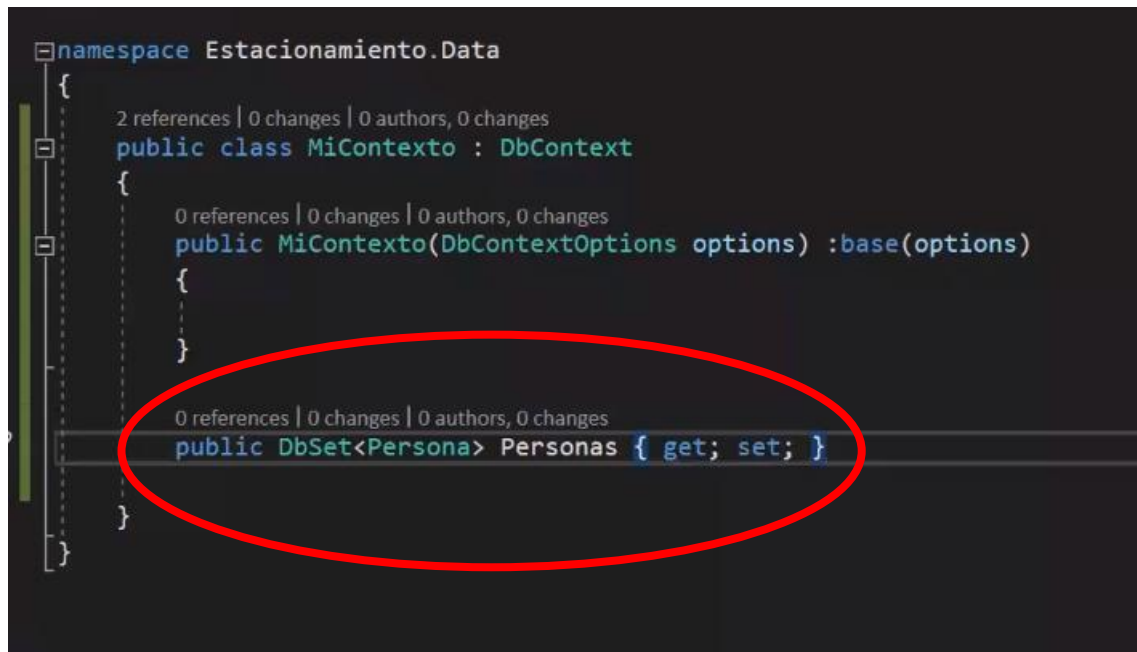


Creamos un DbSet para cada entidad y le decimos a EF canalizá las definiciones de los modelos, en este caso Persona y fijate en la DB que es lo que tenés. Todos los cambios que haya reflejalos en la DB. Eso se llama **MIGRACIÓN**. Partimos del código y “migramos” esas definiciones a otras en la DB

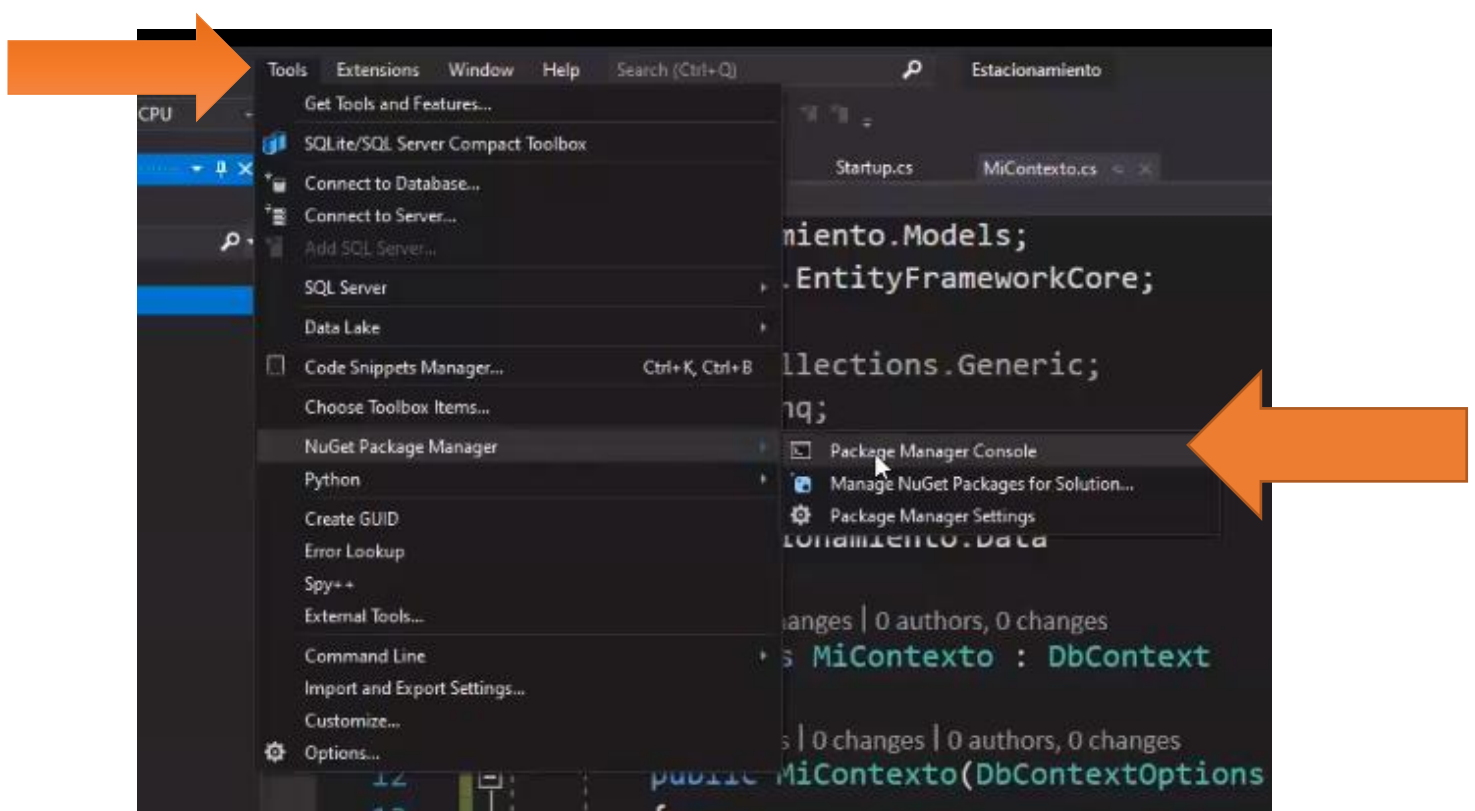
Primero creamos un DbSet dentro del contexto (siempre en plural)



```
namespace Estacionamiento.Data
{
    2 references | 0 changes | 0 authors, 0 changes
    public class MiContexto : DbContext
    {
        0 references | 0 changes | 0 authors, 0 changes
        public MiContexto(DbContextOptions options) :base(options)
        {
        }

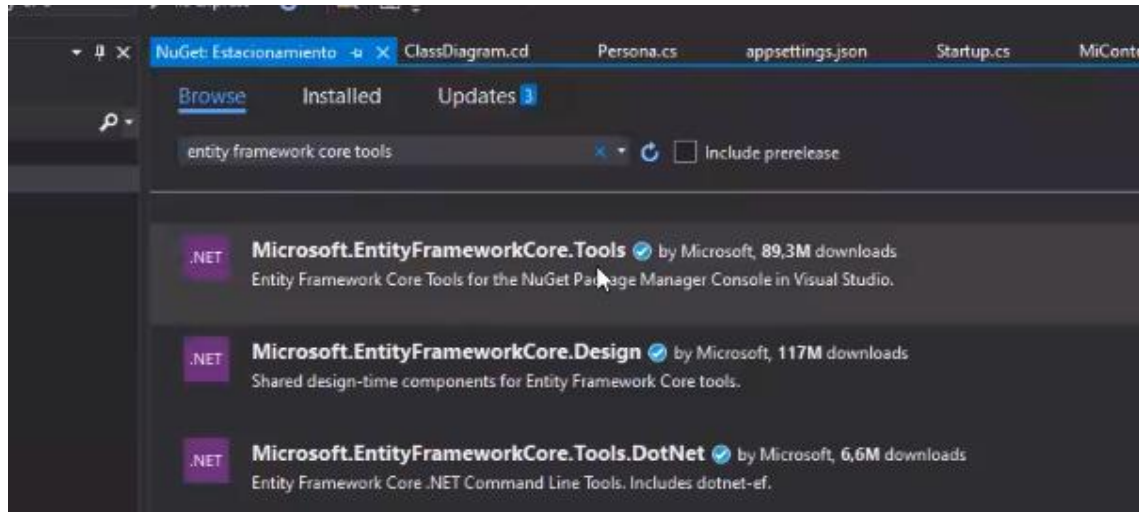
        0 references | 0 changes | 0 authors, 0 changes
        public DbSet<Persona> Personas { get; set; }
    }
}
```

La migración vamos a realizar a través de la **consola** que se encuentra acá:



En este paso primero hay que agregar las **EF Core Tools** si no las tenemos instaladas (es decir si hacemos add-migration en la consola y no lo conoce)

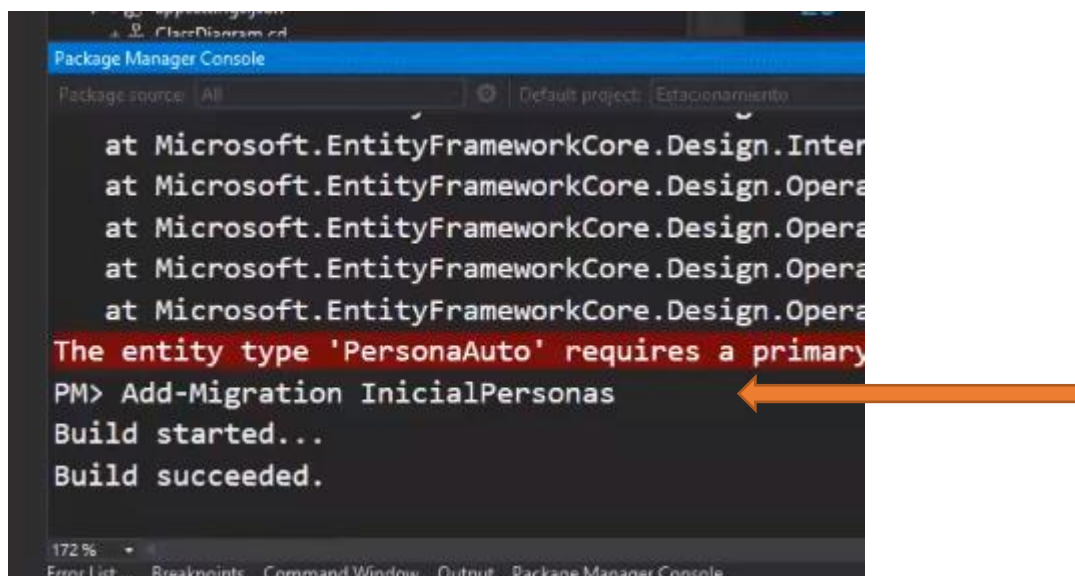
Siempre la version 3.1.14



La migración lo que va a hacer es analizar el código, verifica en la base de datos los delta, es decir las diferencias (en una primera instancia la creación de cero) y luego traduce esos cambios a lenguaje SQL.

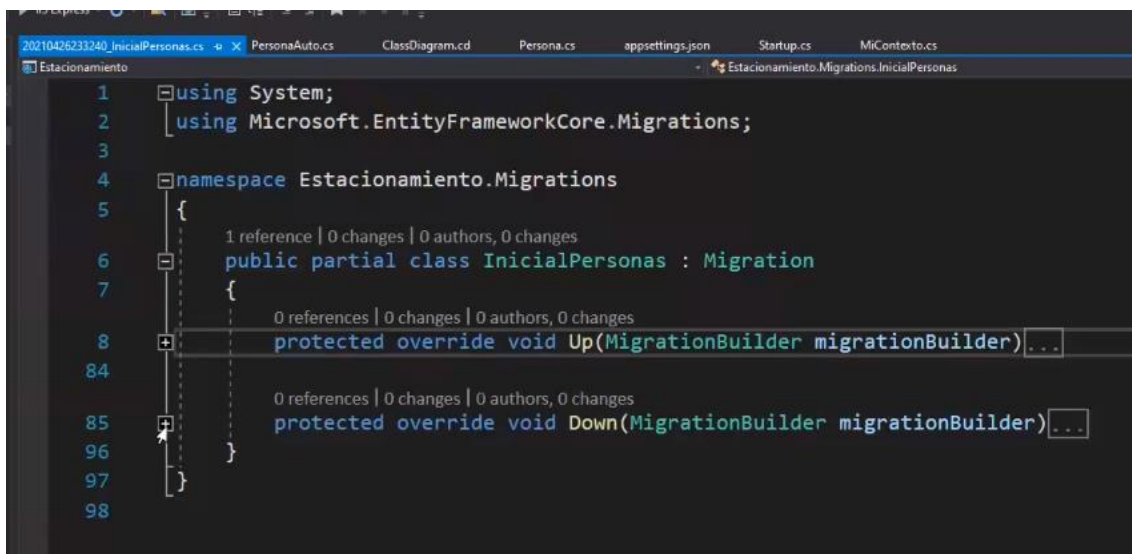
Se realiza con el comando **Add-Migration <nombre>**

El nombre es a elección, puede describir una modificación



Importante revisar la resultante para ver si hizo lo que nosotros queríamos antes de hacer el UPDATE.

En la carpeta **Migrations** crea la clase con los métodos UP y DOWN



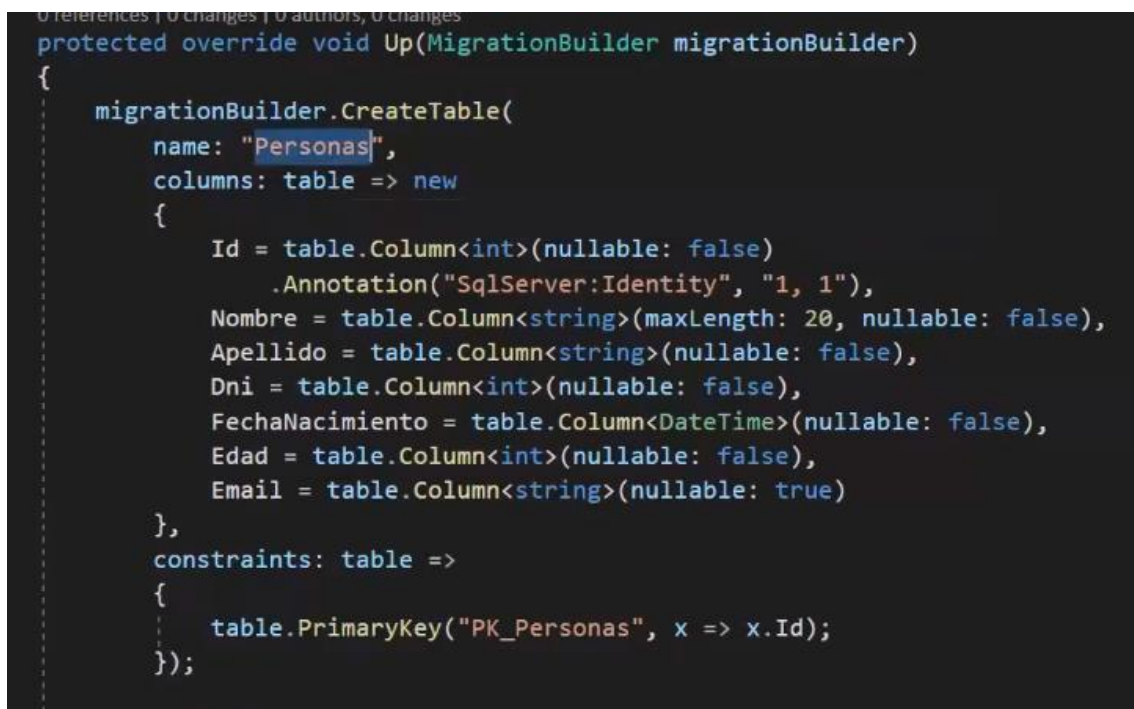
```
1 using System;
2 using Microsoft.EntityFrameworkCore.Migrations;
3
4 namespace Estacionamiento.Migrations
5 {
6     1 reference | 0 changes | 0 authors, 0 changes
7     public partial class InicialPersonas : Migration
8     {
9         0 references | 0 changes | 0 authors, 0 changes
10        protected override void Up(MigrationBuilder migrationBuilder) ...
11
12        0 references | 0 changes | 0 authors, 0 changes
13        protected override void Down(MigrationBuilder migrationBuilder) ...
14    }
15 }
```

UP es lo que tengo que ejecutar para subir hacia la DB los cambios de mi código.

DOWN es para volver atrás

Vemos el método UP. Crea una tabla con el nombre de la Propiedad que le definí en el contexto (DbSet<Persona> **Personas**) y las columnas de acuerdo a las propiedades que definimos en la Entidad.

Nulleable false, es que es required.



```
0 references | 0 changes | 0 authors, 0 changes
protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.CreateTable(
        name: "Personas",
        columns: table => new
        {
            Id = table.Column<int>(nullable: false)
                .Annotation("SqlServer:Identity", "1, 1"),
            Nombre = table.Column<string>(maxLength: 20, nullable: false),
            Apellido = table.Column<string>(nullable: false),
            Dni = table.Column<int>(nullable: false),
            FechaNacimiento = table.Column<DateTime>(nullable: false),
            Edad = table.Column<int>(nullable: false),
            Email = table.Column<string>(nullable: true)
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_Personas", x => x.Id);
        });
}
```

Para definir la primary Key, identificador único, podemos delegar la responsabilidad a SQL Server

```
Id = table.Column<int>(nullable: false)  
    .Annotation("SqlServer:Identity", "1, 1"),  
Nombre = table.Column<string>(maxLength: 255, nullable: true)
```

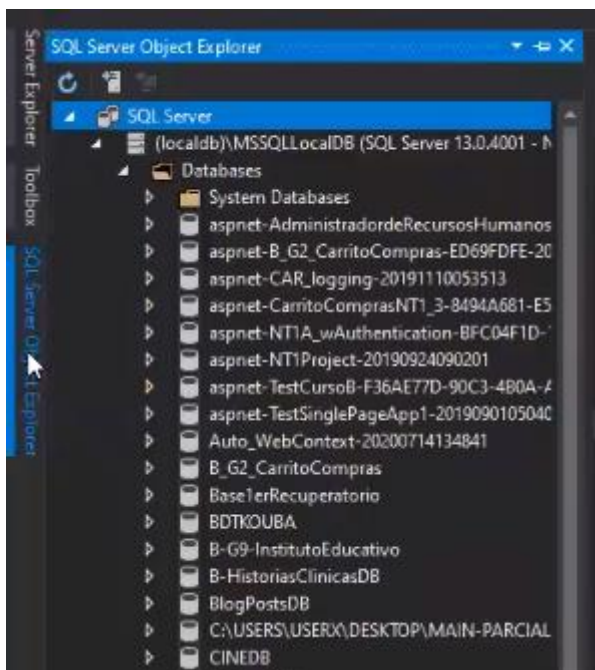
La configuración por defecto es 1,1. Se interpreta que inicia en 1 y los saltos sumando de a 1 para el siguiente identificador. La primera persona va a tener el Id 1 y la segunda el 2, etc. **(NO RECOMENDADO)**

Una vez que se realizó exitosamente la migración, hay que ejecutar el **Update-Database** para impactar la migration y realizar los cambios.

```
PM> Update-Database  
Build started...  
Build succeeded.  
Done.  
PM> |
```

---

Para consultar la DB desde el Visual Studio hacemos **view > SQL Server Object Explorer** y la busco con el nombre que definimos en el archivo JSON de configuración. (si no sale, hacer un refresh arriba izquierda)





## NUNCA TOCAR la que se llama dbo.\_EFMigrationsHistory

Se pueden revisar las tablas con un formato más amigable ingresando por el SQL Server Explorer, donde figuran las tablas, con las columnas y los detalles.

The screenshot displays the Visual Studio IDE with the following components:

- SQL Server Object Explorer:** Shows the database structure. Under 'EstacionamientoDB-D', the 'dbo.Personas' table is selected. The 'Columns' list shows: Id (PK, int, not null), Nombre (nvarchar(20), not null), Apellido (nvarchar(max), not null), Dni (int, not null), FechaNacimiento (datetime2(7), not null), Edad (int, not null), and Email (nvarchar(max), null).
- Solution Explorer:** Shows the project 'Estacionamiento' with folders for Connected Services, Dependencies, Properties, wwwroot, Controllers, Data, Documentos, and Migrations. The 'Migrations' folder contains a file named '20210426233240\_InicialPersonas.cs'.
- SQL Server Enterprise Designer:** The 'dbo.Personas [Design]' view is open, showing a table with the following columns and data types:

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
Nombre	nvarchar(20)	<input type="checkbox"/>	
Apellido	nvarchar(MAX)	<input type="checkbox"/>	
Dni	int	<input type="checkbox"/>	
FechaNacimiento	datetime2(7)	<input type="checkbox"/>	
Edad	int	<input type="checkbox"/>	
Email	nvarchar(MAX)	<input checked="" type="checkbox"/>	
- SQL Script:** The 'T-SQL' view shows the CREATE TABLE statement for 'dbo.Personas':

```
1 CREATE TABLE [dbo].[Personas] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nombre] NVARCHAR (20) NOT NULL,  
4     [Apellido] NVARCHAR (MAX) NOT NULL,  
5     [Dni] INT NOT NULL,  
6     [FechaNacimiento] DATETIME2 (7) NOT NULL,  
7     [Edad] INT NOT NULL,  
8     [Email] NVARCHAR (MAX) NULL,  
9     CONSTRAINT [PK_Personas] PRIMARY KEY CLUSTERED ([Id] ASC)  
10 );  
11
```

Para revisar los datos que está guardando la tabla, podemos hacer botón derecho y **View Data**

Para arrancar de cero, botón derecho sobre la DB, **Delete** y tildar "*Close existing connections*". Luego borrar la carpeta **Migrations**. Arreglar las cosas y arrancar de nuevo.

---