



UT02_ 03 Persistencia de Datos DB First

ASP .NET MVC



- DataBase First
- Conectar con la DB
- Generación automática de modelo + scaffold de los modelos
- Modelo híbrido
- Trabajando con tablas relacionadas
- One to One
- One to Many
- Many to Many



EF Core - DataBase First

En este tipo de enfoque partiremos de una base de datos ya existente.

A partir de esta generamos los modelos, gracias al ORM EF Core, y mediante *scaffolding* se crearán los controladores y vistas asociadas al modelo que elijamos.

Trataremos los siguientes aspectos:

- Importar una BD en SQL Server Express LocalDB
- Generar Modelos de las tablas de la BD
- Generar controlador y vistas del modelo
- Añadir Clase de Contexto al Builder de la Aplicación



EF Core - DataBase First

Importar una BD

Dado que este enfoque parte de una base de datos existente, importaremos una base de datos, con su conjunto de tablas, en nuestra base de datos local SQL Server Express LocalDB.

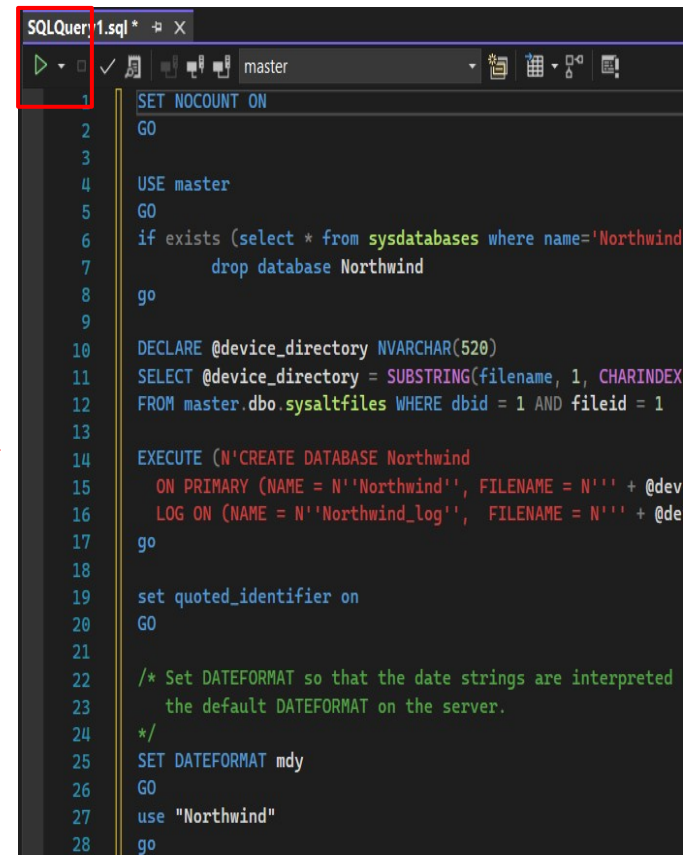
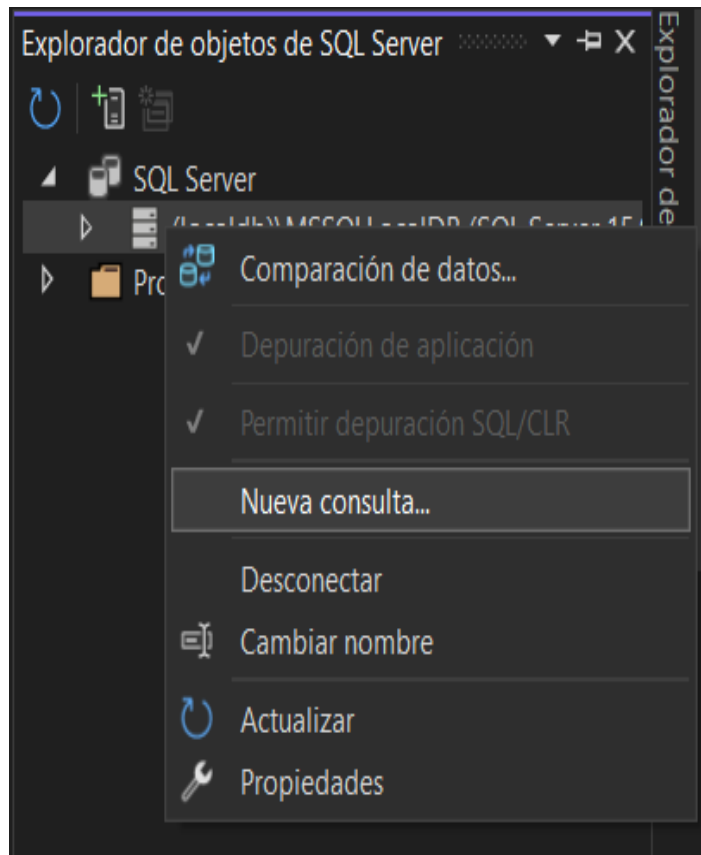
Utilizaremos la base de datos que podemos encontrar [aquí](#), y realizaremos:

- Descarga del archivo .zip y descomprimos.
- Abrimos el archivo de extensión .sql con un editor de texto (Sublime, notepad++, etc) y copiamos todas las consultas
- Realizamos una nueva consulta desde el explorador de objetos SQL Server de Visual Studio



EF Core - DataBase First

Importar una BD

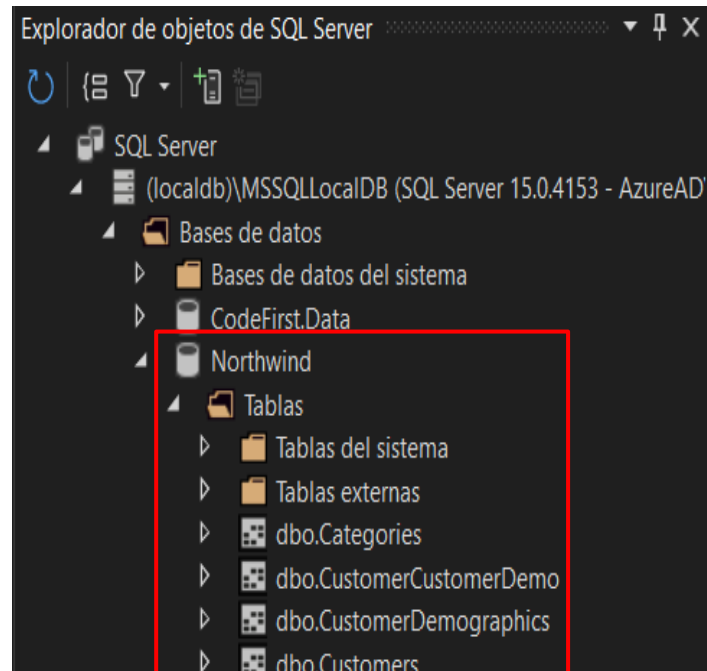




EF Core - DataBase First

Importar una BD

Si inspeccionamos en el “Explorador de objetos de SQL Server” veremos que se ha creado la base de datos (Northwind en este ejemplo) con sus tablas





EF Core - DataBase First

Generar Modelos de la BD

Para realizar esta acción tendremos que hacer “ingeniería inversa” usando el comando Scaffold-DbContext

Esta acción generará las clases de los modelos y la clase de contexto (derivada de DbContext) basándose en el esquema de la base de datos existente.

NOTA: Es necesario tener instalados los paquetes NuGet:

- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools



EF Core - DataBase First

Generar Modelos de la BD

El comando a usar tiene la siguiente Sintaxis:

```
Scaffold-DbContext [-Connection] [-Provider] [-OutputDir] [-Context] [-Schemas>] [-Tables>]  
                  [-DataAnnotations] [-Force] [-Project] [-StartupProject] [<CommonParameters>]
```

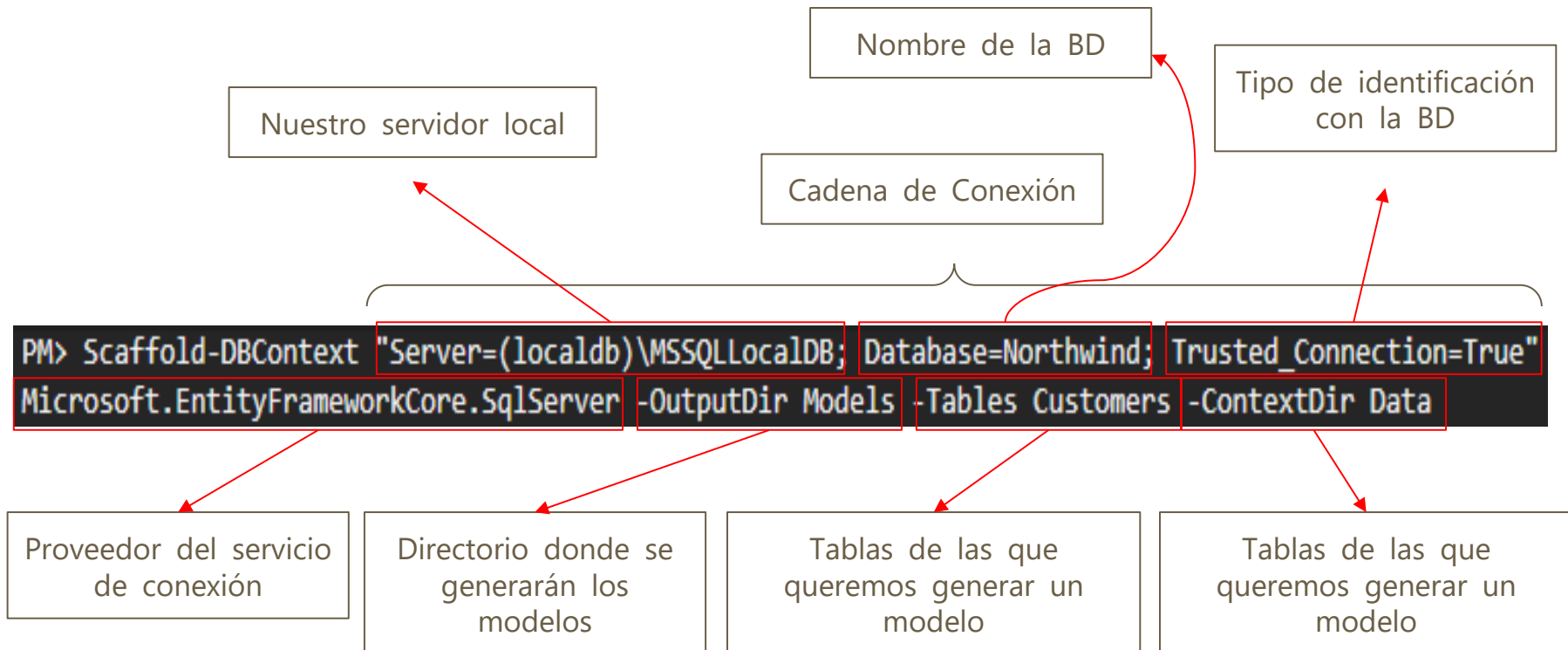
Usaremos la “Consola de paquetes NuGet” para escribir el comando, que tendrá el siguiente aspecto:



EF Core - DataBase First

Generar Modelos de la BD

El comando a usar tiene la siguiente Sintaxis:

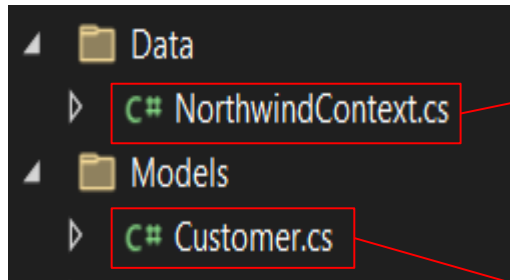




EF Core - DataBase First

Generar Modelos de la BD

Esta acción genera lo siguiente:



Clase del contexto de
conexión con la BD

Clase del modelo de la entidad de
la BD elegida



EF Core - DataBase First

Generar Controlador y Vistas del Modelo

En este punto ya podemos realizar el *scaffolding* en base al modelo, de la misma forma que lo hicimos en el enfoque *Code First* pero esta vez con el modelo generado por ingeniería inversa.

NOTA: Esta vez no tendremos que generar una nueva Clase de contexto de datos, ya que la hemos generado con el comando *Scaffold-DbContext*.

Por ello, tendremos que seleccionar la clase generada, y guardada en el directorio Data, en el desplegable correspondiente al hacer el *Scaffold*



EF Core - DataBase First

Añadir Clase de Contexto al Builder de la Aplicación

Finalmente, para que nuestro proyecto pueda realizar consultas con la base de datos importada, tenemos que añadir nuestra clase de contexto (DbContext) como servicio en el builder de nuestra WebApplication

Esto lo realizaremos en el archivo “Program.cs”

```
builder.Services.AddDbContext<NorthwindContext>();
```

Si miramos en “NorthwindContext.cs” vemos que la cadena de conexión está definida dentro, junto a un mensaje *Warning*.

Lo recomendable es incluir esa cadena en el archivo appsettings.json



Ejercicio

Realizar las siguientes acciones:

- Generar un nuevo proyecto ASP.NET Core MVC
- Importar la base de datos del ejemplo
- Generar, mediante el ORM, un modelo de la entidad *Suppliers*
- Realizar el *scaffold* del controlador y vistas
- Editar lo necesario para conseguir ejecutar el proyecto y que se listen por defecto los elementos de la entidad *Suppliers*