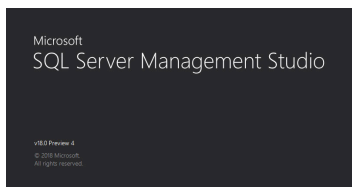


Creación de un API con arquitectura Hexagonal



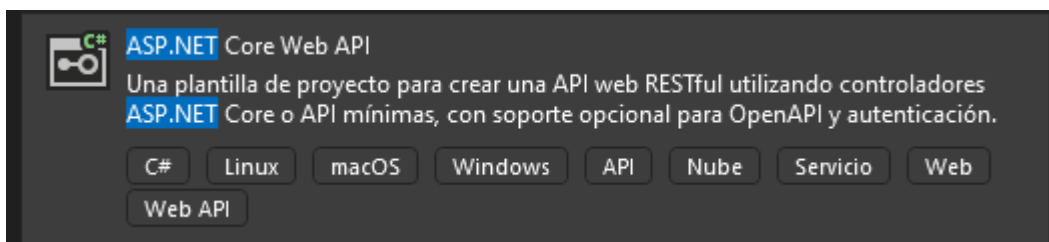
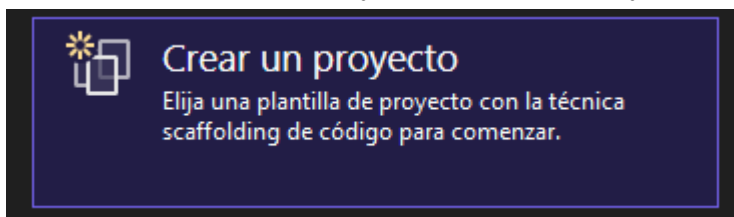
<https://visualstudio.microsoft.com/es/>



<https://learn.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

Como crear una API

Vamos a abrir visual studio y crear un nuevo proyecto “ASP.NET core Web API”



Configure su nuevo proyecto

ASP.NET Core Web API

C#

Linux

macOS

Windows

API

Nube

Servicio

Web

Web API

Nombre del proyecto

aspDotNetBlankProject

Ubicación

C:\Users\docto\source\repos

Nombre de la solución ⓘ

aspDotNetBlankProject

☐ Colocar la solución y el proyecto en el mismo directorio

Proyecto se creará en "C:\Users\docto\source\repos\aspDotNetBlankProject\aspDotNetBlankProject\"

Información adicional

ASP.NET Core Web API

C#

Linux

macOS

Windows

API

Nube

Servicio

Web

Web API

Framework ⓘ

.NET 8.0 (Compatibilidad a largo plazo)

Authentication de campo ⓘ

Ninguno

☒ Configurar para HTTPS ⓘ

☐ Habilitar compatibilidad con el contenedor ⓘ

SO del contenedor ⓘ

Linux

Tipo de compilación de contenedor ⓘ


Dockerfile

☒ Habilitar compatibilidad con OpenAPI ⓘ

☐ No usar instrucciones de nivel superior ⓘ

☒ Utilizar controladores ⓘ

☐ Inscribirse en la orquestación de .NET Aspire ⓘ

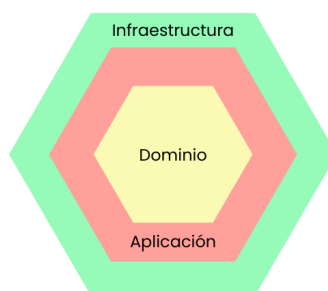
Una vez creado lo podemos ejecutar al darle play  [https](https://) y nos enviará a un swagger con el código de ejemplo del clima, este código lo vamos a borrar.

Borrar los siguientes archivos:

aspDotNetBlankProject\aspDotNetBlankProject\WeatherForecast.cs

aspDotNetBlankProject\aspDotNetBlankProject\Controllers\WeatherForecastController.cs

La arquitectura hexagonal



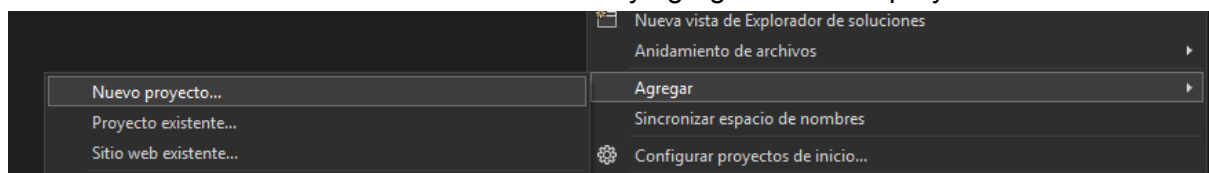
Esta arquitectura consta de 3 capas:

- Dominio: Resolver la lógica de negocio.
- Aplicación: Intermediar entre el dominio y la capa de infraestructura.
- Infraestructura: Conectarse a base de datos.

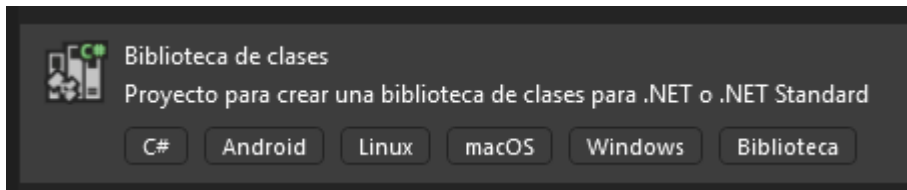
Como crear una capa

Una capa es un proyecto que tiene una única responsabilidad, vamos a crear nuestras 3 capas

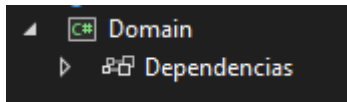
Basta solo con darle click derecho a la solución y agregar un nuevo proyecto:



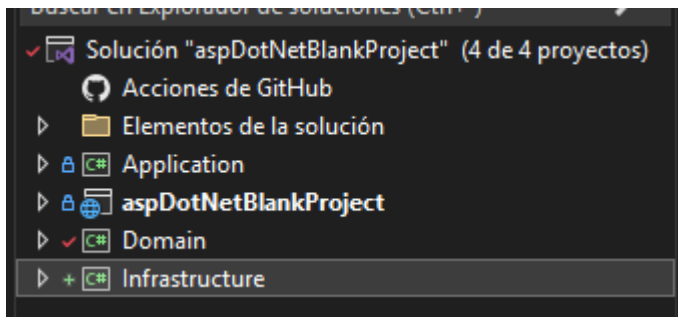
Y Agregamos una biblioteca de clases:



Por ejemplo creamos la capa de dominio el cual vendrá con un archivo de ejemplo el cual procederemos a borrar.



Y luego creamos nuestras capas de Infraestructure y Application



Ahora vamos a definir las capas internas y su responsabilidad:

Nr	Tipo	Ubicación	Descripción y responsabilidad.
01	Entidad de Dominio	Domain	Es la encargada de contener todas las entidades del dominio de negocio como por ejemplo: Usuarios, Clientes, ventas, Productos. Su responsabilidad es modelar las entidades de bases de datos.
02	Puertos	Domain	Serán los encargados de conectar las capas e inyectar dependencias. Su responsabilidad es asegurar que los repositorios tengan una consistencia de métodos.
03	Servicio	Domain	Será el encargado de usar los repositorios para manipular la información. Su responsabilidad es ejecutar la lógica de los casos de uso.
04	Entidades de aplicación	Application	Se utilizará el patrón CQRS para modelar los récords y los handler. Cada 01 tendrá su

			<p>propia carpeta que a su vez va a contener 2 carpetas: “Commands” y “Querys”.</p> <p>Su responsabilidad será inyectar servicios y resolver lógica de operaciones con entidades.</p>
05	Adaptador	Infrastructure	<p>Son los repositorios e implementan el contrato pactado en 02.</p> <p>Su responsabilidad será hacer valer el contrato y ser inyectados.</p>
06	Conector a base de datos	Infrastructure	<p>Se encargará de usar la librería MEntityFrameworkCore para:</p> <ul style="list-style-type: none"> • Conectarse a la base de datos. • Registrar por cada entidad su respectiva tabla en base de datos. • <p>Su responsabilidad será conectar y modelar la base de datos.</p>