C# + NET + VSC + Entity Framework

Pomelo + MySql

ITR – Dic 2023

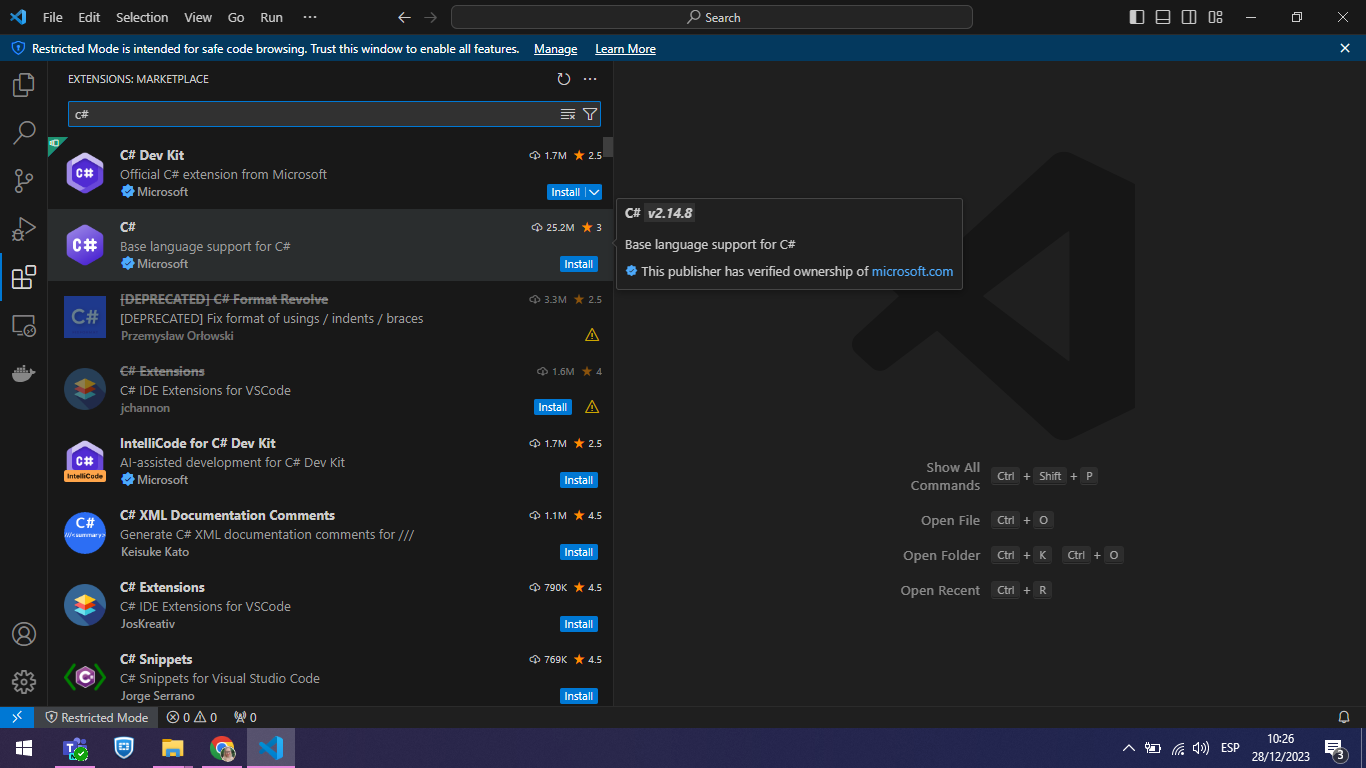
Marina Celeste Chávez

**Entity Framework – C# - VSC – Pomelo - MySql**

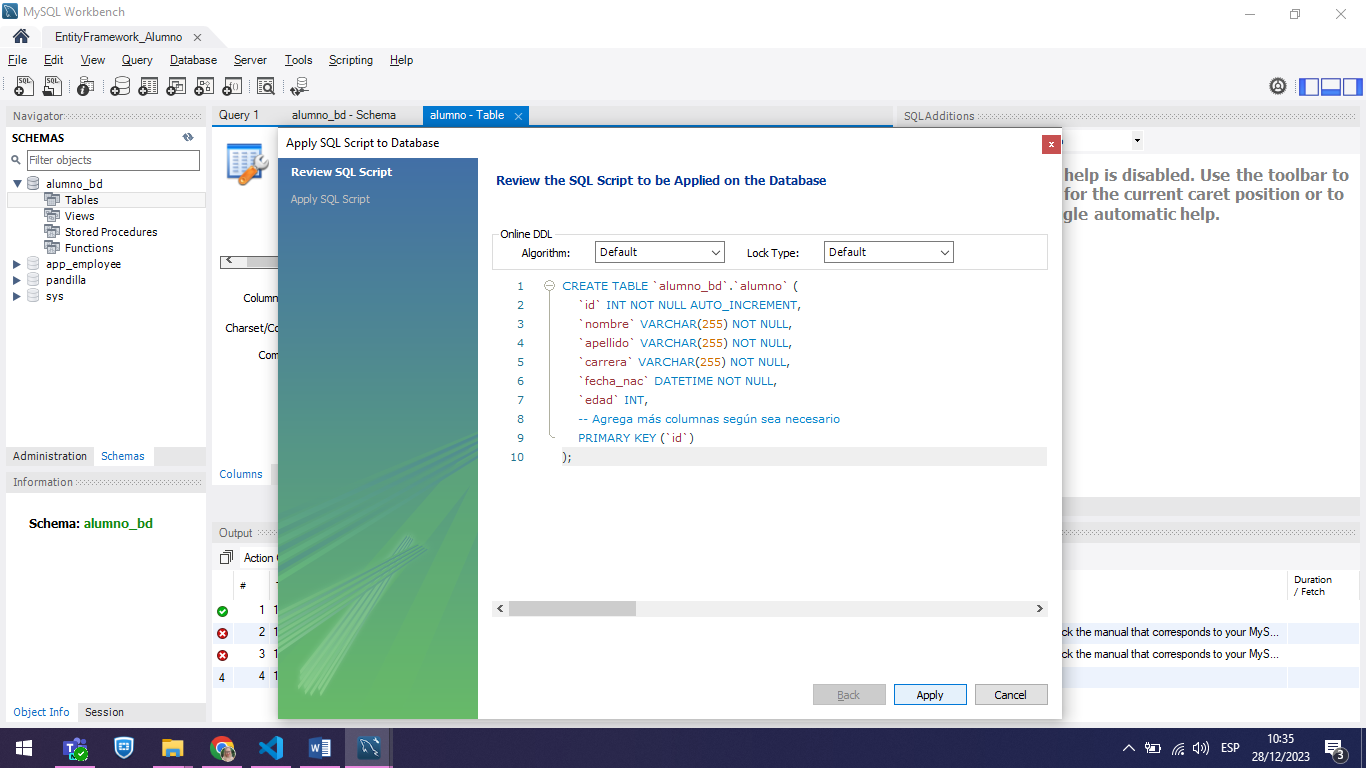
Vamos a comenzar con los primeros pasos para generar una base de datos de ejemplo e instalación del software necesario para poder usar Entity Framework C# + VSC y Pomelo para el dbcontext y poder generar las clases automáticas que posee nuestra base de datos al realizar la conexión.

Comenzamos instalando el plugin básico, pero a medida que avancemos el VSC nos ira diciendo que necesitamos instalar o recomendaciones.

Por otro lado, tener en cuenta las versiones. Primero instalé net 3 pero luego migré todas las versiones a Net 8 por sugerencia del uso en VSC.



Mientras se instala la extensión en VSC para usar C#, voy creando una tabla de ejemplo:



Es mejor usar Date para no tener que poner horario en la fecha, por lo tanto, edite con esta qwery la tabla alumno:

ALTER TABLE `alumno\_bd`.`alumno`

MODIFY COLUMN `fecha\_nac` DATE NOT NULL;

CREATE TABLE `alumno\_bd`.`alumno` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(255) NOT NULL,

`apellido` VARCHAR(255) NOT NULL,

`carrera` VARCHAR(255) NOT NULL,

`fecha\_nac` DATE NOT NULL,

`edad` INT,

PRIMARY KEY (`id`)

);

Dejo estas qwery de ejemplo por si se desea formar la DB:

use alumno\_bd;

INSERT INTO `alumno\_bd`.`alumno` (`nombre`, `apellido`, `carrera`, `fecha\_nac`, `edad`) VALUES

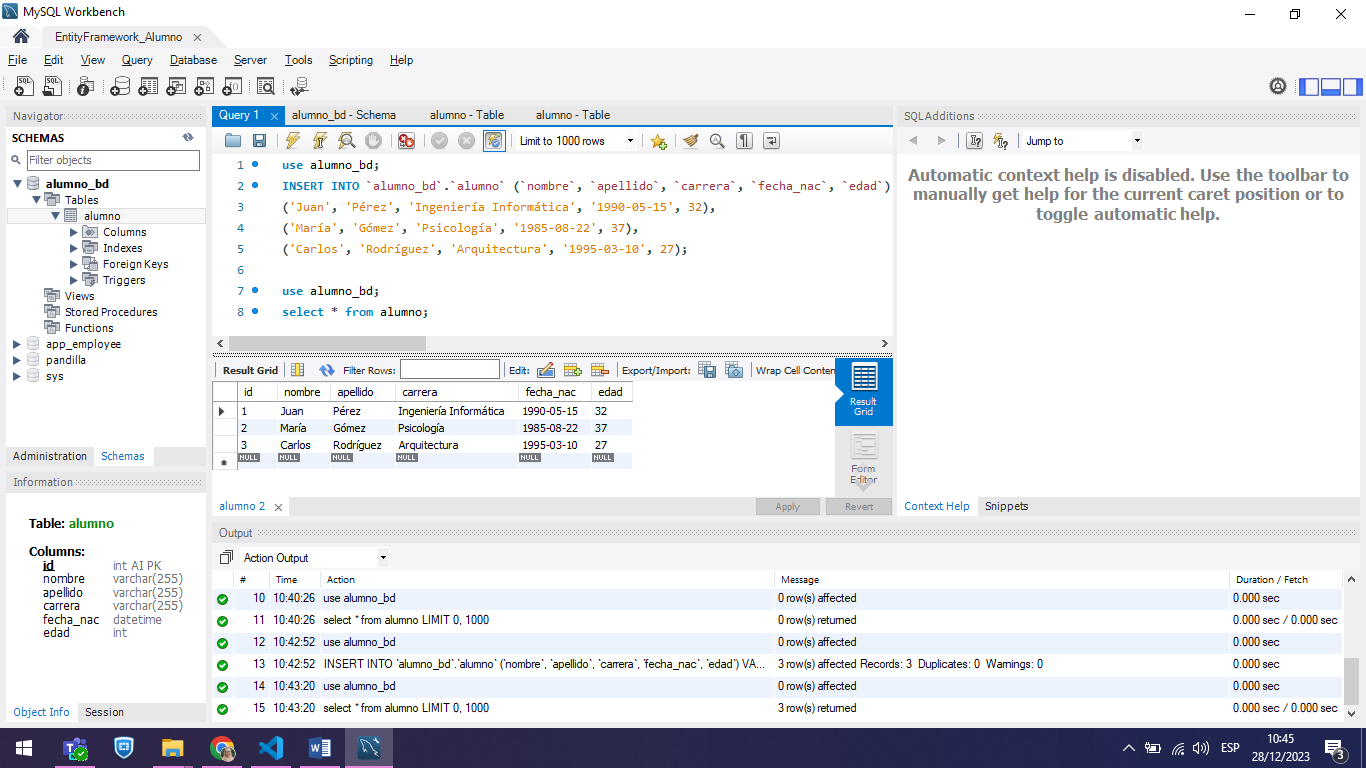
('Juan', 'Pérez', 'Ingeniería Informática', '1990-05-15', 32),

('María', 'Gómez', 'Psicología', '1985-08-22', 37),

('Carlos', 'Rodríguez', 'Arquitectura', '1995-03-10', 27);

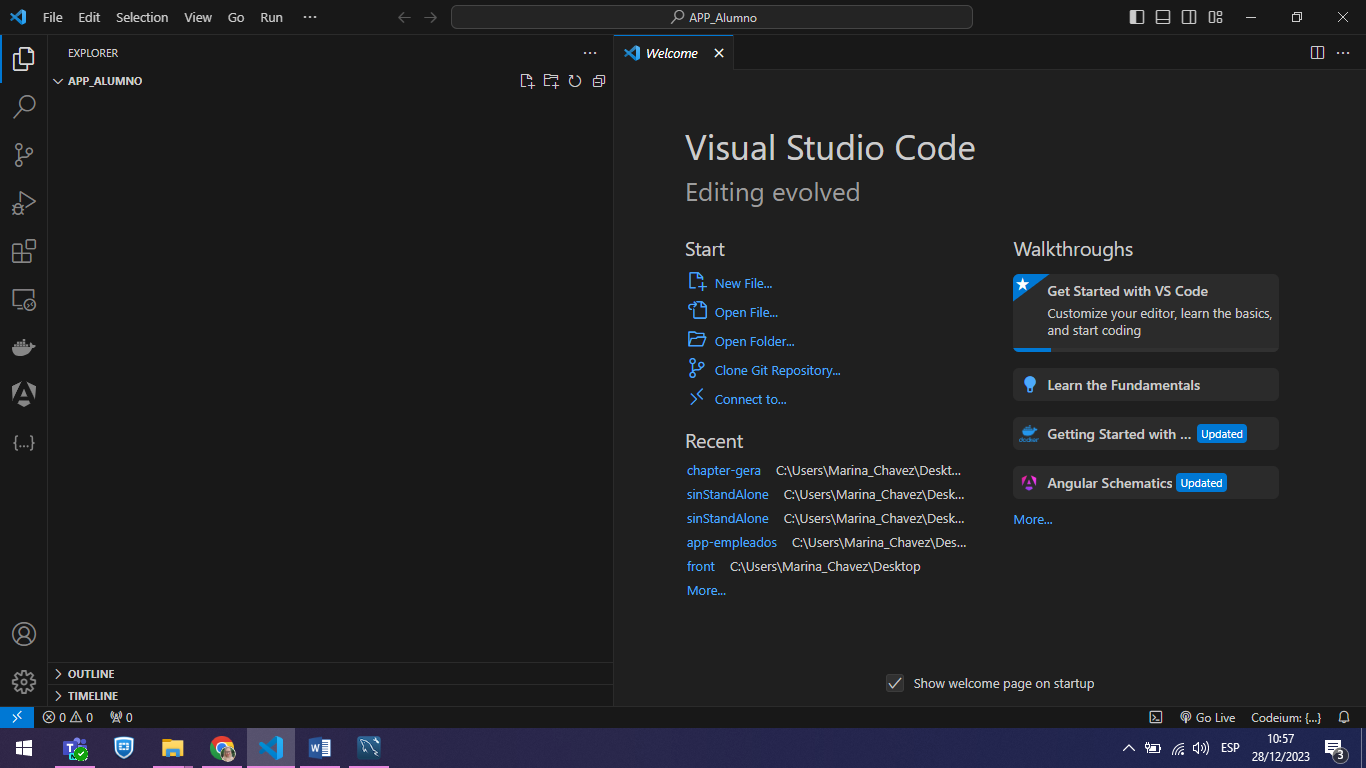
use alumno\_bd;

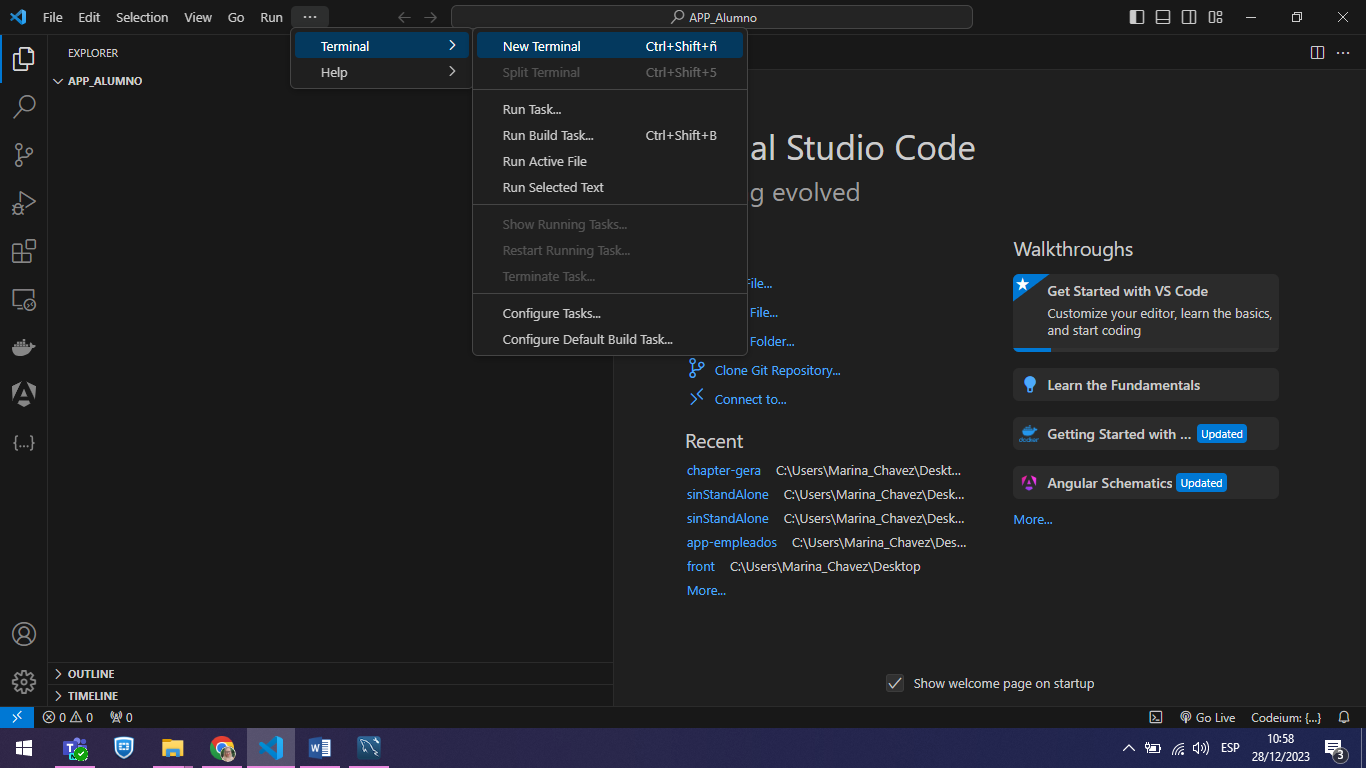
select \* from alumno;



Creo una carpeta llamada APP\_Alumno y la arrastro hasta el VSC para abrirla y comenzar:



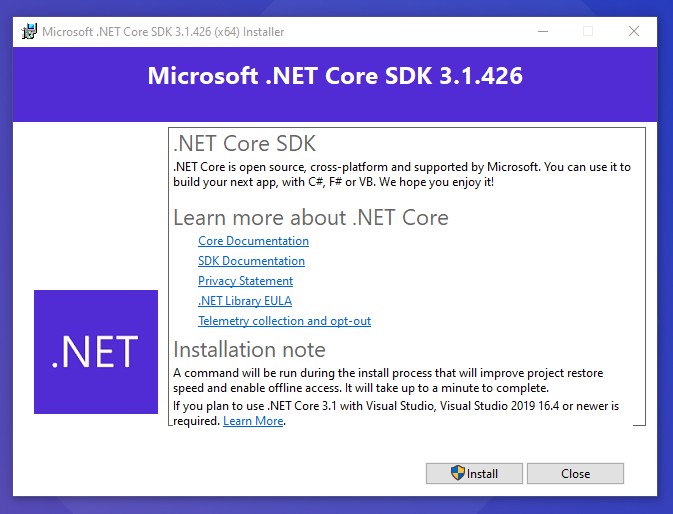


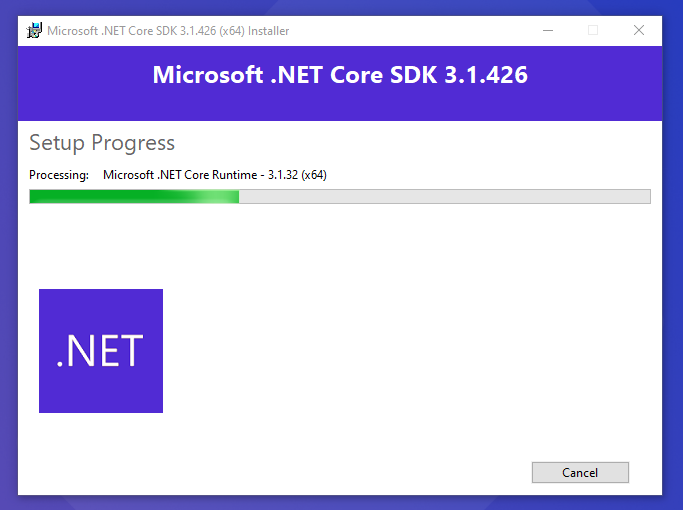


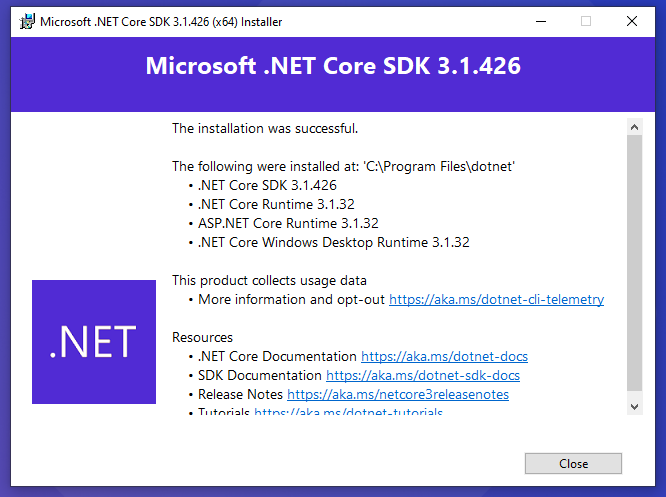
Para poder utilizar el comando a continuación debemos descargar Net Core:

<https://dotnet.microsoft.com/es-es/download/dotnet/3.1>

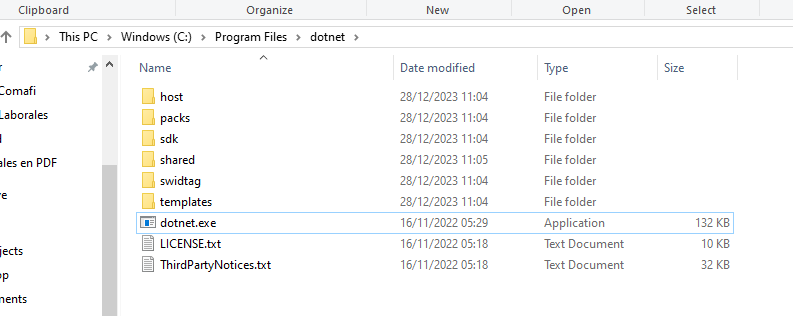
<https://dotnet.microsoft.com/es-es/download/dotnet/thank-you/sdk-3.1.426-windows-x64-installer>

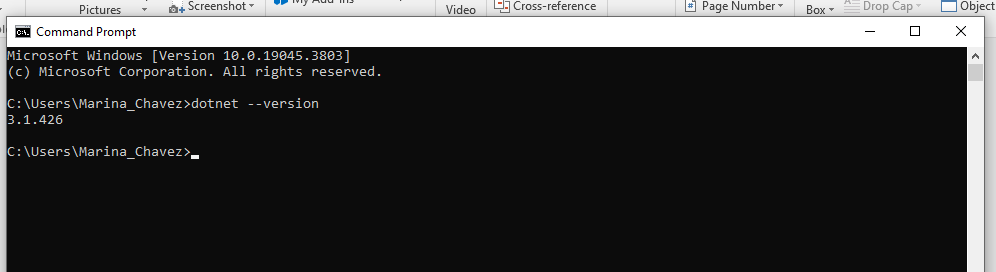




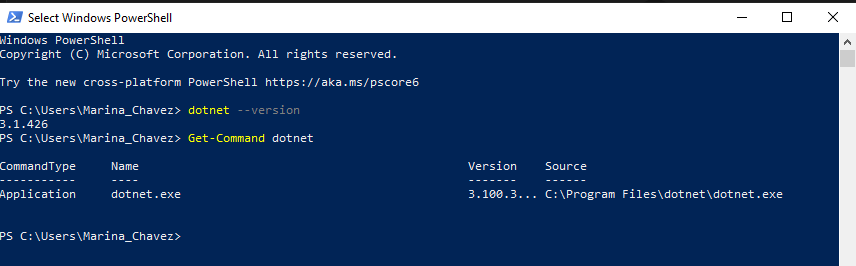


Chequemos en disco C que se haya instalado y en el cmd que ya tengamos la versión:

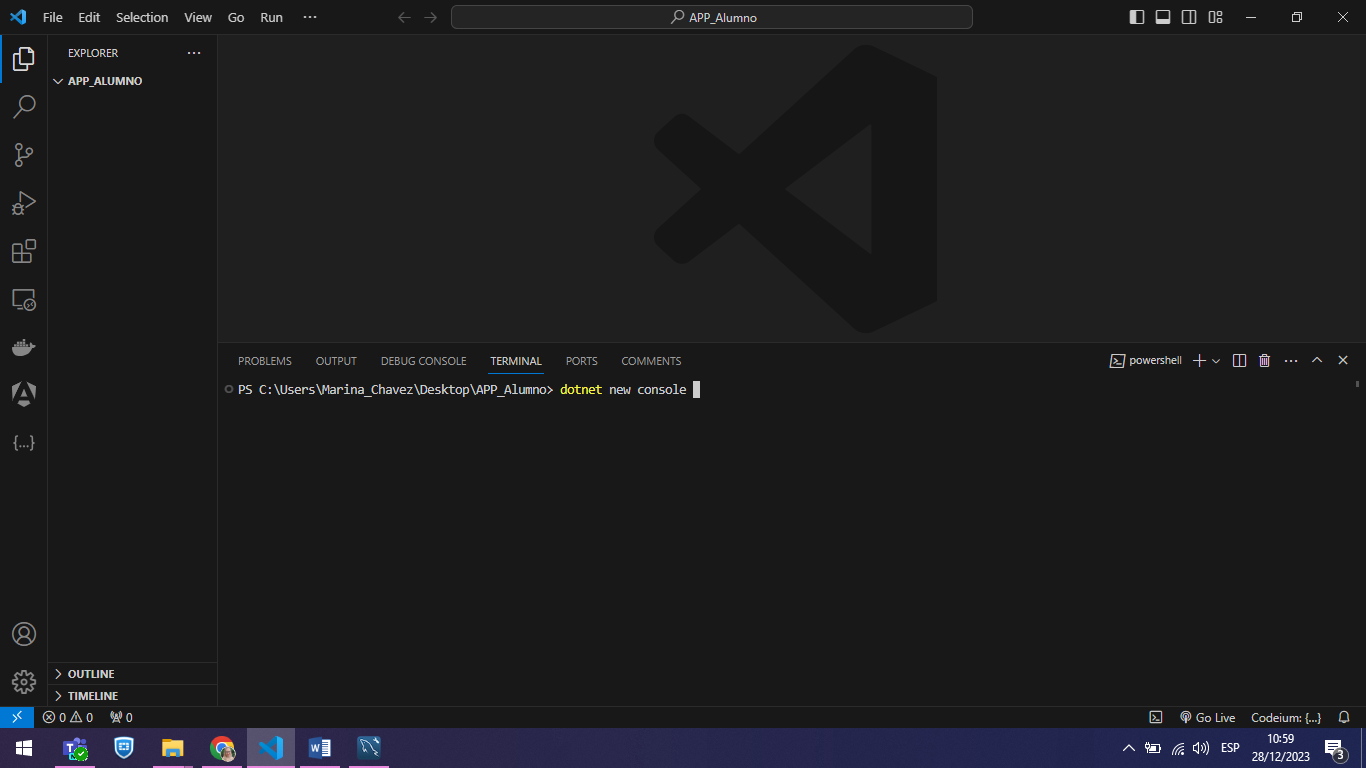


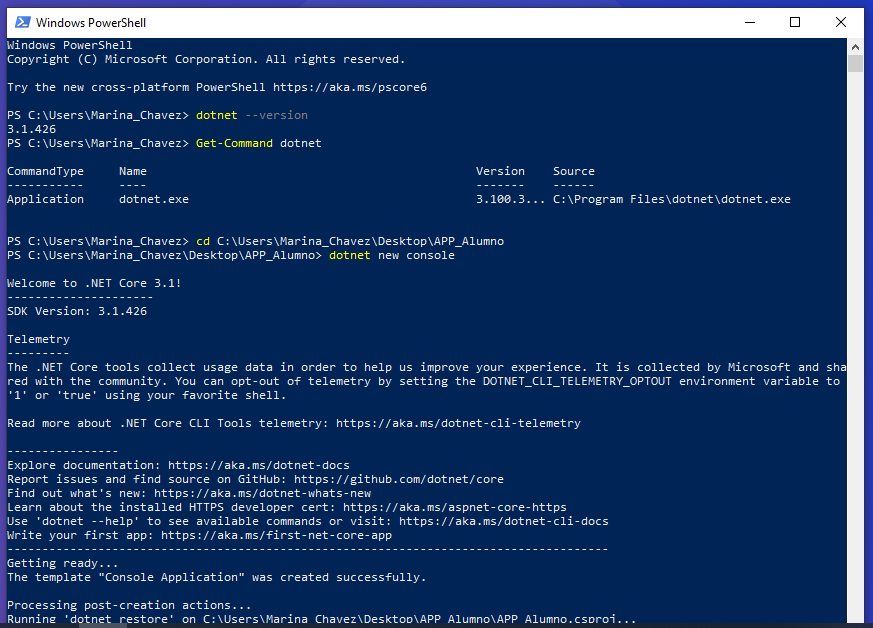


Si se presentan inconvenientes para reconocer el comando dotnet usar powershell:

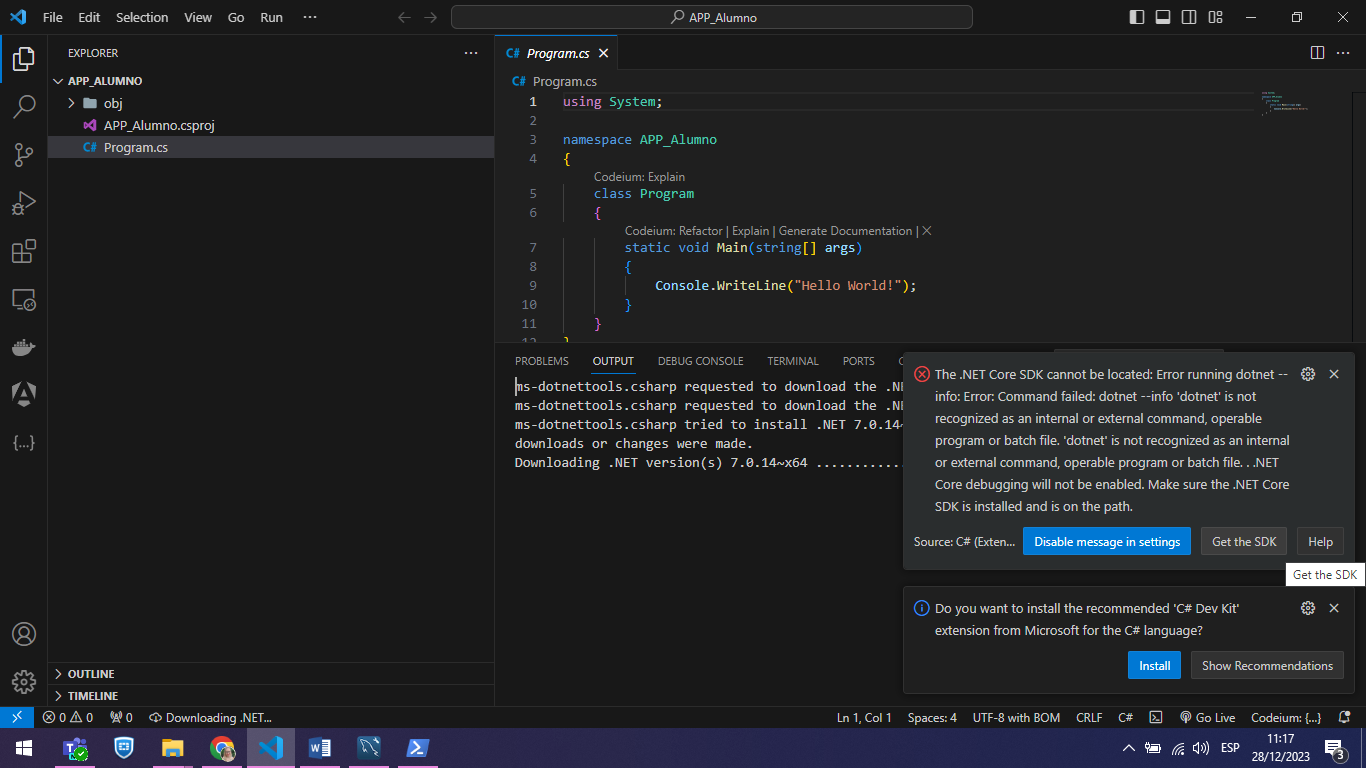


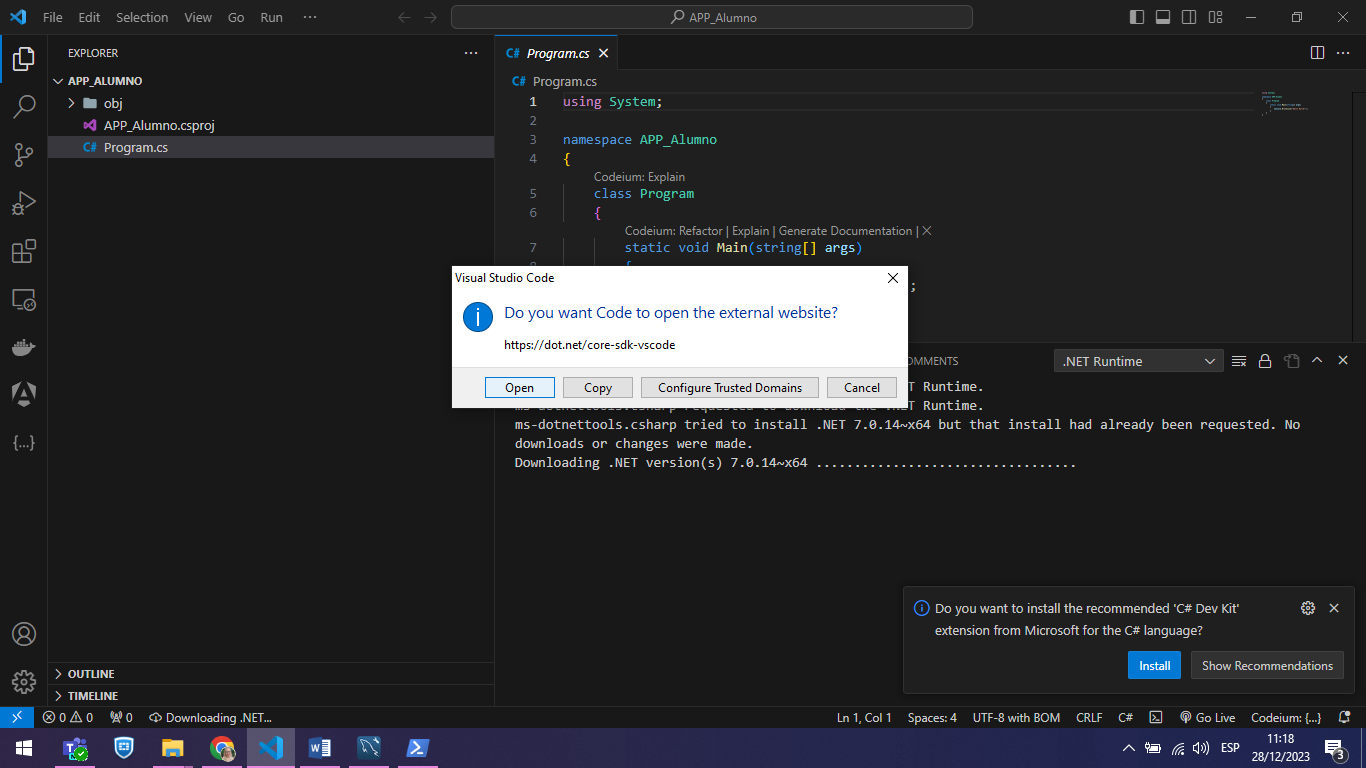
Proseguimos con la terminal del VSC en el directorio de nuestra nueva carpeta para comenzar:



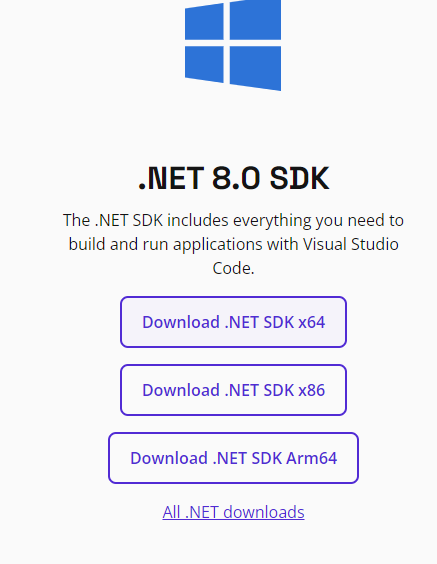


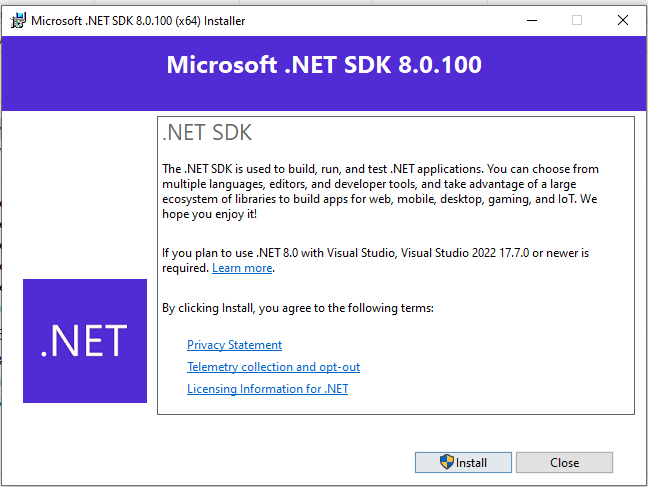
Si da error el dotnet en la terminal de VSC se puede usar powershell y lo reconoce.



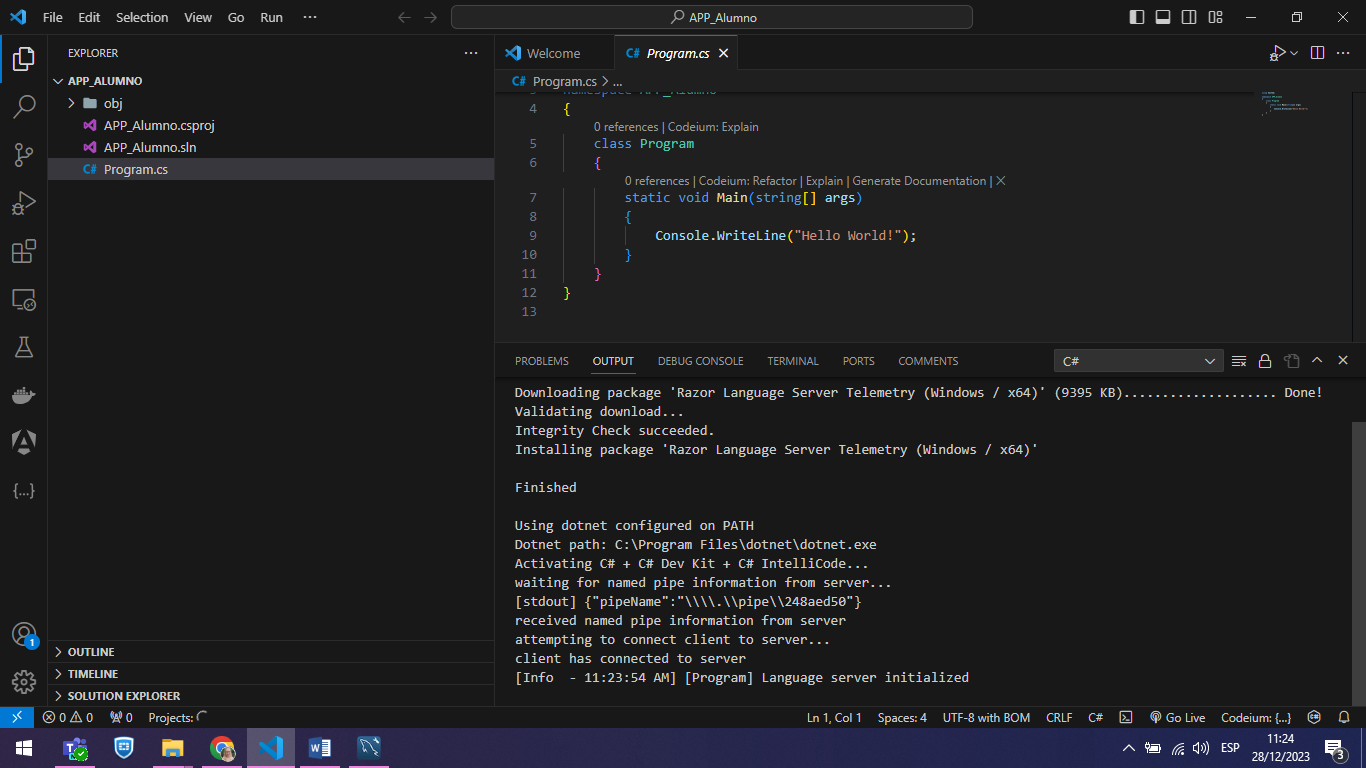


<https://dotnet.microsoft.com/en-us/download/dotnet/sdk-for-vs-code?utm_source=vs-code&amp;utm_medium=referral&amp;utm_campaign=sdk-install>





Al reiniciar VSC se va actualizando todo lo necesario:



Vamos a instalar EntityFramework con el siguiente comando:

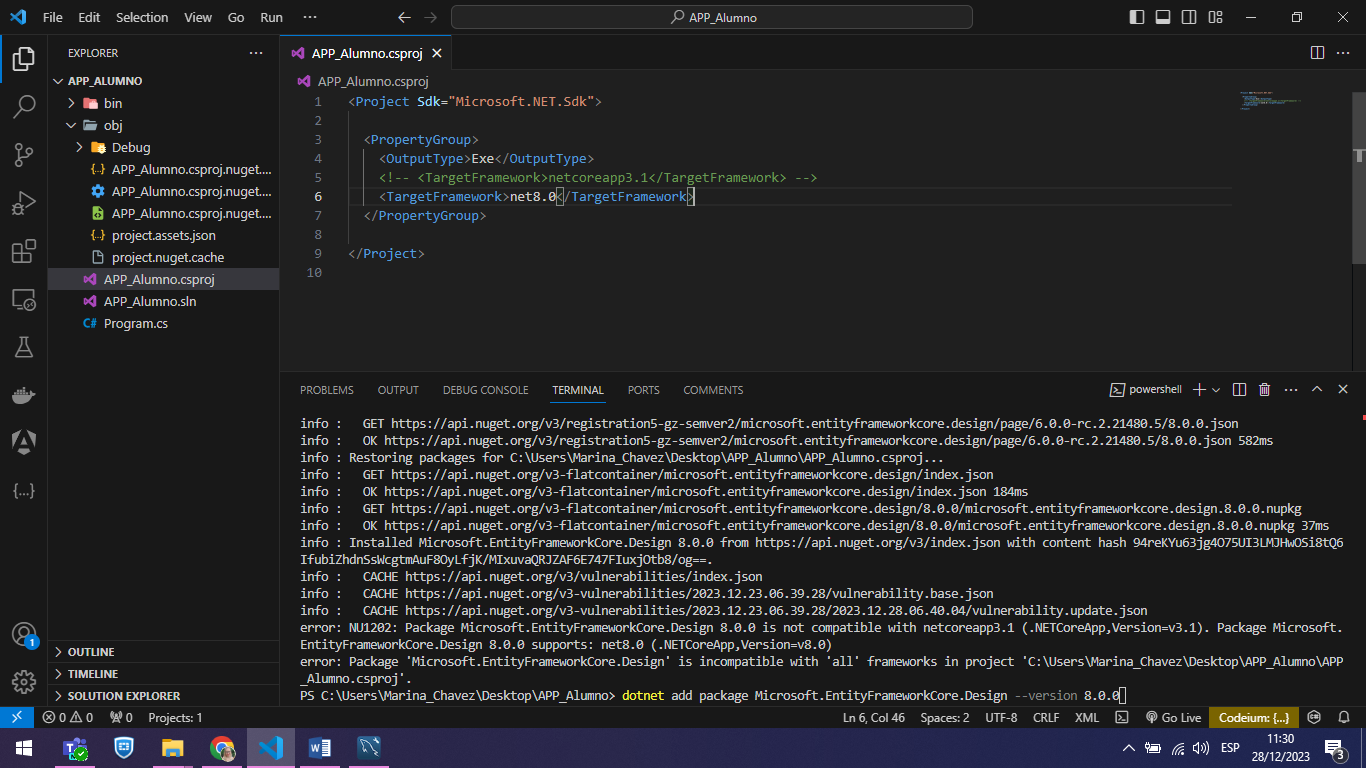
dotnet add package Microsoft.EntityFrameworkCore.Design

Tener en cuenta las versions. Edite la versión 3 por la 8 y ejecute nuevamente el comando diciendo que versión utilizar:

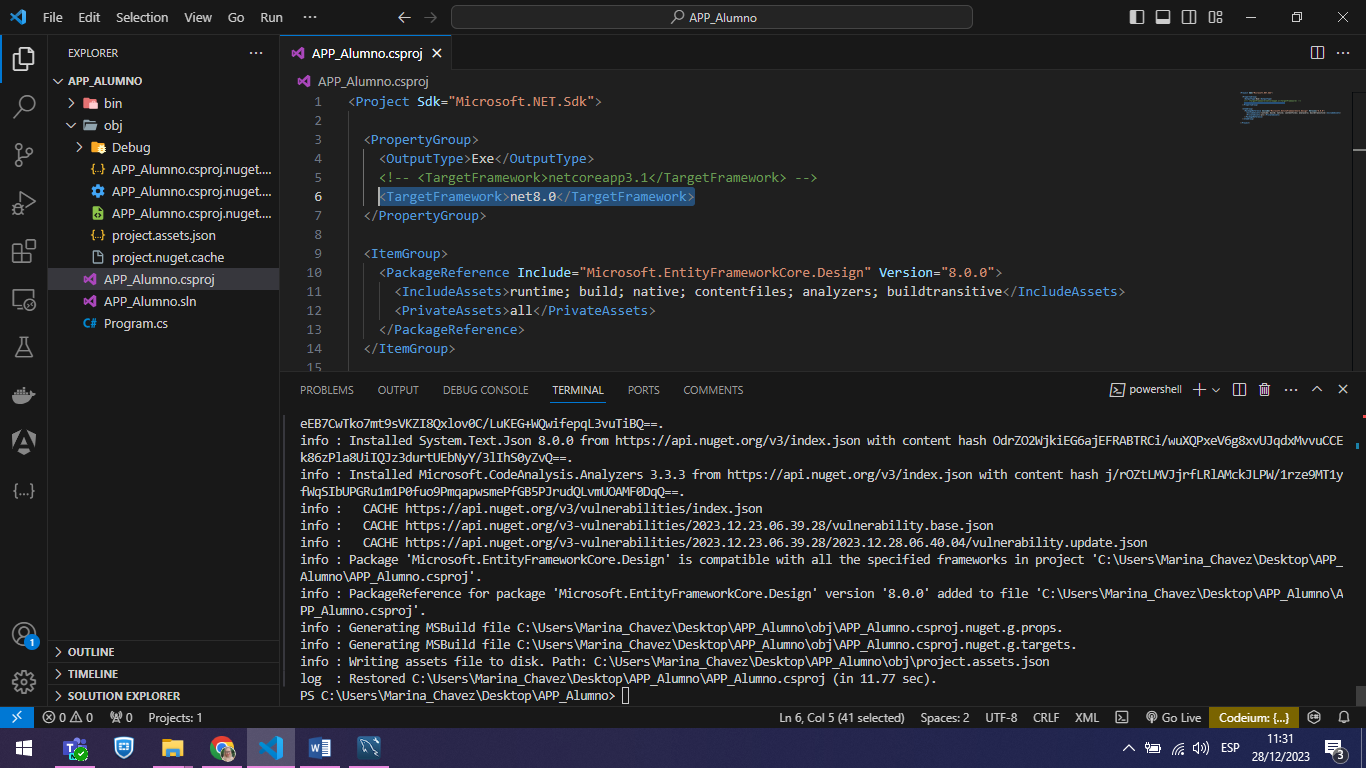
dotnet add package Microsoft.EntityFrameworkCore.Design --version 8.0.0

Agregando net 8 en el archivo correspondiente:

<TargetFramework>net8.0</TargetFramework>



De ese modo, las versiones no dan errores:

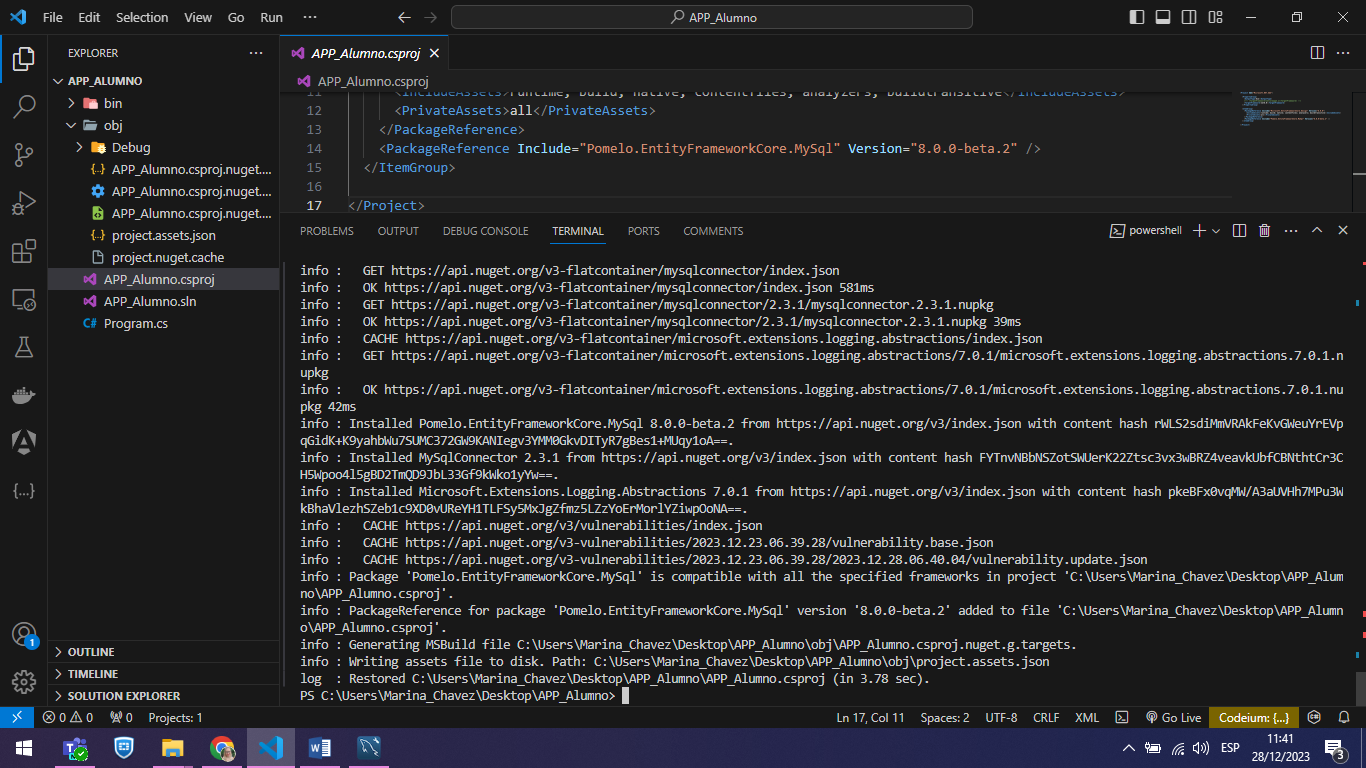


Si se desea usar la versión nro 3 el comando debería ser por ejemplo:

dotnet add package Microsoft.EntityFrameworkCore.Design –version --version 3.1.20

Ahora vamos a instalar Pomelo para poder hacer la conexión con la BD. Con el siguiente comando:

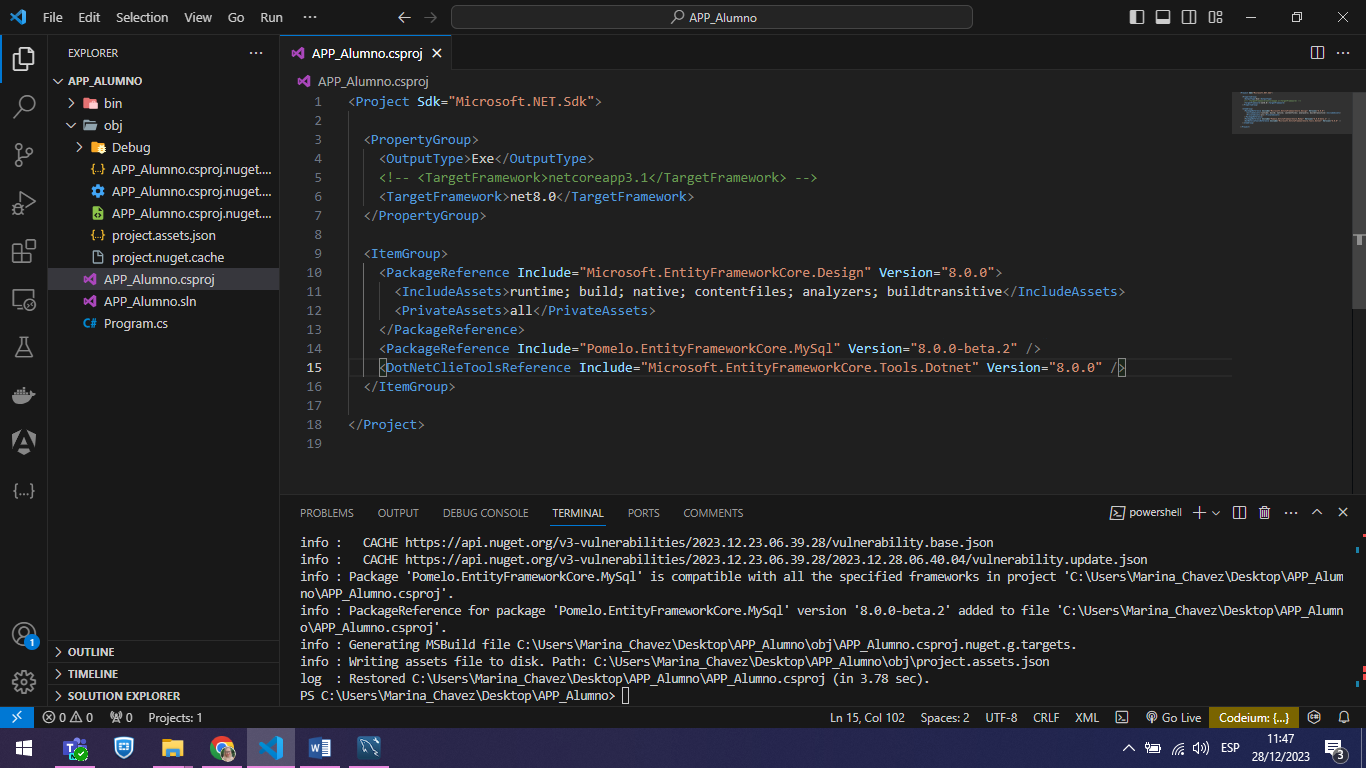
dotnet add package Pomelo.EntityFrameworkCore.MySql --version 8.0.0-beta.2



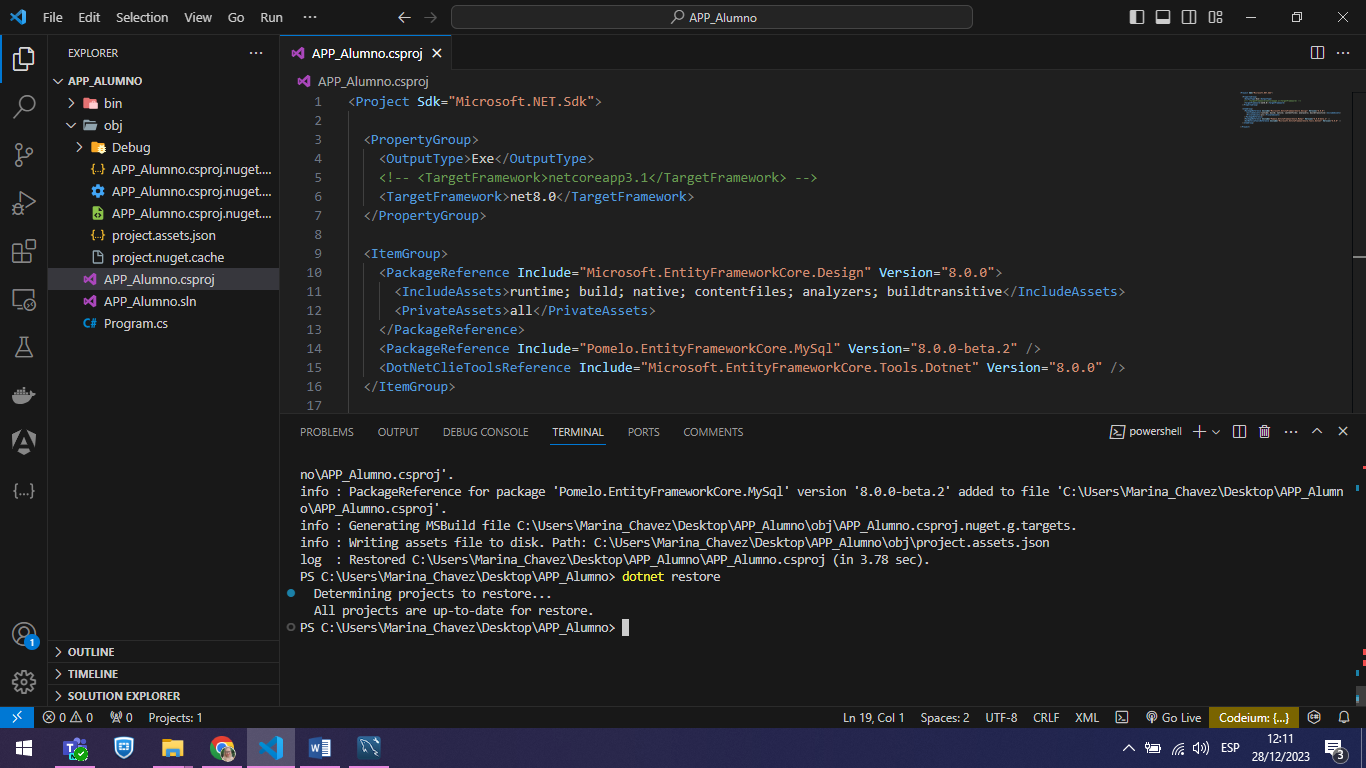
**Tener en cuenta que beta es versión con cambios no probados y no se recomienda para ambientes productivos.**

Agregamos esta línea para las herramientas necesarias a utilizar:

<DotNetClieToolsReference Include="Microsoft.EntityFrameworkCore.Tools.Dotnet" Version="8.0.0" />



Debemos restaurar los paquetes:

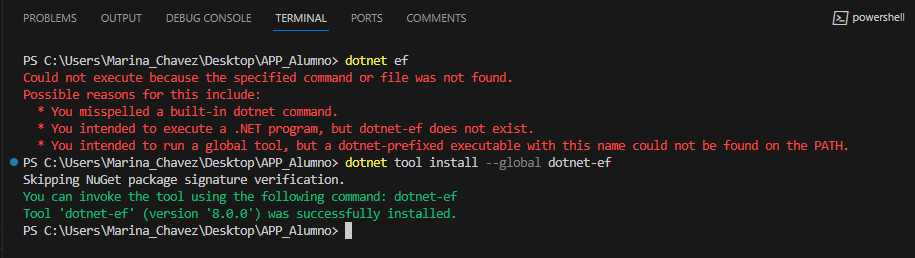


dotnet tool install --global dotnet-ef

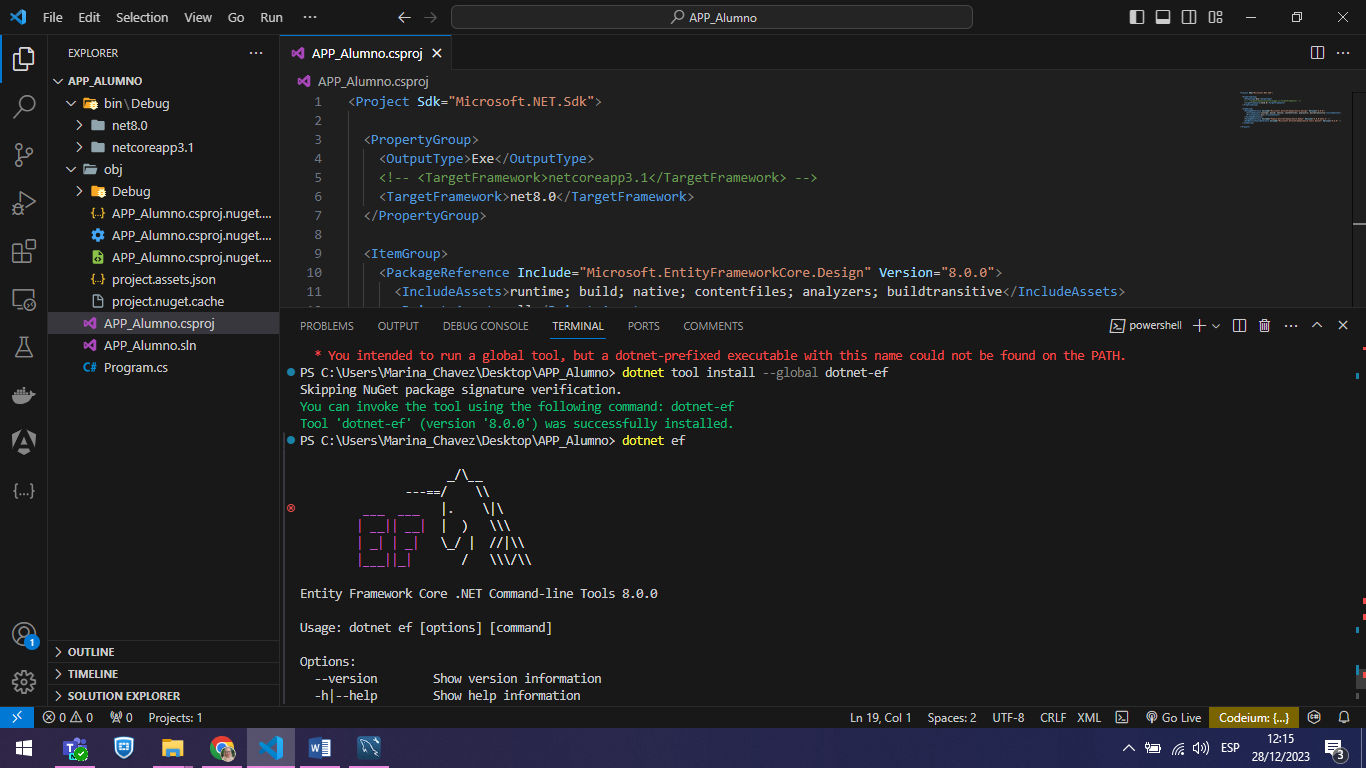
Skipping NuGet package signature verification.

You can invoke the tool using the following command: dotnet-ef

Tool 'dotnet-ef' (version '8.0.0') was successfully installed.

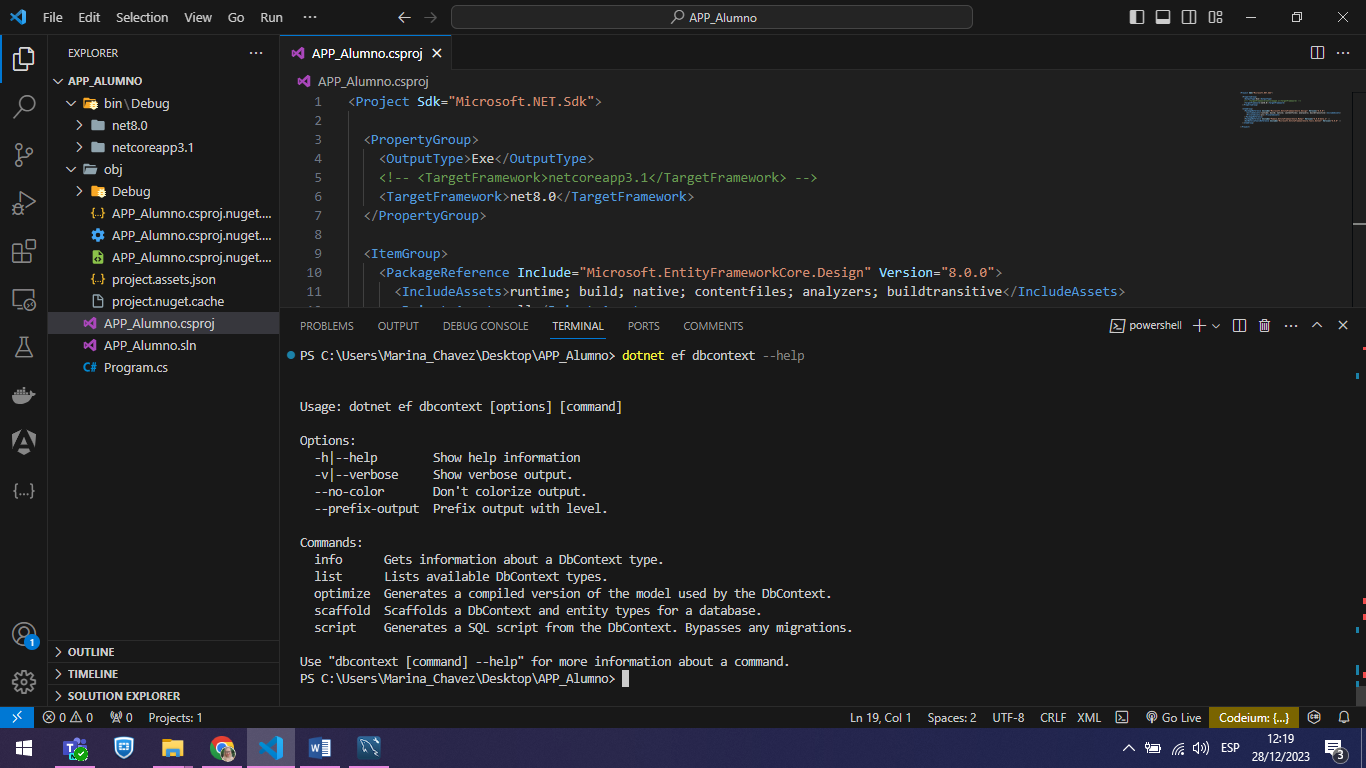


Ahora si funciona dotnet ef:



Vamos a verificar que esté instalado correctamente y que podamos utilizar las herramientas:

Dotnet ef dbcontext –help

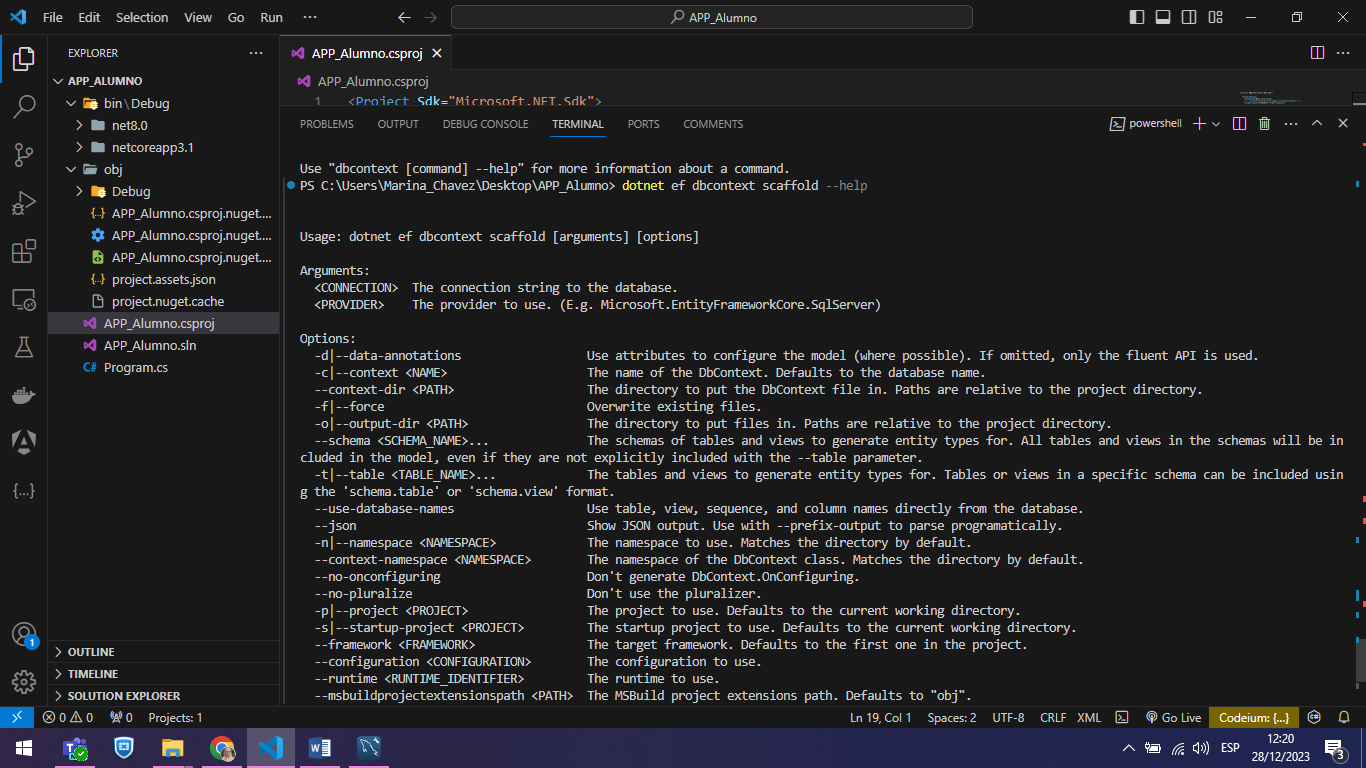


Nos interesa esta herramienta que es la que nos conecta con la base de datos:

scaffold Scaffolds a DbContext and entity types for a database.

dotnet ef dbcontext scaffold –help

Para ver los comandos disponibles:



-o|--output-dir <PATH> The directory to put files in. Paths are relative to the project directory.

Este comando es el que nos interesa para conectar con la base de datos.

Para hacer un poco más interesante este ejemplo. Realizo una tabla carrera en la DB y edito la tabla alumnos con su respectiva clave foránea:

CREATE TABLE `alumno\_bd`.`carrera` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(255) NOT NULL,

PRIMARY KEY (`id`)

);

ALTER TABLE `alumno\_bd`.`alumno`

ADD COLUMN `carrera\_id` INT,

ADD CONSTRAINT `fk\_alumno\_carrera`

FOREIGN KEY (`carrera\_id`)

REFERENCES `alumno\_bd`.`carrera` (`id`);

Inserto algunos datos en la tabla carrera:

INSERT INTO `alumno\_bd`.`carrera` (`nombre`) VALUES

('Ingeniería Informática'),

('Psicología'),

('Arquitectura'),

('Medicina'),

('Contabilidad');

Realizo un update de alumno para agregar ID de carreras:

UPDATE `alumno\_bd`.`alumno` SET `carrera\_id` = '1' WHERE (`id` = '1');

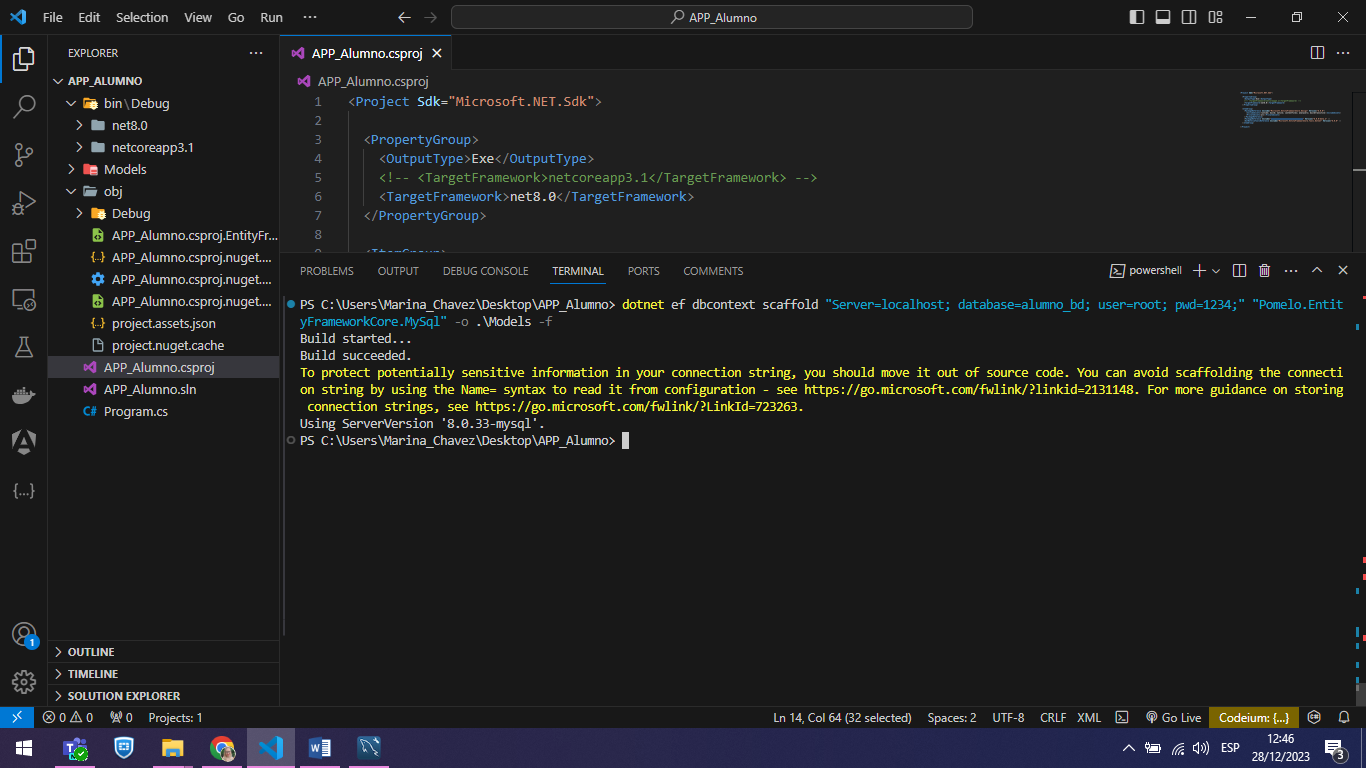
UPDATE `alumno\_bd`.`alumno` SET `carrera\_id` = '2' WHERE (`id` = '2');

UPDATE `alumno\_bd`.`alumno` SET `carrera\_id` = '2' WHERE (`id` = '3');

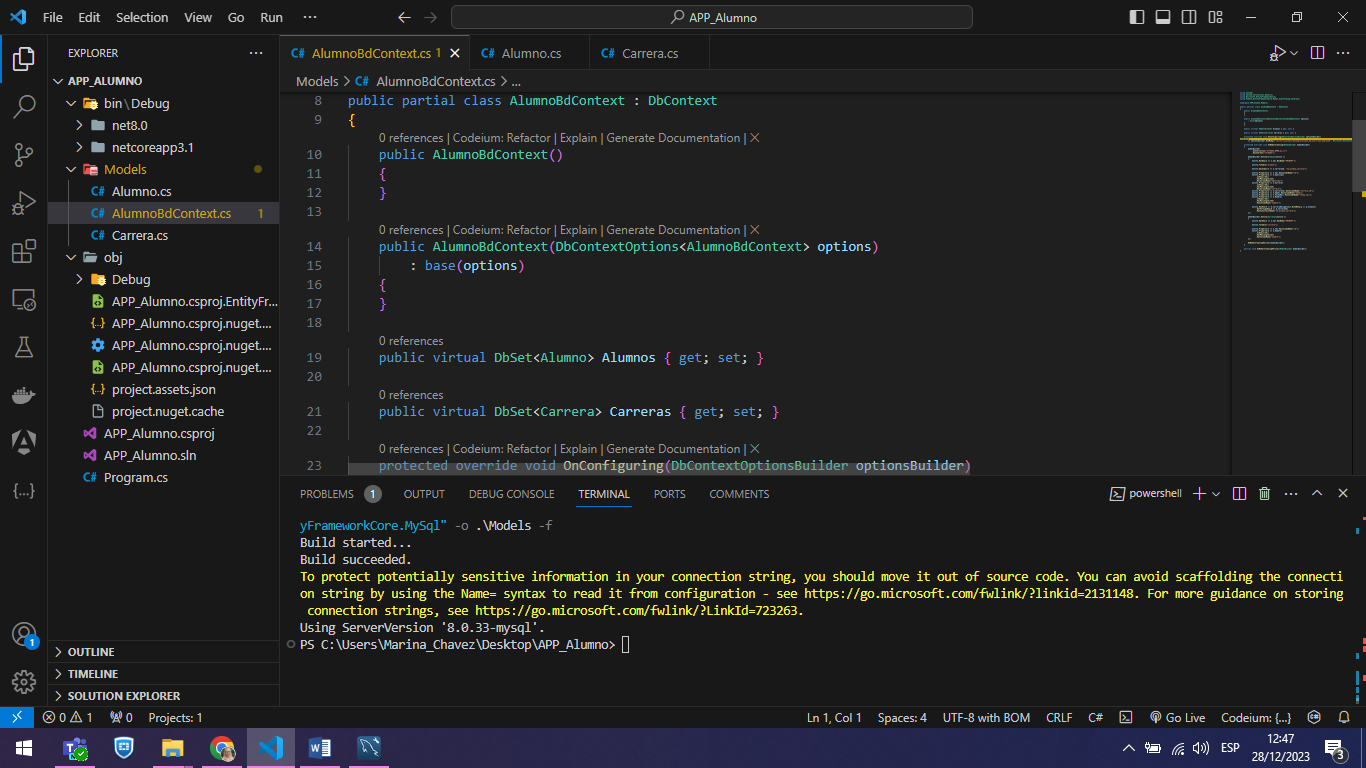
Ahora si, vamos a hacer la conexión a la base de datos con el siguiente comando:

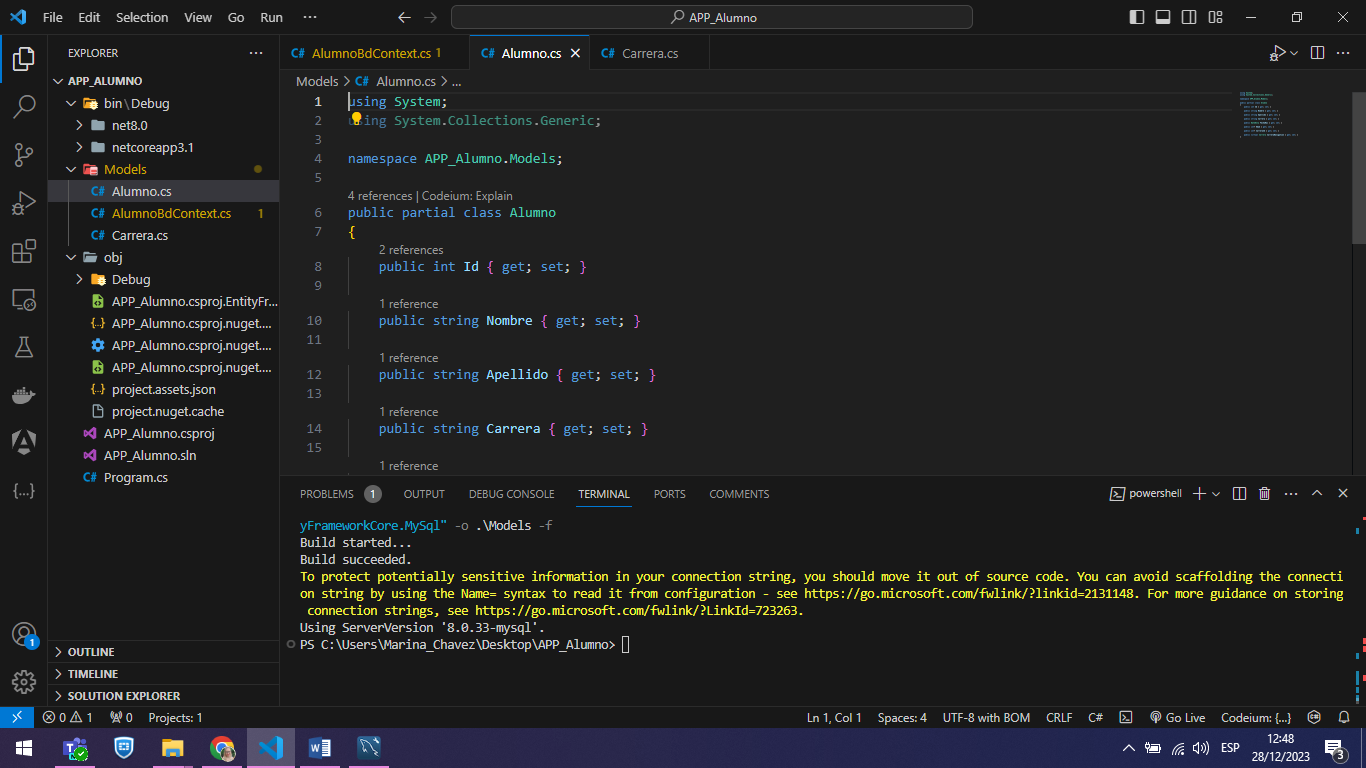
dotnet ef dbcontext scaffold "Server=localhost; database=alumno\_bd; user=root; pwd=1234;" "Pomelo.EntityFrameworkCore.MySql" -o .\Models –f

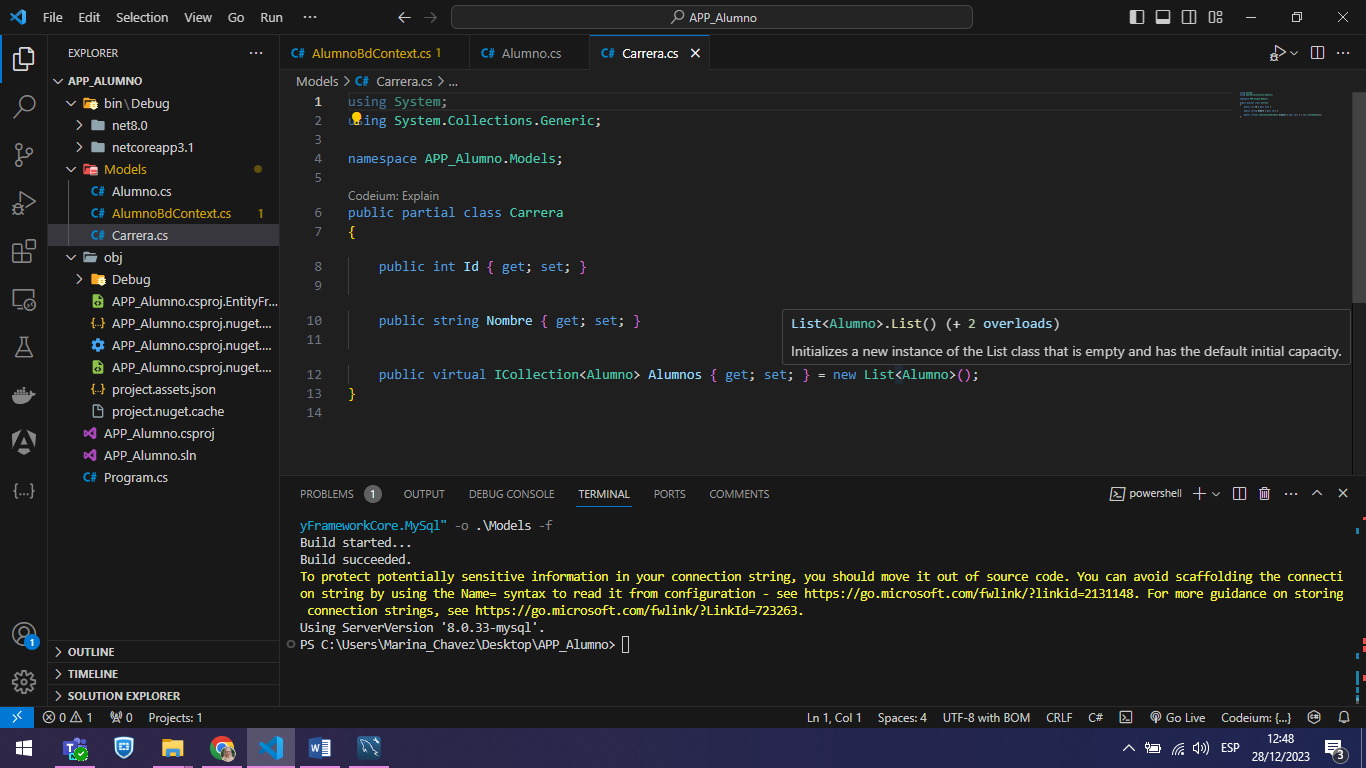
Con este comando, se van a generar las clases y el dbcontext:



Se genera en la carpeta models:





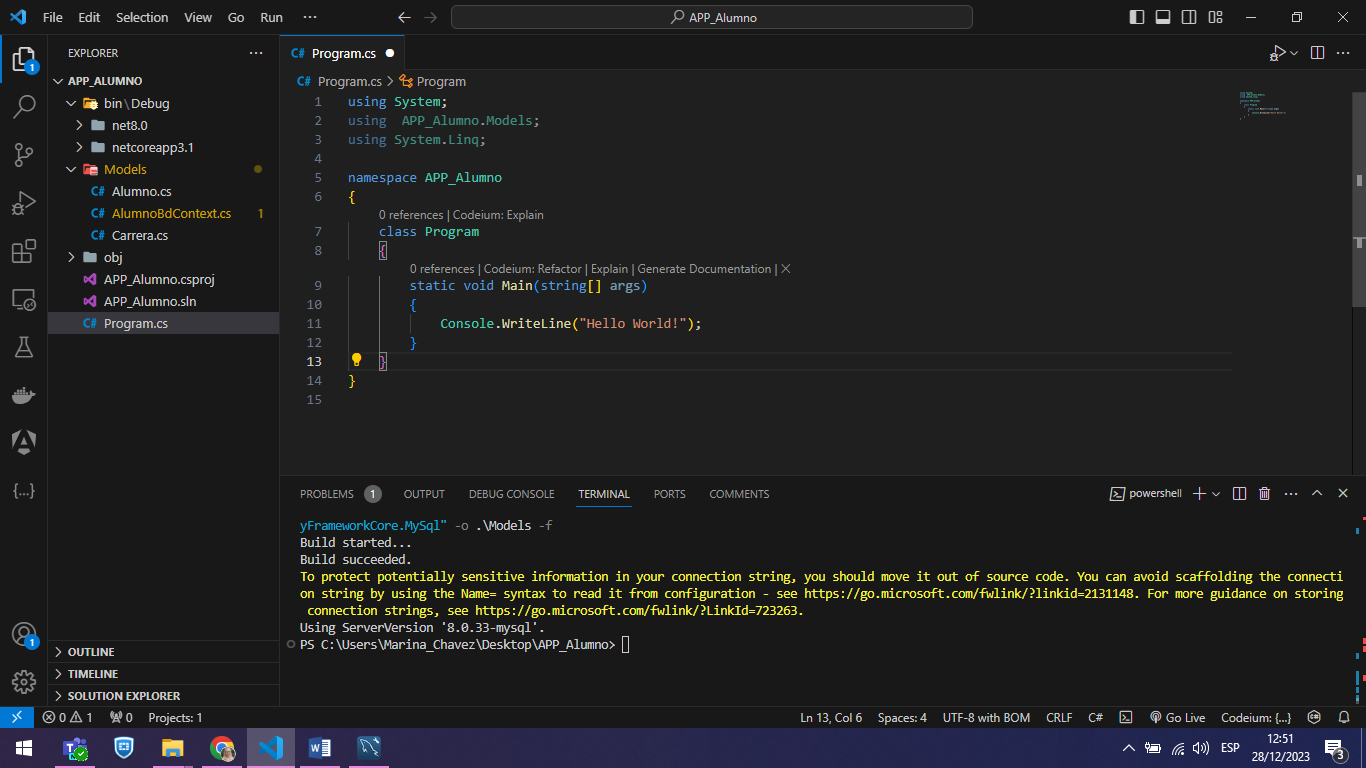


Se crearon cada una de las clases con sus getter and setters y la clase “public partial class AlumnoBdContext : DbContext”

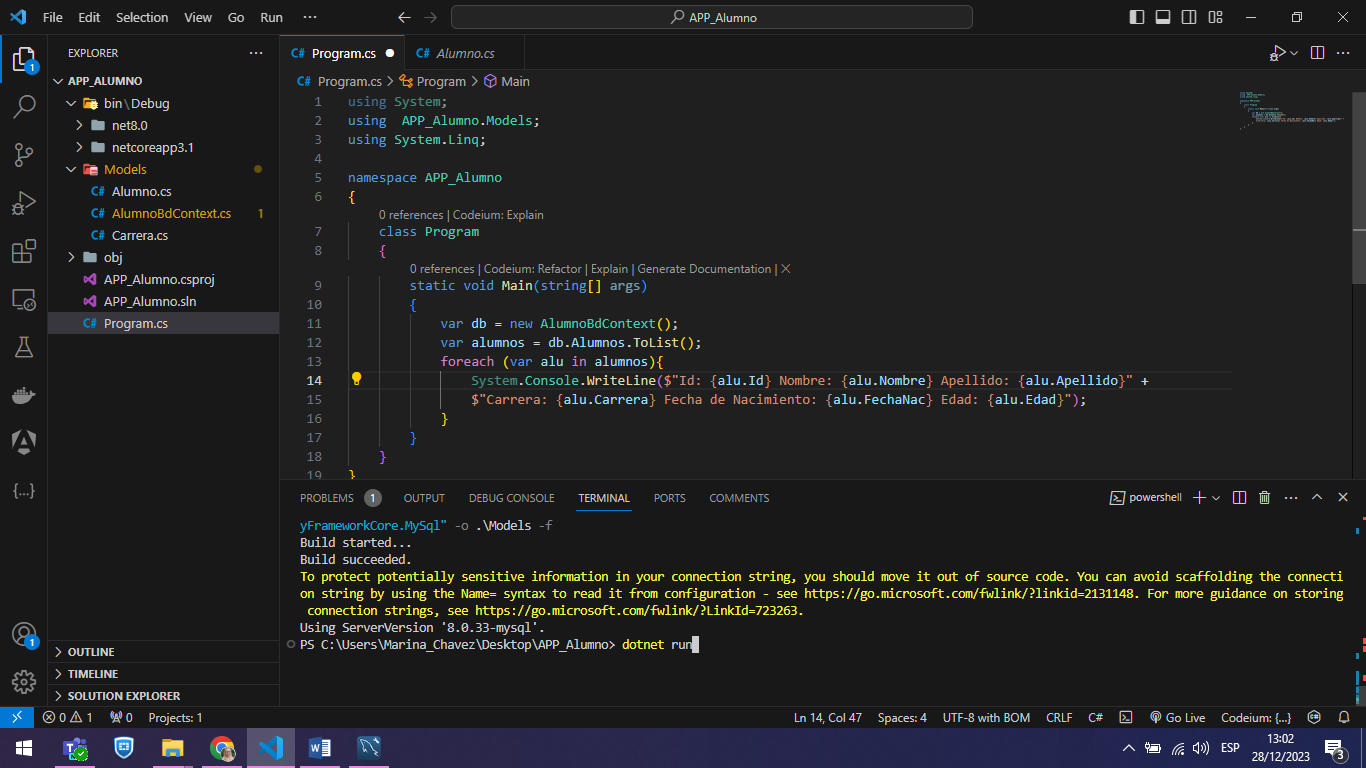
En el archivo “Program.cs” vamos a agregar estos dos using:

using  APP\_Alumno.Models;

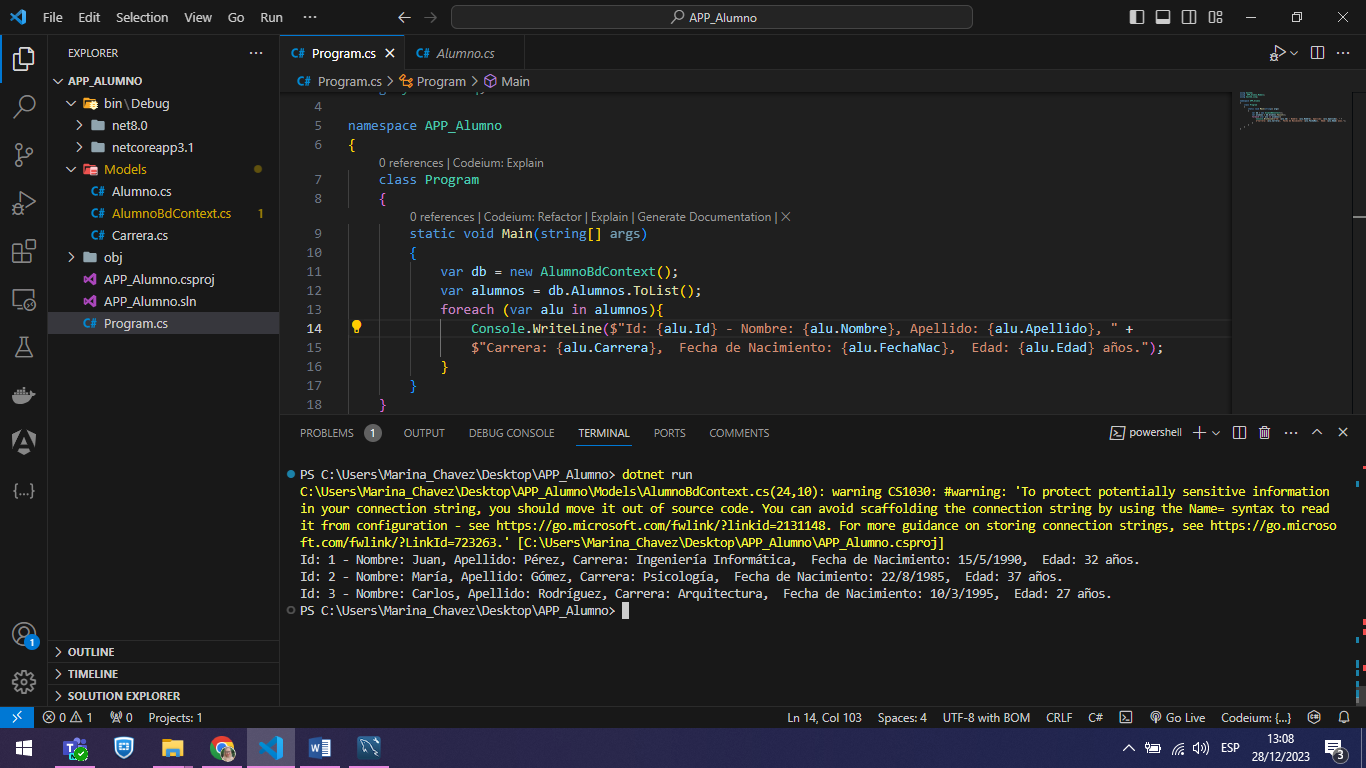
using System.Linq;



Agregamos las variables con sus respectivos usos y hacemos un foreach para poder visuaizar en consola los nombres y datos de nuestros alumnos. Usamos el comando “dotnet run”:



Observamos en la consola la data:



Estos fueron los primeros pasos para conectar a una base de datos y generar las clases automáticas y poder automatizar el proceso de generación de clases con sus getter and setter y además, con herramientas que ya brindan métodos de listado ya disponibles para su uso, por ejemplo.