



UNIVERSIDAD ISRAEL

CIENCIAS DE LA INGENIERÍA

CARRERA DE SISTEMAS DE INFORMACIÓN

PLATAFORMAS DE DESARROLLO 2

SEMESTRE 2020 B

INFORME DE LABORATORIO 1-S5

TEMA: MVC

ESTUDIANTE(S): Marco Antonio Ayala Lituma

CURSO: Octavo

PARALELO: "A"

PROFESOR: Mg. Luis Fernando Aguas Bucheli

QUITO, 2021

1. TEMA: MVC

2. OBJETIVOS:

- Adquirir los conceptos básicos relacionados con MVC
- Reconocer las características de MVC
- Crear una demostración practica
- Aprender como utilizar en visual studio

3. INTRODUCCION:

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés. En este artículo estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

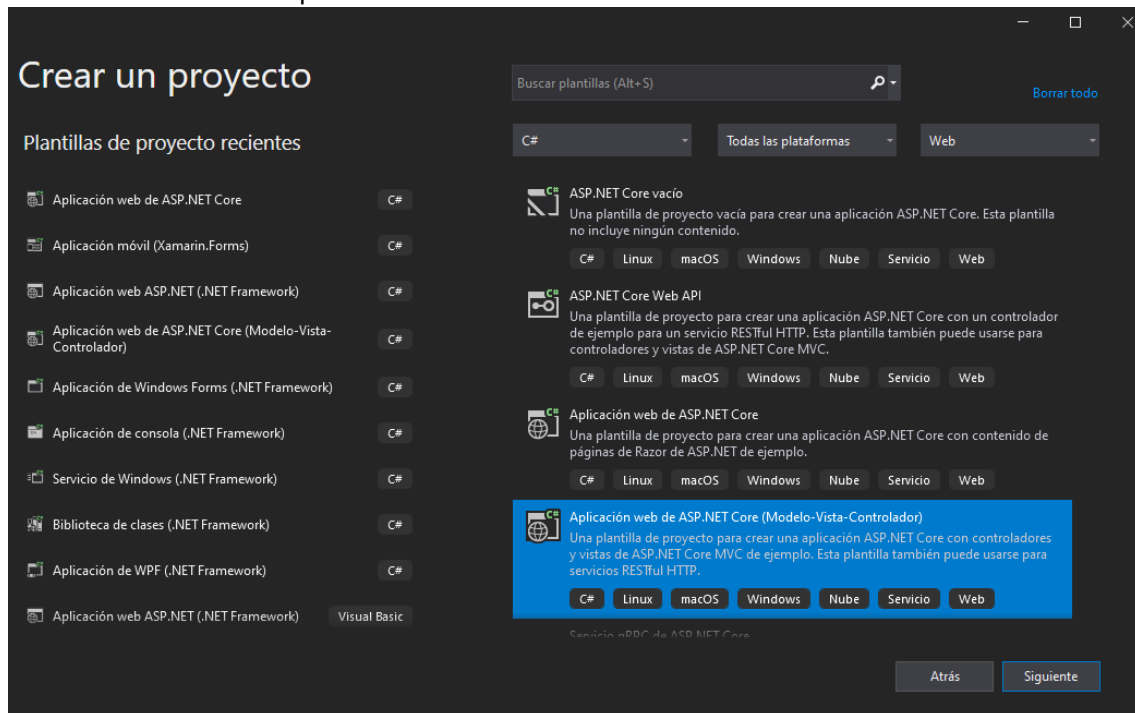
De manera genérica, los componentes de MVC se podrían definir como sigue:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.¹²
- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase Middleware).
- La Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

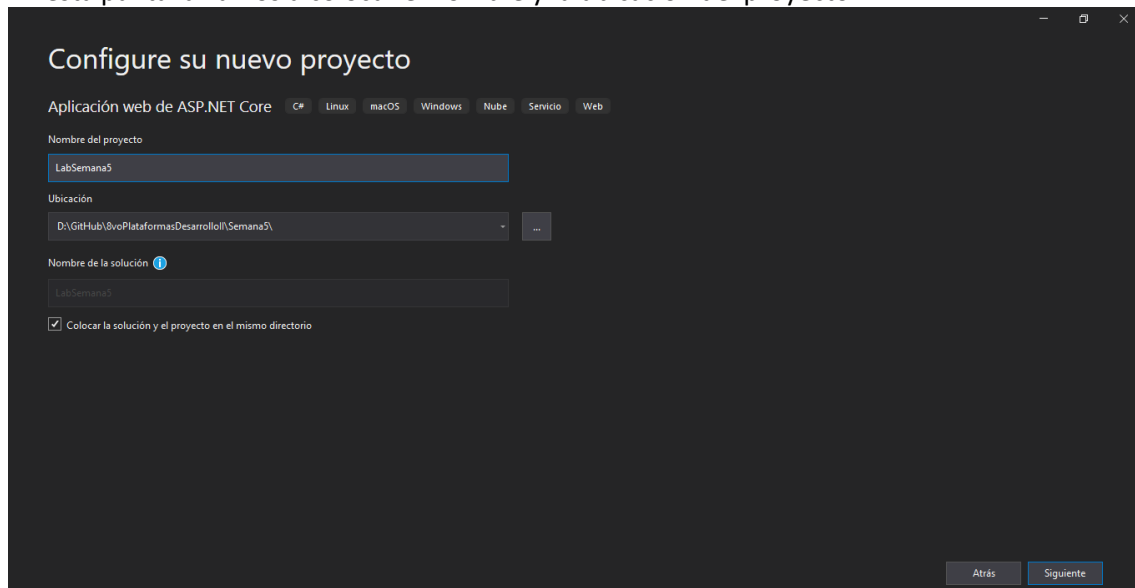


4. DESARROLLO:

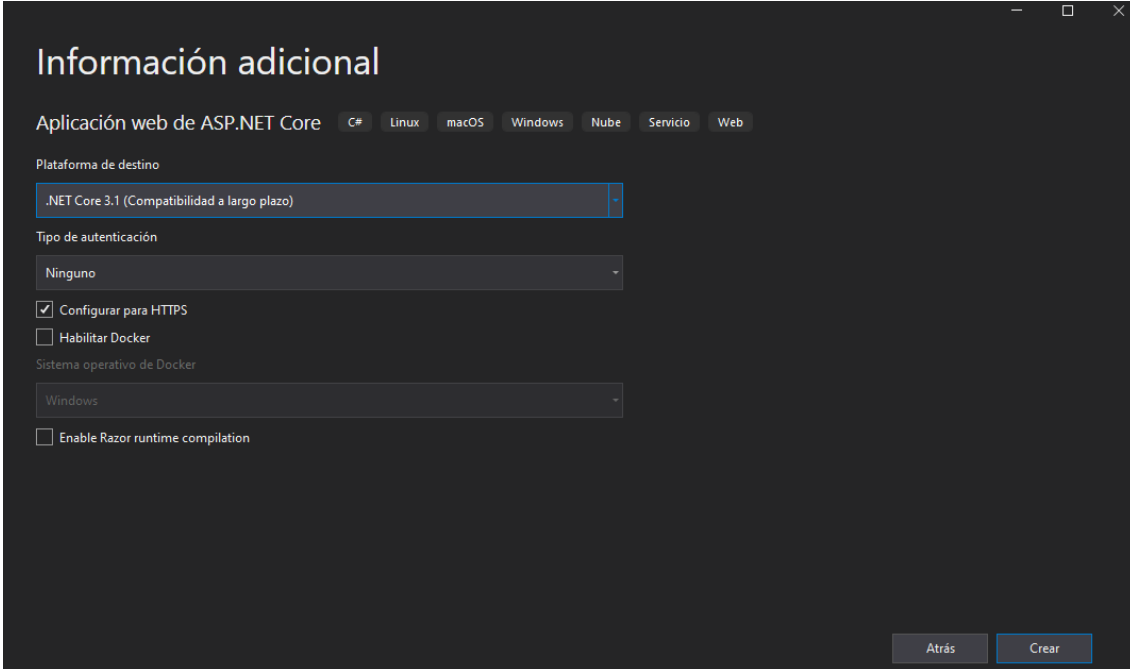
Abrimos visual studio y seleccionamos el tipo de proyecto que vamos a realizar, en este caso vamos a seleccionar el Aplicación Web en modelo vista controlador.



En esta pantalla vamos a colocar el nombre y la ubicación del proyecto.



En esta pantalla vamos a seleccionar el tipo de proyecto de plataforma de destino



Información adicional

Aplicación web de ASP.NET Core C# Linux macOS Windows Nube Servicio Web

Plataforma de destino
.NET Core 3.1 (Compatibilidad a largo plazo)

Tipo de autenticación
Ninguno

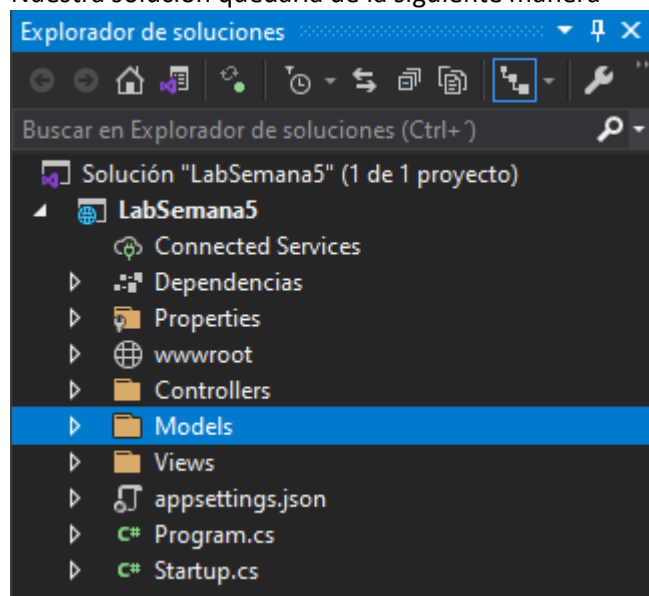
☒ Configurar para HTTPS
☐ Habilitar Docker

Sistema operativo de Docker
Windows

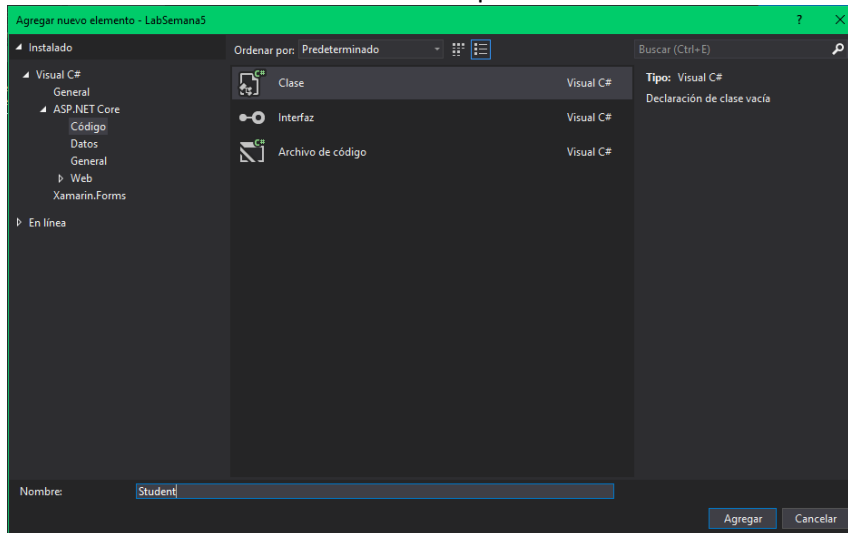
☐ Enable Razor runtime compilation

Atrás Crear

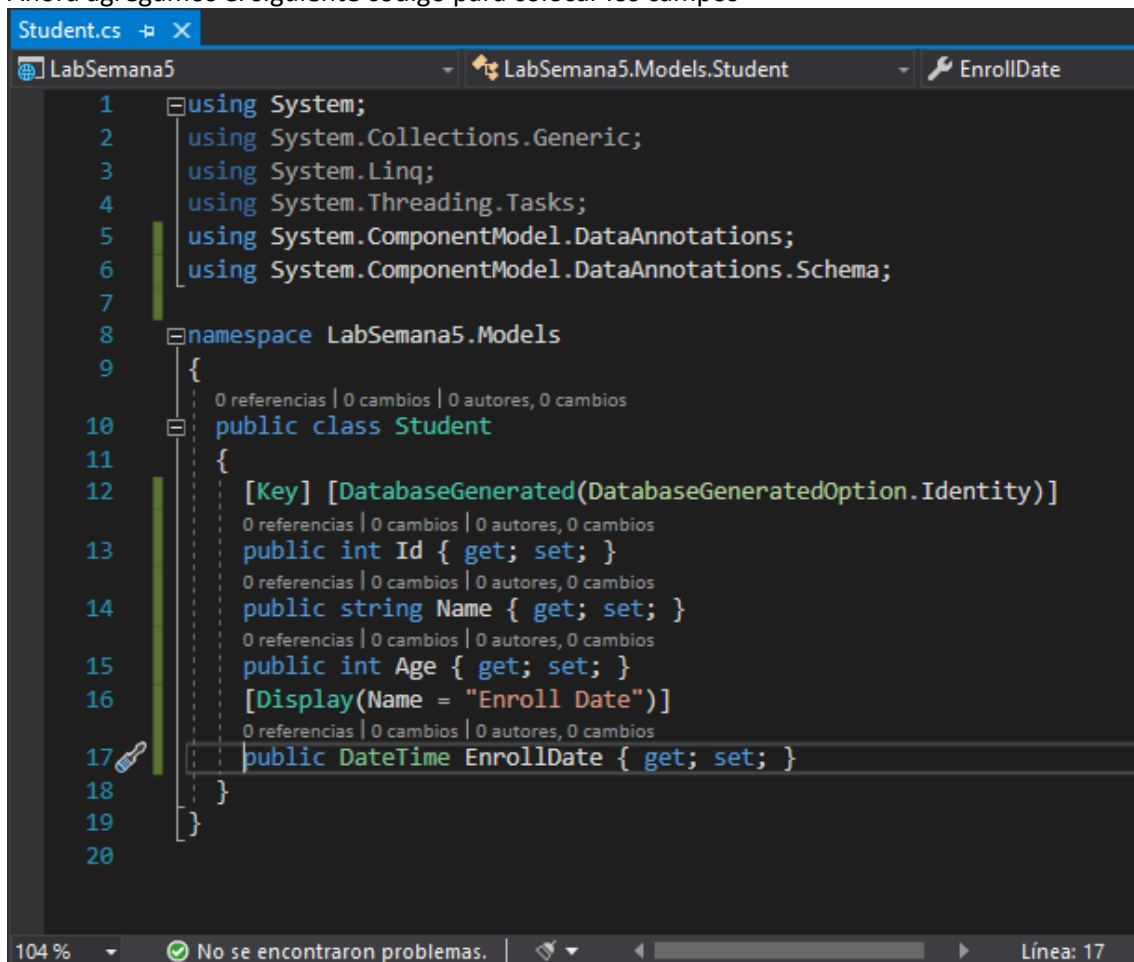
Nuestra solución quedaría de la siguiente manera



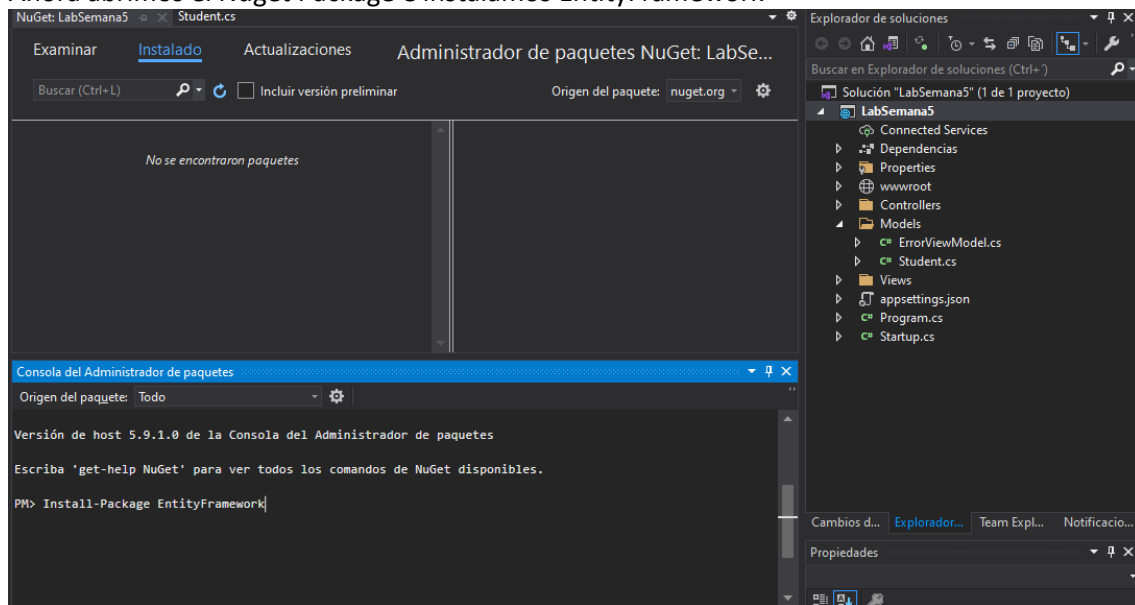
Ahora vamos a crear una clase en la carpeta models



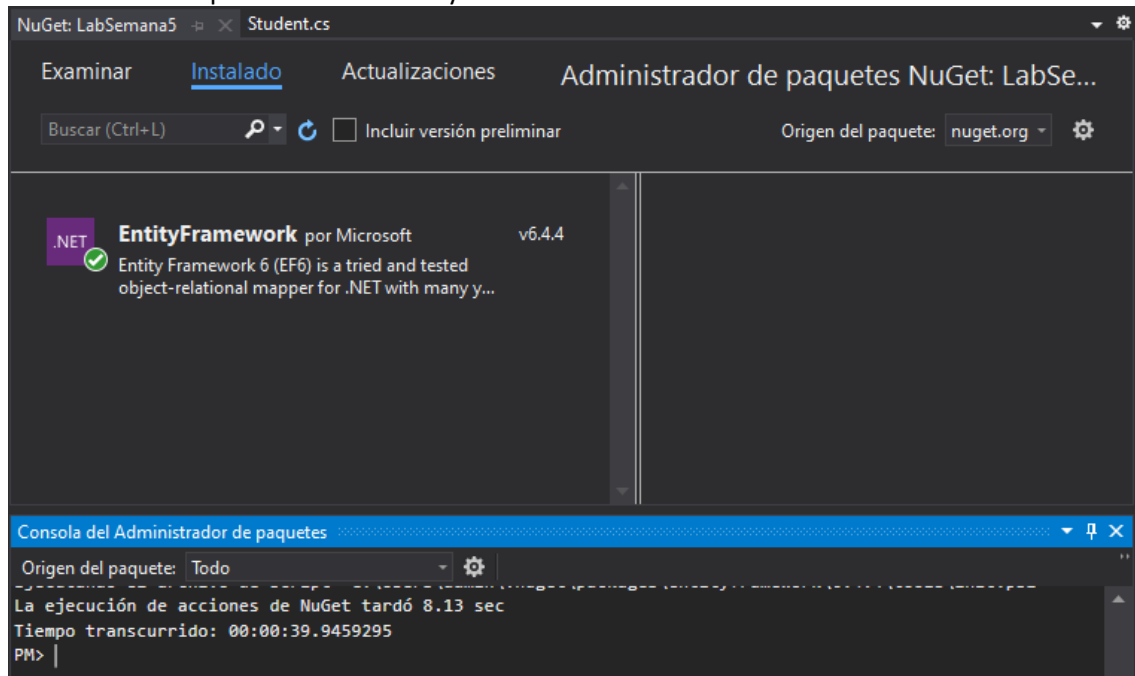
Ahora agregamos el siguiente código para colocar los campos



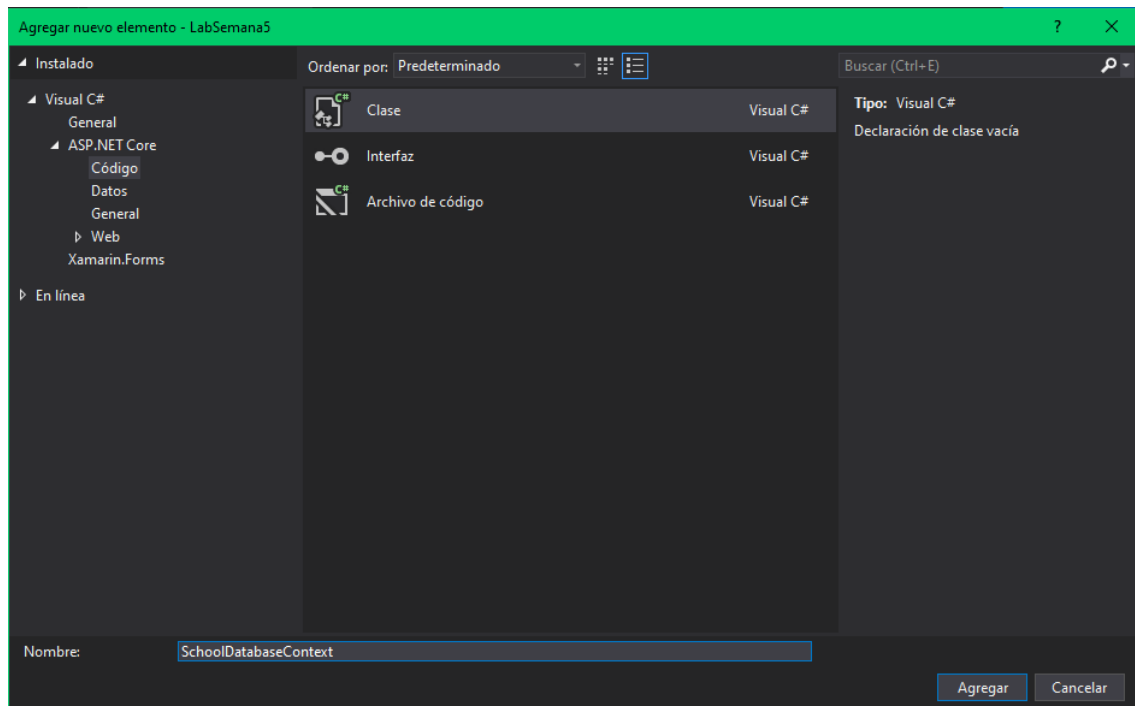
Ahora abrimos el Nuget Package e instalamos EntityFramework



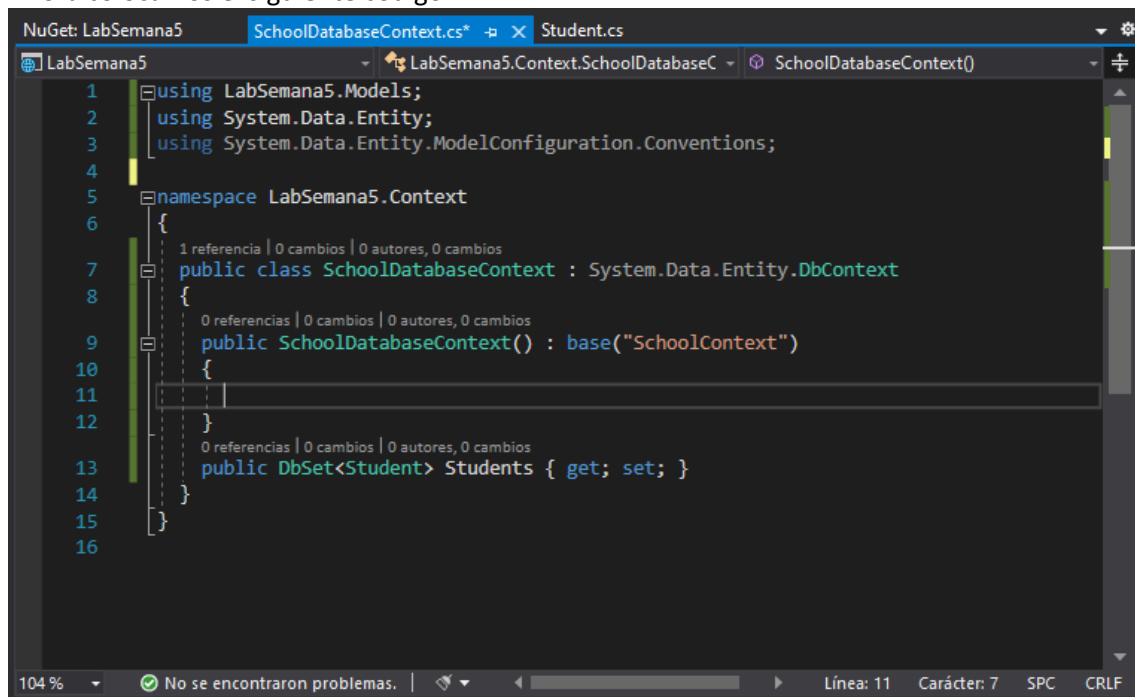
Se mostrara una pantalla así cuando ya este instalado



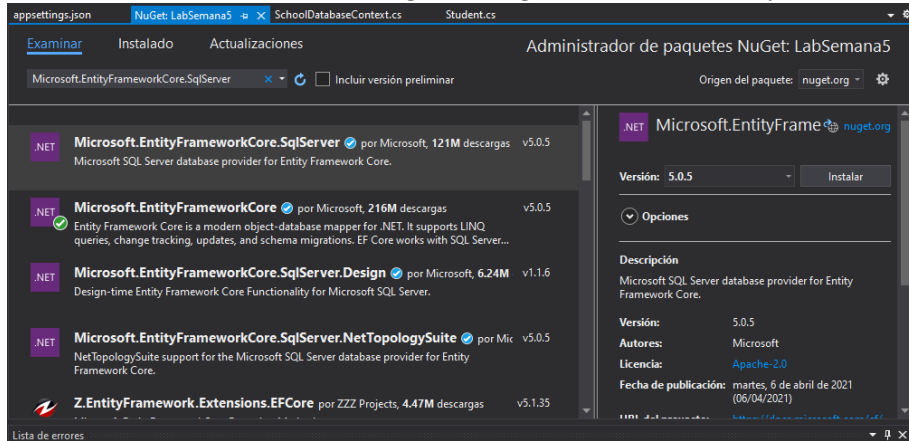
Ahora vamos a crear la clase de conexión con el contexto



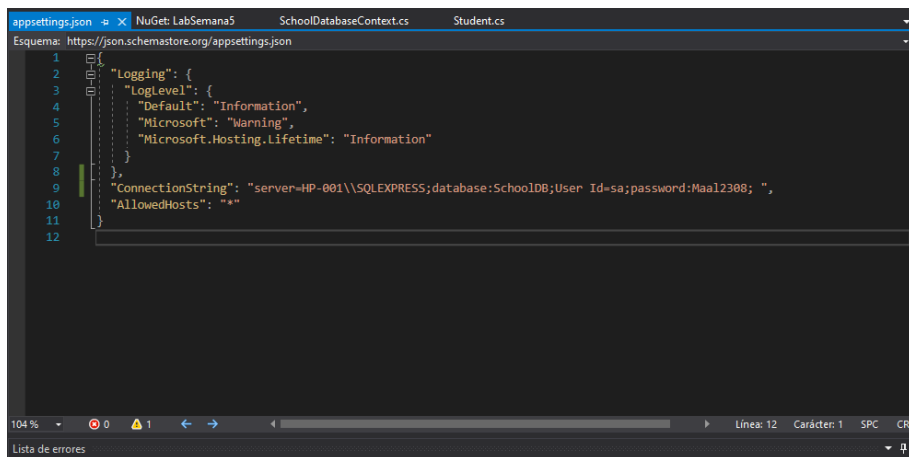
Ahora colocamos el siguiente código



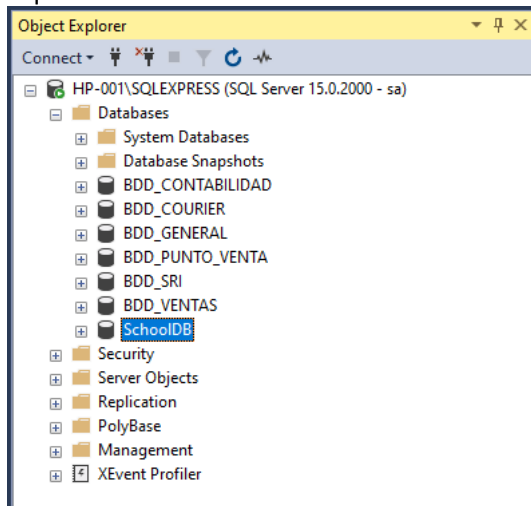
También abrimos de nuevo el NUget Package e instalamos EntityFramework de SQLserver



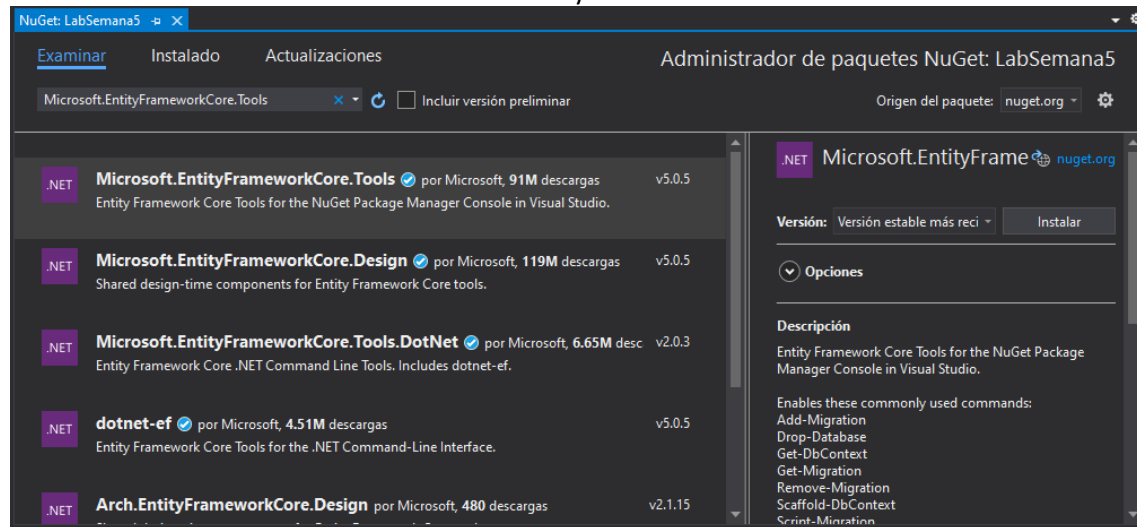
Aquí agregamos la cadena de conexión con la base de datos



Aquí creamos la base de datos



Finalmente instalamos la herramientas de Entity FrameworkCore.Tools



5. CONCLUSIONES:

MVC es un patrón de diseño enfocado a separar las responsabilidades dentro de nuestra aplicación y es muy utilizado en la web por su enfoque y las ventajas que ofrece con respecto a algunas otras formas o patrones de desarrollo de aplicaciones web. Puedes encontrar frameworks prácticamente para cualquier lenguaje web.

6. RECOMENDACIONES:

Utilizando tecnologías Microsoft se puede utilizar desde la conexión de la base de datos creación de modelos en un proyecto con las estructura de MVC que nos ayuda en el ahorro de tiempo de desarrollo y que la organización de código este correcta y fácil de utilizar para el equipo de desarrollo.

7. BIBLIOGRAFIA:

MVC Recuperador en: [Modelo–vista–controlador - Wikipedia, la enciclopedia libre](#)

Iniciando MCV, Recuperador en: [Getting Started with ASP.NET MVC 5 | Microsoft Docs](#)