

## Taller de Lenguajes II – 2023

Programador Universitario / Licenciatura en informática / Ingeniería en Informática  
TP Nro 12

### CONTENIDOS

- ASP Web MVC
- ASP Layout
- Inserción de dependencias
- Validaciones

### INTRODUCCIÓN

Seguiremos trabajando con el repo del Trabajo Práctico Nro.10:

<https://classroom.github.com/a/Hj0gl71t>

En esta nueva iteración de trabajo, sobre el proyecto que venimos desarrollando vamos a agregar validación y las inyecciones de dependencias correspondientes para lograr un desarrollo más robusto.

1. Implemente validaciones en los ViewModels de:
  - CrearUsuario y ModificarUsuario
  - CrearTablero y ModificarTablero
  - CrearTarea y ModificarTablero

Para poder utilizar atributos de validación integrados recuerde qué:

- Se debe agregar las dependencias de validación a los ViewModels, es decir:  
`using System.ComponentModel.DataAnnotations;`
- Utilizar los atributos de validación correspondientes para cada propiedad del ViewModel.
- En las vistas (front-end) donde se quiera utilizar la validación se deben agregar el conjunto de librerías de javascript que validan los datos con las reglas del servidor. Para ello debe agregar la siguiente línea en cada vista  
`@section Scripts{<partial name="_ValidationScriptsPartial"/>}`
- Desde el servidor (back-end) (en los controllers) puede utilizar el objeto **ModelState.IsValid** para asegurar la validez de los campos enviados en los endpoint.

Ej.

```
[HttpPost]
public IActionResult CrearProducto(AltaProductoViewModel productoVM) {
    if(!ModelState.IsValid) return RedirectToAction("Index");
    //continua la definición del endpoint
}
```

Puede ingresar al siguiente link para ver la documentación sobre validaciones:

<https://learn.microsoft.com/es-es/aspnet/core/mvc/models/validation?view=aspnetcore-7.0>

## **Taller de Lenguajes II – 2023**

Programador Universitario / Licenciatura en informática / Ingeniería en Informática

TP Nro 12

2. Implemente Inyección de Dependencias para los repositorios creados anteriormente. Para ello cada repositorio deberá tener la interfaz correspondiente. El tipo de inyección de dependencias que utilizaremos será Scope.

**Nota:** Todos los repositorios deberán crearse en los controladores por inyección de dependencias.