

# CLASE 2

## Laboratorio

---

### Creación de un WebApi simple

1. En la carpeta Descargas de la clase 2, ubique el archivo **Laboratorio\_WebApiEjemplo\_Inicial.zip**, descomprímalo en una carpeta y cargue el proyecto en Visual Studio. Este es el Proyecto inicial y punto de partida para realizar este ejercicio.
2. En la ventana **WebApiEjemplo - Microsoft Visual Studio** window, Seleccione el menú **DEBUG** menu, y clicken **Start Without Debugging**.
3. Observe la barra de direcciones del navegador, note el número de Puerto que aparece al final de la URL [http://localhost:\[port\]](http://localhost:[port]). Usaremos ese número de puerto durante el ejercicio.
4. Cierre el navegador.
5. En la ventana **WebApiEjemplo - Microsoft Visual Studio** , en el Solution Explorer, bajo **Properties**, ubique el archivo **launchSettings.json** y ábralo.
6. En la ventana de Código del archivo **launchSettings.json** , seleccione el siguiente código:

```
"launchUrl": "",
```

7. Reemplace el código seleccionado por este:

```
"launchUrl": "api/persona",
```

8. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, en el menú **DEBUG** , seleccione **Start Without Debugging**.

**Nota:** El navegador muestra el error **HTTP 404**.

9. Cierre el navegador.
10. En el Solution Explorer, click derecho en **WebApiEjemplo**, Seleccione **Add**, y luego click en **New Folder**.
11. En **NewFolder** escriba **Controllers**, y pulse Enter.

12. En la ventana **WebApiEjemplo - Microsoft Visual Studio** , en el Solution Explorer, click derecho en **Controllers**, seleccione **Add**, y luego seleccione **Controller** para crear un nuevo controlador.
13. En el cuadro de diálogo **Add Scaffold**, seleccione **API Controller - Empty**, y luego click en **Add**.
14. En el cuadro de diálogo **Add Empty API Controller** , en **Controller name** escriba **PersonaController** y luego click en **Add**.
15. En la ventana de código del controlador **PersonaController.cs** agregue el siguiente namespace a continuación de los existentes:

```
using WebApiExample.Models;
```

16. En la ventana de código del controlador **PersonaController.cs** , dentro de la clase y a continuación de la segunda { (llave que abre) escriba el siguiente código:

```
private List<Persona> _personas = new List<Persona>();

public PersonaController()
{
    _personas.Add(new Persona() {Id = 1, Nombre = " Juan ", Apellido = "Suarez" });
    _personas.Add(new Persona() {Id = 2, Nombre = " Jaime ", Apellido = "Manto" });
    _personas.Add(new Persona() {Id = 3, Nombre = " Jennifer ", Apellido = "Diaz" });
    _personas.Add(new Persona() {Id = 4, Nombre = " Vanesa ", Apellido = "Dupres" });
}
```

17. Debajo del constructor de **PersonaController** , agregue el siguiente código para la acción **HttpGet**:

```
[HttpGet]
public List<Persona> GetAll()
{
}
```

18. Dentro del código de la acción **GetAll**, escriba el siguiente código para indicar que se retornará la lista de personas:

```
return _personas;
```

19. Debajo de la acción **GetAll** escriba el siguiente código para otra acción **HttpGet** que retorne una persona desde su **Id**:

```
[HttpGet("{id}")]
public ActionResult<Persona> GetPersonaPorId(int id)
{
}
```

20. Dentro de la acción **GetPersonaPorId** escriba el siguiente código:

```
var persona = _personas.FirstOrDefault(p => p.Id == id);

if (persona == null)
{
    return NotFound();
}

return persona;
```

21. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, seleccione el menú **FILE** y click en **Save All**.

22. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, Seleccione el menú **DEBUG**, click en **Start Without Debugging**.

**Nota:** El navegador muestra la lista de personas en formato JSON.

23. En la barra de direcciones del navegador escriba esta URL correspondiente a un endpoint, tipee <http://localhost:port/api/persona/1>, y luego pulse Enter.

**Nota:** El navegador muestra la persona con Id 1 en formato JSON.

24. Cierre el navegador.

25. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, en el Solution Explorer, click derecho en **WebApiExample**, y luego click en **Manage NuGet Packages**.

26. En la ventana **NuGet Package Manager: WebApiEjemplo**, click en **Browse**.

27. En el cuadro de búsqueda, tipee **Microsoft.AspNetCore.Mvc.Formatters.Xml**, para buscar este paquete Nuget.

28. Ubique el paquete **Microsoft.AspNetCore.Mvc.Formatters.Xml**, Seleccione version **2.1.1**, y click en **Install** para instalarlo en el proyecto.

29. Si aparece el cuadro de diálogo **Preview Changes** , click en **OK**.

30. Si aparece el cuadro de diálogo **License Acceptance** , click en **I Accept**.

31. Cierre la ventana **NuGet Package Manager: WebApiEjemplo**.

32. En la ventana **WebApiEjemplo - Microsoft Visual Studio** , en el Solution Explorer, click en **Startup.cs**.

33. En la ventana de Código de **Startup.cs**, seleccione el siguiente código:

```
services.AddMvc();
```

34. Y reemplácelo por este otro, indicando que se agrega el formateador XML para MVC:

```
services.AddMvc().AddXmlSerializerFormatters();
```

35. En la ventana de código del controlador **PersonaController.cs**, en la acción **GetAll** y debajo de su atributo [HttpGet], agregue este otro atributo para esta acción indicando que el tipo de dato resultante de esta acción será XML:

```
[Produces("application/xml")]
```

36. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, en el menú **FILE**, click en **Save All**.

37. En la ventana **WebApiEjemplo - Microsoft Visual Studio**, Seleccione el menú **DEBUG**, click en **Start Without Debugging**

**Nota:** Ahora el navegador muestra la lista de personas en formato XML.