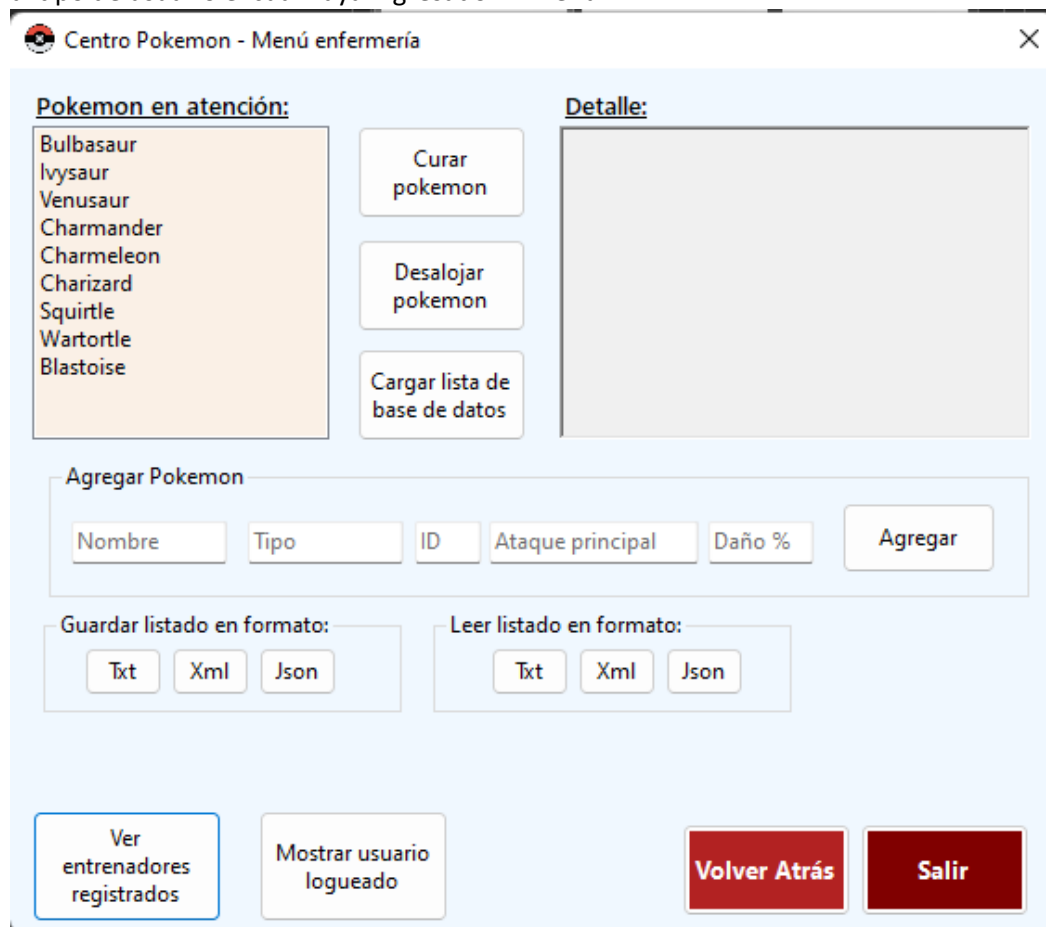


Instructivo – Centro Pokemon App

La app cuenta con un Login principal, donde ingresaremos como administrador o entrenador. Para acceder a todas las funcionalidades recomiendo ingresar como administrador, ya que podrás ingresar con el perfil de la Enfermera Joey, y así poder usar el sistema de atención y gestión que la app provee. Ingresando como entrenador, al momento actual solo podrás ver los pokemon alojados y sus detalles. El mismo consta de diferentes tipos de validaciones para asegurar un ingreso seguro y exitoso.



Una vez avanzado el login, nos encontraremos con un menú principal, el cual diferirá en base al tipo de usuario el cual haya ingresado. El menú:



En este menú, podremos ver la lista de pokemon alojados en el centro, y si presionamos click en algunos de ellos, podremos observar su detalle en la lista de la derecha.

También podremos agregar pokemon de manera manual, completando correctamente todos los campos necesarios para ello. O también podemos cargar pokemon desde una base de datos a través del botón Cargar lista desde base de datos.

En caso de querer preservar la información, podemos a través de los botones del group “guardar listado en formato”, guardar un archivo txt o xml, los mismos se guardarán en una carpeta en el escritorio. A su vez podemos leer esos archivos y visualizarlos en la lista de la derecha a través de los botones del group “Leer listado en formato”. Deberías haber guardado los archivos antes de intentar leerlos.

Tenemos el botón Curar Pokemon, que trabajará según el pokemon que tengas clickeado en ese momento, y si su daño es mayor a 0, efectuará el proceso de curación.

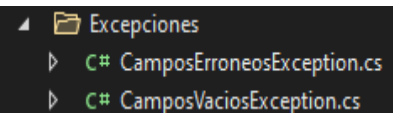
Y no menos importante, contamos con el botón Desalojar Pokemon, que si el daño del pokemon se encuentra en 0, lo desalojará el Centro.

Después tenemos funciones accesorias como “Ver entrenadores registrados”, que visualizará qué usuarios están registrados en el Centro. Y tenemos el botón “Mostrar usuario logueado”, que mostrará qué usuario está usando la app.

Y como toda app necesita, un botón para ir una ventana atrás, o para salir completamente de la app.

Temas aplicados:

Excepciones: Se crearon dos excepciones, se usan en el Login para verificar el correcto ingreso de datos. También se aplican en diferentes partes del código para que el mismo no rompa.



Métodos de extensión: Extendemos un string, el cual en este caso será el nombre de un pokemon, y aplicará el método curar.

```
public static string CurarPokemon(this string strpokemon)
{
    foreach (Pokemon pokemon in Pokemon.ListaPokemon)
    {
        if (strpokemon == pokemon.nombre)
        {
            pokemon.danio = 0;
            return $"{pokemon.nombre} ha sido curado y su daño es 0";
        }
    }
    return "No se encontró el pokemon";
}
```

```
private void btnCurarPokemon_Click(object sender, EventArgs e)
{
    if (this.lstPokemon.SelectedItem is not null)
    {
        string pokemonSeleccionado = (string)this.lstPokemon.SelectedItem;

        foreach (Pokemon pokemon in Pokemon.ListaPokemon)
        {
            if (pokemon.nombre == pokemonSeleccionado)
            {
                if (pokemon.danio == 0)
                {
                    MessageBox.Show("El pokémon no presenta daños.", "Elegi");
                }
                else
                {
                    MessageBox.Show(pokemonSeleccionado.CurarPokemon(), "Est");
                }
            }
        }
    }
}
```

Test unitarios: Se prueba el método de extensión nombrado antes, y un método que prueba la carga manual de un pokemon.

```
[TestClass]
0 referencias
public class PokemonTest
{
    //Pruebo método Carga de datos manual(AgregarPokemonManual)
    [TestMethod]
    0 referencias
    public void ProbarCargaDeDatosManualNuevoPokemon()
    {
        //Act
        bool resultado = Pokemon.AgregarPokemonManual("Chikorita", "Planta", 152, "Látigo Cepa", 33);

        //Assert
        Assert.IsTrue(resultado);
    }

    //Pruebo método de extensión(CurarPokemon)
    [TestMethod]
    0 referencias
    public void ProbarCurarPokemon()
    {
        //Arrange
        string pokemon = "Charmander";
        string resultadoEsperado = "Charmander ha sido curado y su daño es 0";

        //Act
        string resultado = pokemon.CurarPokemon();

        //Assert
        Assert.AreEqual(resultadoEsperado, resultado);
    }
}
```

Interfaces: Creamos una interfaz con el método MostrarDato, el cual se aplica en diferentes clases no relacionadas.





The screenshot shows the Visual Studio IDE. At the top, the `IDatos` interface is defined with a `MostrarDato()` method. Below it, the 'Implementaciones de "MostrarDato"' (Implementations of "MostrarDato") window is open, showing three implementations: `Administrador.MostrarDato()`, `Entrenador.MostrarDato()`, and `Pokemon.MostrarDato()`. The interface definition is as follows:

```
3 referencias
public interface IDatos
{
    /// <summary>
    /// Función que mostrará un dato
    /// </summary>
    /// <returns>Retornará un string</returns>
    12 referencias
    string MostrarDato();
}
```

The implementation window shows:

```
Implementaciones de "MostrarDato"
Toda la solución | Agrupar por: solo Proyecto
Código
Entidades (3)
string Administrador.MostrarDato()
string Entrenador.MostrarDato()
string Pokemon.MostrarDato()
```

Archivos y serialización: Los usamos para guardar y leer la lista de pokemon.

 EscribirTxt()	 LeerTxt(string archivo)
 EscribirXml(List<Pokemon> pokemon)	 LeerXml()

Base de datos:

Tenemos la base de datos CentroPokemon con la tabla PokemonAlojados. En la clase PokemonAccesoDatos se trabaja con su implementación, y así poder traer sus elementos a la lista, agregarlos y visualizarlos.

Delegados: Guardamos el método DesalojarPokemon en el atributo desalojarDelCentro y de esa manera trabajamos con delegados.

```
public delegate bool DesalojarDelCentro(string nombre);

Pokemon.DesalojarDelCentro desalojarDelCentro = Pokemon.DesalojarPokemon;

private void btnDesalojarPokemon_Click(object sender, EventArgs e)
{
    if (this.lstPokemon.SelectedItem is not null)
    {
        string pokemonSeleccionado = (string)this.lstPokemon.SelectedItem;
        if (desalojarDelCentro(pokemonSeleccionado))
        {
            MessageBox.Show("Se desalojó al pokemon.", "Desalojo exitoso",
                MostrarPokemonEnListaPokemon());
        }
    }
}
```