28/10/2018

Generar reportes con c#

Guía paso a paso

UTN FACULTAD REGIONAL CORDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Programación de Aplicaciones Visuales I

Prof. Esp. Ing. Martín Polliotto

Generar reportes con c#

Guía paso a paso

El siguiente documento tiene por objetivo brindar los contenidos teórico-prácticos necesarios para generar reportes utilizando C# y los componentes estándar de la plataforma .NET.

INTRODUCCION

En el ámbito de la informática, los reportes son salidas un sistema que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.
El reporte, de esta forma, confiere una mayor utilidad a los datos. No es lo mismo trabajar con una planilla de cálculos con 10.000 campos que con un dibujo en forma de torta que presenta dichos campos de manera gráfica. De la misma forma, gracias a los reportes cualquier persona puede proceder a realizar un resumen de datos o a clasificar estos en grupos determinados. Así, teniendo en cuenta los datos que abordan y la extensión que tienen, estos reportes pueden clasificarse en:

- **Listados**: generalmente utilizados para dar soporte a procesos operativos utilizados por los niveles más bajos dentro de la estructura organizacional. Son ejemplos de este tipo de reportes: los listados de pedidos pendientes a la fecha, una hoja de ruta, un listado de control de existencias.

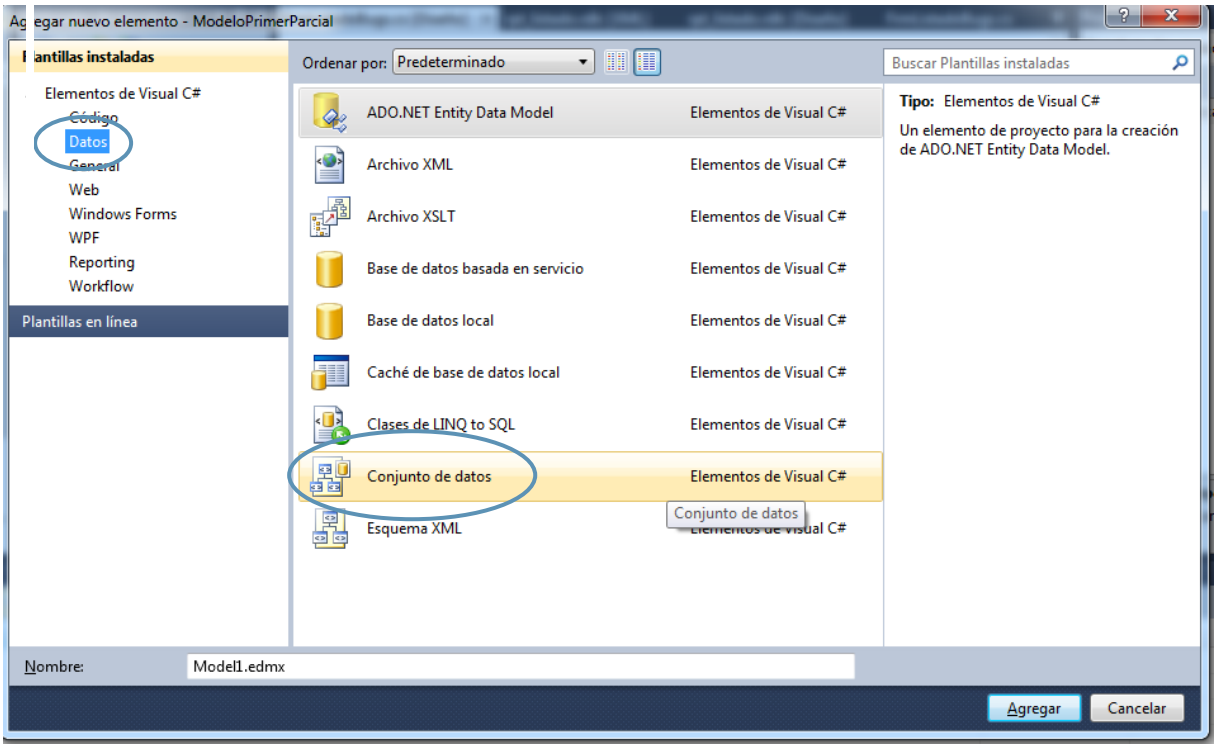
- **Reportes:** utilizados para dar soporte a las necesidades de información de mandos medios permiten, como se indicó anteriormente, clasificar los datos en grupos (denominados en los lenguajes estructurados como cortes de control) y consolidar información de cantidades y/o importes. Éstos incluyen generalmente filtros por fechas, áreas, estados o tipos. Son ejemplos de reportes: reportes de ventas por sucursal y zona, reportes de vuelos por destino o reportes de mantenimiento por tipo de vehículo por mes.

- **Estadísticos:** son salidas que se utilizan fundamentalmente para dar soporte a las decisiones estratégicas mediante la consolidación de grandes volúmenes de datos transaccionales, utilizando gráficos (de barras o de torta) para representar los datos consultados. Por lo general se utilizan para analizar evolución de indicadores de gestión, comparativos de comportamientos de variables de interés entre periodos (anuales) o tendencias de variables críticas de negocio. Son ejemplos de estadísticos: Reporte interanual de ventas o un reporte anual de ganancias netas por sucursal.

PASOS PARA GENERAR UN REPORTE

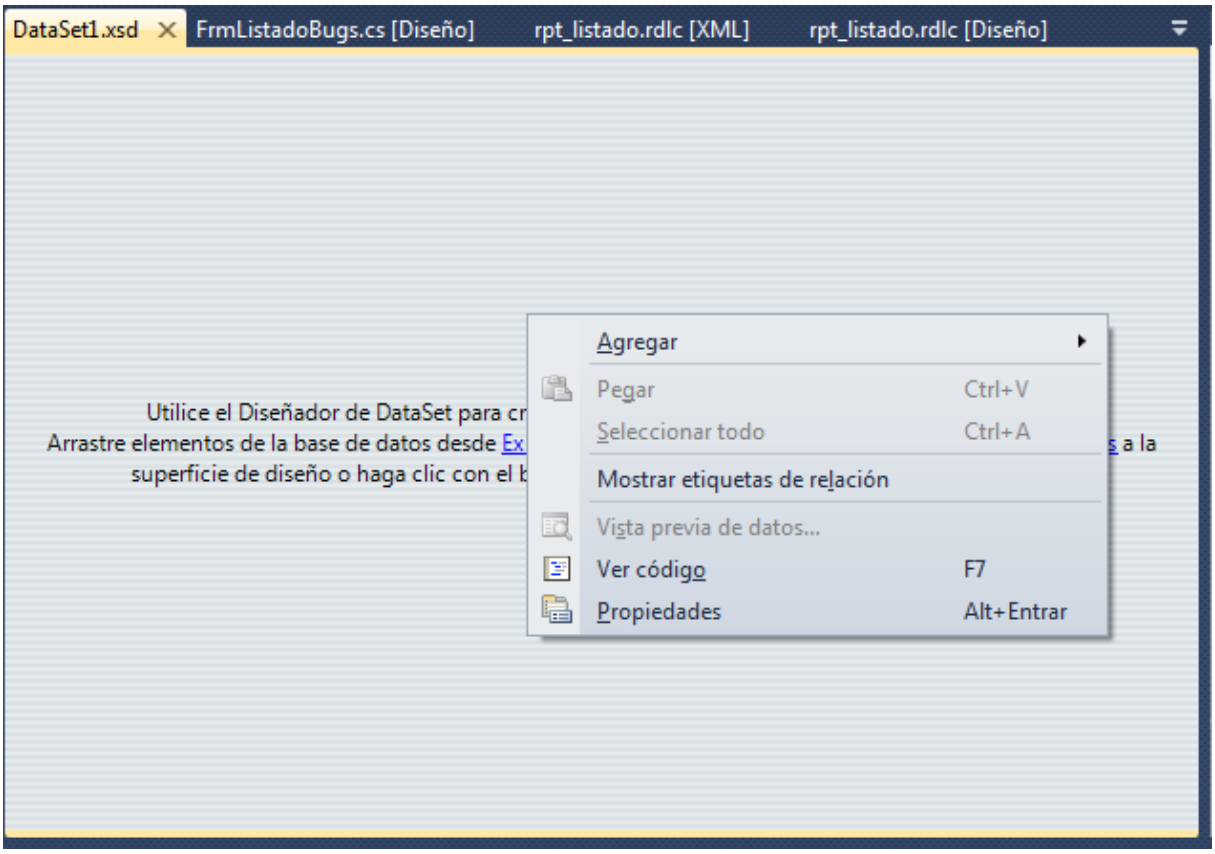
En general para generar nuestros reportes podemos convenir los siguientes pasos:

I. Como primer paso crear un objeto DATASET [1] que utilizaremos como fuente de datos para diseñar el reporte y posteriormente para poblar con datos. Para crear un DATASET hacer click derecho sobre el proyecto y seleccionar *agregar>>nuevo elemento…*



[1] Un DATASET es una representación de datos residente en memoria que proporciona una modelo de programación relacional coherente independientemente del origen de datos que contiene

Una vez creado el DATASET haciendo click derecho sobre la pantalla de visualización del conjunto de datos nuevamente seleccionamos *agregar*, tal como se indica en la siguiente figura:



En este punto tenemos básicamente dos opciones:

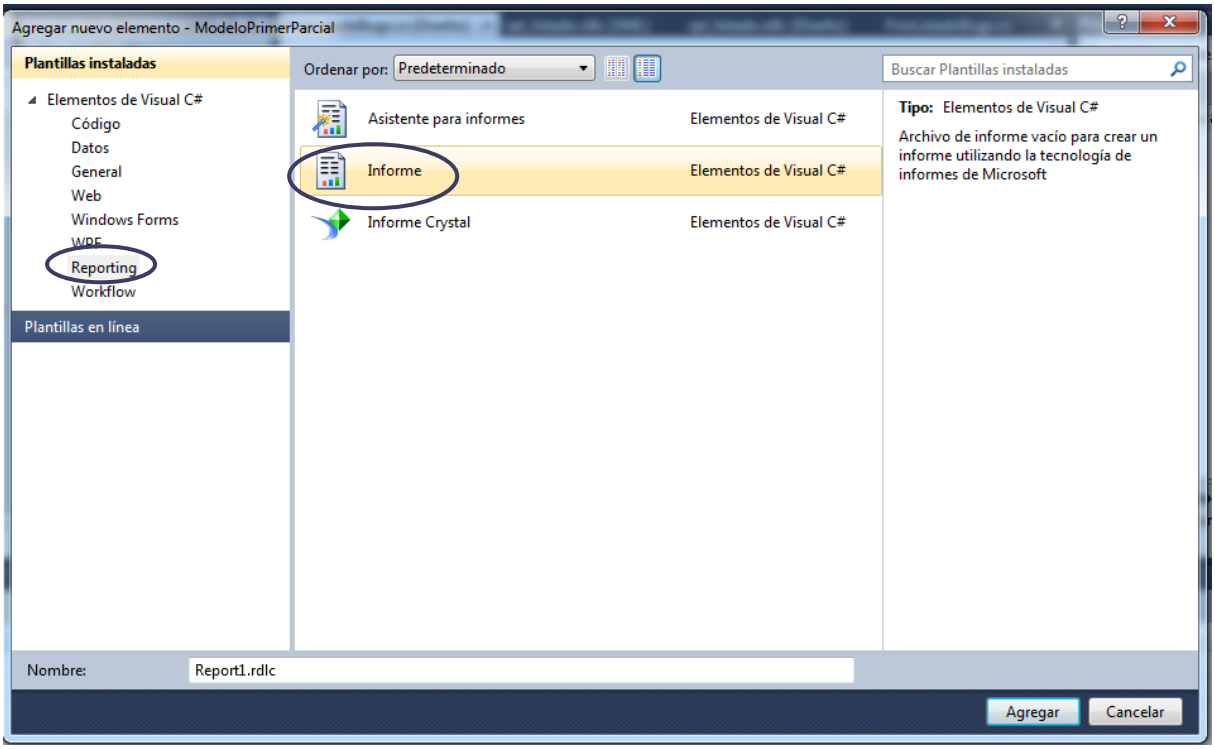
- **TableAdapter**. Con esta opción podemos generar nuestra tabla y mediante el generador de consultas componerla a partir de dos o más tablas de nuestra base de datos. Esta opción será utilizada en la sección siguiente: LISTADOS UTILIZANDO ASISTENTE PARA INFORMES.

- **Tabla de datos.** Con esta opción creamos un DATATABLE [2] propio, al que necesitaremos definir columnas y tipos de datos (como si fuese una nueva tabla). A diferencia de lo anterior, la consulta SELECT con la que llenaremos este objeto será definida por fuera del DATASET y dinámicamente, en tiempo de ejecución, será

[2] Un DATATABLE es un objeto que nos permite representar una determinada tabla en memoria, de modo que podamos interactuar con ella.

asignada por código antes de visualizar el reporte. Esta opción será utilizada en la sección REPORTES.

II. El segundo paso será diseñar nuestro reporte. Para ello hacemos click derecho sobre el proyecto y seleccionar *agregar>>nuevo elemento…*

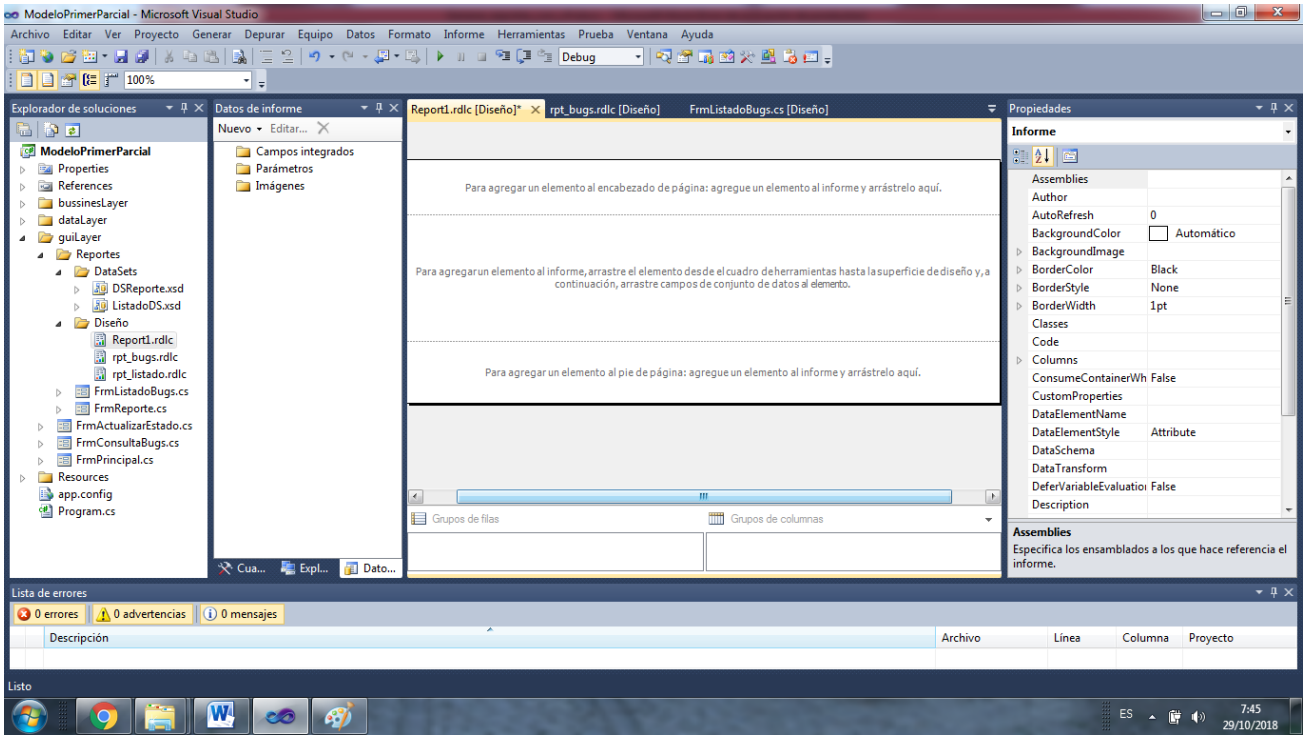


Notar que en este caso, la extensión del objeto Informe es .rdlc.

Se nos crea un área de trabajo donde podremos arrastrar y soltar controles: como tablas, cuadros de textos, líneas o imágenes, de la misma forma que si diseñáramos un formulario *WinForm*.

Se deberá habilitar la opción *Ver>>Cuadro de herramientas,* en caso de que no se muestre por defecto, para disponer de la paleta de controles.

La siguiente figura muestra el área de diseño de nuestro reporte:



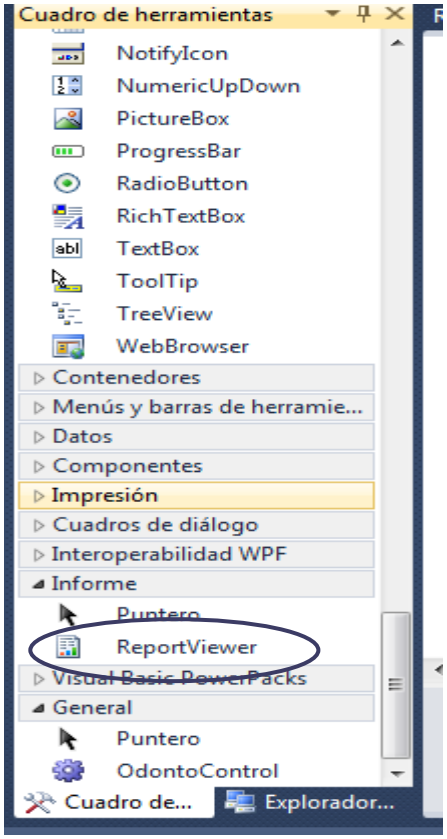
Todo reporte debería tener al menos las siguientes secciones:

1. **Encabezado**: esta sección se utiliza para mostrar datos que no se repiten, tales como el título del reporte, la fecha de generación o el logo de la empresa. Preferentemente los datos de esta sección solo deberían mostrarse en la primera página del reporte. Para indicar esto último hacemos click derecho sobre el encabezado y mediante la opción *propiedades del encabezado*... podemos fijar este comportamiento.
2. **Cuerpo**: permite incluir los datos que se repiten, es decir, debería contener los datos obtenidos del SELECT a la base de datos. Acá se incluyen generalmente las tablas de resultados. Cuando arrastramos una tabla automáticamente el IDE nos permite seleccionar un conjunto de datos para elegir nuestra tabla
3. **Pié de página**: similar a la sección de encabezado pero al pié del informe. Generalmente incluimos el número de página o el nombre del reporte. Esta sección debería repetirse en cada página del documento.

Algo interesante para resaltar es la posibilidad de insertar en un control (generalmente un cuadro de texto o en una columna de tabla) la opción *Expresión*. Mediante esta opción es posible acceder a un abanico de fórmulas y funciones integradas con las que podemos hacer un diseño robusto, sumamente

flexible y fácil de mantener. Queda a cargo del alumno investigar y utilizar todo el potencial de esta herramienta. Si alguna vez utilizó el asistente de fórmulas de una planilla de cálculos EXCEL, entonces se sentirá familiarizado con esta opción.

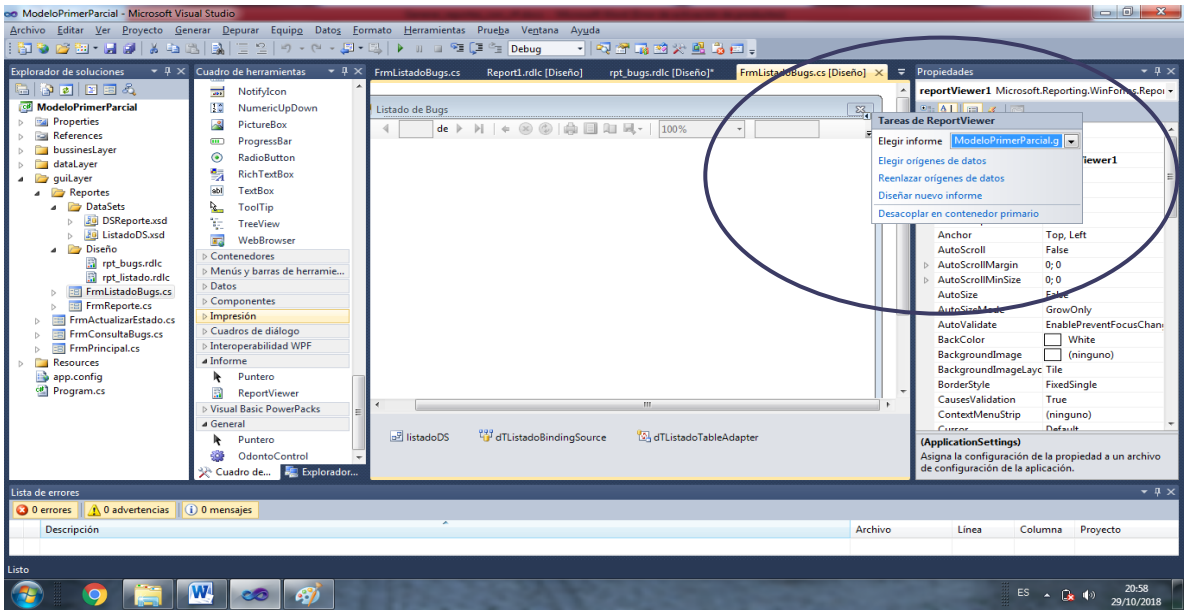
III. Por último necesitamos visualizar nuestro diseño (objeto .rdlc) sobre un control específico para mostrar reportes. Vamos a crear un formulario (WinForm) y agregar un control de reporte desde el cuadro de herramienas: **ReportViewer.**



Cuando se arrastra este componente automáticamente el Visual Studio automáticamente agrega en el evento Load() del formulario:

```
this.reportViewer1.RefreshReport();
```

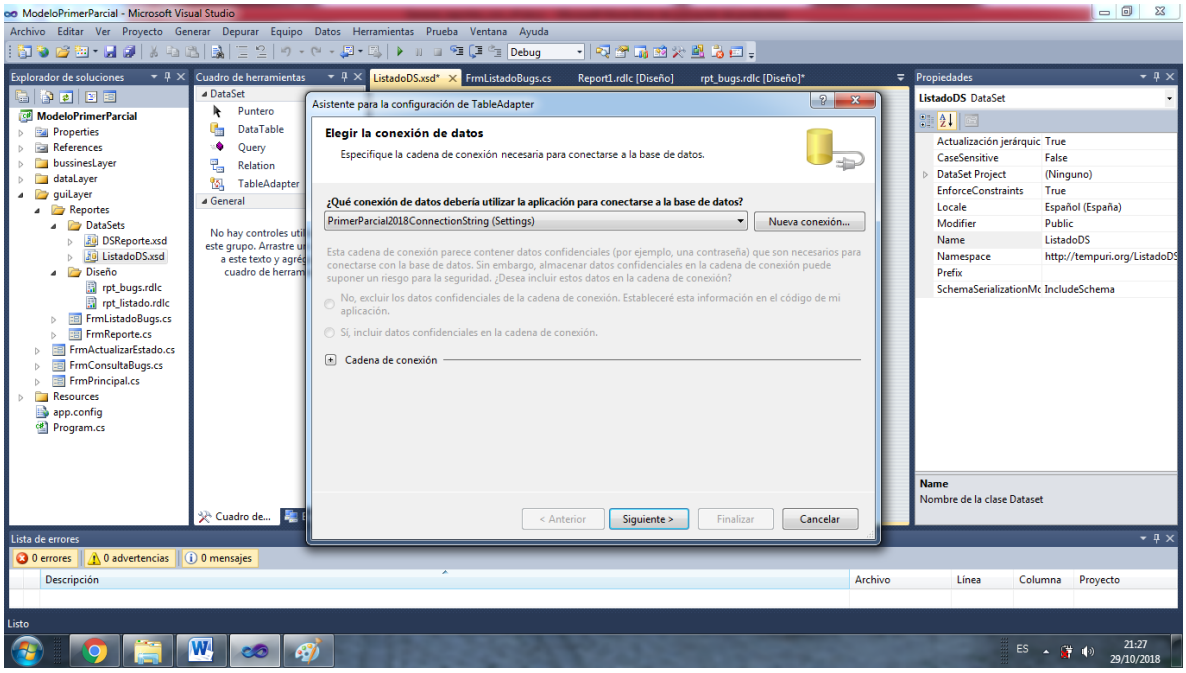
Está línea permite ejecutar el reporte y visualizar el diseño realizado en nuestro objeto .rdlc. Para que esto último funcione correctamente debemos pararnos sobre el objeto reportViewer y enlazarlo con nuestro diseño. Tal como se muestra en la siguiente figura:



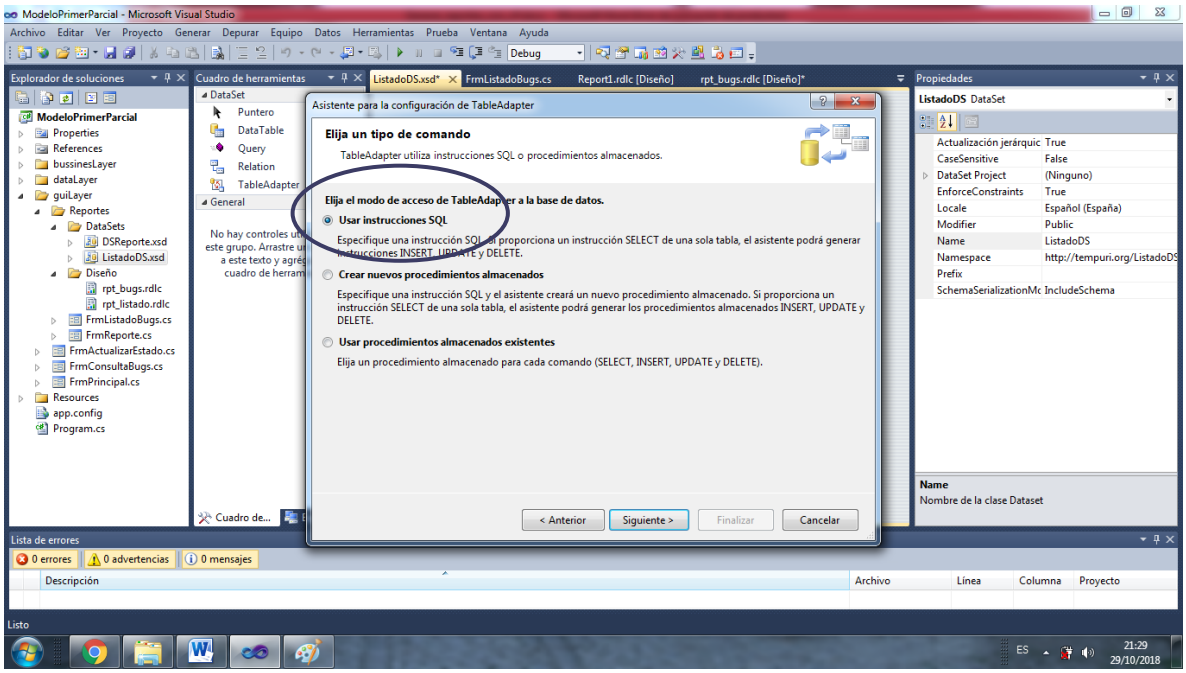
LISTADOS UTILIZANDO UN TABLEADAPTER

Para crear un listado simples vamos a utilizar los siguientes pasos:

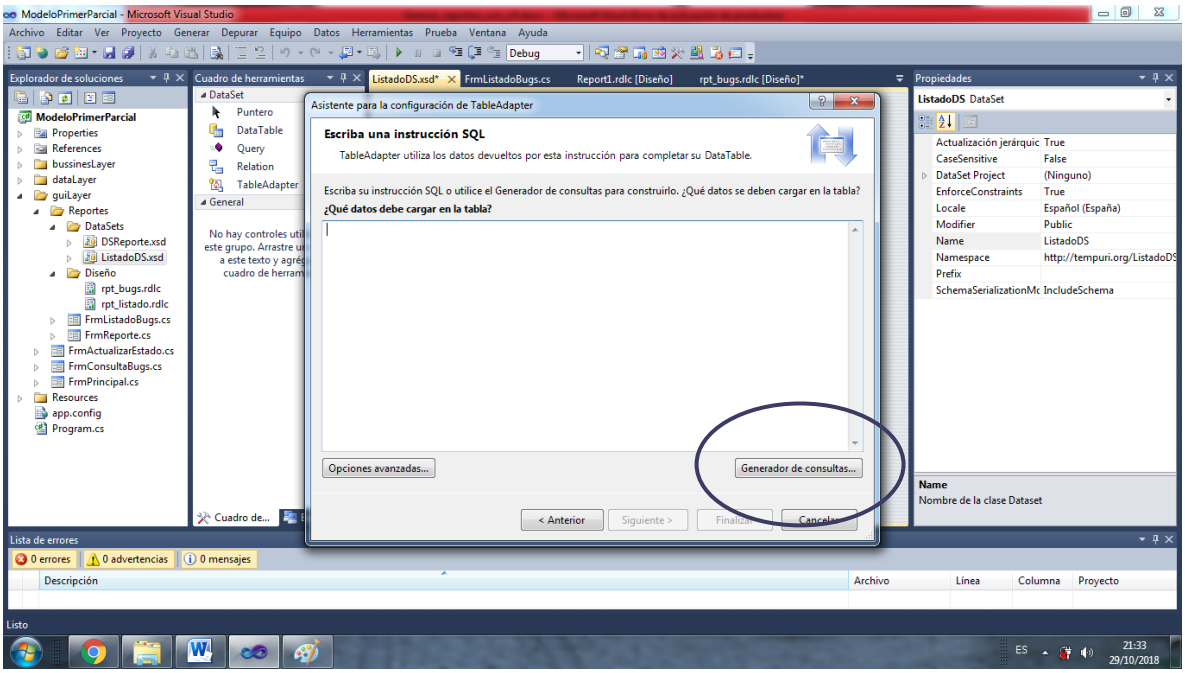
1) Primero vamos a crear un DATASET llamado *DSListado.xsd* y luego vamos a hacer click derecho y seleccionar la opción *agregar>>tableAdapter*.



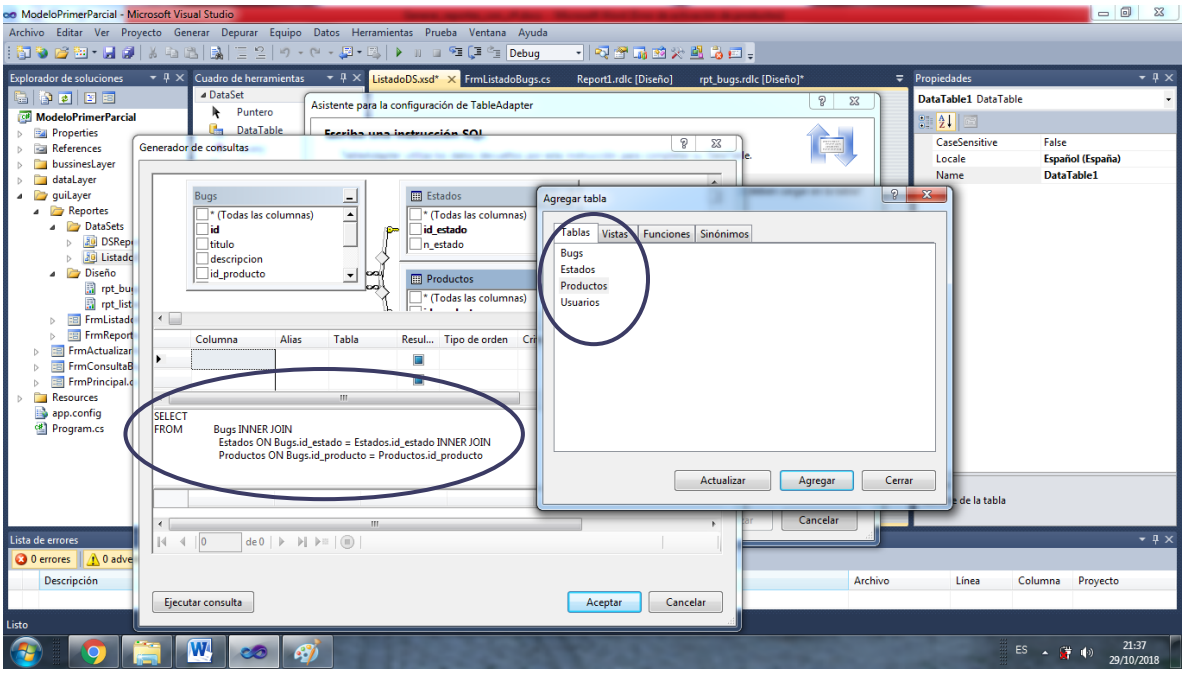
- Lo primero que nos pide es una cadena de conexión a nuestra base. Si no la hemos generado aún podemos seleccionar la opción *Nueva conexión…* y seguir los pasos del asistente.
- Definida la conexión seleccionamos *Siguiente:*



Seleccionamos la primera opción: *Usar instrucciones SQL*. Notar que también podemos utilizar un procedimiento almacenado existente en nuestra base de datos o crear uno desde el asistente. En la siguiente pantalla seleccionamos la opción *Generador de consultas*...

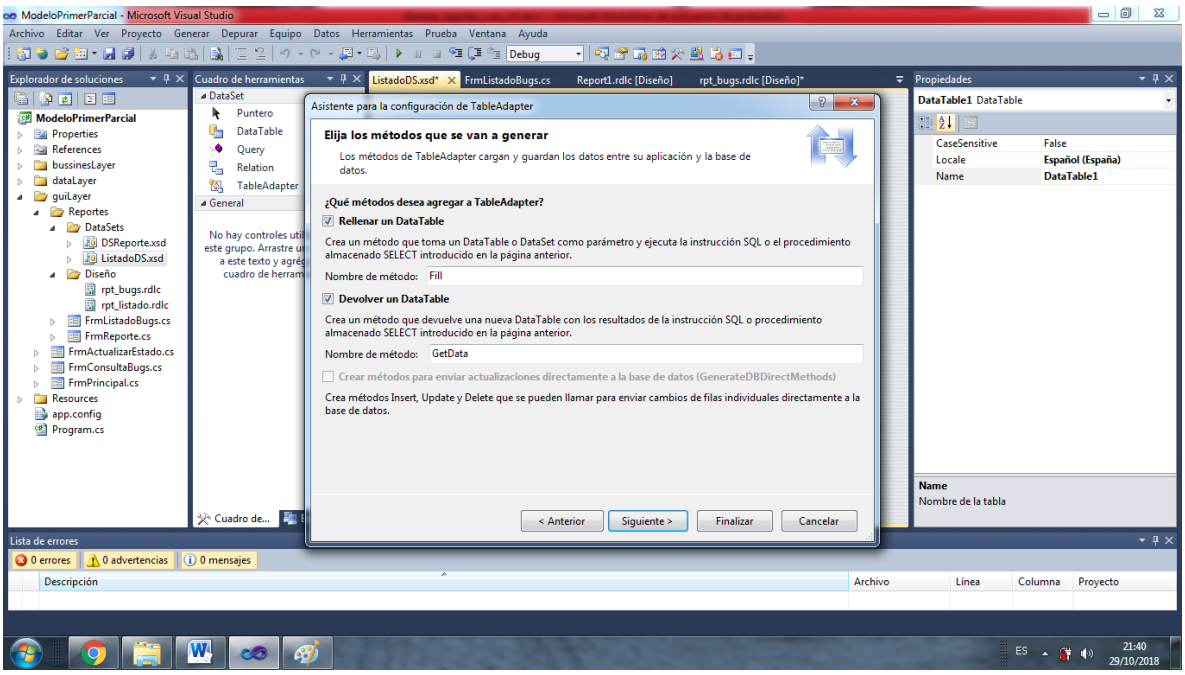


Esta opción nos permite crear una tabla en memoria que surja de la conjunción de varias tablas de nuestra base. Al seleccionar las tablas el asistente automáticamente arma la sentencia SQL con los INNER JOIN respectivo respetando las claves primarias y foráneas definidas en nuestro modelo de datos.



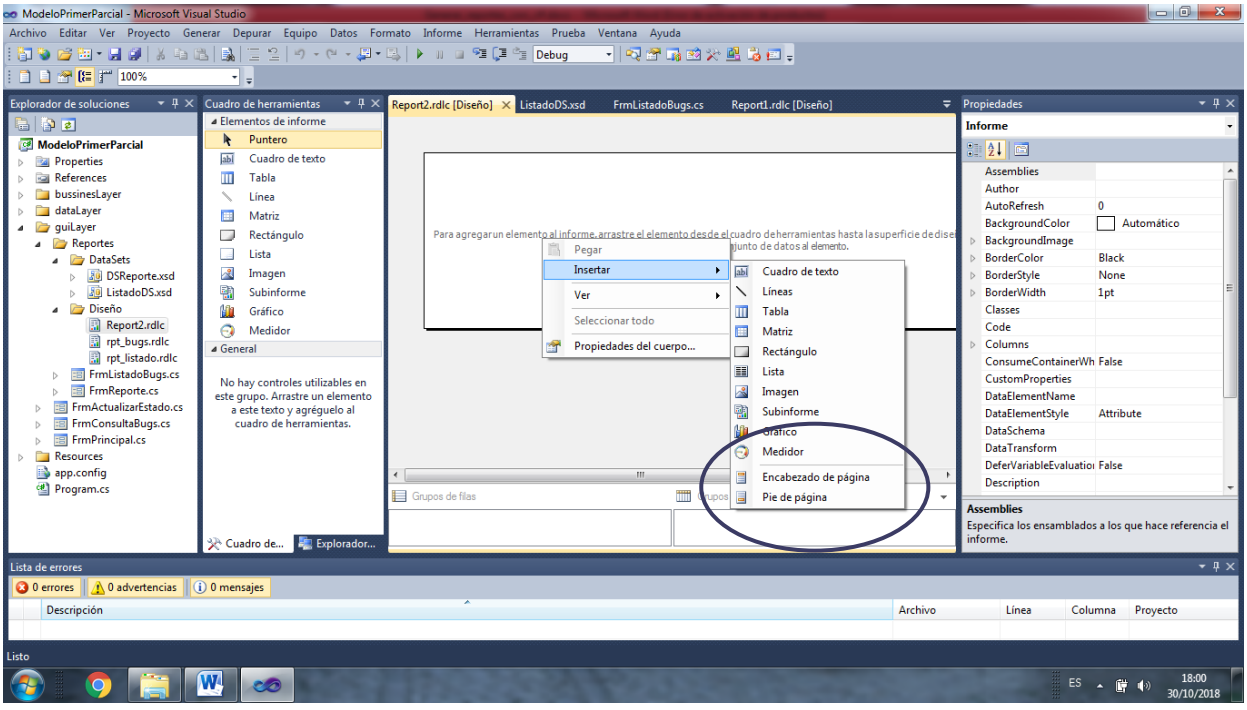
Luego seleccionamos las columnas de las tablas a incluir en el SELECT y con la opción *Ejecutar consulta* podemos validar los resultados obtenidos.

La siguiente pantalla permite configurar los métodos que tendrá el objeto TableAdapter para poder llenar el DATATABLE con la consulta generada.

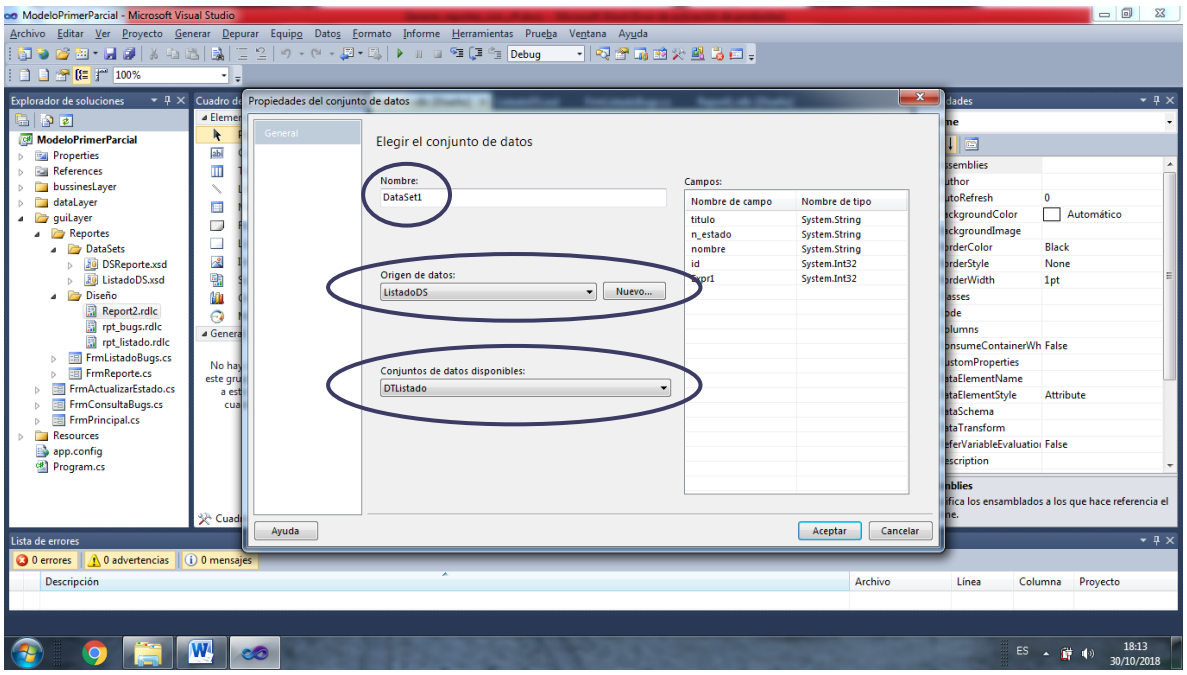


Al finalizar el asistente tendremos ya configurado nuestro DATASET para usarlo en el siguiente paso.

2) El siguiente paso es definir nuestro archivo .rdlc con el diseño del reporte. Para ellos seleccionamos *agregar>>nuevo elemento*, y en la categoría Reporting seleccionamos **Informe.** Luego hacemos click derecho opción *insertar* y seleccionamos **Encabezado y Pié de página.**



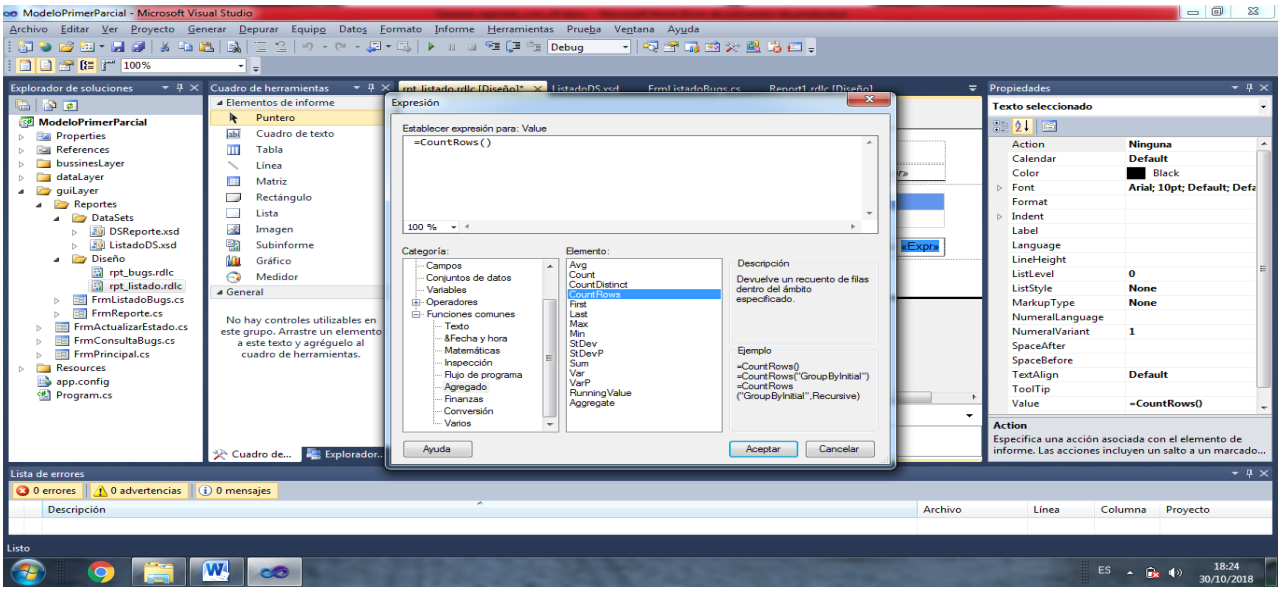
Luego de agregar las secciones mínimas del reporte, arrastramos y soltamos desde el cuadro de herramientas una tabla a la sección central.



Esta pantalla nos permite asociar el DATASET creado en el paso anterior con nuestro diseño. Como se observa podemos elegir ListadoDS y seleccionar el objeto Table generado por el TableAdapter (DTListado).

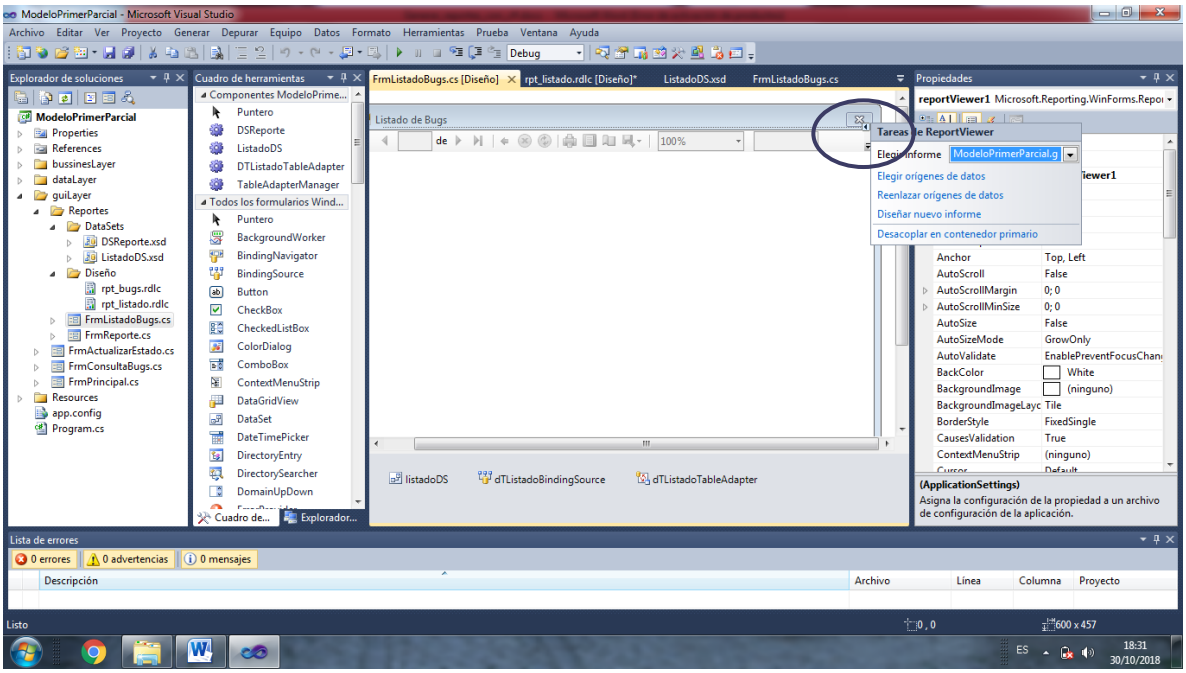
Importante: El nombre que se le asigne a este conjunto de datos será el que luego podrá referenciarse por código en tiempo de ejecución.

Con esto ya podemos armar el detalle de los datos del reporte en función de las columnas del datatable. Como se mencionó anteriormente se sugiere investigar sobre las *opciones de Expresión* que vienen con el VS para agregar elementos al reporte. Si por ejemplo necesitáramos el total de filas recuperadas podemos arrastrar un Cuadro de texto y sobre el componente hacemos click derecho opción *Expresión*



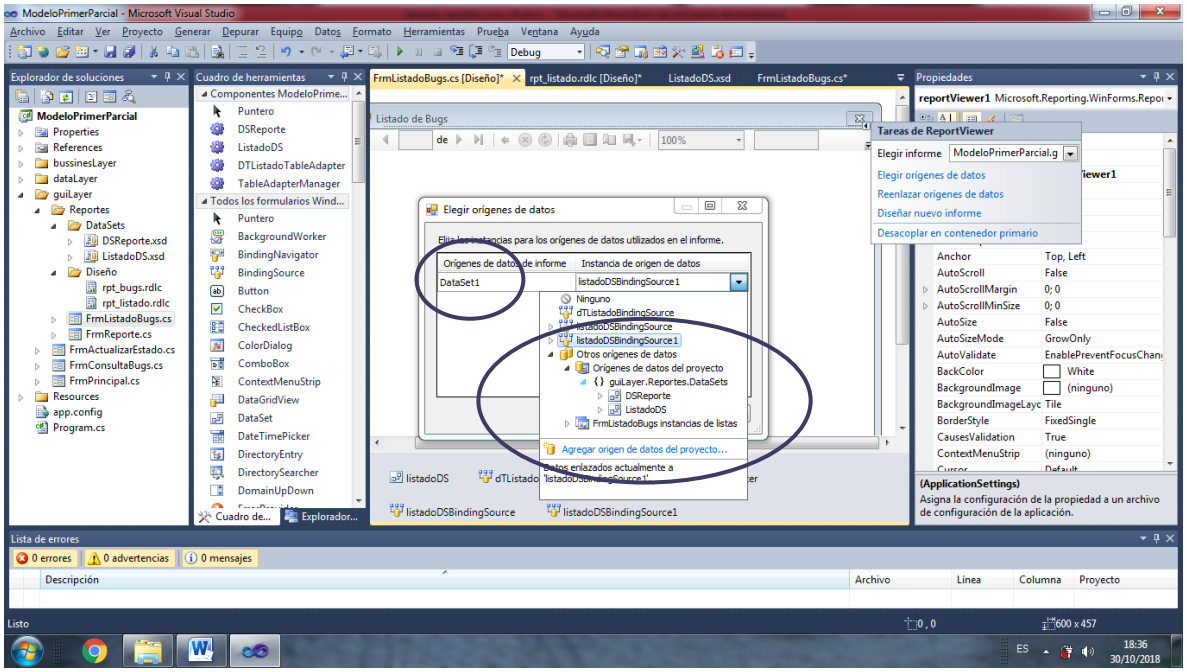
En la categoría *Funciones comunes* seleccionamos CountRows() y podemos generar una expresión que cuente la cantidad de filas del reporte mediante esta función.

3) Por último vamos a crear un formulario donde incluiremos un control **ReportViewer.**



Luego hacemos click en el botón indicado en la imagen y seleccionamos el objeto .rdlc que será asociado con este visualizador de reportes.

Para que nuestro reporte funcione correctamente nos falta solo un último paso: seleccionar el origen de datos para rellenar el dataset del reporte. Seleccionamos opción *Elegir origen de datos* y buscamos el objeto ListadoDS.



Notar que se nos está asociando el DataSet1 (que definimos al arrastrar la tabla al cuerpo del reporte) con el DATASET definido como origen de datos en el paso 2). En el evento Load() del formulario se agrega automáticamente la siguiente línea de código:

```
this.dTListadoTableAdapter.Fill(this.listadoDS.DTListado);
```

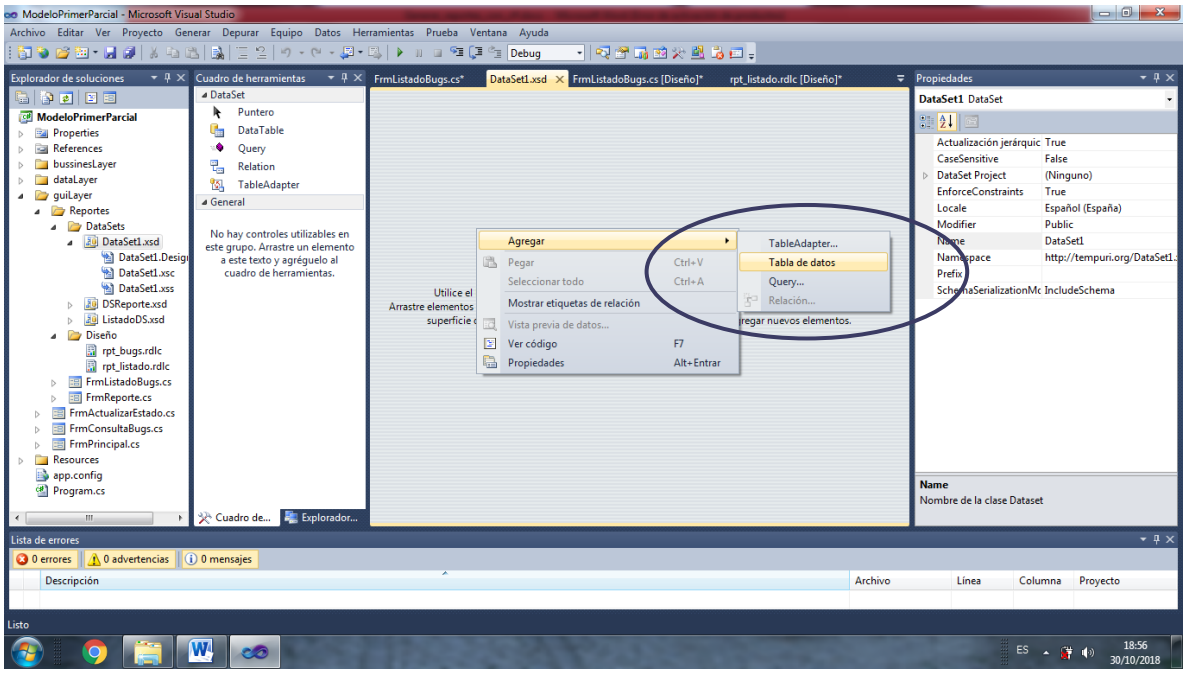
Esta línea permite poblar el dataset del reporte con el datatable creado por el tableadapter del paso 1).

Es importante resaltar que no hemos escrito ni una sola línea de código y el reporte queda funcionando. En la sección siguiente veremos cómo hacer esto mismo escribiendo código y asignando los objetos en tiempo de ejecución.

REPORTES UTILIZANDO UNA TABLA PROPIA

Para crear un reporte personalizado con uno o más grupos o cortes de control vamos a utilizar los siguientes pasos:

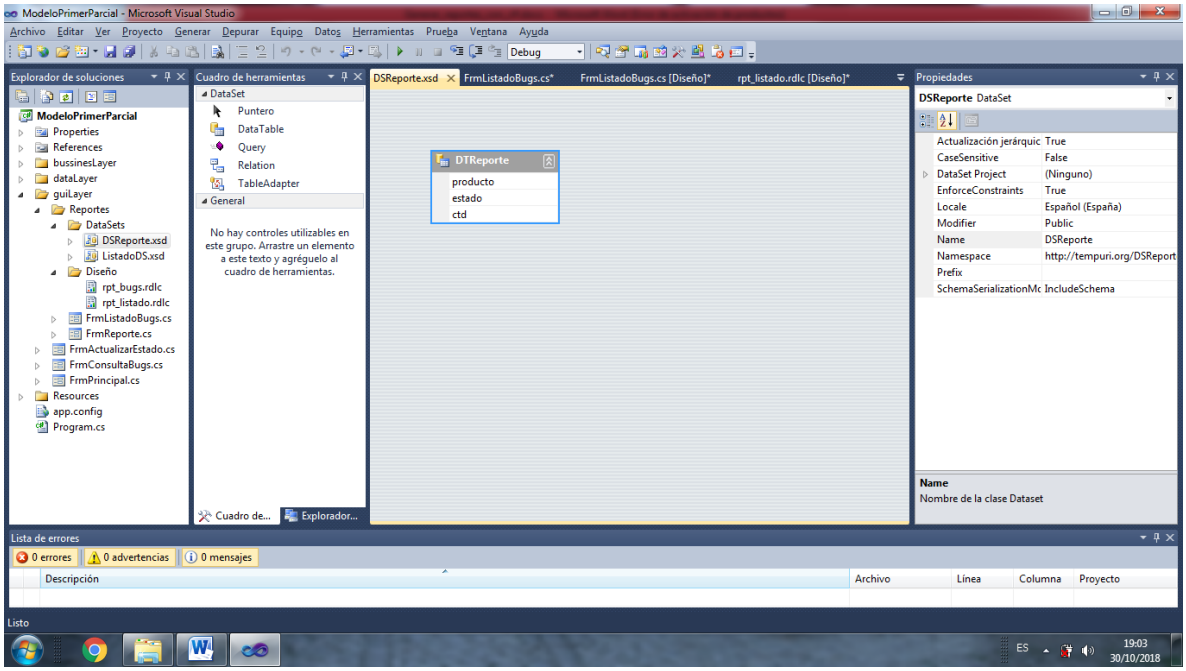
1) Primero vamos a crear un DATASET llamado *DSReporte.xsd* y luego vamos a hacer click derecho y seleccionar la opción *agregar>>Tabla de datos*.



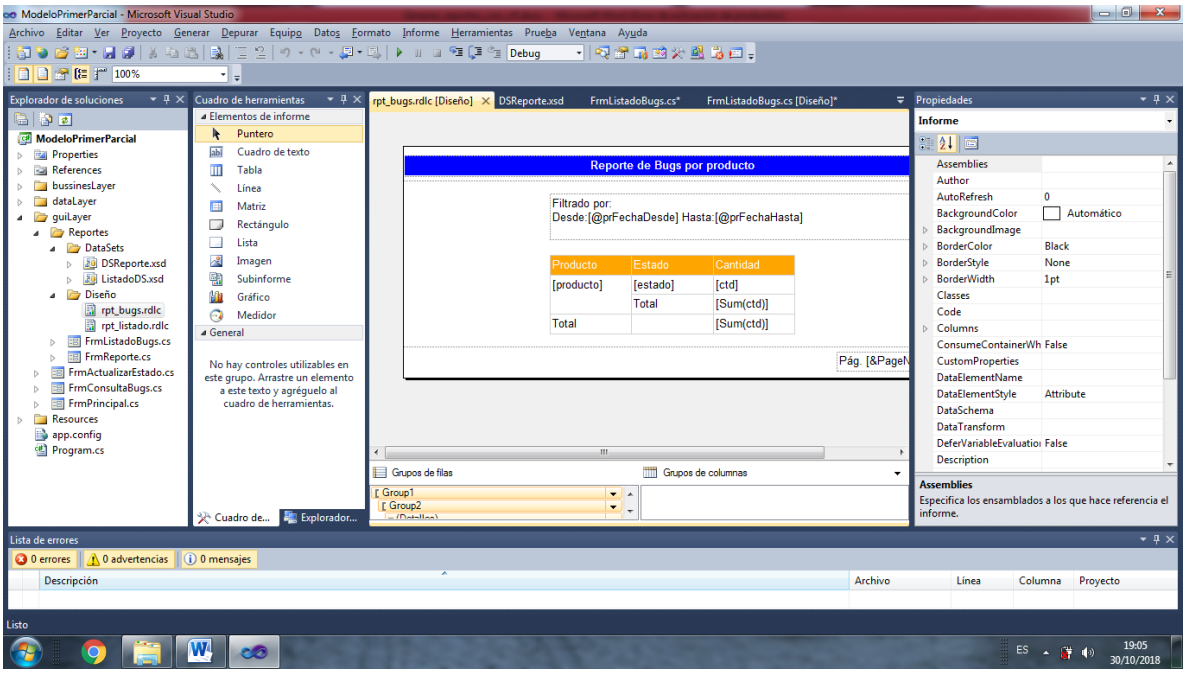
En este caso la estrategia será crear una tabla en memoria que no existe en nuestro modelo de datos pero que va a contener tantas columnas como campos hayamos definido en una consulta SQL diseñada especialmente para crear el reporte y que podremos solicitar a la capa de datos. La consulta debería contar por Producto y por Estado la cantidad de bugs presentes entre dos fechas parametrizadas, tal como se muestra en la sentencia:

```
string strSql = "SELECT nombre as producto, n_estado as estado, COUNT(*) as ctd" +
                " FROM Bugs t1, Productos t2, Estados t3" +
                " WHERE t1.id_producto = t2.id_producto" +
                " AND t1.id_estado = t3.id_estado" +
                " AND t1.fecha_alta between @FecDesde AND @FecHasta"+
                " GROUP BY nombre, n_estado";
```

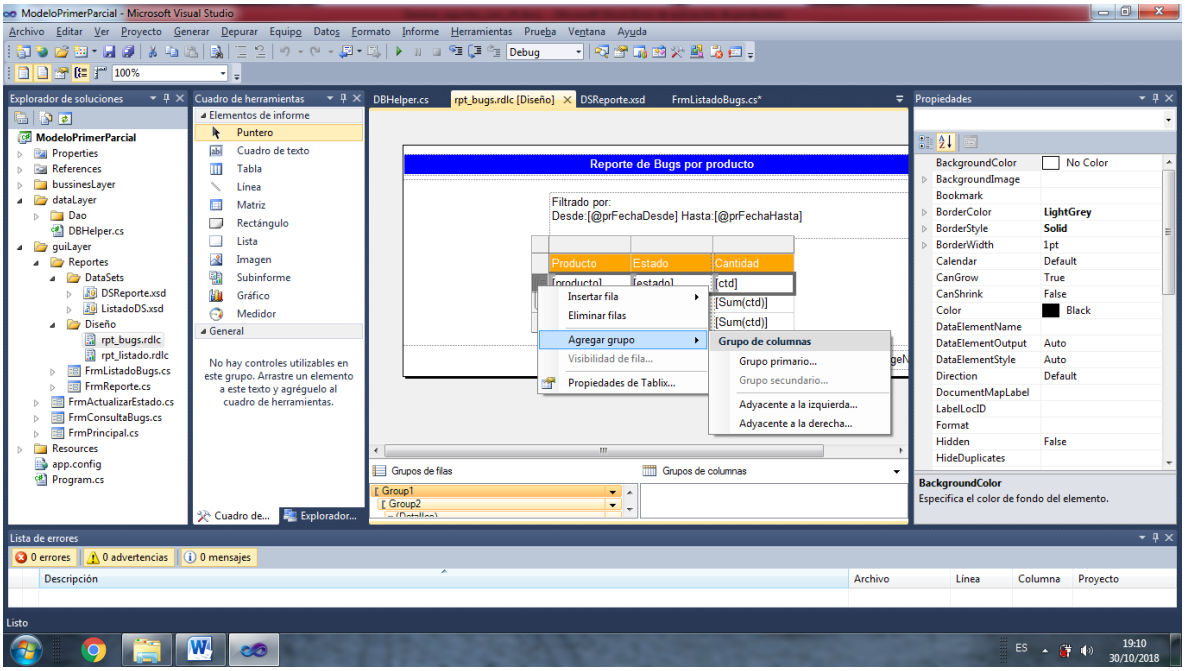
La tabla quedará definida de la siguiente manera:



2) Siguiendo los mismos pasos que en la sección anterior diseñamos el objeto .rdlc. Arrastramos la tabla de contenido y utilizamos el DATASET definido en el punto anterior, resultando:



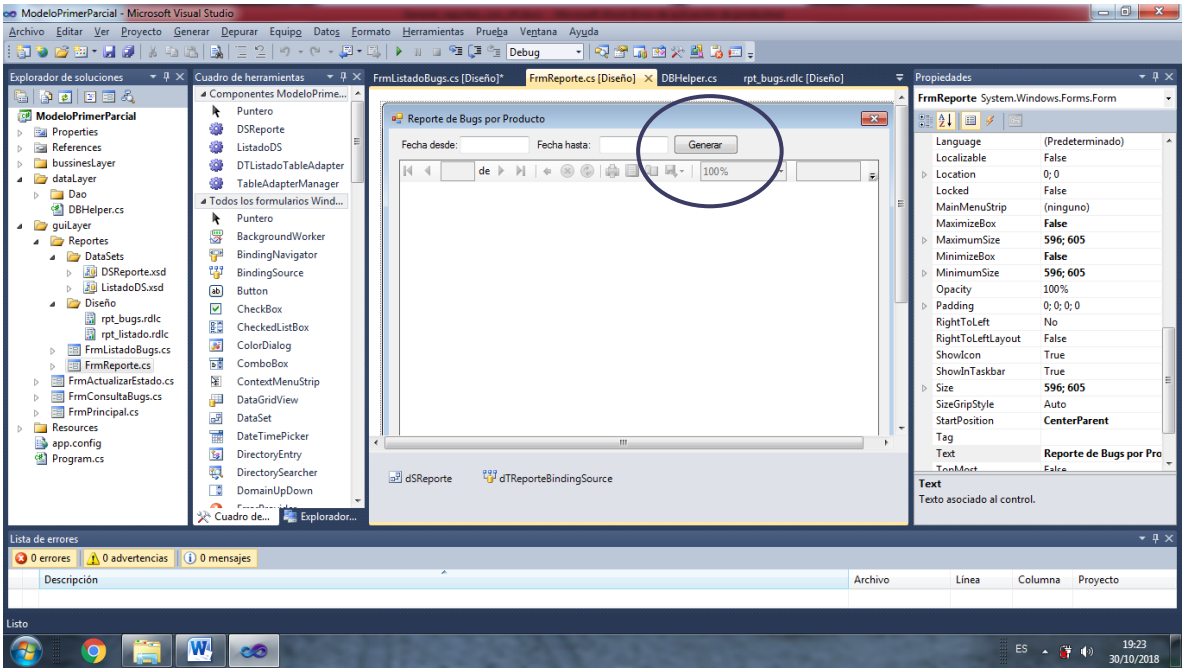
Para poder armar los grupos (o cortes de control) seleccionamos una fila de la tabla elegimos la opción *Agregar Grupo >>Grupo primario…*



Una vez creado el grupo primario, siguiendo los mismos pasos podemos crear un Grupo secundario. De esta forma tendremos para cada producto, por estado el total de bugs.

Solo nos resta agregar los totales a los grupos. Para ello parados sobre un grupo seleccionamos *Agregar Total>> Después de.* Esto nos agrega una fila adicional donde podemos ingresar una **Expresión** con una función integrada Sum() y tendremos los totales del reporte.

3) Por último vamos a enlazar en un formulario un control ReportViewer con un .rdlc del diseño. Ahora bien, en vez de elegir el origen de datos para el DataSet del reporte lo vamos a hacer por código en un botón **Generar**



Registramos un evento de click con el siguiente código:

```
  private void btnGenerar_Click(object sender, EventArgs e)
        {
            if (txtDesde.Text != "" && txtHasta.Text != "") {
                rpvBugs.LocalReport.SetParameters(new ReportParameter []{ new
ReportParameter("prFechaDesde", txtDesde.Text), new ReportParameter("prFechaHasta",
txtHasta.Text) });
                //DATASOURCE
                rpvBugs.LocalReport.DataSources.Clear();
                rpvBugs.LocalReport.DataSources.Add(new ReportDataSource("DataSet1",
DBHelper.getDBHelper().GenerarReporte(txtDesde.Text, txtHasta.Text)));
                rpvBugs.RefreshReport();
            }
        }
```

Básicamente los que estamos diciendo es:

- Limpiar los DataSources del reporte
- Agregar un nuevo DataSource y asociar el "DataSet1" con un datatable devuelto por nuestro `DBHelper` mediante el método `GenerarReporte()`.
- Por último llamamos al RefreshReport() para ejecutar el reporte.

Importante:

Mediante la línea:

```
rpvBugs.LocalReport.SetParameters(new ReportParameter []{ new
ReportParameter("prFechaDesde", txtDesde.Text), new ReportParameter("prFechaHasta",
txtHasta.Text) });
```

Estamos pasando al diseño del reporte los parámetros "`prFechaHasta`" y "`prFechaDesde`" con los valores de las fechas usadas como filtro de consulta.