**Instituto tecnológico Colonia CTC**

**DDA**

**Analista Programador**

**Obligatorio 1**

**Sistema gestor reservas**

**Docente: Agustín Maciel**

**Estudiante:**

**Pablo Marchetti**

**2024**

# Índice

[Índice 2](#_Toc181354645)

[Letra del problema 3](#_Toc181354646)

[Descripción del análisis y la solución 4](#_Toc181354647)

[Alcances obtenidos 5](#_Toc181354648)

[Diagrama de clases 7](#_Toc181354649)

[Casos de uso 7](#_Toc181354650)

[Desarrollo casos de uso 12](#_Toc181354651)

[Clases 14](#_Toc181354652)

[Ejercicio: 14](#_Toc181354653)

[Local: 15](#_Toc181354654)

[Maquina: 16](#_Toc181354655)

[Persona: 17](#_Toc181354656)

[Responsable: 18](#_Toc181354657)

[Rutina: 18](#_Toc181354658)

[RutinaEjercicio: 19](#_Toc181354659)

[RutinaSocio: 20](#_Toc181354660)

[Socio: 21](#_Toc181354661)

[TipoMaquina: 22](#_Toc181354662)

Patron de diseño utilizado ................................... **¡Error! Marcador no definido.**

Letra del problema Texto, Carta

Descripción generada automáticamente

# Descripción del análisis y la solución

**Gestión de Reservas**:

* + Manejo de reservas, incluyendo creación, modificación y cancelación.

**Información del Huésped**:

* + Registro detallado de los huéspedes (ID, nombre, apellidos, tipo de documento, etc.).

**Identificación de Habitaciones Ocupadas**:

* + Identificación de habitaciones ocupadas y el responsable de la reserva realizada a través del sistema.

**Detalles del Hotel**:

* + Información completa del hotel, incluyendo ubicación, características y disponibilidad de habitaciones.

**Gestión de Tarifas**:

* + Tarifas con fecha de vigencia hasta la actualización de una nueva.

**Solución Propuesta:**

**Base de Datos Relacional**:

* + Tablas para hoteles, habitaciones, huéspedes, reservas y tarifas.

**Consultas Dinámicas**:

* + Consultas SQL para buscar y reservar habitaciones.

**Interfaz de Usuario Intuitiva**:

* + Facilitar la consulta y reserva de habitaciones.

**Validación de Datos**:

* + Asegurar datos correctos y consistentes mediante validaciones.

**Resultado Esperado:**

* **Centralización**: Acceso fácil a datos de reservas, tarifas y ocupación.
* **Facilidad de Uso**: Experiencia intuitiva para usuarios.
* **Flexibilidad**: Adaptable a futuras necesidades.

# Alcances obtenidos

El equipo pudo concretar con éxito lo pautado por el docente, cumplimos con la parte de agregar, eliminar, modificar y listar los

modelos(Responsable,Local,Socio,TipoMaquina,Maquina,Rutina,Ejercicio,Rutin -aEjercicio,RutinaSocio), aplicamos los siguientes patrones de diseño

**Creador**:

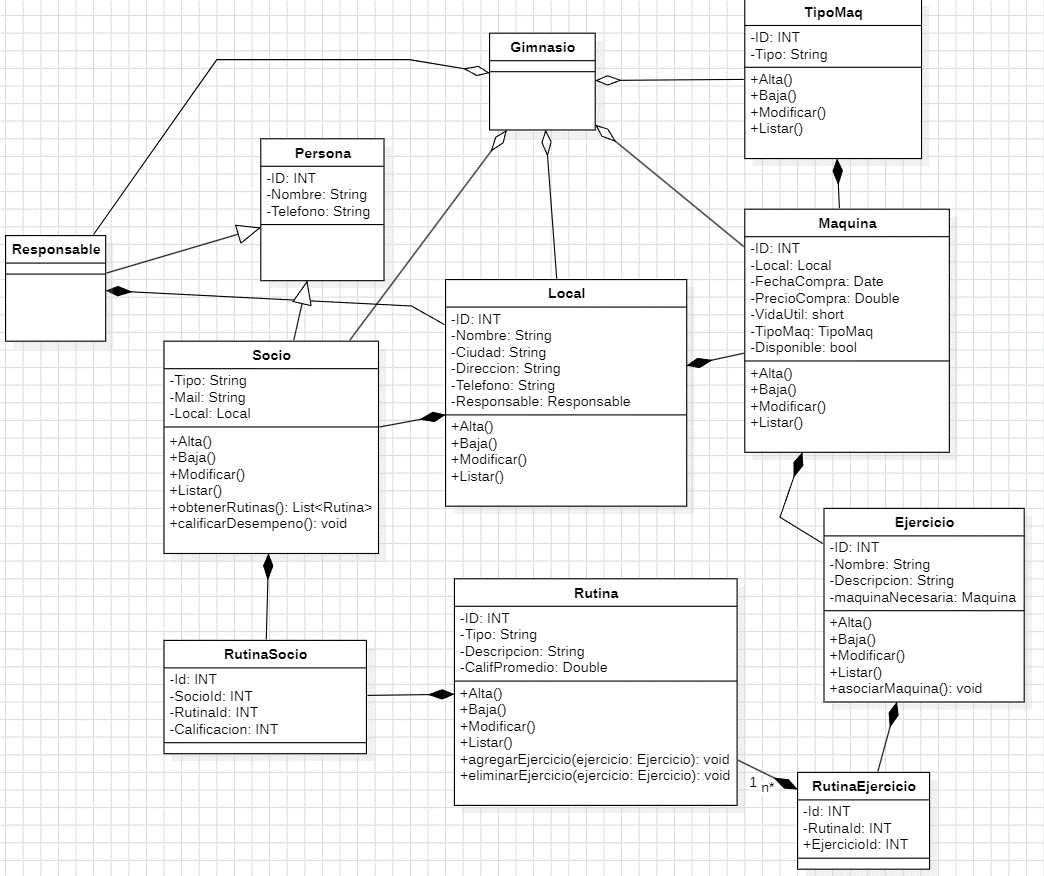
* **Definición**: Asigna la responsabilidad de crear una instancia de la clase A a la clase B si B contiene, agrega, utiliza, o tiene la información necesaria para crear instancias de A.
  + ApplicationDBContext crea instancias de DbSet para cada una de las entidades (Ejercicio, Local, Maquina, etc.). ApplicationDBContext necesita crear estas instancias para poder gestionar las operaciones CRUD en la base de datos.

**Alta Cohesión:**

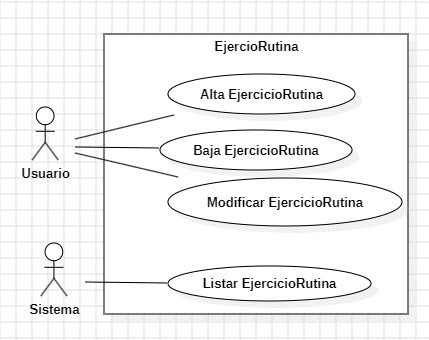
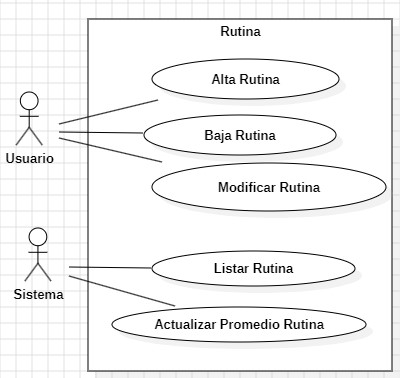
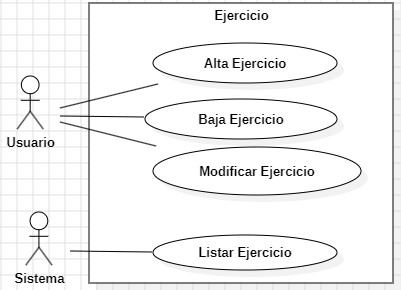
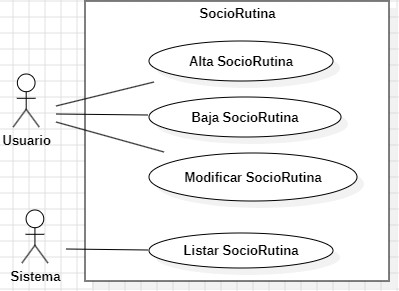
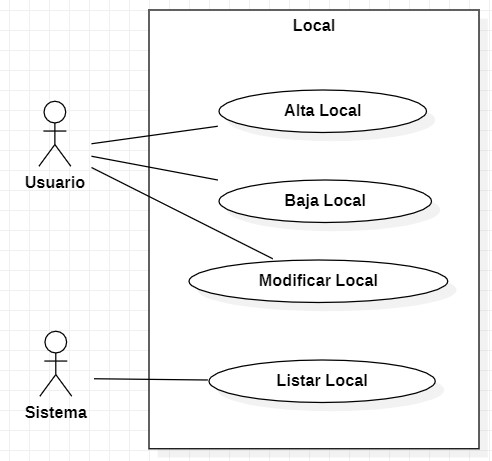
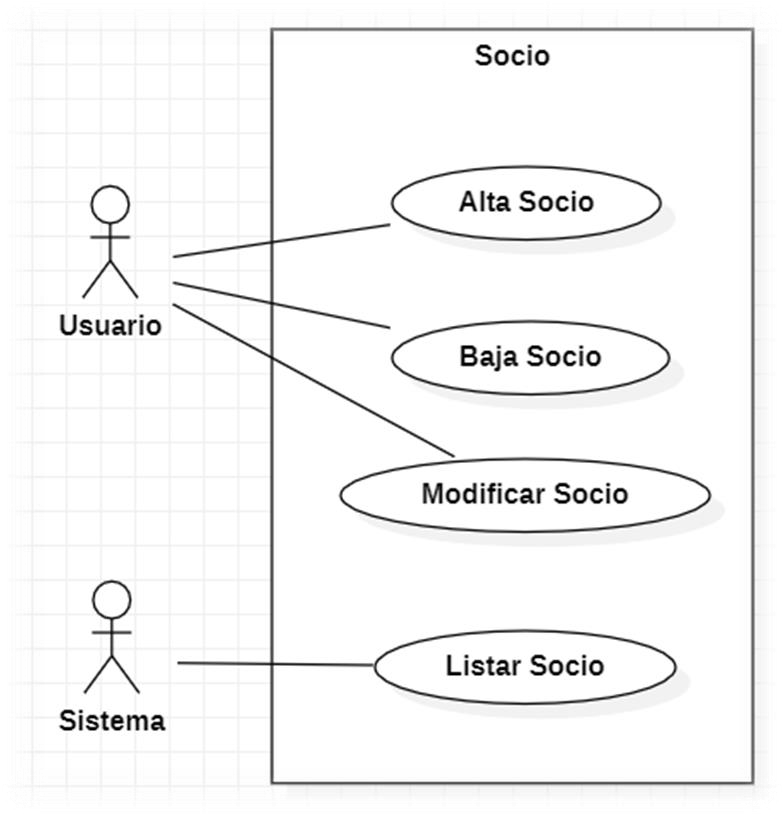
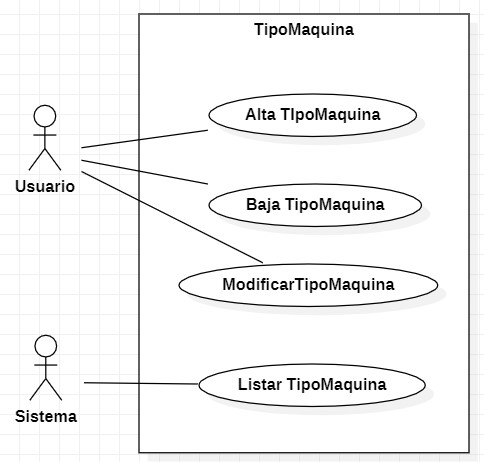
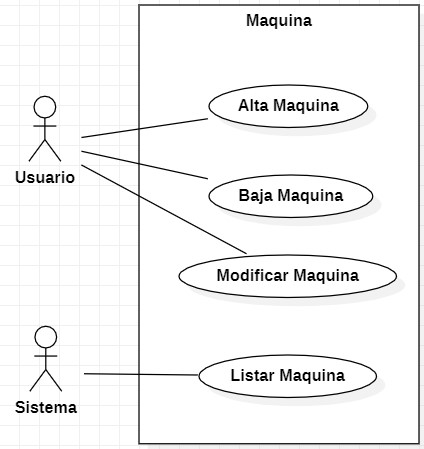
* **Definición**: Se garantizó que cada módulo o clase tuviera una responsabilidad específica y bien definida
  + ApplicationDBContext tiene una alta cohesión, ya que su única responsabilidad es gestionar la conexión a la base de datos y las configuraciones relacionadas. No se mezclan otras responsabilidades dentro de esta clase.

Y aplicamos la estructura de diseño dictada en clase

# Diagrama de clases



# Casos de uso



# Desarrollo casos de uso

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Alta socio |
| **Actores** | Usuario |
| **Descripción**  Permite dar de alta a los socios, los socios son personas ya agregadas al sistema que se harán “socios” de un local | |
| **Flujo normal**   * El Usuario selecciona la opción "Alta de Socio". * El sistema muestra un formulario para que el usuario complete los datos del nuevo socio (Tipo de socio, mail, local al que se asocia). * El usuario completa los datos y selecciona "Guardar". * El sistema valida los datos ingresados y muestra un mensaje de confirmación si el socio fue registrado exitosamente. | |
| **Flujo alternativo:**  2a. El sistema falla al cargar el listado de locales y muestra un error informando del problema.  3a. Si el empleado no completa algún campo obligatorio, el sistema muestra un mensaje de error y solicita que se completen los campos faltantes. | |
|  | |

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Alta maquina |
| **Actores** | Usuario |
| **Descripción:**  Permite dar de alta a las maquinas registrando su tipo, fecha de adquisición, y otros datos. | |
| **Flujo normal**   * El empleado selecciona la opción "Alta de Maquina". * El sistema muestra un formulario para que el Usuario complete los datos de la nueva maquina (local, fecha adquisición, precio, vida útil y disponibilidad). * El Usuario completa los datos y selecciona "Guardar". * El sistema valida los datos ingresados y muestra un mensaje de confirmación si la maquina fue registrada exitosamente | |
| **Flujo alternativo:**  2a. El sistema no carga el listado de tipo maquina y devuelve un mensaje notificando el error  3a. Si el empleado no completa algún campo obligatorio, el sistema muestra un mensaje de error y solicita que se completen los campos faltantes. | |

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Alta Rutina |
| **Actores** | Usuario |
| **Descripción**  Permite dar de alta a las rutinas | |
| **Flujo normal**   * El Usuario selecciona la opción "Alta de Rutina". * El sistema muestra un formulario para que el usuario complete los datos de la nueva rutina(Tipo, Descripcion, CalifPromedio). * El usuario completa los datos y selecciona "Guardar". * El sistema valida los datos ingresados y muestra un mensaje de confirmación si la rutina fue registrada exitosamente. | |
| **Flujo alternativo:**  2a. Si el empleado no completa algún campo obligatorio, el sistema muestra un mensaje de error y solicita que se completen los campos faltantes. | |
|  | |

|  |  |
| --- | --- |
| **Nombre del caso de uso** | Alta Local |
| **Actores** | Usuario |
| **Descripción:**  Permite dar de alta de los locales registrando, Nombre, Ciudad, y otros datos. | |
| **Flujo normal**   * El empleado selecciona la opción "Alta de Local". * El sistema muestra un formulario para que el Usuario complete los datos del nuevo Local (Nombre, Ciudad, Dirección, Teléfono y Responsable). * El Usuario completa los datos y selecciona "Guardar". * El sistema valida los datos ingresados y muestra un mensaje de confirmación si la maquina fue registrada exitosamente | |
| **Flujo alternativo:**  2a. El sistema no carga el listado de responsable y devuelve un mensaje notificando el error  3a. Si el empleado no completa algún campo obligatorio, el sistema muestra un mensaje de error y solicita que se completen los campos faltantes. | |

# Clases

## Ejercicio:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using System.Text.RegularExpressions;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class Ejercicio

{ [Key]

public int Id { get; set; }

[Required(ErrorMessage = "El campo Nombre es obligatorio")]

[MaxLength(50, ErrorMessage = "El nombre no puede contener más de 50 caracteres")] public string Nombre { get; set; }

[Required(ErrorMessage = "El campo Descripcion es obligatorio")]

[MaxLength(150, ErrorMessage = "La Descripcion no puede contener más de 150 caracteres")]

public string Descripcion { get; set; }

[ForeignKey("Maquina")]

public int IdMaquina { get; set; }

[ValidateNever]

public Maquina Maquina { get; set; }

[NotMapped]

public List<SelectListItem> MaquinasPosibles { get; set;

}

[ValidateNever]

public List<RutinaEjercicio> RutinaEjercicios { get; set;

}

}

}

## Local:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using System.Text.RegularExpressions;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering; using Microsoft.EntityFrameworkCore;

namespace Obligatorio2Programacion3.Models

{

public class Local

{ [Key]

public int Id { get; set; }

[Required(ErrorMessage = "El campo nombre es obligatorio")] [MaxLength(50, ErrorMessage = "El nombre no puede exceder los 50 caracteres")]

public string Nombre { get; set; }

[Required(ErrorMessage = "El campo ciudad es obligatorio")] [MaxLength(50, ErrorMessage = "La ciudad no puede exceder los 50 caracteres")]

public string Ciudad { get; set; }

[Required(ErrorMessage = "El campo direccion es obligatorio")] [MaxLength(100, ErrorMessage = "La direccion no puede exceder los 100 caracteres")]

public string Direccion { get; set; }

[Required(ErrorMessage = "El campo telefono es obligatorio")]

[MaxLength(20, ErrorMessage = "El telefono no puede exceder los

20 caracteres")]

public string Telefono { get; set; }

[Required(ErrorMessage = "El campo Responsable es Obligatorio")]

[ForeignKey("Responsable")]

public int IdResponsable { get; set; }

[ValidateNever]

public Responsable Responsable { get; set; }

[ValidateNever]

public List<Socio> Socios { get; set; }

[NotMapped]

public List<SelectListItem> ResponsablesPosibles;

}

}

## Maquina:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema;

using System.Text.RegularExpressions;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation;

using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class Maquina

{ [Key]

public int Id { get; set; }

[ForeignKey("Local")]

[Required(ErrorMessage = "el local es obligatoria")] public int IdLocal { get; set; }

[ValidateNever]

public Local Local { get; set; }

[Required(ErrorMessage = "el FechaCompra es obligatoria")] public DateTime FechaCompra { get; set; }

[Required(ErrorMessage = "el PrecioCompra es obligatoria")]

public int PrecioCompra { get; set; }

[Required(ErrorMessage = "el VidaUtil es obligatoria")] public int VidaUtil { get;set; }

[ForeignKey("TipoMaquina")]

[Required(ErrorMessage = "el tipo de maquina es obligatoria")]

public int IdTipoMaquina { get; set; }

[ValidateNever]

public TipoMaquina TipoMaquina { get; set; }

[Required(ErrorMessage = "el Disponible es obligatoria")] public bool Disponible { get; set; }

[NotMapped]

public List<SelectListItem> LocalesDisponibles;

[NotMapped]

public List<SelectListItem> TipoMaquinasDisponibles;

[ValidateNever]

public List<Ejercicio> Ejercicios{ get; set; }

}

}

## Persona:

using System.ComponentModel.DataAnnotations;

namespace Obligatorio2Programacion3.Models

{

public abstract class Persona

{ [Key]

public int Id { get; set; }

[Required(ErrorMessage = "El nombre es Requerido")]

public string Nombre { get; set; }

[Required(ErrorMessage = "El Telefono es Requerido")]

public string Telefono { get; set; }

}

}

## Responsable:

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering; using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema;

namespace Obligatorio2Programacion3.Models

{

public class Responsable : Persona

{

[ValidateNever]

public List<Local> Locales { get; set; }

}

}

## Rutina:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using System.Text.RegularExpressions;

using

Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class Rutina

{ [Key]

public int Id { get; set; }

[Required(ErrorMessage = "El campo Tipo es obligatorio")]

[MaxLength(50, ErrorMessage = "El tipo no puede contener más de 50 caracteres")] public string Tipo { get; set; }

[Required(ErrorMessage = "El campo

Descripcion es obligatorio")]

[MaxLength(150, ErrorMessage = "La Descripcion no puede contener más de 150 caracteres")]

public string Descripcion { get; set; }

public double CalifPromedio { get; set; }

[ValidateNever]

public List<RutinaEjercicio>

RutinaEjercicios { get; set; }

[ValidateNever]

public List<RutinaSocio> RutinaSocio { get; set; } }

## RutinaEjercicio:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using System.Text.RegularExpressions;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class RutinaEjercicio

{

public int Id { get; set; }

[ForeignKey("Rutina")]

[Required(ErrorMessage = "La Rutina es obligatoria")] public int IdRutina { get; set; }

[ValidateNever]

public Rutina Rutina { get; set; }

[ForeignKey("Ejercicio")]

[Required(ErrorMessage = "El Ejercicio es obligatorio")] public int IdEjercicio { get; set; }

[ValidateNever]

public Ejercicio Ejercicio { get; set; }

[NotMapped]

public List<SelectListItem> RutinasPosibles { get; set; }

[NotMapped]

public List<SelectListItem> EjerciciosPosibles { get; set; } }

}

## RutinaSocio:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using System.Text.RegularExpressions;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class RutinaSocio

{

public int Id { get; set; }

[ForeignKey("Socio")]

[Required(ErrorMessage = "El socio es obligatorio")] public int IdSocio { get; set; }

[ValidateNever]

public Socio Socio { get; set; }

[ForeignKey("Rutina")]

[Required(ErrorMessage = "La rutina es obligatoria")]

public int IdRutina { get; set; }

[ValidateNever]

public Rutina Rutina { get; set; }

[Required(ErrorMessage = "La calificacion es obligatoria")]

[Range(1, 5, ErrorMessage = "La calificacion debe ser entre 1 y 5")]

public int Calificacion { get; set; }

[NotMapped]

public List<SelectListItem> RutinasPosibles;

[NotMapped]

public List<SelectListItem> SociosPosibles;

[ValidateNever]

public List<RutinaEjercicio> RutinaEjercicios { get; set; } }

}

## Socio:

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class Socio:Persona

{

[Required]

[MaxLength(20 , ErrorMessage = "El tipo no puede exceder los 20 caractres")]

public string Tipo { get; set; }

[Required]

[EmailAddress]

[MaxLength(100, ErrorMessage = "El correo no puede exceder los 100 caracteres")]

public string Mail { get; set; }

[ForeignKey("Local")]

[Required(ErrorMessage = "El Local es obligatorio")] public int IdLocal { get; set; }

[ValidateNever]

public Local Local { get; set; }

[NotMapped] [ValidateNever]

public List<SelectListItem> LocalesPosibles { get; set; }

[ValidateNever]

public List<Local> Locales { get; set; }

[ValidateNever]

public List<RutinaSocio> RutinaSocios { get; set; }

/\*Estrategia tpt\*/

}

}

## TipoMaquina:

using System.ComponentModel.DataAnnotations; using System.ComponentModel.DataAnnotations.Schema; using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; using Microsoft.AspNetCore.Mvc.Rendering;

namespace Obligatorio2Programacion3.Models

{

public class TipoMaquina

{ [Key]

public int Id { get; set; }

[Required(ErrorMessage = "El Campo tipo es obligatorio")]

[MaxLength(50, ErrorMessage = "No puede exceder los

50 caracteres")]

public string Tipo { get; set; }

[ValidateNever]

public List<Maquina> Maquinas { get; set; }

}

}