**Desarrollador de Aplicaciones Web**

**Programación Web III**

**Guía Práctica : ASP.NET MVC**

1. Crear un proyecto de tipo Aplicación ASP.NET MVC y crear un formulario Web. En este formulario agregar una html input y un botón de type=submit. En el action retornar un mensaje de éxito que debe ser mostrado en un control HTML p dentro de la misma vista.
2. Crear un proyecto de tipo Aplicación ASP.NET MVC.
   1. Crear un controller llamado calculadora (recordar la convención de MVC).
   2. Agregar una vista con un form con los siguientes campos:
      1. Input con id “Operador1”
      2. Input con id “Operador2”
      3. Input type=submit
      4. Control HTML p con id “Resultado”
      5. Agregar un action a donde será recepcionado el submit del formulario para mostrar el mensaje en el control HTML p con id “Resultado”.
3. Crear un vista para registrar el alta de una nueva encuesta. La vista debe tener 2 inputs de tipo text (nombre y descripción) y 1 input de tipo check (campo “estoy de acuerdo”). Luego de crear la vista, crear un action de tipo Post en un controller de nombre EncuestasController que reciba un FormCollection. Acceder a la información de cada input y, luego, enviarla por querystring a un action MostrarEncuesta (dentro del mismo controller) que mostrará los datos ingresados en una nueva vista (el diseño de la vista queda a decisión del alumno).
4. Realizar el ejercicio anterior pero utilizando una clase Encuesta en lugar del FormCollection. ¿Qué diferencias hay? ¿Qué tienen en común ambos tipos de bindings?
5. Crear un controller con el nombre ProductosController, crear las siguientes actions y views para resolver los siguientes puntos:
   1. Vista para Crear un producto.
   2. Vista para Modificar un producto.
   3. Vista para Listar todos los productos almacenados.
   4. Vista para Eliminar un producto existente.

Comentarios:

1. Se debe utilizar una clase de nombre Producto con los siguientes atributos: id (int), nombre (string), cantidad (int), precio(decimal), fecha alta (dateTime).
2. Todos los datos indicados en el punto anterior, deben poder ser cargados por el usuario. Así como todos deben visualizarse en la vista que lista.
3. Para almacenar los productos que se van creando debe utilizarse una lista estáticas de tipo List<T> (esto será visto en clase con un ejemplo). Utilizar los métodos de Add, Remove para dar de alta, modificar o eliminar un producto).
4. No se debe utilizar FormCollection sino que se deben utilizar binding a través de clases.
5. Modificar el ejercicio del punto 5) para utilizar un ViewModel. El ViewModel contendrá el producto creado/modificado junto a una colección de tipos de productos. Los posibles tipos de productos son: Librería, Hogar, Tecnología y Deportes. Al momento de crear y modificar un producto el usuario debe elegir el tipo de producto asociado (Agregar un atributo a la clase Producto para el Tipo de Producto seleccionado).

En la pantalla de Listar sólo mostrar el ID de tipo de Producto.