

Gestión de Donaciones para una ONG

Apellido y Nombre:

DNI:

Realizar una aplicación "**Gestión de Donaciones**", en **ASP.NET MVC**, que incluya las siguientes funcionalidades y restricciones:

1) Inicio de la aplicación

- Al ejecutar la aplicación, esta debe iniciar por defecto en un controlador llamado **InicioController**, en una acción llamada **Saludo**.
 - La vista correspondiente debe incluir un mensaje de bienvenida al sitio:
"**Bienvenido al sistema de Gestión de Donaciones para ONG**"
-

2) Layout y navegación

2.1) Agregar un layout llamado **_ONGLayout.cshtml**, que debe incluir:

- Un **header fijo** con la siguiente información del alumno (visible en todas las páginas):
 - Nombre y Apellido
 - DNI

2.2) Incluir en la vista "Saludo" dos links:

- a) Registrar Donación
- b) Listado de Donaciones

La aplicación debe contar con un controlador **DonacionesController**, que tenga la lógica necesaria para mostrar las vistas según la opción elegida por el usuario:

- a) **Registrar**
 - b) **Listado**
-

3) Vista: Registrar Donación

La vista **Registrar Donación** debe permitir ingresar los siguientes campos:

- **Nombre del Donante**
- **Motivo de la Donación** (campo de texto)
- **Monto de Donación** (número decimal)

4) Validaciones requeridas

Aplicar las siguientes validaciones al formulario:

- **Nombre del Donante:** Campo obligatorio, máximo 50 caracteres.
- **Motivo de la Donación:** Campo obligatorio, máximo 100 caracteres.
- **Monto:** Obligatorio, mayor a 0 y menor o igual a 1.000.000

! No debe permitirse registrar donaciones que no cumplan con estas validaciones.

5) Vista: Listado de Donaciones

Debe mostrar todas las donaciones registradas con los siguientes datos:

- Nombre del Donante
- Motivo de la Donación
- Monto
- **Porcentaje de participación** respecto al total

⚠ El cálculo del porcentaje debe implementarse en un método separado para que refleje siempre el valor actualizado a medida que se agregan nuevas donaciones.

Aclaraciones finales

- El proyecto puede usar una lista en memoria (por ejemplo, una `static List<Donacion>` en una clase de lógica).

- El método de cálculo del porcentaje debe reutilizarse en ambas vistas (listado).
- Implementar validaciones con atributos ([Required], [StringLength], [Range], etc.).
- El header debe estar visible en todas las vistas (no usar footer).
- El código debe impedir el registro si hay errores de validación.

Entregar un archivo .zip con nombre:

Apellido-Nombre-Parcial-ONG.zip

- Incluir el proyecto completo.
- Excluir carpetas /bin y /obj.