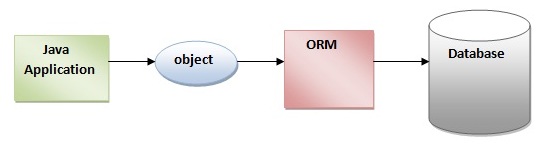
Entity Framework (EF) nos sirve en este caso para realizar una conexión de una BBDD existente a un proyecto en C#.

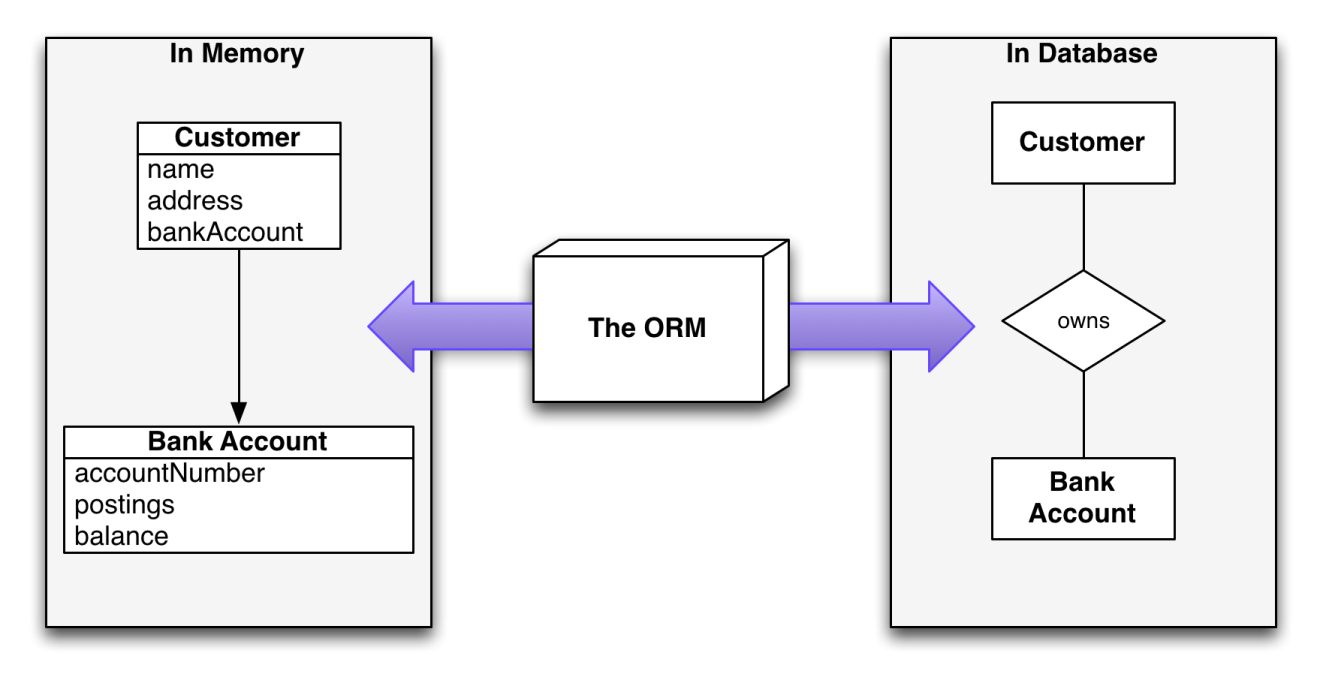
Para realizar esa conversión EF, utiliza ORM Mapeo Objeto-Relacional (Object Realtional Mapping). ORM, es una técnica para poder comunicar datos de una BBDD relacional a un modelo orientado a objetos. Para poder crear/instanciar esos objetos como ya sabemos, necesitamos crear previamente las clases.

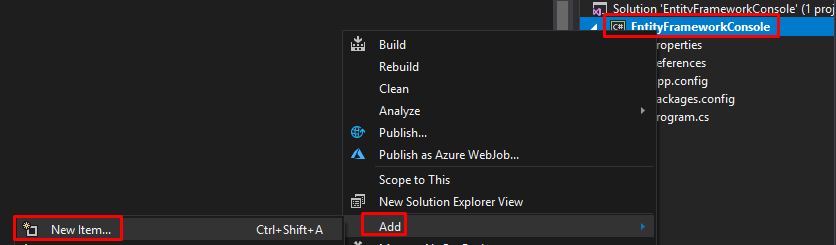
* Realizar el ORM de forma manual nosotros mismos picando código.
* Utilizar ADO.NET que es lo equivalente a lo anterior, pero realizado automáticamente por Visual Studio para nosotros. El principal beneficio de ADO.NET es la rapidez, sobre todo cuando si la BBDD es de grandes dimensiones.

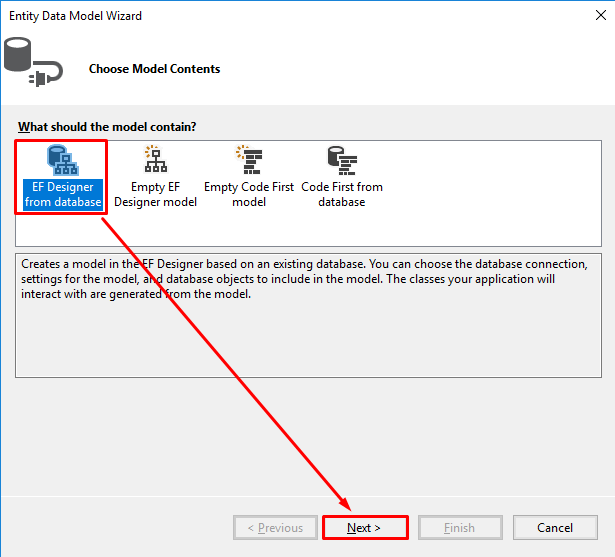
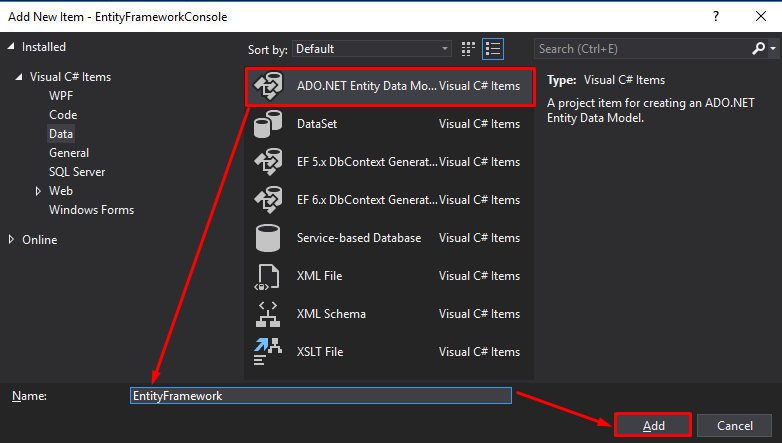
El principal beneficio de realizar está gestión automáticamente es que nos será independiente de la BBDD que estemos usando la podríamos cambiar si la BBDD es la misma.

Ado por tanto, es una herramienta de Visual Studio disponible para el framework de .NET y aunque se usa principalmente para acceder a datos relacionales, también podemos acceder a datos no relacionales.

Y es ideal para realizar trabajar con BBDD ya que podemos hacer un CRUD (Create, Read, Update, Delete)

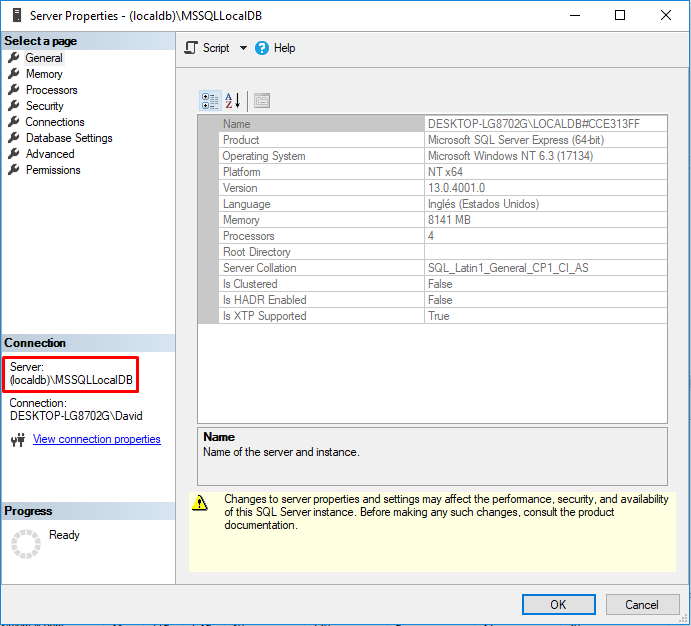
* Create: INSERT
* Read: SELECT
* Update: UPDATE
* Delete: DELETE

Creamos un proyecto de consola New Project>Console y añadimos un nuevo elemento de tipo Data > ADO.NET Entity Data Model



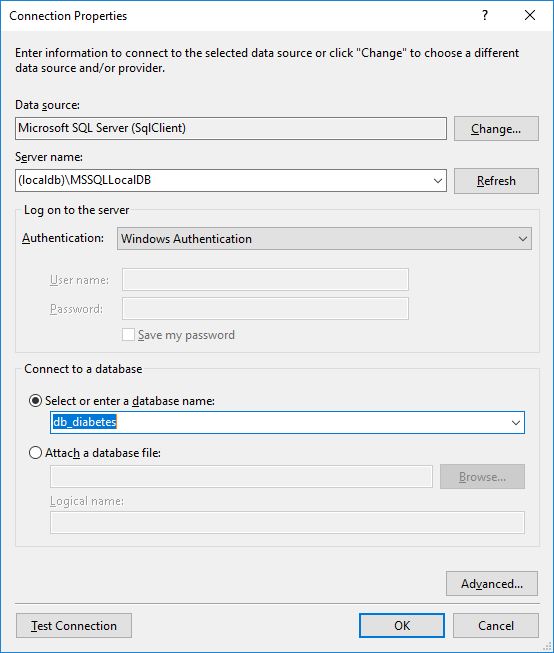
Bien, realizamos una nueva conexión para ello, primeramente necesitamos saber el nombre de la conexión para poder seleccionar la BBDD. Al ser una BBDD en local y tenerla alojada en Microsoft SQL Server Management Studio, abrimos el SQL Server y hacemos clic en propiedades sobre él. (o también podemos el nombre del server desde aquí ya que lo estamos viendo)



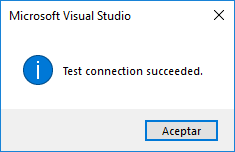


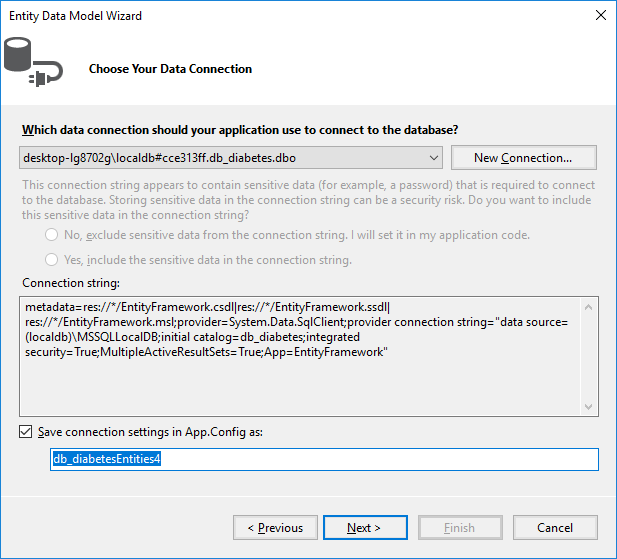
Bien, finalmente, una vez tenemos el nombre ahora sí, vamos a crear la conexión.



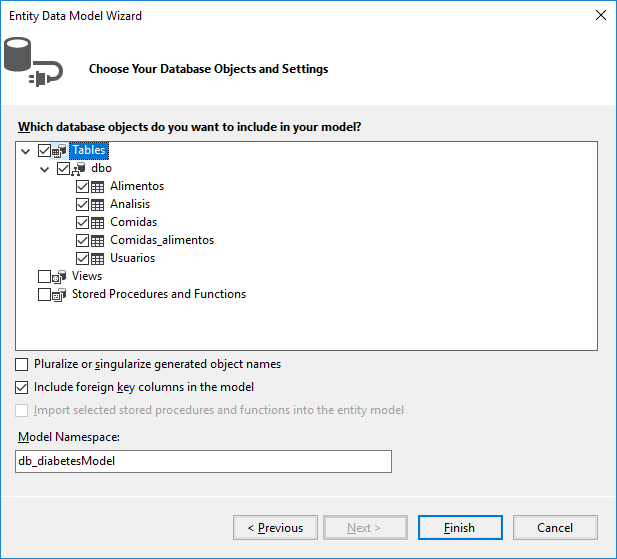


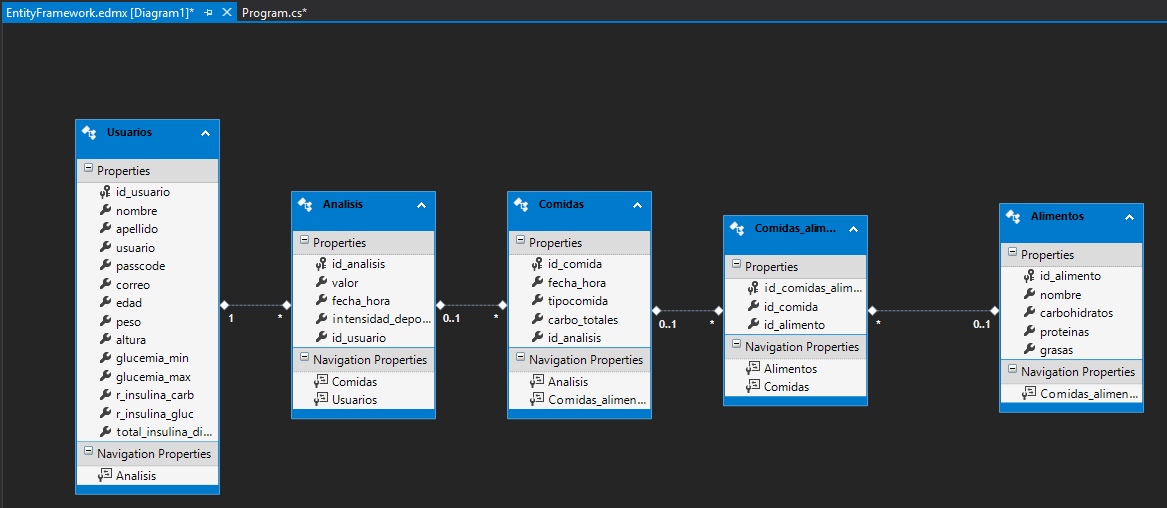
Seleccionamos la BBDD y testeamos la conexión.



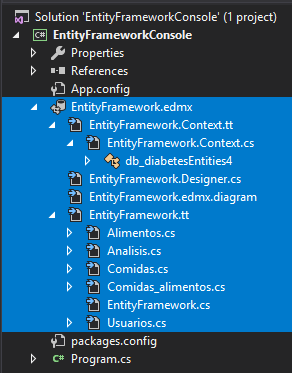


Esta conexión la necesitaremos posteriormente para poder realizar las query. Aunque no hace falta guardarla ya que como veremos antes de realizar las querys, se guarda en app.config.



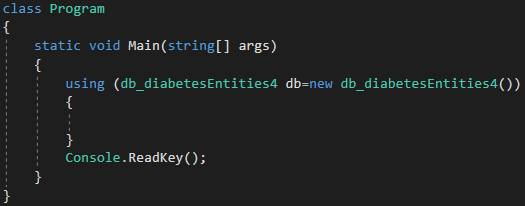


Si no nos aparecieran todas la tablas, posiblemente es que algunas de las tabla no tenga PK o que no sea autoincremental, que también puede dar problemas.



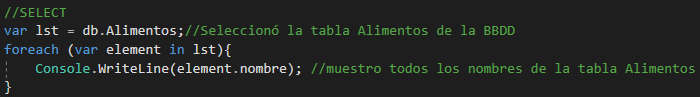
Dentro de la solución, tenemos db\_diabetesEntities4, que será lo que usaremos para realizar nuestra primera query.

Vamos a ello, para ello vamos al main y primeramente creamos una conexión mediante al using. Para ello, necesitamos el string del que hemos hablado anteriormente. Si no lo hemos guardado, no pasa nada ya que lo podemos encontrar dentro de EntityFramework > EntityFramework>db\_diabetesEntities4 (en mi caso al menos).

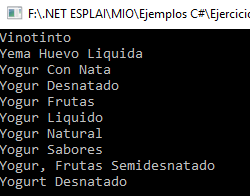


Bien, con esto realizamos la conexión, una vez realizada la conexión, empezamos con las queries.

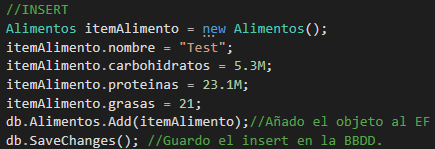
**1.SELECT**



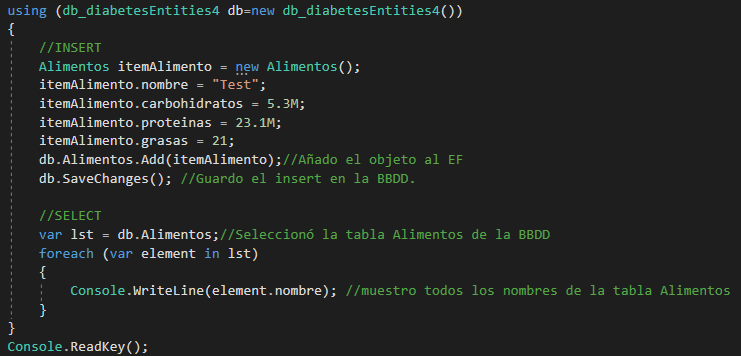
**Resultado:**



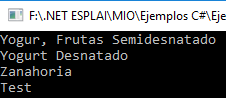
**2.INSERT**



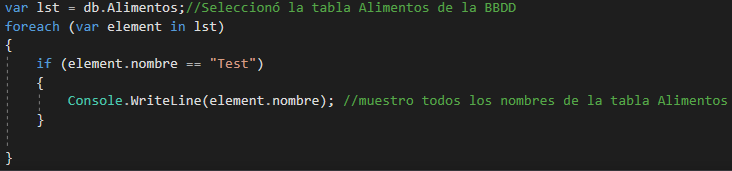
**3.INSERT + SELECT**



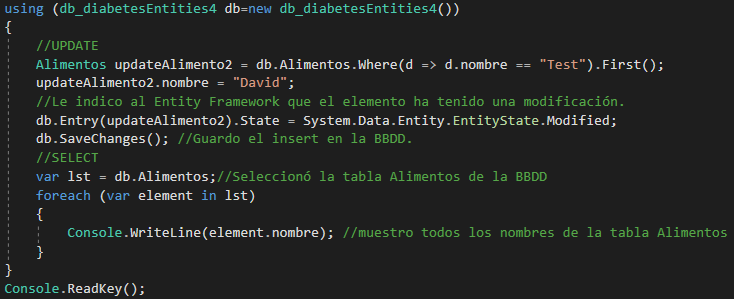
**Resultado:**

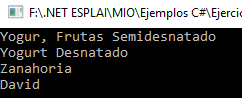


**4. SELECT DE LOS ALIMENTOS QUE SE LLAMEN TEST**

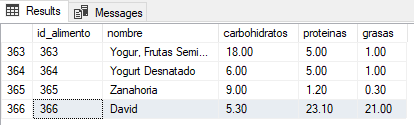


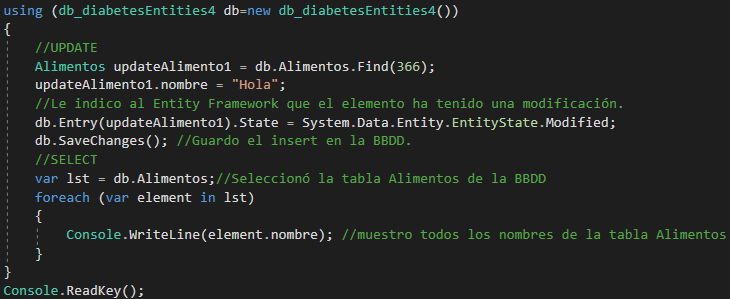
**5. UPDATE CON WHERE + SELECT**

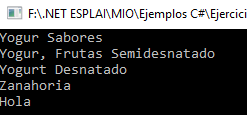




**5. UPDATE CON FIND + SELECT**







**6. DELETE (CRUD)**

