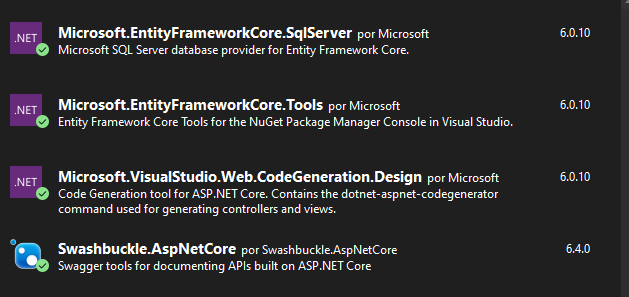
**API BASICA**

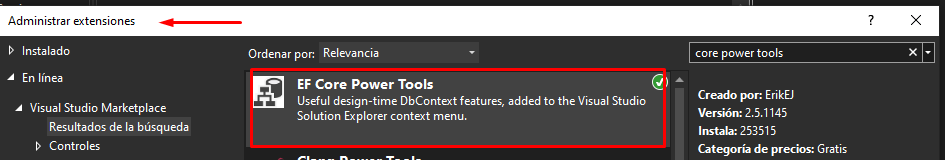
1 – **Conexión String** en appsettings.json



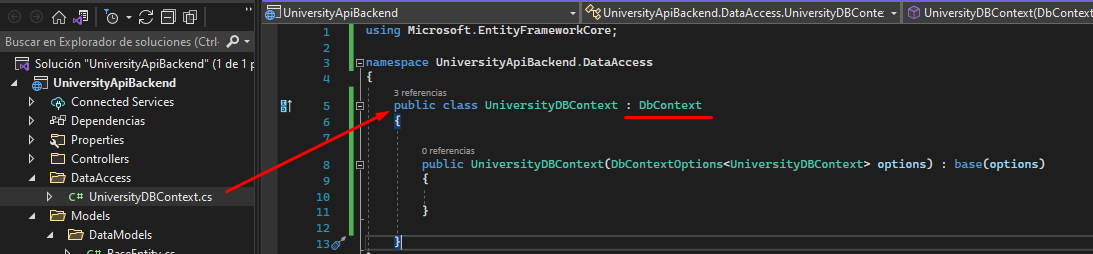
**2 – Instalamos Nuggets y extensiones**



\*La de CodeGeneration.Design sirve para crear CRUDS automáticamente con los endpoints necesarios.

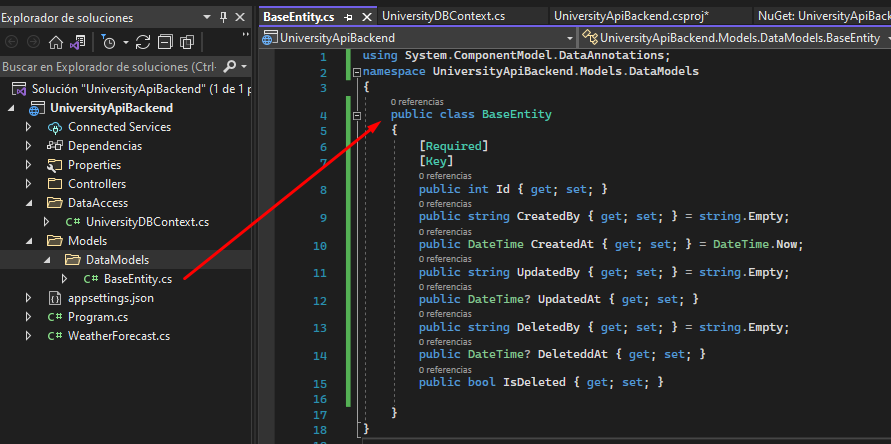


**3- Creamos nuestro contexto de la B.D.** en una nueva carpeta llamada DataAccess

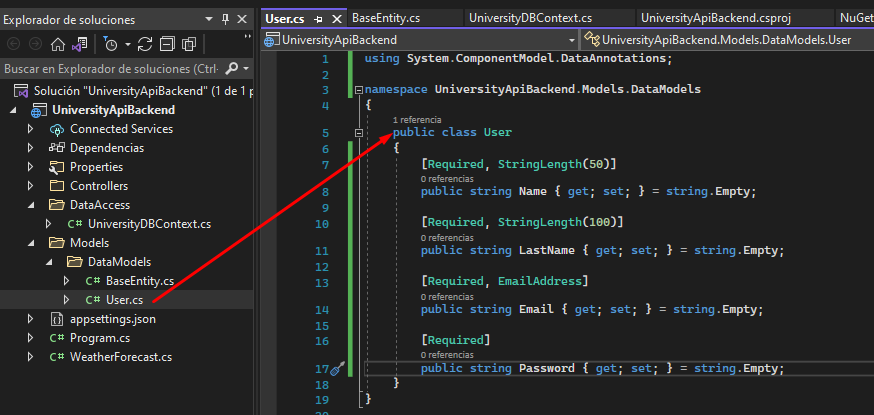


\*originalmente vacio, luego añadiremos los DBSet.

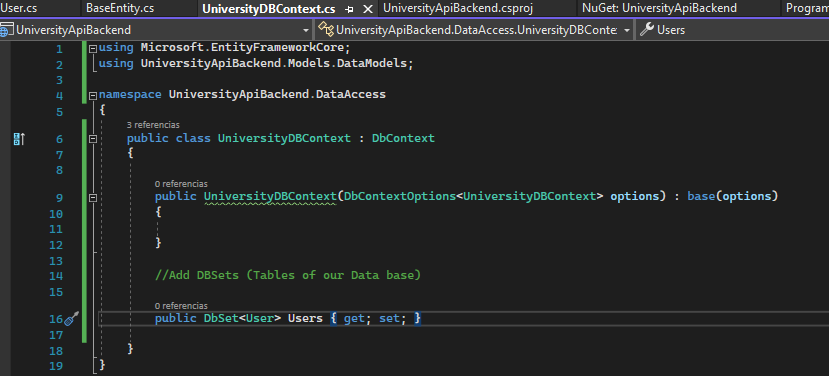
**4- Creamos una carpeta models/DataModels** y ahí creamos nuestro primer modelo (una entidad base con los campos mínimos que deben contener el resto de modelos).



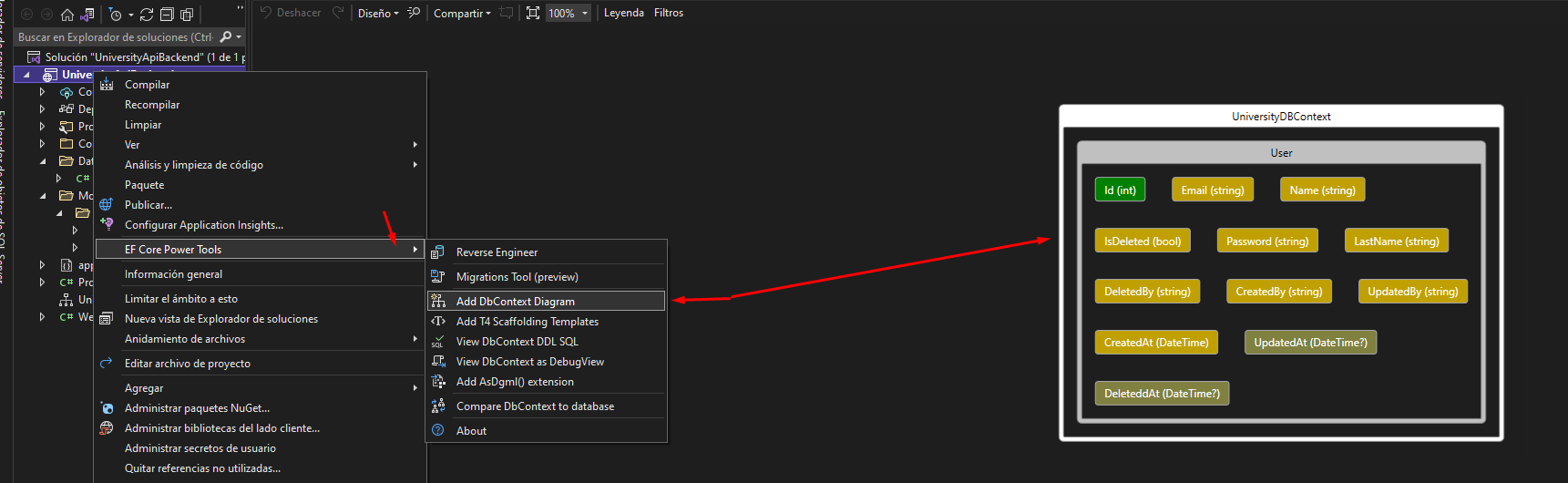
5 – Creamos la clase usuario que herede de la anterior



**6 – Agregamos DBContext**



Si usamos EF Core Power Tools, podemos ver y modificar el diagrama de la B.D. que se generaría



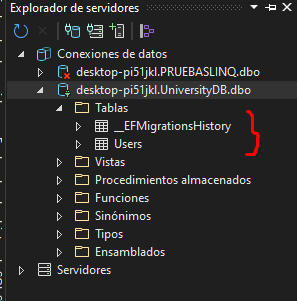
**7 - Desde la consola del administrador de paquetes, creamos la migración y agregamos la B.D.**

add-migration "Create Users Table"

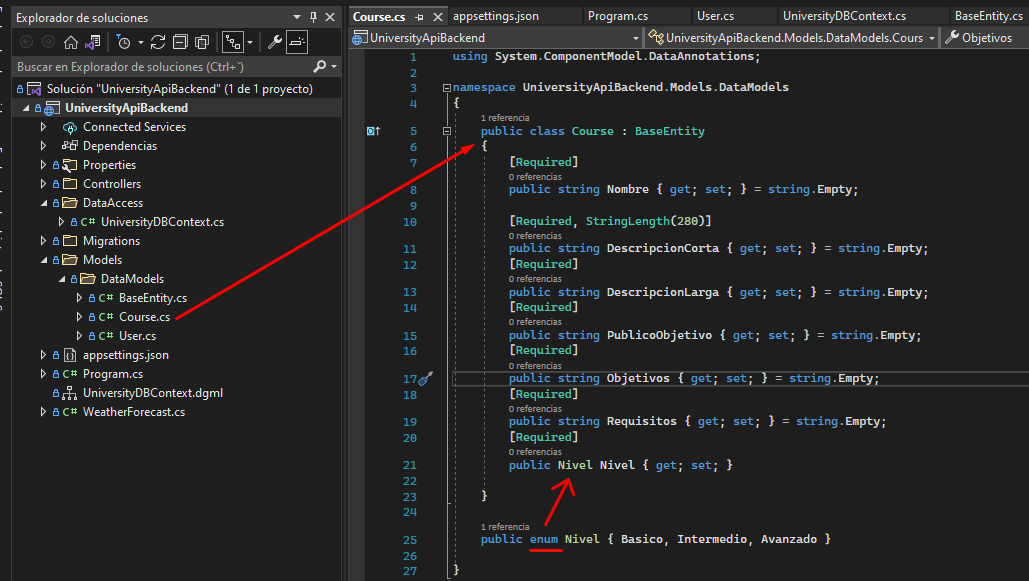
Update-Database –Verbose

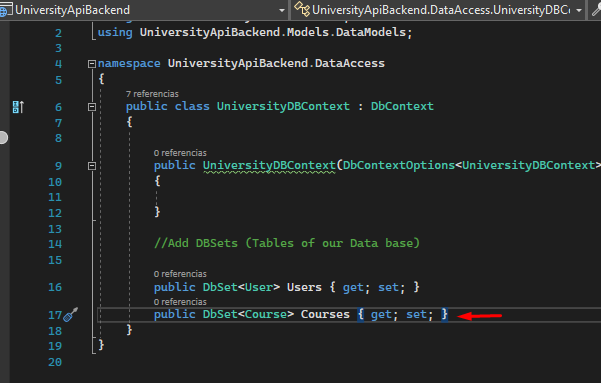
\*Con el comando “Script-Migration” podemos ver el script que se va a generar en la B.D.

\*Desde el explorador de servidores podemos conectarnos a nuestras B.D. para comprobar que todo está correcto

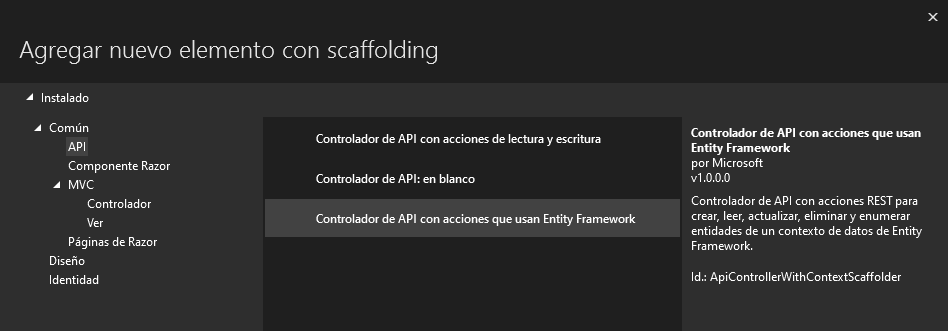


-Hacemos lo mismo para Cursos

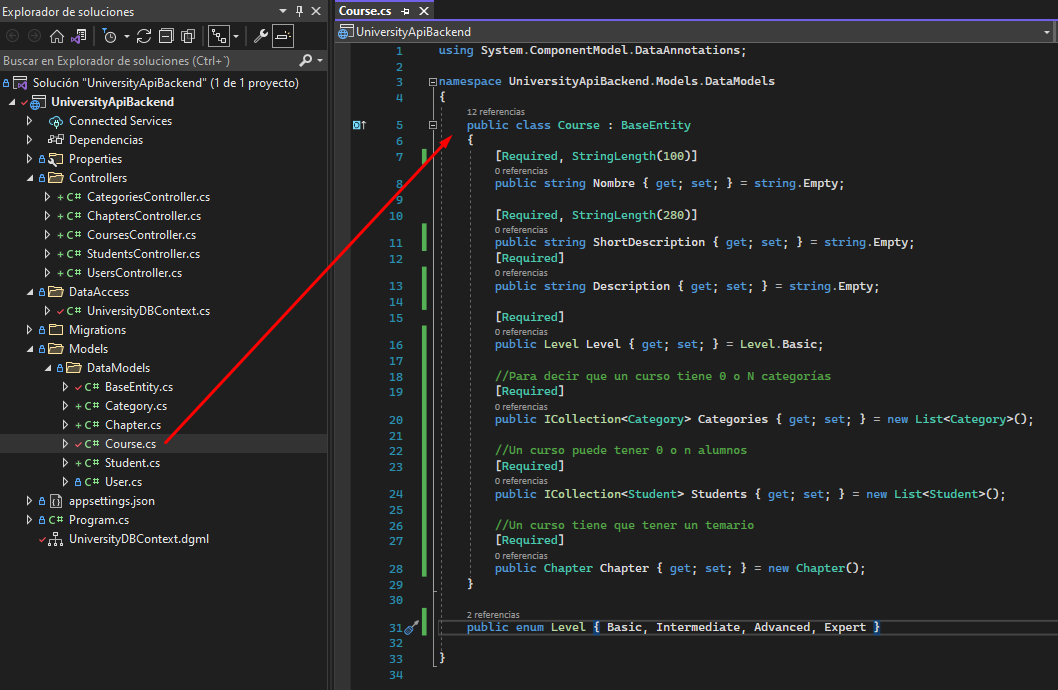


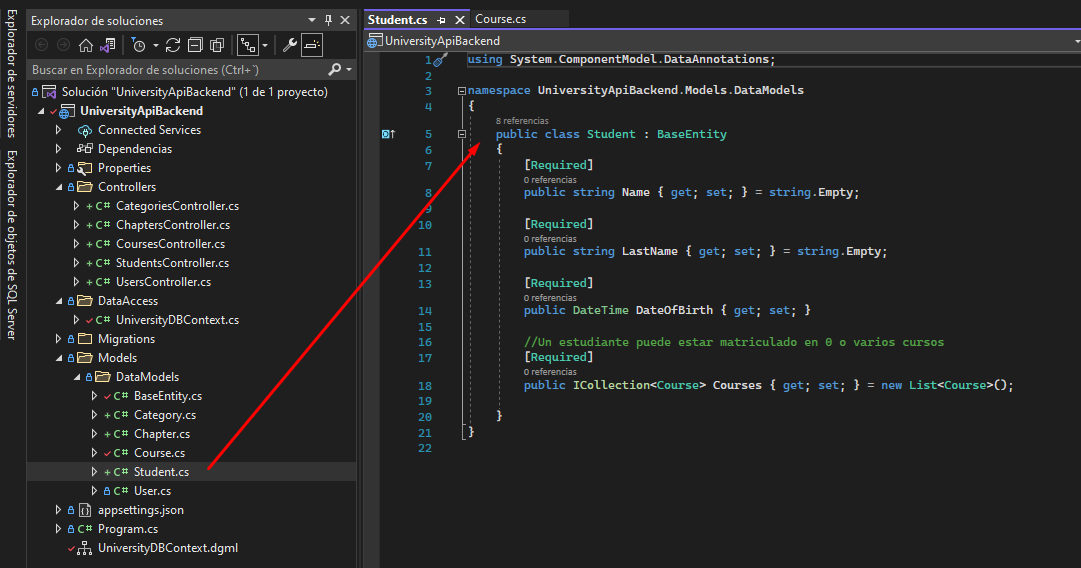


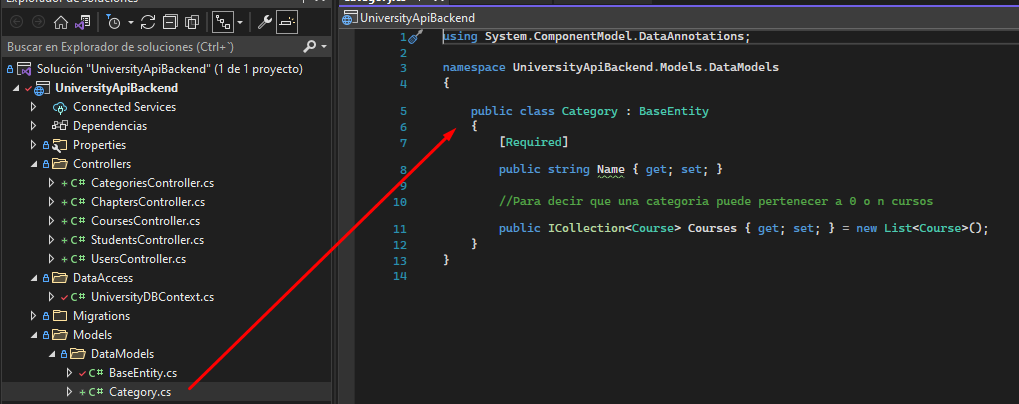
-Agregamos controlador con Scafolding

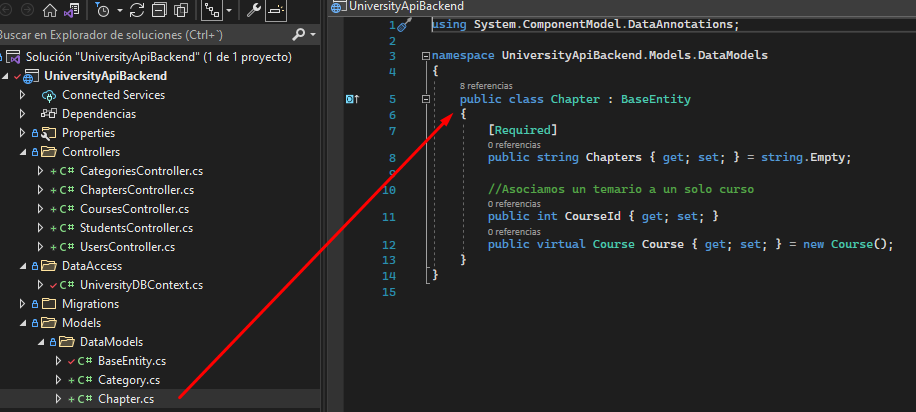


**8- Ahora iremos complicando con nuevas clases y relaciones entre ellas:**

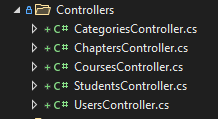




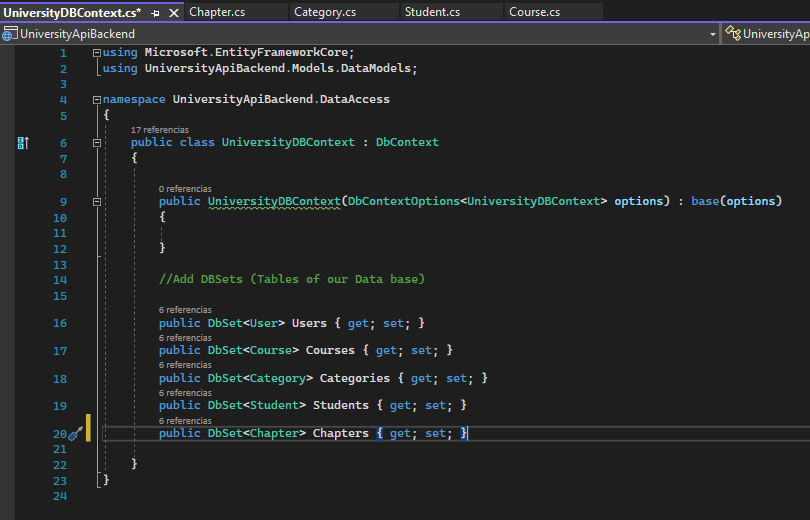




Y creamos con scalfolding sus respectivos controladores



Agregamos sus contextos a la B.D.

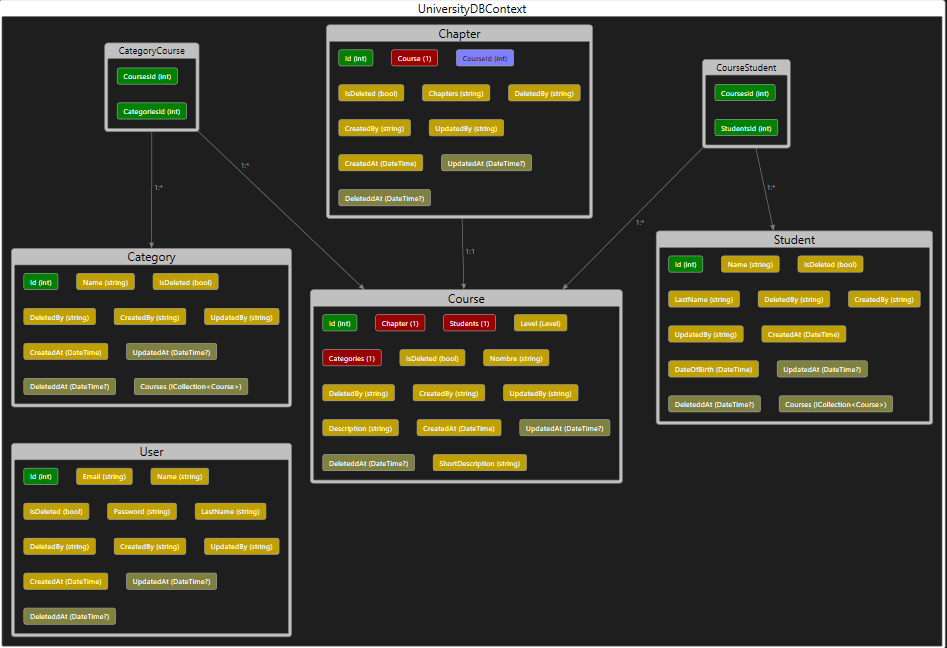


Y hacemos una migración

Add-Migration “Actualizando DB”

Update-Database –Verbose

9 – Actualmente el diagrama de la B.D. quedaría como sigue:



10- Ahora **crearemos nuestra propia biblioteca de clases usando LINQ** que nos permitan hacer consultas más avanzadas de manera recurrente.