

## Práctica 2 - My First Web API

### Paso 1. Crea los objetos de la capa de dominio

- Inicia Visual Studio y Abre el proyecto **AnimalSpawn**.
- Selecciona el proyecto **AnimalSpawn.Domain**. y elimina la clase Class1.cs, que se encuentra en la raíz del proyecto.
- Dentro de la carpeta **Entities** agrega una clase con el nombre **Animal.cs** agrega las propiedades de acuerdo al diagrama de clases proporcionado por tu maestro (Para este primer ejercicio omitiremos las Asociaciones de la clase).

**Nota:** Recuerda que el snippet “Prop” puede ayudarte a crear de manera más rápida las propiedades en tu clase. También recuerda que una buena práctica a la hora de programar es remover las clausulas using que no se esten usando.

```
public class Animal
{
    public string Name { get; set; }
    public bool Gender { get; set; }
    public DateTime CaptureDate { get; set; }
    public string CaptureCondition { get; set; }
    public double Weight { get; set; }
    public double Height { get; set; }
    public int EstimatedAge { get; set; }
    public string Description { get; set; }
}
```

**Nota:** Recuerda que para poder acceder a una clase desde un proyecto externo, debemos marcar la clase con el modificador de acceso public.

### Paso 2. Crea los objetos para la capa de persistencia

- Selecciona el proyecto **AnimalSpawn.Infraestructure** y elimina la clase con el nombre Class1.cs que se encuentra en la raíz del proyecto.
- Dentro de la carpeta **Repositories** crea una nueva clase con el nombre **AnimalRepository.cs**.
- En la clase **AnimalRepository** crea un método con el nombre **GetAnimals**, el cual debe devolver una colección de animales (**IEnumerable<Animal>**).

**Nota:** Para poder usar la clase **Enumerable** se debe agregar la referencia a la clase **System.Linq** y debido a que estaremos usando un objeto de dominio también debemos agregar la referencia al ensamblado **AnimalSpawn.Domain.Entities**.

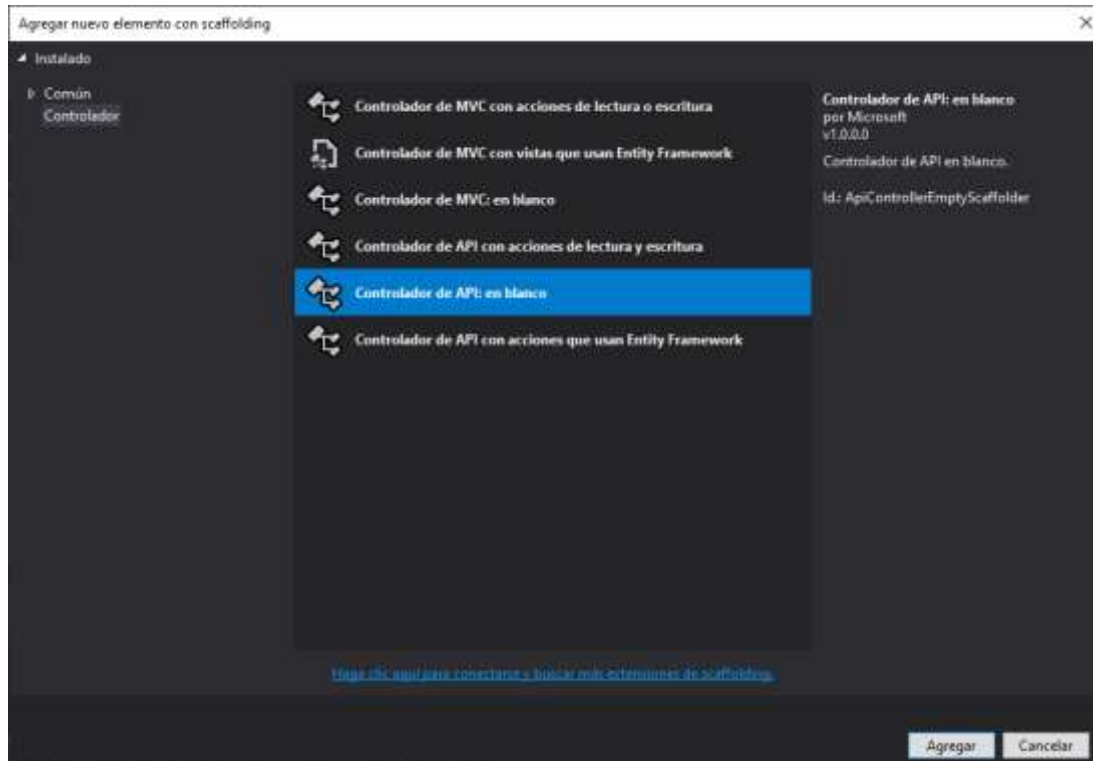
```
public class AnimalRepository
{
    public IEnumerable<Animal> GetAnimals()
    {
        var animals = Enumerable.Range(1, 10).Select(x => new Animal {
            Name = $"animal-{x}",
            CaptureCondition = "Good",
            CaptureDate = DateTime.Now,
            Description = $"Description of animal-{x}",
            EstimatedAge = (int)Math.Truncate(DateTime.Now.Minute * 2.5),
            Gender = x % 2 == 0,
            Height = Math.Round(DateTime.Now.Minute * 1.16, 2),
            Weight = Math.Round(DateTime.Now.Minute * 4.5, 2)
        });

        return animals;
    }
}
```

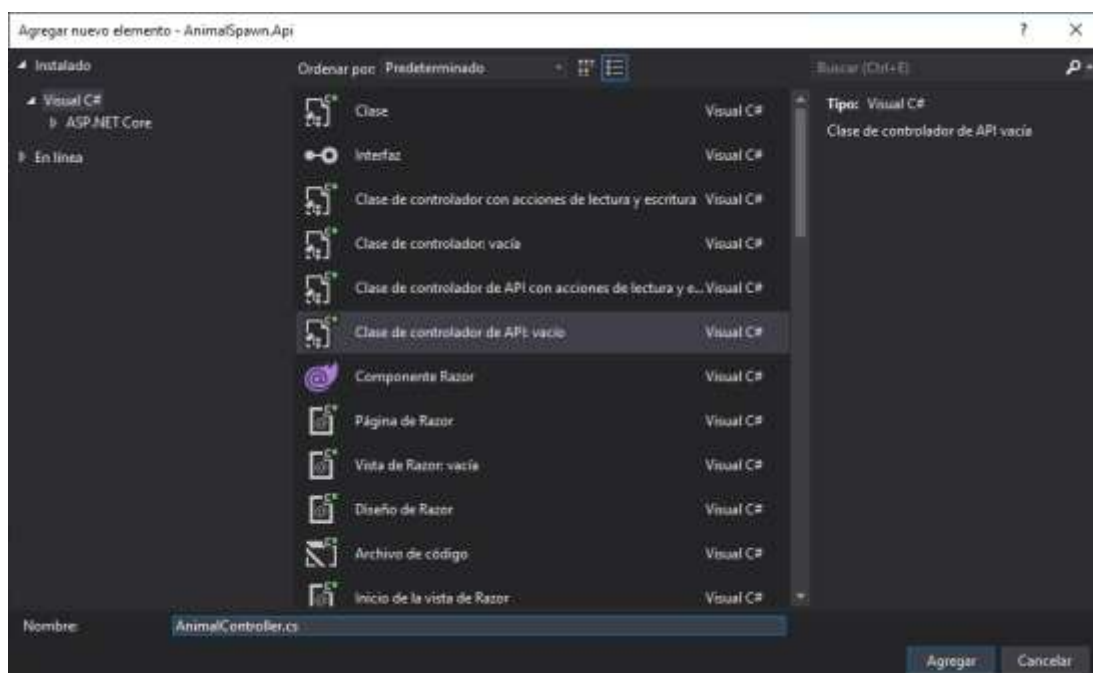
**Nota:** Recuerda que una buena práctica a la hora de programar es remover las clausulas using que no se esten usando.; también recuerda que para poder acceder a una clase desde un proyecto externo, debemos marcar la clase con el modificador de acceso public.

### Paso 3. Crear los objetos de la capa de presentación

- Selecciona el proyecto **AnimalSpawn.Api** y elimina la clase y el controlador de ejemplo que crea visual studio al momento de crear el proyecto web, recuerda que los controladores se ubican en la carpeta **Controllers**.
- Selecciona la carpeta **Controllers** y agrega un nuevo controlador al proyecto, para ello selecciona la opción API Controller – Empty



- Proporciona el nombre **AnimalController.cs** para el nombre del controlador.



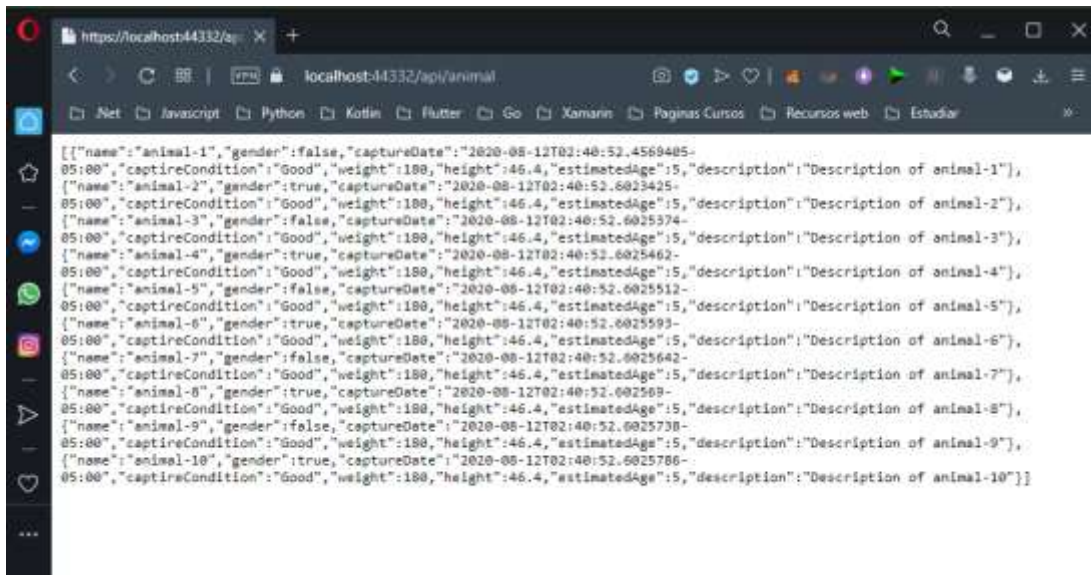
- En el controlador que recién se ha creado agrega un método con el nombre **Get** el cual debe retornar un objeto de tipo **IActionResult**, dentro del método invoca a la clase **AnimalRepository.cs** para retornar el listado de todos los animales; no olvides decorar el método con el verbo **HttpGet**.  
**Nota:** Para poder acceder a la clase **AnimalRepository**, debemos agregar a las clausulas using la referencia al ensamblado **AnimalSpawn.Infraestructure.Repositories**.

```
[Route("api/[controller]")]
[ApiController]
public class AnimalController : ControllerBase
{
    [HttpGet]
    public IActionResult Get() {
        var animals = new AnimalRepository().GetAnimal();
        return Ok(animals);
    }
}
```

**Nota:** Recuerda que una buena práctica a la hora de programar es remover las clausulas using que no se esten usando.; también recuerda que para poder acceder a una clase que se encuentra en un espacio de nombres diferente al de la clase que lo invoca, se debe agregar un using al espacio de nombres donde se encuentra la clase destino.

### Paso 5. Prueba el funcionamiento de tu aplicación

- Ejecuta tu aplicación y en el navegador de tu preferencia proporciona la siguiente ruta:  
<https://localhost:XXXXX/api/animal>



- Para tener una vista más legible ejecutaremos la consulta en el Postman, ejecuta la aplicación y crea una nueva petición Get a la ruta: <https://localhost:44332/api/animal>

NEW Runner Import Builder Team Library SYNC OFF

Chrome apps are being deprecated. [Download](#) our free native apps for continued support and better performance. [Learn more](#)

http://bitengine.ddns. http://bitengine.ddns. https://localhost:4433 + ... No Environment

GET https://localhost:44332/api/animal Pa

Key	Value	Description	...	Bulk Edit
New key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
33     "name": "animal-4",
34     "gender": true,
35     "captureDate": "2020-08-12T02:49:47.8783799-05:00",
36     "captureCondition": "Good",
37     "weight": 220.5,
38     "height": 56.839999999999996,
39     "estimatedAge": 5,
40     "description": "Description of animal-4"
41   },
42   {
43     "name": "animal-5",
44     "gender": false,
```

- Suerte y felices compilaciones