

Taller de Lenguajes II – 2024

Programador Universitario / Licenciatura en informática / Ingeniería en Informática

TP Nro 7

CONTENIDOS

- ASP Web MVC
- ASP Layout
- Validaciones
- View Models

INTRODUCCIÓN

1) En el mismo repositorio del TP6 (<https://classroom.github.com/a/HCjszOX5>) cree una rama que se llame TP7 y trabaje en ella para el presente trabajo práctico

2) Refactorización del Sistema de Presupuestos

Se presentó un nuevo requisito: Que los Clientes tengan más información relacionada a los Clientes.

Para ello tendrá que incorporar la Tabla Clientes en la base de datos para almacenar información necesaria.

Define la tabla Cliente con los siguientes campos:

- Clienteld (int, clave primaria)
- Nombre (string)
- Email (string)
- Telefono (string)

Relaciona la tabla Clientes con la tabla Presupuestos para que cada presupuesto esté asociado a un cliente específico. Para ello en la tabla Presupuesto, añada el campo Clienteld (como clave externa). El campo NombreDestinatario se dejará de utilizar. (debe borrarlo de la tabla).

- En el sistema cree la clase Cliente con los mismos campos de la tabla Clientes
- En la clase presupuestos agregue el objeto Cliente como campo de la clase y borre el campo NombreDestinatario
- Incorpore el controlador para administrar los Clientes

3) Incorpore los ViewModels

- Para Presupuestos:** debe incluir una lista de clientes para poder elegirlos en el formulario de alta de presupuestos
- Para Alta de producto a presupuesto:** Incluir el listado de los productos para que se pueda seleccionar entre ellos.
- Para Productos:** Implementa validaciones en el modelo Producto para asegurar la integridad de los datos y evitar la inserción de registros incompletos o incorrectos.

4) Añadir las siguientes validaciones:

En Productos:

- **Descripción** (string): opcional, con una longitud máxima de 250 caracteres.
- **Precio** (decimal): requerido y debe ser un valor positivo.

Taller de Lenguajes II – 2024

Programador Universitario / Licenciatura en informática / Ingeniería en Informática
TP Nro 7

En Clientes:

- **Nombre (string)** es obligatorio
- **Email** (string) validar el tipo email
- **Telefono** (string) validar el tipo teléfono

Recordando validación:

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-8.0>

NOTA: Para correr la aplicación puede usar el comando

- **dotnet watch**

Este comando permite la recarga activa de la aplicación que está desarrollando.

Algunos links útiles

Introducción a bootstrap (layout de vistas):

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

Cheatsheet de razor:

<https://gist.github.com/jwill9999/655533b6652418bd3bc94d864a5e2b49#Index>

Más links útiles entre los recursos en la página de la cátedra.