Antes de empezar el proyecto defino los patrones de arquitectura y de diseño que se aplicaran a la prueba y queda formalizado de la siguiente manera

**Patrón de arquitectura**

El patrón seleccionado para trabajar en esta prueba técnica será el patrón de capas, para ello cada capa tendrá una responsabilidad única en la aplicación por ejemplo la capa de persistencia, la capa de funcionabilidad y la capa del modelo de negocio, también con esta arquitectura es posible dividir las tareas de alto nivel y bajo nivel siendo las ultimas aquellas que se conectan a la base de datos y obtiene información.

**Patrón de diseño**

Los patrones de diseño utilizados para esta prueba técnica fueron los siguientes:

* **MVC**

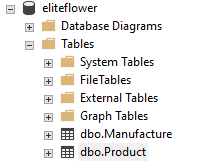
Se dividió el proyecto en 4 capas, las cuales se encargan de funciones diferentes como los principios SOLID lo expresan, la capa llamada EliteFlower.Core es la encargada de mantener el conjunto de los modelos relacionados a la base de datos. La capa EliteFlower.Data es la encargada de generar la conexión a la base de datos y la asociación de los modelos de Core directamente a las tablas de la base de datos, los servicios que interactúan con Core y Data es llamada EliteFlower.Services, estos serán los que se inyectaran a la solución y quienes se encargaran con la comunicación a la base de datos, la ultima capa es llamada EliteFlower.Web quien es la encargada de mostrar todo al usuarios final, es el frontend de la aplicación, aquí se inyectaran los servicios y todas las referencias a las demás capas para ser utilizadas cuando sean necesario.

* **Patrón de inyección de dependencia**

Se aplico este patrón por medio del Entity Framework con la versión adaptada para funcionar con .NET 5, esta inyección de dependencias asegura que la comunicación se realice solo por medio de los servicios del proyecto EliteFlower.Services

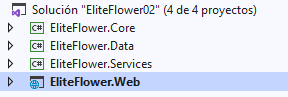
Una vez definido los patrones a utilizar, se resume las acciones tomadas en la construcción de la prueba técnica.

La base de datos se crea en el motor de base de datos SQLServer Management Studio Version 18, se usa el modelo entregado con una leve variación y es la de agregar el campo pathImage que servirá para poder guardar la ruta de la imagen de cada producto.



El proyecto se crea en Visual Studio Community 2019, se crea un proyecto MVC .NET CORE vacío con la versión .NET 5.0, por lo que cualquier nuget que se necesite se descargara para que tenga compatibilidad para esta versión.

Se crean para el proyecto 3 proyectos de bibliotecas y 1 proyecto web que aplica MVC optimo para el uso de las capas que se crearan.



Instalar Entity Framework para ello se instalan los siguientes nugets

* Microsoft.EntityFrameworkCore.SqlServer Versión 5.0
* Microsoft.EntityFrameworkCore.Tools Versión 5.0

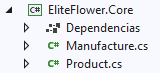
Luego en consola obtener la clase que servirá para la conexiona la base de datos por medio de la siguiente instrucción

* Scaffold-DbContext “Server= DESKTOP-77M474K\SQLEXPRESS;Database=Blogging;Trusted\_Connection=True;” Microsoft.EntityFrameworkCore.SqlServer -OutputDir ModelsEliteFlower.Core

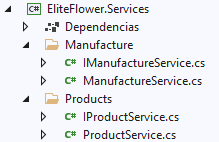
Crear la clase y pasar al archivo appsettings.json la información de conexión del DataContext



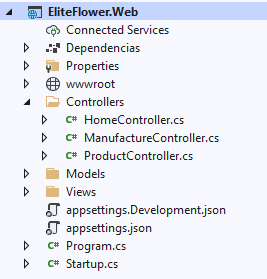
Crear los modelos en proyecto Core



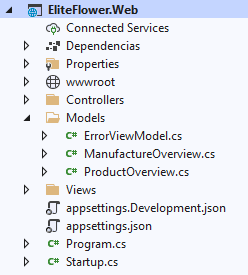
Crear los servicios necesarios para la comunicación entre el proyecto Web, Core y Data por medio de Services



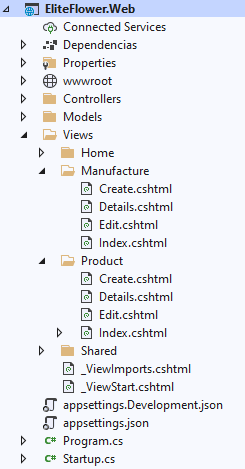
Crear los controladores y vistas necesarias para realizar los CRUD para Manufactures y para los Products



Como buena practica se deben crear modelos en el proyecto Web para no usar los de Core, entonces se procede a crearlos

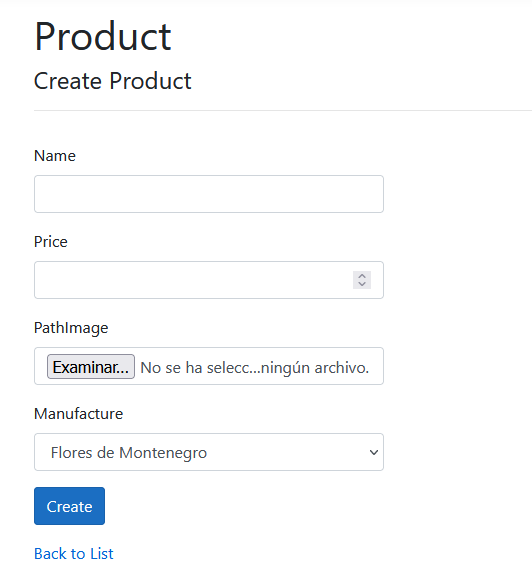


En el proceso de creación de las vistas se Instalar RuntimeCompilation de Razor para poder ver las actualizaciones en las vistas con solo actualizar y no reiniciar el proyecto.



Instalar **Microsoft.AspNetCore.Mvc.TagHelpers para poder validar los modelos del lado del servidor cuando se requiera más adelante.**

**Se procede a crear controles y métodos para realziar el CRUD de ambas entidades y adicionar la carga de imágenes.**



**El proyecto tiene anexo la copia de la base de datos y solo es modificar la cadena de conexión de appsettings.json por el server local donde se este probando una vez este creada la base de datos en SQLServer.**

