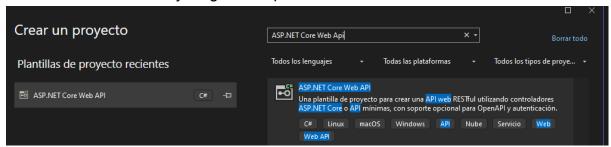
Taller .NET

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

```
節 + 🛎 🖺 📵 🖖 → ୯ → | ▶ Documento actual (Program.cs) → | ▶ | 📭 | 👼 👼 | 🌭 偱 | 厘 🧏 | 📵 匆 匆 匆 ↓
                                        ▼ ‡ X Program.cs ‡ X
  var builder = WebApplication.CreateBuilder(args);
                                                    { <u>}</u>
dotNet__Practices (C:\Coding\dotNet__Practices)
                                                                   // Add services to the container
  Tarea_01
    bin
Controllers
obj
                                                                   builder.Services.AddControllers();
                                                                                                               gger/OpenAPI at https://aka.ms/aspnetcore/swashb
                                                                   builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
     Properties
    △ 🕡 appsettings.Development.json
                                                                   var app = builder.Build():

△ (i) appsettings.json

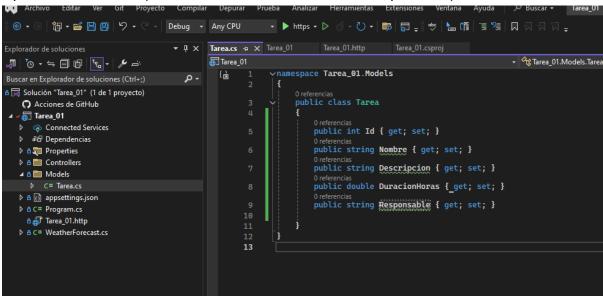
△ C# Program.cs

                                                                 // Configure the HTTP request pipeline
vif (app.Environment.IsDevelopment())
                                                           12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
   △ □ Tarea_01.csproj
   △ 🗟 Tarea_01.csproj.user
   △ ☐ Tarea_01.http
                                                                        app.UseSwagger();
app.UseSwaggerUI();

△ C# WeatherForecast.cs

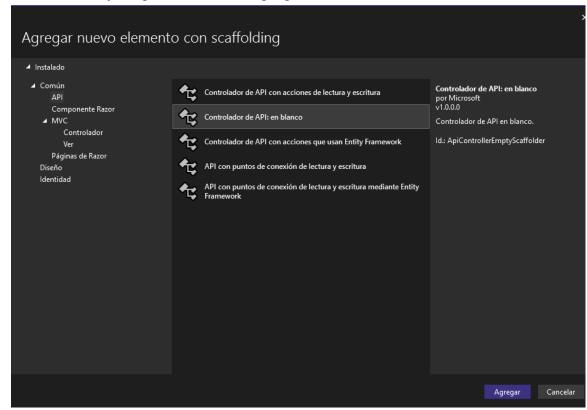
                                                                   app.UseHttpsRedirection();
                                                                   app.UseAuthorization();
                                                                   app.MapControllers();
                                                                   app.Run();
```

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.

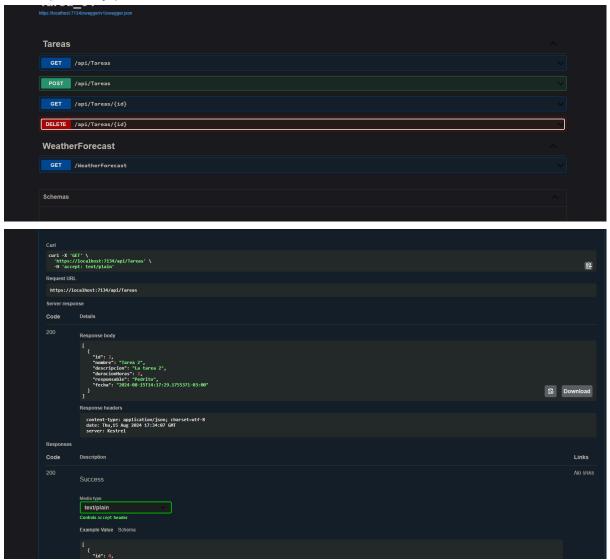
- 3) Como se aprecia en la anterior captura, podemos ver que hay atributos string, int y double.
- 4) Para este siguiente punto podemos ir a la carpeta de controladores, darle click derecho y luego le damos a "agregar". De ahí seleccionamos uno API.

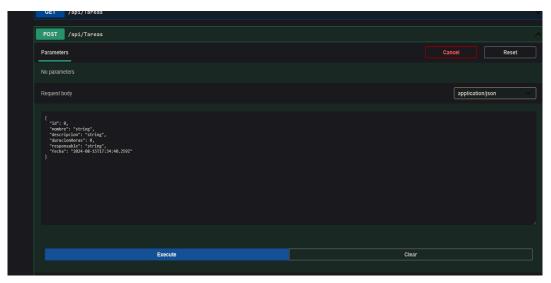


Luego de eso en el controlador, procedemos a crear una lista de tareas y cargarla con 2 tareas como se muestra en la siguiente imagen.

```
WeatherforcatController.cs v X Inrea.Controller.cs v X
```

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





Podemos ver que todos están funcionando correctamente.

```
Tarea.cs + X Tarea_01
               Tarea_01
                                                                                                             🕶 🕰 Tarea_01.Models.Tarea
                                 ∨namespace Tarea_01.Models
                  {a}
es (Ctrl+ 🔑 🕶
proyecto)
                                        8 referencias
public class Tarea
                                             5 referencias
public int Id { get; set; }
                                             2 referencias
public string Nombre { get; set; }
                                             2 referencias
public string Descripcion { get; set; }
                                             2 referencias
public double DuracionHoras { get; set; }
                                             2 referencias
public string Responsable { get; set; }
ontroller.cs
                                             2 referencias
public DateTime Fecha { get; set; }
                         13
```

De esta manera le agregamos un atributo de fecha y podemos pre cargarlo también.

```
Tarea_Ol. Tarea_
```

Repo de GitHub con el trabajo:

6)

https://github.com/JuanmaPilon/dotNet Practices