

Manipulación de datos en Silverlight mediante ADO.NET Data Services (I)

Luis Miguel Blanco Ancos

Un claro exponente de toda esta amalgama de tecnologías de acceso a datos lo constituye ADO.NET Data Services –proyecto inicialmente conocido con el nombre clave Astoria–, que a lo largo de las sucesivas entregas que componen este artículo abordaremos como mecanismo para la manipulación de datos, dentro del contexto de aplicaciones Silverlight.

En esta primera entrega realizaremos una descripción general de ADO.NET Data Services, así como un ejemplo de creación de un modelo y servicio de datos, sobre el que posteriormente efectuaremos pruebas de acceso y consultas. Si bien es cierto que el título del artículo indica que utilizaremos Silverlight, por cuestiones de espacio no lo haremos en esta parte, reservando los ejemplos de acceso al servicio desde la interfaz de usuario para la segunda entrega.

ADO.NET Data Services a grandes rasgos

Según se explica en [1], podemos definir ADO.NET Data Services como una tecnología basada en la combinación de una serie de librerías y patrones, cuyo principal objetivo radica en posibilitar la creación y consumo de servicios orientados a datos, los cuales se integren de forma fácil y natural en aplicaciones ejecutadas dentro del entorno Web.

Los servicios creados con ADO.NET Data Services, posibilitan la obtención, filtrado, ordenación y demás operaciones habituales sobre los datos. Para lograrlo, se utilizan identificadores URI, que mediante los verbos HTTP: GET, PUT, POST y DELETE, permiten interactuar con los datos, los cuales se representan como una colección de recursos de tipo REST, que se transmiten utilizando formatos como AtomPub (Atom Publishing Protocol) o JSON (JavaScript Object Notation). Adicionalmente, para organizar y representar conceptualmente el conjunto de entidades y relaciones (asociaciones) que componen un servicio de datos, podemos utilizar Entity Data Model como herramienta de modelado.

A continuación, una vez realizada la introducción teórica a esta tecnología, pasaremos a la parte práctica, en la cual describiremos las principales acciones que necesitaremos implementar para crear un servicio de tipo ADO.NET Data Services, cuyos datos puedan ser consumidos en un futuro por una aplicación cliente desarrollada en Silverlight.

Definiendo el Entity Data Model

Es posible crear un servicio de datos de ADO.NET a partir de un origen de datos que implemente la interfaz `IQueryable<T>` y, opcionalmente, la interfaz `IUpdatable`, ya sea el origen de datos relacional o no [2]. En nuestro caso emplearemos una fuente de datos relacional como es la base de datos de ejemplo Northwind de SQL Server.

Debido a la naturaleza relacional de esta fuente de datos, es preciso definir un modelo de las entidades que utilizaremos desde el servicio de datos que crearemos posteriormente.

En primer lugar iniciaremos Visual Studio 2008 y crearemos un nuevo proyecto de tipo Silverlight, al que daremos el nombre `ServicioDatosConsultaSL` (todos los ejemplos expuestos en el presente artículo se han desarrollado con Visual Studio 2008 y .NET Framework 3.5 Service Pack 1, pudiendo ser descargados desde el sitio web <http://www.dotnetmania.com>). A continuación nos situaremos en el Explorador de Soluciones, haremos clic derecho sobre el proyecto Web de esta solución y seleccionaremos la opción *Add > New Item*, que abrirá el correspondiente cuadro de diálogo, dentro del cual haremos clic en el elemento *ADO.NET Entity Data Model* para crear un nuevo modelo de entidades, que grabaremos en un archivo con el nombre `mdlNorthwind.edmx`.

A continuación se iniciará el asistente para la creación del modelo de entidades, en el que, como primer paso, indicaremos que nuestro modelo se generará a partir de una base de datos, seleccionando el icono *Generate from database*.

En el siguiente paso, que vemos en la figura 1, proporcionaremos la información de conexión a la base de datos, obteniendo como resultado la cadena de conexión que utilizará el servicio para acceder a la fuente de datos, así como el nombre con el que dicha conexión se almacenará en el archivo `Web.config`.

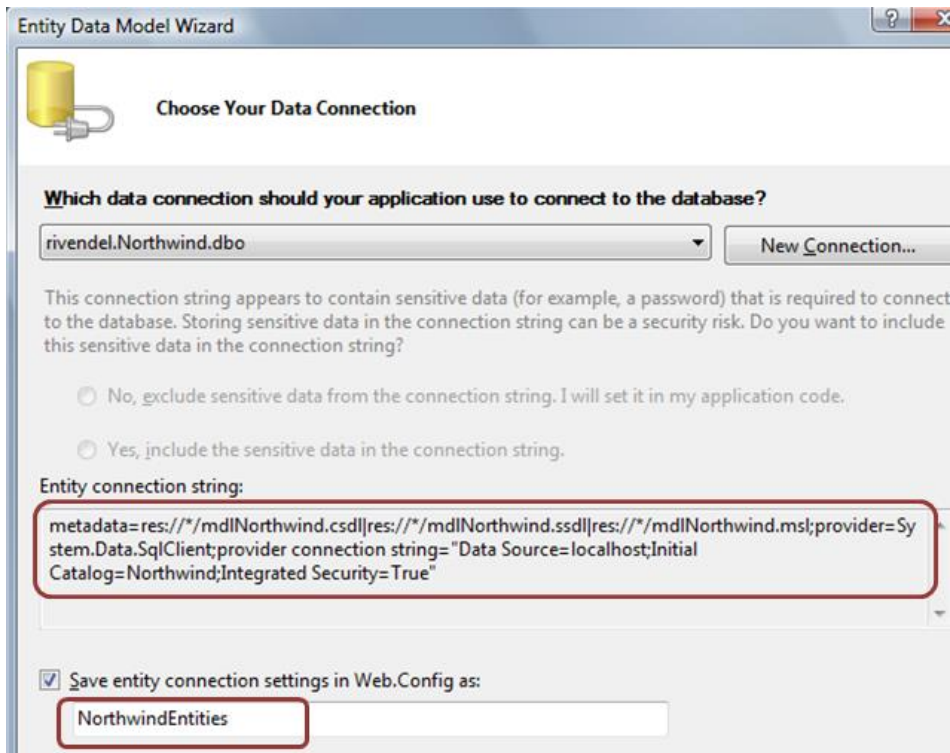


Figura 1. Cadena de conexión del modelo de datos.

Al llegar al final de este asistente, seleccionaremos aquellos elementos de la base de datos que conformarán las entidades de nuestro modelo, que en nuestro caso serán las tablas Customers, Orders, Products, Shippers y Suppliers.

Pulsando el botón Finish, el asistente generará el modelo a partir de los valores suministrados, ofreciéndonos el diseñador *Entity Data Model* (EDM a partir de ahora), con la representación conceptual en forma de entidades, de las tablas seleccionadas. En el caso de que entre las tablas agregadas al modelo existan relaciones, estas serán denominadas asociaciones dentro del diseñador.

Al igual que ocurre con otros diseñadores, el resultado del diseño que realicemos con el EDM se reflejará en un archivo de código, conteniendo una clase que representa al propio modelo o contexto de entidades, cuyo nombre, NorthwindEntities, como podemos apreciar, se compone del nombre de la base de datos más el sufijo Entities. Por otro lado, se crearán tantas clases como entidades hayamos definido en el modelo. Este conjunto de clases será utilizado por el servicio de datos en las operaciones de mapeo y transmisión de datos entre el cliente y el servidor.

También resulta importante destacar que el modelo obtenido no es algo rígido ni inmutable, sino que podemos editar las entidades, modificando sus propiedades, añadir nuevas entidades, etc.

Creación del servicio de datos

Una vez creado el modelo de entidades, el siguiente paso consiste en crear el servicio de datos, para lo cual volveremos al explorador de soluciones, y haciendo de nuevo clic derecho en el proyecto Web, seleccionaremos la opción *Add > New Item*, eligiendo esta vez el icono *ADO.NET Data Service*, al que daremos el nombre *wdsNorthwind.svc*.

Después de aceptar este cuadro de diálogo, nuestro servicio de datos será creado y Visual Studio 2008 abrirá un editor con el código fuente del mismo, donde tendremos que añadir código adicional para completar su configuración.

Por una parte, podemos observar que la clase correspondiente al servicio hereda del tipo genérico *DataService*, debiendo sustituir el comentario que aparece con el texto *"/*TODO: put your data source class name here*/"* por la clase que representa al modelo de entidades que recientemente acabamos de crear: *NorthwindEntities*.

Por otro lado, en el método *InitializeService* tomaremos el parámetro *config*, de tipo *IDataServiceConfiguration*, para establecer las reglas de acceso al servicio llamando a su método *SetEntitySetAccessRule*; al que pasaremos como primer parámetro el nombre de las entidades a las que permitiremos acceso —el asterisco indica que se permiten todas las entidades—, mientras que en el segundo parámetro indicaremos el tipo de permiso, que para este ejemplo será sin restricciones. Todo ello lo vemos en el listado 1.

```
public class wdsNorthwind : DataService<NorthwindEntities>
{
    // This method is called only once to initialize service-wide policies.
    public static void InitializeService(IDataServiceConfiguration config)
    {
        config.SetEntitySetAccessRule("*", EntitySetRights.All);
    }
}
```

Listado 1. Inicialización del servicio de datos.

Verificando el funcionamiento del servicio de datos

Tras finalizar la creación del servicio, el siguiente paso lógico consistirá en comprobar su funcionamiento.

El modo más inmediato de realizar estas pruebas pasa por escribir la ruta del servicio de datos en nuestro navegador Web.

Si nos encontramos en Visual Studio 2008, desde el Explorador de Soluciones haremos clic derecho en el servicio, seleccionando la opción *Set As Start Page*, con lo que al ejecutar el proyecto desde el entorno de desarrollo, la ruta que mostrará el navegador, debido a que estamos utilizando el servidor Web de desarrollo, será similar a la siguiente:

<http://localhost:55721/wdsNorthwind.svc/>

En el caso de que ya tengamos alojado el servicio en un directorio virtual mediante IIS, la ruta tendrá el formato habitual, pero apuntando, como es lógico, al servicio:

<http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/>

En ambos casos, el navegador mostrará, como resultado de la llamada al servicio, un listado con la representación de las entidades que lo componen en Atom –formato predeterminado de salida empleado por ADO.NET Data Services—, como vemos en la figura 2.

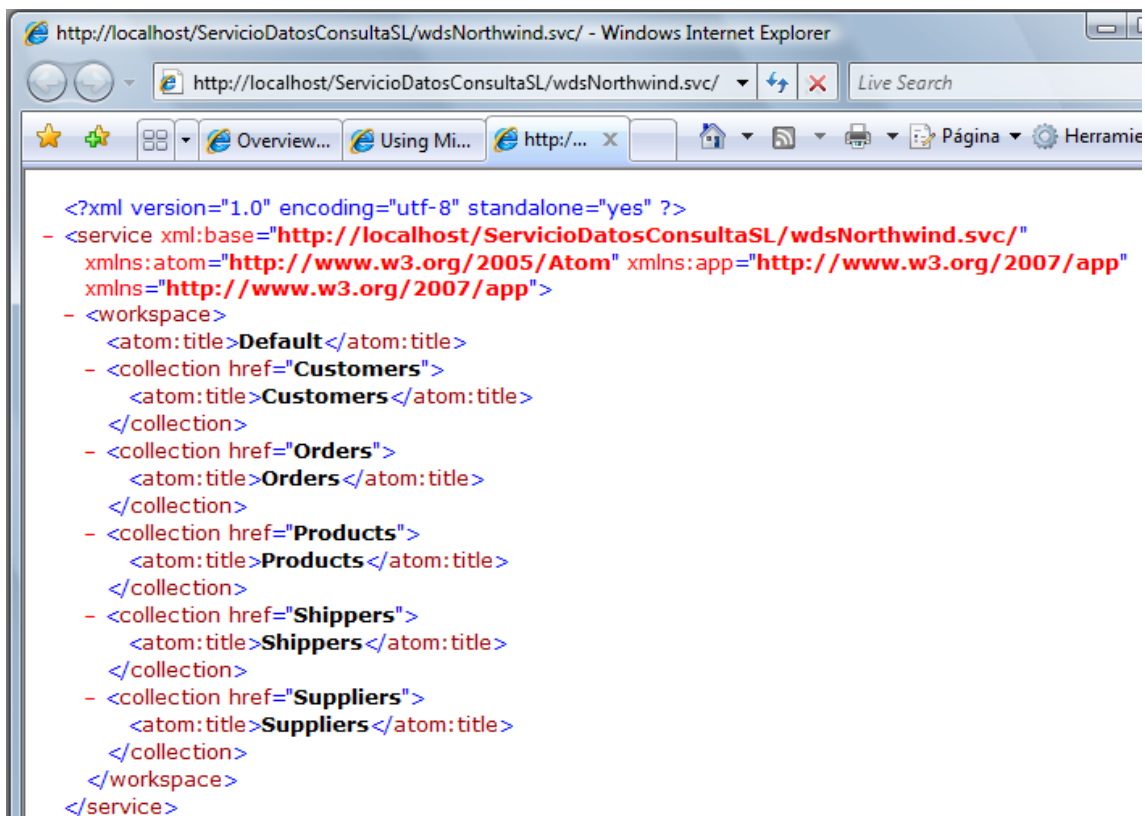


Figura 2. Listado de entidades del servicio en formato Atom.

Acceso al contenido de las entidades del servicio

Pero como es natural, además de haber conseguido llegar a la puerta de entrada de nuestro servicio de datos, lo lógico es que queramos acceder a su interior, para así obtener la información contenida en sus entidades. Esto será algo que conseguiremos mediante la combinación de un conjunto de parámetros, operadores y funciones, que pasados al query string de la ruta de acceso al servicio, nos permitirán moldear los datos resultantes de acuerdo a nuestras necesidades.

El primer ejemplo en este sentido consistirá en obtener el contenido al completo de una de las entidades del servicio, para lo que partiremos de cualquiera de las rutas

expuestas en el apartado anterior, añadiéndole el nombre de la entidad –Suppliers en este caso—de la siguiente manera.

<http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers>

La respuesta obtenida en el navegador –parte de la cual vemos en la figura 3— después de ejecutar esta llamada al servicio, muestra cómo cada uno de los elementos <entry> representa un registro de la tabla Suppliers, conteniendo todo el detalle relativo a ese elemento en particular: identificador único de acceso, propiedades, etc.

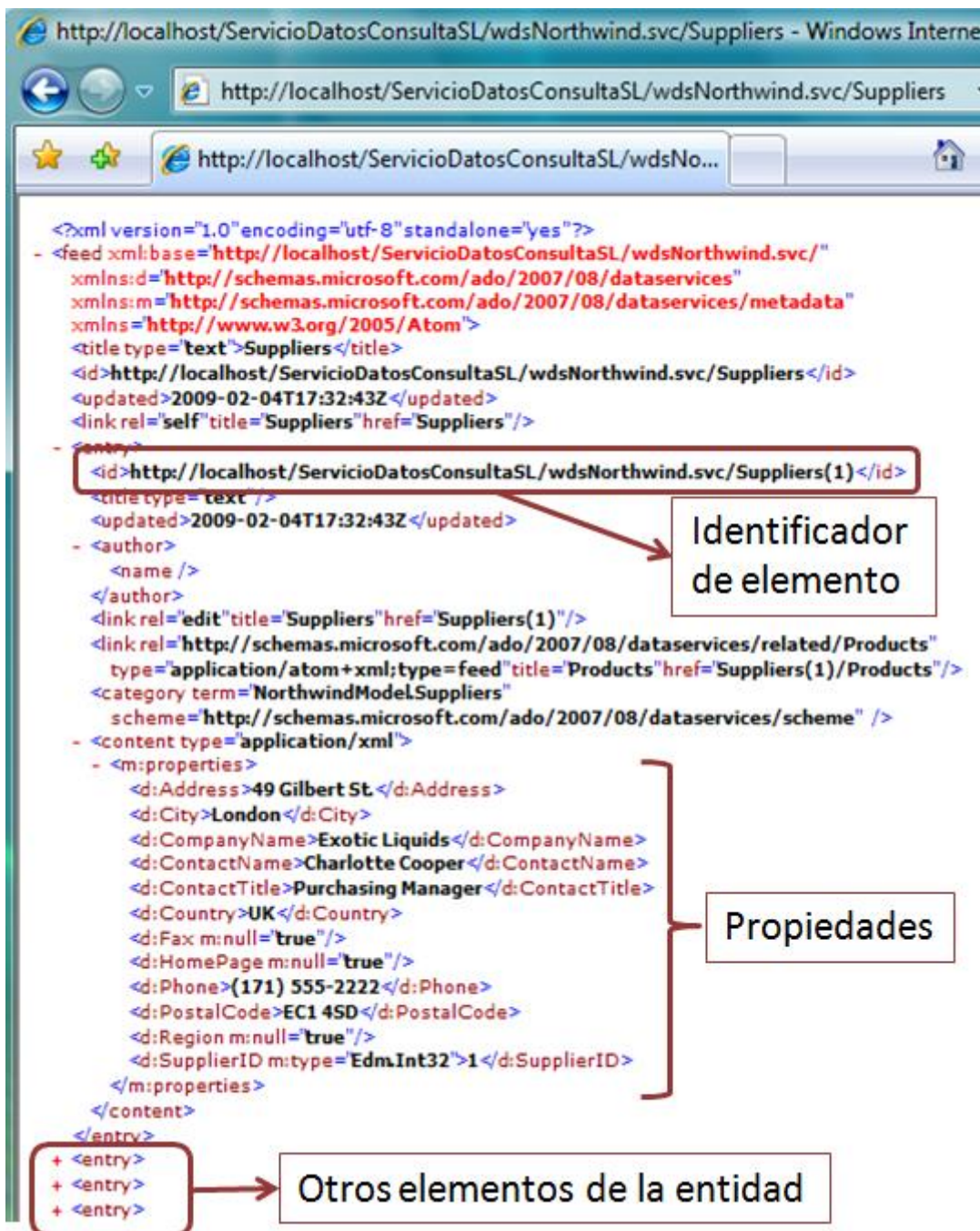


Figura 3. Listado de los elementos contenidos en una entidad.

Antes de proseguir debemos hacer un inciso acerca de la visualización en el navegador del contenido de las entidades, ya que en el caso de utilizar Internet Explorer, es probable que la salida no se muestre en la forma que hemos indicado. Para poder visualizar correctamente el resultado, seleccionaremos en el navegador la opción de menú *Herramientas > Opciones de Internet*; haremos clic en la pestaña Contenido, y dentro del apartado Fuentes, pulsaremos el botón Configuración, desmarcando a continuación la casilla *Activar la vista de lectura de fuentes*.

Supongamos a continuación que solamente queremos recuperar un único registro de la entidad. La forma de conseguirlo consiste en adjuntar al nombre de la entidad, encerrado entre paréntesis, el valor de la clave primaria correspondiente al registro; siendo en este caso el campo SupplierID la clave primaria de la tabla/entidad.

[http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers\(7\)](http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers(7))

Si observamos la entidad Suppliers en el EDM de nuestro proyecto, nos percataremos de que existe una asociación entre dicha entidad y Products, a través de la propiedad SupplierID. Esto significa que a partir de un registro de Suppliers, podemos obtener los registros relacionados de la entidad Products.

El modo de implementar esta característica en la llamada al servicio pasa por utilizar el parámetro expand, pasándole como valor la entidad asociada de la que queremos obtener los registros relacionados.

[http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers\(4\)?\\$expand=Products](http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers(4)?$expand=Products)

Otra variante en la selección de los datos la constituye el parámetro filter, que combinado con alguno de los operadores lógicos eq (equal), ne (not equal), gt (greater than), etc., nos permiten aplicar un filtro basado en una condición, como vemos en el siguiente ejemplo.

[http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers?\\$filter=Country eq 'Germany'](http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers?$filter=Country eq 'Germany')

Y finalizamos esta serie de ejemplos con una petición a nuestro servicio de datos que combina un filtro con los operadores lógicos ge (greater than or equal), and (y lógico), le (less than or equal), y finalmente el parámetro orderby, con el cual ordenamos el resultado en base a una de las propiedades de la entidad.

[http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers?\\$filter=SupplierID ge 10 and SupplierID le 20&\\$orderby=CompanyName](http://localhost/ServicioDatosConsultaSL/wdsNorthwind.svc/Suppliers?$filter=SupplierID ge 10 and SupplierID le 20&$orderby=CompanyName)

Conclusión

Una discusión en profundidad, sobre todas y cada una de las características que podemos emplear en la URL, a la hora de consultar un servicio de datos, es una cuestión que excede el ámbito de este artículo, por lo que recomendamos al lector la consulta de [3], donde encontrará todos estos aspectos tratados con un mayor grado de amplitud. En la segunda parte de esta serie abordaremos el acceso al servicio desde Silverlight, en aquellos aspectos relacionados con las consultas y presentación de resultados en la interfaz de usuario.

BIBLIOGRAFIA

[1] Overview: ADO.NET Data Services

<http://msdn.microsoft.com/en-us/library/cc956153.aspx>

[2] Data Model and Data Service Implementations (ADO.NET Data Services/Silverlight)

[http://msdn.microsoft.com/en-us/library/cc838239\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc838239(VS.95).aspx)

[3] Using Microsoft ADO.NET Data Services

<http://msdn.microsoft.com/en-us/library/cc907912.aspx>