

# Unidad 6 – Capítulo 1 – Laboratorio 3

## Consumiendo los servicios de una Web API

### Objetivos

Consumir los servicios REST desde un cliente (aplicación de escritorio Windows Forms).

### Duración Aproximada

60 minutos

### Consignas

Crear una aplicación de escritorio Windows Forms para consumir los servicios REST disponibles en la aplicación ASP.NET Core construida en el Laboratorio de la Unidad sobre Servicios Web, donde continuaremos extendiendo el ejemplo de la entidad allí planteada.

La aplicación de escritorio debe contener uno o más formularios que permitan realizar las operaciones de ABMC / CRUD interactuando con los servicios REST disponibles.

### Pasos sugeridos

- 1) Utilizando Visual Studio, abrir la solución creada en el Laboratorio de construcción de Servicios Web, y agregar dentro de la misma un nuevo proyecto de tipo Windows Forms App.
- 2) Agregar en el proyecto de Windows Forms una referencia al proyecto “Dominio”, ya que será necesario utilizar la clase Alumno que allí reside.
- 3) Dentro del proyecto Windows Forms, crear uno o más formularios para realizar el ABMC / CRUD del Alumno utilizando los servicios REST expuestos en el proyecto Web API.

Para esto, cada formulario deberá contar con su propia instancia de la clase HttpClient para interactuar con la Web API; por ejemplo, declarar como variable de instancia del form:

```
private readonly HttpClient _httpClient = new()  
{  
    BaseAddress = new Uri("http://localhost:5000")  
};
```

**NOTA:** ajustar el valor del número de puerto en la URL (en el ejemplo, 5000) para que coincida con el número de puerto asignado por Visual Studio a su Web API.

El formulario principal (o el único que se cree para resolver el ABMC / CRUD completo) deberá contar con una grilla que muestre todos los Alumnos existentes; se sugiere cargar la grilla en el evento Load del form utilizando la instancia de HttpClient de la siguiente manera:

```
IEnumerable<Alumno> alumnos =  
    await _httpClient.GetFromJsonAsync<IEnumerable<Alumno>>("alumnos");  
this.dgvAlumnos.DataSource = alumnos;
```

- 4) Proseguir con la creación de las operaciones de Alta, Baja, Modificación y Consulta, recordando que se corresponden respectivamente con las operaciones POST, DELETE, PUT y GET

expuestas en la Web API. A modo de ejemplo, dentro del manejador del evento de un hipotético botón Aceptar de la operación de alta, podríamos tener un fragmento de código como éste:

```
Alumno alumno = new Alumno()
{
    Apellido = ...
    Nombre = ...
    Direccion = ...
    Legajo = ...
};

await _httpClient.PostAsJsonAsync("alumnos", alumno);
```