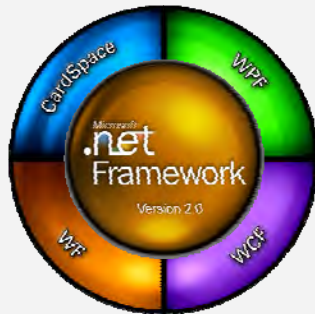


## Visual Studio 2008, LINQ Acceso a datos con LINQ usando ListView

### Antecedentes

Actualmente la versión en producción del Framework .NET es la 3.0 que comprende un núcleo conformado por el Framework 2.0, del cual hemos hablado anteriormente, con una capa de módulos especializados en presentación, comunicación, seguridad y flujos de trabajo. A este nuevo conjunto lo llamamos .Net Framework 3.0.



Este esquema se extiende aún más con la presentación del Framework 3.5 que presenta nuevos módulos sobre el framework 3.0, este viene acompañado de la herramienta de desarrollo Visual Studio 2008 –VS2008-.

Actualmente la versión con la que contamos es la edición Beta 2, la cual siendo una versión de pruebas debería ser cargada en una máquina virtual.

La versión de producción será liberada en Febrero de 2008, según lo publicado por Microsoft.

### Objetivos del Laboratorio

Dar a conocer la nueva tecnología para acceso a datos LINQ y su utilización, así como utilizar uno de los nuevos controles que nos brinda Microsoft dentro de VS2008, el control ListView.

### Funcionamiento de la Aplicación

La aplicación final deberá mostrar un listado de fotos organizadas por álbumes dentro de un control List View y accediendo a datos mediante LINQ.

### Prerrequisitos

Para realizar este laboratorio deberemos tener instalado además del Beta 2 de Visual Studio 2008 o de Visual Web Developer 2008 la versión Express de la base de datos SQL con el que se realizará el Laboratorio.

Para descargar estas distribuciones visitar el sitio:

<http://msdn2.microsoft.com/es-es/express/default.aspx>  
<http://www.microsoft.com/sql/editions/express/default.msp>

**COBUS** 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a  
[jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

Para este laboratorio se utilizará la versión Express de Visual Web Developer 2008.

## Solución Parcial

La solución parcial de este laboratorio presenta un sitio preparado para el inicio del desarrollo, cuenta con las imágenes necesarias para realizar la práctica, así como con la base de datos y archivos de presentación que serán usados para darle forma a la aplicación de acceso a datos con LINQ.

## Trabajo por realizar

La Solución final deberá presentar un listado de imágenes por Álbum, para ello debemos:

1. Crear clases para acceso a datos con LINQ.
2. Revisar y entender la estructura generada por la fuente de datos.
3. Crear la lógica del negocio necesaria para acceder a los datos según los requerimientos.
4. Crear las plantillas de presentación dentro del ListView.
5. Atar la fuente de datos al control de presentación
6. Mejorar las plantillas de presentación dentro del ListView

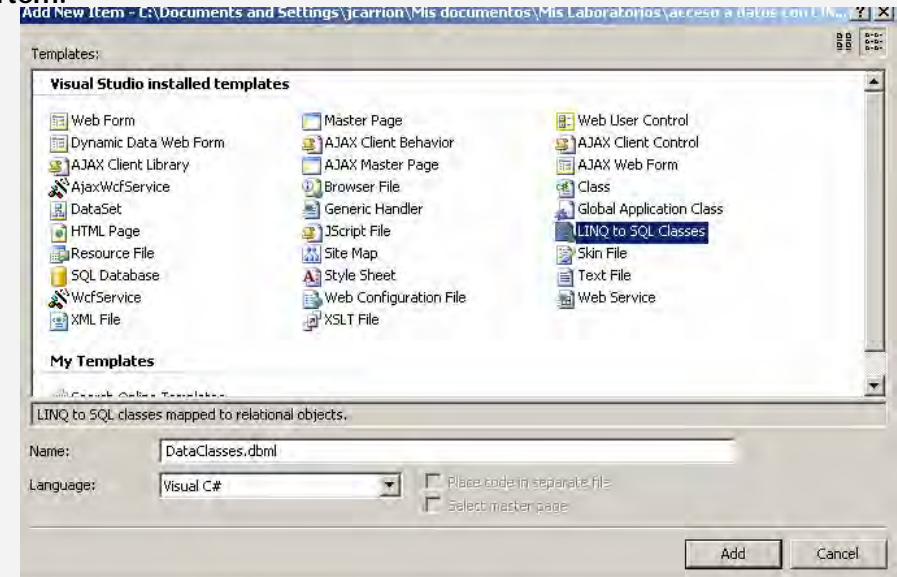
## Guión de desarrollo

### Tarea 1: Crear clases para acceso a datos con LINQ.

Visual Studio 2008 cuenta con nuevos tipos de ítems que nos ayudarán con

esta tarea, para esto debemos agregar al Sitio Web un ítem del tipo **LINQ to SQL Classes**.

Desde el menú contextual sobre el nombre del proyecto en el Explorador de soluciones podemos desplegar la ventana **Add New Item/ Agregar nuevo ítem**.



A este nuevo ítem le nombraremos **AlbumFotos.dbml** y lo agregaremos al proyecto. Los archivos de código fuente se agregarán a la carpeta **App\_Code** y el archivo de diseño dentro de la raíz del proyecto. La interfase de usuario de VS2008 advertirá sobre este particular a través de un mensaje en pantalla

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

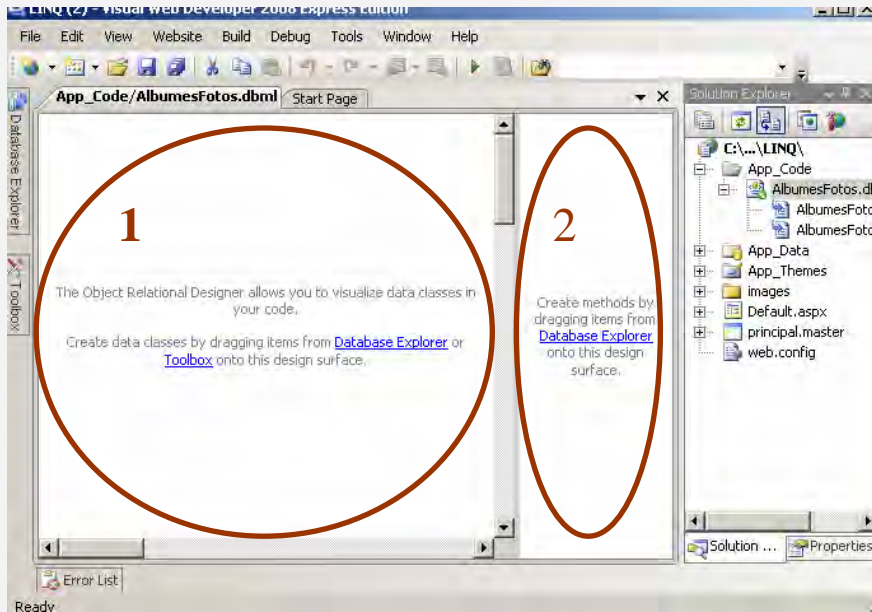
[www.msndcuenca.com](http://www.msndcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

al que debemos responder afirmativamente.

Luego de crear el nuevo ítem se presentará un lienzo de diseño con dos partes, la primera para crear un mapa relacional de objetos y la segunda para la creación de métodos.

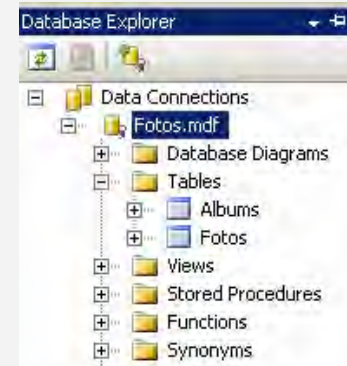


## [::] LINQ

Language-Integrated Query (LINQ) es una innovación en el manejo de datos que promete revolucionar el manejo de datos. Introduce el concepto de Query como lenguaje de primera clase dentro de C# o VB para acceder a todo tipo de datos.

A continuación haremos uso del explorador de datos para agregar las tablas desde nuestra base de datos al diseñador de relacional de objetos.

Para ello utilizaremos la base de datos adjunta al Sitio Web que maneja un grupo de imágenes organizados por Álbum de Fotos. Para ello manejaremos un esquema de cabecera-detalle en el cual usaremos dos tablas: **Albums** que será la cabecera y **Fotos** el detalle.



Para agregar las tablas al Diseñador de Relacional de Objetos arrastraremos y

**COBUS** 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

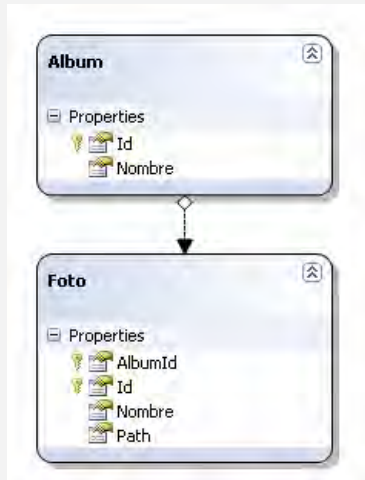
[www.msndcuenca.com](http://www.msndcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

pegaremos cada tabla sobre la primera zona del elemento **LINQ to SQL Classes**.

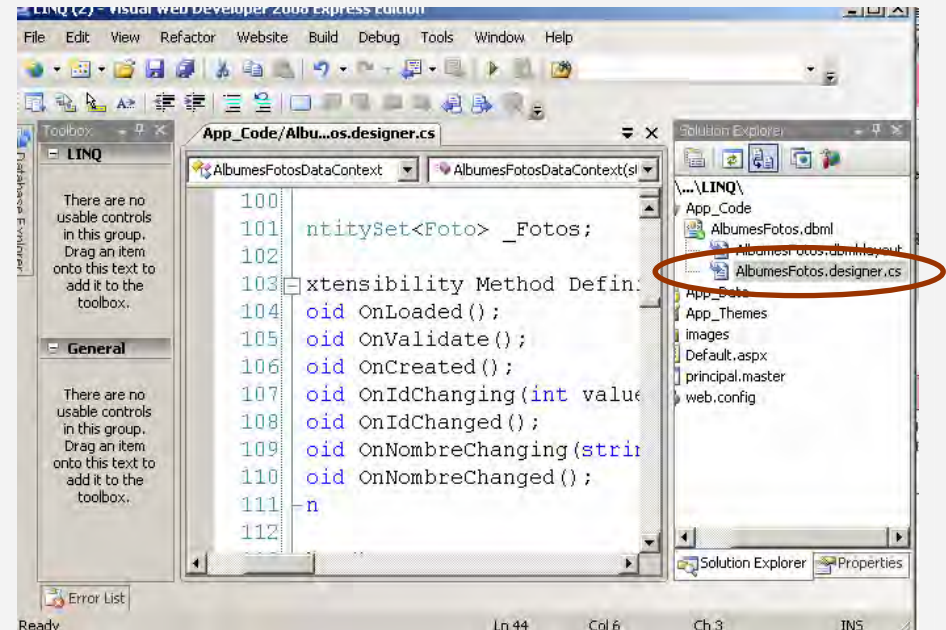
Automáticamente el diseñador reconocerá las relaciones y creará un esquema que represente las clases que se desprenden de las tablas de la base de datos. El esquema presentado a continuación será el resultado de realizar esta operación.



Para asegurarnos que los cambios realizados en el diseñador se vean reflejados en el código fuente procederemos a construir el sitio Web, ya sea utilizando la opción **Build Web Site/Generar Sitio Web** del menú **Build/Generar** o utilizando el menú contextual sobre el nombre del sitio Web con la opción **Build Web Site/Generar Sitio Web**.

**Tarea 2. Revisar y entender la estructura generada por la fuente de datos.**

En este momento analizaremos el código generado por el diseñador, para esto veamos el código fuente del Archivo **AlbumesFotos.designer.cs**



Lo primero que vemos en el código fuente que se generó es la clase AlbumesFotosDataContext.

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de COBUS. COBUS es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdnecuena.com](http://www.msdnecuena.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

```

em.Data.Linq.Mapping.DatabaseAttribute (Name="E386CB4E0051295B6A5A8A755
CON LINQ\ \ SOLUTION_ BEMBA\ LINQ\ APP_ DATA\ FOTOS.MDF ")
c partial class AlbumesFotosDataContext : System.Data.Linq.DataContext
{
private static System.Data.Linq.Mapping.MappingSource mappingSource = n
}

```

Esta clase es el elemento central dentro de la consulta que realizaremos a través de LINQ.

Adicionalmente veremos dos clases adicionales que corresponderán a las tablas de la base de datos que arrastramos para crear el elemento LINQ.

```

85 {
86     return this.GetTable<Foto>();
87 }
88 }
89 }
90 }
91 [Table(Name="dbo.Albums")]
92 public partial class Album : INotifyPropertyChanging, INotifyPropertyCh
206 }
207 [Table(Name="dbo.FOTOS")]
208 public partial class Foto : INotifyPropertyChanging, INotifyPropertyCh
381 }

```

Así tendremos las clases **Album** y **Foto** que internamente manejarán las propiedades necesarias para controlar todos los campos de las tablas en base de datos, así como los métodos que nos permitan trabajar con la base de datos.

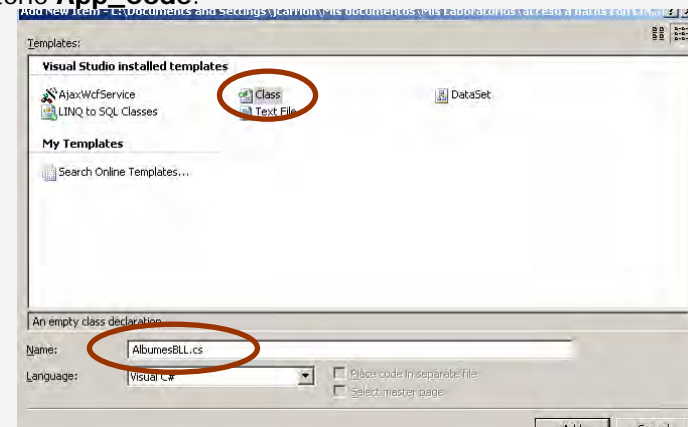
En este punto me permito sugerir que se haga un estudio más exhaustivo del contenido de este archivo para entender completamente la funcionalidad de LINQ.

Por ahora diremos que este modelo sería comparado con un híbrido entre el acceso a través de las fuentes de datos SQLClient y Object. Tomando esto como base de nuestro análisis, además podríamos decir que la mejor forma para consumir datos en una arquitectura por capas sería un modelo LINQ ya que generaría la lógica suficiente para abstraer los datos a clases usando lo mejor de los dos mundos.

### Tarea 3: Crear la lógica del negocio necesaria para acceder a los datos según los requerimientos.

Crearemos una capa intermedia en donde haremos uso del lenguaje de consulta para acceder a los datos de los álbumes y dentro de estos a cada dato de las fotos.

Primero agregaremos una clase a la cual llamaremos **AlbumesBLL.cs** dentro del directorio **App\_Code**.



A la clase **AlbumesBLL** agregaremos un método con el nombre **getFotosXAlbum** en el cual implementaremos la consulta. Este método devolverá un listado de datos que representará a las fotos agrupadas por álbum.

```
public System.Collections.IEnumerable  
getFotosXAlbum()  
{  
    var bd = new  
AlbumesFotosDataContext();  
    var query =  
    from a in bd.Albums  
    select new  
    {  
        a.Nombre,  
        a.Fotos.Count,  
        a.Fotos  
    };  
    return query.ToList();  
}
```

La lógica implementada dentro de este método devolverá una colección indefinida `-IEnumerable-` que será retornada al pasar el conjunto de datos a tipo Lista.

Para el acceso a través de consulta LINQ debemos crear un objeto del contexto de datos para poder acceder, para ello creamos el objeto **bd**. Usando un objeto consulta accederemos a los álbumes mediante la consulta "from a in db.Albums select new {a.Nombre, a.Fotos.Count, a.Fotos}".

La consulta nos dará como resultado los nombres de los álbumes, la cantidad de fotos dentro de cada álbum y las fotos dentro del álbum.

#### [::] System.Collections.IEnumerable

Expone una colección. La cual soporta iteración simple sobre una colección no genérica

#### [::] from

La cláusula from es llamada un generador porque es siempre la primera cláusula en una expresión de consulta, esta especifica la fuente de datos sobre la cual la consulta correrá.

#### [::] select

En una expresión de consulta, la cláusula select especifica el tipo de valores que se producirán cuando la consulta se ejecute. El resultado es basado en la evaluación de todas las cláusulas previas en la expresión del select.

#### Tarea 4: Crear las plantillas de presentación dentro del ListView.

Para consumir la lógica de negocio vamos a agregar un nuevo formulario Web

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de COBUS. COBUS es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

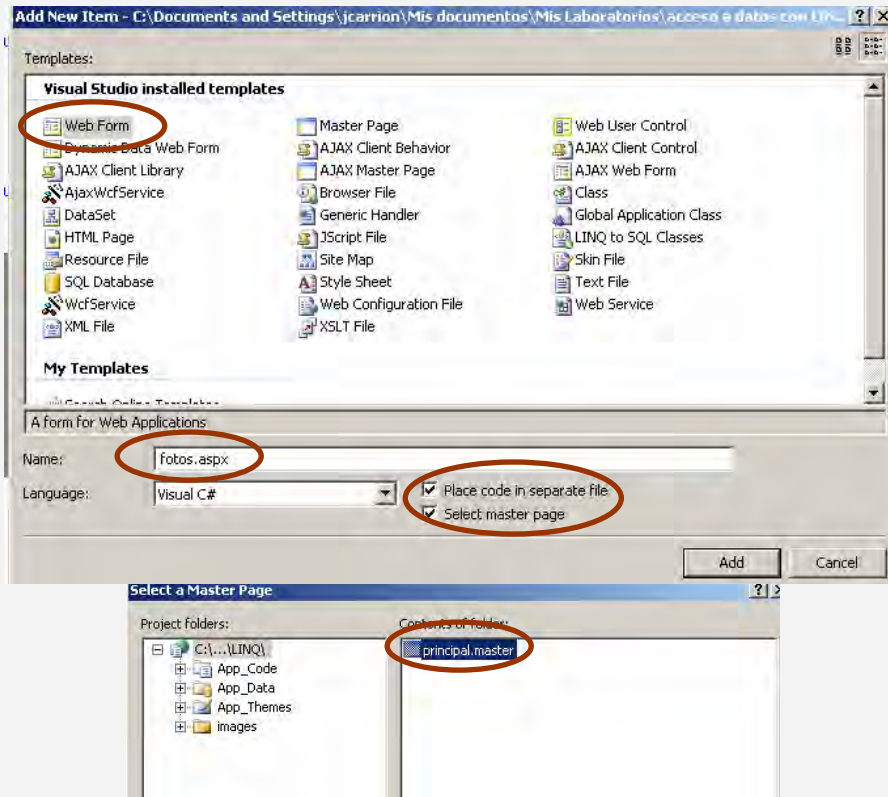
[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

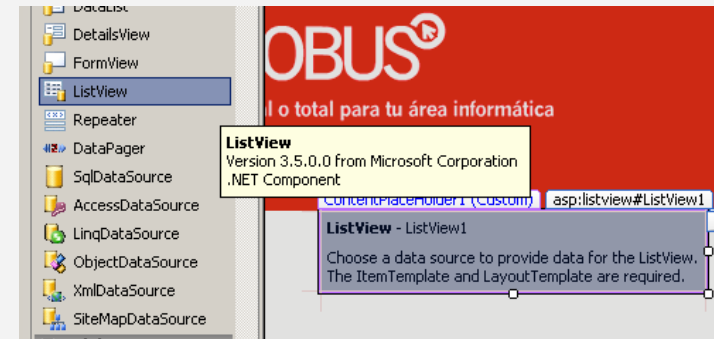
(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarri@cobus.com.ec](mailto:jcarri@cobus.com.ec)

al que llamaremos **fotos.aspx** y haremos uso de la **página principal/masterpage principal.master**.



En este formulario Web agregaremos el control **ListView** desde el **toolBox/Herramientas Data/Datos**.



El Control **ListView1** necesariamente debe tener configuradas las secciones **LayoutTemplate** e **ItemTemplate**, las cuales darán la organización para que los datos se muestren.

Para este laboratorio configuraremos manualmente estas secciones con el objetivo de que quede claro la estructura necesaria para un ListView.

*Usando la nueva vista dividida que trae VS2008 podemos realizar la configuración mientras tenemos el diseño gráfico también presente.*

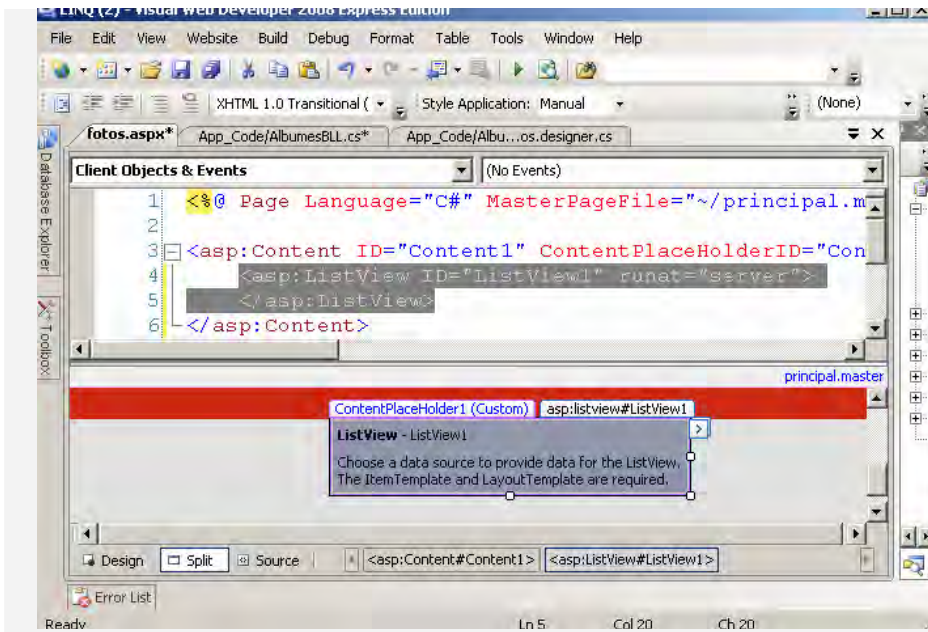
COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdnecuena.com](http://www.msdnecuena.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)



Lo primero que haremos es crear las dos secciones dentro del **ListView1** para ello agregaremos manualmente las etiquetas **<LayoutTemplate>** e **<ItemTemplate>**

```
<asp:ListView ID="ListView1"
runat="server">
  <LayoutTemplate>
  </LayoutTemplate>
  <ItemTemplate>
  </ItemTemplate>
</asp:ListView>
```

La sección **LayoutTemplate** nos permite especificar cual va a ser la estructura que tendrá nuestra lista, en este caso usaremos un listado común HTML a través de la etiqueta **<ul>**. Debido a que este elemento debe ser el primario en la estructura del **ListView** debemos cumplir con dos condiciones, este debe tener como **Id** el valor de **itemContainer** y debe ser un **control de servidor**.

```
<LayoutTemplate>
  <ul id="itemContainer"
  runat="server"></ul>
</LayoutTemplate>
```

Para la sección de presentación de cada ítem usaremos la etiqueta **<li>** y por ahora presentaremos el nombre del Álbum y cuantas fotos existen por álbum.

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msndcuenca.com](http://www.msndcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

Para esto agregaremos dentro de la etiqueta <li> un código de evaluación de expresiones para ayudarnos a mostrar cada uno de los items.

```
<ItemTemplate>
  <li>
    <%# Eval("Nombre") %>,
    <%# Eval("Count") %>
  </li>
</ItemTemplate>
```

#### [::] Data-Binding Expression Syntax

Data-Binding Expression Syntax le permite enlazar valores de propiedades a datos evaluando la expresión desde la fuente de datos. Eval es usada como función de una sola vía para usos de solo lectura.

#### Tarea 5: Atar la fuente de datos al control de presentación.

Para esto agregaremos código al manejador del evento **Load** del formulario Web, en donde ataremos el **datasource** del **ListView** a nuestra lógica de negocio creada en la **tarea 3**.

Entonces, creamos un objeto de tipo **AlbumesBLL** con el cual podamos acceder al método **getFotosXAlbum()**, lo atamos a la propiedad **DataSource** del **Listview1** y luego refrescamos la presentación del

#### ListView1 con DataBind().

```
protected void Page_Load(object sender,
EventArgs e)
{
    AlbumesBLL album = new AlbumesBLL();
    ListView1.DataSource =
    album.getFotosXAlbum();
    ListView1.DataBind();
}
```

A esta altura podemos probar la ejecución de nuestra aplicación y lo haremos mediante **F5** o desde el menú **Debug/Depurar** la opción **Start Debuggin/Iniciar Depuración**.

COBUS 2007 © Todos los derechos reservados

El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**

Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarri@cobus.com.ec](mailto:jcarri@cobus.com.ec)



El resultado de nuestra aplicación será un listado con el **Nombre del Álbum** y la **cantidad de fotos** dentro de cada uno. Recordemos que en la consulta habíamos especificado un contador de fotos.

```
public System.Collections.IEnumerable getFotosXAlbum ()
{
    var bd = new AlbumesFotosDataContext ();
    var query =
    from a in bd.Albums
    select new
    {
        a.Nombre,
        a.Fotos.Count,
        a.Fotos
    };
    return query.ToList ();
}
```

#### Tarea 6: Mejorar las plantillas de presentación dentro del ListView.

Hasta el momento hemos presentado una parte de los datos que fueron recolectados en la consulta, pero no hemos mostrado las fotos de los álbumes, objeto principal de nuestra consulta.

Para mostrar las imágenes de nuestras fotos vamos a agregar un **ListView** anidado dentro de la sección **itemTemplate** del control **ListView** previamente agregado.

Para esto arrastraremos un control **ListView** luego de la evaluación del contador de Fotos. Usaremos un salto de línea para dividir visualmente los campos.

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de COBUS. COBUS es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdnecuena.com](http://www.msdnecuena.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

```

<ItemTemplate>
<li>
    <%# Eval("Nombre") %>,
    <%# Eval("Count") %>
    <br />
    <asp:ListView ID="ListView2"
    runat="server">
    </asp:ListView>
</li>
</ItemTemplate>

```

Como se puede observar, en vista de código, hemos agregado un control **ListView** al que debemos configurar de la misma forma que lo hicimos con el control agregado previamente. Agregaremos las secciones **<LayoutTemplate>** e **<ItemTemplate>**, a la primera sección agregaremos como estructura una lista HTML normal **<ul>** y como estructura de cada elemento un elemento de lista **<li>**.

```

<ItemTemplate>
<li>
    <%# Eval("Nombre") %>,
    <%# Eval("Count") %>
    <br />
    <asp:ListView ID="ListView2"
    runat="server">
        <LayoutTemplate>
            <ul id="itemContainer"
            runat="server"></ul>
        </LayoutTemplate>
        <ItemTemplate>
            <li></li>
        </ItemTemplate>
    </asp:ListView>

```

Para que **ListView2** pueda acceder a los datos debemos atarlo al elemento **Fotos** al que hacemos referencia en la consulta.

COBUS 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a  
[jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)

```

}
public System.Collections.IEnumerable getFotosXAlbur
{
    var bd = new AlbumesFotosDataContext();
    var query =
    from a in bd.Albums
    select new
    {
        a.Nombre,
        a.Fotos.Count,
        a.Fotos
    };
    return query.ToList();
}

```

```

<asp:ListView ID="ListView2"
runat="server" DataSource="<%#
Eval("Fotos") %>">

```

Con esto cada iteración de **Álbum** tendrá su correspondiente fuente de datos a **Fotos**.

Como podemos ver en el código HTML, anteriormente presentado, se ha agregado un elemento **<li>** para el **<itemTemplate>**. Este elemento será usado como repositorio de un *texto* y una *imagen* por lo que agregaremos

más código HTML para realizar esta operación.

```

<ItemTemplate>
    <li><%# Eval("Nombre") %>
    <br />
    " />
</li>
</ItemTemplate>

```

Como vemos hemos evaluado la propiedad **Nombre** de la Foto y su **Path** dentro de un elemento **<img>**.

Para probar la solución final de nuestro laboratorio la ejecutaremos mediante **F5** o desde el menú **Debug/Depurar** la opción **Start Debuggin/Iniciar Depuración**.

El resultado será una página Web que presente un listado de fotos con una ruptura por nombre de Álbum y número de Fotos.

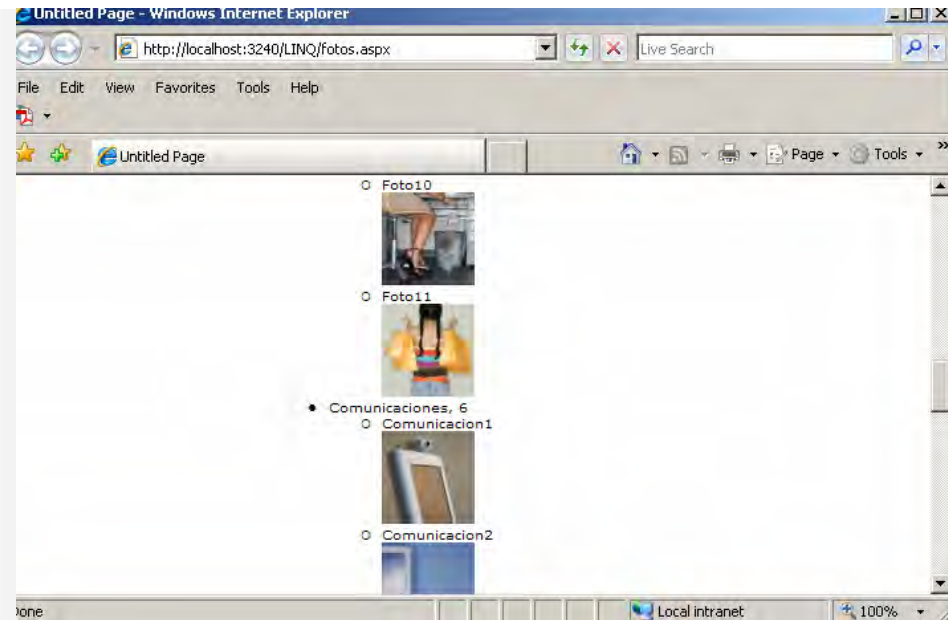
**COBUS** 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**  
Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)



## Conclusiones

Usando **LINQ** podemos acceder a cualquier fuente de datos y lo hacemos a través de una arquitectura de capas usando un lenguaje de consulta.

**ListView** es un control que nos permite presentar datos estructurándolos según nuestras necesidades y nos permite crear presentaciones más apegadas al diseño del sitio.

**COBUS** 2007 © Todos los derechos reservados  
El uso y distribución de este material es únicamente derecho y responsabilidad de **COBUS**. **COBUS** es parte de la comunidad de desarrolladores profesionales de Microsoft MSDNCuenca.

[www.cobus.com.ec](http://www.cobus.com.ec)

[www.msdcuenca.com](http://www.msdcuenca.com)

(593) 094049167

**Ing. Jorge Carrión Quezada**

Ingeniero de Sistemas  
Microsoft Speaker  
Desarrollador 4 estrellas  
Coordinador Académico  
Para comentarios, aportes o sugerencias por favor dirigirse a [jcarrion@cobus.com.ec](mailto:jcarrion@cobus.com.ec)