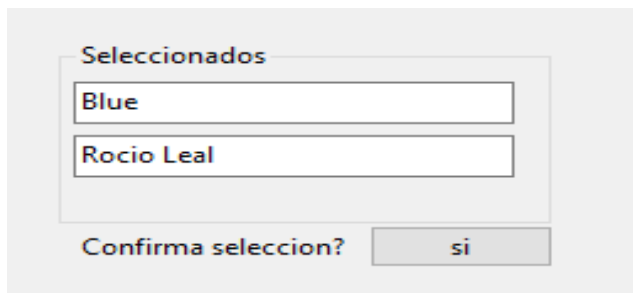


TP3 Laboratorio

El objetivo del app es la gestión de adopción de perros que se encuentran en distintos refugios.

Se le pide al empleado que seleccione un adoptante y un perro. Si el perro está disponible para la adopción aparecerá en los campos



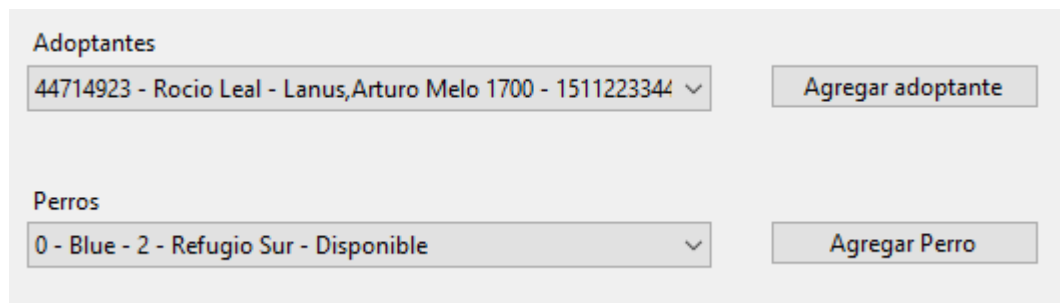
Seleccionados

Blue

Rocio Leal

Confirma seleccion?

En este sector del formulario aparecen las dos listas de-serializadas y cargadas respectivamente en sus comboBoxes.



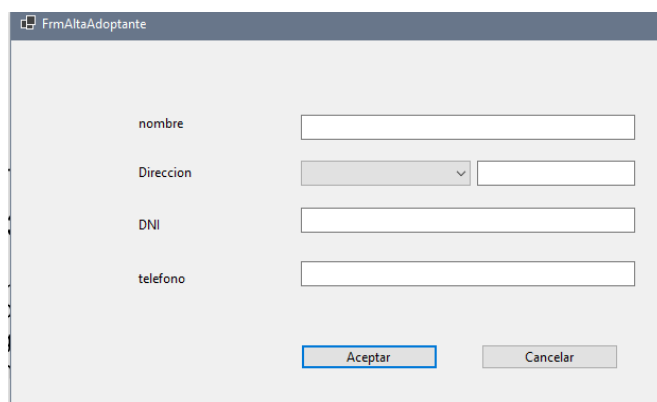
Adoptantes

44714923 - Rocio Leal - Lanus, Arturo Melo 1700 - 1511223344 ▼

Perros

0 - Blue - 2 - Refugio Sur - Disponible ▼

Si se desea agregar a la lista de adoptantes al hacer click se muestra un nuevo formulario , en formato ShowDialog() , se captura los datos Pero no se lo agregara a la lista si no que se lo guarda en forma de .txt en la ruta del proyecto,ensamblado.



FrmAltaAdoptante


nombre

Direccion

DNI

telefono

Si se desea agregar un perro aparece otro formulario con los campos a rellenar , una vez completados si se acepta se agregara a la lista de perros.



FrmAltaPerro

Nombre

Edad aproximada

Refugio

Si se tiene al adoptante cargado en la lista , puede seleccionar de la lista perro y en el caso de no estar disponible no aparecerá en el cuadro de la selección. Una vez seleccionados aparecerán en este sector

Si se confirma la lista se actualiza y el perro tendrá por estado Adoptado. Y finalmente si se hace clic en Guardar se serializa formato json la lista de los perros como la de los adoptantes.

Temas requeridos:

Excepciones: crea clase “CargaDeDatosExceptions.cs” principalmente para el error en la carga de datos del formulario.

```

1 // Referencias
2 public void ValidarCampos()
3 {
4     if((textBox_nombre.Text==string.Empty) || (textBox_direccion.Text==string.Empty) || textBox_dni.Text== string.Empty || te
5     {
6         throw new CargaDeDatosExceptions("completar todos los datos requeridos");
7     }
8     if(!int.TryParse(this.textBox_dni.Text, out int dni) || (int.Parse(this.textBox_dni.Text) <= 0 || int.Parse(this.textBox_
9     {
10        throw new CargaDeDatosExceptions("Error cargar de dni");
11    }
12    if(!int.TryParse(this.textBox_telefono.Text,out int telefono))
13    {
14        throw new CargaDeDatosExceptions("Error en carga de telefono");
15    }
16 }

```

Prueba Unitaria de la clase Serializadora

Explorador de pruebas				Buscar en el explorador de pruebas (Ctrl + B)	
Prueba	Duración	Rasgos	Mensaje de error	Resumen de los detalles de la prueba	
Test (7)	120 ms			DeSerializar_json_CuandoLePasoRutaPerro_Deberia	
Test (7)	120 ms			Origen: ClaseSerializadoraTest.cs línea 4	
ClaseSerializadoraTest (7)	120 ms			Duración: 11 ms	
Deserializadora_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	5 ms				
DeSerializar_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	54 ms				
DeSerializar_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	11 ms				
PropiedadGetRutaJson_deberiaRetornarJson	< 1 ms				
Serializadora_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	2 ms				
Serializadora_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	38 ms				
Serializadora_json_CuandoLePasoRutaPerro_DeberiaRetornarJson	10 ms				

Tipos Genericos : la clase Serializadora es genérica debido a que serializo tanto Adoptantes como Perros .La Interfaz IObtenerDatos es genérica ya que la uso para obtener los distintos tipos de datos que me retornar los formularios,

```

1 referencia
public FrmInicio()

InitializeComponent();
listaPerros = ClaseSerializadora<List<Perro>>.DeSerializar_json("listaPerros.json", EDirectorio.DirectorioBaseProyecto);
listaAdoptantes = ClaseSerializadora<List<Adoptante>>.DeSerializar_json("listaAdoptantes.json", EDirectorio.DirectorioBaseProyecto);
seleccionDelPerro = null;
seleccionAdoptante = null;

```

Interfaces : Principalmente las uso en el formulario para validar Campos y Obtener Datos de un formulario a otro y actualizar datos(cargo comboboxes).

```

public partial class FrmAltaAdoptante : Form, IActualizarDatos, IValidarCampos, IObtenerDatos<Adoptante>
{
    Adoptante adoptanteAux = null;
    1 referencia
    public FrmAltaAdoptante()
    {
        InitializeComponent();
        ActualizarDatos();
    }
}

```

Archivos la uso para guardar el adoptante en formato .txt. ya que no lo agrego a la lista.

```

private void button_agregarAdoptante_Click(object sender, EventArgs e)
{
    FrmAltaAdoptante frmAltaAdoptante = new FrmAltaAdoptante();
    frmAltaAdoptante.ShowDialog();
    if (frmAltaAdoptante.DialogResult == DialogResult.OK)
    {
        Adoptante aux = frmAltaAdoptante.ObtenerDatos();
        string ruta = Archivo.Guardar($"Adoptante_{DateTime.Now.ToString("HH-mm-ss")}.txt", aux.ToString(), EDirectorio.DirectorioBaseProyecto);
        MessageBox.Show($"Se a guardado dato de posible Adoptante en {ruta}");
    }
}

```

Rutas del Proyecto :

Se encuentra agregado a proyecto dos archivos.json con la lista de perros y adoptantes.Están en la biblioteca de Clase Entidades con la propiedad copiar siempre.

La primera de-serialización se realiza desde el ensamblado con los archivos.json:

listaPerros.json

listaAdoptantes.json

