

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 2

DOCUMENTO DE ANÁLISIS



Andrés Alvarez – 295384

69195-MAR24-P2 – V2A

Docente: Pablo Guillen

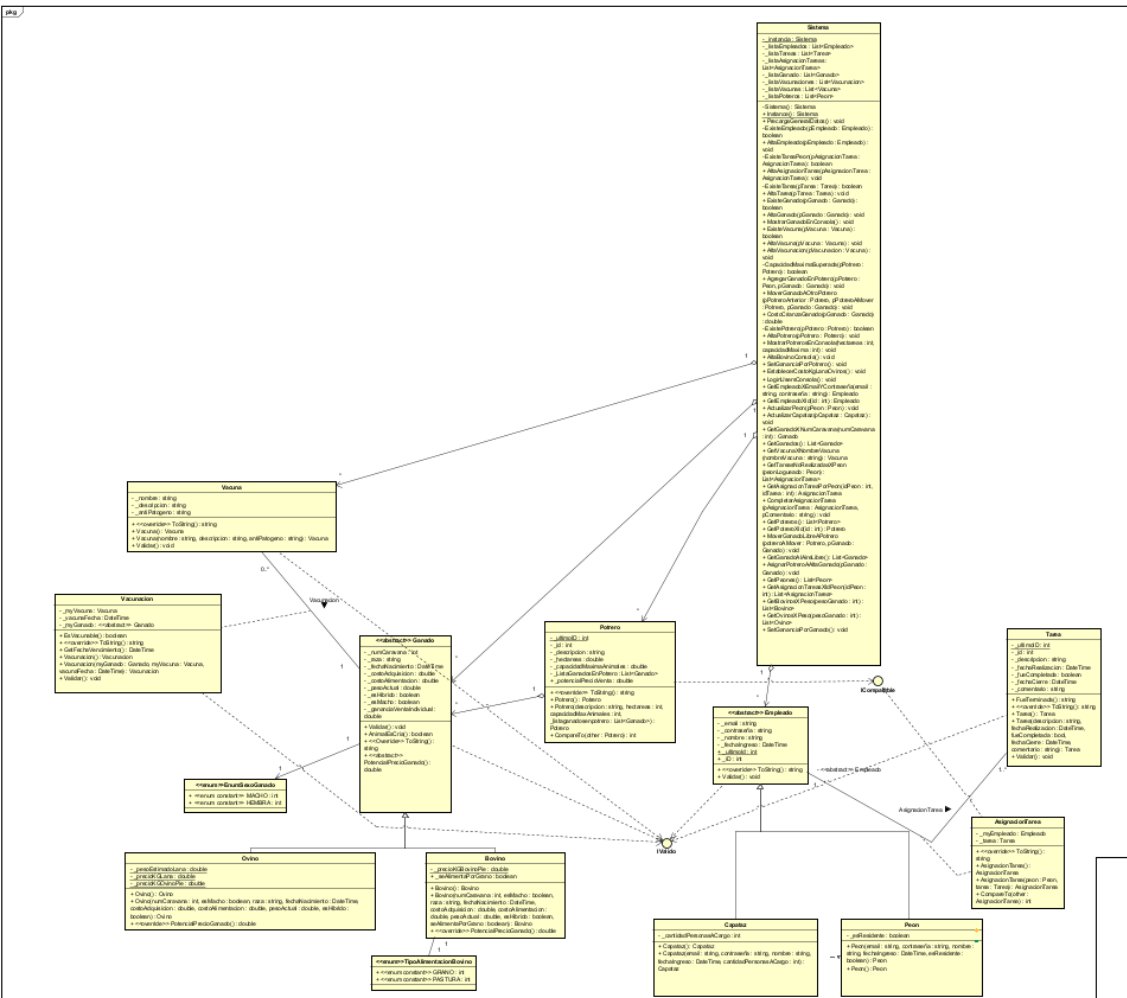
Analista en Tecnologías de la Información

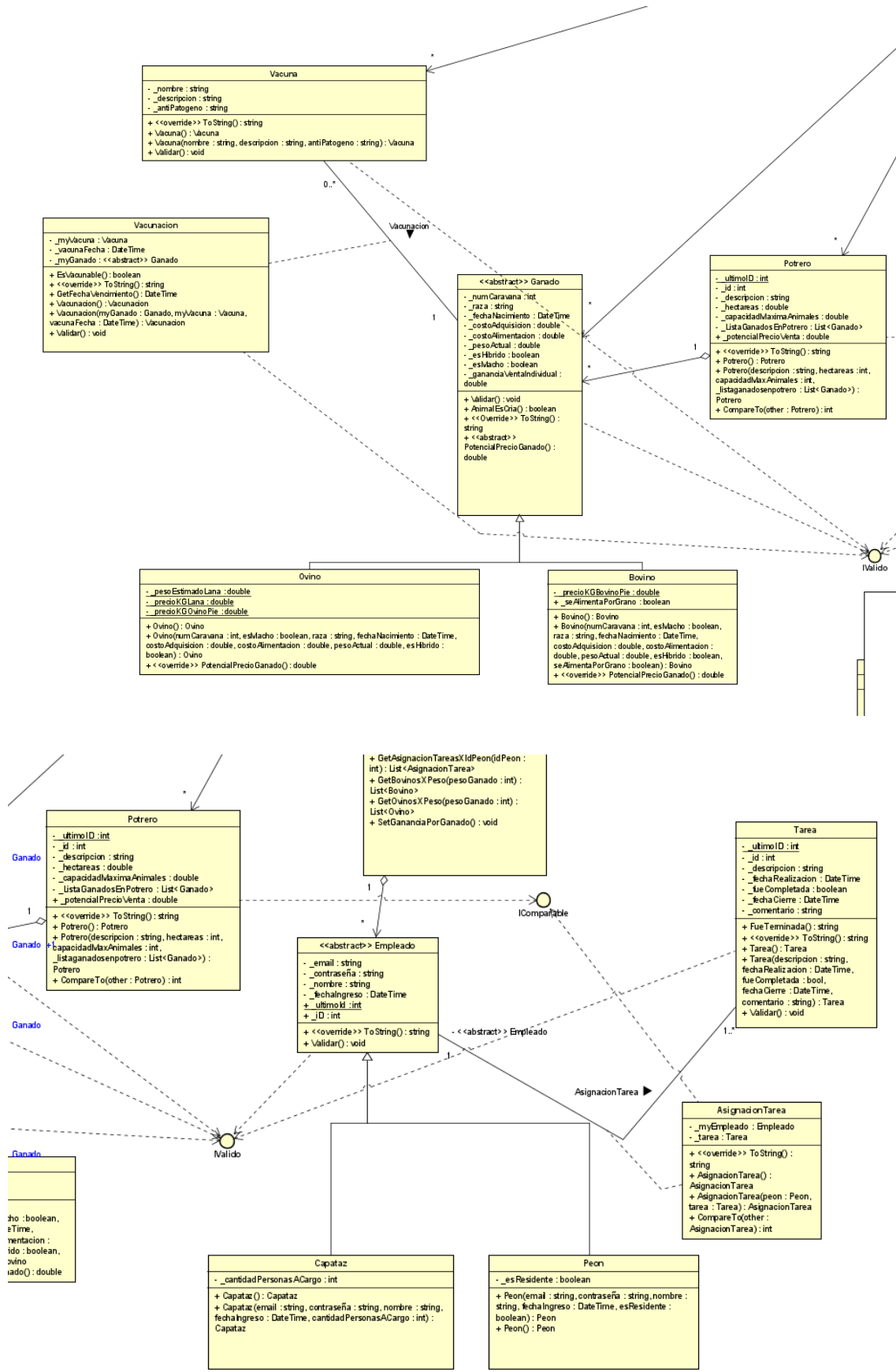
20/06/2024

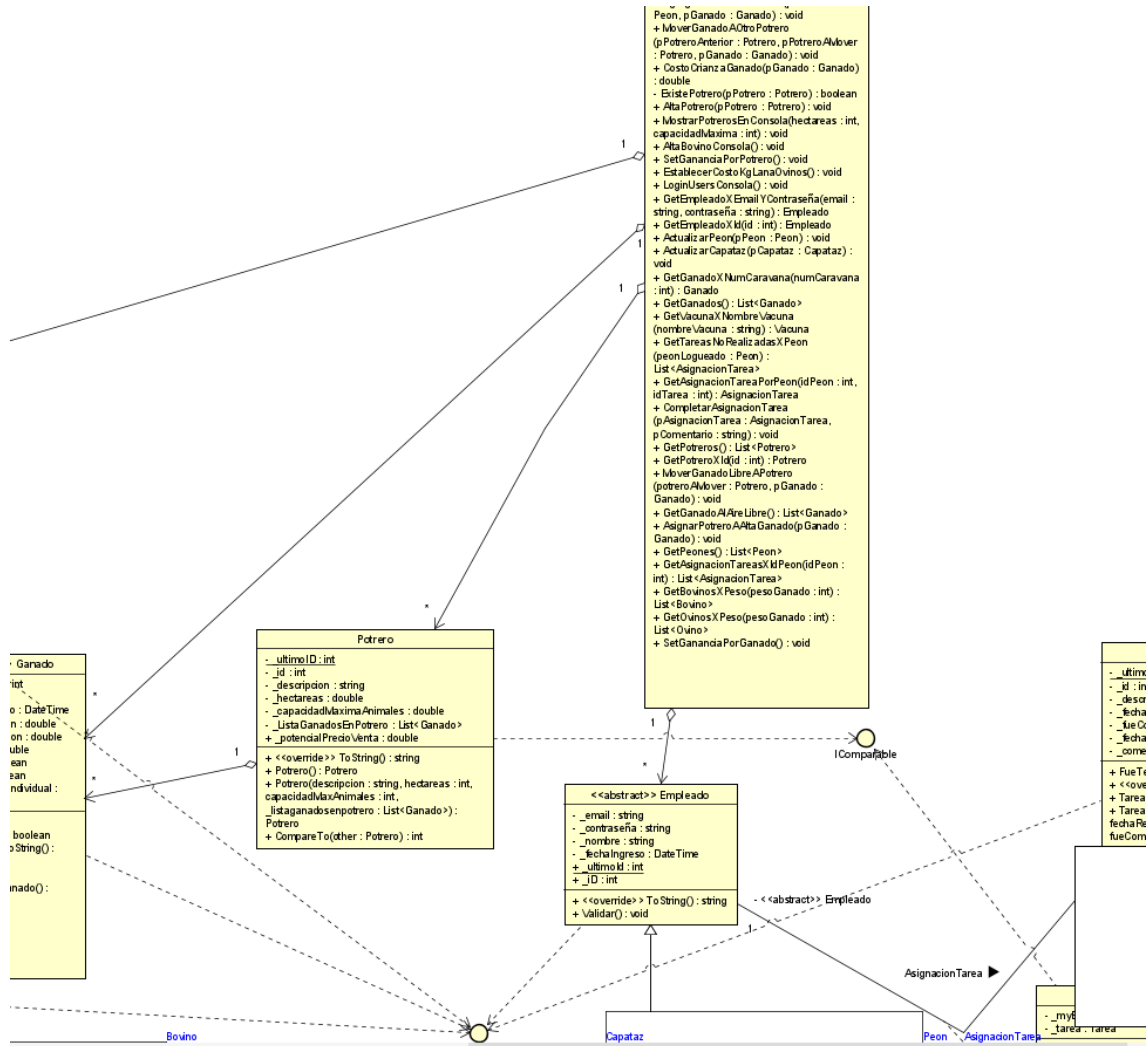
Contenido

Diagrama de clases definitivo del Dominio del problema.....	3
Diagrama de casos de uso que incluya todas las funcionalidades solicitadas	16
Una planilla con los casos de prueba necesarios para probar la totalidad de la aplicación y evidenciar el testing.....	17
Código fuente COMENTADO de toda la aplicación	25
.....	25

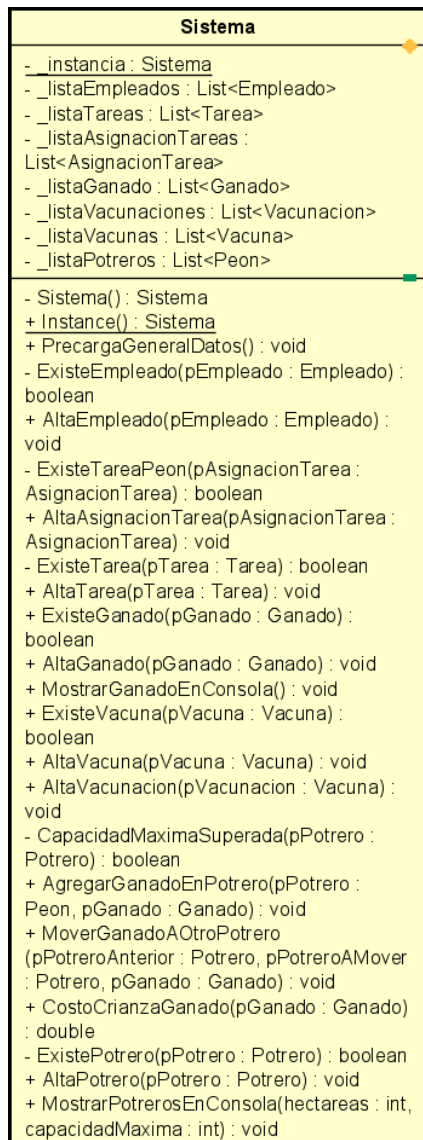
Diagrama de clases definitivo del Dominio del problema







1) Sistema



```

+ AltaBovinoConsola() : void
+ SetGananciaPorPotrero() : void
+ EstablecerCostoKgLanaOvinos() : void
+ LoginUsersConsola() : void
+ GetEmpleadoXEmailYContraseña(email :
string, contraseña : string) : Empleado
+ GetEmpleadoXId(id : int) : Empleado
+ ActualizarPeon(pPeon : Peon) : void
+ ActualizarCapataz(pCapataz : Capataz) :
void
+ GetGanadoXNumCaravana(numCaravana
: int) : Ganado
+ GetGanados() : List<Ganado>
+ GetVacunaXNombreVacuna
(nombreVacuna : string) : Vacuna
+ GetTareasNoRealizadasXPeon
(peonLogueado : Peon) :
List<AsignacionTarea>
+ GetAsignacionTareaPorPeon(idPeon : int,
idTarea : int) : AsignacionTarea
+ CompletarAsignacionTarea
(pAsignacionTarea : AsignacionTarea,
pComentario : string) : void
+ GetPotreros() : List<Potrero>
+ GetPotreroXId(id : int) : Potrero
+ MoverGanadoLibreAPotrero
(potreroAMover : Potrero, pGanado :
Ganado) : void
+ GetGanadoAlAireLibre() : List<Ganado>
+ AsignarPotreroAAItaGanado(pGanado :
Ganado) : void
+ GetPeones() : List<Peon>
+ GetAsignacionTareasXIdPeon(idPeon :
int) : List<AsignacionTarea>
+ GetBovinosXPeso(pesoGanado : int) :
List<Bovino>
+ GetOvinosXPeso(pesoGanado : int) :
List<Ovino>
+ SetGananciaPorGanado() : void

```

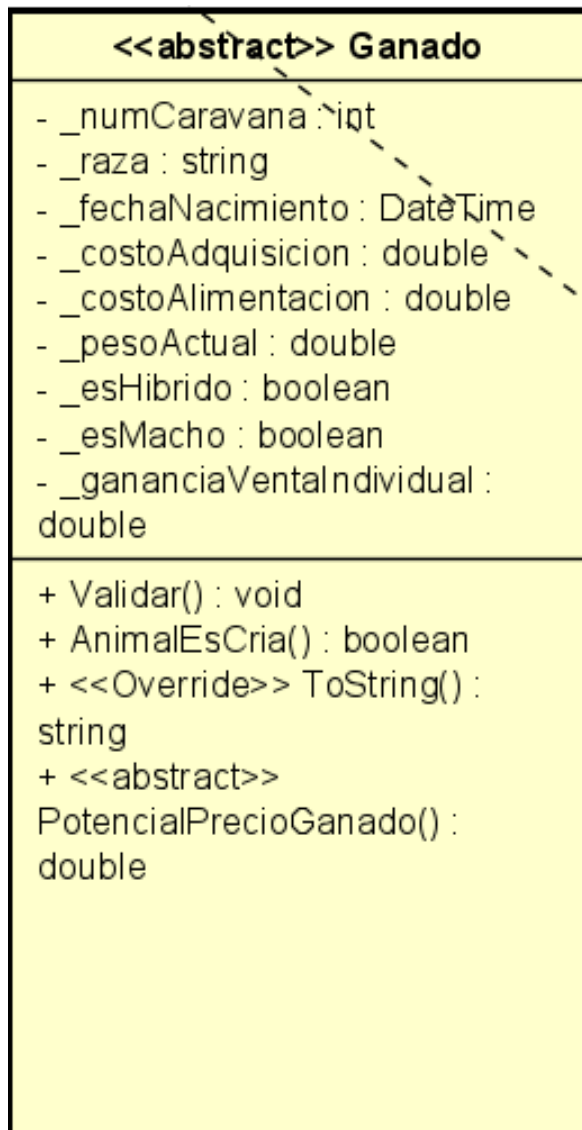
2)Vacuna

Vacuna
- _nombre : string - _descripcion : string - _antiPatogeno : string
+ <<override>> ToString() : string + Vacuna() : Vacuna + Vacuna(nombre : string, descripcion : string, antiPatogeno : string) : Vacuna + Validar() : void

3)Vacunación

Vacunacion
<ul style="list-style-type: none">- _myVacuna : Vacuna- _vacunaFecha : DateTime- _myGanado : <<abstract>> Ganado
<ul style="list-style-type: none">+ EsVacunable() : boolean+ <<override>> ToString() : string+ GetFechaVencimiento() : DateTime+ Vacunacion() : Vacunacion+ Vacunacion(myGanado : Ganado, myVacuna : Vacuna, vacunaFecha : DateTime) : Vacunacion+ Validar() : void

4)Ganado



5)Potrero

Potrero
<ul style="list-style-type: none">- <u>_ultimoID</u> : int- <u>_id</u> : int- <u>_descripcion</u> : string- <u>_hectareas</u> : double- <u>_capacidadMaximaAnimales</u> : double- <u>_ListaGanadosEnPotrero</u> : List<Ganado>+ <u>_potencialPrecioVenta</u> : double
<ul style="list-style-type: none">+ <<override>> ToString() : string+ Potrero() : Potrero+ Potrero(descripcion : string, hectareas : int, capacidadMaxAnimales : int, _listaganadosenpotrero : List<Ganado>) : Potrero+ CompareTo(other : Potrero) : int

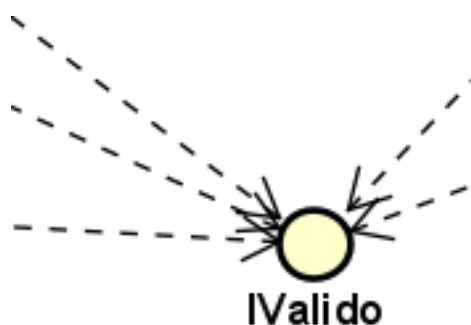
6)Ovino

Ovino
<ul style="list-style-type: none">- <u>_pesoEstimadoLana</u> : double- <u>_precioKGLana</u> : double- <u>_precioKGOvinoPie</u> : double
<ul style="list-style-type: none">+ Ovino() : Ovino+ Ovino(numCaravana : int, esMacho : boolean, raza : string, fechaNacimiento : DateTime, costoAdquisicion : double, costoAlimentacion : double, pesoActual : double, esHibrido : boolean) : Ovino+ <<override>> PotencialPrecioGanado() : double

7)Bovino

Bovino
<u>- __precioKGBovinoPie : double</u> + __seAlimentaPorGrano : boolean
+ Bovino() : Bovino + Bovino(numCaravana : int, esMacho : boolean, raza : string, fechaNacimiento : DateTime, costoAdquisicion : double, costoAlimentacion : double, pesoActual : double, esHibrido : boolean, seAlimentaPorGrano : boolean) : Bovino + <<override>> PotencialPrecioGanado() : double

8)InterfaceIValido

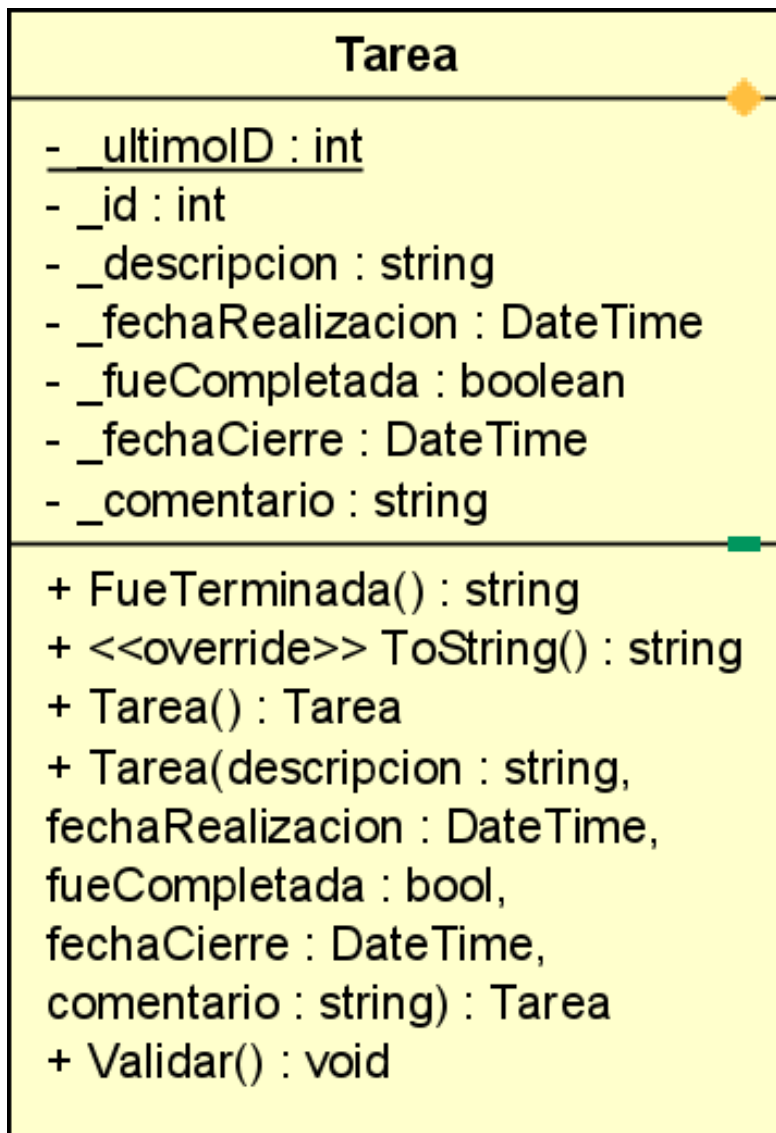


9)Empleado

<<abstract>> Empleado

- _email : string
 - _contraseña : string
 - _nombre : string
 - _fechaIngreso : DateTime
 - + _ultimold : int
 - + _iD : int
-
- + <<override>> ToString() : string
 - + Validar() : void

10)Tarea



11)AsignacionTarea

AsignacionTarea
- _myEmpleado : Empleado - _tarea : Tarea
+ <<override>> ToString() : string + AsignacionTarea() : AsignacionTarea + AsignacionTarea(peon : Peon, tarea : Tarea) : AsignacionTarea + CompareTo(other : AsignacionTarea) : int

12)Peon

Peon
- _esResidente : boolean
+ Peon(email : string, contraseña : string, nombre : string, fechaIngreso : DateTime, esResidente : boolean) : Peon + Peon() : Peon

13)Capataz

Capataz
- _cantidadPersonasACargo : int
+ Capataz() : Capataz + Capataz(email : string, contraseña : string, nombre : string, fechaIngreso : DateTime, cantidadPersonasACargo : int) : Capataz

14)IComparable

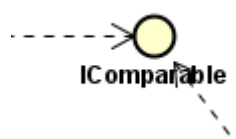
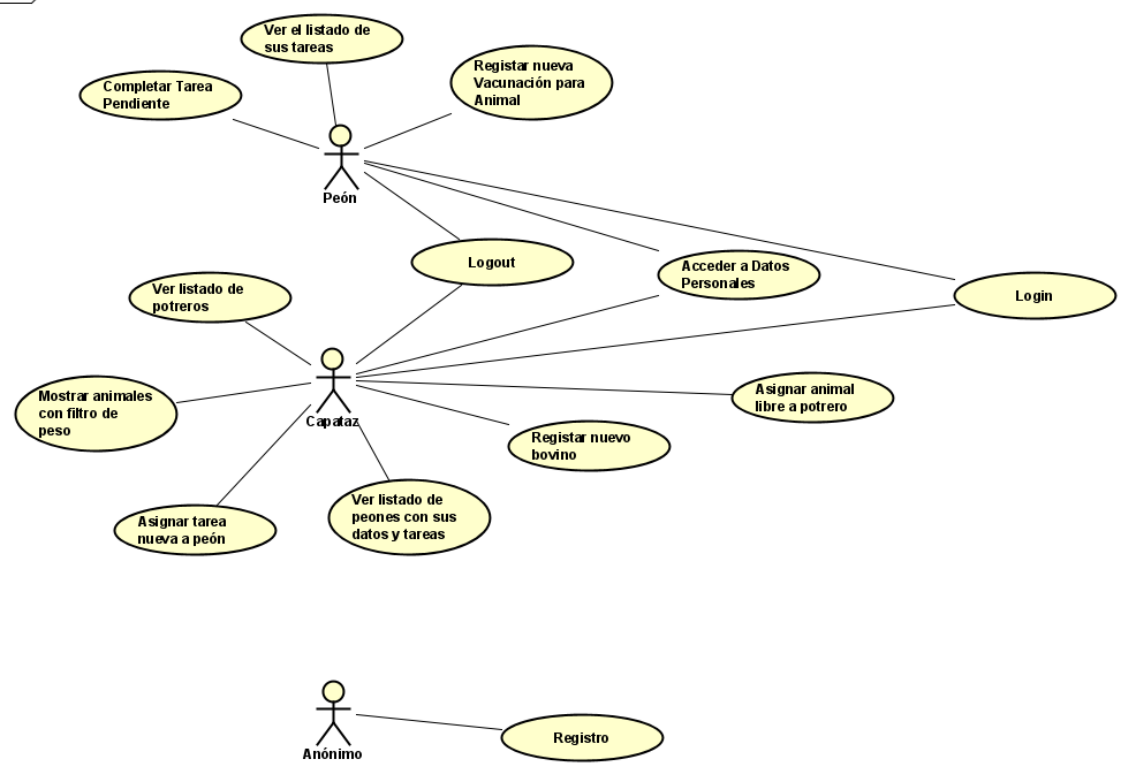


Diagrama de casos de uso que incluya todas las funcionalidades solicitadas

uc



Una planilla con los casos de prueba necesarios para probar la totalidad de la aplicación y evidenciar el testing

<u>Usuario</u>	<u>¿Qué se prueba?</u>	<u>Escenario de Prueba</u>	<u>Pasos</u>	<u>Fecha</u>	<u>Resultado Esperado</u>	<u>Resultado Obtenido</u>
Anónimo	Login. Ingresará email y contraseña y en caso de que sean válidos quedará autorizado a acceder a las funcionalidades correspondientes a su rol	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en Botón de Barra "Inicio de Sesión"	20/6/2024	True	True
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Anónimo	Registrarse como peón. Ingresar su email y una contraseña, el nombre, y si es residente o no. El email y contraseña serán utilizados para	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en Botón de Barra "Inicio de Sesión"	20/6/2024	True	True
Peón		Verificar Funcionalidad de Prueba, ingresando	Ingresar URL Correspondiente		No cumple	Index, Inicio

	verificar su identidad para ingresar con el rol “Peón”	o por botón o URL				
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Anónimo	Acceder a sus datos personales	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clicar en Menú Principal Usuario "Editar"		True	True
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clicar en Menú Principal Usuario "Editar"		True	True
Anónimo	Registrar nueva vacunación para	Verificar Funcionalidad de Prueba, ingresando por	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio

	cualquier animal	botón o URL				
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en Barra Usuario "Vacunacion"		True	True
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Anónimo	Ver el listado de todas sus tareas aún no completadas ordenadas por fecha pactada ascendente. Se deben mostrar todos los datos de la tarea	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en Barra Usuario "Tareas"		True	True
Capataz		Verificar Funcionalidad de Prueba, ingresando por	Ingresar URL Correspondiente		No cumple	Index, Inicio

		botón o URL				
Anónimo	Completar una tarea pendiente. Cuando se completa una tarea se debe agregar un comentario de cierre	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en botón "Completar Tarea Pendiente"		True	True
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Anónimo	Registrar un nuevo bovino en el sistema	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por	Ingresar URL Correspondiente		No cumple	Index, Inicio

		botón o URL				
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Registrar Bovino"		True	True
Anónimo	Asignar un animal libre a un potrero	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Animal Libre a Potrero"		True	True
Anónimo	Ver listado de todos los peones. Se deben mostrar todos sus	Verificar Funcionalidad de Prueba, ingresando por	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio

	datos. Se podrá acceder a un detalle de todas las	botón o URL				
Peón	tareas que le han sido asignadas (completas o no)	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Lista Peones", una vez en la lista Clickear en lista botón "Ver" Para ver Tarea de cada Peón Correspondientemente		True	True
Anónimo	Asignar una nueva tarea a un peón	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio

Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en lista botón "Agregar" Para asignar Tarea a cada Peón Correspondientemente		True	True
Anónimo	Ver listado de todos los potreros ordenados por capacidad ascendente y cantidad de animales descendente . Se debe mostrar el Identificador, la descripción, la cantidad de animales	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Capataz	que contiene y el potencial precio de venta	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Registrar Bovino"		True	True
Anónimo	Dado un peso y un tipo mostrar todos los animales de ese tipo que superen el peso dado.	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio

Peón	De cada animal se debe mostrar el identificador de la caravana, el tipo de animal, el peso, sexo y la potencial ganancia de venta individual	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente		No cumple	Index, Inicio
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Mostrar Animal"		True	True
Anónimo	Logout	Verificar Funcionalidad de Prueba, ingresando por botón o URL	Ingresar URL Correspondiente	20/6/2024	No cumple	Index, Inicio
Peón		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Salir"		True	True
Capataz		Verificar Funcionalidad de Prueba, ingresando por botón o URL	Clickear en barra botón "Salir"		True	True

[Código fuente COMENTADO de toda la aplicación](#)

AsignacionTarea

```
namespace Obligatorio.Dominio
{
    public class AsignacionTarea : IComparable<AsignacionTarea>
    {
        public Empleado MyEmpleado { get; set; }
        public Tarea Tarea { get; set; }
        public AsignacionTarea()
        {
            MyEmpleado = null;
            Tarea = null;
        }
        public AsignacionTarea(Empleado myEmpleado, Tarea tarea)
        {
            MyEmpleado = myEmpleado;
            Tarea = tarea;
        }
        public override string ToString()
        {
            return $"La tarea {Tarea.Descripcion} de ID: {Tarea.ID}. Fue
Asignada a: {MyEmpleado.Nombre} de Email: {MyEmpleado.Email}";
        }
        public int CompareTo(AsignacionTarea? other)
        {
            return
Tarea.FechaRealizacion.CompareTo(other.Tarea.FechaRealizacion);
        }
    }
}
```

Bovino

```
namespace Obligatorio.Dominio
{
    public class Bovino : Ganado
    {
        public bool SeAlimentaPorGrano { get; set; }
        public static double PrecioKGBovinoPie { get; set; } = 160;
        public Bovino()
        {
            NumCaravana = 00000000;
            EsMacho = false;
            Raza = "";
            FechaNacimiento = DateTime.Today;
            CostoAdquisicion = 0;
            CostoAlimentacion = 0;
            PesoActual = 0;
            EsHibrido = false;
            SeAlimentaPorGrano = false;
            GananciaVentaIndividual = 0;
        }
        public Bovino(int numCaravana, bool esMacho, string raza, DateTime
fechaNacimiento, double costoAdquisicion,
            double costoAlimentacion, double pesoActual, bool esHibrido,
bool seAlimentaPorGrano)
```

```

    {
        NumCaravana = numCaravana;
        EsMacho = esMacho;
        Raza = raza;
        FechaNacimiento = fechaNacimiento;
        CostoAdquisicion = costoAdquisicion;
        CostoAlimentacion = costoAlimentacion;
        PesoActual = pesoActual;
        EsHibrido = esHibrido;
        SeAlimentaPorGrano = seAlimentaPorGrano;
        GananciaVentaIndividual = 0;
    }
    public override double PotencialPrecioGanado()
    {
        double aux = 0;
        aux = PesoActual * PrecioKGBovinoPie;
        if (SeAlimentaPorGrano == true)
        {
            aux = aux * 1.30;
        }
        if (EsMacho == false)
        {
            aux = aux * 1.10;
        }
        return aux;
    }
}
}
}

```

Capataz

```

namespace Obligatorio.Dominio
{
    public class Capataz : Empleado
    {
        public int CantidadPersonasACargo { get; set; }
        public Capataz()
        {
            Id = UltimoId++;
            Email = "";
            Contraseña = "";
            Nombre = "";
            FechaIngreso = DateTime.Today;
            CantidadPersonasACargo = 0;
        }
        public Capataz(string email, string contraseña, string nombre,
            DateTime fechaIngreso, int cantidadPersonasACargo)
        {
            Id = UltimoId++;
            Email = email;
            Contraseña = contraseña;
            Nombre = nombre;
            FechaIngreso = fechaIngreso;
            CantidadPersonasACargo = cantidadPersonasACargo;
        }
    }
}
}

```

Empleado

```
namespace Obligatorio.Dominio
{
    public abstract class Empleado : IValido
    {
        public static int UltimoId { get; set; } = 1;
        public int Id { get; set; }
        public string Email { get; set; }
        public string Contraseña { get; set; }
        public string Nombre { get; set; }
        public DateTime FechaIngreso { get; set; }
        public void Validar()
        {
            char aux;
            int valorAux = 0;
            string nuevoTexto;
            bool arroba = false;
            if (Contraseña == null || Email == null || Nombre == null)
            {
                throw new Exception("Todos los campos son obligatorios.");
            }
            if (FechaIngreso == DateTime.Today)
            {
                throw new Exception("Ingrese correctamente su fecha de
ingreso.");
            }
            if (Contraseña.Length < 8)
            {
                throw new Exception("La Contraseña debe ser mayor a 8
caracteres.");
            }
            for (int i = 0; i < Email.Length; i++)
            {
                aux = Email[i];
                if (aux == '@')
                {
                    valorAux = i;
                    arroba = true;
                }
            }
            if (arroba == false)
            {
                throw new Exception("El email debe contener un dominio.");
            }
            nuevoTexto = Email.Substring(0, valorAux);
            if (nuevoTexto.Length < 8)
            {
                throw new Exception("El largo del Email debe ser de al menos
8 caracteres.");
            }
        }
        public override string ToString()
        {
            return $"{Nombre}";
        }
    }
}
```

Ganado

```
namespace Obligatorio.Dominio
{
    public abstract class Ganado : IValido
    {
        public int NumCaravana { get; set; }
        public string Raza { get; set; }
        public DateTime FechaNacimiento { get; set; }
        public double CostoAdquisicion { get; set; }
        public double CostoAlimentacion { get; set; }
        public double PesoActual { get; set; }
        public bool EsHibrido { get; set; }
        public bool EsMacho { get; set; }
        public double GananciaVentaIndividual { get; set; } = 0;
        public Ganado() { }
        public void Validar()
        {
            if (NumCaravana == null || Raza == null || FechaNacimiento ==
DateTime.Today || CostoAdquisicion == null || CostoAlimentacion == null ||
PesoActual == null)
            {
                throw new Exception("Todos los campos son OBLIGATORIOS");
            }
            if (NumCaravana.ToString().Length != 8)
            {
                throw new Exception("Los dígitos del número de Caravana
deben ser 8");
            }
            if (CostoAdquisicion == 0 || CostoAlimentacion == 0 ||
PesoActual == 0)
            {
                throw new Exception("Ingrese correctamente el valor
solicitado.");
            }
        }
        public bool AnimalEsCria()
        {
            bool bandera = false;
            if (DateTime.Today.Year - FechaNacimiento.Year < 1)
            {
                bandera = true;
            }
            return bandera;
        }
        public abstract double PotencialPrecioGanado();

        public override string ToString()
        {
            return $"Sexo: {EsMacho}. Nació en el: {FechaNacimiento.Year} en
el mes {FechaNacimiento.Month}. De raza: {Raza}." +
                $"Con un peso de: {PesoActual} Kgs. Su número de caravana
es: {NumCaravana}";
        }
    }
}
```

Interface IValido

```
namespace Obligatorio.Dominio
{
    internal interface IValido
    {
        public void Validar();
    }
}
```

Ovino

```
namespace Obligatorio.Dominio
{
    public class Ovino : Ganado
    {
        public static double PesoEstimadoLana { get; set; } = 160;
        public static double PrecioKGLana { get; set; } = 110;
        public static double PrecioKGOvinoPie { get; set; } = 150;
        public Ovino()
        {
            NumCaravana = 00000000;
            EsMacho = false;
            Raza = "";
            FechaNacimiento = new DateTime(0000, 00, 00);
            CostoAdquisicion = 0;
            CostoAlimentacion = 0;
            PesoActual = 0;
            EsHibrido = false;
            GananciaVentaIndividual = 0;
        }
        public Ovino(int numCaravana, bool esMacho, string raza, DateTime
fechaNacimiento,
            double costoAdquisicion, double costoAlimentacion, double
pesoActual,
            bool esHibrido)
        {
            NumCaravana = numCaravana;
            EsMacho = esMacho;
            Raza = raza;
            FechaNacimiento = fechaNacimiento;
            CostoAdquisicion = costoAdquisicion;
            CostoAlimentacion = costoAlimentacion;
            PesoActual = pesoActual;
            EsHibrido = esHibrido;
            GananciaVentaIndividual = 0;
        }
        public override double PotencialPrecioGanado()
        {
            double aux = 0;
            aux = PesoEstimadoLana * PrecioKGLana + PesoActual *
PrecioKGOvinoPie;
            if (EsHibrido)
            {
                aux = aux * 0.95;
            }

            return aux;
        }
    }
}
```

```

    }
}

```

Peon

```

namespace Obligatorio.Dominio
{
    public class Peon : Empleado
    {
        public bool EsResidente { get; set; }
        public Peon()
        {
            Id = UltimoId++;
            Email = "";
            Contraseña = "";
            Nombre = "";
            FechaIngreso = DateTime.Today;
            EsResidente = false;
        }
        public Peon(string email, string contraseña, string nombre, DateTime
fechaIngreso, bool esResidente)
        {
            Id = UltimoId++;
            Email = email;
            Contraseña = contraseña;
            Nombre = nombre;
            FechaIngreso = fechaIngreso;
            EsResidente = esResidente;
        }
    }
}

```

Potrero

```

namespace Obligatorio.Dominio
{
    public class Potrero : IComparable<Potrero>
    {
        public static int UltimoID { get; set; } = 1;
        public int ID { get; set; }
        public string Descripcion { get; set; }
        public int Hectareas { get; set; }
        public int CapacidadMaxAnimales { get; set; }
        public List<Ganado> _ListaGanadosEnPotrero { get; set; }
        public double PotencialPrecioVenta { get; set; } = 0;
        public Potrero()
        {
            ID = UltimoID++;
            Descripcion = "";
            Hectareas = 1;
            CapacidadMaxAnimales = 0;
            PotencialPrecioVenta = 0;
        }
        public Potrero(string descripcion, int hectareas, int
capacidadMaxAnimales, List<Ganado> _listaganadosenpotrero)
        {
            ID = UltimoID;

```

```

        UltimoID++;
        Descripcion = descripcion;
        Hectareas = hectareas;
        CapacidadMaxAnimales = capacidadMaxAnimales;
        _ListaGanadosEnPotrero = _listaganadosenpotrero;
        PotencialPrecioVenta = 0;
    }
    public override string ToString()
    {
        return $"Este potrero cuenta con {Hectareas} de hectareas. Con
una capacidad máxima de {CapacidadMaxAnimales} de animales." +
        $"Es el potero número: {ID}";
    }
    public int CompareTo(Potrero? potrero)
    {
        int retorno;
        retorno =
CapacidadMaxAnimales.CompareTo(potrero.CapacidadMaxAnimales);
        if (retorno == 0)
        {
            retorno =
_ListaGanadosEnPotrero.Count.CompareTo(potrero._ListaGanadosEnPotrero.Count)
* -1;
        }
        return retorno;
    }
}

```

Sistema

```

namespace Obligatorio.Dominio

{

    public class Sistema

    {

        private Sistema()

        {

            PrecargaGeneralDatos();

        }

        private static Sistema _instancia;

        public static Sistema Instance()

        {

```

```
        if (_instancia == null)
        {
            _instancia = new Sistema();
        }

        return _instancia;
    }
}
```

```
private void PrecargaGeneralDatos()
```

```
{
    PrecargarCapataz();

    PrecargarPeon();

    PrecargarTarea();

    PrecargarAsignacionTarea();

    PrecargarBovino();

    PrecargarOvino();

    PrecargarVacuna();

    PrecargarVacunacion();

    PrecargarPotrero();

    PrecargaAgregarGanadoEnPotrero();
}
```

```
//EMPLEADOS
```

```
private List<Empleado> _listaEmpleados = new List<Empleado>();
```

```
private bool ExisteEmpleado(Empleado pEmpleado)
```

```
{
    bool existe = false;
```



```

        foreach (Empleado iEmpleado in _listaEmpleados)
        {
            if (iEmpleado.Email.ToLower() == pEmpleado.Email.ToLower())
            {
                existe = true; break;
            }
        }

        return existe;
    }

    public void AltaEmpleado(Empleado pEmpleado)
    {
        pEmpleado.Validar();

        if (!ExisteEmpleado(pEmpleado))
        {
            _listaEmpleados.Add(pEmpleado);
        }

        else
        {
            throw new Exception("Este Empleado ya existe.");
        }
    }

}

//TAREA - ASIGNACION TAREA

private List<Tarea> _listaTareas = new List<Tarea>();

```

```

        private List<AsignacionTarea> _listaAsignacionTareas = new
List<AsignacionTarea>();

        private bool ExisteTareaEnPeon(AsignacionTarea pAsignacionTarea)
        {
            bool existe = false;

            for (int i = 0; i < _listaAsignacionTareas.Count; i++)
            {
                AsignacionTarea auxAsignacionTarea =
                _listaAsignacionTareas[i];

                if (auxAsignacionTarea.MyEmpleado ==
                pAsignacionTarea.MyEmpleado && auxAsignacionTarea.Tarea ==
                pAsignacionTarea.Tarea)
                {
                    existe = true;
                }
            }

            return existe;
        }

        public void AltaAsignacionTarea(AsignacionTarea pAsignacionTarea)
        {
            if (!ExisteTareaEnPeon(pAsignacionTarea))
            {
                _listaAsignacionTareas.Add(pAsignacionTarea);

                _listaAsignacionTareas.Sort();
            }

            else

```

```

        {

            throw new Exception("Este Peon ya cuenta con esta tarea.");

        }

    }

    private bool ExisteTarea(Tarea pTarea)

    {

        bool existe = false;

        for (int i = 0; i < _listaTareas.Count; i++)

        {

            Tarea auxTarea = _listaTareas[i];

            if (pTarea.Descripcion == auxTarea.Descripcion &&
pTarea.FechaRealizacion == auxTarea.FechaRealizacion)

            {

                existe = true;

            }

        }

        return existe;

    }

    public void AltaTarea(Tarea pTarea)

    {

        pTarea.Validar();

        if (!ExisteTarea(pTarea))

        {

            _listaTareas.Add(pTarea);

        }

    }

}

```

```

else

{

    throw new Exception("La tarea que desea agregar ya existe.");

}

}

//GANADO

private List<Ganado> _listaGanado = new List<Ganado>();

public bool ExisteGanado(Ganado pGanado)

{

    bool boolean = false;

    for (int i = 0; i < _listaGanado.Count; i++)

    {

        Ganado auxGanado = _listaGanado[i];

        if (auxGanado.NumCaravana == pGanado.NumCaravana)

        {

            boolean = true;

        }

        else

        {

            boolean = false;

        }

    }

    return boolean;

}

```

```

public void AltaGanado(Ganado pGanado)
{
    pGanado.Validar();

    if (!ExisteGanado(pGanado))
    {
        _listaGanado.Add(pGanado);
    }
    else
    {
        throw new Exception("Este ganado ya existe en la lista.");
    }
}

public void MostrarGanadoEnConsola()
{
    int aux = 0;

    for (int i = 0; i < _listaGanado.Count; i++)
    {
        Ganado auxGanado = _listaGanado[i];

        Console.WriteLine($"{auxGanado.NumCaravana} {auxGanado.Raza}
{auxGanado.PesoActual}" +

            $"{auxGanado.EsMacho}");

        aux++;
    }
}

```

```
//VACUNA

private List<Vacunacion> _listaVacunaciones = new List<Vacunacion>();

private List<Vacuna> _listaVacunas = new List<Vacuna>();

public bool ExisteVacuna(Vacuna pVacuna)
{
    bool existe = false;

    foreach (Vacuna iVacuna in _listaVacunas)
    {
        if (pVacuna.Nombre == iVacuna.Nombre)
        {
            existe = true;
        }
    }

    return existe;
}

public void AltaVacuna(Vacuna pVacuna)
{
    pVacuna.Validar();

    if (!ExisteVacuna(pVacuna))
    {
        _listaVacunas.Add(pVacuna);
    }

    else
    {
        throw new Exception("Esta vacuna ya existe!.");
    }
}
```

```

        }

    }

    public void AltaVacunacion(Vacunacion pVacunacion)
    {
        pVacunacion.Validar();

        if (pVacunacion.EsVacunable())
        {
            _listaVacunaciones.Add(pVacunacion);
        }

        else
        {
            throw new Exception("Este animal aún no puede ser vacunado");
        }
    }

//POTRERO

private bool CapacidadMaximaSuperada(Potrero pPotrero)
{
    bool retorno = false;

    if (pPotrero.CapacidadMaxAnimales <=
pPotrero._ListaGanadosEnPotrero.Count)
    {
        retorno = true;
    }

    return retorno;
}

```

```

    }

    public void AgregarGanadoEnPotrero(Potrero pPotrero, Ganado pGanado)
    {
        if (!CapacidadMaximaSuperada(pPotrero))
        {
            pPotrero._ListaGanadosEnPotrero.Add(pGanado);
        }
        else
        {
            throw new Exception("No hay más espacio para agregar animales
en este Potrero.");
        }
    }

    public void MoverGanadoA OtroPotrero(Potrero pPotreroAnterior, Potrero
pPotreroAMover, Ganado pGanado)
    {
        if (!pGanado.AnimalEsCria())
        {
            for (int i = 0; i <
pPotreroAnterior._ListaGanadosEnPotrero.Count; i++)
            {
                Ganado auxGanado =
pPotreroAnterior._ListaGanadosEnPotrero[i];
                if (auxGanado.NumCaravana == pGanado.NumCaravana)
                {
                    pPotreroAnterior._ListaGanadosEnPotrero.RemoveAt(i);

```



```

        break;
    }

}

AgregarGanadoEnPotrero(pPotreroAMover, pGanado);

}

else

{

    throw new Exception("Este animal aún es una cria. No puede
Moverse de Potrero.");

}

}

//CALCULOS

public double CostoCrianzaGanado(Ganado pGanado)

{

    double aux = 0;

    for (int i = 0; i < _listaVacunaciones.Count; i++)

    {

        if      (_listaVacunaciones[i].MyGanado.NumCaravana    ==
pGanado.NumCaravana)

        {

            aux = aux + 200;

        }

    }

    return      aux      +      pGanado.CostoAdquisicion      +
pGanado.CostoAlimentacion;

```

```
}
```

```
//MUESTRA CONSOLA PUNTO 3
```

```
private List<Potrero> _listaPotreros = new List<Potrero>();
```

```
private bool ExistePotrero(Potrero pPotrero)
```

```
{
```

```
    bool existe = false;
```

```
    for (int i = 0; i < _listaPotreros.Count; i++)
```

```
    {
```

```
        if (pPotrero.ID == _listaPotreros[i].ID)
```

```
        {
```

```
            existe = true; break;
```

```
        }
```

```
    }
```

```
    return existe;
```

```
}
```

```
public void AltaPotrero(Potrero pPotrero)
```

```
{
```

```
    if (!ExistePotrero(pPotrero))
```

```
    {
```

```
        _listaPotreros.Add(pPotrero);
```

```
    }
```

```
    else
```

```

        {

            throw new Exception("Este potrero ya existe!");

        }

    }

    public void MostrarPotrerosEnConsola(int hectareas, int
capacidadMaxima)

    {

        for (int i = 0; i < _listaPotreros.Count; i++)

        {

            Potrero aux = _listaPotreros[i];

            if (aux.Hectareas > hectareas && aux.CapacidadMaxAnimales >
capacidadMaxima)

            {

                Console.WriteLine(aux.ID + " " + aux.Descripcion + " " +
aux.Hectareas + " " + aux.CapacidadMaxAnimales);

            }

        }

    }

//ALTA BOVINO CONSOLA

public void AltaBovinoConsola()

{

    //CODIGO IDENTIFICADOR

    Console.WriteLine("Desea agregar un bovino? \n" +

        "s/S para SI\n" +

        "cualquier letra para NO");

```

```

string texto = Console.ReadLine();

while (texto == "s" || texto == "S" || texto == "1")
{
    bool boolCaravana;

    int codigoCaravana;

    Console.WriteLine("Ingrese Código identificador NÚMÉRICO de
Caravana de Bovino de 8 digitos.");

    try
    {
        boolCaravana = int.TryParse(Console.ReadLine(), out
codigoCaravana);
    }

    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }

    while (!boolCaravana || codigoCaravana.ToString().Length !=
8)
    {
        Console.WriteLine("Ingrese Código identificador NÚMÉRICO
de Caravana de Bovino de 8 digitos.");

        try
        {
            boolCaravana = int.TryParse(Console.ReadLine(), out
codigoCaravana);
        }
    }
}

```

```

        catch (Exception ex)
        {
            throw new Exception(ex.Message);
        }
    }

    bool segundaBandera = true;

    for (int i = 0; i < _listaGanado.Count; i++)
    {
        if (codigoCaravana == _listaGanado[i].NumCaravana)
        {
            Console.WriteLine("Este código ya existe. Ingrese uno
nuevo por favor.");

            segundaBandera = false;

            break;
        }
    }

    while (!segundaBandera)
    {
        codigoCaravana = int.Parse(Console.ReadLine());

        for (int i = 0; i < _listaGanado.Count; i++)
        {
            if (codigoCaravana == _listaGanado[i].NumCaravana)
            {
                Console.WriteLine("Este código ya existe. Ingrese
uno nuevo por favor.");
            }
        }
    }
}

```

```

        segundaBandera = false;

        break;

    }

    else

    {

        segundaBandera = true;

    }

}

}

if (segundaBandera)

{

    {

        Console.WriteLine("Usted acaba de compeltar el primer
paso del registro de su bovino \n" +

            " FELICIDADES.");

    }

}

//

//SEXO BOVINO

bool macho_hembraBool;

int macho_hembraInt;

bool macho_hembraValor = false;

Console.WriteLine("Su bovino es Macho o Hembra? \n" +

    "Coloque el número 1 si es: HEMBRA \n " +

    "Coloque el número 2 si es: MACHO");

```

```

        try

        {

            macho_hembraBool = int.TryParse(Console.ReadLine(), out
macho_hembraInt);

        }

        catch (Exception ex)

        {

            throw new Exception(ex.Message);

        }

        while (!macho_hembraBool || (macho_hembraInt != 1 &&
macho_hembraInt != 2))

        {

            Console.WriteLine("Su bovino es Macho o Hembra? \n" +

            "Coloque el número 1 si es: HEMBRA \n " +

            "Coloque el número 2 si es: MACHO");

            try

            {

                macho_hembraBool = int.TryParse(Console.ReadLine(),
out macho_hembraInt);

            }

            catch (Exception ex)

            {

                throw new Exception(ex.Message);

            }

        }

        if (macho_hembraInt == 1)

```

```

    {
        macho_hembraValor = false;
    }

    else if (macho_hembraInt == 2)
    {
        macho_hembraValor = true;
    }

    //

    //RAZA BOVINO

    string stringRaza = "";

    Console.WriteLine("Que raza es su bovino?\n" +

        " 1 - ANGUS \n" +

        " 2 - HEREFORD \n" +

        " 3 - SIMMENTAL \n" +

        " 4 - LIMOUSIN \n" +

        " 5 - CHARLOAIS \n" +

        " 6 - BRAHMAN \n" +

        " 7 - GELBVIEH \n");

    bool booleanRaza = int.TryParse(Console.ReadLine(), out int
intRaza);

    while (!booleanRaza || (intRaza != 1 && intRaza != 2 && intRaza
!= 3

        && intRaza != 4 && intRaza != 5 && intRaza != 6 && intRaza
!= 7))

    {

        Console.WriteLine("Que raza es su bovino?\n" +

```



```

        " 1 - ANGUS \n" +

        " 2 - HEREFORD \n" +

        " 3 - SIMMENTAL \n" +

        " 4 - LIMOUSIN \n" +

        " 5 - CHARLOAIS \n" +

        " 6 - BRAHMAN \n" +

        " 7 - GELBVIEH \n");

        booleanRaza = int.TryParse(Console.ReadLine(), out
intRaza);

    }

    switch (intRaza)
    {

        case 1:

            stringRaza = "ANGUS";

            break;

        case 2:

            stringRaza = "HEREFORD";

            break;

        case 3:

            stringRaza = "SIMMENTAL";

            break;

        case 4:

            stringRaza = "LIMOUSIN";

            break;

        case 5:

```

```

        stringRaza = "CHARLOAIS";

        break;

    case 6:

        stringRaza = "BRAHMAN";

        break;

    case 7:

        stringRaza = "GELBVIEH";

        break;

    }

    //

    //FECHA DE NACIMIENTO BOVINO

    DateTime dateFecha;

    Console.WriteLine("En que fecha nació su bovino?\n Seguir el
siguiente formato:" +

        "(YYYY,MM,DD)");

    bool booleanFecha = DateTime.TryParse(Console.ReadLine(), out
dateFecha);

    while (!booleanFecha || (dateFecha > DateTime.Today))

    {

        Console.WriteLine("En que fecha nació su bovino?\n Seguir
el siguiente formato:" +

            "(YYYY,MM,DD)");

        booleanFecha = DateTime.TryParse(Console.ReadLine(), out
dateFecha);

    }

    //

```

```

//COSTO BOVINO

Console.WriteLine("Cuánto costó adquirir su bovino?");

bool                booleanCostoAdquisicion                =
double.TryParse(Console.ReadLine(), out double doubleCostoAdquisicion);

while (!booleanCostoAdquisicion || (doubleCostoAdquisicion ==
0))

{

    Console.WriteLine("Cuánto costó adquirir su bovino?");

    booleanCostoAdquisicion                =
double.TryParse(Console.ReadLine(), out doubleCostoAdquisicion);

}

//

//COSTO ALIMENTACIÓN BOVINO

Console.WriteLine("Cuánto cuesta alimentar a su bovino?");

bool                booleanCostoAlimentacin                =
double.TryParse(Console.ReadLine(), out double doubleCostoAlimentacion);

while (!booleanCostoAdquisicion || (doubleCostoAlimentacion
== 0))

{

    Console.WriteLine("Cuánto    cuesta    alimentar    a    su
bovino?");

    booleanCostoAlimentacin                =
double.TryParse(Console.ReadLine(), out doubleCostoAlimentacion);

}

//

//PESO BOVINO

Console.WriteLine("Cuánto pesa su bovino? (en KG)");

```

```

        bool booleanPesoBovino = double.TryParse(Console.ReadLine(),
out double doublePesoBovino);

        while (!booleanPesoBovino || (doublePesoBovino == 0))
        {

            Console.WriteLine("Cuánto pesa su bovino? (en KG)");

            booleanPesoBovino = double.TryParse(Console.ReadLine(),
out doublePesoBovino);

        }

        //

        //BOVINO HIBRIDO

        bool booleanHibrido = false;

        Console.WriteLine("Su bovino es hibrido? \n s/S para SI \n
n/N para NO");

        string stringHibrido = Console.ReadLine().ToLower();

        while (stringHibrido != "s" && stringHibrido != "n")
        {

            Console.WriteLine("Su bovino es hibrido? \n s/S para SI
\n n/N para NO");

            stringHibrido = Console.ReadLine();

        }

        if (stringHibrido == "s")
        {

            booleanHibrido = true;

        }

        else if (stringHibrido == "n")
        {

```

```

        booleanHibrido = false;

    }

    //

    //ALIMENTACIÓN BOVINO

    bool tipoAlimentacionBovino = true;

    Console.WriteLine("Que tipo de alimentación recibe su Bovino?
\n" +

        " Coloque el número 1 si es: GRANO \n " +

        " Coloque el número 2 si es: PASTURA ");

    bool booleanAlimentacion = int.TryParse(Console.ReadLine(),
out int intAlimentacion);

    while (!booleanAlimentacion || (intAlimentacion != 1 &&
intAlimentacion != 2))

    {

        Console.WriteLine("Que tipo de alimentación recibe su
Bovino? \n" +

            " Coloque el número 1 si es: GRANO \n " +

            " Coloque el número 2 si es: PASTURA ");

        booleanAlimentacion = int.TryParse(Console.ReadLine(),
out intAlimentacion);

    }

    if (intAlimentacion == 1)

    {

        tipoAlimentacionBovino = true;

    }

    else if (intAlimentacion == 2)

    {

```

```

        tipoAlimentacionBovino = false;

    }

    //

    Bovino bovino = new Bovino(codigoCaravana, macho_hembraValor,
stringRaza, dateFecha, doubleCostoAdquisicion, doubleCostoAlimentacion
        , doublePesoBovino, booleanHibrido,
tipoAlimentacionBovino);

    AltaGanado(bovino);

    Console.WriteLine(bovino);

    Console.WriteLine("Presione 1 - Para Agregar Otro BOVINO. " +
        "\n 2 - Para Obtener La Lista De BOVINOS. " +
        "\n 3 o Cualquier Tecla - Para Salir.");

    texto = Console.ReadLine().ToLower();

    if (texto == "2")
    {

        int auxContador = 0;

        for (int i = 0; i < _listaGanado.Count; i++)
        {

            Ganado auxGanado = _listaGanado[i];

            Console.WriteLine(auxGanado);

            auxContador++;

        }

        Console.WriteLine($"Contamos con: {auxContador} Animales
Actualmente.");

    }

}

```

```

    }

    //GANANCIAS POR POTRERO(COSTO CRIANZA - POTENCIALES PRECIOS)

    public void SetGananciaPorPotrero()

    {

        for (int i = 1; i < _listaPotreros.Count; i++)

        {

            double valorXUnidad = 0;

            Potrero auxPotrero = _listaPotreros[i];

            for (int j = 0; j < auxPotrero._ListaGanadosEnPotrero.Count;
j++)

            {

                Ganado auxGanado = auxPotrero._ListaGanadosEnPotrero[j];

                valorXUnidad = valorXUnidad +
auxGanado.PotencialPrecioGanado();

            }

            auxPotrero.PotencialPrecioVenta = valorXUnidad;

        }

    }

    //ESTABLECER COSTO POR KILOGRAMO DE LANA DE LOS OVINOS

    public void EstablecerCostoKgLanaOvinos()

    {

        Console.WriteLine("Qué precio desea establecer por kilogramo de
lana de los" +

            " ovinos?.");

```

```

        bool booleanValor = double.TryParse(Console.ReadLine(), out
double doubleValor);

        while (!booleanValor || doubleValor == 0)

        {

            Console.WriteLine("(RECUERDE COLOCAR UN VALOR NUMÉRICO MAYOR
A CERO). Qué precio desea establecer por kilogramo de lana de los" +

                " ovinos?.");

            booleanValor = double.TryParse(Console.ReadLine(), out
doubleValor);

        }

        Ovino.PrecioKGLana = doubleValor;

    }

```

//LOGIN USUARIOS - POR CONSOLA - NUEVO NOMBRE MÉTODO

```

public void LoginUsersConsola()

{

    int iLista = 0;

    //EMAIL

    bool booleanEmail = false;

    Console.WriteLine("Ingrese su Email.");

    string emailEmpleado = Console.ReadLine();

    for (int i = 0; i < _listaEmpleados.Count; i++)

    {

        Empleado auxEmpleado = _listaEmpleados[i];

        if (auxEmpleado.Email == emailEmpleado)

        {

```



```

        booleanEmail = true;

        iLista = i;

        break;
    }
}

while (!booleanEmail)
{
    Console.WriteLine("Ingrese su Email correctamente.");

    emailEmpleado = Console.ReadLine();

    for (int i = 0; i < _listaEmpleados.Count; i++)
    {
        Empleado auxEmpleado = _listaEmpleados[i];

        if (auxEmpleado.Email == emailEmpleado)
        {
            booleanEmail = true;

            iLista = i;

            break;
        }
    }
}

//CONTRASEÑA

bool booleanPass = false;

Console.WriteLine("Ingrese su Contraseña.");

string passEmpleado = Console.ReadLine();

if (_listaEmpleados[iLista].Contraseña == passEmpleado)

```

```

{
    booleanPass = true;
}

while (!booleanPass)
{
    Console.WriteLine("Ingreso incorrecto de Contraseña.");
    passEmpleado = Console.ReadLine();

    if (_listaEmpleados[iLista].Contraseña == passEmpleado)
    {
        booleanPass = true;
    }
}

if (booleanPass)
{
    if (_listaEmpleados[iLista] is Capataz)
    {
        Console.WriteLine($"Bienvenido
{_listaEmpleados[iLista].Nombre}. Es un gusto verte por aquí nuevamente.");
    }

    else
    {
        Console.WriteLine($"Bienvenido
{_listaEmpleados[iLista].Nombre}. Es un gusto verte por aquí nuevamente.");
    }
}
}

```

```
}
```

```
//-----MÉTODOS NUEVOS-----
```

```
public Empleado GetEmpleadoXEmailYContraseña(string email, string  
contraseña)
```

```
{
```

```
    foreach (Empleado item in _listaEmpleados)
```

```
    {
```

```
        if (item.Email == email && item.Contraseña == contraseña)
```

```
        {
```

```
            return item;
```

```
        }
```

```
    }
```

```
    throw new Exception("Email y/o Contraseña incorrecto/s");
```

```
}
```

```
public Empleado GetEmpleadoXId(int? id)
```

```
{
```

```
    Empleado retorno = null;
```

```
    foreach (Empleado item in _listaEmpleados)
```

```
    {
```

```
        if (item.Id == id)
```

```
        {
```

```
            retorno = item;
```

```
            break;
```

```
        }
```

```

    }

    return retorno;
}

public void ActualizarPeon(Peon pPeon)
{
    pPeon.Validar();

    Peon peonBuscado = (Peon)GetEmpleadoXId(pPeon.Id);

    peonBuscado.Email = pPeon.Email;

    peonBuscado.Contraseña = pPeon.Contraseña;

    peonBuscado.Nombre = pPeon.Nombre;

    peonBuscado.FechaIngreso = pPeon.FechaIngreso;

    peonBuscado.EsResidente = pPeon.EsResidente;
}

public void ActualizarCapataz(Capataz pCapataz)
{
    pCapataz.Validar();

    Capataz capatazBuscado = (Capataz)GetEmpleadoXId(pCapataz.Id);

    capatazBuscado.Email = pCapataz.Email;

    capatazBuscado.Contraseña = pCapataz.Contraseña;

    capatazBuscado.Nombre = pCapataz.Nombre;

    capatazBuscado.FechaIngreso = pCapataz.FechaIngreso;

    capatazBuscado.CantidadPersonasACargo =
pCapataz.CantidadPersonasACargo;
}

public Ganado GetGanadoXNumCaravana(int numCaravana)

```

```
{

    Ganado retorno = null;

    foreach (Ganado item in _listaGanado)

    {

        if (item.NumCaravana == numCaravana)

        {

            retorno = item;

            break;

        }

    }

    return retorno;

}

public List<Ganado> GetGanados()

{

    return _listaGanado;

}

public Vacuna GetVacunaXNombreVacuna(string nombreVacuna)

{

    Vacuna retorno = null;

    foreach (Vacuna item in _listaVacunas)

    {

        if (item.Nombre == nombreVacuna)

        {

            retorno = item;

        }

    }

}
```

```

        }

        return retorno;
    }

    public List<AsignacionTarea> GetTareasNoRealizadasXPeon(Peon
peonLogueado)
    {
        List<AsignacionTarea> retorno = new List<AsignacionTarea>();

        foreach (AsignacionTarea item in _listaAsignacionTareas)
        {
            if (item.MyEmpleado.Id == peonLogueado.Id &&
!item.Tarea.FueCompletada)
            {
                retorno.Add(item);
            }
        }

        retorno.Sort();

        return retorno;
    }

    public AsignacionTarea GetAsignacionTareaPorPeon(int? idPeon, int
idTarea)
    {
        AsignacionTarea retorno = null;

        foreach (AsignacionTarea item in _listaAsignacionTareas)
        {
            if (item.MyEmpleado.Id == idPeon && idTarea == item.Tarea.ID)

```

```

        {

            retorno = item;

            break;

        }

    }

    return retorno;

}

public void CompletarAsignacionTarea(AsignacionTarea
pAsignacionTarea, string pComentario)

{

    AsignacionTarea asignacionTareaV =
GetAsignacionTareaPorPeon(pAsignacionTarea.MyEmpleado.Id,
pAsignacionTarea.Tarea.ID);

    if (pComentario != null)
    {

        asignacionTareaV.Tarea.FueCompletada = true;

        asignacionTareaV.Tarea.Comentario = pComentario;

        asignacionTareaV.Tarea.FechaCierre = DateTime.Today;

    }

    else

    {

        throw new Exception("Comentarios Obligatorios");

    }

}

public List<Potrero> GetPotreros()

{

```

```

        List<Potrero> retorno = new List<Potrero>();

        for (int i = 1; i < _listaPotreros.Count; i++)
        {
            if (!CapacidadMaximaSuperada(_listaPotreros[i]))
            {
                retorno.Add(_listaPotreros[i]);
            }
        }

        return retorno;
    }

    public Potrero GetPotreroXId(int id)
    {
        Potrero retorno = null;

        foreach (Potrero item in _listaPotreros)
        {
            if (item.ID == id)
            {
                retorno = item;

                break;
            }
        }

        return retorno;
    }

    public void MoverGanadoLibreAPotrero(Potrero potreroAMover, Ganado
pGanado)

```



```

{
    if (pGanado != null)
    {
        if (!pGanado.AnimalEsCria())
        {
            if (!CapacidadMaximaSuperada(potreroAMover))
            {
                potreroAMover._ListaGanadosEnPotrero.Add(pGanado);

                for (int i = 0; i <
_listraPotreros[0]._ListaGanadosEnPotrero.Count; i++)
                {
                    Ganado auxGanado =
_listraPotreros[0]._ListaGanadosEnPotrero[i];

                    if (auxGanado == pGanado)
                    {
_listraPotreros[0]._ListaGanadosEnPotrero.RemoveAt(i);

                    }

                }

            }
        }
        else
        {
            throw new Exception("Capacidad de Potrero
Superada.");
        }
    }
}

```

```

        else

        {

            throw new Exception("Este animal es Cría.");

        }

    }

    else

    {

        throw new Exception("No tiene Animales Libres para Mover");

    }

}

public List<Ganado> GetGanadoAlAireLibre()

{

    List<Ganado> retorno = new List<Ganado>();

    Potrero auxPotrero = _listaPotreros[0];

    for (int i = 0; i < auxPotrero._ListaGanadosEnPotrero.Count; i++)

    {

        Ganado auxGanado = auxPotrero._ListaGanadosEnPotrero[i];

        retorno.Add(auxGanado);

    }

    return retorno;

}

public void AsignarPotreroAAltaGanado(Ganado pGanado)

{

    _listaPotreros[0]._ListaGanadosEnPotrero.Add(pGanado);

}

```

```

public List<Peon> GetPeones()
{
    List<Peon> retorno = new List<Peon>();

    foreach (Empleado item in _listaEmpleados)
    {
        if (item is Peon)
        {
            retorno.Add((Peon)item);
        }
    }

    return retorno;
}

public List<AsignacionTarea> GetAsignacionTareasXIdPeon(int idPeon)
{
    List<AsignacionTarea> retorno = new List<AsignacionTarea>();

    foreach (AsignacionTarea item in _listaAsignacionTareas)
    {
        if (item.MyEmpleado.Id == idPeon)
        {
            retorno.Add(item);
        }
    }

    return retorno;
}

public List<Bovino> GetBovinosXPeso(int? pesoGanado)

```

```

{

    List<Bovino> retorno = new List<Bovino>();

    foreach (Ganado item in _listaGanado)
    {

        if (item is Bovino && item.PesoActual > pesoGanado)

        {

            retorno.Add((Bovino)item);

        }

    }

    return retorno;
}

public List<Ovino> GetOvinosXPeso(int? pesoGanado)
{

    List<Ovino> retorno = new List<Ovino>();

    foreach (Ganado item in _listaGanado)
    {

        if (item is Ovino && item.PesoActual > pesoGanado)

        {

            retorno.Add((Ovino)item);

        }

    }

    return retorno;
}

public void SetGananciaPorGanado()
{

```

```

        for (int i = 0; i < _listaGanado.Count; i++)
        {
            double valorXUnidad = 0; ;

            Ganado auxGanado = _listaGanado[i];

            valorXUnidad = auxGanado.PotencialPrecioGanado() -
CostoCrianzaGanado(auxGanado);

            auxGanado.GananciaVentaIndividual = valorXUnidad;

        }
    }

//-----
-----

private void PrecargarCapataz()
{
    Capataz capataz1 = new Capataz("capataz0011@gmail.com",
"capatazcatapaz1", "JuanPedro", new DateTime(2020, 01, 01), 5);

    Capataz capataz2 = new Capataz("capataz0022@gmail.com",
"capatazcatapaz2", "RaulMaria", new DateTime(2019, 01, 01), 5);

    AltaEmpleado(capataz1);

    AltaEmpleado(capataz2);

}

private void PrecargarPeon()
{
    Peon peon1 = new Peon("email011@jmail.com", "passw1234ord",
"Juan", new DateTime(2023, 01, 15), true);

    Peon peon2 = new Peon("email022@jmail.com", "passw23ord",
"Maria", new DateTime(2023, 03, 01), false);

```

```

        Peon peon3 = new Peon("email033@jmail.com", "contr123ase",
"Carlos", new DateTime(2019, 01, 01), true);

        Peon peon4 = new Peon("email044@jmail.com", "secre123eto1",
"Laura", new DateTime(2021, 06, 17), false);

        Peon peon5 = new Peon("email055@jmail.com", "12345123678",
"Pedro", new DateTime(2020, 02, 019), true);

        Peon peon6 = new Peon("email066@jmail.com", "pass123pass", "Ana",
new DateTime(2019, 01, 01), false);

        Peon peon7 = new Peon("email077@jmail.com", "cont123ra12",
"Pablo", new DateTime(2019, 01, 01), true);

        Peon peon8 = new Peon("email088@jmail.com", "qwer123tyui",
"Sofia", new DateTime(2019, 01, 01), false);

        Peon peon9 = new Peon("email099@jmail.com", "abcd123ef12",
"Luis", new DateTime(2023, 12, 11), true);

        Peon peon10 = new Peon("email100@jmail.com", "pas123spass",
"Elena", new DateTime(2019, 01, 01), false);

        AltaEmpleado(peon1);

        AltaEmpleado(peon2);

        AltaEmpleado(peon3);

        AltaEmpleado(peon4);

        AltaEmpleado(peon5);

        AltaEmpleado(peon6);

        AltaEmpleado(peon7);

        AltaEmpleado(peon8);

        AltaEmpleado(peon9);

        AltaEmpleado(peon10);

    }

    private void PrecargarTarea()

```

```
{

    Tarea tarea1 = new Tarea("Alimentar al ganado bovino en el
pastizal", DateTime.Today.AddDays(1), false, DateTime.Today.AddDays(6),
"NG");

    Tarea tarea2 = new Tarea("Ordenar y limpiar el establo de ovejas",
DateTime.Today.AddDays(2), false, DateTime.Today.AddDays(7), "NG");

    Tarea tarea3 = new Tarea("Separar terneros de las vacas lecheras",
DateTime.Today.AddDays(3), false, DateTime.Today.AddDays(8), "NG");

    Tarea tarea4 = new Tarea("Cortar la lana de las ovejas",
DateTime.Today.AddDays(4), false, DateTime.Today.AddDays(9), "NG");

    Tarea tarea5 = new Tarea("Administrar suplementos nutricionales
al ganado bovino", DateTime.Today.AddDays(5), false,
DateTime.Today.AddDays(10), "NG");

    Tarea tarea6 = new Tarea("Revisar cercas para contener el ganado
bovino", DateTime.Today.AddDays(6), false, DateTime.Today.AddDays(11), "NG");

    Tarea tarea7 = new Tarea("Ordenar los corrales de ovejas",
DateTime.Today.AddDays(7), false, DateTime.Today.AddDays(12), "NG");

    Tarea tarea8 = new Tarea("Vacunar corderos recién nacidos",
DateTime.Today.AddDays(8), false, DateTime.Today.AddDays(13), "NG");

    Tarea tarea9 = new Tarea("Supervisar el parto de las vacas
preñadas", DateTime.Today.AddDays(9), false, DateTime.Today.AddDays(14),
"NG");

    Tarea tarea10 = new Tarea("Mover el ganado bovino al pasto
fresco", DateTime.Today.AddDays(10), false, DateTime.Today.AddDays(15),
"NG");

    Tarea tarea11 = new Tarea("Contar cabezas de ganado bovino para
el inventario", DateTime.Today.AddDays(11), false,
DateTime.Today.AddDays(16), "NG");

    Tarea tarea12 = new Tarea("Revisar la salud de los carneros
reproductores", DateTime.Today.AddDays(12), false,
DateTime.Today.AddDays(20), "NG");

    Tarea tarea13 = new Tarea("Vacunar el ganado bovino",
DateTime.Today.AddDays(13), true, DateTime.Today.AddDays(13), "NG");
```

```
Tarea tarea14 = new Tarea("Revisar la condición corporal de las  
ovejas", DateTime.Today.AddDays(14), true, DateTime.Today.AddDays(14), "NG");
```

```
Tarea tarea15 = new Tarea("Ordenar y clasificar terneros por edad  
y peso", DateTime.Today.AddDays(15), true, DateTime.Today.AddDays(15), "NG");
```

```
Tarea tarea16 = new Tarea("Desparasitar el ganado ovino",  
DateTime.Today.AddDays(16), true, DateTime.Today.AddDays(16), "NG");
```

```
Tarea tarea17 = new Tarea("Supervisar el parto de una vaca  
preñada", DateTime.Today.AddDays(17), true, DateTime.Today.AddDays(17),  
"NG");
```

```
Tarea tarea18 = new Tarea("Revisar la valla perimetral del corral  
de ovejas", DateTime.Today.AddDays(18), true, DateTime.Today.AddDays(18),  
"NG");
```

```
Tarea tarea19 = new Tarea("Alimentar al ganado bovino con  
suplementos nutritivos", DateTime.Today.AddDays(19), true,  
DateTime.Today.AddDays(19), "NG");
```

```
Tarea tarea20 = new Tarea("Contar el número de ovejas en el  
rebaño", DateTime.Today.AddDays(20), true, DateTime.Today.AddDays(20), "NG");
```

```
Tarea tarea21 = new Tarea("Podar árboles en el área de pastoreo  
de ovejas", DateTime.Today.AddDays(21), true, DateTime.Today.AddDays(21),  
"NG");
```

```
Tarea tarea22 = new Tarea("Limpiar bebederos del ganado bovino",  
DateTime.Today.AddDays(22), true, DateTime.Today.AddDays(22), "NG");
```

```
Tarea tarea23 = new Tarea("Revisión del sistema de riego en los  
pastizales", DateTime.Today.AddDays(23), false, DateTime.Today.AddDays(30),  
"NG");
```

```
Tarea tarea24 = new Tarea("Aplicación de suplementos minerales al  
ganado bovino", DateTime.Today.AddDays(24), false,  
DateTime.Today.AddDays(31), "NG");
```

```
Tarea tarea25 = new Tarea("Clasificación de corderos para venta  
en el mercado", DateTime.Today.AddDays(25), false,  
DateTime.Today.AddDays(32), "NG");
```



```
Tarea tarea26 = new Tarea("Control de malezas en los pastizales  
de invierno", DateTime.Today.AddDays(26), false, DateTime.Today.AddDays(31),  
"NG");
```

```
Tarea tarea27 = new Tarea("Revisión de la instalación eléctrica  
en el establo", DateTime.Today.AddDays(27), false,  
DateTime.Today.AddDays(32), "NG");
```

```
Tarea tarea28 = new Tarea("Entrenamiento de perros pastores para  
el manejo del ganado", DateTime.Today.AddDays(28), false,  
DateTime.Today.AddDays(33), "NG");
```

```
Tarea tarea29 = new Tarea("Revisiones de rutina del estado de  
salud del ganado", DateTime.Today.AddDays(29), false,  
DateTime.Today.AddDays(34), "NG");
```

```
Tarea tarea30 = new Tarea("Evaluación de la eficacia del programa  
de alimentación", DateTime.Today.AddDays(30), false,  
DateTime.Today.AddDays(35), "NG");
```

```
AltaTarea(tarea1);
```

```
AltaTarea(tarea2);
```

```
AltaTarea(tarea3);
```

```
AltaTarea(tarea4);
```

```
AltaTarea(tarea5);
```

```
AltaTarea(tarea6);
```

```
AltaTarea(tarea7);
```

```
AltaTarea(tarea8);
```

```
AltaTarea(tarea9);
```

```
AltaTarea(tarea10);
```

```
AltaTarea(tarea11);
```

```
AltaTarea(tarea12);
```

```
AltaTarea(tarea13);
```

```
AltaTarea(tarea14);
```

```
        AltaTarea(tarea15);

        AltaTarea(tarea16);

        AltaTarea(tarea17);

        AltaTarea(tarea18);

        AltaTarea(tarea19);

        AltaTarea(tarea20);

        AltaTarea(tarea21);

        AltaTarea(tarea22);

        AltaTarea(tarea23);

        AltaTarea(tarea24);

        AltaTarea(tarea25);

        AltaTarea(tarea26);

        AltaTarea(tarea27);

        AltaTarea(tarea28);

        AltaTarea(tarea29);

        AltaTarea(tarea30);

    }

    private void PrecargarAsignacionTarea()

    {

        AsignacionTarea tar1 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[0]);

        AsignacionTarea tar2 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[3]);

        AsignacionTarea tar3 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[4]);
```

```
        AsignacionTarea tar4 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[5]);

        AsignacionTarea tar5 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[6]);

        AsignacionTarea tar6 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[7]);

        AsignacionTarea tar7 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[8]);

        AsignacionTarea tar8 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[9]);

        AsignacionTarea tar9 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[10]);

        AsignacionTarea tar10 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[11]);

        AsignacionTarea tar11 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[12]);

        AsignacionTarea tar12 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[13]);

        AsignacionTarea tar13 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[14]);

        AsignacionTarea tar14 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[15]);

        AsignacionTarea tar15 = new AsignacionTarea(_listaEmpleados[2],
        _listaTareas[16]);

        //

        AsignacionTarea tar16 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[2]);

        AsignacionTarea tar17 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[16]);

        AsignacionTarea tar18 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[17]);
```

```
        AsignacionTarea tar19 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[18]);

        AsignacionTarea tar20 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[20]);

        AsignacionTarea tar21 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[21]);

        AsignacionTarea tar22 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[22]);

        AsignacionTarea tar23 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[23]);

        AsignacionTarea tar24 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[24]);

        AsignacionTarea tar25 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[27]);

        AsignacionTarea tar26 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[26]);

        AsignacionTarea tar27 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[29]);

        AsignacionTarea tar28 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[28]);

        AsignacionTarea tar29 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[1]);

        AsignacionTarea tar30 = new AsignacionTarea(_listaEmpleados[3],
        _listaTareas[7]);

        //

        AsignacionTarea tar31 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[1]);

        AsignacionTarea tar32 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[29]);

        AsignacionTarea tar33 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[28]);
```

```
        AsignacionTarea tar34 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[27]);

        AsignacionTarea tar35 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[26]);

        AsignacionTarea tar36 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[25]);

        AsignacionTarea tar37 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[24]);

        AsignacionTarea tar38 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[23]);

        AsignacionTarea tar39 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[22]);

        AsignacionTarea tar40 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[21]);

        AsignacionTarea tar41 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[20]);

        AsignacionTarea tar42 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[19]);

        AsignacionTarea tar43 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[18]);

        AsignacionTarea tar44 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[17]);

        AsignacionTarea tar45 = new AsignacionTarea(_listaEmpleados[4],
        _listaTareas[16]);

        //

        AsignacionTarea tar46 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[15]);

        AsignacionTarea tar47 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[14]);

        AsignacionTarea tar48 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[13]);
```

```
        AsignacionTarea tar49 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[12]);

        AsignacionTarea tar50 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[11]);

        AsignacionTarea tar51 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[10]);

        AsignacionTarea tar52 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[9]);

        AsignacionTarea tar53 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[8]);

        AsignacionTarea tar54 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[7]);

        AsignacionTarea tar55 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[1]);

        AsignacionTarea tar56 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[2]);

        AsignacionTarea tar57 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[5]);

        AsignacionTarea tar58 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[4]);

        AsignacionTarea tar59 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[18]);

        AsignacionTarea tar60 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[28]);

        AsignacionTarea tar61 = new AsignacionTarea(_listaEmpleados[5],
        _listaTareas[29]);

        //

        AsignacionTarea tar62 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[29]);

        AsignacionTarea tar63 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[20]);
```

```
        AsignacionTarea tar64 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[21]);

        AsignacionTarea tar65 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[22]);

        AsignacionTarea tar66 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[23]);

        AsignacionTarea tar67 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[24]);

        AsignacionTarea tar68 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[25]);

        AsignacionTarea tar69 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[19]);

        AsignacionTarea tar70 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[2]);

        AsignacionTarea tar71 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[3]);

        AsignacionTarea tar72 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[4]);

        AsignacionTarea tar73 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[5]);

        AsignacionTarea tar74 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[1]);

        AsignacionTarea tar75 = new AsignacionTarea(_listaEmpleados[6],
        _listaTareas[7]);

        //

        AsignacionTarea tar76 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[8]);

        AsignacionTarea tar77 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[12]);

        AsignacionTarea tar78 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[13]);
```

```
        AsignacionTarea tar79 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[14]);

        AsignacionTarea tar80 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[15]);

        AsignacionTarea tar81 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[16]);

        AsignacionTarea tar82 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[17]);

        AsignacionTarea tar83 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[18]);

        AsignacionTarea tar84 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[26]);

        AsignacionTarea tar85 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[1]);

        AsignacionTarea tar86 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[29]);

        AsignacionTarea tar87 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[27]);

        AsignacionTarea tar88 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[23]);

        AsignacionTarea tar89 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[2]);

        AsignacionTarea tar90 = new AsignacionTarea(_listaEmpleados[7],
        _listaTareas[5]);

        //

        AsignacionTarea tar91 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[5]);

        AsignacionTarea tar92 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[6]);

        AsignacionTarea tar93 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[7]);
```



```
        AsignacionTarea tar94 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[8]);

        AsignacionTarea tar95 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[9]);

        AsignacionTarea tar96 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[10]);

        AsignacionTarea tar97 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[11]);

        AsignacionTarea tar98 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[12]);

        AsignacionTarea tar99 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[13]);

        AsignacionTarea tar100 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[14]);

        AsignacionTarea tar101 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[15]);

        AsignacionTarea tar102 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[16]);

        AsignacionTarea tar103 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[17]);

        AsignacionTarea tar104 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[18]);

        AsignacionTarea tar105 = new AsignacionTarea(_listaEmpleados[8],
        _listaTareas[19]);

        //

        AsignacionTarea tar106 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[10]);

        AsignacionTarea tar107 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[11]);
```

```
        AsignacionTarea tar108 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[12]);

        AsignacionTarea tar109 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[13]);

        AsignacionTarea tar110 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[14]);

        AsignacionTarea tar111 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[15]);

        AsignacionTarea tar112 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[26]);

        AsignacionTarea tar113 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[27]);

        AsignacionTarea tar114 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[28]);

        AsignacionTarea tar115 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[29]);

        AsignacionTarea tar116 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[7]);

        AsignacionTarea tar117 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[1]);

        AsignacionTarea tar118 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[2]);

        AsignacionTarea tar119 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[3]);

        AsignacionTarea tar120 = new AsignacionTarea(_listaEmpleados[9],
        _listaTareas[4]);

        //

        AsignacionTarea tar121 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[5]);

        AsignacionTarea tar122 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[6]);
```

```
        AsignacionTarea tar123 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[7]);

        AsignacionTarea tar124 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[8]);

        AsignacionTarea tar125 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[9]);

        AsignacionTarea tar126 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[10]);

        AsignacionTarea tar127 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[11]);

        AsignacionTarea tar128 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[21]);

        AsignacionTarea tar129 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[22]);

        AsignacionTarea tar130 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[23]);

        AsignacionTarea tar131 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[24]);

        AsignacionTarea tar132 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[25]);

        AsignacionTarea tar133 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[26]);

        AsignacionTarea tar134 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[27]);

        AsignacionTarea tar135 = new AsignacionTarea(_listaEmpleados[10],
        _listaTareas[28]);

        //

        AsignacionTarea tar136 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[22]);

        AsignacionTarea tar137 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[9]);
```

```
        AsignacionTarea tar138 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[1]);

        AsignacionTarea tar139 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[11]);

        AsignacionTarea tar140 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[12]);

        AsignacionTarea tar141 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[13]);

        AsignacionTarea tar142 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[17]);

        AsignacionTarea tar143 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[14]);

        AsignacionTarea tar144 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[15]);

        AsignacionTarea tar145 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[16]);

        AsignacionTarea tar146 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[18]);

        AsignacionTarea tar147 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[19]);

        AsignacionTarea tar148 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[20]);

        AsignacionTarea tar149 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[21]);

        AsignacionTarea tar150 = new AsignacionTarea(_listaEmpleados[11],
        _listaTareas[24]);

        AltaAsignacionTarea(tar1);

        AltaAsignacionTarea(tar2);

        AltaAsignacionTarea(tar3);

        AltaAsignacionTarea(tar4);
```

AltaAsignacionTarea(tar5);
AltaAsignacionTarea(tar6);
AltaAsignacionTarea(tar7);
AltaAsignacionTarea(tar8);
AltaAsignacionTarea(tar9);
AltaAsignacionTarea(tar10);
AltaAsignacionTarea(tar11);
AltaAsignacionTarea(tar12);
AltaAsignacionTarea(tar13);
AltaAsignacionTarea(tar14);
AltaAsignacionTarea(tar15);
AltaAsignacionTarea(tar16);
AltaAsignacionTarea(tar17);
AltaAsignacionTarea(tar18);
AltaAsignacionTarea(tar19);
AltaAsignacionTarea(tar20);
AltaAsignacionTarea(tar21);
AltaAsignacionTarea(tar22);
AltaAsignacionTarea(tar23);
AltaAsignacionTarea(tar24);
AltaAsignacionTarea(tar25);
AltaAsignacionTarea(tar26);
AltaAsignacionTarea(tar27);
AltaAsignacionTarea(tar28);
AltaAsignacionTarea(tar29);

AltaAsignacionTarea(tar30);
AltaAsignacionTarea(tar31);
AltaAsignacionTarea(tar32);
AltaAsignacionTarea(tar33);
AltaAsignacionTarea(tar34);
AltaAsignacionTarea(tar35);
AltaAsignacionTarea(tar36);
AltaAsignacionTarea(tar37);
AltaAsignacionTarea(tar38);
AltaAsignacionTarea(tar39);
AltaAsignacionTarea(tar40);
AltaAsignacionTarea(tar41);
AltaAsignacionTarea(tar42);
AltaAsignacionTarea(tar43);
AltaAsignacionTarea(tar44);
AltaAsignacionTarea(tar45);
AltaAsignacionTarea(tar46);
AltaAsignacionTarea(tar47);
AltaAsignacionTarea(tar48);
AltaAsignacionTarea(tar49);
AltaAsignacionTarea(tar50);
AltaAsignacionTarea(tar51);
AltaAsignacionTarea(tar52);
AltaAsignacionTarea(tar53);
AltaAsignacionTarea(tar54);

AltaAsignacionTarea(tar55);
AltaAsignacionTarea(tar56);
AltaAsignacionTarea(tar57);
AltaAsignacionTarea(tar58);
AltaAsignacionTarea(tar59);
AltaAsignacionTarea(tar60);
AltaAsignacionTarea(tar61);
AltaAsignacionTarea(tar62);
AltaAsignacionTarea(tar63);
AltaAsignacionTarea(tar64);
AltaAsignacionTarea(tar65);
AltaAsignacionTarea(tar66);
AltaAsignacionTarea(tar67);
AltaAsignacionTarea(tar68);
AltaAsignacionTarea(tar69);
AltaAsignacionTarea(tar70);
AltaAsignacionTarea(tar71);
AltaAsignacionTarea(tar72);
AltaAsignacionTarea(tar73);
AltaAsignacionTarea(tar74);
AltaAsignacionTarea(tar75);
AltaAsignacionTarea(tar76);
AltaAsignacionTarea(tar77);
AltaAsignacionTarea(tar78);
AltaAsignacionTarea(tar79);

AltaAsignacionTarea(tar80);
AltaAsignacionTarea(tar81);
AltaAsignacionTarea(tar82);
AltaAsignacionTarea(tar83);
AltaAsignacionTarea(tar84);
AltaAsignacionTarea(tar85);
AltaAsignacionTarea(tar86);
AltaAsignacionTarea(tar87);
AltaAsignacionTarea(tar88);
AltaAsignacionTarea(tar89);
AltaAsignacionTarea(tar90);
AltaAsignacionTarea(tar91);
AltaAsignacionTarea(tar92);
AltaAsignacionTarea(tar93);
AltaAsignacionTarea(tar94);
AltaAsignacionTarea(tar95);
AltaAsignacionTarea(tar96);
AltaAsignacionTarea(tar97);
AltaAsignacionTarea(tar98);
AltaAsignacionTarea(tar99);
AltaAsignacionTarea(tar100);
AltaAsignacionTarea(tar101);
AltaAsignacionTarea(tar102);
AltaAsignacionTarea(tar103);
AltaAsignacionTarea(tar104);

AltaAsignacionTarea(tar105);
AltaAsignacionTarea(tar106);
AltaAsignacionTarea(tar107);
AltaAsignacionTarea(tar108);
AltaAsignacionTarea(tar109);
AltaAsignacionTarea(tar110);
AltaAsignacionTarea(tar111);
AltaAsignacionTarea(tar112);
AltaAsignacionTarea(tar113);
AltaAsignacionTarea(tar114);
AltaAsignacionTarea(tar115);
AltaAsignacionTarea(tar116);
AltaAsignacionTarea(tar117);
AltaAsignacionTarea(tar118);
AltaAsignacionTarea(tar119);
AltaAsignacionTarea(tar120);
AltaAsignacionTarea(tar121);
AltaAsignacionTarea(tar122);
AltaAsignacionTarea(tar123);
AltaAsignacionTarea(tar124);
AltaAsignacionTarea(tar125);
AltaAsignacionTarea(tar126);
AltaAsignacionTarea(tar127);
AltaAsignacionTarea(tar128);
AltaAsignacionTarea(tar129);

```
        AltaAsignacionTarea(tar130);

        AltaAsignacionTarea(tar131);

        AltaAsignacionTarea(tar132);

        AltaAsignacionTarea(tar133);

        AltaAsignacionTarea(tar134);

        AltaAsignacionTarea(tar135);

        AltaAsignacionTarea(tar136);

        AltaAsignacionTarea(tar137);

        AltaAsignacionTarea(tar138);

        AltaAsignacionTarea(tar139);

        AltaAsignacionTarea(tar140);

        AltaAsignacionTarea(tar141);

        AltaAsignacionTarea(tar142);

        AltaAsignacionTarea(tar143);

        AltaAsignacionTarea(tar144);

        AltaAsignacionTarea(tar145);

        AltaAsignacionTarea(tar146);

        AltaAsignacionTarea(tar147);

        AltaAsignacionTarea(tar148);

        AltaAsignacionTarea(tar149);

        AltaAsignacionTarea(tar150);

    }

    private void PrecargarOvino()

    {
```

```
Ovino ovino1 = new Ovino(10000001, true, "Merino", new
DateTime(2023, 4, 10), 200.50, 100.25, 40.5, true);

Ovino ovino2 = new Ovino(10000002, false, "Dorper", new
DateTime(2019, 2, 15), 180.75, 95.20, 38.0, true);

Ovino ovino3 = new Ovino(10000003, true, "Suffolk", new
DateTime(2018, 3, 20), 220.0, 110.0, 45.0, false);

Ovino ovino4 = new Ovino(10000004, false, "Rambouillet", new
DateTime(2017, 4, 25), 190.25, 98.50, 42.0, false);

Ovino ovino5 = new Ovino(10000005, true, "Romney", new
DateTime(2016, 5, 30), 205.0, 105.75, 43.5, false);

Ovino ovino6 = new Ovino(10000006, false, "Lincoln", new
DateTime(2015, 6, 5), 195.80, 97.25, 41.0, true);

Ovino ovino7 = new Ovino(10000007, true, "Columbia", new
DateTime(2014, 7, 10), 215.60, 108.20, 44.0, true);

Ovino ovino8 = new Ovino(10000008, false, "Merino", new
DateTime(2013, 8, 15), 200.50, 100.25, 40.5, false);

Ovino ovino9 = new Ovino(10000009, false, "Dorper", new
DateTime(2012, 9, 20), 180.75, 95.20, 38.0, true);

Ovino ovino10 = new Ovino(10000010, true, "Suffolk", new
DateTime(2011, 10, 25), 220.0, 110.0, 45.0, false);

Ovino ovino11 = new Ovino(10000011, false, "Rambouillet", new
DateTime(2010, 11, 30), 190.25, 98.50, 42.0, false);

Ovino ovino12 = new Ovino(10000012, false, "Romney", new
DateTime(2009, 12, 5), 205.0, 105.75, 43.5, true);

Ovino ovino13 = new Ovino(10000013, false, "Lincoln", new
DateTime(2008, 1, 10), 195.80, 97.25, 41.0, false);

Ovino ovino14 = new Ovino(10000014, true, "Columbia", new
DateTime(2007, 2, 15), 215.60, 108.20, 44.0, true);

Ovino ovino15 = new Ovino(10000015, false, "Merino", new
DateTime(2006, 3, 20), 200.50, 100.25, 40.5, false);
```

```
Ovino ovino16 = new Ovino(10000016, false, "Dorper", new
DateTime(2005, 4, 25), 180.75, 95.20, 38.0, true);
```

```
Ovino ovino17 = new Ovino(10000017, true, "Suffolk", new
DateTime(2004, 5, 30), 220.0, 110.0, 45.0, false);
```

```
Ovino ovino18 = new Ovino(10000018, false, "Rambouillet", new
DateTime(2003, 6, 5), 190.25, 98.50, 42.0, false);
```

```
Ovino ovino19 = new Ovino(10000019, false, "Romney", new
DateTime(2002, 7, 10), 205.0, 105.75, 43.5, true);
```

```
Ovino ovino20 = new Ovino(10000020, false, "Lincoln", new
DateTime(2001, 8, 15), 195.80, 97.25, 41.0, true);
```

```
Ovino ovino21 = new Ovino(10000021, true, "Merino", new
DateTime(2020, 1, 10), 200.50, 100.25, 40.5, true);
```

```
Ovino ovino22 = new Ovino(10000022, false, "Dorper", new
DateTime(2019, 2, 15), 180.75, 95.20, 38.0, false);
```

```
Ovino ovino23 = new Ovino(10000023, false, "Suffolk", new
DateTime(2018, 3, 20), 220.0, 110.0, 45.0, true);
```

```
Ovino ovino24 = new Ovino(10000024, false, "Rambouillet", new
DateTime(2017, 4, 25), 190.25, 98.50, 42.0, false);
```

```
Ovino ovino25 = new Ovino(10000025, true, "Romney", new
DateTime(2016, 5, 30), 205.0, 105.75, 43.5, true);
```

```
Ovino ovino26 = new Ovino(10000026, false, "Lincoln", new
DateTime(2015, 6, 5), 195.80, 97.25, 41.0, false);
```

```
Ovino ovino27 = new Ovino(10000027, false, "Columbia", new
DateTime(2014, 7, 10), 215.60, 108.20, 44.0, true);
```

```
Ovino ovino28 = new Ovino(10000028, true, "Merino", new
DateTime(2013, 8, 15), 200.50, 100.25, 40.5, true);
```

```
Ovino ovino29 = new Ovino(10000029, false, "Dorper", new
DateTime(2012, 9, 20), 180.75, 95.20, 38.0, false);
```

```
Ovino ovino30 = new Ovino(10000030, true, "Suffolk", new
DateTime(2011, 10, 25), 220.0, 110.0, 45.0, true);
```

```
AltaGanado(ovino1);
```

AltaGanado(ovino2);
AltaGanado(ovino3);
AltaGanado(ovino4);
AltaGanado(ovino5);
AltaGanado(ovino6);
AltaGanado(ovino7);
AltaGanado(ovino8);
AltaGanado(ovino9);
AltaGanado(ovino10);
AltaGanado(ovino11);
AltaGanado(ovino12);
AltaGanado(ovino13);
AltaGanado(ovino14);
AltaGanado(ovino15);
AltaGanado(ovino16);
AltaGanado(ovino17);
AltaGanado(ovino18);
AltaGanado(ovino19);
AltaGanado(ovino20);
AltaGanado(ovino21);
AltaGanado(ovino22);
AltaGanado(ovino23);
AltaGanado(ovino24);
AltaGanado(ovino25);
AltaGanado(ovino26);

```

        AltaGanado(ovino27);

        AltaGanado(ovino28);

        AltaGanado(ovino29);

        AltaGanado(ovino30);

    }

    private void PrecargarBovino()

    {

        Bovino bovino1 = new Bovino(10000121, true, "Angus", new
DateTime(2023, 04, 15), 500.0, 250.0, 700.0, true, false);

        Bovino bovino2 = new Bovino(10000122, false, "Hereford", new
DateTime(2020, 3, 20), 480.0, 230.0, 680.0, false, true);

        Bovino bovino3 = new Bovino(10000123, true, "Simmental", new
DateTime(2018, 12, 5), 520.0, 260.0, 720.0, false, true);

        Bovino bovino4 = new Bovino(10000124, false, "Limousin", new
DateTime(2021, 7, 10), 450.0, 220.0, 650.0, true, false);

        Bovino bovino5 = new Bovino(10000125, true, "Charolais", new
DateTime(2019, 5, 15), 530.0, 270.0, 730.0, false, false);

        Bovino bovino6 = new Bovino(10000126, false, "Brahman", new
DateTime(2020, 9, 20), 470.0, 240.0, 670.0, false, true);

        Bovino bovino7 = new Bovino(10000127, true, "Gelbvieh", new
DateTime(2018, 11, 25), 510.0, 250.0, 710.0, false, false);

        Bovino bovino8 = new Bovino(10000128, false, "Brown Swiss", new
DateTime(2021, 4, 5), 460.0, 230.0, 660.0, false, false);

        Bovino bovino9 = new Bovino(10000129, true, "Shorthorn", new
DateTime(2020, 2, 10), 540.0, 280.0, 740.0, false, false);

        Bovino bovino10 = new Bovino(10001201, false, "Murray Grey", new
DateTime(2019, 8, 15), 440.0, 210.0, 630.0, false, false);

        Bovino bovino11 = new Bovino(11001211, true, "Red Angus", new
DateTime(2021, 1, 20), 520.0, 260.0, 720.0, true, false);

```

```
Bovino bovino12 = new Bovino(12001212, false, "Jersey", new
DateTime(2018, 10, 25), 480.0, 240.0, 680.0, true, true);
```

```
Bovino bovino13 = new Bovino(13001213, true, "Holstein", new
DateTime(2020, 6, 10), 550.0, 290.0, 750.0, false, false);
```

```
Bovino bovino14 = new Bovino(14001214, false, "Texas Longhorn",
new DateTime(2019, 12, 15), 430.0, 220.0, 620.0, false, false);
```

```
Bovino bovino15 = new Bovino(15001215, true, "Santa Gertrudis",
new DateTime(2021, 3, 20), 530.0, 270.0, 730.0, false, false);
```

```
Bovino bovino16 = new Bovino(16001216, false, "Belgian Blue", new
DateTime(2019, 9, 25), 450.0, 230.0, 650.0, false, false);
```

```
Bovino bovino17 = new Bovino(17001217, true, "Highland", new
DateTime(2020, 2, 10), 540.0, 280.0, 740.0, true, true);
```

```
Bovino bovino18 = new Bovino(18001218, false, "Dexter", new
DateTime(2018, 11, 15), 420.0, 210.0, 610.0, true, false);
```

```
Bovino bovino19 = new Bovino(19001219, true, "Devon", new
DateTime(2021, 5, 20), 560.0, 300.0, 760.0, true, false);
```

```
Bovino bovino20 = new Bovino(20001220, false, "Aberdeen Angus",
new DateTime(2019, 7, 25), 470.0, 240.0, 670.0, true, true);
```

```
Bovino bovino21 = new Bovino(10001221, false, "Murray Grey", new
DateTime(2019, 8, 15), 440.0, 210.0, 630.0, false, false);
```

```
Bovino bovino22 = new Bovino(11001222, true, "Red Angus", new
DateTime(2021, 1, 20), 520.0, 260.0, 720.0, true, true);
```

```
Bovino bovino23 = new Bovino(12001223, false, "Jersey", new
DateTime(2018, 10, 25), 480.0, 240.0, 680.0, false, false);
```

```
Bovino bovino24 = new Bovino(13001224, true, "Holstein", new
DateTime(2020, 6, 10), 550.0, 290.0, 750.0, true, false);
```

```
Bovino bovino25 = new Bovino(14001225, false, "Texas Longhorn",
new DateTime(2019, 12, 15), 430.0, 220.0, 620.0, true, false);
```

```
Bovino bovino26 = new Bovino(15001226, true, "Santa Gertrudis",
new DateTime(2021, 3, 20), 530.0, 270.0, 730.0, true, true);
```

```
Bovino bovino27 = new Bovino(16001227, false, "Belgian Blue", new  
DateTime(2019, 9, 25), 450.0, 230.0, 650.0, true, false);
```

```
Bovino bovino28 = new Bovino(17001228, true, "Highland", new  
DateTime(2020, 2, 10), 540.0, 280.0, 740.0, true, true);
```

```
Bovino bovino29 = new Bovino(18001229, false, "Dexter", new  
DateTime(2018, 11, 15), 420.0, 210.0, 610.0, true, true);
```

```
Bovino bovino30 = new Bovino(19001230, true, "Devon", new  
DateTime(2021, 5, 20), 560.0, 300.0, 760.0, false, true);
```

```
AltaGanado(bovino1);
```

```
AltaGanado(bovino2);
```

```
AltaGanado(bovino3);
```

```
AltaGanado(bovino4);
```

```
AltaGanado(bovino5);
```

```
AltaGanado(bovino6);
```

```
AltaGanado(bovino7);
```

```
AltaGanado(bovino8);
```

```
AltaGanado(bovino9);
```

```
AltaGanado(bovino10);
```

```
AltaGanado(bovino11);
```

```
AltaGanado(bovino12);
```

```
AltaGanado(bovino13);
```

```
AltaGanado(bovino14);
```

```
AltaGanado(bovino15);
```

```
AltaGanado(bovino16);
```

```
AltaGanado(bovino17);
```

```
AltaGanado(bovino18);
```



```
        AltaGanado(bovino19);

        AltaGanado(bovino20);

        AltaGanado(bovino21);

        AltaGanado(bovino22);

        AltaGanado(bovino23);

        AltaGanado(bovino24);

        AltaGanado(bovino25);

        AltaGanado(bovino26);

        AltaGanado(bovino27);

        AltaGanado(bovino28);

        AltaGanado(bovino29);

        AltaGanado(bovino30);

    }

    private void PrecargarVacuna()

    {

        Vacuna vacuna1 = new Vacuna("Fiebre aftosa", "Protege contra el
virus de la fiebre aftosa", "Virus de la fiebre aftosa");

        Vacuna vacuna2 = new Vacuna("Brucelosis", "Ayuda a prevenir la
infección por Brucella abortus", "Brucella abortus");

        Vacuna vacuna3 = new Vacuna("Tuberculosis bovina", "Reduce el
riesgo de infección por Mycobacterium bovis", "Mycobacterium bovis");

        Vacuna vacuna4 = new Vacuna("Leptospirosis", "Previene la
infección por Leptospira interrogans serovar Hardjo", "Leptospira interrogans
serovar Hardjo");

        Vacuna vacuna5 = new Vacuna("Fiebre del Valle del Rift", "Protege
contra el virus de la fiebre del Valle del Rift", "Virus de la fiebre del
Valle del Rift");
```

```
Vacuna vacuna6 = new Vacuna("Clostridiosis", "Ayuda a prevenir  
las enfermedades causadas por Clostridium spp.", "Clostridium spp.");
```

```
Vacuna vacuna7 = new Vacuna("Enfermedad de la mucosa", "Protege  
contra Mycoplasma mycoides subsp. mycoides SC", "Mycoplasma mycoides subsp.  
mycoides SC");
```

```
Vacuna vacuna8 = new Vacuna("Peste Bovina", "La vacunación puede  
ayudar a controlar su propagación en los rebaños", "Peste Bovina");
```

```
Vacuna vacuna9 = new Vacuna("Rabia", "La vacunación puede ayudar  
a prevenir la propagación de esta enfermedad zoonótica", "Rabia");
```

```
Vacuna vacuna10 = new Vacuna("Amigdalita de vaca", "Reduce el riesgo  
de infección por Mycobacterium bovis amigdalitis", "Mycobacterium bovis");
```

```
AltaVacuna(vacuna1);
```

```
AltaVacuna(vacuna2);
```

```
AltaVacuna(vacuna3);
```

```
AltaVacuna(vacuna4);
```

```
AltaVacuna(vacuna5);
```

```
AltaVacuna(vacuna6);
```

```
AltaVacuna(vacuna7);
```

```
AltaVacuna(vacuna8);
```

```
AltaVacuna(vacuna9);
```

```
AltaVacuna(vacuna10);
```

```
}
```

```
private void PrecargarVacunacion()
```

```
{
```

```
Vacunacion vacunacion1 = new Vacunacion(_listaGanado[0],  
_listaVacunas[2], new DateTime(2024, 02, 02)); //PRUEBA ERROR
```

```
Vacunacion vacunacion2 = new Vacunacion(_listaGanado[1],  
_listaVacunas[3], new DateTime(2024, 03, 17));
```

```
        Vacunacion  vacunacion3  =  new  Vacunacion(_listaGanado[2],
_listaVacunas[4], new DateTime(2024, 01, 02));

        Vacunacion  vacunacion4  =  new  Vacunacion(_listaGanado[3],
_listaVacunas[5], new DateTime(2023, 08, 18));

        Vacunacion  vacunacion5  =  new  Vacunacion(_listaGanado[4],
_listaVacunas[6], new DateTime(2022, 07, 02));

        Vacunacion  vacunacion6  =  new  Vacunacion(_listaGanado[5],
_listaVacunas[7], new DateTime(2021, 06, 19));

        Vacunacion  vacunacion7  =  new  Vacunacion(_listaGanado[6],
_listaVacunas[8], new DateTime(2020, 05, 02));

        Vacunacion  vacunacion8  =  new  Vacunacion(_listaGanado[7],
_listaVacunas[9], new DateTime(2024, 02, 11));

        Vacunacion  vacunacion9  =  new  Vacunacion(_listaGanado[8],
_listaVacunas[8], new DateTime(2020, 04, 10));

        Vacunacion  vacunacion10 =  new  Vacunacion(_listaGanado[9],
_listaVacunas[9], new DateTime(2022, 03, 02));

        Vacunacion  vacunacion11 =  new  Vacunacion(_listaGanado[10],
_listaVacunas[0], new DateTime(2023, 04, 02));

        Vacunacion  vacunacion12 =  new  Vacunacion(_listaGanado[11],
_listaVacunas[0], new DateTime(2023, 03, 17));

        Vacunacion  vacunacion13 =  new  Vacunacion(_listaGanado[12],
_listaVacunas[0], new DateTime(2023, 01, 02));

        Vacunacion  vacunacion14 =  new  Vacunacion(_listaGanado[13],
_listaVacunas[0], new DateTime(2023, 01, 02));

        Vacunacion  vacunacion15 =  new  Vacunacion(_listaGanado[14],
_listaVacunas[1], new DateTime(2024, 01, 27));

        Vacunacion  vacunacion16 =  new  Vacunacion(_listaGanado[15],
_listaVacunas[0], new DateTime(2022, 02, 02));

        Vacunacion  vacunacion17 =  new  Vacunacion(_listaGanado[16],
_listaVacunas[0], new DateTime(2022, 01, 15));
```

```
Vacunacion vacunacion18 = new Vacunacion(_listaGanado[17],
_listaVacunas[1], new DateTime(2022, 01, 02));

Vacunacion vacunacion19 = new Vacunacion(_listaGanado[18],
_listaVacunas[9], new DateTime(2023, 02, 02));

Vacunacion vacunacion20 = new Vacunacion(_listaGanado[19],
_listaVacunas[1], new DateTime(2023, 02, 21));

Vacunacion vacunacion21 = new Vacunacion(_listaGanado[30],
_listaVacunas[8], new DateTime(2022, 01, 20));

Vacunacion vacunacion22 = new Vacunacion(_listaGanado[31],
_listaVacunas[8], new DateTime(2023, 07, 02));

Vacunacion vacunacion23 = new Vacunacion(_listaGanado[32],
_listaVacunas[0], new DateTime(2023, 08, 02));

Vacunacion vacunacion24 = new Vacunacion(_listaGanado[33],
_listaVacunas[1], new DateTime(2023, 09, 02));

Vacunacion vacunacion25 = new Vacunacion(_listaGanado[34],
_listaVacunas[2], new DateTime(2023, 10, 02));

Vacunacion vacunacion26 = new Vacunacion(_listaGanado[35],
_listaVacunas[2], new DateTime(2024, 02, 02));

Vacunacion vacunacion27 = new Vacunacion(_listaGanado[36],
_listaVacunas[9], new DateTime(2024, 02, 02));

Vacunacion vacunacion28 = new Vacunacion(_listaGanado[37],
_listaVacunas[2], new DateTime(2022, 11, 27));

Vacunacion vacunacion29 = new Vacunacion(_listaGanado[38],
_listaVacunas[8], new DateTime(2022, 12, 26));

Vacunacion vacunacion30 = new Vacunacion(_listaGanado[39],
_listaVacunas[1], new DateTime(2023, 07, 25));

Vacunacion vacunacion31 = new Vacunacion(_listaGanado[40],
_listaVacunas[2], new DateTime(2023, 02, 02));

Vacunacion vacunacion32 = new Vacunacion(_listaGanado[41],
_listaVacunas[1], new DateTime(2024, 01, 02));
```

```
Vacunacion vacunacion33 = new Vacunacion(_listaGanado[42],
_listaVacunas[1], new DateTime(2023, 06, 22));

Vacunacion vacunacion34 = new Vacunacion(_listaGanado[43],
_listaVacunas[9], new DateTime(2024, 02, 17));

Vacunacion vacunacion35 = new Vacunacion(_listaGanado[44],
_listaVacunas[9], new DateTime(2023, 02, 15));

Vacunacion vacunacion36 = new Vacunacion(_listaGanado[45],
_listaVacunas[1], new DateTime(2024, 02, 19));

Vacunacion vacunacion37 = new Vacunacion(_listaGanado[46],
_listaVacunas[2], new DateTime(2023, 10, 18));

Vacunacion vacunacion38 = new Vacunacion(_listaGanado[47],
_listaVacunas[1], new DateTime(2022, 10, 02));

Vacunacion vacunacion39 = new Vacunacion(_listaGanado[48],
_listaVacunas[0], new DateTime(2023, 02, 02));

Vacunacion vacunacion40 = new Vacunacion(_listaGanado[49],
_listaVacunas[4], new DateTime(2024, 01, 20));

Vacunacion vacunacion41 = new Vacunacion(_listaGanado[50],
_listaVacunas[4], new DateTime(2023, 12, 21));

Vacunacion vacunacion42 = new Vacunacion(_listaGanado[51],
_listaVacunas[4], new DateTime(2024, 01, 16));

Vacunacion vacunacion43 = new Vacunacion(_listaGanado[52],
_listaVacunas[0], new DateTime(2024, 01, 15));

Vacunacion vacunacion44 = new Vacunacion(_listaGanado[53],
_listaVacunas[6], new DateTime(2023, 05, 18));

Vacunacion vacunacion45 = new Vacunacion(_listaGanado[54],
_listaVacunas[0], new DateTime(2024, 02, 17));

Vacunacion vacunacion46 = new Vacunacion(_listaGanado[55],
_listaVacunas[6], new DateTime(2024, 02, 16));

Vacunacion vacunacion47 = new Vacunacion(_listaGanado[56],
_listaVacunas[4], new DateTime(2023, 02, 15));
```

```
        Vacunacion  vacunacion48 = new  Vacunacion(_listaGanado[57],  
_listaVacunas[6], new DateTime(2023, 05, 14));
```

```
        Vacunacion  vacunacion49 = new  Vacunacion(_listaGanado[58],  
_listaVacunas[0], new DateTime(2023, 06, 13));
```

```
        Vacunacion  vacunacion50 = new  Vacunacion(_listaGanado[59],  
_listaVacunas[6], new DateTime(2023, 07, 12));
```

```
        AltaVacunacion(vacunacion1);
```

```
        AltaVacunacion(vacunacion2);
```

```
        AltaVacunacion(vacunacion3);
```

```
        AltaVacunacion(vacunacion4);
```

```
        AltaVacunacion(vacunacion5);
```

```
        AltaVacunacion(vacunacion6);
```

```
        AltaVacunacion(vacunacion7);
```

```
        AltaVacunacion(vacunacion8);
```

```
        AltaVacunacion(vacunacion9);
```

```
        AltaVacunacion(vacunacion10);
```

```
        AltaVacunacion(vacunacion11);
```

```
        AltaVacunacion(vacunacion12);
```

```
        AltaVacunacion(vacunacion13);
```

```
        AltaVacunacion(vacunacion14);
```

```
        AltaVacunacion(vacunacion15);
```

```
        AltaVacunacion(vacunacion16);
```

```
        AltaVacunacion(vacunacion17);
```

```
        AltaVacunacion(vacunacion18);
```

```
        AltaVacunacion(vacunacion19);
```

```
        AltaVacunacion(vacunacion20);
```

AltaVacunacion(vacunacion21);
AltaVacunacion(vacunacion22);
AltaVacunacion(vacunacion23);
AltaVacunacion(vacunacion24);
AltaVacunacion(vacunacion25);
AltaVacunacion(vacunacion26);
AltaVacunacion(vacunacion27);
AltaVacunacion(vacunacion28);
AltaVacunacion(vacunacion29);
AltaVacunacion(vacunacion30);
AltaVacunacion(vacunacion31);
AltaVacunacion(vacunacion32);
AltaVacunacion(vacunacion33);
AltaVacunacion(vacunacion34);
AltaVacunacion(vacunacion35);
AltaVacunacion(vacunacion36);
AltaVacunacion(vacunacion37);
AltaVacunacion(vacunacion38);
AltaVacunacion(vacunacion39);
AltaVacunacion(vacunacion40);
AltaVacunacion(vacunacion41);
AltaVacunacion(vacunacion42);
AltaVacunacion(vacunacion43);
AltaVacunacion(vacunacion44);
AltaVacunacion(vacunacion45);

```

        AltaVacunacion(vacunacion46);

        AltaVacunacion(vacunacion47);

        AltaVacunacion(vacunacion48);

        AltaVacunacion(vacunacion49);

        AltaVacunacion(vacunacion50);

    }

    private void PrecargarPotrero()
    {

        List<Ganado> _listaGanadoenPotrero1 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero2 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero3 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero4 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero5 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero6 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero7 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero8 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero9 = new List<Ganado>();

        List<Ganado> _listaGanadoenPotrero10 = new List<Ganado>();

        Potrero potrero1 = new Potrero("Potrero Aire Libre", int.MaxValue,
int.MaxValue, _listaGanadoenPotrero1);

        Potrero potrero2 = new Potrero("Potrero 1", 10, 20,
_listaNadaoenPotrero2);

        Potrero potrero3 = new Potrero("Potrero 2", 20, 40,
_listaNadaoenPotrero3);

        Potrero potrero4 = new Potrero("Potrero 3", 30, 60,
_listaNadaoenPotrero4);

```



```

        Potrero potrero5 = new Potrero("Potrero 4", 20, 40,
_listaGanadoenPotrero5);

        Potrero potrero6 = new Potrero("Potrero 5", 10, 20,
_listaGanadoenPotrero6);

        Potrero potrero7 = new Potrero("Potrero 6", 20, 40,
_listaGanadoenPotrero7);

        Potrero potrero8 = new Potrero("Potrero 7", 30, 60,
_listaGanadoenPotrero8);

        Potrero potrero9 = new Potrero("Potrero 8", 40, 80,
_listaGanadoenPotrero9);

        Potrero potrero10 = new Potrero("Potrero 9", 30, 60,
_listaGanadoenPotrero10);

        AltaPotrero(potrero1);

        AltaPotrero(potrero2);

        AltaPotrero(potrero3);

        AltaPotrero(potrero4);

        AltaPotrero(potrero5);

        AltaPotrero(potrero6);

        AltaPotrero(potrero7);

        AltaPotrero(potrero8);

        AltaPotrero(potrero9);

        AltaPotrero(potrero10);

    }

    private void PrecargaAgregarGanadoEnPotrero()

    {

        AgregarGanadoEnPotrero(_listaPotreros[0], _listaGanado[0]);

        AgregarGanadoEnPotrero(_listaPotreros[0], _listaGanado[1]);

```

```
AgregarGanadoEnPotrero(_listaPotreros[1], _listaGanado[2]);  
AgregarGanadoEnPotrero(_listaPotreros[1], _listaGanado[3]);  
AgregarGanadoEnPotrero(_listaPotreros[1], _listaGanado[4]);  
AgregarGanadoEnPotrero(_listaPotreros[1], _listaGanado[5]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[6]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[7]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[8]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[9]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[10]);  
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[11]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[12]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[13]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[14]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[15]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[16]);  
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[17]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[18]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[19]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[20]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[21]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[22]);  
AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[23]);  
AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[24]);  
AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[25]);  
AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[26]);
```

```
AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[27]);
AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[28]);
AgregarGanadoEnPotrero(_listaPotreros[6], _listaGanado[29]);
AgregarGanadoEnPotrero(_listaPotreros[6], _listaGanado[30]);
AgregarGanadoEnPotrero(_listaPotreros[6], _listaGanado[31]);
AgregarGanadoEnPotrero(_listaPotreros[6], _listaGanado[32]);
AgregarGanadoEnPotrero(_listaPotreros[6], _listaGanado[33]);
AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[34]);
AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[35]);
AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[36]);
AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[37]);
AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[38]);
AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[39]);
AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[40]);
AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[41]);
AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[42]);
AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[43]);
AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[44]);
AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[45]);
AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[46]);
AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[47]);
AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[48]);
AgregarGanadoEnPotrero(_listaPotreros[1], _listaGanado[49]);
AgregarGanadoEnPotrero(_listaPotreros[2], _listaGanado[50]);
AgregarGanadoEnPotrero(_listaPotreros[3], _listaGanado[51]);
```

```
        AgregarGanadoEnPotrero(_listaPotreros[4], _listaGanado[52]);  
        AgregarGanadoEnPotrero(_listaPotreros[5], _listaGanado[53]);  
        AgregarGanadoEnPotrero(_listaPotreros[0], _listaGanado[54]);  
        AgregarGanadoEnPotrero(_listaPotreros[0], _listaGanado[55]);  
        AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[56]);  
        AgregarGanadoEnPotrero(_listaPotreros[7], _listaGanado[57]);  
        AgregarGanadoEnPotrero(_listaPotreros[8], _listaGanado[58]);  
        AgregarGanadoEnPotrero(_listaPotreros[9], _listaGanado[59]);  
    }  
}  
}
```

Tarea

```
namespace Obligatorio.Dominio
{
    public class Tarea : IValido
    {
        public static int UltimoID { get; set; } = 1;
        public int ID { get; set; }
        public string Descripcion { get; set; }
        public DateTime FechaRealizacion { get; set; }
        public bool FueCompletada { get; set; }
        public DateTime FechaCierre { get; set; }
        public string Comentario { get; set; }
        public Tarea()
        {
            ID = UltimoID++;
            Descripcion = "";
            FechaRealizacion = DateTime.Today;
            FueCompletada = false;
            FechaCierre = DateTime.Today;
            Comentario = null;
        }
        public Tarea(string descripcion, DateTime fechaRealizacion, bool
fueCompletada, DateTime fechaCierre, string comentario)
        {
            ID = UltimoID++;
            Descripcion = descripcion;
            FechaRealizacion = fechaRealizacion;
            FueCompletada = fueCompletada;
            FechaCierre = fechaCierre;
            Comentario = comentario;
        }
        public string FueTerminada()
        {
            if (FueCompletada)
            {
                return "si";
            }
            else
            {
                return "No";
            }
        }
        public void Validar()
        {
            if (Descripcion == null || FechaRealizacion == DateTime.Today ||
FechaCierre == DateTime.Today || Comentario == null)
            {
                throw new Exception("Campos Obligatorios.");
            }
        }
        public override string ToString()
        {
            return $"Tarea Número: {ID} - " +
                $"Descripción de la Tarea: {Descripcion} - Fecha de
Realización: {FechaRealizacion}" +
                $" - La misma fue Compeltada?: {FueTerminada()} - Fecha de
Cierre: {FechaCierre} - " +
                $" "Devolución del Peón: {Comentario}.";
        }
    }
}
```

Vacuna

```
namespace Obligatorio.Dominio
{
    public class Vacuna : IValido
    {
        public string Nombre { get; set; }
        public string Descripcion { get; set; }
        public string AntiPatogeno { get; set; }
        public Vacuna()
        {
            Nombre = "";
            Descripcion = "";
            AntiPatogeno = "";
        }
        public Vacuna(string nombre, string descripcion, string
antiPatogeno)
        {
            Nombre = nombre;
            Descripcion = descripcion;
            AntiPatogeno = antiPatogeno;
        }
        public void Validar()
        {
            if (Nombre == null || Descripcion == null || AntiPatogeno ==
null)
            {
                throw new Exception("Todos los campos son obligatorios");
            }
        }
        public override string ToString()
        {
            return $"La vacuna: {Nombre}. La misma sirve para curar el
{AntiPatogeno}";
        }
    }
}
```

Vacunacion

```
namespace Obligatorio.Dominio
{
    public class Vacunacion : IValido
    {
        public Vacuna MyVacuna { get; set; }
        public DateTime VacunaFecha { get; set; }
        public Ganado MyGanado { get; set; }
        public Vacunacion()
        {
            MyVacuna = null;
            VacunaFecha = new DateTime(0000, 00, 00);

            MyGanado = null;
        }
        public Vacunacion(Ganado myGanado, Vacuna myVacuna, DateTime
vacunaFecha)
        {
            MyGanado = myGanado;
            MyVacuna = myVacuna;
            VacunaFecha = vacunaFecha;
        }
        public DateTime GetFechaVencimiento()
        {
            return VacunaFecha.AddYears(1);
        }
        public bool EsVacunable()
        {
            bool result = false;
            int año = DateTime.Now.Year;
            int mes = DateTime.Now.Month;
            if (año - MyGanado.FechaNacimiento.Year < 1)
            {
                if (mes - MyGanado.FechaNacimiento.Month < 3)
                {
                    result = false;
                }
                else
                {
                    result = true;
                }
            }
            else
            {
                result = true;
            }
            return result;
        }
        public void Validar()
        {
            if (MyGanado == null || MyVacuna == null)
            {
                throw new Exception("Todos los campos son obligatorios");
            }
        }
        public override string ToString()
        {
            return $"Vacunamos al ganado de número de caravana:
{MyGanado.NumCaravana}. Con la vacuna{MyVacuna.Nombre}. " +
                $"Siendo hoy: {VacunaFecha.Date}. La misma vence el:
{GetFechaVencimiento()}";
        }
    }
}
```

}
}
}

Program

```
namespace Obligatorio.MiApp
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Sistema _sistema = Sistema.Instance();
            _sistema.MostrarPotrerosEnConsola(10, 10);
            _sistema.MostrarGanadoEnConsola();
            _sistema.EstablecerCostoKgLanaOvinos();
            _sistema.LoginUsersConsola();
            _sistema.AltaBovinoConsola();
        }
    }
}
```

EmpleadoController

```
namespace WebApp.Controllers
{
    public class EmpleadoController : Controller
    {
        Sistema sistema = Sistema.Instance();

        public IActionResult IngresoEmpleados()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
null)
            {
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult IngresoEmpleados(string email, string
contraseña)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
null)
            {
                try
                {
                    Empleado empleadoLogueado =
sistema.GetEmpleadoXEmailYContraseña(email, contraseña);
                    HttpContext.Session.SetString("rolEmpleadoLogueado",
empleadoLogueado.GetType().Name);
                    HttpContext.Session.SetString("nombreEmpleadoLogueado",
empleadoLogueado.Nombre);
                    HttpContext.Session.SetInt32("idEmpleadoLogueado",
empleadoLogueado.Id);
                    if (empleadoLogueado is Peon)
                    {
                        return RedirectToAction("InicioPeon");
                    }
                    else
                    {
                        return RedirectToAction("InicioCapataz");
                    }
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                }
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult Salir()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon" || HttpContext.Session.GetString("rolEmpleadoLogueado") == "Capataz")
            {
                HttpContext.Session.Clear();
            }
            return RedirectToAction("Index", "Inicio");
        }
    }
}
```

```

        public IActionResult EditPeon(int id)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                Peon peonLogueado = (Peon)sistema.GetEmpleadoXId(id);
                return View(peonLogueado);
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult EditPeon(Peon pPeon)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                try
                {
                    sistema.ActualizarPeon(pPeon);
                    ViewBag.Mensaje = "Actualización con éxito";
                    return View(pPeon);
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                }
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult RegistroPeon()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
null)
            {
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult RegistroPeon(Peon pPeon)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
null)
            {
                try
                {
                    sistema.AltaEmpleado(pPeon);
                    ViewBag.Mensaje = "Registro realizado con éxito!";
                    return View();
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                    return View();
                }
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult InicioPeon()

```

```

        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                int? idEmpleadoLogueado =
HttpContext.Session.GetInt32("idEmpleadoLogueado");
                Peon peonLogueado =
(Peon)sistema.GetEmpleadoXId(idEmpleadoLogueado);
                ViewBag.NombreEmpleado = peonLogueado.Nombre;
                ViewBag.EmailEmpleado = peonLogueado.Email;
                ViewBag.ContraseñaEmpleado = peonLogueado.Contraseña;
                ViewBag.FechaIngresoEmpleado =
peonLogueado.FechaIngreso.ToShortDateString();
                ViewBag.EsResidente = peonLogueado.EsResidente;
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult InicioCapataz()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                int? idEmpleadoLogueado =
HttpContext.Session.GetInt32("idEmpleadoLogueado");
                Capataz capatazLogueado =
(Capataz)sistema.GetEmpleadoXId(idEmpleadoLogueado);
                ViewBag.NombreEmpleado = capatazLogueado.Nombre;
                ViewBag.EmailEmpleado = capatazLogueado.Email;
                ViewBag.ContraseñaEmpleado = capatazLogueado.Contraseña;
                ViewBag.FechaIngresoEmpleado =
capatazLogueado.FechaIngreso.ToShortDateString();
                ViewBag.PersonasACargo =
capatazLogueado.CantidadPersonasACargo;
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult EditCapataz(int id)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                Capataz capatazLogueado =
(Capataz)sistema.GetEmpleadoXId(id);
                return View(capatazLogueado);
            }
            return RedirectToAction("Index", "Inicio");
        }

        [HttpPost]
        public IActionResult EditCapataz(Capataz pCapataz)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                try
                {
                    sistema.ActualizarCapataz(pCapataz);
                    ViewBag.Mensaje = "Actualización con éxito";
                    return View(pCapataz);
                }
                catch (Exception ex)

```

```
        {
            ViewBag.Mensaje = ex.Message;
            return View();
        }
    }
    return RedirectToAction("Index", "Inicio");
}
public IActionResult ListaPeones()
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        return View(sistema.GetPeones());
    }
    return RedirectToAction("Index", "Inicio");
}
}
```

GanadoController

```
namespace WebApp.Controllers
{
    public class GanadoController : Controller
    {
        Sistema sistema = Sistema.Instance();
        public IActionResult Index()
        {
            return View();
        }
        public IActionResult RegistrarNuevoBovino()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                return View();
            }

            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult RegistrarNuevoBovino(Bovino pBovino)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                try
                {
                    sistema.AltaGanado(pBovino);
                    sistema.AsignarPotreroAAaltaGanado(pBovino);
                    ViewBag.Mensaje = "Animal Registrado con éxito!";
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                }
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        public IActionResult MostrarAnimal()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult MostrarAnimal(string tipoAnimal, int
pesoGanado)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                sistema.SetGananciaPorGanado();
                TempData["pesoGanado"] = pesoGanado;
                if (tipoAnimal == "Bovino")
                {
                    return RedirectToAction("MostrarBovinos");
                }
            }
        }
    }
}
```

```

        }
        else
        {
            return RedirectToAction("MostrarOvinos");
        }
    }
    return RedirectToAction("Index", "Inicio");
}

public IActionResult MostrarBovinos()
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        int? pesoGanado = TempData["pesoGanado"] as int?;
        return View(sistema.GetBovinosXPeso(pesoGanado));
    }
    return RedirectToAction("Index", "Inicio");
}

public IActionResult MostrarOvinos()
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        int? pesoGanado = TempData["pesoGanado"] as int?;
        return View(sistema.GetOvinosXPeso(pesoGanado));
    }
    return RedirectToAction("Index", "Inicio");
}
}
}
}

```

InicioController

```
namespace WebApp.Controllers
{
    public class InicioController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
    }
}
```


PotreroController

```
namespace WebApp.Controllers
{
    public class PotreroController : Controller
    {
        Sistema sistema = Sistema.Instance();
        public IActionResult Index()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                ViewBag.Potreros = sistema.GetPotreros();
                return View(sistema.GetGanadoAlAireLibre());
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult Index(int idPotrero, int numCaravana)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                Ganado pGanado = sistema.GetGanadoXNumCaravana(numCaravana);
                Potrero potreroAMover = sistema.GetPotreroXId(idPotrero);
                ViewBag.Potreros = sistema.GetPotreros();
                try
                {
                    sistema.MoverGanadoLibreAPotrero(potreroAMover,
pGanado);
                    ViewBag.Mensaje = "Animal movido con éxito";
                    return View(sistema.GetGanadoAlAireLibre());
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                    return View();
                }
            }
            return RedirectToAction("Index", "Inicio");
        }
        public IActionResult ListaPotreros()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
            {
                sistema.SetGananciaPorPotrero();
                List<Potrero> listPotreros = new List<Potrero>();
                listPotreros = sistema.GetPotreros();
                listPotreros.Sort();
                return View(listPotreros);
            }
            return RedirectToAction("Index", "Inicio");
        }
    }
}
```

TareasController

```
namespace WebApp.Controllers
{
    public class TareasController : Controller
    {
        Sistema sistema = Sistema.Instance();
        public IActionResult Index()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                int? idLogueado =
HttpContext.Session.GetInt32("idEmpleadoLogueado");
                Peon peonLogueado =
(Peon)sistema.GetEmpleadoXId(idLogueado);
                return
View(sistema.GetTareasNoRealizadasXPeon(peonLogueado));
            }
            return RedirectToAction("Index", "Inicio");
        }

        public IActionResult CompletarTarea()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                int? idLogueado =
HttpContext.Session.GetInt32("idEmpleadoLogueado");
                Peon peonLogueado =
(Peon)sistema.GetEmpleadoXId(idLogueado);
                ViewBag.AsignacionTareas =
sistema.GetTareasNoRealizadasXPeon(peonLogueado);
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult CompletarTarea(int idTarea, string pComentario)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                int? idLogueado =
HttpContext.Session.GetInt32("idEmpleadoLogueado");
                Peon peonLogueado =
(Peon)sistema.GetEmpleadoXId(idLogueado);
                AsignacionTarea tareaAcompletar =
sistema.GetAsignacionTareaPorPeon(idLogueado, idTarea);
                ViewBag.AsignacionTareas =
sistema.GetTareasNoRealizadasXPeon(peonLogueado);
                try
                {
                    sistema.CompletarAsignacionTarea(tareaAcompletar,
pComentario);
                    ViewBag.Mensaje = "Tarea Finalizada con éxito";
                    ViewBag.AsignacionTareas =
sistema.GetTareasNoRealizadasXPeon(peonLogueado);
                    return View();
                }
                catch (Exception ex)
                {

```

```

        ViewBag.Mensaje = ex.Message + " - Verifique Tarea
Seleccionada - ";
    }
    return View();
}
return RedirectToAction("Index", "Inicio");
}
public IActionResult ListaTareasPeon(int idPeon)
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        return View(sistema.GetAsignacionTareasXIdPeon(idPeon));
    }
    return RedirectToAction("Index", "Inicio");
}

public IActionResult AsignarTarea(int idPeon)
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        TempData["idPeon"] = idPeon;
        return View();
    }
    return RedirectToAction("Index", "Inicio");
}
[HttpPost]
public IActionResult AsignarTarea(Tarea pTarea)
{
    if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Capataz")
    {
        int? idPeon = TempData["idPeon"] as int?;
        try
        {
            sistema.AltaTarea(pTarea);
            Peon _peon = (Peon)sistema.GetEmpleadoXId(idPeon);
            AsignacionTarea _asignacionTarea = new
AsignacionTarea(_peon, pTarea);
            sistema.AltaAsignacionTarea(_asignacionTarea);
            ViewBag.Mensaje = "Tarea Agregada a peón con éxito!";
        }
        catch (Exception ex)
        {
            ViewBag.Mensaje = ex.Message;
        }
        return View();
    }
    return RedirectToAction("Index", "Inicio");
}
}
}
}

```

VacunacionController

```
namespace WebApp.Controllers
{
    public class VacunacionController : Controller
    {
        Sistema sistema = Sistema.Instance();

        public IActionResult Index()
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                ViewBag.Ganados = sistema.GetGanados();
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
        [HttpPost]
        public IActionResult Index(int numCaravana, string nombreVacuna)
        {
            if (HttpContext.Session.GetString("rolEmpleadoLogueado") ==
"Peon")
            {
                ViewBag.Ganados = sistema.GetGanados();
                Ganado ganadoPorVacunar =
sistema.GetGanadoXNumCaravana(numCaravana);
                if (ganadoPorVacunar == null)
                {
                    ViewBag.Mensaje = "Número de caravana incorrecto";
                }
                Vacuna vacunaPorUsar =
sistema.GetVacunaXNombreVacuna(nombreVacuna);
                Vacunacion vacunacionPorRealizar = new
Vacunacion(ganadoPorVacunar, vacunaPorUsar, DateTime.Today);
                try
                {
                    sistema.AltaVacunacion(vacunacionPorRealizar);
                    ViewBag.Mensaje = "Vacunación registrada con ÉXITO!";
                    return View();
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = ex.Message;
                }
                return View();
            }
            return RedirectToAction("Index", "Inicio");
        }
    }
}
```

View EditCapataz

@model Obligatorio.Dominio.Capataz

<h1>Editar Datos Personales</h1>

<form method="post">

<table>

<tbody>

<tr>

<td><label>Email:</label></td>

<td><input asp-for="Email" /></td>

</tr>

<tr>

<td><label>Contraseña:</label></td>

<td><input asp-for="Contraseña" /></td>

</tr>

<tr>

<td><label>Nombre:</label></td>

<td><input asp-for="Nombre" /></td>

</tr>

<tr>

<td><label>Fecha de Ingreso:</label></td>

<td><input asp-for="FechaIngreso" /></td>

</tr>

<tr>

<td><label>Cantidad de Personas a Su Cargo</label></td>

<td>

<input asp-for="CantidadPersonasACargo" checked />

</td>

</tr>

<tr>

<td><input type="submit" value="Editar" /></td>

</tr>

</tbody>

</table>

@ViewBag.Mensaje

<hr />

</form>

View EditPeon

```
@model Obligatorio.Dominio.Peon
<h1>Editar Datos Personales</h1>
<br />
<br />
<form method="post">
    <table>
        <thead>
        </thead>
        <tbody>
            <tr>
                <td><label>Email:</label></td>
                <td><input asp-for="Email" /></td>
            </tr>
            <tr>
                <td><label>Contraseña:</label></td>
                <td><input asp-for="Contraseña"/></td>
            </tr>
            <tr>
                <td><label>Nombre:</label></td>
                <td><input asp-for="Nombre" /></td>
            </tr>
            <tr>
                <td><label>Fecha de Ingreso:</label></td>
                <td><input asp-for="FechaIngreso" /></td>
            </tr>
            <tr>
                <td><label>Es Usted Residente?</label></td>
                <td>
                    <input asp-for="EsResidente" checked />
                </td>
            </tr>
            <tr>
                <td><input type="submit" value="Editar" /></td>
            </tr>
        </tbody>
    </table>
    <br />
    @ViewBag.Mensaje
    <hr />
</form>
```

View IngresoEmpleados

@model Obligatorio.Dominio.Empleado

<h1>INGRESO EMPLEADOS</h1>

<form method="post">

<table>

<thead>

</thead>

<tbody>

<tr>

<td><label>Email:</label></td>

<td><input type="text" name="email" /></td>

</tr>

<tr>

<td><label>Contraseña:</label></td>

<td><input type="password" name="contraseña" /></td>

</tr>

<tr>

<td><input type="submit" value="Ingresar" /></td>

</tr>

</tbody>

</table>

</form>

@ViewBag.Mensaje

View InicioCapataz

```
@model Obligatorio.Dominio.Capataz
<h1>BIENVENIDO @ViewBag.NombreEmpleado</h1>
<hr />
<style>
    body {
        font-family: Arial, sans-serif;
    }

    h1 {
        text-align: center;
    }

    table {
        width: 80%;
        margin: 0 auto;
        border-collapse: collapse;
        margin-top: 20px;
    }

    th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
        text-align: center;
    }

    tr:hover {
        background-color: #f5f5f5;
    }
</style>
<table>
    <thead>
        <tr>
            <th>EMAIL</th>
            <th>CONTRASEÑA</th>
            <th>NOMBRE</th>
            <th>FECHA DE INGRESO</th>
            <th>PERSONAL A CARGO</th>
            <th>EDITAR DATOS</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>@ViewBag.EmailEmpleado</td>
            <td>@ViewBag.ContraseñaEmpleado</td>
            <td>@ViewBag.NombreEmpleado</td>
            <td>@ViewBag.FechaIngresoEmpleado</td>
            <td>@ViewBag.PersonasACargo</td>
            <td>@Html.ActionLink("Editar", "EditCapataz", "Empleado", new {
id = Context.Session.GetInt32("idEmpleadoLogueado") })</td>
        </tr>
    </tbody>
</table>
```


View InicioPeon

```
@model Obligatorio.Dominio.Peon
<h1>BIENVENIDO @ViewBag.NombreEmpleado</h1>
<hr />
<style>
    body {
        font-family: Arial, sans-serif;
    }

    h1 {
        text-align: center;
    }

    table {
        width: 80%;
        margin: 0 auto;
        border-collapse: collapse;
        margin-top: 20px;
    }

    th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
        text-align: center;
    }

    tr:hover {
        background-color: #f5f5f5;
    }
</style>
<table>
    <thead>
        <tr>
            <th>EMAIL</th>
            <th>CONTRASEÑA</th>
            <th>NOMBRE</th>
            <th>FECHA DE INGRESO</th>
            <th>ES RESIDENTE</th>
            <th>EDITAR DATOS</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>@ViewBag.EmailEmpleado</td>
            <td>@ViewBag.ContraseñaEmpleado</td>
            <td>@ViewBag.NombreEmpleado</td>
            <td>@ViewBag.FechaIngresoEmpleado</td>
            <td>@ViewBag.EsResidente</td>
            <td>@Html.ActionLink("Editar", "EditPeon", "Empleado", new { id
= Context.Session.GetInt32("idEmpleadoLogueado") })</td>
        </tr>
    </tbody>
</table>
<p>@ViewBag.IdEmpleado</p>
```

View ListaPeones

```
@model List<Obligatorio.Dominio.Peon>
<h1>Lista Peones</h1>
<style>
    table {
        width: 100%;
        border-collapse: collapse;
        text-align: center;
    }

    th, td {
        border: 1px solid #000;
        padding: 8px;
    }

    th {
        background-color: #f2f2f2;
    }
</style>
<table>
    <thead>
        <tr>
            <th>Id</th>
            <th>Email</th>
            <th>Contraseña</th>
            <th>Nombre</th>
            <th>Fecha de Ingreso</th>
            <th>EsResidente</th>
            <th>Tareas</th>
            <th>Agregar Tarea</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@item.Id</td>
                <td>@item.Email</td>
                <td>*****</td>
                <td>@item.Nombre</td>
                <td>@item.FechaIngreso.ToShortDateString()</td>
                <td>@item.EsResidente</td>
                <td>@Html.ActionLink("Ver", "ListaTareasPeon", "Tareas", new
{ idPeon = item.Id })</td>
                <td>@Html.ActionLink("Agregar", "AsignarTarea", "Tareas",
new { idPeon = item.Id })</td>
            </tr>
        }
    </tbody>
</table>
```

View RegistroPeon

@model Obligatorio.Dominio.Peon

```
<h1>REGISTRO EMPLEADOS</h1>
<br />
<br />
<form method="post">
  <table>
    <thead>
    </thead>
    <tbody>
      <tr>
        <td><label>Email:</label></td>
        <td><input asp-for="Email" /></td>
      </tr>
      <tr>
        <td><label>Contraseña:</label></td>
        <td><input asp-for="Contraseña" type="password"/></td>
      </tr>
      <tr>
        <td><label>Nombre:</label></td>
        <td><input asp-for="Nombre" /></td>
      </tr>
      <tr>
        <td><label>Fecha de Ingreso:</label></td>
        <td><input asp-for="FechaIngreso" /></td>
      </tr>
      <tr>
        <td><label>Es Usted Residente?</label></td>
        <td>
          <input asp-for="EsResidente" checked />
        </td>
      </tr>
      <tr>
        <td><input type="submit" value="Registrarse" /></td>
      </tr>
    </tbody>
  </table>
  <hr />
</form>
@ViewBag.Mensaje
```

View MostrarAnimal

@model Obligatorio.Dominio.Ganado

<h1>Mostrar Animal</h1>

<form method="post">

<label name="pesoGanado">Ingrese mínimo peso a Mostrar:</label>

<input type="number" name="pesoGanado" />

<label>Tipo de Animal:</label>

<select name="tipoAnimal">

<option value="Bovino">Bovino</option>

<option value="Ovino">Ovino</option>

</select>

<input type="submit" value="Mostrar" />

<hr />

</form>

View MostrarBovinos

```
@model List<Obligatorio.Dominio.Bovino>
<h1>Lista Bovinos</h1>
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 20px;
    }

    table {
        width: 100%;
        border-collapse: collapse;
    }

    th, td {
        padding: 8px 12px;
        border: 1px solid #ddd;
        text-align: center;
    }

    th {
        background-color: #f4f4f4;
    }

    tr:nth-child(even) {
        background-color: #f9f9f9;
    }
</style>
<table>
    <thead>
        <tr>
            <th>Numero de Caravana</th>
            <th>Es Macho?</th>
            <th>Raza</th>
            <th>Fecha de Nacimiento</th>
            <th>Costo de Adquisición</th>
            <th>Costo de Alimentación</th>
            <th>Peso Actual</th>
            <th>Es Hibrido?</th>
            <th>Se Alimenta Por Grano?</th>
            <th>Potencial Ganancia de Venta</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@item.NumCaravana</td>
                <td>@item.EsMacho</td>
                <td>@item.Raza</td>
                <td>@item.FechaNacimiento.ToShortDateString()</td>
                <td>@item.CostoAdquisicion</td>
                <td>@item.CostoAlimentacion</td>
                <td>@item.PesoActual</td>
                <td>@item.EsHibrido</td>
                <td>@item.SeAlimentaPorGrano</td>
                <td>@item.GananciaVentaIndividual.ToString("0.0")</td>
            </tr>
        }
    </tbody>
</table>
```

View MostrarOvinos

@model List<Obligatorio.Dominio.Ovino>

<h1>Lista Ovinos</h1>

<style>

```
body {
    font-family: Arial, sans-serif;
    margin: 20px;
}
```

```
table {
    width: 100%;
    border-collapse: collapse;
}
```

```
th, td {
    padding: 8px 12px;
    border: 1px solid #ddd;
    text-align: center;
}
```

```
th {
    background-color: #f4f4f4;
}
```

```
tr:nth-child(even) {
    background-color: #f9f9f9;
}
```

</style>

<table>

<thead>

<tr>

```
<th>Numero de Caravana</th>
<th>Es Macho?</th>
<th>Raza</th>
<th>Fecha de Nacimiento</th>
<th>Costo de Adquisición</th>
<th>Costo de Alimentación</th>
<th>Peso Actual</th>
<th>Es Híbrido?</th>
<th>Potencial Ganancia de Venta</th>
```

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

```
<td>@item.NumCaravana</td>
<td>@item.EsMacho</td>
<td>@item.Raza</td>
<td>@item.FechaNacimiento.ToShortDateString()</td>
<td>@item.CostoAdquisicion</td>
<td>@item.CostoAlimentacion</td>
<td>@item.PesoActual</td>
<td>@item.EsHibrido</td>
<td>@item.GananciaVentaIndividual.ToString("0.0")</td>
```

</tr>

}

</tbody>

</table>

View RegistrarNuevoBovino

@model Obligatorio.Dominio.Bovino

```
<h1>Registro de nuevo bovino</h1>
<br />
<form method="post">
  <label asp-for="NumCaravana">Número de Caravana:</label>
  <input asp-for="NumCaravana" />
  <br />
  <br />
  <label asp-for="Raza">Nombre de Raza:</label>
  <select name="Raza">
    <option value="Angus">Angus</option>
    <option value="Hereford">Hereford</option>
    <option value="Holstein">Holstein</option>
    <option value="Jersey">Jersey</option>
    <option value="Brahman">Brahman</option>
    <option value="Simmental">Simmental</option>
    <option value="Charolais">Charolais</option>
    <option value="Limousin">Limousin</option>
    <option value="Shorthorn">Shorthorn</option>
    <option value="Gelbvieh">Gelbvieh</option>
  </select>
  <br />
  <br /><label asp-for="FechaNacimiento">Fecha de Nacimiento:</label>
  <input asp-for="FechaNacimiento" />
  <br />
  <br /><label asp-for="CostoAdquisicion">Costo de Adquisición:</label>
  <input asp-for="CostoAdquisicion" />
  <br />
  <br /><label asp-for="CostoAlimentacion">Costo de Alimentación:</label>
  <input asp-for="CostoAlimentacion" />
  <br />
  <br /><label asp-for="PesoActual">Peso Actual:</label>
  <input asp-for="PesoActual" />
  <br />
  <br /><label asp-for="EsHibrido">Es Híbrido?</label>
  <input asp-for="EsHibrido" />
  <br />
  <br /><label asp-for="EsMacho">Es Macho?</label>
  <input asp-for="EsMacho" />
  <br />
  <br />
  <label asp-for="SeAlimentaPorGrano">Se Alimenta Por
Grano/Pastura?</label>
  <input asp-for="SeAlimentaPorGrano" />
  <p>(Check para Grano)</p>
  <input type="submit" value="Registrar" />
  <hr />
</form>
@ViewBag.Mensaje
```

View Index (Inicio)

<h1>BIENVENIDOS A NUESTRA GRANJA</h1>

View Index(Potrero)

@model List<Obligatorio.Dominio.Ganado>

```
<h1>Asignar un Animal Libre a un Potrero</h1>
<hr />
<br />
<form method="post">
    <table>
        <thead>
        </thead>
        <tbody>
            <tr>
                <td><label>Número de Caravana - Raza </label></td>
                <td>
                    <select name="numCaravana">
                        @if (Model != null)
                        {
                            @foreach (var item in Model)
                            {
                                <option
value="@item.NumCaravana">@item.NumCaravana - @item.Raza</option>
                            }
                        }
                    </select>
                </td>
            </tr>
            <tr>
                <td><label>Potrero a Mover: </label></td>
                <td>
                    <select name="idPotrero">
                        @foreach (var item in @ViewBag.Potreros)
                        {
                            <option
value="@item.ID">@item.Descripcion</option>
                        }
                    </select>
                </td>
            </tr>
            <tr>
                <td><input type="submit" name="Mover" value="Mover" /></td>
            </tr>
        </tbody>
    </table>
    <hr />
</form>
@ViewBag.Mensaje
```

View ListaPotreros

```
@model List<Obligatorio.Dominio.Potrero>
<h1>Potreros</h1>
<style>
    table {
        border-collapse: collapse;
        width: 75%;
    }

    th, td {
        border: 1px solid #dddddd;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
    }

    tr:nth-child(even) {
        background-color: #ffffff;
    }
</style>
<table border="1">
    <thead>
        <tr>
            <th>Id</th>
            <th>Descripcion</th>
            <th>Hectareas</th>
            <th>Capacidad Máxima de Animales</th>
            <th>Cantidad de Animales Actualmente</th>
            <th>Potencial Precio de Venta</th>
        </tr>
    </thead>
    <tbody>
        @foreach(var item in Model)
        {
            <tr>
                <td>@item.ID</td>
                <td>@item.Descripcion</td>
                <td>@item.Hectareas</td>
                <td>@item.CapacidadMaxAnimales</td>
                <td>@item._ListaGanadosEnPotrero.Count</td>
                <td>@item.PotencialPrecioVenta.ToString("0.0")</td>
            </tr>
        }
    </tbody>
</table>
```

View Layout

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"]Nuestra Granja</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css"
/>
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true"
/>
    <link rel="stylesheet" href="~/WebApp.styles.css" asp-append-
version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-
light bg-white border-bottom box-shadow mb-3">
            <div class="container-fluid">
                <a class="navbar-brand" asp-area="" asp-controller="" asp-
action="">Nuestra Granja</a>
                <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle
navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex
justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        @if
(Context.Session.GetString("rolEmpleadoLogueado") == null)
                        {
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="IngresoEmpleados">Inicio de Sesión</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="RegistroPeon">Registro</a>
                            </li>
                        }
                        @if
(Context.Session.GetString("rolEmpleadoLogueado") == "Capataz")
                        {
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="InicioCapataz">Inicio</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Ganado" asp-action="RegistrarNuevoBovino">Registrar
Bovino</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Potrero" asp-action="Index">Animal Libre a Potrero</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="IngresoEmpleados">Inicio de Sesión</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="RegistroPeon">Registro</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="InicioCapataz">Inicio</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Ganado" asp-action="RegistrarNuevoBovino">Registrar
Bovino</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area=""
asp-controller="Potrero" asp-action="Index">Animal Libre a Potrero</a>
                            </li>
                        }
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <div class="row">
            <div class="col">
                <div class="card">
                    <div class="card-header">
                        <h3>Inicio de Sesión</h3>
                    </div>
                    <div class="card-body">
                        <div class="form-group">
                            <input type="text" value="Usuario" />
                        </div>
                        <div class="form-group">
                            <input type="password" value="Contraseña" />
                        </div>
                        <div class="form-group">
                            <input type="button" value="Iniciar Sesión" />
                        </div>
                    </div>
                </div>
            </div>
            <div class="col">
                <div class="card">
                    <div class="card-header">
                        <h3>Registro</h3>
                    </div>
                    <div class="card-body">
                        <div class="form-group">
                            <input type="text" value="Nombre" />
                        </div>
                        <div class="form-group">
                            <input type="text" value="Apellido" />
                        </div>
                        <div class="form-group">
                            <input type="text" value="Cédula" />
                        </div>
                        <div class="form-group">
                            <input type="text" value="Cargo" />
                        </div>
                        <div class="form-group">
                            <input type="button" value="Registrar" />
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

```

        <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="ListaPeones">Lista Peones</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area=""
asp-controller="Potrero" asp-action="ListaPotreros">Lista Potreros</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area=""
asp-controller="Ganado" asp-action="MostrarAnimal">Mostrar Animal</a>
    </li>
}
@if
(Context.Session.GetString("rolEmpleadoLogueado") == "Peon")
{
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="InicioPeon">Inicio</a>
    </li>
    <li style="text-align:center">
        <a class="nav-link text-dark" asp-area=""
asp-controller="Vacunacion" asp-action="Index">Vacunación</a>
    </li>
    <li>
        <a class="nav-link text-dark" asp-area=""
asp-controller="Tareas" asp-action="Index">Tareas</a>
    </li>
}
@if
(Context.Session.GetString("rolEmpleadoLogueado") != null)
{
    <li class="nav-item ms-auto">
        <a class="nav-link text-dark" asp-area=""
asp-controller="Empleado" asp-action="Salir">Salir</a>
    </li>
}

</ul>
</div>
</div>
</nav>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>
<footer class="border-top footer text-muted">
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

View ValidationScriptsPartial

```
<script src="/lib/jquery-validation/dist/jquery.validate.min.js"></script>  
<script src="/lib/jquery-validation-  
unobtrusive/jquery.validate.unobtrusive.min.js"></script>
```

View Error

```
@model ErrorViewModel
@{
    ViewData["Title"] = "Error";
}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while processing your
request.</h2>

@if (Model.ShowRequestId)
{
    <p>
        <strong>Request ID:</strong> <code>@Model.RequestId</code>
    </p>
}

<h3>Development Mode</h3>
<p>
    Swapping to <strong>Development</strong> environment will display more
    detailed information about the error that occurred.
</p>
<p>
    <strong>The Development environment shouldn't be enabled for deployed
    applications.</strong>
    It can result in displaying sensitive information from exceptions to end
    users.
    For local debugging, enable the <strong>Development</strong> environment
    by setting the <strong>ASPNETCORE_ENVIRONMENT</strong> environment variable
    to <strong>Development</strong>
    and restarting the app.
</p>
```

View AsignarTarea

```
@model Obligatorio.Dominio.Tarea
<h1>Asignar nueva tarea a peón</h1>
<br />
<form method="post">
    <label asp-for="Descripcion">Ingrese Descripción:</label>
    <input asp-for="Descripcion" />
    <br />
    <br />
    <label asp-for="FechaRealizacion">Ingrese Fecha para
Realización:</label>
    <input asp-for="FechaRealizacion" />
    <br />
    <br />
    <label asp-for="FueCompletada">Fue Completada?</label>
    <input asp-for="FueCompletada" />
    <br />
    <br />
    <label asp-for="FechaCierre">Ingrese Fecha Estimada para
Cierre</label>
    <input asp-for="FechaCierre" />
    <br />
    <br />
    <label asp-for="Comentario">Ingrese Comentario</label>
    <input asp-for="Comentario" />
    <br />
    <br />
    <input type="submit" value="Asignar" />
    <hr />
</form>
@ViewBag.Mensaje
```

View CompletarTarea

```
@model Obligatorio.Dominio.AsignacionTarea
<h1>Completar Tarea</h1>
<br />
<br />
<form method="post">
    <table>
        <thead>
        </thead>
        <tbody>
            <tr>
                <td><label>Ingrese ID de Tarea:</label></td>
                <td>
                    <select name="idTarea">
                        @foreach (var item in ViewBag.AsignacionTareas)
                        {
                            <option value="@item.Tarea.ID">@item.Tarea.ID -
@item.Tarea.Descripcion</option>
                        }
                    </select>
                </td>
            </tr>
            <tr>
                <td><label>Ingrese Comentario:</label></td>
                <td><input asp-for="Tarea.Comentario" name="pComentario"
/></td>
            </tr>
        </tbody>
    </table>
    <br />
    <input type="submit" value="Completar" />
    @ViewBag.Mensaje
    <hr />
</form>
```


View Index(Tarea)

```
@model IEnumerable<Obligatorio.Dominio.AsignacionTarea>
<h1>TAREAS PENDIENTES</h1>
<style>
    table {
        text-align: center;
    }
</style>
<table class="centered-table">
    <thead>
        <tr>
            <th>ID TAREA</th>
            <th>DESCRIPCIÓN</th>
            <th>FECHA DE REALIZACIÓN</th>
            <th>FUE COMPLETADA</th>
            <th>FECHA CIERRE</th>
            <th>COMENTARIOS</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@item.Tarea.ID</td>
                <td>@item.Tarea.Descripcion</td>

                <td>@item.Tarea.FechaRealizacion.ToString("dd/MM/yyyy")</td>
                <td>@item.Tarea.FueCompletada</td>

                <td>@item.Tarea.FechaCierre.ToShortDateString()</td>
                <td>@item.Tarea.Comentario</td>
            </tr>
        }
    </tbody>
</table>
<a href="Tareas/CompletarTarea">Completar Tarea Pendiente</a>
```

View ListaTareasPeon

```
@model List<Obligatorio.Dominio.AsignacionTarea>
<h1>Tareas por Peon</h1>
<style>
    table {
        width: 100%;
        border-collapse: collapse;
        text-align: center;
    }

    th, td {
        border: 1px solid #000;
        padding: 8px;
    }

    th {
        background-color: #f2f2f2;
    }
</style>
<table>
    <thead>
        <tr>
            <th>Descripcion</th>
            <th>Fecha De Realizacion</th>
            <th>FueCompletada</th>
            <th>Fecha de Cierre</th>
            <th>Comentario</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@item.Tarea.Descripcion</td>
                <td>@item.Tarea.FechaRealizacion.ToShortDateString()</td>
                <td>@item.Tarea.FueCompletada</td>
                <td>@item.Tarea.FechaCierre.ToShortDateString()</td>
                <td>@item.Tarea.Comentario</td>
            </tr>
        }
    </tbody>
</table>
```

View Index(Vacunacion)

@model Obligatorio.Dominio.Vacunacion

<h1>REGISTRO DE VACUNACIÓN</h1>

<hr />

<form method="post">

<table>

<thead>

</thead>

<tbody>

<tr>

<td><label>Número de Caravana: </label></td>

<td>

<select name="numCaravana">

@foreach (var item in ViewBag.Ganados)
{

<option

value="@item.NumCaravana">@item.NumCaravana</option>

}

</select>

</td>

</tr>

<tr>

<td><label>Nombre de Vacuna: </label></td>

<td>

<select name="nombreVacuna">

<option value="Fiebre aftosa">Fiebre aftosa</option>

<option value="Brucelosis">Brucelosis</option>

<option value="Tuberculosis bovina">Tuberculosis

bovina</option>

<option value="Leptospirosis">Leptospirosis</option>

<option value="Fiebre del Valle del Rift">Fiebre del

Valle del Rift</option>

<option value="Clostridiosis">Clostridiosis</option>

<option value="Enfermedad de la mucosa">Enfermedad

de la mucosa</option>

<option value="Peste Bovina">Peste Bovina</option>

<option value="Rabia">Rabia</option>

<option value="Amigdalita de vaca">Amigdalita de

vaca</option>

</select>

</td>

</tr>

<tr>

<td><input type="submit" name="Vacunar" value="Vacunar"

</tr>

</tbody>

</table>

<hr />

</form>

@ViewBag.Mensaje

View ViewImports

```
@using WebApp
@using WebApp.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

View ViewStart

```
@{
    Layout = "_Layout";
}
```

View appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

View Program

```
namespace WebApp
{
    public class Program
    {
        public static void Main(string[] args)
        {
            var builder = WebApplication.CreateBuilder(args);

            // Add services to the container.
            builder.Services.AddControllersWithViews();
            builder.Services.AddSession();

            var app = builder.Build();

            // Configure the HTTP request pipeline.
            if (!app.Environment.IsDevelopment())
            {
                app.UseExceptionHandler("/Home/Error");
                // The default HSTS value is 30 days. You may want to change
                this for production scenarios, see https://aka.ms/aspnetcore-hsts.
                app.UseHsts();
            }

            app.UseHttpsRedirection();
            app.UseStaticFiles();

            app.UseRouting();
            app.UseSession();

            app.UseAuthorization();

            app.MapControllerRoute(
                name: "default",
                pattern: "{controller=Inicio}/{action=Index}/{id?}");

            app.Run();
        }
    }
}
```