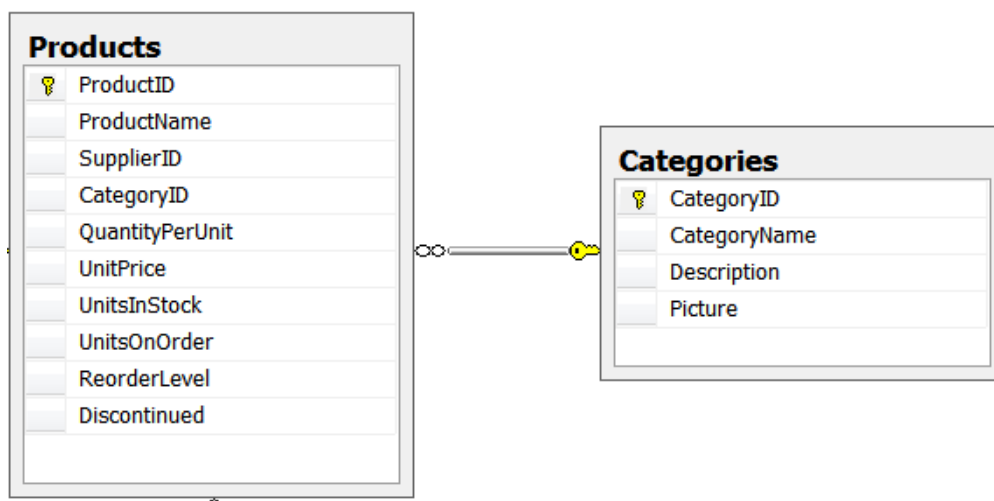


Creación de Reportes con Crystal Report con Visual Studio 2010 en Windows Form.

En esta ocasión crearemos un reporte en Windows Form con Crystal Report en Visual Studio 2010 y Net Framework 4. Como vimos en la entrada de instalación de Crystal Report para VS 2010, tenemos que [instalar CR for VS 2010](#), aunque también podemos usar Visual Studio 2005 y 2008 sin instalar nada ya que por defecto CR está en estas ediciones.

Para este ejemplo usaremos la base de datos Northwind en las tablas Productos y Categorías. El reporte mostrara los productos agrupados por categoría además de un gráfico en forma de torta o circulo para mostrar cuantos productos existen por categoría.



Materiales

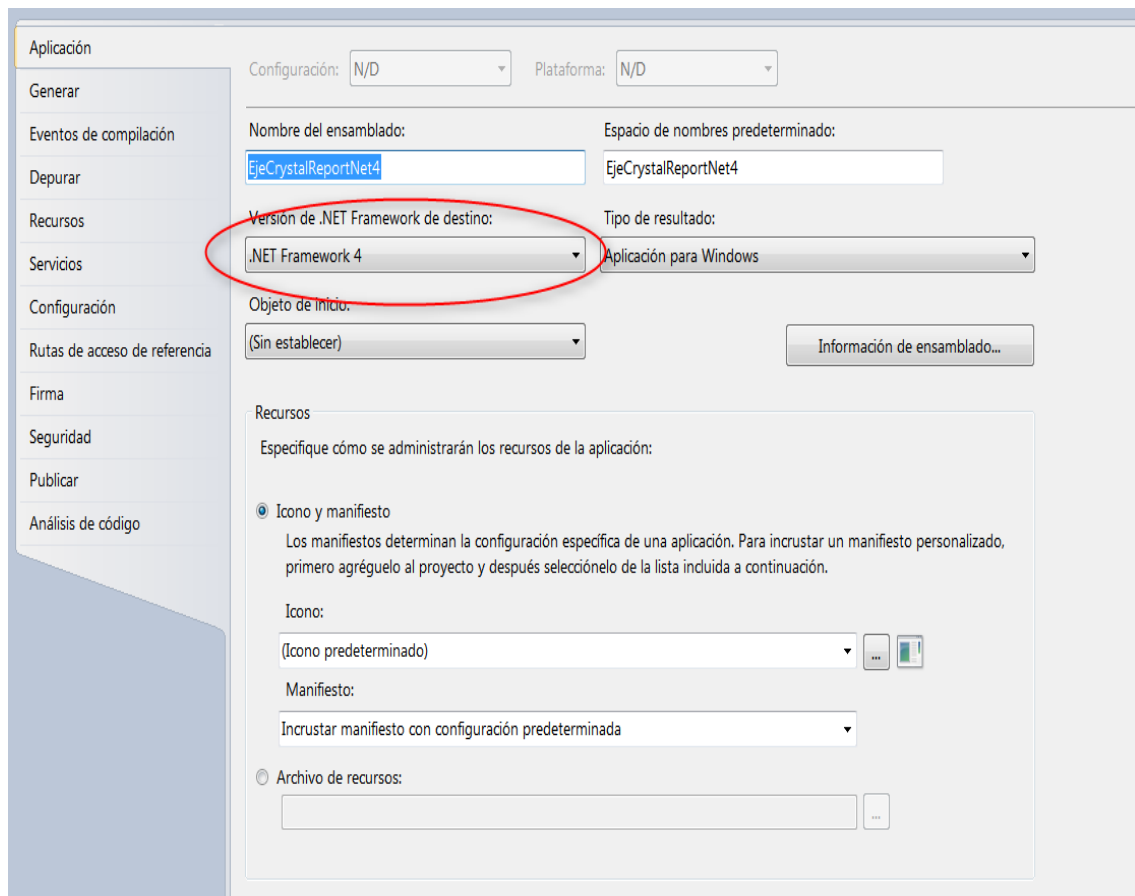
1. [Base de Datos Northwind](#)
2. [Visual Studio 2010](#)
3. [Crystal Report para Visual Studio 2010](#)
4. [Sql Server 2008 Express o superior](#)

Para realizar un reporte con Crystal Report tenemos muchas formas, usando clases, directamente con la base de datos usando tablas o vistas y usando DataSet tipados. Esta será la forma que usaremos para generar el reporte.

Creemos un nuevo Proyecto de Windows form con Net 4.0

Por defecto VS 2010 ejecuta el proyecto con Net Framework 4 Client Profile, esto se debe a que el Framework es muy grande y muchas veces no usamos todo el potencial así que podemos recurrir al Net Framework 4 Client Profile ya que este es una versión reducida del .Net Framework 4 y por ende más fácil de transportar y de instalar ya que no usa todas las librerías y nuestra aplicación puede ser más liviana para la instalación.

Pero si queremos usar Crystal Report no podemos usar la versión Reducida de .Net Framework 4 así que nos vamos al Menú→Proyecto→Propiedades del Proyecto. Y cambiamos de Motor de Ejecución a .Net Framework 4, Para poder usar CR for VS 2010



Creando el Data Set

Crearemos el data set a partir de la base de datos. Para poder realizar esto primero vamos a Explorador de Servidores en cual es muy útil para manejar un servidor de base de datos en este caso SQL SERVER 2008.

Para poder visualizar el explorador de servidores en el Menú Ver→Explorador de Servidores o (Ctrl+Alt+S).

Luego presionamos conectar con base de datos y conectamos con la base de datos Northwind

Agregar conexión

Especifique la información para establecer conexión con el origen de datos seleccionado o haga clic en "Cambiar" para elegir un origen y/o un proveedor de datos diferente.

Origen de datos:

Microsoft SQL Server (SqlClient) Cambiar...

Nombre del servidor:

localhost Actualizar

Conexión con el servidor

☒ Usar autenticación de Windows

☐ Usar autenticación de SQL Server

Nombre de usuario:

Contraseña:

☐ Guardar mi contraseña

Establecer conexión con una base de datos

☒ Seleccione o escriba el nombre de la base de datos:

Northwind

☐ Asociar con un archivo de base de datos:

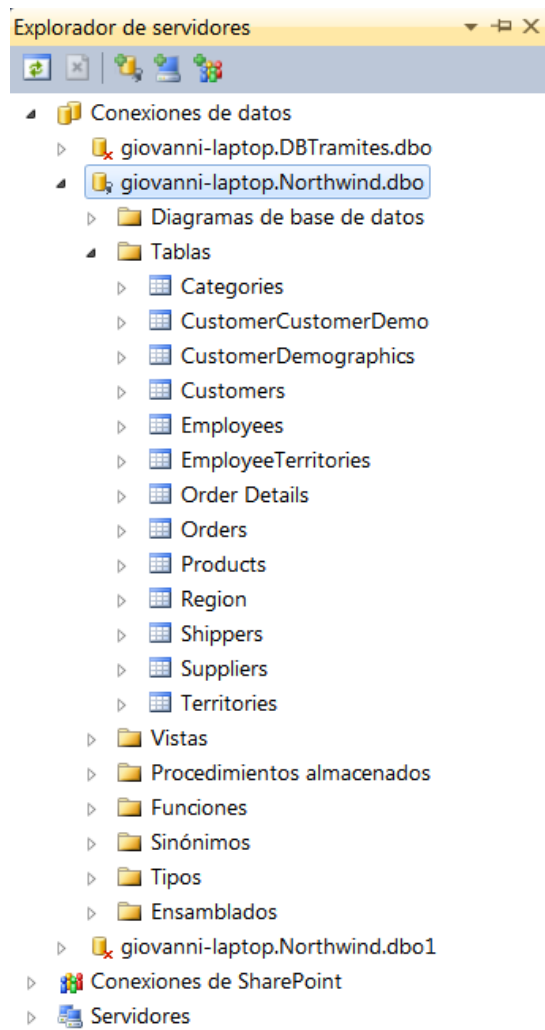
Examinar...

Nombre lógico:

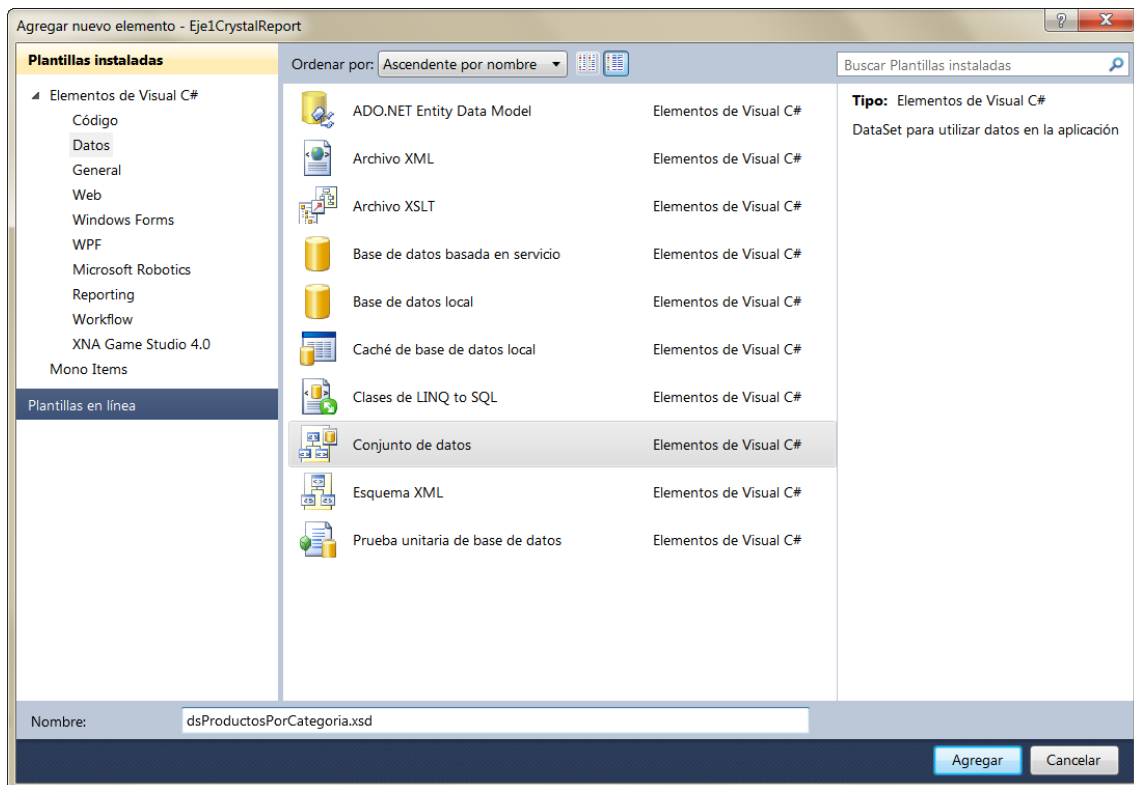
Avanzadas...

Probar conexión Aceptar Cancelar

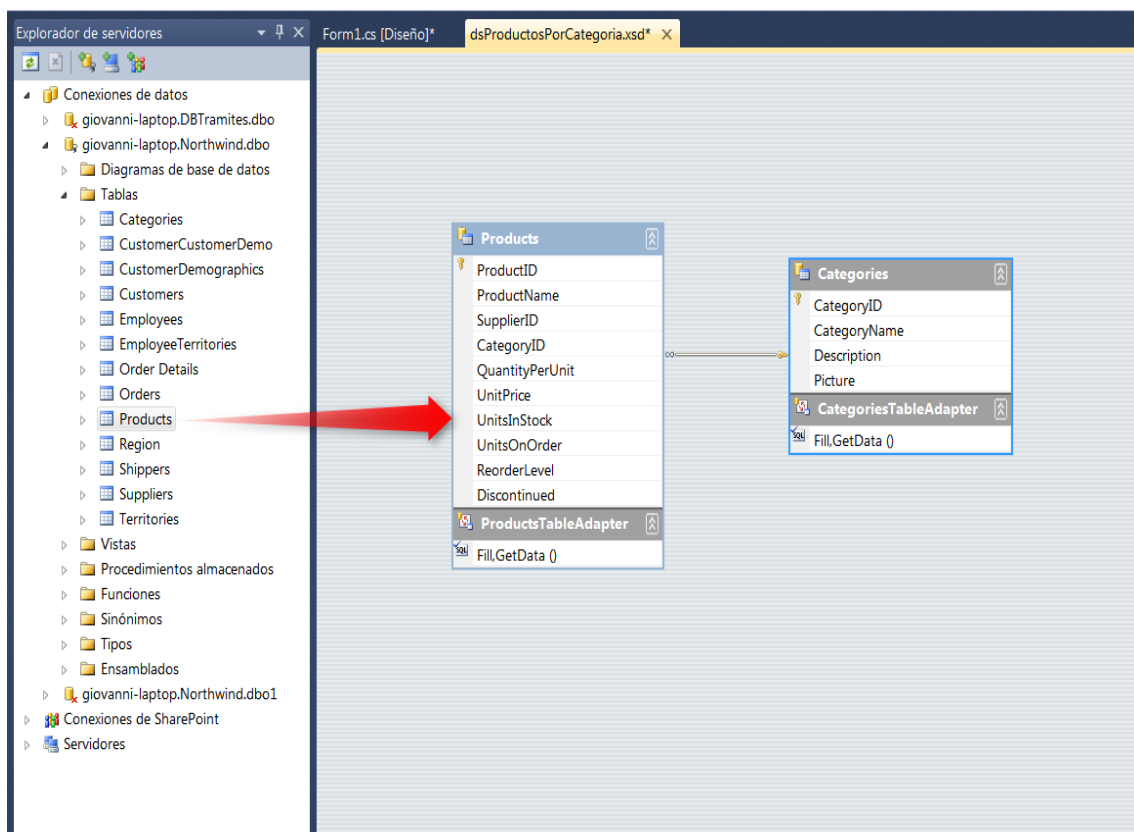
Una vez conectada la base de datos podemos ver las tablas que esta BD contiene.



Ahora agregamos un Data Set a nuestro proyecto.



Colocamos el nombre de dsProductoPorCategoria.xsd. Arrastramos la tabla para que el reporte pueda usarlo como fuente de datos.

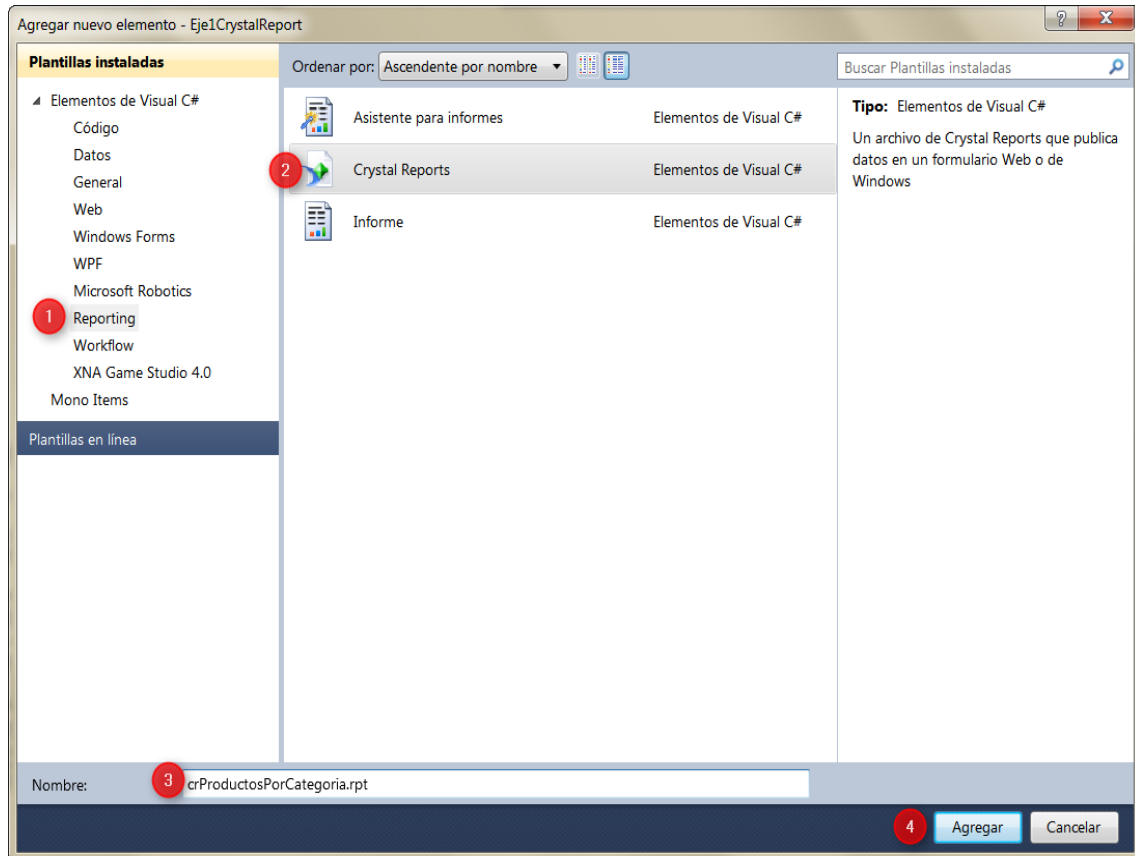


La relación se construirá automáticamente con esta estructura definida el reporte podrá tomar los datos y ordenarlos de forma que podamos agrupar los productos por categorías.

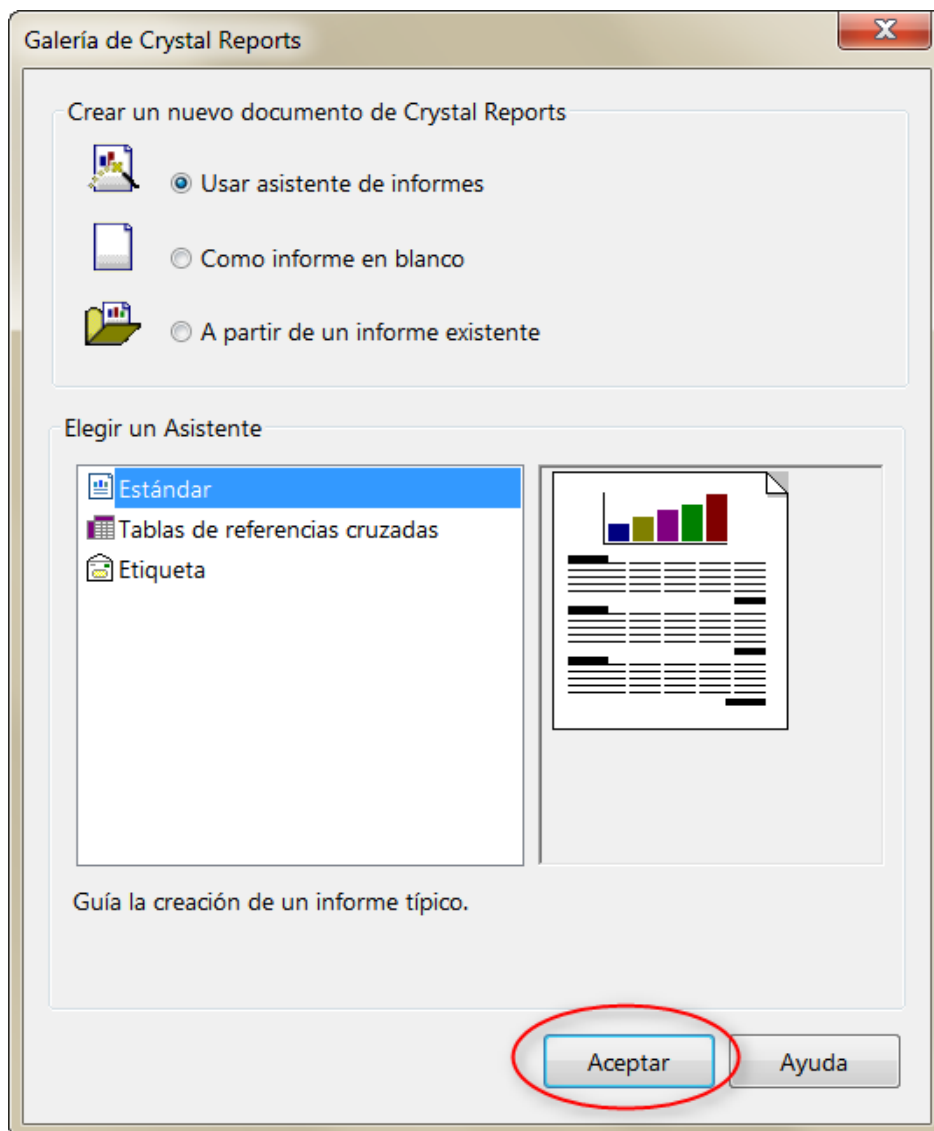
Creamos el Reporte

Para crear el reportes usaremos el asistente el cual nos permitirá elegir entre varias opciones la más adecuada para nuestro reporte.

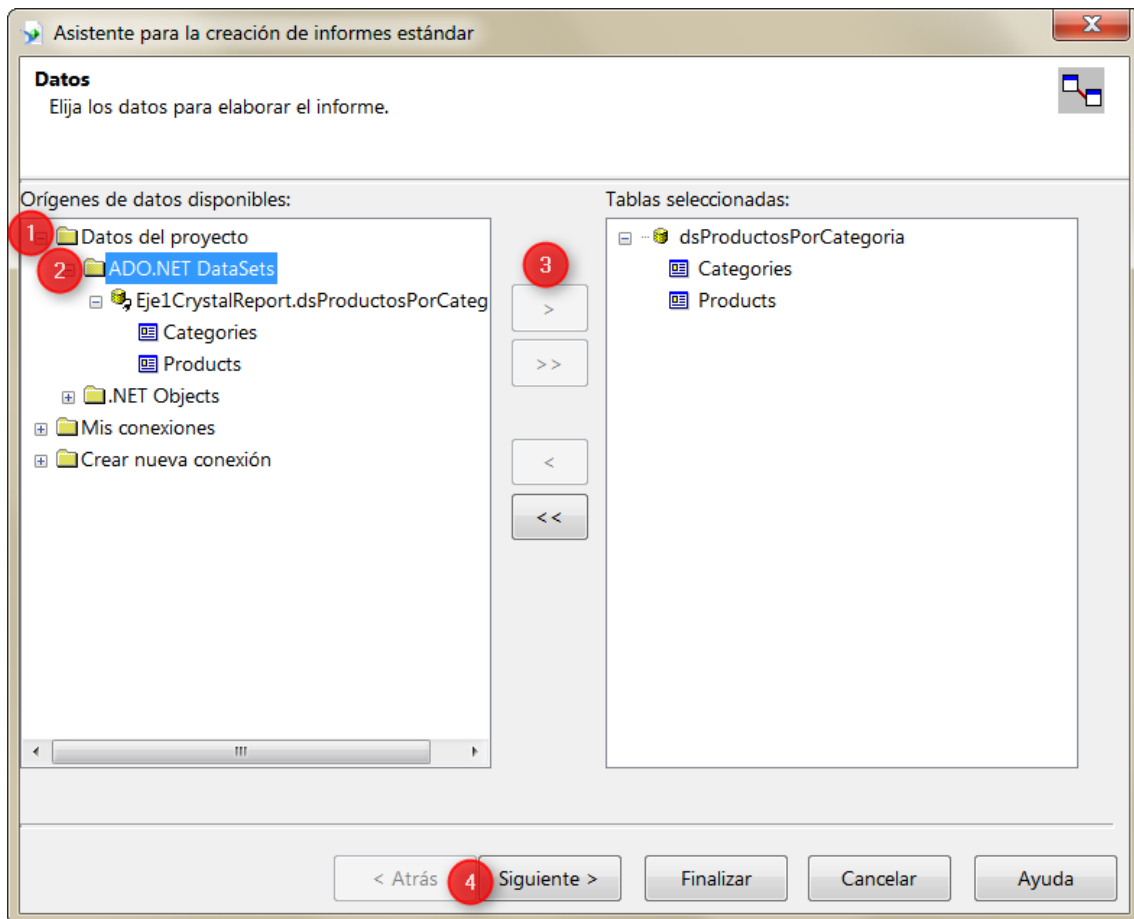
Agregamos el reporte de Crystal Report



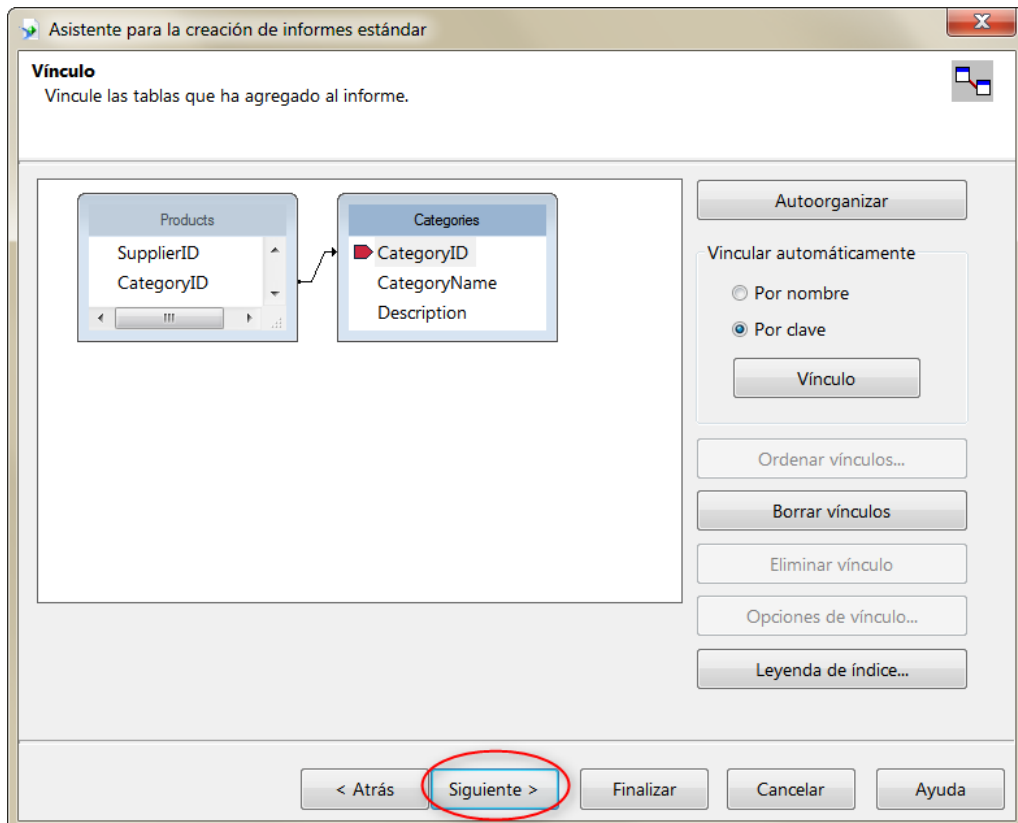
Cuando agregamos el reporte usaremos el asistente de informe para ayudarnos en la configuración del reporte.



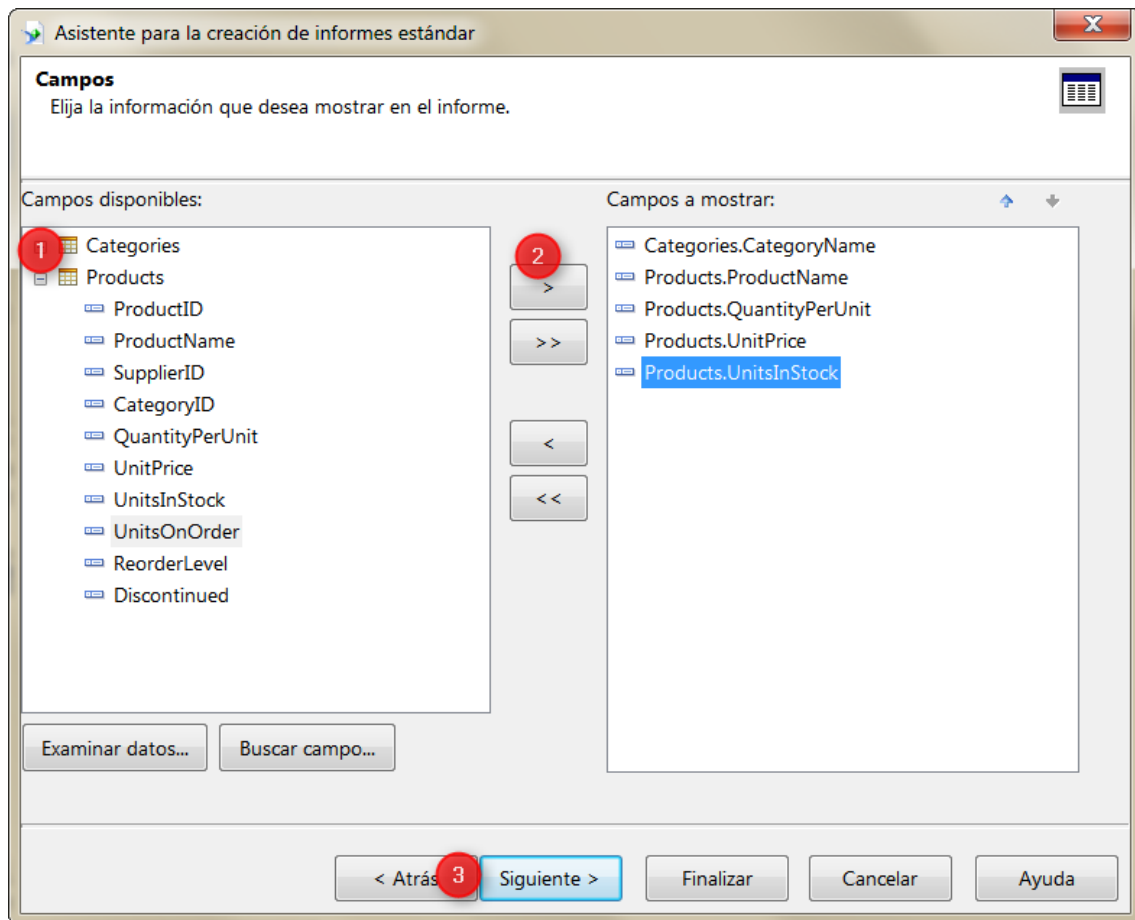
Elegimos la conexión del reporte del cual tomara la estructura para definir el informe, para este paso solo tenemos que elegir el data set tipado que construimos anteriormente



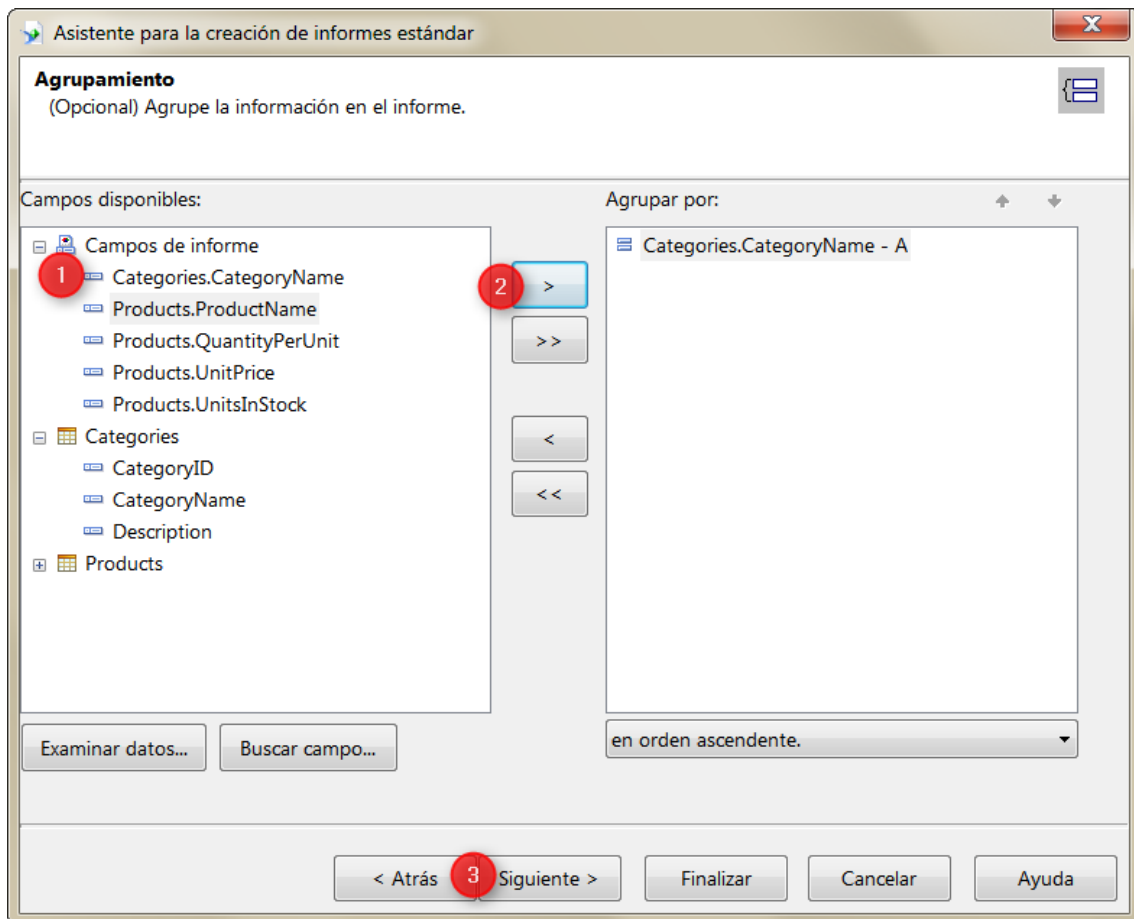
Vemos la tablas relacionadas y damos siguiente



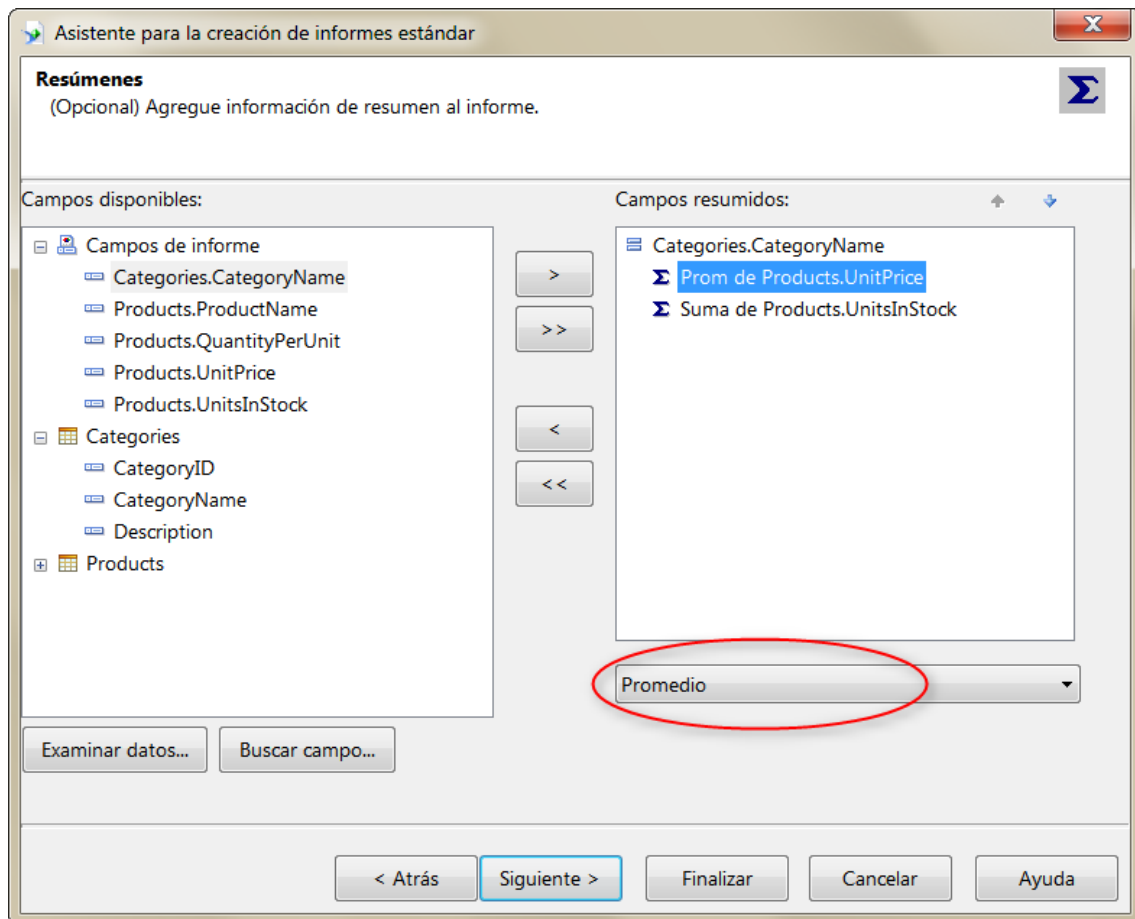
Seleccionamos los campos a mostrar esto es importante ya que si elegimos muchos campos nuestro informe se verá muy lleno y eso puede provocar que el informe no sea tan legible, así que solo tenemos que usar los campos que realmente necesitamos.



Ahora podemos agrupar los datos del informe esto es opcional pero como deseamos tener un informe que muestre los productos por categoría debemos agrupar los datos por la categoría.



Luego también podemos insertar a nuestro informe campos de resúmenes (opcional). En estos campos de resumen, indicaremos que calcule el promedio de precios de una categoría y la sumatoria del stock de los productos de esa categoría.



En esta sección podemos ordenar por los grupos superiores o inferiores pero por defecto elegimos ninguno.

Asistente para la creación de informes estándar

Ordenación de grupos
(Opcional) Ordene los grupos según los totales resumidos.

Grupo que se va ordenar:

Categories.CategoryName

¿Qué tipo de clasificación de grupo desea visualizar?

☒ Ninguno

☐ Los 5 grupos superiores

☐ Los últimos 5 grupos

Comparar valores de resumen de los grupos superiores o inferiores:

< Atrás **Siguiente >** Finalizar Cancelar Ayuda

Agregamos el grafico al reporte (grafico circular o de torta)

Asistente para la creación de informes estándar

Gráfico
(Opcional) Incluya un gráfico en el informe.

¿Qué tipo de gráfico desea visualizar?

☐ Ningún gráfico ☒ Gráfico de barras ☐ Gráfico de líneas ☒ Gráfico circular 1

Título del gráfico:

2 Suma de UnitsInStock / CategoryName

A cambio de:

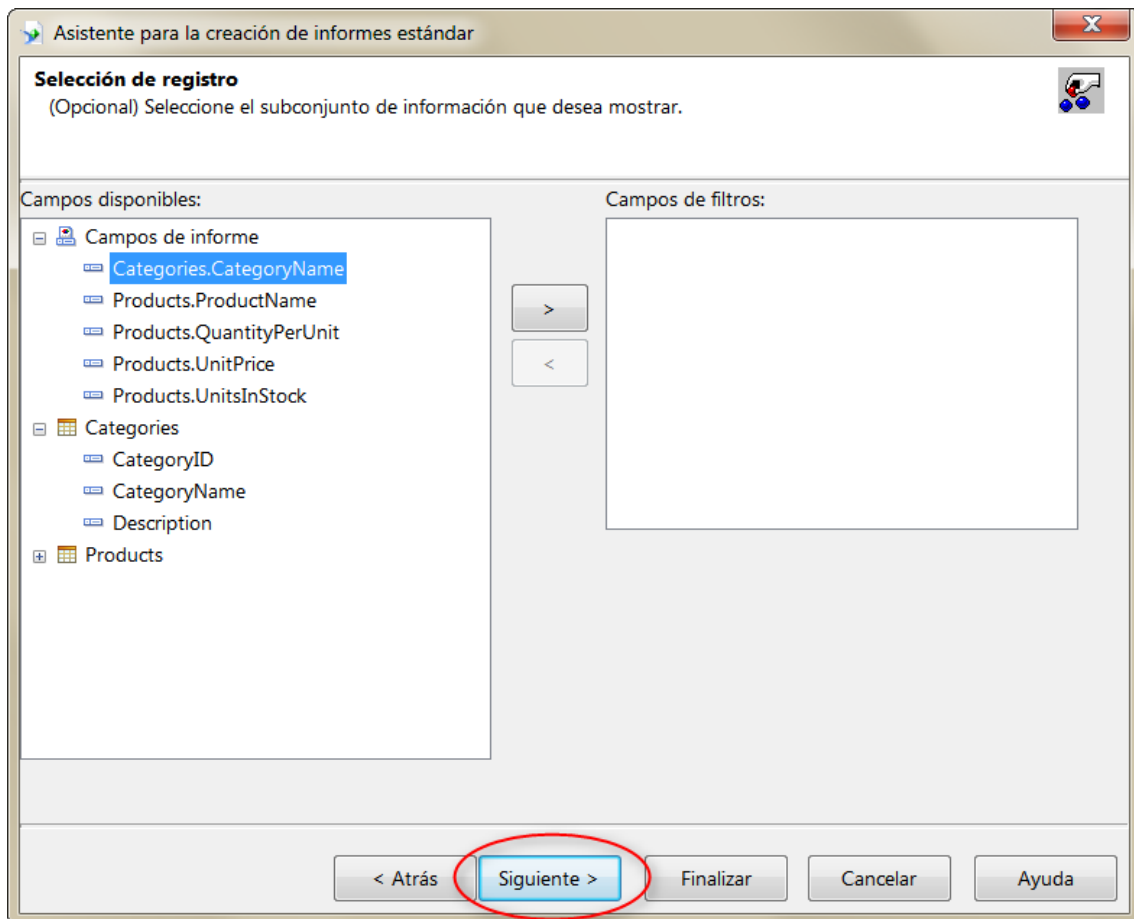
Categories.CategoryName

Mostrar resumen:

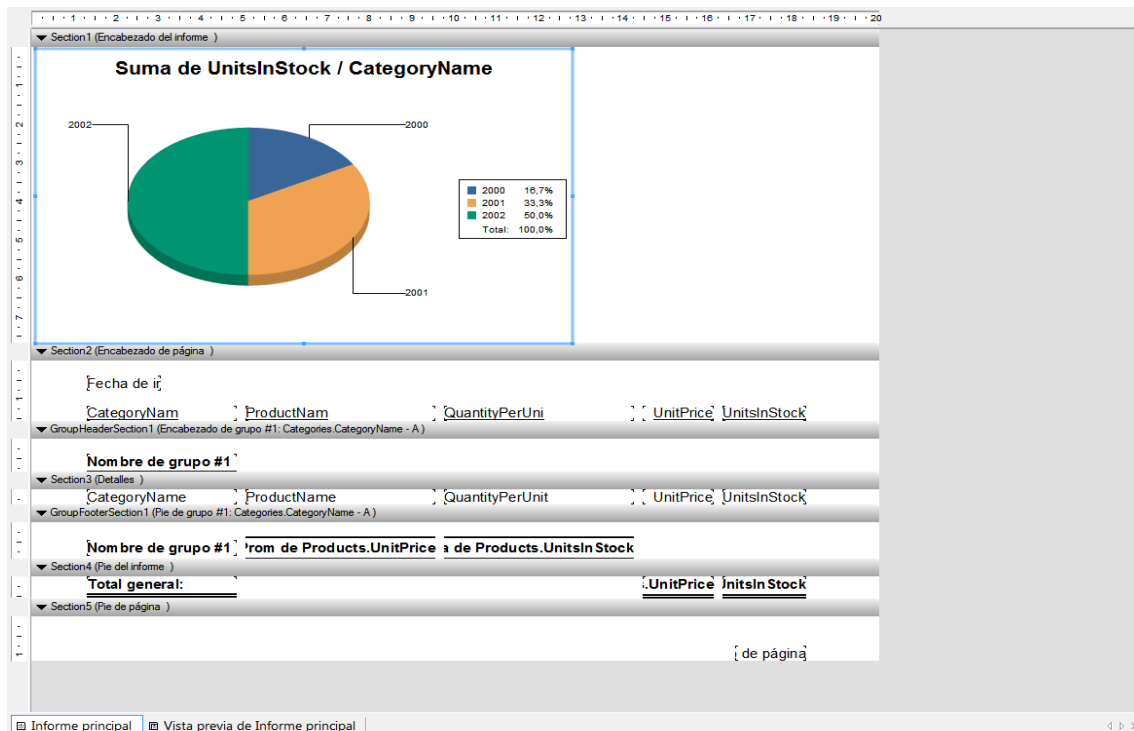
3 Suma de Products.UnitsInStock

< Atrás 4 Siguiente > Finalizar Cancelar Ayuda

Si deseamos realizar una sub agrupación podemos seleccionar en esta ventana pero para nuestro caso es innecesario.

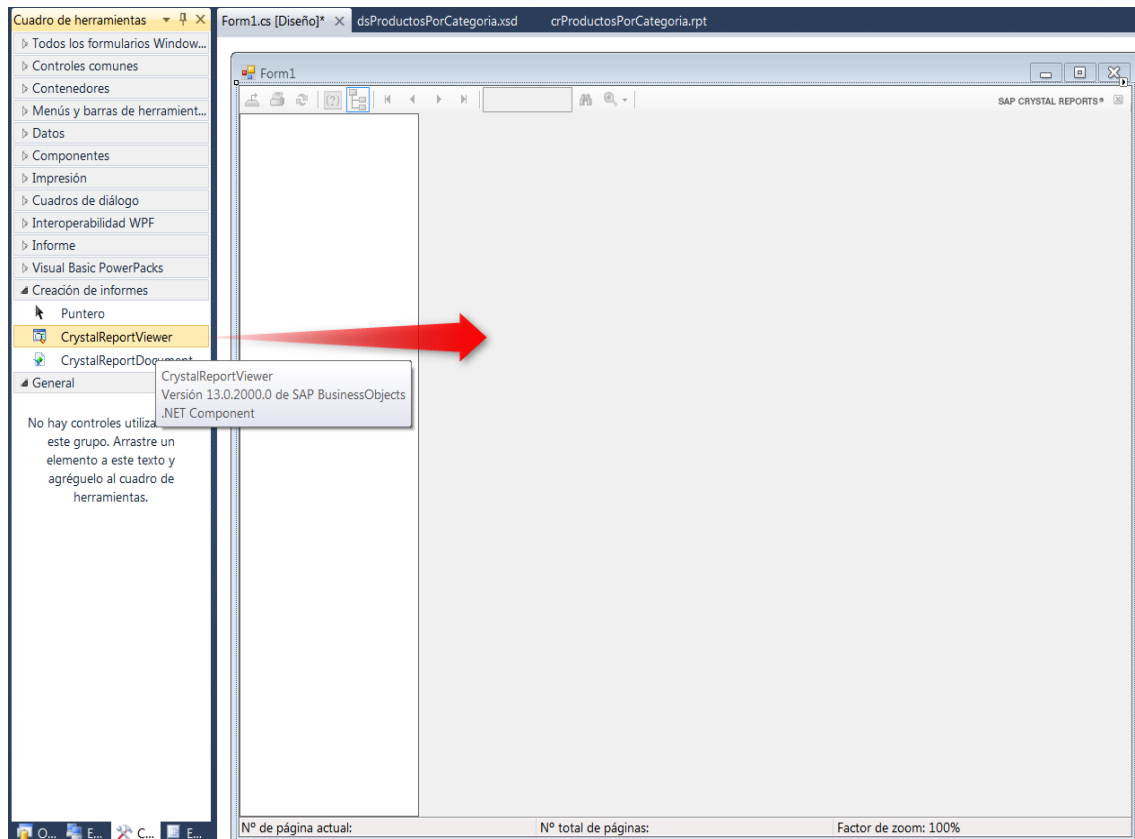


Finalmente nuestro informe está concluido.



Si requerimos colocar el informe al último del informe podemos hacerlo con solo arrastra el grafico a la parte inferior de la sección pie de informe (Section 4).

Pero nuestro trabajo no termina ahí tenemos que agregar un visor del reporte al formulario Windows Form.



Poblar el Reporte

Ahora que tenemos definida la estructura y tenemos el visor de informe. Esto no significa que el reporte esté listo ya que el reporte por si solo aun no está lleno, si bien usamos el data set para definir la estructura del reporte este dataset no está lleno. Ahora toca llenar el data set con los datos de la base de datos.

Creamos el método PoblarReporte el cual llena el data set con las sentencias SQL que se requieran, como tenemos que llenar dos tablas usaremos dos distintos DataAdapters.

```

//Metodo Para poblar el reporte
private void PoblarReporte()
{
    //Definimos las sentencias las cuales seleccionan todo los datos
    string SqlProductos = "Select * from Products";
    string SqlCategorias = "Select * from Categories";

    dsProductosPorCategoria dsPC = new dsProductosPorCategoria();
    try
    {
        //creamos los data adapters
        SqlConnection Conne = new SqlConnection("Data Source=localhost;Initial Catalog=Northwind;Integrated Security=true");
        SqlDataAdapter sqlDaProductos = new SqlDataAdapter(SqlProductos, Conne);
        SqlDataAdapter sqlDaCategorias = new SqlDataAdapter(SqlCategorias, Conne);

        // empezamos el a poblar las tablas del Data Set Tipado
        sqlDaCategorias.Fill(dsPC, "Categories");
        sqlDaProductos.Fill(dsPC, "Products");

        //Ahora Poblamos el informe y lo mostramos
        crProductosPorCategoria Informe = new crProductosPorCategoria();
        Informe.SetDataSource(dsPC);
        crvwProductosPorCategoria.ReportSource = Informe;

    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }
}

```

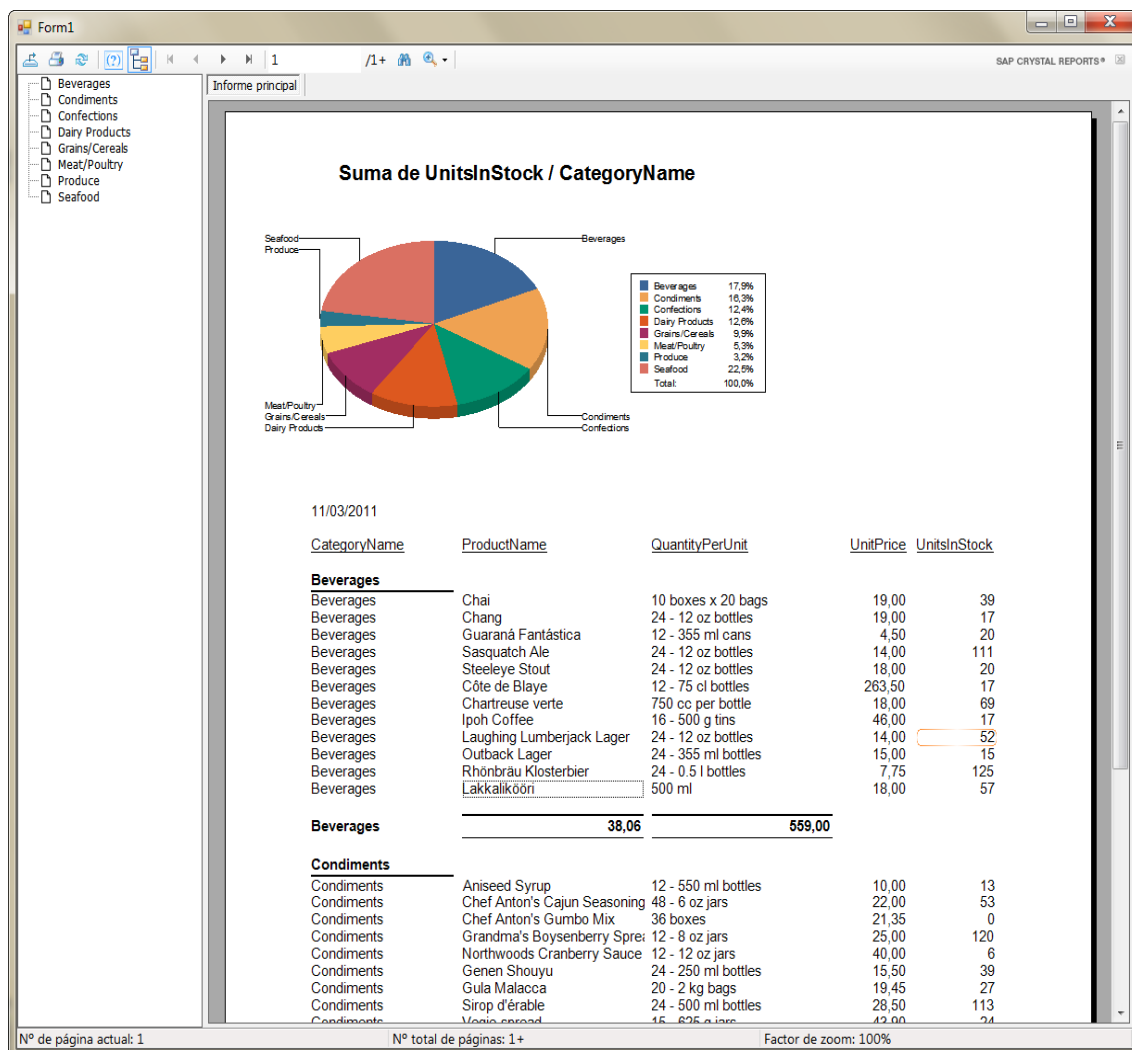
Ahora llamamos este método en el evento Load del formulario.

```

private void Form1_Load(object sender, EventArgs e)
{
    PoblarReporte();
}

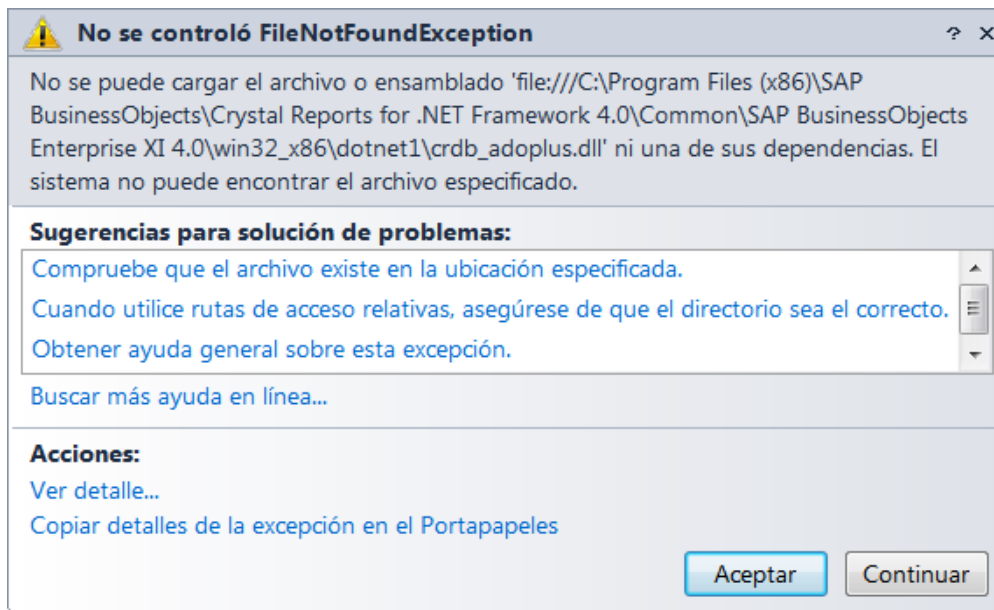
```

Finalmente ejecutamos el reporte y vemos como nuestro reporte se genero con éxito.



Nota Importante para la ejecución en VS 2010.

Cuando creamos de esta manera el reporte en VS 2010 tendría que agregar automáticamente las librerías necesarias, además configurar el App.Config de manera que la ejecución resulte exitosa. Pero por alguna razón en algunos VS 2010 no configura de manera correcta el App.config y nuestro reporte no se ejecuta, en otros no existe el problema. Estuve probando en diferentes VS 2010 sobre todo ultimate pero me sale un error, en otros no y todo es normal, tal vez sea porque algunos PC eran de X64 y otros de X32 pero, no hay problema si tenemos este error de ejecución el cual indica que no podemos compilar porque no se reconoce las librerías de CR.



Para que se ejecute de manera correcta solo tenemos que configurar el App.Config, agregando esta línea.

```
<startup useLegacyV2RuntimeActivationPolicy="true">  
<supportedRuntime version="v4.0"/>  
</startup>
```

Y comentando la línea

```
<!--<startup><supportedRuntime version="v4.0"  
sku=".NETFramework,Version=v4.0"/></startup>-->
```

Quedando el App.config como sigue:

```
<?xml version="1.0"?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="EjeCrystalReportNet4.Properties.Settings.NorthwindConnectionString"
      connectionString="Data Source=localhost;Initial Catalog=Northwind;Integrated Security=True"
      providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <!--<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/></startup>-->
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

Con esto ya se ejecuta nuestro reporte.