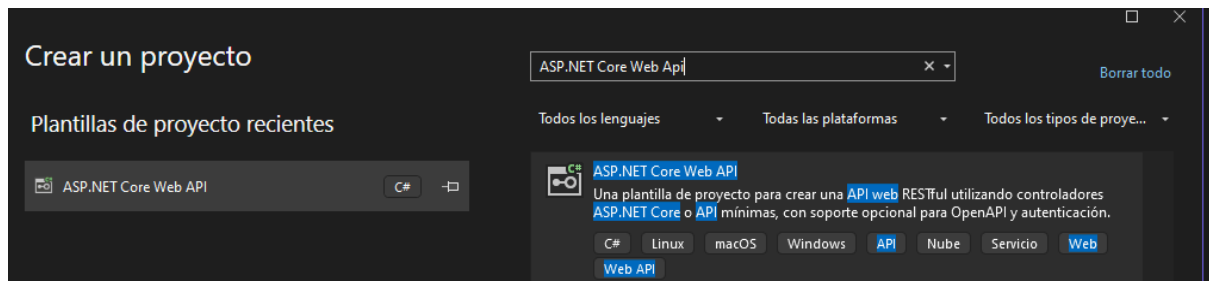


Taller .NET

Tarea_01

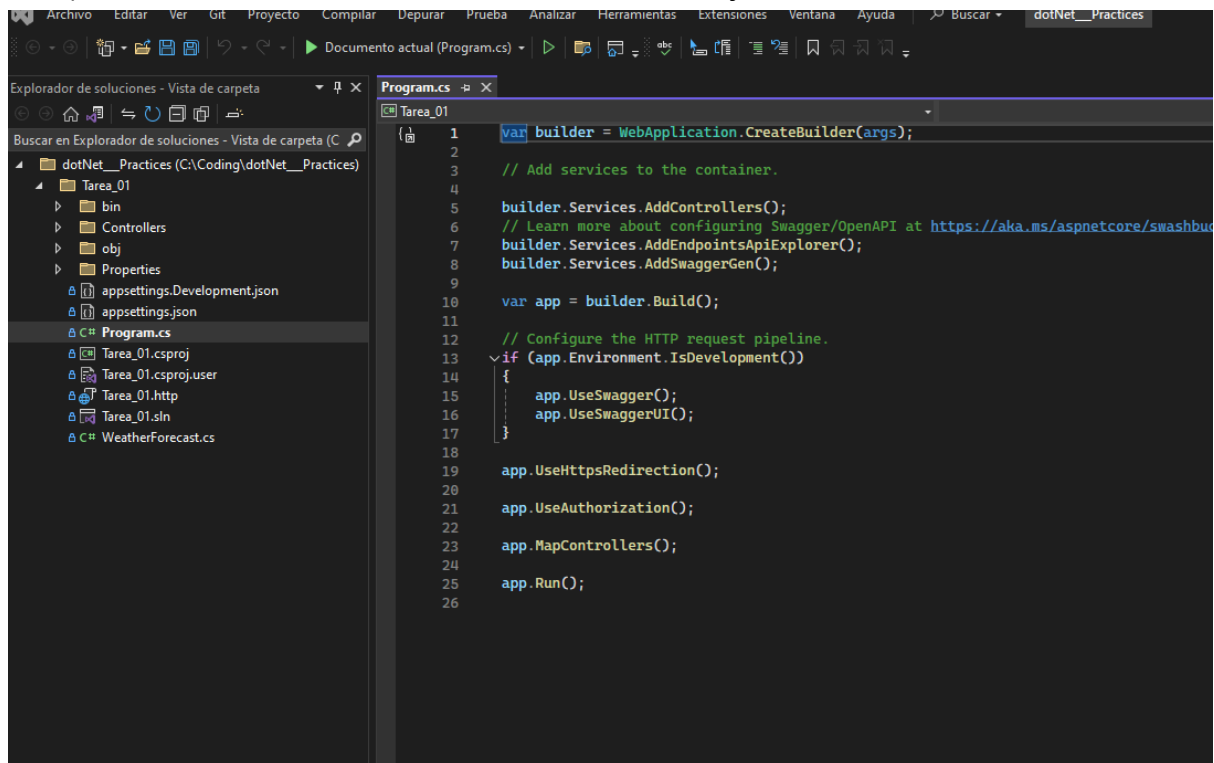
- 1) Primero procederemos a crear una plantilla de la misma manera que está documentado en el material del curso.a

Para ello abrimos el VS y elegimos la plantilla.

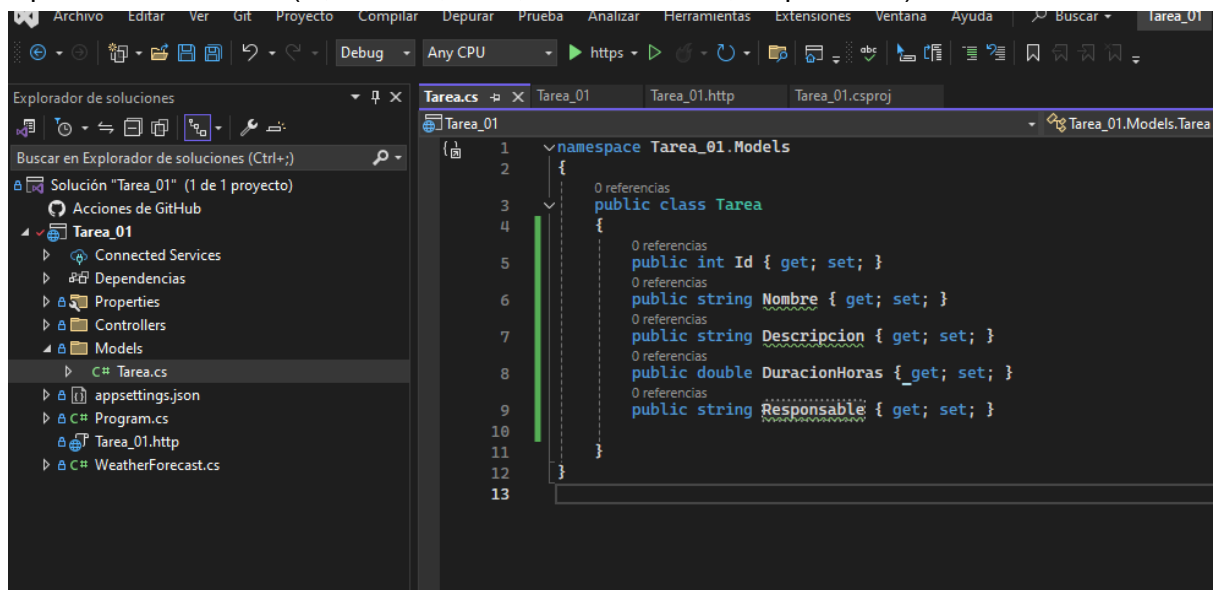


Luego de tener el proyecto creado podemos irnos referenciando con el material del curso.

- 2) Una vez tenemos nuestro ambiente de trabajo

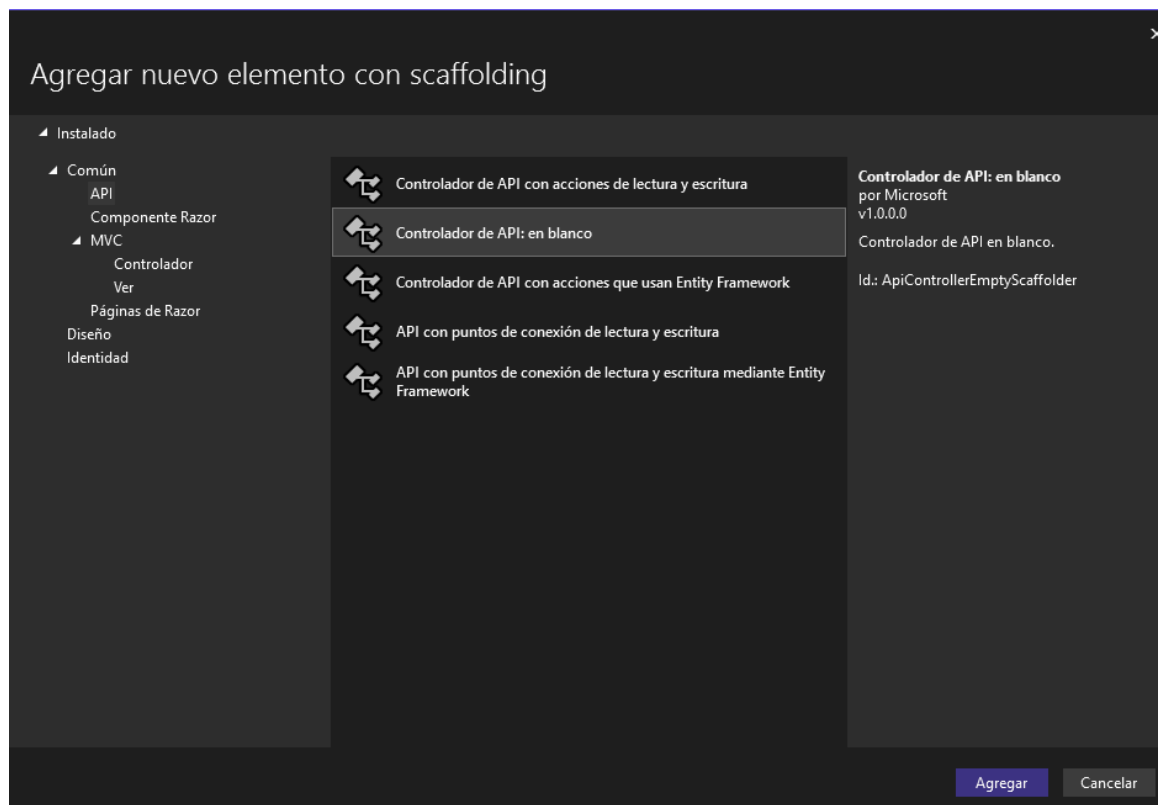


Procedemos a implementar dentro de la carpeta Models, una clase de dominio que represente una Tarea (id, nombre, desc, duración horas, responsable).



Procedemos a agregar la carpeta Models y crear la clase Tarea.cs
Dentro de la clase Tarea.cs creamos sus atributos con sus getters y setters.

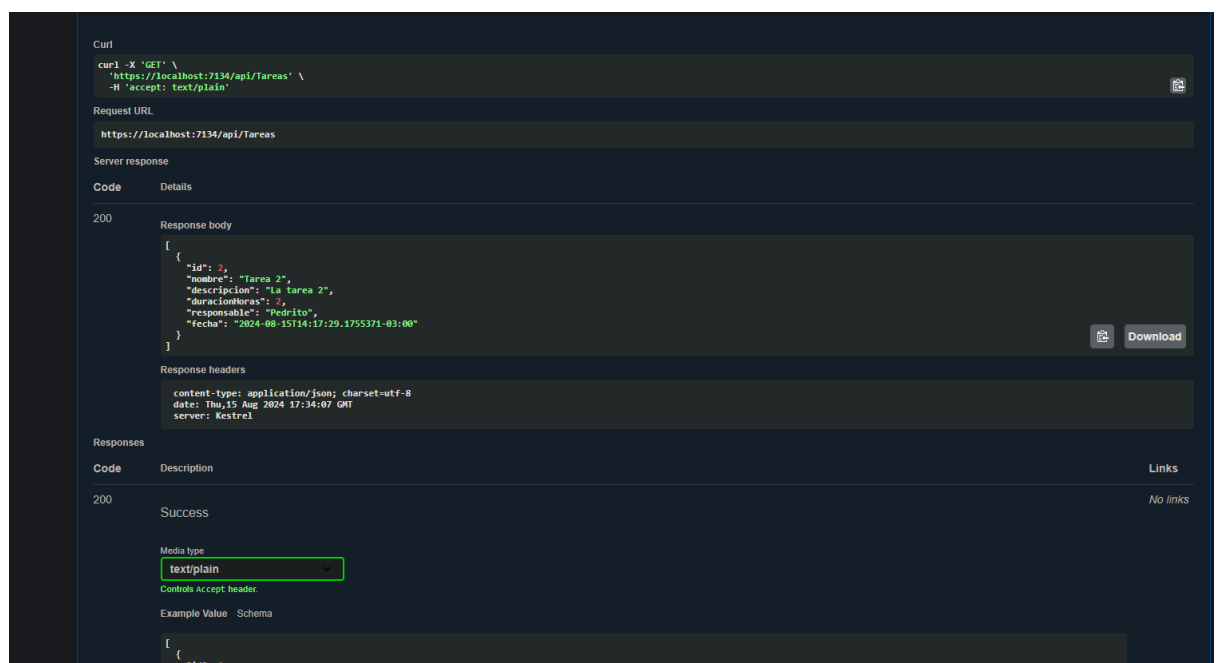
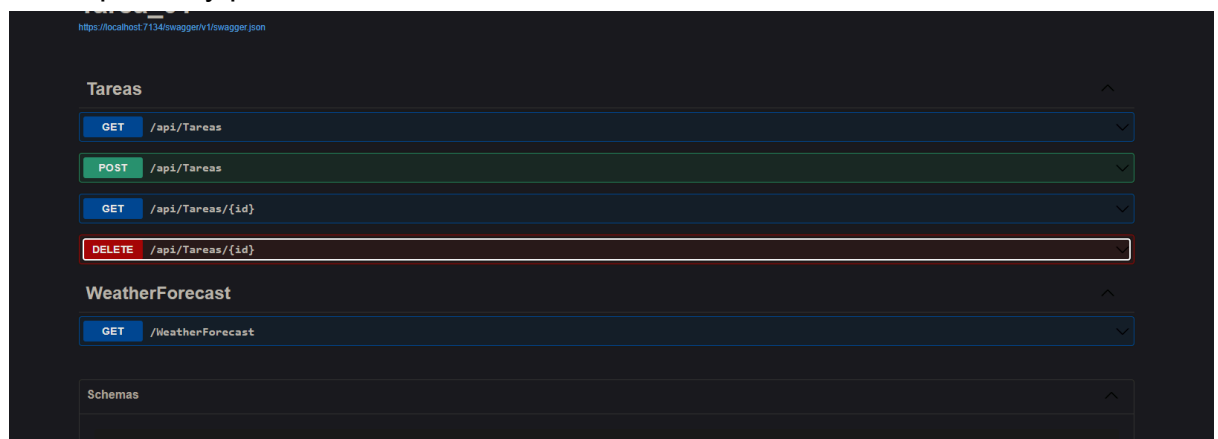
- 3) Como se aprecia en la anterior captura, podemos ver que hay atributos string, int y double.
- 4) Para este siguiente punto podemos ir a la carpeta de controladores, darle click derecho y luego le damos a “agregar”. De ahí seleccionamos uno API.

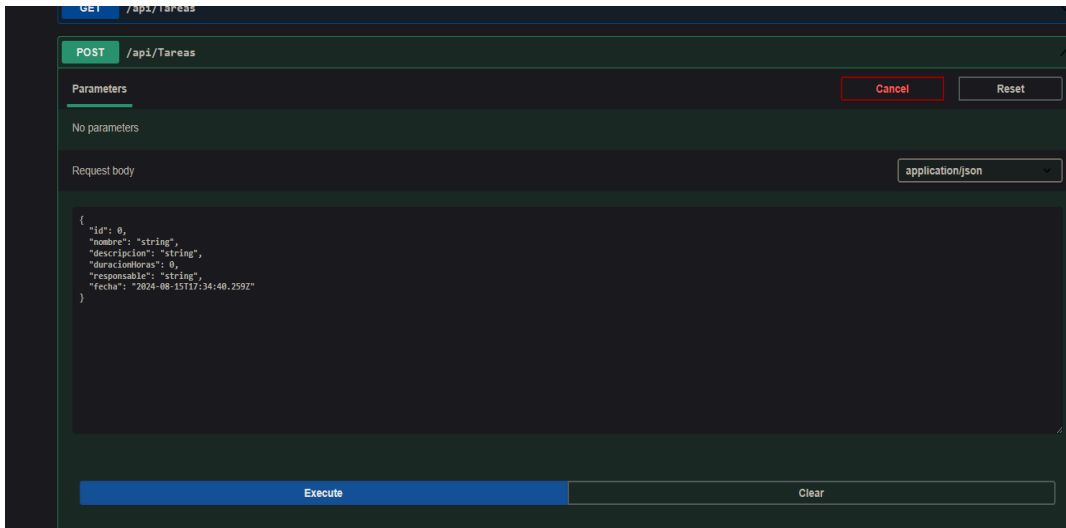


Luego de eso en el controlador, procedemos a crear una lista de tareas y cargarla con 2 tareas como se muestra en la siguiente imagen.

```
1 using Microsoft.AspNetCore.Http;
2 using Microsoft.AspNetCore.Mvc;
3 using Tarea_01.Models;
4
5 namespace Tarea_01.Controllers
6 {
7     [Route("api/[controller]")]
8     [ApiController]
9     public class TareasController : ControllerBase
10     {
11         private static List<Tarea> tareas = new List<Tarea>
12         {
13             new Tarea { Id = 1, Nombre = "Tarea 1", Descripcion = "La tarea 1", DuracionHoras = 1, Responsable = "Juansito" },
14             new Tarea { Id = 2, Nombre = "Tarea 2", Descripcion = "La tarea 2", DuracionHoras = 2, Responsable = "Pedrito" }
15         };
16     }
17 }
```

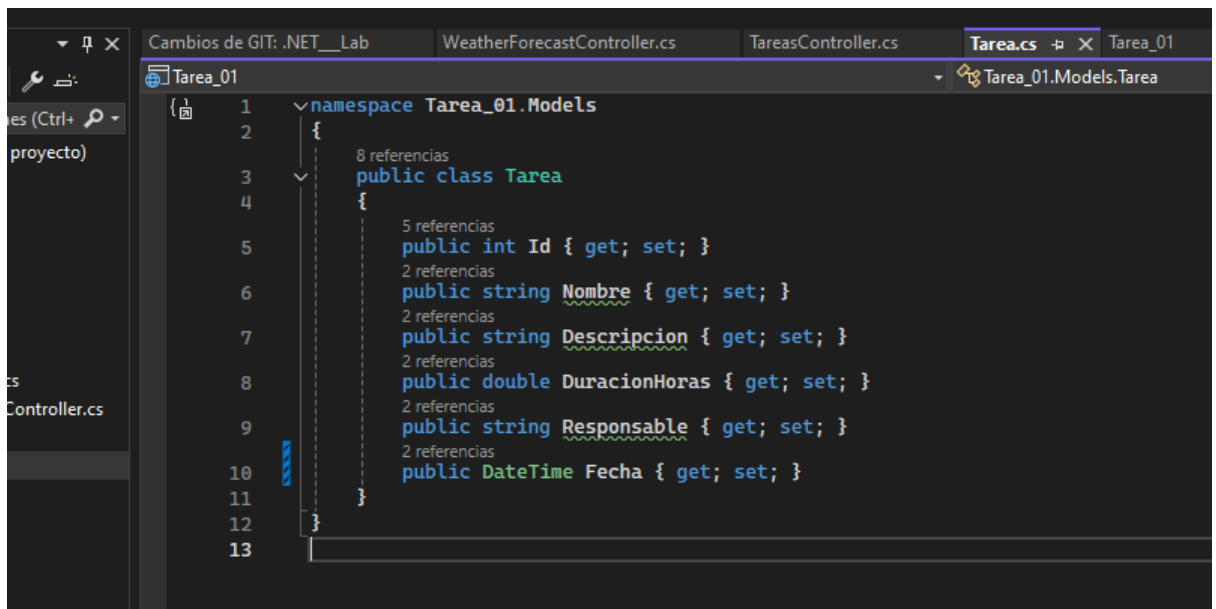
5) Seguido a implementar los endpoints, podemos proceder a levantar nuestra API expuesta y probar cada método con SWAGGER.



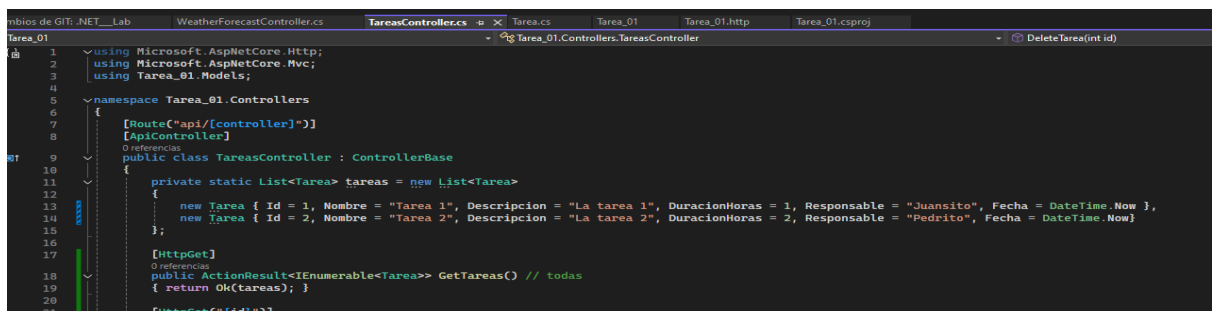


Podemos ver que todos estan funcionando correctamente.

6)



De esta manera le agregamos un atributo de fecha y podemos pre cargarlo tambien.



Repo de GitHub con el trabajo:

https://github.com/JuanmaPilon/dotNet___Practices