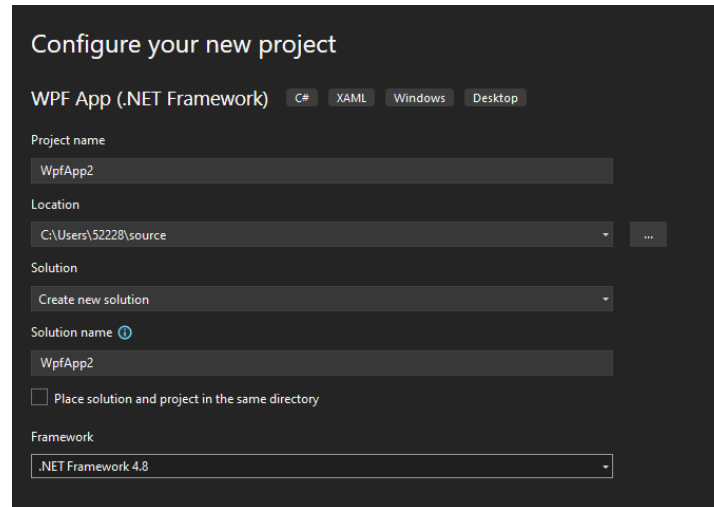


# Databinding with WPF

Alejandro Chacón

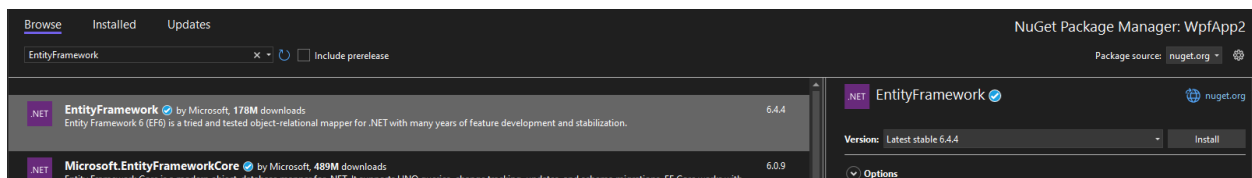
## 1. Crear proyecto

Se debe crear el proyecto usando la plantilla WPF App (.NET Framework), cuidando que el framework sea .NET 4.8. Mi primer intento Salió mal, pues escogí una plantilla que usaba .NET 6.0, y eso ocasionó problemas en el siguiente paso.



## 2. Instalar paquete Entity Framework NuGet

Clic derecho en el proyecto muestra la opción para administrar los paquetes NuGet de dicho proyecto. Clic en esta opción muestra la ventana para buscar los paquetes que se necesitan instalar. En la pestaña Browse, se selecciona busca “EntityFramework” y se selecciona la primer opción. El panel de la muestra la descripción del paquete seleccionado y el botón de descarga.



## 3. Definir modelo

Para el enfoque Code first, se escriben primero las clases en C# que se van a utilizar. Cada clase tiene dentro de sí propiedades que permiten la navegación entre las clases, representando programáticamente sus **relaciones y cardinalidades**.

Un Product tiene una Category	Una Category tiene muchos Products
<pre>namespace WPFwithEFSample {     public class Product     {         public int ProductId { get; set; }         public string Name { get; set; }     } }</pre>	<pre>using System.Collections.ObjectModel;  namespace WPFwithEFSample {     public class Category     {     } }</pre>

```

public int CategoryId { get; set; }
public virtual Category Category { get; set; }
}

public Category()
{
    this.Products = new
    ObservableCollection<Product>();
}

public int CategoryId { get; set; }
public string Name { get; set; }

public virtual ObservableCollection<Product>
Products { get; private set; }
}

```

También es necesario crear una clase derivada de DbContext para que el modelo funcione.

```

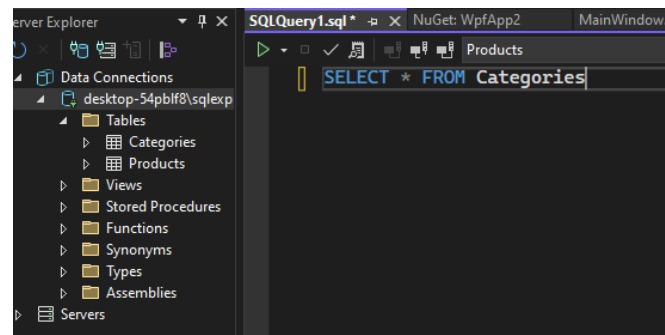
using System.Data.Entity;

namespace WPFwithEFSample
{
    public class ProductContext : DbContext
    {
        public DbSet<Category> Categories { get; set; }
        public DbSet<Product> Products { get; set; }
    }
}

```

Si se hace Code First, es importante compilar el proyecto antes de continuar.

Para Database First, primero se debe definir la base de datos del proyecto antes de crear el modelo. Se establece una conexión al servidor en donde queramos trabajar y se selecciona la base de datos correspondiente; si queremos crear una nueva, basta con escribir el nombre de una base que no exista y Visual Studio nos preguntará si la queremos crear. Visual Studio tiene, también, módulos para interactuar con el servidor como si se estuviese usando SQL Server Management Studio. Desde aquí se pueden ejecutar los comandos en SQL para crear las tablas del proyecto.



Cuando se ha hecho esto, ya se puede generar el modelo de entidades usando el Entity Framework Designer, el cual genera el código de las clases de manera automática usando ingeniería inversa y el esquema de la base de datos previamente definida.

## 4. Databinding

Una vez construido el modelo, se deben ligar nuestros controladores GUI con las entidades que creamos, esto se hace asignando Data Sources (fuentes de datos) a cada ventana que los necesite. Si el modelo fue creado Code First, es muy importante compilar antes de intentar asignar Data Sources, pues del contrario, las entidades no aparecerán en la ventana para seleccionarl

Es importante mencionar que si dos entidades están relacionadas, asignar el Data Source de una de ellas al controlador asignará, también, el de las entidades relacionadas a esta entidad que arrastramos. En este caso, como Product se relaciona con Category, asignar Category al controlador asigna automáticamente a Product también.

Cuando la ventana tiene asignadas sus Data Sources, estas pueden ser arrastradas a las ventanas y automáticamente se generan tablas (con su código XAML) que muestran los atributos de todas las entidades de esa fuente de datos. Ya no hay que diseñarla a mano, como había que hacerlo en JavaFX.

Finalmente, se escribe el código C# para hacer funcionar los controladores y los datos que manejan. Una curiosidad del proyecto es que la tabla de Products cambia sus contenidos dependiendo qué tupla se seleccione en la tabla Category. Puede resultar algo confuso al principio.

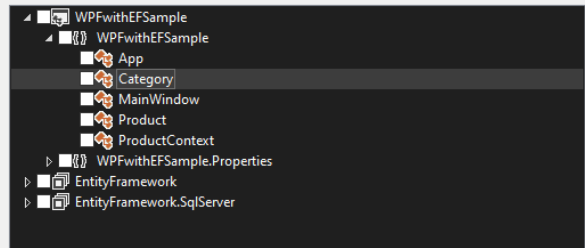
Data Source Configuration Wizard



### Select the Data Objects

Expand the referenced assemblies and namespaces to select your objects. If an object is missing from the list, expand the assembly, cancel the wizard and rebuild the project that contains the object.

What objects do you want to bind to?



☒ Hide system assemblies

< Previous

Next >

Finish

MainWindow

Category Id	Name	Product Id	
1	qwer	1	
1	fdsa	3	
1	4321	4	

MainWindow

Category Id	Name	Product Id	
1	qwer		
2	asdf	2	
2	6534	5	