.Service;

class Program
{
    static void Main(string[] args)
    {
        MiClase Prueba = new MiClase();
        string _result = string.Empty;
        AdminServiceClient serviceSEA = new AdminServiceClient();

        _result = serviceSEA.Total_SaveDocument(Prueba.Documento());
        Console.Write(_result);
        Console.Read();
    }
}

/* Este código genera el siguiente resultado:
 * {"status":"OK" }"
 *
 * Mensaje en caso de error: {"error":"Desconocido"}"
 */
```

## Total\_PaymentInstruments()

### Total\_PaymentInstruments()

Devuelve información de los instrumentos de pago..

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: codtarj, descrip, clase, activo, impuesto, diasdif, basereten, reteniva.

### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Total_PaymentInstruments();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CODTARJ":"000","DESCRIP":"COMPROBANTES DE RETENCION DE IMPTO.,"CLASE":"","ACTIVO":1,"IMPUESTO":0.000,"DIASDIF":0,
 * "BASERETEN":0.000,"RETENIVA":0.000},{"CODTARJ":"0001","DESCRIP":"COMPROBANTES DE RETENCION IVA","CLASE":"","ACTIVO":1,
 * "IMPUESTO":0.000,"DIASDIF":0,"BASERETEN":0.000,"RETENIVA":0.000},{"CODTARJ":"0002","DESCRIP":"COMPROBANTE DE IMPTO MUNICIPAL 1%",
 * "CLASE":"","ACTIVO":1,"IMPUESTO":0.000,"DIASDIF":0,"BASERETEN":0.000,"RETENIVA":0.000},{"CODTARJ":"0003","DESCRIP":"NOTAS DE CREDITO",
 * "CLASE":"","ACTIVO":1,"IMPUESTO":0.000,"DIASDIF":0,"BASERETEN":0.000,"RETENIVA":0.000},{"CODTARJ":"0004","DESCRIP":"NOTAS DE DEBITO",
 * "CLASE":"","ACTIVO":1,"IMPUESTO":0.000,"DIASDIF":0,"BASERETEN":0.000,"RETENIVA":0.000}]
 *
 * Mensaje en caso de error: {"error":"Desconocido"}" */
```

## Total\_WithHoldingTax

Total\_WithHoldingTax()

Devuelve una lista de los impuestos de retención.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: codtaxs, descrip, mtotax, activo, esporct, escosto, eslibroi, espartic, tipoiva, estaxventa, estaxcompra, esfijo, esreten, codoper, montomax, sustraendo.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Total_WithHoldingTax();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CODTAXS":"002","DESCRIP":"RET. ISLR HON.PROF.P.N. 3%","MTOTAX":0.000,"ACTIVO":1,"ESPORCT":0,"ESCOSTO":0,"ESLIBROI":0,
 * "ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,"CODOPER":"ISLR","MONTOMAX":7500.000,
 * "SUSTRANDO":225.000},{ "CODTAXS":"004","DESCRIP":"RET. ISLR HON.PROF.P.J. 5%","MTOTAX":0.000,"ACTIVO":1,"ESPORCT":0,
 * "ESCOSTO":0,"ESLIBROI":0,"ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,"CODOPER":"ISLR",
 * "MONTOMAX":25.000,"SUSTRANDO":0.000},{ "CODTAXS":"018","DESCRIP":"RET. ISLR COMIS. P.N. 3%","MTOTAX":0.000,"ACTIVO":1,
 * "ESPORCT":0,"ESCOSTO":0,"ESLIBROI":0,"ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,
 * "CODOPER":"ISLR","MONTOMAX":7500.000,"SUSTRANDO":225.000},{ "CODTAXS":"020","DESCRIP":"RET. ISLR COMIS. P.J. 5%","MTOTAX":0.000,
 * "ACTIVO":1,"ESPORCT":0,"ESCOSTO":0,"ESLIBROI":0,"ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,
 * "CODOPER":"ISLR","MONTOMAX":25.000,"SUSTRANDO":0.000},{ "CODTAXS":"053","DESCRIP":"RET. ISLR SERVICIOS P.N. 1%","MTOTAX":0.000,
 * "ACTIVO":1,"ESPORCT":0,"ESCOSTO":0,"ESLIBROI":0,"ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,
 * "CODOPER":"ISLR","MONTOMAX":7500.000,"SUSTRANDO":75.000},{ "CODTAXS":"055","DESCRIP":"RET. ISLR SERVICIOS P.J. 2%","MTOTAX":0.000,
 * "ACTIVO":1,"ESPORCT":0,"ESCOSTO":0,"ESLIBROI":0,"ESPARTIC":0,"TIPOIVA":0,"ESTAXVENTA":0,"ESTAXCOMPRA":1,"ESFIJO":0,"ESRETE":1,
 * "CODOPER":"ISLR","MONTOMAX":0.000,"SUSTRANDO":0.000}]
 *
 */
/* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
```

## Total\_WaitDocumentTypeFac()

Total\_WaitDocumentTypeFac(string tipoFact)

Devuelve una lista de los documentos en espera de acuerdo al tipo de documento.

Parámetros	Tipo	Descripción
<i>tipoFact</i>	<b>string</b>	Tipo de documento.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: tipofac, numerod, nroctrol, status, codsuc, codesta, codusua, signo, otip, onumero, numeroc, numerot, numeror, fechad1, numerod1, numerok, numero, numero, esexpickup, moneda, factor, montomex, codclie, codvend, codubic, descrip, direc1,

Código de ejemplo:

54      Parte III: Métodos del SDK

# Región ReportProcess

Esta sección contiene los métodos relacionados a la generación de reportes, para la toma de decisiones. Los siguientes métodos integran esta sección:

- ✓ Report\_SalesDetails.
- ✓ Report\_AnalyticalSuppliers.
- ✓ Report\_AnalyticalSupplierH.
- ✓ Report\_AnalyticalSupplierD.
- ✓ Report\_AnalyticalCustomer.
- ✓ Report\_AnalyticalCustomerH.
- ✓ Report\_AnalyticalCustomerD.
- ✓ Report\_AdvSupplier.
- ✓ Report\_AdvSupplierD.
- ✓ Report\_AdvCustomer.
- ✓ Report\_AdvCustomerD.
- ✓ Report\_EffectsReceivable.
- ✓ Report\_EffectsReceivableD.
- ✓ Report\_EffectsReceivableDO.
- ✓ Report\_GeneralSales.
- ✓ Report\_EffectsPayable.
- ✓ Report\_EffectsPayableD.
- ✓ Report\_TotEffectsPayable.
- ✓ Report\_TotEffectsReceivable.
- ✓ Report\_DeferredDebits.
- ✓ Report\_DeferredCredits.
- ✓ Report\_TotPaymentCxC.
- ✓ Report\_TotPaymentsCxC.
- ✓ Report\_Payments.
- ✓ Report\_PaymentsLastMonths.
- ✓ Report\_Sales.
- ✓ Report\_SalesLastMonths.
- ✓ Report\_SalesInvoiceCreditD.
- ✓ Report\_SalesInvoiceCashD.
- ✓ Report\_TotAcctPayable.
- ✓ Report\_TotAcctsPayable.
- ✓ Report\_ItemsInvoice.
- ✓ Report\_TotReleases.
- ✓ Report\_ServiceFees.
- ✓ Report\_ServiceFeesD.
- ✓ Report\_BillsPayableSupp.
- ✓ Report\_BillsReceivableCust.
- ✓ Report\_TotDistAcctsPayable.
- ✓ Report\_TotDistAcctsPayableD.
- ✓ Report\_TotDistAcctsReceivable.
- ✓ Report\_TotDistAcctsReceivableD.
- ✓ Report\_BalanceBanks.
- ✓ Report\_ExplorationBanks.
- ✓ Report\_CashBanks.

- ✓ Report\_BalanceDBanks.
- ✓ Report\_DebitCreditNotes.
- ✓ Report\_DebitNotesD.
- ✓ Report\_CreditNotesD.
- ✓ Report\_AcctsReceivable.
- ✓ Report\_TotAcctReceivable.
- ✓ Report\_TotAcctReceivableD.
- ✓ Report\_PurchasesIVA.
- ✓ Report\_SalesIVA.
- ✓ Report\_WithholdingIVAReceivableD.
- ✓ Report\_WithholdingIVASalesD.
- ✓ Report\_WithholdingIVAPurchasesD.
- ✓ Report\_WithholdingIVA.
- ✓ Report\_InventoryFixtures.
- ✓ Report\_InventoryFixturesD.
- ✓ Report\_DischargeInventoryD.
- ✓ Report\_SalesProd.
- ✓ Report\_SalesServ.
- ✓ Report\_SalesProdD.
- ✓ Report\_PurchasesProdD.
- ✓ Report\_TotalSales.
- ✓ Report\_SalesDiscounts.
- ✓ Report\_CostsProdServD.
- ✓ Report\_FeesCosts.
- ✓ Report\_PurchaseLastMonths.
- ✓ Report\_SalesNumber.
- ✓ Report\_ComsnSellers.
- ✓ Report\_ComsnSellersD.
- ✓ Report\_TotComsnSeller.

## Report\_SalesDetails()

Report\_SalesDetails(*string* startDate, *string* endDate, *string* where, *int* pageSize, *int* page)

Muestra el detalle de ventas en periodo determinado. El método incluye control de paginación en los resultados.

Parámetros	Tipo	Descripción
<i>startDate</i>	<i>string</i>	Fecha de inicio.
<i>endDate</i>	<i>string</i>	Fecha final.
<i>where</i>	<i>string</i>	Sentencia(s) where aplicadas a la consulta.
<i>pageSize</i>	<i>Int</i>	Tamaño de la página de resultados.
<i>page</i>	<i>Int</i>	Número de la página de resultados que se desea visualizar.
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: fila, fecha, factura, montoneto, impuestos, totaloperacion, contado, credito, costos, utilidad, porcutil.



El siguiente ejemplo muestra el resultado del método **Report\_SalesDetails**.

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesDetails("2013-01-01", "2013-12-31", "", 4, 2);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"FILA":5,"FECHA":"2013-05-30","FACTURA":"000293","MONTONETO":3600.000,"IMPUESTOS":432.000,"TOTALOPERACION":4032.000,
 * "CONTADO":0.000,"CREDITO":4032.000,"COSTOS":568.080,"UTILIDAD":3463.920,"PORCUTIL":96.220000},{FILA":6,"FECHA":"2013-05-30",
 * "FACTURA":"000294","MONTONETO":760.000,"IMPUESTOS":91.200,"TOTALOPERACION":851.200,"CONTADO":0.000,"CREDITO":851.200,
 * "COSTOS":190.000,"UTILIDAD":661.200,"PORCUTIL":87.000000},{FILA":7,"FECHA":"2013-05-30","FACTURA":"000295",
 * "MONTONETO":661071.000,"IMPUESTOS":79328.520,"TOTALOPERACION":740399.520,"CONTADO":0.000,"CREDITO":740399.520,
 * "COSTOS":558865.833,"UTILIDAD":181533.687,"PORCUTIL":27.460000},{FILA":8,"FECHA":"2013-05-30","FACTURA":"000296",
 * "MONTONETO":661071.000,"IMPUESTOS":79328.520,"TOTALOPERACION":740399.520,"CONTADO":0.000,"CREDITO":740399.520,
 * "COSTOS":558865.833,"UTILIDAD":181533.687,"PORCUTIL":27.460000}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_AnalyticalSuppliers()

Report\_AnalyticalSuppliers(**string** date)

Reporte analítico de proveedores vencidos a la fecha.

Parámetros	Tipo	Descripción
<i>date</i>	<b>string</b>	Fecha final.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: descrip, telef, clase, codprov, tipocxp, numerod, detalle, fechav, dias, montopv, monto130, monto3160, monto6190, monto91, montov, total.

El siguiente ejemplo muestra el resultado:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalSuppliers("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"DESCRIP":"INVERSIONES CONSULTORES, C.A","TELEF":"0412 3461455","CLASE":"PROVEEDOR","CODPROV":"J294098649",
 * "TIPOCXP":"FAC","NUMEROD":"001537","DETALLE":"001537 00-001287","FECHAV":"2012-10-23","DIAS":70,"MONTOPV":0.000,
 * "MONTOT130":0.000,"MONTOT3160":0.000,"MONTOT6190":1078.000,"MONTOT91":0.000,"MONTOTOV":1078.000,"TOTAL":1078.000},
 * {"DESCRIP":"INVERSIONES CONSULTORES, C.A","TELEF":"0412 3478865","CLASE":"PROVEEDOR","CODPROV":"J299023649",
 * "TIPOCXP":"FAC","NUMEROD":"001569","DETALLE":"001569 00-001319","FECHAV":"2012-12-03","DIAS":29,"MONTOPV":0.000,
 * "MONTOT130":1078.000,"MONTOT3160":0.000,"MONTOT6190":0.000,"MONTOT91":0.000,"MONTOTOV":1078.000,"TOTAL":1078.000},
 * {"DESCRIP":"INDUSTRIA VZLANA","TELEF":"","CLASE":"PROVEEDOR","CODPROV":"J309983326",
 * "TIPOCXP":"FAC","NUMEROD":"0413","DETALLE":"0413 00-0413","FECHAV":"2012-01-16","DIAS":351,"MONTOPV":0.000,
 * "MONTOT130":0.000,"MONTOT3160":0.000,"MONTOT6190":0.000,"MONTOT91":1500.000,"MONTOTOV":1500.000,"TOTAL":1500.000},
 * {"DESCRIP":"INDUSTRIA VZLANA","TELEF":"","CLASE":"PROVEEDOR","CODPROV":"J3089993326",
 * "TIPOCXP":"FAC","NUMEROD":"0421","DETALLE":"0421 00-0421","FECHAV":"2012-02-03","DIAS":333,"MONTOPV":0.000,
 * "MONTOT130":0.000,"MONTOT3160":0.000,"MONTOT6190":0.000,"MONTOT91":1800.000,"MONTOTOV":1800.000,"TOTAL":1800.000}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_AnalyticalSupplierH()

Report\_AnalyticalSupplierH([string](#) date)

Encabezado del reporte analítico de proveedores vencidos a la fecha.

Parámetros	Tipo	Descripción
<i>date</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codprov, descrip, observa, telef, clase, total.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalSupplierH("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODPROV":"J299993649","DESCRIP":"INVERSIONES CONSULTORES, C.A","OBSERVA":"","TELEF":"0412 3999865",
    * "CLASE":"PROVEEDOR","TOTAL":2156.000},{ "CODPROV":"J987183326","DESCRIP":"INDUSTRIA VZLANA",
    * "OBSERVA":"","TELEF":"","CLASE":"PROVEEDOR","TOTAL":6900.000},{ "CODPROV":"V151022720",
    * "DESCRIP":"PEDRO PEREZ P","OBSERVA":"","TELEF":"0241 9992650 / 8249992","CLASE":"PROVEEDOR",
    * "TOTAL":850.000}]
    */
    /* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
}
```

## Report\_AnalyticalSupplierD()

Report\_AnalyticalSupplierD([string](#) codProv, [string](#) date)

Detalle del reporte analítico de proveedores vencidos a una fecha.

Parámetros	Tipo	Descripción
<i>codProv</i>	<a href="#">string</a>	Código del proveedor.
<i>date</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codprov, tipocxp, numerod, detalle, fechav, dias, montopv, monto130, monto3160, monto6190, monto91, montov, total.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalSupplierD("J999000111", "2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CODPROV": "J999000111", "TIPOCXPC": "FAC", "NUMEROD": "0413", "DETALLE": "0413 00-0413", "FECHAV": "2012-01-16", "DIAS": 351,
 * "MONTOPV": 0.000, "MONTOT130": 0.000, "MONTOT3160": 0.000, "MONTOT6190": 0.000, "MONTOT91": 1500.000, "MONTOTV": 1500.000,
 * "TOTAL": 1500.000}, {"CODPROV": "J308183326", "TIPOCXPC": "FAC", "NUMEROD": "0421", "DETALLE": "0421 00-0421", "FECHAV": "2012-02-03",
 * "DIAS": 333, "MONTOPV": 0.000, "MONTOT130": 0.000, "MONTOT3160": 0.000, "MONTOT6190": 0.000, "MONTOT91": 1800.000, "MONTOTV": 1800.000,
 * "TOTAL": 1800.000}, {"CODPROV": "J308183326", "TIPOCXPC": "FAC", "NUMEROD": "0429", "DETALLE": "0429 00-0429", "FECHAV": "2012-03-12",
 * "DIAS": 295, "MONTOPV": 0.000, "MONTOT130": 0.000, "MONTOT3160": 0.000, "MONTOT6190": 0.000, "MONTOT91": 1800.000, "MONTOTV": 1800.000,
 * "TOTAL": 1800.000}]
 */
/* Mensaje en caso de error: {"error": "Desconocido"} */
```

## Report\_AnalyticalCustomer()

Report\_AnalyticalCustomer(string date)

Reporte analítico de las cuentas vencidas de un cliente, a una fecha.

Parámetros	Tipo	Descripción
<i>date</i>	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: descrip, telef, clase, codclie, numerod, detalle, fecha, fechav, dias, total, tipocxc, montopv, monto130, monto3160, monto6190, monto91, montov.

## Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalCustomer("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"DESCRIP": "MOTORS, C. A.", "TELEF": "(0212) 1234789 / 9840", "CLASE": "CLIENTE", "CODCLIE": "J999000111", "NUMEROD": "714",
 * "DETALLE": "BUS 26 PUESTOS. SERIAL 45754575", "FECHAE": "2011-12-31", "FECHAV": "2011-05-31",
 * "DIAS": 581, "TOTAL": 156800.000, "TIPOCXPC": "FAC", "MONTOPV": 0.000, "MONTOT130": 0.000, "MONTOT3160": 0.000, "MONTOT6190": 0.000,
 * "MONTOT91": 156800.000, "MONTOTV": 156800.000}, {"DESCRIP": "DELFIN MOTORS, C. A.", "TELEF": "(0212) 6909808 / 9840",
 * "CLASE": "CLIENTE", "CODCLIE": "J002234768", "NUMEROD": "000790", "DETALLE": "000790", "FECHAE": "2011-11-22T14:20:22.36",
 * "FECHAV": "2011-11-22T14:20:22.39", "DIAS": 406, "TOTAL": 22288.000, "TIPOCXPC": "FAC", "MONTOPV": 0.000, "MONTOT130": 0.000,
 * "MONTOT3160": 0.000, "MONTOT6190": 0.000, "MONTOT91": 22288.000, "MONTOTV": 22288.000}]
 */
/* Mensaje en caso de error: {"error": "Desconocido"} */
```

## Report\_AnalyticalCustomerH()

Report\_AnalyticalCustomerH(string date)

Encabezado del reporte analítico de clientes vencidos a la fecha.

Parámetros	Tipo	Descripción
<i>date</i>	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codclie, descrip, observa, telef, clase, saldo.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalCustomerH("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CODCLIE":"J999000111","DESCRIP":"MOTORS, C. A.,"OBSERVA":"","TELEF":"(0212) 1112233 / 9840","CLASE":"CLIENTE",
 * "SALDO":179088.000},{ "CODCLIE":"J999000222","DESCRIP":"INVERSIONES, C.A.,"OBSERVA":"","TELEF":"","
 * "CLASE":"CONTRATIST","SALDO":267290.060},{ "CODCLIE":"J999000333",      * "DESCRIP":"COOPERATIVA CARROCEERÍAS PRIMERA, R. L.",
 * "OBSERVA":"","TELEF":"","CLASE":"CONTRATIST","SALDO":230927.610},{ "CODCLIE":"J999000444",
 * "DESCRIP":"ASOC. COOP. VIDRIOS SUAREZ R.L.,"OBSERVA":"","TELEF":"0412-12345674","CLASE":"CONTRATIST","SALDO":302299.490}]
 */
/* Mensaje en caso de error: "{"error":"Desconocido"}" */
```

## Report\_AnalyticalCustomerD()

Report\_AnalyticalCustomerD(string codClie, string date)

Detalle del reporte analítico de clientes vencidos a una fecha.

Parámetros	Tipo	Descripción
<i>codClie</i>	string	Código del cliente.
<i>date</i>	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: tipocxc, numerod, detalle, fechav, dias, montopv, monto130, monto3160, monto6190, monto91, montov, total.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AnalyticalCustomerD("J999000111", "2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"TIPOCXK":"FAC","NUMEROD":"714","DETALLE":"BUS 26 PUESTOS. SERIAL 54GF254D42G8H7","FECHAV":"2011-05-31",
 * "DIAS":581,"MONTOPV":0.000,"MONT0130":0.000,"MONT03160":0.000,"MONT06190":0.000,"MONT091":156800.000,"MONT0V":156800.000,
 * "TOTAL":156800.000},{ "TIPOCXK":"FAC","NUMEROD":"000790","DETALLE":"000790","FECHAV":"2011-11-22","DIAS":406,
 * "MONTOPV":0.000,"MONT0130":0.000,"MONT03160":0.000,"MONT06190":0.000,"MONT091":22288.000,"MONT0V":22288.000,
 * "TOTAL":22288.000}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"}" */
```

## Report\_AdvSupplier()

Report\_AdvSupplier( [string](#) endDate)

Muestra el total de los adelantos realizados a los proveedores a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: adelprov.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AdvSupplier("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"ADELPROV":2130721.600}
 */
/* Mensaje en caso de error: {"error":"Desconocido"}" */
```

## Report\_AdvSupplierD()

Report\_AdvSupplierD( [string](#) endDate)

Muestra el total detallado de los adelantos realizados a los proveedores a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: proveedor, fechae, numerod, saldo, monto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AdvSupplierD("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"PROVEEDOR":"COOL, C.A.", "FECHAE":"2011-11-29", "NUMEROD":"71400849", "SALDO":200000.000, "MONTO":200000.000},
    * {"PROVEEDOR":"CONSTRUVA, C.A.", "FECHAE":"2011-12-02", "NUMEROD":"4497", "SALDO":20000.000, "MONTO":20000.000},
    * {"PROVEEDOR":"COOL, C.A.", "FECHAE":"2011-12-27", "NUMEROD":"1215", "SALDO":500000.000, "MONTO":500000.000},
    * {"PROVEEDOR":"CONSTRUVA, C.A.", "FECHAE":"2012-01-19", "NUMEROD":"2425", "SALDO":10000.000, "MONTO":10000.000},
    * {"PROVEEDOR":"COOL, C.A.", "FECHAE":"2012-02-29", "NUMEROD":"2441", "SALDO":100000.000, "MONTO":100000.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_AdvCustomer()

Report\_AdvCustomer( [string](#) endDate)

Muestra el total de los adelantos realizados por los clientes a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: adelcli.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AdvCustomer("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"ADELCLI":5032878.400}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_AdvCustomerD()

Report\_AdvCustomerD( [string](#) endDate)

Muestra el total detallado de los adelantos realizados por los clientes a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codclie, cliente, document, fecha, numerod, saldo.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AdvCustomerD("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODCLIE":"98765432","CLIENTE":"JUANA MARIA","DOCUMENT":"ANTICIPO OP 806 NPR","FECHA":"2013-06-07",
    * "NUMEROD":"001060","SALDO":500000.000},{ "CODCLIE":"98765432","CLIENTE":"JUANA MARIA",
    * "DOCUMENT":"ANTICIPO OP 806 NPR","FECHA":"2013-06-07","NUMEROD":"001061","SALDO":98278.400},
    * {"CODCLIE":"J999000111","CLIENTE":"RUSTICOS C. A. ","DOCUMENT":"ANTICIPO 3 UND A/A OP 859/860/861 RUSTIC",
    * "FECHA":"2013-05-28","NUMEROD":"000007","SALDO":2322000.000},
    * {"CODCLIE":"J999000111","CLIENTE":"RUSTICOS C. A. ","DOCUMENT":"ANT DE 2 UND OP 862/863 RUSTICOS",
    * "FECHA":"2013-05-28","NUMEROD":"000008","SALDO":810000.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_EffectsReceivable()

Report\_EffectsReceivable( [string](#) endDate)

Muestra el total de efectos por cobrar a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: efectosxcobrar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_EffectsReceivable("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"EFECTOSXCOBRAR":1000,000}
    */
    /* Mensaje en caso de error: "{"error":"Desconocido"}" */
}
```

## Report\_EffectsReceivableD()

Report\_EffectsReceivableD( [string](#) endDate)

Muestra el total detallado de los efectos por cobrar a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codclie, descrip, codoper, document, fechav, monto, saldo.



Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_EffectsReceivableD("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODCLIE":"CLIENTE 01","DESCRIP":"ANTENA CUBICA","CODOPER":"VTA","DOCUMENT":"","FECHAV":"2013-06-07","MONTO":500.00
    * "SALDO":500.00},
    * [{"CODCLIE":"CLIENTE 02","DESCRIP":"ROUTER ALAMBRICO","CODOPER":"VTA","DOCUMENT":"","FECHAV":"2013-03-28","MONTO":500.00
    * "SALDO":500.00}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_EffectsReceivableDO()

Report\_EffectsReceivableDO( [string](#) endDate)

Muestra el total de los efectos por cobrar vencidos y por vencer a una fecha.

Parámetros	Tipo	Descripción
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: vencido, porvencer.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_EffectsReceivableDO("2013-07-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"VENCIDO":0.00,"PORVENCER":0.00}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_GeneralSales()

Report\_GeneralSales(string startDate, string endDate)

Muestra los montos totales de ventas realizadas en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: montoneto, montoa, iva, costo, total, descuento, fletes, adelantos, reteniva, contado, credito, totalnd, totaliva_nd, totalnc, totaliva_nc, devoluciones.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.ReportGeneralSales("2013-01-01", "2013-12-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"MONTONETO":40391059.65,"MONTOA":40391059.65,"IVA":5000631.31,"COSTO":14094078.75,"TOTAL":45391690.96,
 * "DESCUENTO":0.00,"FLETES":0.00,"ADELANTOS":5121591.04,"RETENIVA":0.00,"CONTADO":5121591.04,"CREDITO":40270099.92,
 * "TOTALND":0.00,"TOTALIVA_ND":0.00,"TOTALNC":0.00,"TOTALIVA_NC":0.00,"DEVOLUCIONES":19864074.46}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_EffectsPayable()

Report\_EffectsPayable(string endDate)

Muestra el total de efectos por pagar a una fecha.

Parámetros	Tipo	Descripción
endDate	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: efectosxpagar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_EffectsPayable("2013-01-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"EFECTOSXPAGAR":0.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_EffectsPayableD()

Report\_EffectsPayableD(string endDate)

Muestra el total detallado de los efectos por pagar a una fecha.

Parámetros	Tipo	Descripción
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: codprov, descrip, numerod, fechav, saldo.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_EffectsPayableD("2013-03-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODPROV":"025639","DESCRIP":"PROVEDOR 001","NUMEROD":"00236","FECHAV":"2013-02-05","SALDO":150.00}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotEffectsPayable()

Report\_TotEffectsPayable()

Muestra el total de los efectos por pagar a la fecha.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: tefectosxpagar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotEffectsPayable();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"TEFECTOSXPAGAR":0.000}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_TotEffectsReceivable()

Report\_TotEffectsReceivable()

Muestra el total de los efectos por cobrar a la fecha.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: tefectosxcobrar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotEffectsReceivable();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"TEFECTOSXCOBRAR":0.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_DeferredDebits()

Report\_DeferredDebits([string](#) startDate)

Devuelve el total de débitos diferidos, desde una fecha de inicio a la fecha actual.

Parámetros	Tipo	Descripción
<i>startDate</i>	<a href="#">string</a>	Fecha inicial.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: debitosdiferidos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_DeferredDebits("2013-05-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"DEBITOSDIFERIDOS":594663.090}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_DeferredCredits()

Report\_DeferredCredits(string startDate)

Devuelve el total de créditos diferidos, desde una fecha de inicio a la fecha actual.

Parámetros	Tipo	Descripción
<i>startDate</i>	string	Fecha inicial.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: creditosdiferidos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_DeferredCredits("2013-05-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"CREDITOSDIFERIDOS":236989.350}
 */

/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_TotPaymentCxC()

Report\_TotPaymentCxC(string startDate, string endDate)

Devuelve el total de pagos de cuentas por cobrar en un periodo dado.

Parámetros	Tipo	Descripción
<i>startDate</i>	string	Fecha inicial.
<i>endDate</i>	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: totalpagoscxc.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotPaymentCxC("2013-01-01", "2013-05-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"TOTALPAGOSCxC":10897721.820}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotPaymentsCxC()

Report\_TotPaymentsCxC(int month, int year)

Devuelve el total de pagos de cuentas por cobrar para un año y mes específico.

Parámetros	Tipo	Descripción
<i>month</i>	int	Mes.
<i>year</i>	int	Año.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: totalpagoscxc.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotPaymentsCxC(5, 2013);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"TOTALPAGOSCxC":114427.650}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_Payments()

Report\_Payments(string startDate, string endDate)

Devuelve el total de pagos mensuales realizados para un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<hr/>		
Valor devuelto	string	Cadena de texto en formato Json. Campos: monto, anio, mes.
<hr/>		

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_Payments("2013-01-01", "2013-03-01");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"MONTO":1601489.500,"ANIO":2013,"MES":1},{ "MONTO":1747836.000,"ANIO":2013,"MES":2}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_PaymentsLastMonths()

Report\_PaymentsLastMonths()

Devuelve el monto total mensual de los pagos realizados los últimos 6 meses, sin incluir el mes actual.

Parámetros	Tipo	Descripción
no requiere		
<hr/>		
Valor devuelto	string	Cadena de texto en formato Json. Campos: monto, anio, mes.
<hr/>		



### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_PaymentsLastMonths();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"MES":8,"ANIO":2013,"MONTO":1500.000},
    * {"MES":9,"ANIO":2013,"MONTO":1600.000},
    * {"MES":10,"ANIO":2013,"MONTO":1689.000},
    * {"MES":11,"ANIO":2013,"MONTO":1785.000},
    * {"MES":12,"ANIO":2013,"MONTO":1952.000},
    * {"MES":1,"ANIO":2014,"MONTO":963.000},]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

### Report\_Sales()

Report\_Sales(string startDate, string endDate)

Devuelve el total mensual de ventas en un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: monto, anio, mes.

### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_Sales("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"MONTO":1370956.480,"ANIO":2013,"MES":1},
    * {"MONTO":3838545.720,"ANIO":2013,"MES":2},
    * {"MONTO":3845988.000,"ANIO":2013,"MES":3}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_SalesLastMonths()

Report\_SalesLastMonths()

Devuelve el monto total mensual de las ventas realizadas en los últimos 6 meses, sin incluir el mes actual.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: monto, anio, mes.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesLastMonths();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"MES":8,"ANIO":2013,"MONTO":50.000},
 * {"MES":9,"ANIO":2013,"MONTO":100.000},
 * {"MES":10,"ANIO":2013,"MONTO":150.000},
 * {"MES":11,"ANIO":2013,"MONTO":360.000},
 * {"MES":12,"ANIO":2013,"MONTO":90.000},
 * {"MES":1,"ANIO":2014,"MONTO":80.000}]
 */
/* Mensaje en caso de error: "{"error":"Desconocido"}" */
```

## Report\_SalesInvoiceCreditD()

Report\_SalesInvoiceCreditD(string startDate, string endDate)

Devuelve el detalle de las facturas de ventas a crédito, en un periodo.

Parámetros	Tipo	Descripción
<i>startDate</i>	<b>string</b>	Fecha inicial.
<i>endDate</i>	<b>string</b>	Fecha final.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: factura, cliente, descripcion, fecha, credito.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesInvoiceCreditD("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    * [{"FACTURA":"000982","CLIENTE":"J999000555","DESCRIPCION":"TRANSPORTE CA","FECHA":"2013-01-11",
    * "CREDITO":560000.000},{FACTURA:"000983","CLIENTE":"J999000666",
    * "DESCRIPCION":"CONSEJO MACHODO","FECHA":"2013-01-24","CREDITO":1470.000},
    * {"FACTURA":"000984","CLIENTE":"J999000777","DESCRIPCION":"EQUIPAMIENTOS C. A.", "FECHA":"2013-01-30",
    * "CREDITO":683636.500},{FACTURA:"000985","CLIENTE":"V999000888","DESCRIPCION":"ARNALDO ALVE",
    * "FECHA":"2013-01-30","CREDITO":260.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_SalesInvoiceCashD()

Report\_SalesInvoiceCashD([string](#) startDate, [string](#) endDate)

Devuelve el detalle de las facturas de ventas de contado, en un periodo.

Parámetros	Tipo	Descripción
<i>startDate</i>	<a href="#">string</a>	Fecha inicial.
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: factura, cliente, descripcion, fecha, contado.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesInvoiceCashD("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    * [{"FACTURA":"000982","CLIENTE":"J999000555","DESCRIPCION":"TRANSPORTE CA","FECHA":"2013-01-11",
    * "CONTADO":560000.000},{FACTURA:"000983","CLIENTE":"J999000666",
    * "DESCRIPCION":"CONSEJO MACHODO","FECHA":"2013-01-24","CONTADO":1470.000},
    * {"FACTURA":"000984","CLIENTE":"J999000777","DESCRIPCION":"EQUIPAMIENTOS C. A.", "FECHA":"2013-01-30",
    * "CONTADO":683636.500},{FACTURA:"000985","CLIENTE":"V999000888","DESCRIPCION":"ARNALDO ALVE",
    * "FECHA":"2013-01-30","CONTADO":260.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotAcctPayable()

Report\_TotAcctPayable ()

Devuelve el monto total de cuentas por pagar a la fecha.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: tctasxpagar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotAcctPayable();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * {"TCTASXPAGAR":734536.770}
 */

/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_TotAcctsPayable()

Report\_TotAcctsPayable(*string* endDate)

Devuelve el monto total de cuentas por pagar a una fecha dada.

Parámetros	Tipo	Descripción
<i>endDate</i>	<i>string</i>	Fecha final.
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: ctasxpagar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotAcctsPayable("2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"CTASXPAGAR":31532.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_ItemsInvoice()

Report\_ItemsInvoice([string](#) invoiceNumber)

Devuelve los ítems de una factura.

Parámetros	Tipo	Descripción
<i>invoiceNumber</i>	<a href="#">string</a>	Numero de factura.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: coditem, descrip, cantidad, precio, totalitem.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ItemsInvoice("000060");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODITEM":"SFTTV","DESCRIP":"SERVICIO DE TAPAS VISOR","CANTIDAD":6.00000,"PRECIO":650.00000,
    * "TOTALITEM":3900.00000},{"CODITEM":"29C3","DESCRIP":"FAB. E INST. BUSS (4815MM) 32 PTOS","CANTIDAD":1.00000,
    * "PRECIO":77000.00000,"TOTALITEM":77000.00000},{"CODITEM":"29CC18","DESCRIP":"MAXI URBANO 26 PUESTOS",
    * "CANTIDAD":1.00000,"PRECIO":1699486.50000,"TOTALITEM":1699486.50000},{"CODITEM":"29000003",
    * "DESCRIP":"ESPECIFICACIONES MINI-BUSS URBANO 26PT","CANTIDAD":1.00000,"PRECIO":0.00000,"TOTALITEM":0.00000},
    * {"CODITEM":"42000580","DESCRIP":"PLACAS (E)","CANTIDAD":1.00000,"PRECIO":513.50000,"TOTALITEM":513.50000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotReleases()

Report\_TotReleases(*string* startDate, *string* endDate)

Devuelve el monto total de descargos en un periodo.

Parámetros	Tipo	Descripción
<i>startDate</i>	<i>string</i>	Fecha inicial.
<i>endDate</i>	<i>string</i>	Fecha final.
<hr/>		
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: descargos.
<hr/>		

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotReleases("2012-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"DESCARGOS":472253.064}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_ServiceFees()

Report\_ServiceFees(*string* startDate, *string* endDate)

Devuelve el monto total de comisiones por servicios en un periodo dado.

Parámetros	Tipo	Descripción
<i>startDate</i>	<i>string</i>	Fecha inicial.
<i>endDate</i>	<i>string</i>	Fecha final.
<hr/>		
<b>Valor devuelto</b>	<i>string</i>	Cadena de texto en formato Json. Campos: comionserv.
<hr/>		

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ServiceFees("2012-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"COMISIONSERV":15236.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_ServiceFeesD()

Report\_ServiceFeesD([string](#) startDate, [string](#) endDate)

Devuelve el monto total detallado de comisiones por servicios en un periodo dado.

Parámetros	Tipo	Descripción
<i>startDate</i>	<a href="#">string</a>	Fecha inicial.
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: numerod, coditem, descrip1, fecha, monto, precio, cantidad, subtotal, comision, tipofac.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ServiceFeesD("2012-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"NUMEROD":"000060","CODITEM":"29CC18","DESCRIP1":"FAB. E INST.BUSS (4815MM)32 PTOS","FECHAE":"2013-02-19",
    * "MONTO":7700.000,"PRECIO":16996.500,"CANTIDAD":1,"SUBTOTAL":16996.500,"COMISION":0.000,"TIPOFAC":"A"}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_BillsPayableSupp()

Report\_BillsPayableSupp(string codProv, string date)

Devuelve las facturas por pagar de un proveedor a una fecha dada.

Parámetros	Tipo	Descripción
<i>codProv</i>	string	Código del proveedor.
<i>date</i>	string	Fecha.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codprov, numerod, detalle, fecha, fechav, tipocxp, dias, grupo, total.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_BillsPayableSupp("J000999111", "2013-12-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODPROV": "J000999111", "NUMEROD": "117878", "DETALLE": "117878 008178", "FECHAE": "2013-07-25T14:53:18.473",
 * "FECHAV": "2013-08-12T14:53:18.503", "TIPOCXP": "FAC", "DIAS": 141, "GRUPO": "G91+", "TOTAL": 48951.830},
 * {"CODPROV": "J000999111", "NUMEROD": "117962", "DETALLE": "117962 008178", "FECHAE": "2013-07-31T09:40:28.467",
 * "FECHAV": "2013-08-18T09:40:28.5", "TIPOCXP": "FAC", "DIAS": 135, "GRUPO": "G91+", "TOTAL": 55880.600}]
 */
/* Mensaje en caso de error: {"error": "Desconocido"} */
```

## Report\_BillsReceivableCust()

Report\_BillsReceivableCust(string codClie, string date)

Devuelve las facturas por cobrar de un cliente a una fecha dada.

Parámetros	Tipo	Descripción
<i>codClie</i>	string	Código del cliente.
<i>date</i>	string	Fecha.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codclie, numerod, detalle, fecha, fechav, tipocxc, dias, grupo, total.



Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_BillsReceivableCust("10189992", "2013-12-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    * [{"CODCLIE":"10189992","NUMEROD":"001032","DETALLE":"001032","FECHAE":"2013-06-19T13:03:12.843",
    * "FECHAV":"2013-06-19T13:03:12.877","TIPOCX":"FAC","DIAS":195,"GRUPO":"G91+","TOTAL":1903938.380}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotDistAcctsPayable()

Report\_TotDistAcctsPayable(string date)

Devuelve los montos totales vencidos distribuidos en cuentas por pagar.

Parámetros	Tipo	Descripción
<i>date</i>	string	Fecha.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: montoporvencer, monto130, monto3160, monto6190, monto91, montovencido.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotDistAcctsPayable("2013-12-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    * {"MONTOPORVENCER":0.000,"MONT0130":0.000,"MONT03160":0.000,"MONT06190":0.000,"MONT091":734536.770,
    * "MONTOVENCIDO":734536.770}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotDistAcctsPayableD()

Report\_TotDistAcctsPayableD([string](#) date)

Devuelve los montos totales detallados vencidos distribuidos en cuentas por pagar, a una fecha.

Parámetros	Tipo	Descripción
<i>date</i>	<a href="#">string</a>	Fecha.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codprov, descrip, montoporvencer, monto130, monto3160, monto6190, monto91, montovencido.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotDistAcctsPayableD("2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"CODPROV":"J999000111","DESCRIP":"CAMARA VZLANA","MONTOPORVENCER":0.000,"MONTO130":0.000,
 * "MONTO3160":0.000,"MONTO6190":0.000,"MONTO91":6900.000,"MONTOVENCIDO":6900.000},{"CODPROV":"J999000222",
 * "DESCRIP":"INVERSIONES CONSULTORES","MONTOPORVENCER":0.000,"MONTO130":0.000,"MONTO3160":1078.000,
 * "MONTO6190":0.000,"MONTO91":1078.000,"MONTOVENCIDO":2156.000},{"CODPROV":"J999000333",
 * "DESCRIP":"REPRESENTACIONES MARIA C.A","MONTOPORVENCER":0.000,"MONTO130":19950.000,
 * "MONTO3160":0.000,"MONTO6190":0.000,"MONTO91":0.000,"MONTOVENCIDO":19950.000},{"CODPROV":"V999000444",
 * "DESCRIP":"VICTOR SILVA","MONTOPORVENCER":0.000,"MONTO130":0.000,"MONTO3160":0.000,"MONTO6190":0.000,
 * "MONTO91":850.000,"MONTOVENCIDO":850.000}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_TotDistAcctsReceivable()

Report\_TotDistAcctsReceivable([string](#) date)

Devuelve los montos totales vencidos distribuidos en cuentas por cobrar.

Parámetros	Tipo	Descripción
<i>date</i>	<a href="#">string</a>	Fecha.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: montoporvencer, monto130, monto3160, monto6190, monto91, montovencido.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotDistAcctsReceivable("2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"MONTOPORVENCER":0.000,"MONTO130":1140863.110,"MONTO3160":0.000,"MONTO6190":0.000,"MONTO91":182322.560,
    * "MONTOVENCIDO":1323185.670}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotDistAcctsReceivableD()

Report\_TotDistAcctsReceivableD(string date)

Devuelve los montos totales detallados vencidos distribuidos en cuentas por cobrar, a una fecha.

Parámetros	Tipo	Descripción
date	string	Fecha.
Valor devuelto	string	Cadena de texto en formato Json. Campos: codclie, descrip, montoporvencer, monto130, monto3160, monto6190 monto91, montovencido.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotDistAcctsReceivableD("2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODCLIE":"J999000111","DESCRIP":"ASOC. COOP. VIDRIOS","MONTOPORVENCER":0.000,"MONTO130":302299.490,
    * "MONTO3160":0.000,"MONTO6190":0.000,"MONTO91":0.000,"MONTOVENCIDO":302299.490},{"CODCLIE":"J999000222",
    * "DESCRIP":"ASOC. COOPERATIVA PRIMERA","MONTOPORVENCER":0.000,"MONTO130":230927.610,
    * "MONTO3160":0.000,"MONTO6190":0.000,"MONTO91":0.000,"MONTOVENCIDO":230927.610},{"CODCLIE":"J999000333",
    * "DESCRIP":"CARROCERIAS C.A.", "MONTOPORVENCER":0.000,"MONTO130":69071.990,"MONTO3160":0.000,
    * "MONTO6190":0.000,"MONTO91":0.000,"MONTOVENCIDO":69071.990}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_BalanceBanks()

Report\_BalanceBanks([string](#) date)

Devuelve los saldos en banco a una fecha.

Parámetros	Tipo	Descripción
<i>date</i>	<a href="#">string</a>	Fecha.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: saldos, debitos, creditos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_BalanceBanks("2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * {"SALDOS":3042225.510,"DEBITOS":34079864.340,"CREDITOS":37284581.450}
 */
/* Mensaje en caso de error: {"error":"Desconocido"}" */
```

## Report\_ExplorationBanks()

Report\_ExplorationBanks([string](#) codeBank, [string](#) date)

Realiza una exploración en bancos.

Parámetros	Tipo	Descripción
<i>codeBank</i>	<a href="#">string</a>	Código del banco.
<i>date</i>	<a href="#">string</a>	Fecha.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: descripcion, documento, feclib, cdcd, tipoope, tipopgo, descpggo, dctopgo, monto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ExplorationBanks("110110002", "2013-03-05");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"DESCRIPCION":"DISTRIBUIDORA ELI-C.A.", "DOCUMENTO":"0016460", "FECLIB":"2020-01-15", "CDCD":0,
    * "TIPOOPE":1, "TIPOPGO":"", "DESCPGO":"", "DCTOPGO":"", "MONTO":-427.500}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"}" */
}
```

## Report\_CashBanks()

Report\_CashBanks()

Devuelve el total de efectivo en bancos a la fecha.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: efectivobancos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_CashBanks();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"EFFECTIVOBANCOS":3042225.510}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"}" */
}
```

## Report\_BalanceDBanks()

Report\_BalanceDBanks(string date, string currency)

Devuelve el detalle de los saldos en banco a una fecha dada, y filtrado por el tipo de moneda.

Parámetros	Tipo	Descripción
<i>date</i>	string	Fecha.
<i>currency</i>	string	Tipo de moneda.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: codbanc, descripcion, tdc, saldoact, debitos_dif, creditos_dif, disponible, saldoc1.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_BalanceDBanks("2013-01-01", "BS");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 * [{"COdBANC": "110110003", "DESCRIPCION": "EFFECTIVO POR DEPOSITAR", "TDC": 0, "SALDOACT": 3750905.030, "DEBITOS_DIF": 0.000,
 * "CREDITOS_DIF": 1059751.570, "DISPONIBLE": 2691153.460, "SALDOC1": 0.000},
 * {"COdBANC": "110120001", "DESCRIPCION": "BANCO PROVINCIAL 01080927410100000000", "TDC": 0, "SALDOACT": -1284921.670,
 * "DEBITOS_DIF": 7382155.240, "CREDITOS_DIF": 7109694.870, "DISPONIBLE": -1012461.300, "SALDOC1": 457625.160},
 * {"COdBANC": "110120002", "DESCRIPCION": "BANCO PROVINCIAL prueba", "TDC": 0, "SALDOACT": -276.000, "DEBITOS_DIF": 0.000,
 * "CREDITOS_DIF": 0.000, "DISPONIBLE": -276.000, "SALDOC1": 0.000}, {"COdBANC": "110120003",
 * "DESCRIPCION": "BANCO BANFOANDES (NO USAR)", "TDC": 0, "SALDOACT": -30000.000, "DEBITOS_DIF": 0.000, "CREDITOS_DIF": 0.000,
 * "DISPONIBLE": -30000.000, "SALDOC1": 261507.370}, {"COdBANC": "110120004", "DESCRIPCION": "BANCO DE VENEZUELA 0584 GUACARA",
 * "TDC": 0, "SALDOACT": 70973.640, "DEBITOS_DIF": 1010812.260, "CREDITOS_DIF": 1074688.050, "DISPONIBLE": 7097.850, "SALDOC1": 224.150}]
 *
 * /* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
 */
```

## Report\_DebitCreditNotes()

Report\_DebitCreditNotes(string startDate, string endDate)

Devuelve el monto total de las notas de débito y crédito en un periodo dado.

Parámetros	Tipo	Descripción
<i>startDate</i>	string	Fecha inicial.
<i>endDate</i>	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: notasdeb, notascre.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_DebitCreditNotes("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"NOTASDEB":0.000,"NOTASCRE":0.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_DebitNotesD()

Report\_DebitNotesD(string startDate, string endDate)

Devuelve el detalle de las notas de débito en un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: codclie, document, eslibroi,fechae, numerod, tipocxc, afectavta, monto, montoneto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_DebitNotesD("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODCLIE":"15986321","DOCUMENT":"documento 001","ESLIBROI":1,"FECHAE":"2013-02-05","NUMEROD":"002563",
    * "TIPOCXC":20,"AFECTAVTA":1,"MONTO":150.000,"MONTONETO":150.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_CreditNotesD()

Report\_CreditNotesD([string](#) startDate, [string](#) endDate)

Devuelve el detalle de las notas de crédito en un periodo dado.

Parámetros	Tipo	Descripción
<i>startDate</i>	<a href="#">string</a>	Fecha inicial.
<i>endDate</i>	<a href="#">string</a>	Fecha final.
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: codclie, document, eslibroi,fechae, numerod, tipocxc, afectavta, monto, montoneto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_CreditNotesD("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"CODCLIE":"15986321","DOCUMENT":"documento 001","ESLIBROI":1,"FECHAE":"2013-02-05","NUMEROD":"002563",
    * "TIPOCX":30,"AFECTAVTA":1,"MONTO":150.000,"MONTONETO":150.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_AcctsReceivable()

Report\_AcctsReceivable()

Devuelve el saldo de las cuentas por cobrar.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<a href="#">string</a>	Cadena de texto en formato Json. Campos: tctasxcobrar.



Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_AcctsReceivable();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"TCTASXCOBRAR":23166879.620}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotAcctReceivable()

Report\_TotAcctReceivable(string endDate)

Devuelve el saldo de las cuentas por cobrar vencido a una fecha dada.

Parámetros	Tipo	Descripción
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: ctasxcobrar.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotAcctReceivable("2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"CTASXCOBRAR":1323185.670}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_TotAcctReceivable()

Report\_TotAcctReceivable(string endDate)

Devuelve el detalle del total de cuentas por cobrar vencido a una fecha.

Parámetros	Tipo	Descripción
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: codclie, descrip, documento,, fechavenc, monto, saldo.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotAcctReceivable("2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"CODCLIE":"J000111999","DESCRIP":"MOTORS, C. A.,"DOCUMENTO":"714","FEHAVENC":"2011-05-31","MONTO":156800.000,
 * "SALDO":156800.000}, {"CODCLIE":"J000111999","DESCRIP":"MOTORS, C. A.,"DOCUMENTO":"000790","FEHAVENC":"2011-11-22",
 * "MONTO":22288.000,"SALDO":22288.000}, {"CODCLIE":"V000222999","DESCRIP":"REINALDO HERNANDEZ",
 * "DOCUMENTO":"000875","FEHAVENC":"2012-05-23","MONTO":3234.560,"SALDO":3234.560}, {"CODCLIE":"J000333999",
 * "DESCRIP":"FRANK C.A.,"DOCUMENTO":"31122012","FEHAVENC":"2013-01-01","MONTO":267823.960,"SALDO":267823.960},
 * {"CODCLIE":"J000444999","DESCRIP":"COOPERATIVA PRIMERA, R. L.,"DOCUMENTO":"31122012",
 * "FEHAVENC":"2013-01-01","MONTO":230927.610,"SALDO":230927.610}, {"CODCLIE":"J000555999",
 * "DESCRIP":"ASOC. COOP. VIDRIOS","DOCUMENTO":"31122012","FEHAVENC":"2013-01-01","MONTO":302299.490,
 * "SALDO":302299.490}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_PurchasesIVA()

Report\_PurchasesIVA(string startDate, string endDate)

Devuelve el detalle del IVA en compras para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: orden, descripcion, fecha, signo, monto, iva.

### Código de ejemplo:

```
using Saint.Web.Admin;
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_PurchasesIVA("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
/* Este código genera el siguiente resultado:
 *
 * [{"ORDEN":"000000032","DESCRIPCION":"COMERCIALIZADORA REPUESTOS S","FECHA":"2013-01-18","SIGNO":1,"MONTO":447594.000,
 * "IVA":53711.280},{ "ORDEN":"000000033","DESCRIPCION":"COMERCIALIZADORA REPUESTOS S","FECHA":"2013-01-18","SIGNO":1,
 * "MONTO":97750.000,"IVA":11730.000},{ "ORDEN":"02660","DESCRIPCION":"ICE C.A.", "FECHA":"2013-01-22","SIGNO":1,"MONTO":553.660,
 * "IVA":66.440},{ "ORDEN":"02663","DESCRIPCION":"ICE WORLD C.A.", "FECHA":"2013-01-22","SIGNO":1,"MONTO":821.560,"IVA":98.590},
 * {"ORDEN":"0866","DESCRIPCION":"LUNCHERIA EL SAZON","FECHA":"2013-01-22","SIGNO":1,"MONTO":594.720,"IVA":71.370}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

### Report\_SalesIVA()

Report\_SalesIVA(string startDate,string endDate)

Devuelve el detalle del IVA en ventas para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: factura, codclie, descrip, fecha, monto, iva.

### Código de ejemplo:

```
using Saint.Web.Admin;
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesIVA("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
/* Este código genera el siguiente resultado:
 *
 * [{"FACTURA":"000982","CODCLIE":"J314480235","DESCRIP":"TRANSPORTE FEDETOUR'S CA","FECHA":"2013-01-11","MONTO":758975.340,
 * "IVA":91024.660},{ "FACTURA":"000983","CODCLIE":"J2000000020","DESCRIP":"CONSEJO COMUNAL LA FORTALEZA LA TRILLA-CANSAMACHO",
 * "FECHA":"2013-01-24","MONTO":1312.500,"IVA":157.500},{ "FACTURA":"000984","CODCLIE":"J400767385","DESCRIP":"EQUIPAMIENTOS ",
 * "FECHA":"2013-01-30","MONTO":610436.500,"IVA":73200.000},{ "FACTURA":"000985","CODCLIE":"V137703854","DESCRIP":"ARNALDO SALVES",
 * "FECHA":"2013-01-30","MONTO":232.140,"IVA":27.860},{ "FACTURA":"000986","CODCLIE":"J400767385","DESCRIP":"EQUIPAMIENTOS ",
 * "FECHA":"2013-02-04","MONTO":680436.500,"IVA":81600.000},
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_WithholdingIVAReceivableD()

Report\_WithholdingIVAReceivableD(string startDate,string endDate)

Devuelve el detalle de retención del IVA en cuentas por cobrar para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<hr/>		
Valor devuelto	string	Cadena de texto en formato Json. Campos: documento, descripcion, fecha, retencion.

Código de ejemplo:

```
using Saint.Web.Admin;
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_WithholdingIVAReceivableD("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
/* Este código genera el siguiente resultado:
* [{"DOCUMENTO":"RET DE IVA FACT N° 992","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RET DE IVA FACT N° 993","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RET DE IVA FACT N° 994","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RET DE IVA FACT N° 995","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RET IVA, FACT NRO. 000989","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RETEN. IVA FAC: 000989","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RETEN. IVA FAC: 000992","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RETEN. IVA FAC: 000993","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RETEN. IVA FAC: 000994","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240},
* [{"DOCUMENTO":"RETEN. IVA FAC: 000995","DESCRIPCION":"RUSTICOS C. A.", "FECHA":"2013-03-21", "RETENCION":80256.240}]
*/
/* Mensaje en caso de error: "{"error":"Desconocido"}" */
```

## Report\_WithholdingIVASalesD()

Report\_WithholdingIVASalesD(string startDate,string endDate)

Devuelve el detalle de retención del IVA en venta para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<hr/>		
Valor devuelto	string	Cadena de texto en formato Json. Campos: documento, descripcion, fecha, retencion.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_WithholdingIVASalesD("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"DOCUMENTO":"RET DE IVA FACT Nº 992","DESCRIPCION":"RUSTICOS C. A.,"FECHA":"2013-03-21","RETENCION":80256.240},
    * [{"DOCUMENTO":"RET DE IVA FACT Nº 993","DESCRIPCION":"RUSTICOS C. A.,"FECHA":"2013-03-21","RETENCION":80256.240},
    * [{"DOCUMENTO":"RET DE IVA FACT Nº 994","DESCRIPCION":"RUSTICOS C. A.,"FECHA":"2013-03-21","RETENCION":80256.240},
    * [{"DOCUMENTO":"RET DE IVA FACT Nº 995","DESCRIPCION":"RUSTICOS C. A.,"FECHA":"2013-03-21","RETENCION":80256.240},
    * [{"DOCUMENTO":"RET IVA, FACT NRO. 000989","DESCRIPCION":"RUSTICOS C. A.,"FECHA":"2013-03-21","RETENCION":80256.240}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_WithholdingIVAPurchasesD()

Report\_WithholdingIVAPurchasesD(string startDate,string endDate)

Devuelve el detalle de retención del IVA en compra para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: orden, descripcion, fecha, signo, monto, retencion.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_WithholdingIVAPurchasesD("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"ORDEN":"207425","DESCRIPCION":"FERRUMA, C.A.,"FECHA":"2013-01-25","SIGNO":1,"MONTO":2257.000,"RETENCION":203.130},
    * [{"ORDEN":"004645","DESCRIPCION":"VIGILANTES C.A.,"FECHA":"2013-01-22","SIGNO":1,"MONTO":9161.000,"RETENCION":824.490},
    * [{"ORDEN":"0181","DESCRIPCION":"QUIMISELA, C.A.,"FECHA":"2013-01-23","SIGNO":1,"MONTO":6000.000,"RETENCION":540.000},
    * [{"ORDEN":"0197","DESCRIPCION":"QUIMISELA, C.A.,"FECHA":"2013-01-23","SIGNO":1,"MONTO":15660.000,"RETENCION":1409.400},
    * [{"ORDEN":"003642","DESCRIPCION":"SERMEDOCA C.A.,"FECHA":"2013-01-22","SIGNO":1,"MONTO":10600.000,"RETENCION":0.000},
    * [{"ORDEN":"003667","DESCRIPCION":"SERMEDOCA C.A.,"FECHA":"2013-01-23","SIGNO":1,"MONTO":1150.000,"RETENCION":0.000},
    * [{"ORDEN":"0866","DESCRIPCION":"LUNCHERIA Y REST","FECHA":"2013-01-22","SIGNO":1,"MONTO":594.720,"RETENCION":53.530},
    * [{"ORDEN":"0867","DESCRIPCION":"LUNCHERIA Y REST","FECHA":"2013-01-22","SIGNO":1,"MONTO":886.600,"RETENCION":79.790}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_WithholdingIVA()

Report\_WithholdingIVA(string startDate,string endDate)

Devuelve el monto total de retención del IVA en compra y venta, para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: ivaventa, retivaventa, ivaventav, ivacompra, retivacompra.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_WithholdingIVA("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}
/* Este código genera el siguiente resultado:
 *
 * {"IVAVENTA":280683.01,"RETIVAVENTA":0.00,"IVAVENTAV":164410.02,"IVACOMPRA":236555.44,"RETIVACOMPRA":160085.94}
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_InventoryFixtures()

Report\_InventoryFixtures()

Devuelve el monto total del inventario y enseres actual.

Parámetros	Tipo	Descripción
no requiere		
Valor devuelto	string	Cadena de texto en formato Json. Campos: inveenser, inveactual.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_InventoryFixtures();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"INVEENSER":9696.211000,"INVEACTUAL":197569244.156300}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_InventoryFixturesD()

Report\_InventoryFixturesD(int type)

Devuelve el detalle del total del inventario y enseres actual.

Parámetros	Tipo	Descripción
type	int	Tipo: 0=Inventario,1=Enser
Valor devuelto	string	Cadena de texto en formato Json. Campos: fila, codprod, descrip, [inveactual   inveenser].

Código de ejemplo:

```
using Saint.Web.Admin;
using System;
using TestSaintWebAdmin.AdminServiceClient;
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_InventoryFixturesD(0, 1, 2);
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"FILA":1,"CODPROD":"20C05001","DESCRIP":"CHASIS CHEVROLET MOD NPR BUS","INVEACTUAL":5186143.92},
    * {"FILA":2,"CODPROD":"20C05002","DESCRIP":"CHASIS CHEVROLET SILVERADO C3500 4X4","INVEACTUAL":303562.96}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_DischargeInventoryD()

Report\_DischargeInventoryD(string startDate, string endDate)

Muestra los detalles de descargo de inventario para un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: documento, fecha, uso, operacion, descargos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_DischargeInventoryD("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"DOCUMENTO": "001833", "FECHA": "2013-01-09", "USO": "BONO 09895", "OPERACION": "DINV", "DESCARGOS": 397.890},
    * {"DOCUMENTO": "001834", "FECHA": "2013-01-10", "USO": "BONO", "OPERACION": "DINV", "DESCARGOS": 5634.710},
    * {"DOCUMENTO": "001835", "FECHA": "2013-01-11", "USO": "BONO 1170", "OPERACION": "DINV", "DESCARGOS": 359.560},
    * {"DOCUMENTO": "001849", "FECHA": "2013-01-11", "USO": "UNIDAD ENTREGADA A VENTAS", "OPERACION": "DINV", "DESCARGOS": 230426.130},
    * {"DOCUMENTO": "001836", "FECHA": "2013-01-12", "USO": "BONO 013666", "OPERACION": "DINV", "DESCARGOS": 50.420}]
    */
    /* Mensaje en caso de error: "{\"error\": \"Desconocido\"}" */
}
```

## Report\_SalesProd()

Report\_SalesProd(string startDate, string endDate)

Muestra el total de ventas brutas de productos para un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: monto.



Código de ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesProd("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"MONTO":1370023.358}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_SalesServ()

Report\_SalesServ(string startDate, string endDate)

Muestra el total de ventas brutas por servicios para un periodo dado.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: montoservicios.

Código de ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesServ("2013-01-01", "2013-01-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"MONTOSERVICIOS":870023.358}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_SalesProdD()

### Report\_SalesProdD()

Muestra el detalle de las ventas por producto de los últimos 6 meses.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: monto, anio, mes, producto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesProdD();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"MONTO":1983213.000,"ANIO":2013,"MES":5,"PRODUCTO":"FABR. E INST. MIDI-BUSS 32 PUESTOS"},
 * {"MONTO":1620000.000,"ANIO":2013,"MES":5,"PRODUCTO":"MIDI-BUSS 32 PUESTOS"},
 * {"MONTO":1290000.000,"ANIO":2013,"MES":6,"PRODUCTO":"MIDI-BUSS INTERURBANO DE 32 PUESTOS"},
 * {"MONTO":120000.000,"ANIO":2013,"MES":7,"PRODUCTO":"FABRICACION DE UNIDAD PLATAFORMA TIPO"},
 * {"MONTO":1290000.000,"ANIO":2013,"MES":7,"PRODUCTO":"TANQUE CISTERNA 12.000 LTS."},
 * {"MONTO":1290000.000,"ANIO":2013,"MES":7,"PRODUCTO":"MINI-BUS IVECO 3360MM 20 PUESTOS"},
 * {"MONTO":1290000.000,"ANIO":2013,"MES":8,"PRODUCTO":"AMBULANCIA TIPO I"},
 * {"MONTO":1700000.000,"ANIO":2013,"MES":9,"PRODUCTO":"FAB. E INST. DE FURGON DE ALUMINIO 23 M³"}]
 *
 */
/* Mensaje en caso de error: "{"error":"Desconocido"}" */
```

## Report\_PurchasesProdD()

### Report\_PurchasesProdD()

Muestra el detalle de las compras detalladas por producto de los últimos 6 meses.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: monto, anio, mes, producto.

### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_PurchasesProdD();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    * [{"MONTO":1351102.410,"ANIO":2013,"MES":7,"PRODUCTO":"SERVICIOS GENERALES"}, {"MONTO":787500.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"TUBO H.N. 1½" x 1½ * 1.4mm * 6MT"}, {"MONTO":787500.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"TUBO H.N. 3" x 1½ * 1.4mm * 6MT"}, {"MONTO":546632.120,"ANIO":2013,"MES":7,"PRODUCTO":"ASIENTO 5 PUESTOS"},
    * {"MONTO":540272.120,"ANIO":2013,"MES":7,"PRODUCTO":"ASIENTO 1 PUESTO"}, {"MONTO":495143.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"SERVICIOS GENERALES (EXENTO)"}, {"MONTO":441629.560,"ANIO":2013,"MES":7,
    * "PRODUCTO":"TRABAJOS DE ELECTRICIDAD A UNIDADES"}, {"MONTO":391908.110,"ANIO":2013,"MES":7,"PRODUCTO":"FLETES"},
    * {"MONTO":365124.040,"ANIO":2013,"MES":7,"PRODUCTO":"SISTEMA ELECTRICO MINI-BUSS 26 PUESTOS"}, {"MONTO":291115.000,
    * "ANIO":2013,"MES":7,"PRODUCTO":"ASIENTO 2 PUESTOS L. D. (26 PUESTOS)"}, {"MONTO":291115.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"ASIENTO 2 PUESTOS L. I. (26 PUESTOS)"}, {"MONTO":286048.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"GOMA TIPO \"U\" (REF 621-01)"}, {"MONTO":272518.000,"ANIO":2013,"MES":7,
    * "PRODUCTO":"GOMA BORDE VENTANA (REF 482-01)"}]
    */
    /* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
}
```

### Report\_TotalSales()

Report\_TotalSales(string startDate, string endDate)

Muestra el total de ventas para un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: totalfacturacion, impuestos, contado, credito, totaloperacion, costos, utilidad.

### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotalSales("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"TOTALFACTURACION":9055490.200,"IMPUESTOS":1086384.500,"CONTADO":2546472.000,"CREDITO":7595402.700,
    * "TOTALOPERACION":10141874.700,"COSTOS":1814242.860,"UTILIDAD":7241247.340}
    */
    /* Mensaje en caso de error: "{\"error\":\"Desconocido\"}" */
}
```

## Report\_SalesDiscounts()

Report\_SalesDiscounts(string startDate, string endDate)

Muestra el monto total de descuentos de ventas en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: descuento.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesDiscounts("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"DESCUENTO":0.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_CostsProdServD()

Report\_CostsProdServD(string startDate, string endDate)

Devuelve el monto detallado de costos de productos y servicios en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
<b>Valor devuelto</b>	string	Cadena de texto en formato Json. Campos: factura, cliente, descripcion, fecha, costos.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_CostsProdServD("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * [{"FACTURA":"000982","CLIENTE":"J999000111","DESCRIPCION":"TRANSPORTE S CA","FECHA":"2013-01-11",
    * "COSTOS":366542.170},{ "FACTURA":"000983","CLIENTE":"J999000222",
    * "DESCRIPCION":"CONSEJO COMUNAL ", "FECHA":"2013-01-24","COSTOS":327.680},
    * {"FACTURA":"000984","CLIENTE":"J999000333","DESCRIPCION":"EQUIPAMIENTOS C. A.", "FECHA":"2013-01-30",
    * "COSTOS":366542.170},{ "FACTURA":"000985","CLIENTE":"V999000444","DESCRIPCION":"ARNALDO SALVES",
    * "FECHA":"2013-01-30","COSTOS":115.000}]
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_FeesCosts()

Report\_FeesCosts(string startDate, string endDate)

Devuelve el monto total de comisiones y costos en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: comisiones, costomes.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_FeesCosts("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"COMISIONES":0.00,"COSTOMES":0.00}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_PurchaseLastMonths()

Report\_PurchaseLastMonths()

Devuelve el monto total de las compras de los últimos 6 meses.

Parámetros	Tipo	Descripción
<i>no requiere</i>		
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: mes, anio, monto.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_PurchaseLastMonths();
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"MES":1,"ANIO":2014,"MONTO":0.000}, {"MES":8,"ANIO":2013,"MONTO":1100.000}, {"MES":9,"ANIO":2013,"MONTO":0.000},
 * {"MES":10,"ANIO":2013,"MONTO":0.000}, {"MES":11,"ANIO":2013,"MONTO":0.000}, {"MES":12,"ANIO":2013,"MONTO":0.000}]
 */
/* Mensaje en caso de error: {"error":"Desconocido"} */
```

## Report\_SalesNumber()

Report\_SalesNumber(string startDate, string endDate)

Devuelve el número de ventas en un periodo.

Parámetros	Tipo	Descripción
<i>startDate</i>	<b>string</b>	Fecha inicial.
<i>endDate</i>	<b>string</b>	Fecha final.
<b>Valor devuelto</b>	<b>string</b>	Cadena de texto en formato Json. Campos: cuenta.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_SalesNumber("2013-01-01", "2013-03-31");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"CUENTA":17}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_ComsmSellers()

Report\_ComsmSellers(string startDate, string endDate)

Devuelve los montos totales de las comisiones de vendedores en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: comicob, comivta, total.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ComsmSellers("2013-01-01", "2013-06-30");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"COMICOB":1200.000,"COMIVTA":589.000,"TOTAL":1789.000}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

## Report\_ComsnSellers()

Report\_ComsnSellers(string startDate, string endDate)

Devuelve los montos detallados de las comisiones de vendedores en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: numerod, fecha, descripcion, mtocomicob, mtocomivta, total.

Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_ComsnSellersd("2013-01-01", "2013-06-30");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
}

/* Este código genera el siguiente resultado:
 *
 * [{"NUMEROD": "000982", "FECHA": "2013-01-11", "DESCRIPCION": "TRANSPORTE CA", "MTOCOMICOB": 0.000,
 * "MTOCOMIVTA": 0.000, "TOTAL": 0.000}, {"NUMEROD": "000983", "FECHA": "2013-01-24",
 * "DESCRIPCION": "CONSEJO COMUNAL ", "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000},
 * {"NUMEROD": "000984", "FECHA": "2013-01-30", "DESCRIPCION": "EQUIPAMIENTOS C. A.",
 * "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000}, {"NUMEROD": "000985", "FECHA": "2013-01-30",
 * "DESCRIPCION": "ARNALDO SALVES", "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000}, {"NUMEROD": "000986",
 * "FECHA": "2013-02-04", "DESCRIPCION": "EQUIPAMIENTOS C. A.", "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000,
 * "TOTAL": 0.000}, {"NUMEROD": "000987", "FECHA": "2013-02-04", "DESCRIPCION": "EQUIPAMIENTOS C. A.",
 * "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000}, {"NUMEROD": "000988", "FECHA": "2013-02-08",
 * "DESCRIPCION": "GABRIEL MARTINEZ", "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000}, {"NUMEROD": "000989",
 * "FECHA": "2013-02-14", "DESCRIPCION": "RUSTICOS C. A.", "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000,
 * "TOTAL": 0.000}, {"NUMEROD": "000990", "FECHA": "2013-02-20T00:00:00", "DESCRIPCION": "EQUIPAMIENTOS C. A.",
 * "MTOCOMICOB": 0.000, "MTOCOMIVTA": 0.000, "TOTAL": 0.000}
 *
 */
/* Mensaje en caso de error: "{"error": "Desconocido"}" */
```

## Report\_TotComsnSeller()

Report\_TotComsnSeller(string startDate, string endDate)

Devuelve el monto total de la comisión de vendedores en un periodo.

Parámetros	Tipo	Descripción
startDate	string	Fecha inicial.
endDate	string	Fecha final.
Valor devuelto	string	Cadena de texto en formato Json. Campos: comisiones_vend.



### Código de ejemplo:

```
using System;
using TestSaintWebAdmin.AdminServiceClient;

class Program
{
    static void Main(string[] args)
    {
        AdminServiceClient _serviceClient = new AdminServiceClient();
        string _result = string.Empty;
        _result = _serviceClient.Report_TotComsnSeller("2013-01-01", "2013-06-30");
        Console.WriteLine(_result);
        Console.ReadKey();
    }
    /* Este código genera el siguiente resultado:
    *
    * {"COMISIONES_VEND":4520.00}
    */
    /* Mensaje en caso de error: {"error":"Desconocido"} */
}
```

