INFORME ASP.NET

2º DAW – Dual Eric Moros Pérez

Tabla de contenido

[Definición 2](#_Toc8856924)

[Herramientas 2](#_Toc8856925)

[Framework elegido para el informe 3](#_Toc8856926)

[Inicialización del proyecto 4](#_Toc8856927)

[Resumen de la operativa del proyecto 7](#_Toc8856928)

# Definición

Es el lenguaje de scripting del lado del servidor creado por Microsoft. ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje “Visual Basic Script” o “J Script” (JavaScript de Microsoft).



# Herramientas

Para realizar este informe y crear un proyecto en ASP.NET utilizaremos un IDE bastante conocido “*Microsoft Visual Studio 2017 Enterprise*”.

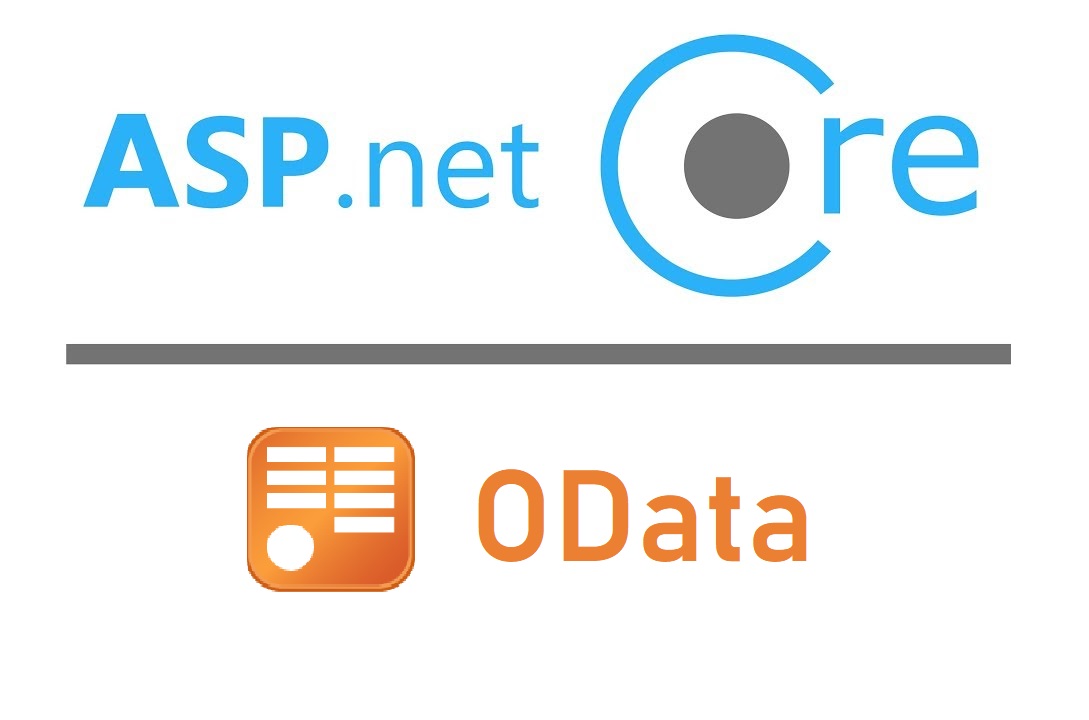
Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002).



# Framework elegido para el informe

ASP.NET contiene varios “frameworks” con los que organizar y agilizar nuestra forma de trabajar en nuestro proyecto, para este caso he elegido crear un proyecto API MVC 5 en .net Core 2.2.5 con un modelo con estructura “ncapas” con inyección de dependencias y ODATA 4. El acceso a datos se realizará contra una base de datos SQL Server.

ASP.NET MVC Es un “framework” que nos permite desarrollar aplicaciones web bastante robustas. Nos permite tener una clara separación entre la presentación de una web y su parte lógica. MVC significa: Modelo, Vista (View) y Controlador (Controller).

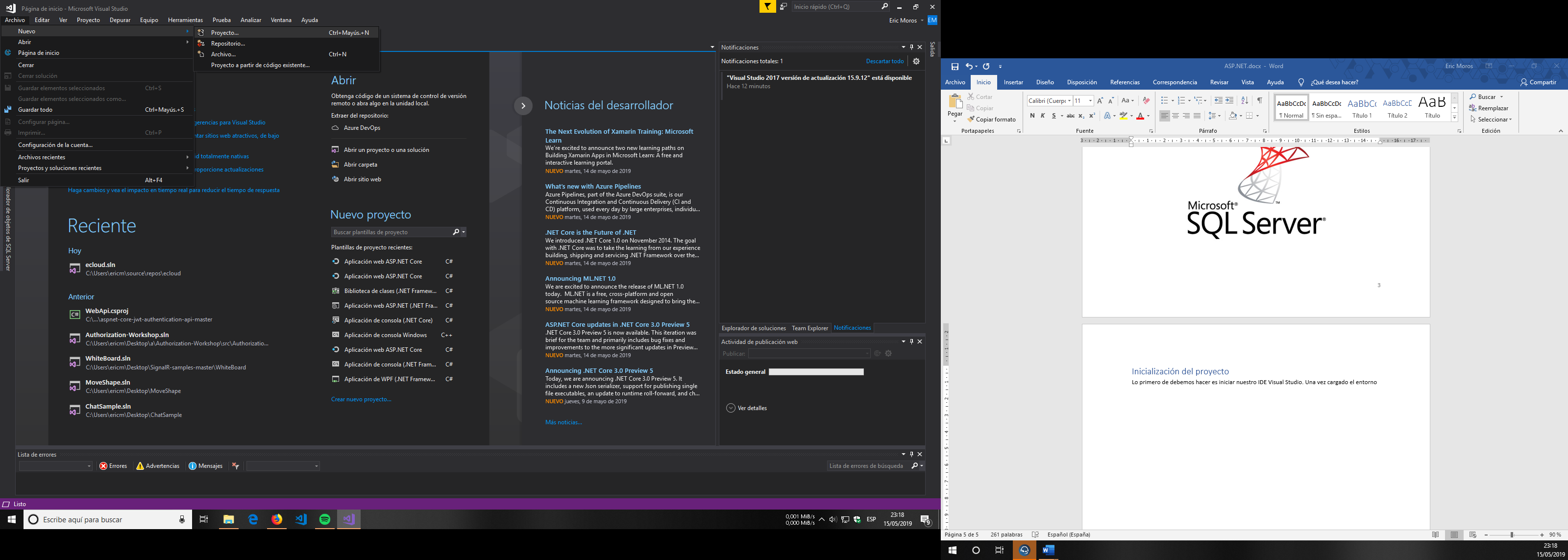




# Inicialización del proyecto

Lo primero de debemos hacer es iniciar nuestro IDE Visual Studio. Una vez cargado el entorno, crearemos un nuevo proyecto desde el menú.

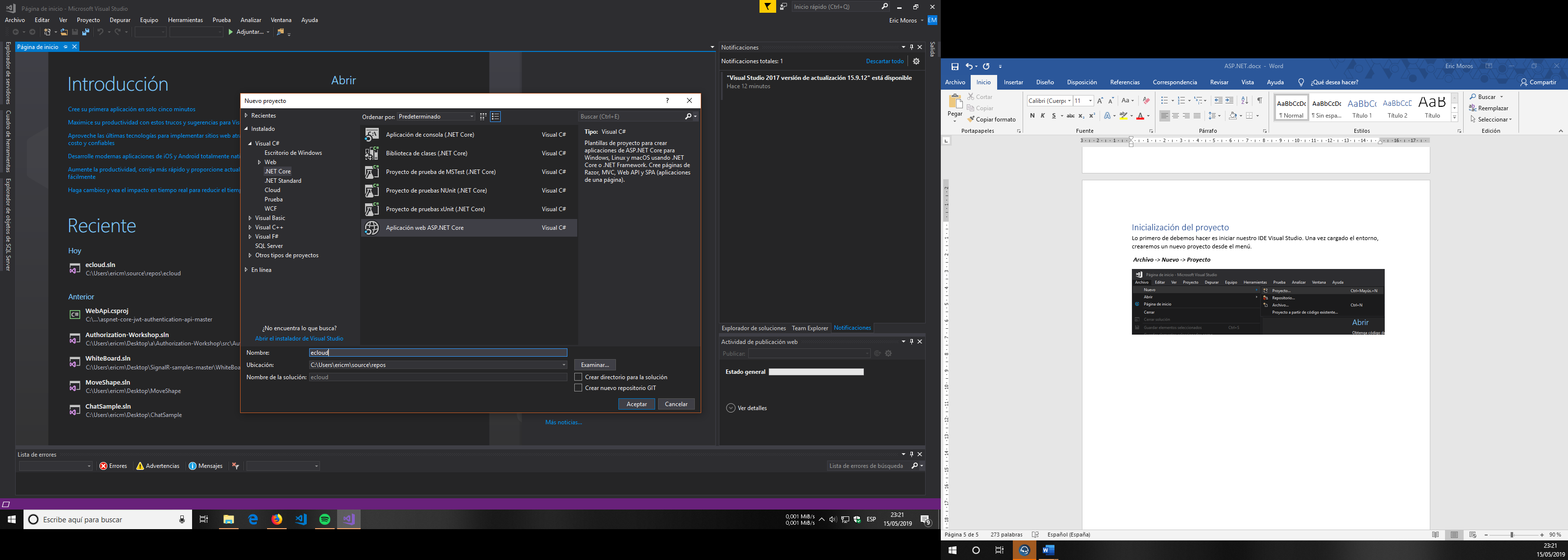
***Archivo -> Nuevo -> Proyecto***



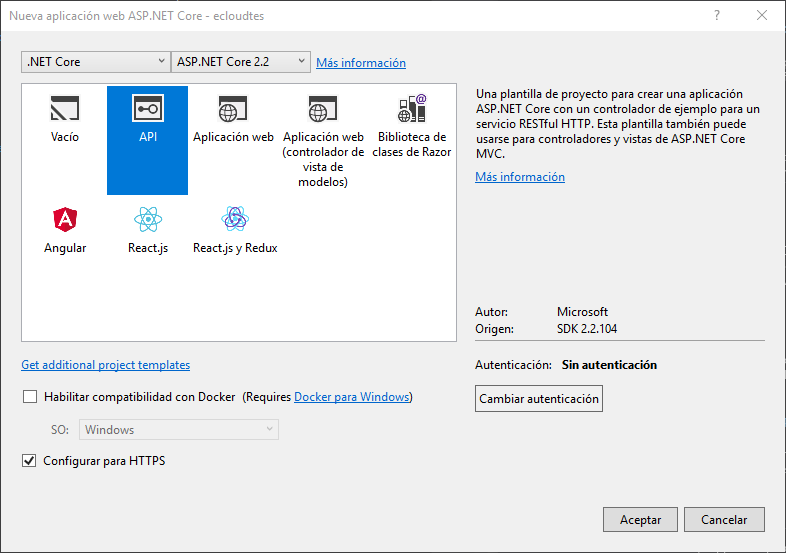
Seleccionamos **“*Aplicación web ASP.NET Core”*** e introducimos el nombre deseado a la aplicación.

Tenga en cuenta que para poder compilar y depurar este proyecto tedrá que tener instalado el SDK y Runtime de .NET Core 2.2, en caso de no tenerlo instalado podra encontrar su intalador en la siguiente página:

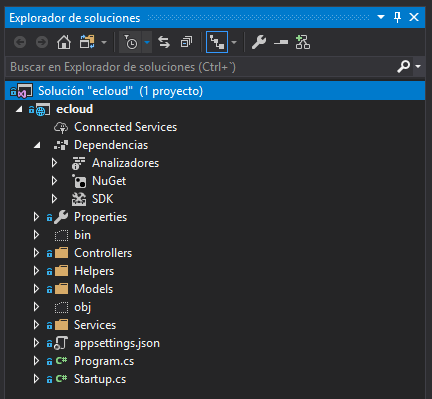
<https://dotnet.microsoft.com/download/dotnet-core/2.2>



Seguidamente seleccionaremos la planilla de proyecto ***“API”***



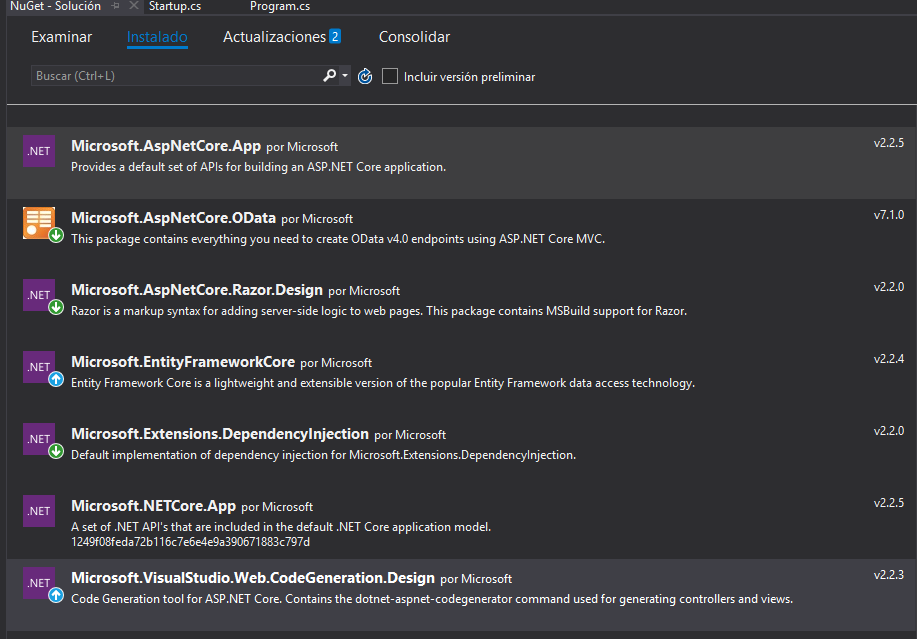
Si ha realizado los pasos correctamente comprobará que en el explorador de soluciones se visualizará la generación de su proyecto con una estructura similar al de la imagen de a continuación.

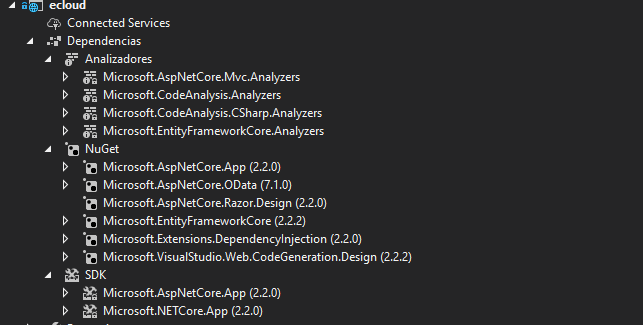


Para finalizar la inicialización instalaremos con “NuGet Manager“ todos los paquetes dependientes de nuestro proyecto:

* Microsoft.AspNetCore.App
* Microsoft.AspNetCore.OData
* Microsoft.AspNetCore.Razor.Design
* Microsoft.EntityFrameworkCore
* Microsoft.Extensions.DependencyInjection
* Microsoft.NETCore.App
* Microsoft.VisualStudio.Web.CodeGeneration.Design

Haga clic derecho en la solución y pinche en la opción ***“Administrar paquetes NuGet para la solución”*** importe todas las dependencias a la solución y tendría que acabar con un resultado parecido al de las 2 imágenes de a continuación.

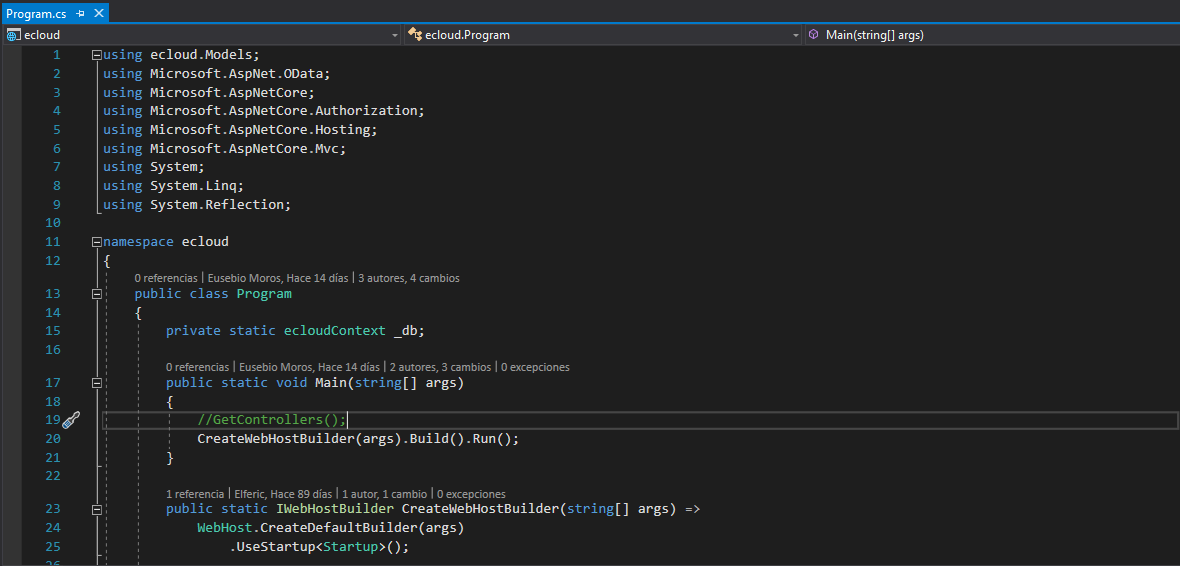




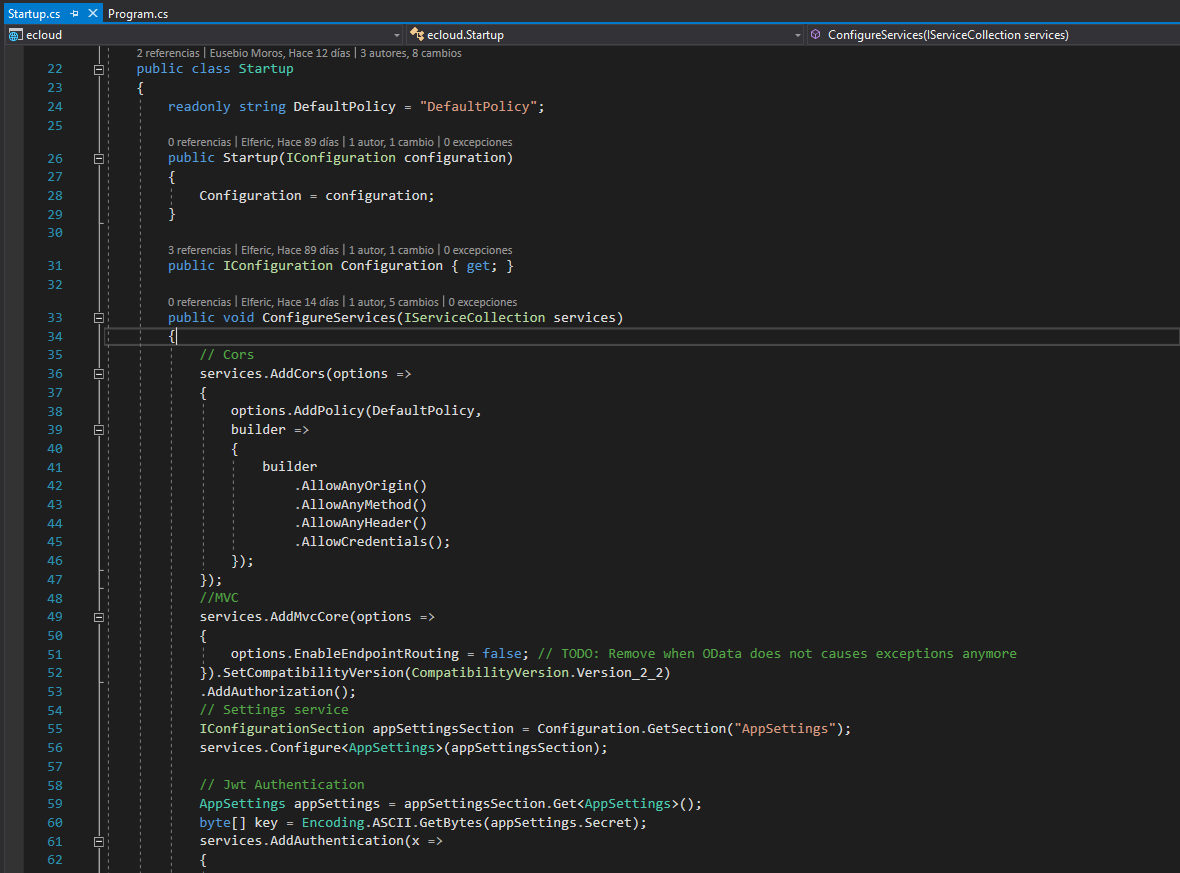
# Resumen de la operativa del proyecto

Archivos a tener en cuenta del proyecto.

* Program.cs: Esta será la clase principal del proyecto desde esta se arrancará el proyecto en un servidor IIS Express que Visual Studio trae preinstalado y preconfigurado.



* Startup.cs: Aquí es donde se definen varias configuraciones de los servicios de la aplicación, normalmente las dependencias se deben invocar desde esta clase, y todos los servicios con los que se vayan a operar.



* Appsettings.json: desde este archivo se podrán añadir algunas configuraciones para el servidor que ejecute la aplicación.

