

Ordenación de datos en Reporting Services

Luis Miguel Blanco Ancos

Algunas consideraciones iniciales

En este artículo, tal y como apunta su título, nos ceñiremos al modo en cómo podemos ordenar los datos mostrados por un informe, describiendo las técnicas que a nuestro alcance pone Reporting Services para la consecución de dicho fin.

Por tal motivo se asume que el lector conoce los aspectos elementales en el proceso de creación de un informe básico, a efectos de poder entrar directamente en la descripción de los procedimientos necesarios para ordenar un listado.

Respecto a los informes de ejemplo que utilizaremos, todos se encontrarán situados dentro de un proyecto de informes de Visual Studio .NET, y tomarán como origen de datos la tabla Customers de la base de datos Northwind de SQL Server. Igualmente, todos visualizarán los datos en una región de datos de tipo tabla de Reporting Services. Estos ejemplos se encuentran disponibles para su descarga como material de apoyo en la dirección www.dotnetmania.com.

Y ya sin más dilación comencemos nuestra andadura por el universo de los informes ordenados.

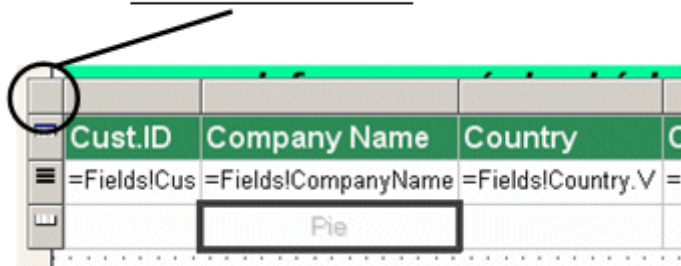
Aplicando un orden simple

El modo más básico de ordenar los datos de un informe consiste en establecer el criterio de ordenación de forma fija en la región de datos utilizada para visualizar la información.

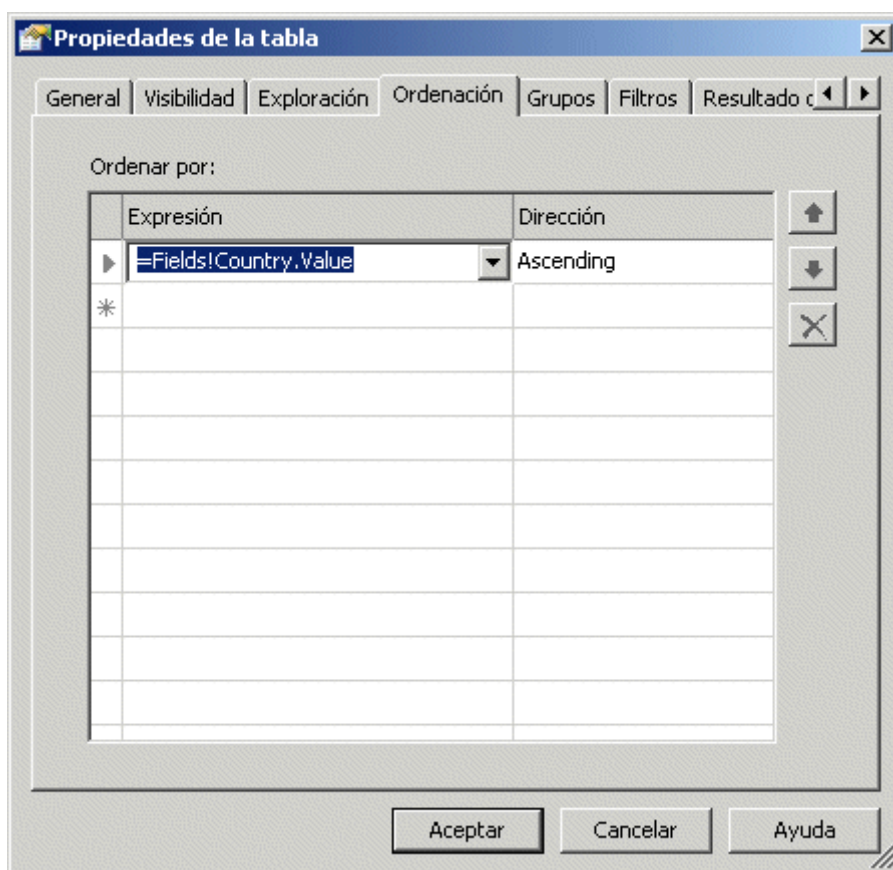
Como ejemplo de este tipo de caso crearemos un informe con el nombre rptOrdenBasico; una vez creado su origen de datos y añadidos los campos a mostrar en la tabla del diseñador, daremos los siguientes pasos para crear un orden por el campo Country:

En primer lugar, situados en la pestaña Diseño del informe seleccionaremos la región de datos tabla, y a continuación haremos clic derecho en el indicador de tabla, que es el recuadro situado en su esquina superior izquierda, como muestra la siguiente figura.

Indicador de tabla



Al aparecer el menú contextual elegiremos la opción Propiedades, que abrirá un cuadro de diálogo en el que nos situaremos en la pestaña Ordenación. En la columna Expresión seleccionaremos de la lista desplegable el campo Country, en la columna Dirección dejaremos el valor por defecto, Ascending, como vemos en la siguiente figura.



Pulsaremos el botón Aceptar y pasaremos a la pestaña *Vista previa* del informe para ver el resultado en ejecución, comprobando que los registros se ordenan por el campo anteriormente establecido.

Como habrá observado, podemos construir nuestro orden para el informe basado en varios campos, tan sólo hemos de ir agregándolos en la forma que acabamos de explicar, y establecer su prioridad utilizando los botones de flecha arriba/abajo suministrados a tal efecto en esta ventana. La siguiente figura muestra este informe en ejecución con un orden basado en los campos Country y City.

WARTH	Wartian Herkku	Finland	Oulu
FOLIG	Folies gourmandes	France	Lille
VICTE	Victuailles en stock	France	Lyon
BONAP	Bon app'	France	Marseille
DUMON	Du monde entier	France	Nantes
FRANR	France restauration	France	Nantes
PARIS	Paris spécialités	France	Paris
SPECD	Spécialités du monde	France	Paris
VINET	Vins et alcools Chevalier	France	Reims
BLONP	Blondesd'sl père et fils	France	Strasbourg
LAMAI	La maison d'Asie	France	Toulouse
LACOR	La corne d'abondance	France	Versailles
DRACD	Drachenblut Delikatessen	Germany	Aachen
ALFKI	Alfreds Futterkiste	Germany	Berlin
KOENE	Königlich Essen	Germany	Brandenburg
QUICK	QUICK-Stop	Germany	Cunewalde
LEHMS	Lehmanns Marktstand	Germany	Frankfurt a.M.
OTTIK	Ottilies Käseladen	Germany	Köln
MORGK	Morgenstern Gesundkost	Germany	Leipzig
BLAUS	Blauer See	Germany	Mannheim

Establecimiento del orden en tiempo de ejecución

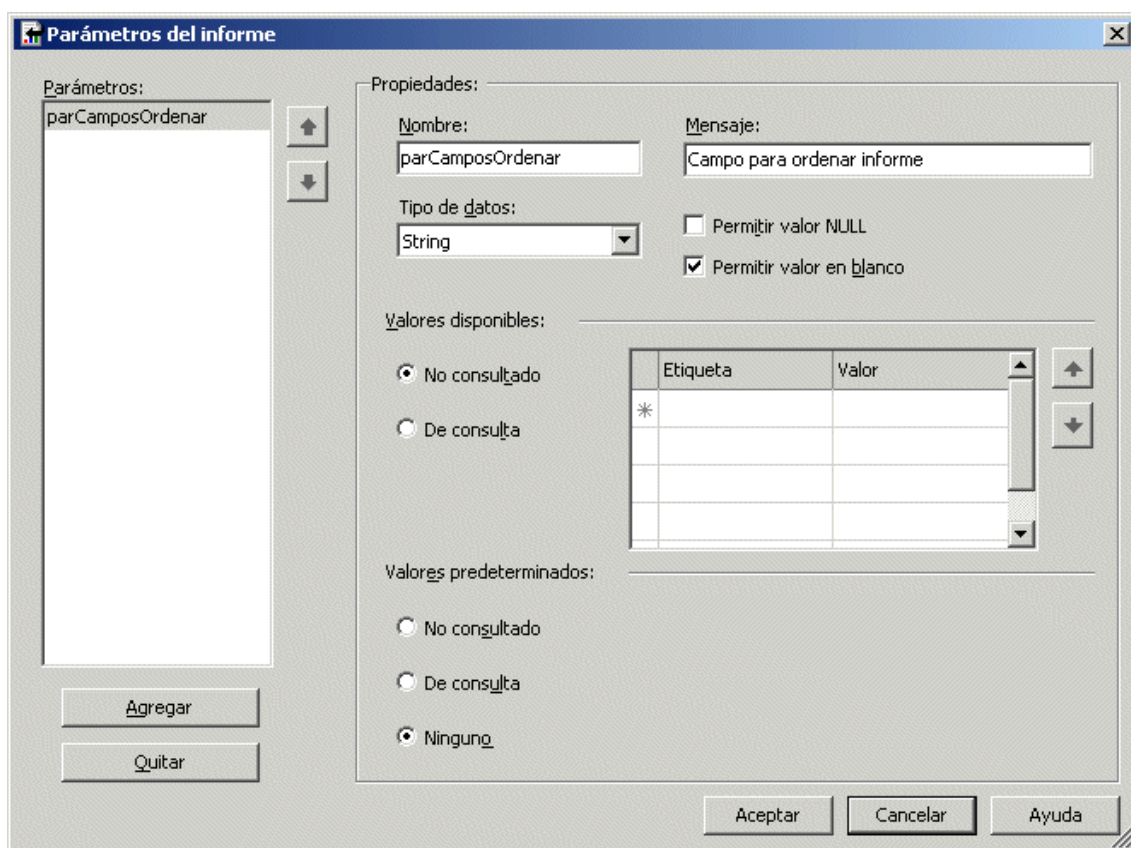
Puede ser puntualmente interesante establecer el orden de un informe de forma fija e inamovible en un campo, pero la práctica nos dice en la mayoría de las ocasiones que el mejor diseño en este sentido es aquel que permite al usuario realizar la selección de aquellos campos por los que necesita ordenar el listado.

Para solventar este tipo de situaciones Reporting Services pone a nuestra disposición los denominados *parámetros de informe*. Gracias a un parámetro tendremos la capacidad de asignar el campo de ordenación al ejecutar el informe, evitando la rigidez que supone crear un orden en diseño que no podemos alterar, como ocurría en el caso anterior.

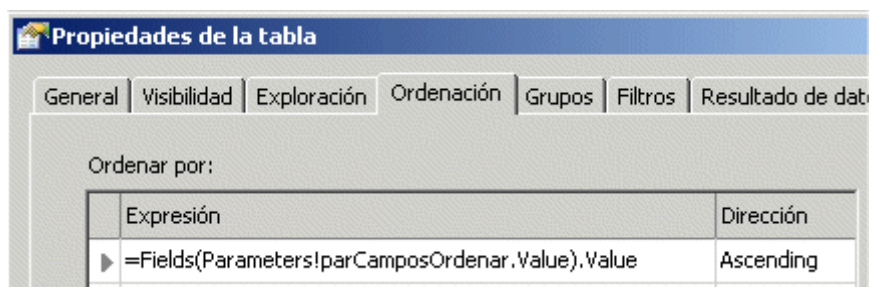
Ilustraremos este escenario de trabajo creando un nuevo informe con el nombre rptOrdenParametro, en el que incluiremos un parámetro asociado al orden de su región de datos, que nos permita seleccionar qué campo queremos utilizar para ordenar.

Crear un parámetro es una labor muy sencilla; una vez que nos situemos en la pestaña Diseño del informe, seleccionaremos la opción de menú *Informe + Parámetros del informe*, que abrirá el cuadro de diálogo para el mantenimiento de parámetros. Pulsando el botón Agregar, crearemos un parámetro con el nombre parCamposOrdenar, de tipo String, como muestra la siguiente figura.

Ordenación de datos en Reporting Services



Tras asignar los valores de configuración necesarios en este cuadro de diálogo y pulsar el botón Aceptar, el parámetro será creado, por lo que nuestro siguiente paso consistirá en asociarlo con la expresión de orden del informe. Para ello, abriremos la ventana de propiedades de la tabla, y en la pestaña Ordenación escribiremos la expresión de forma que la colección Fields del informe utilice el parámetro en lugar de un campo fijo, como puede ver en la siguiente figura.



Al ejecutar este informe desde la pestaña *Vista previa* no se mostrarán inmediatamente los datos como sucedía en el pasado ejemplo, ya que el motor de Reporting Services no sabe a priori el campo por el que queremos ordenar. En esta ocasión aparecerá en la barra de herramientas del informe una caja de texto perteneciente al parámetro creado, en la que teclearemos el nombre del campo de orden y pulsaremos el botón *Ver informe* para ejecutar el listado, como vemos en la siguiente figura.



The screenshot shows a report viewer interface. At the top, there are tabs for 'Datos', 'Diseño', and 'Vista previa'. Below the tabs, there is a field labeled 'Campo para ordenar informe' with the value 'CompanyName' and a 'Ver informe' button. The main content area displays a table with the following data:

Cust.ID	Company Name	Country	City
ALFKI	Alfreds Futterkiste	Germany	Berlin
ANATR	Ana Trujillo Emparedados y helados	Mexico	México D.F.

Téngase en cuenta que utilizando esta técnica se distinguen mayúsculas y minúsculas, por lo que al escribir el nombre del campo en el parámetro debemos hacerlo exactamente igual que aparece en la base de datos, o se producirá un error.

Empleando una consulta dinámica

Nuestra siguiente forma de ordenación consiste en el uso de una consulta dinámica de Reporting Services, característica que nos permite modificar en tiempo de ejecución la consulta SQL incluida en el informe, encargada de extraer los datos a visualizar.

Una consulta dinámica se destaca porque dispone de una parte variable que está representada por un parámetro al que, como ya hemos visto, podemos asignar valor en tiempo de ejecución, siendo lo que dota a este tipo de consultas de su verdadera potencia.

Como ejemplo de este tipo de ordenación crearemos el informe rptOrdenConsultaDinamica, que inicialmente elaboraremos de igual modo que los ejemplos anteriores en lo que respecta a su conjunto de datos y diseño.

Al crear el parámetro para este informe lo definiremos como una lista de valores, en la que asignaremos una etiqueta identificativa y el nombre del campo de orden para cada elemento de la lista. La ventaja de crear el parámetro de esta forma radica en que facilitamos al usuario la selección del campo de orden, evitamos potenciales errores por una incorrecta escritura del nombre del campo, y sólo permitimos que la ordenación se haga por los campos que realmente nos interesa exponer. Adicionalmente, y para que al comenzar a ejecutar el informe, este no aparezca vacío, introduciremos también como valor predeterminado uno de los nombres de campo. Véase la siguiente figura.

Ordenación de datos en Reporting Services

The screenshot shows the 'Propiedades' (Properties) dialog box for a report parameter. The 'Nombre' (Name) field is 'parCamposOrdenar' and the 'Mensaje' (Message) field is 'Campos para ordenar informe'. The 'Tipo de datos' (Data type) is set to 'String'. There are two checkboxes: 'Permitir valor NULL' (unchecked) and 'Permitir valor en blanco' (checked). Under 'Valores disponibles' (Available values), the 'No consultado' (Not queried) radio button is selected. A table lists the available values:

Etiqueta	Valor
Código	CustomerID
Empresa	CompanyName
País	Country
Ciudad	City

Under 'Valores predeterminados' (Default values), the 'No consultado' radio button is selected. The 'Company Name' field is populated with 'CompanyName' and has a formula icon to its right.

Seguidamente volveremos a la pestaña Datos del informe y cambiaremos la consulta SQL para convertirla a dinámica, lo cual conseguimos encerrando entre comillas la parte fija de la consulta, concatenándola a continuación al parámetro del informe. Como resultado obtenemos una expresión evaluable por el motor de informes. Ver el siguiente código fuente.

```
= "SELECT * FROM CUSTOMERS ORDER BY " & Parameters!parCamposOrdenar.Value
```

Es importante hacer notar en este punto, que para que una consulta dinámica se ejecute adecuadamente, debemos eliminar todos los retornos de carro que pudieran existir. En nuestro ejemplo no se da tal circunstancia, pero téngase en cuenta para el caso de consultas SQL con mucho código.

Completados todos los pasos de creación de este informe, al ser ejecutado desde la vista previa nos ofrecerá el parámetro como una lista desplegable, de la que seleccionaremos el campo para ordenar los datos. Ver la siguiente figura.



Cuando diseñemos un informe que vaya a necesitar una consulta dinámica, recuerde que es recomendable comenzar el diseño del modo habitual, utilizando una consulta normal que devuelva los campos que vayamos a visualizar. Una vez que hayamos depositado dichos campos en el diseñador del informe, volveremos a la pestaña Datos y convertiremos la consulta en dinámica. Si empezamos la creación del informe directamente con una consulta dinámica, no se generarán automáticamente los campos que necesitemos emplear en la zona de diseño, y tendremos que crearlos manualmente.

Ataque directo al código RDL

Todo el proceso de creación de un informe a través del diseñador tiene como resultado la producción de un archivo con la extensión .RDL (Report Language Definition), que como su propio nombre indica, consiste en un lenguaje de definición basado en XML, que se utiliza en Reporting Services para escribir el código de los informes.

Cuando estamos situados en el diseñador de informes, podemos visualizar el código RDL del mismo, para ello seleccionaremos la opción de menú de Visual Studio .NET *Ver + Código*. Adicionalmente, si el informe tiene sus datos ordenados, existirá un nodo con el nombre *Sorting*, que incluirá el subnodo *SortBy*, el cual a su vez que contendrá los subnodos *SortExpression* y *Direction*, que son los que definen respectivamente el campo y dirección de ordenación para el informe, como vemos en la siguiente figura.

```

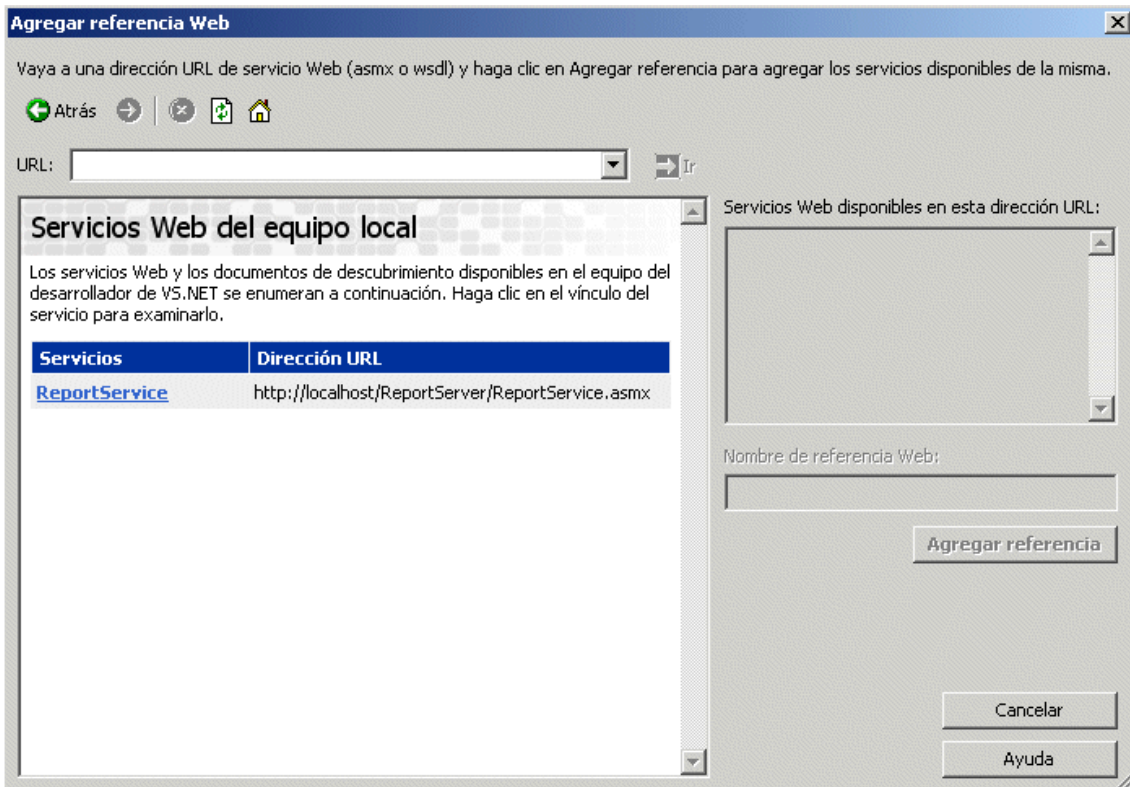
rptOrdenExterno.rdl [Design] | rptOrdenExterno.rdl [XML]
    </Style>
    <rd:DefaultName>City</rd:DefaultName>
    <CanGrow>true</CanGrow>
    <Value>=Fields!City.Value</Value>
  </Textbox>
</ReportItems>
</TableCell>
</TableCells>
</TableRow>
</TableRows>
<Sorting>
  <SortBy>
    <SortExpression>=Fields!Country.Value</SortExpression>
    <Direction>Ascending</Direction>
  </SortBy>
</Sorting>
</Details>
<DataSetName>Northwind</DataSetName>
  
```

La generación de código RDL del informe es un aspecto del que no tenemos que preocuparnos, ya que es el diseñador de informes el que se encarga de esta tarea, sin embargo, vamos a plantear a continuación un escenario de trabajo un tanto particular.

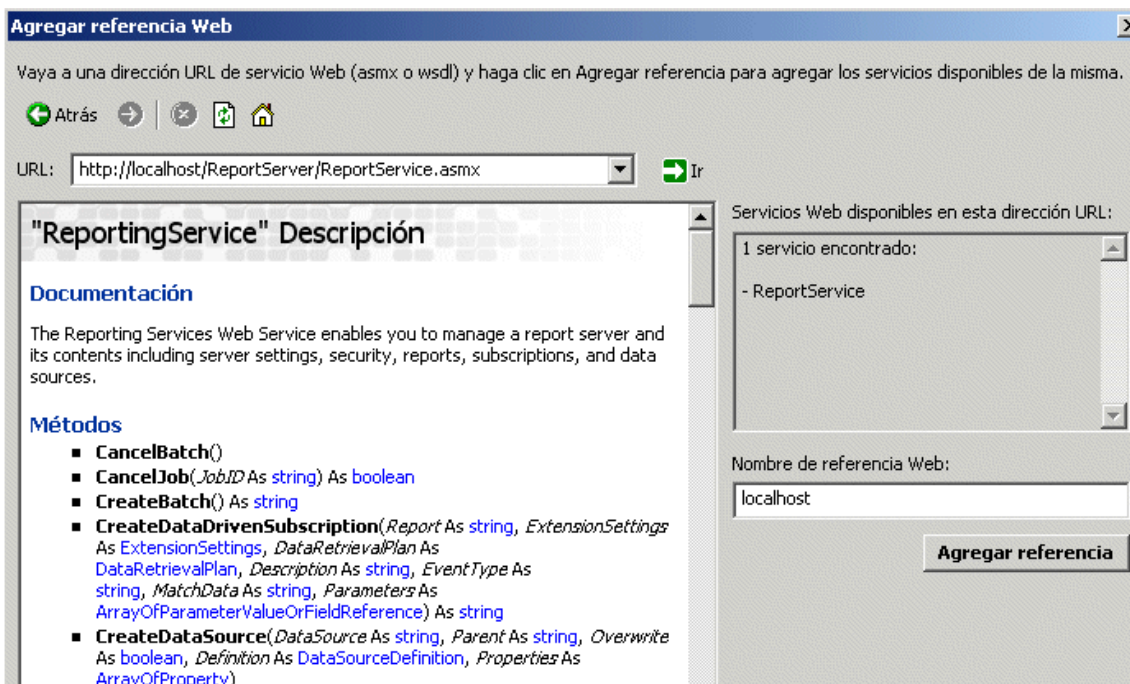
Supongamos que hemos creado un informe con el nombre rptOrdenExterno, ordenado de forma fija por el campo Country, lo que hace que no podemos cambiar su orden mediante parámetros; pero tenemos la necesidad puntual de cambiar dicha ordenación a otro campo distinto desde una aplicación externa que acceda al servidor de informes y haga esta operación.

Para lograr este propósito tenemos que comunicarnos con el servicio web del servidor de informes, y a través del mismo acceder a nuestro informe, para poder manipularlo por código. A continuación describimos los pasos requeridos para llevar esta operación a cabo.

En primer lugar crearemos un nuevo proyecto de tipo aplicación Windows (o ASP.NET según se prefiera) con el nombre CambiarOrdenInforme, y seleccionaremos la opción de menú *Proyecto + Agregar referencia web*, que mostrará el cuadro de diálogo para la búsqueda de servicios web. Haciendo clic en el enlace *Servicios web del equipo local*, se realizará un rastreo de los instalados en nuestra máquina, localizando el correspondiente al servidor de informes: ReportService, como vemos en la siguiente figura.

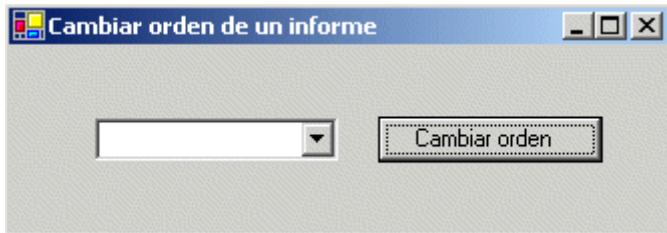


Tras hacer clic sobre el nombre de este servicio web pulsaremos el botón *Agregar referencia*, lo que establecerá el enlace entre nuestro proyecto y dicho servicio, para así poder manipular los informes publicados en el servidor de informes a través de los métodos proporcionados por este servicio. Ver la siguiente figura.



Seguidamente añadiremos al formulario de la aplicación un ComboBox, que rellenaremos con los nombres de los campos sobre los que ordenaremos, y un Button,

que al ser pulsado, ejecutará el código encargado de modificar el orden de los datos del informe.



Como paso final del proceso, en el evento Click del botón escribiremos el siguiente código fuente, en el cual, tras instanciar un objeto correspondiente al servicio web e identificarnos como un usuario válido, obtendremos la definición del informe. Posteriormente, gracias a las clases del espacio de nombres System.Xml, cargaremos la definición del informe en un documento XML, y obtendremos el valor del nodo que contiene la expresión de ordenación, modificándola por el campo que hemos seleccionado en el formulario. Finalmente, pasaremos el documento XML del informe a un stream en memoria, y este a un array de bytes que grabaremos como nueva definición del informe. En el siguiente código fuente se muestran las acciones que acabamos de describir.

```
Imports CambiarOrdenInforme.localhost
Imports System.Xml
Imports System.Text
Imports System.Net
Imports System.IO
'....
Private Sub btnCambiarOrden_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCambiarOrden.Click
    Dim oRS As ReportingService
    Dim sInforme As String
    Dim aBytDefInforme() As Byte
    Dim sCodInforme As String
    Dim oXMLInforme As XmlDocument
    Dim oNodoOrden As XmlNode
    Dim mslInforme As MemoryStream
    Dim aBytNuevoInforme() As Byte
    Dim aAvisos() As Warning

    ' instanciar un objeto del servicio web
    ' del servidor de informes
    oRS = New ReportingService

    ' asignar credenciales de identificación
    oRS.Credentials = CredentialCache.DefaultCredentials

    ' informe a manipular incluyendo ruta
```

```
sInforme = "/InformesOrdenados/rptOrdenExterno"  
  
' obtener el contenido del informe en formato binario  
aBytDefInforme = oRS.GetReportDefinition(sInforme)  
  
' convertir el contenido del informe a texto  
sCodInforme = Encoding.Default.GetString(aBytDefInforme)  
  
' crear un documento xml y cargar dentro el texto del informe  
oXMLInforme = New XmlDocument  
oXMLInforme.LoadXml(sCodInforme)  
  
' localizar el elemento que contiene el campo de orden  
' y cambiarlo por el valor del ComboBox  
oNodoOrden = oXMLInforme.GetElementsByTagName("SortExpression")(0)  
oNodoOrden.InnerText = "=Fields!" & Me.cboCampos.Text & ".Value"  
  
' pasar el informe a un stream  
msInforme = New MemoryStream  
oXMLInforme.Save(msInforme)  
  
' posicionar el stream saltando  
' los códigos de control  
msInforme.Position = 3  
  
' pasar el stream a un array de bytes  
aBytNuevoInforme = New Byte(msInforme.Length - 4) {}  
msInforme.Read(aBytNuevoInforme, 0, aBytNuevoInforme.Length)  
  
' grabar el array de bytes que contiene  
' el informe modificado en el servidor de informes  
aAvisos = oRS.SetReportDefinition(sInforme, aBytNuevoInforme)  
  
' si hay problemas al actualizar  
' se muestran con el array de objetos Warning  
If Not (aAvisos Is Nothing) Then  
    For Each oAviso As Warning In aAvisos  
        MessageBox.Show(oAviso.Message)  
    Next  
Else  
    MessageBox.Show("Establecido nuevo orden de datos en el informe")  
End If  
End Sub
```

Una vez modificado el orden del informe de esta manera, podemos abrir una ventana del navegador web e introducir la siguiente dirección, para comprobar que efectivamente, los datos se ordenan ahora en base al nuevo campo elegido.

<http://localhost/ReportServer?/InformesOrdenados/rptOrdenExterno>

Informe de conclusiones

A lo largo de este artículo hemos introducido al lector en las capacidades de ordenación de datos básica y avanzada con Reporting Services, una de las muchas características de esta interesante herramienta de generación de informes. Esperamos que todo lo aquí comentado le sea de utilidad en la creación de sus propios listados.