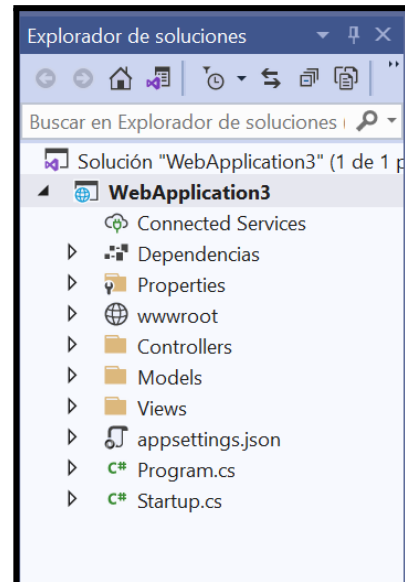
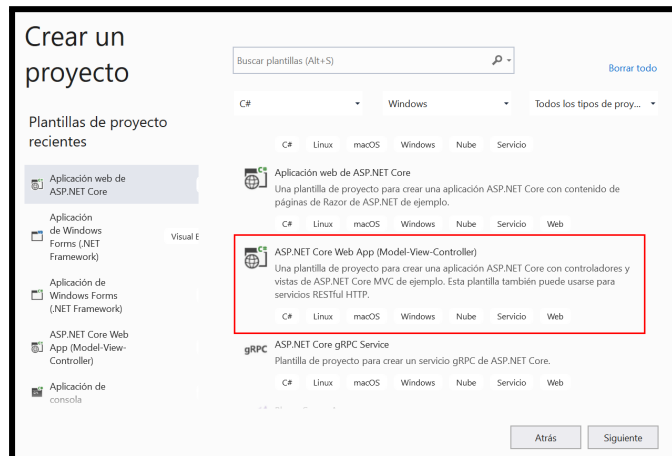


A continuación se muestra un esquema / resumen de los modelos/plantillas de desarrollo disponibles en ASP.net además de ASP.NET Core Razor Pages y ASP.NET Core MVC. Recuerda que ambos son marcos basados en servidor para compilar aplicaciones web con .NET

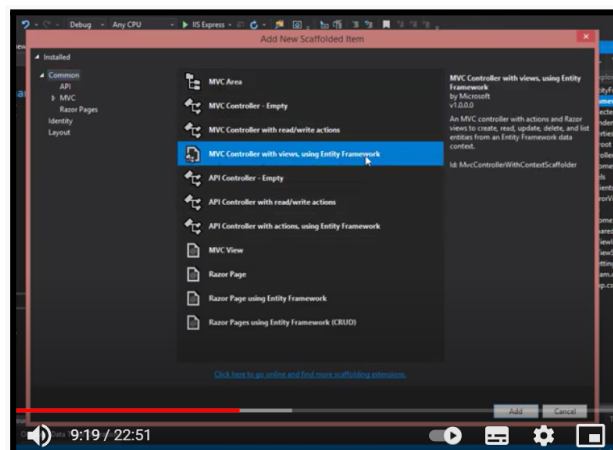
ASP.NET Core Web App (Model-View-Controller)

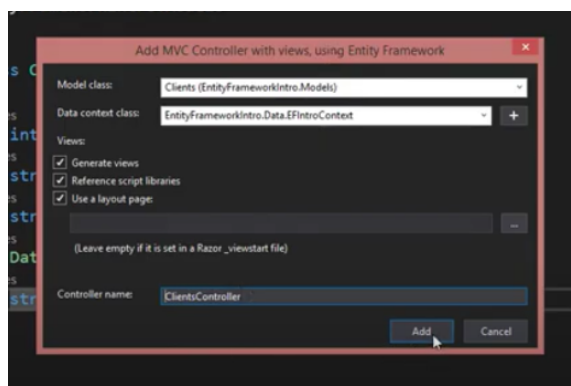
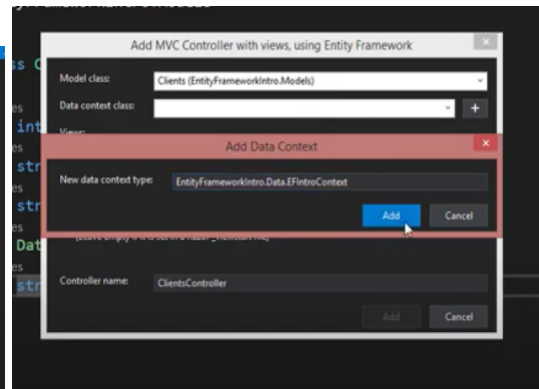
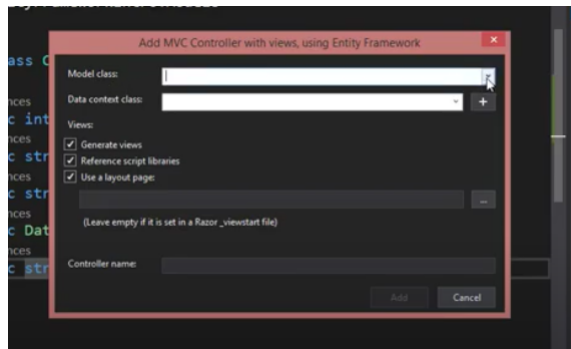
Cuando se crea el proyecto usando **MVC** se crearán las carpetas Models, Views, Controllers.



Ejemplo base partiendo de una bd ya existente

- **en models** crear una nueva clase con propiedades coincidentes con los campos de la tabla de la bd que vamos a mapear
- mediante **scaffolding** (técnica de generación de código) crear un controlador, una vista y el acceso a datos con **EF** (botón derecho en la solución /nuevo scaffolding item/ MVC controlador y vista con EF)





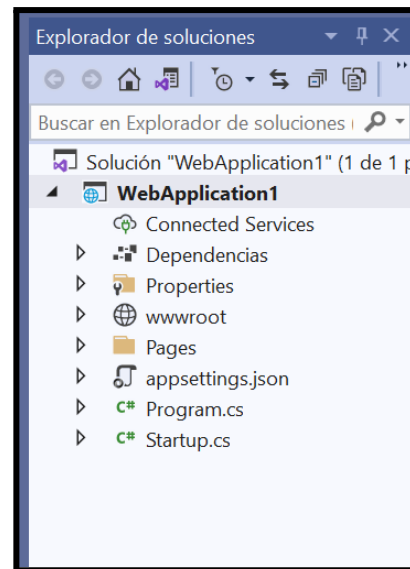
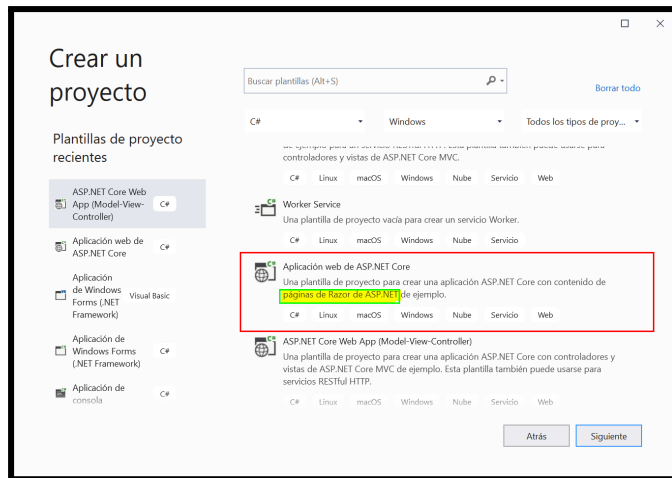
tras este paso Visual Studio crea por nosotros las vistas y controladores (*básicamente el crud está creado; ver código linq generado*)

- en **appsettings.json** indicar la connection string
- añadir nuevas opciones al menú de navegación (*vistas compartidas archivo **_Layout.cshtml***)
- recuerda que ha de haber una columna de tipo **identity** autoincremental en la BD!! (*exigencia de EF Core como lo era de EF6*)

Aplicación web de ASP.NET Core

- Razor Pages hace que la programación resulte más productiva que con controladores y vistas (MVC)
- Razor Pages hace que la programación resulte más productiva que con controladores y vistas (MVC)
- Razor Pages es un modelo basado en páginas. La interfaz de usuario y la lógica de negocios se mantienen separados, pero dentro de la página (*lo que da lugar a una compilación y actualización rápida de la interfaz de usuario*)
- Es la manera recomendada de crear nuevas aplicaciones basadas en páginas o en formularios para desarrolladores que no están familiarizados con ASP.NET Core.
- Proporciona un punto de partida más sencillo que ASP.NET Core MVC (mantiene las páginas de ASP.NET Core organizadas de una manera más sencilla que ASP.NET MVC)

En **Razor pages** se crea una carpeta "pages" donde estarán las vistas (*extensión **CSHTML***) y las páginas de modelo para cada vista. En este caso **no hay controladores**, sino que cada página tiene un archivo de C# asociado que se encarga de procesar las peticiones HTTP que recibe dicha página.



ojo! a elegir **Net core 5.0**