

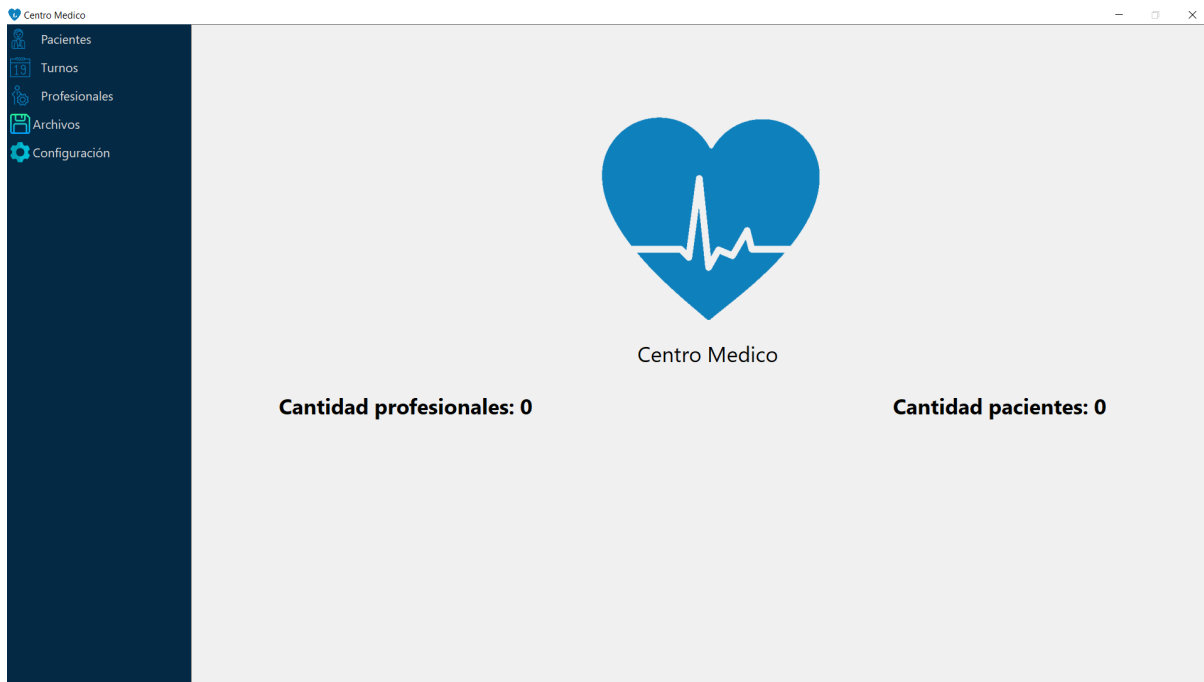
Trabajo Práctico N° 3 - Laboratorio de Computación II

Ledesma Leonel Franco - DIV 2E

Descripción del Programa:

El programa es un sistema para la atención al cliente de un Centro Médico. Donde se puede dar de alta pacientes, profesionales, crear turnos y buscar estas entidades, eliminarlas o modificarlas.

Funcionamiento del programa:



Al abrir el programa por primera vez, este no va a tener ningún dato cargado.

En la sección

Configuración:

- Se puede habilitar la opción de autoguardado
- La dirección donde se quiere guardar los datos del Centro Médico.
- Se va a poder elegir el nombre del Centro Médico.
- Se debe cargar las especialidades que va a poder registrar a cada profesional a la hora de darle el alta.
- Se puede modificar el horario de atención, ya que por defecto no atiende ningún día.

Centro Medico

Pacientes

Turnos

Profesionales

Archivos

Configuración

Configuración

Configuración

Autoguardado

Ubicación:

C:\Users\leone\Desktop\Backups\centro2.json

Abrir

Nombre del Centro:

Centro Medico

Guardar

Especialidades:

Cirugia

+

X

Horario de Atención del Centro Medico:

Lunes: No atiende

Martes: No atiende

Miercoles: No atiende

Jueves: No atiende

Viernes: No atiende

Sabado: No atiende

Domingo: No atiende

Cambiar horarios

Pacientes:

- Nuevo: Se podrá dar de alta un nuevo paciente (1)
- Ver/Buscar/Modificar: Se podrá buscar los pacientes por apellido, documento, nombre, teléfono o todos. Una vez seleccionado un paciente, se puede crear un turno, modificar (haciendo doble click en la tabla o a través del botón), eliminar, exportar, ver turnos o confirmar turno. (2)

(1)

Centro Medico

Pacientes

Turnos

Profesionales

Archivos

Configuración

Nuevo paciente

Nombre (*)

Apellido (*)

N° Documento (*)

Teléfono (*)

Fecha de nacimiento

Obra Social

N° Afiliado

Nacionalidad (*)

Argentina

Genero (*)

Femenino

Teléfono Contacto

Vaciar campos

Agregar paciente

(2)

Centro Medico

Buscar paciente

Buscar por: Apellido: pere

Coincidencias:

	Documento	Nombre	Apellido	Telefono	Obra Social	N° Afiliado	Edad
▶	42396381	Juan	Perez	1166666666			22

Ver turnos Modificar Eliminar

Nuevo turno Exportar Confirmar turno

Profesionales:

- Nuevo: Se podrá dar de alta un nuevo profesional, seleccionando sus especialidades haciendo doble click en la especialidad y horarios de atención, donde abre en pantalla el formulario para la selección de horarios. (1)
- Ver/Buscar/Modificar: Se puede ver turnos, modificar (haciendo doble click en la tabla o a través del botón), modificar horarios, exportar o eliminar. (2)

Centro Medico

Nuevo profesional

Nombre (*)

Apellido (*)

N° Documento (*)

Telefono (*)

Fecha de nacimiento

N° Matricula (*)

Genero (*)

Femenino

Nacionalidad (*)

Argentina

Especialidades

Cirugia

Medicina familiar

Horarios:

Seleccionar horarios

Vaciar campos

Agregar profesional

(1)

Centro Medico

Pacientes
Turnos
Profesionales
Archivos
Configuración

Buscar profesional

Buscar por: Apellido:

Coincidencias:

Documento	Nombre	Apellido	Telefono	DiasDeAtencion	Matricula	Edad
45856966	Micaela	Perez	1154256361	Lunes: Atiend...	1542859	22
42396935	Juan	Perez	113139060	Lunes: Atiend...	45857485	22

Ver turnos Modificar Eliminar

Modificar horarios Exportar

(2)

Turnos:

- **Nuevo turno:** Abre el formulario de búsqueda de pacientes, para hacer click en el botón de Nuevo Turno. Una vez seleccionado el paciente y hecho clic en Nuevo turno, se puede seleccionar por Especialidad o Profesional, abriendo así diferentes menús para la elección del turno, teniendo en cuenta los días que atiende el centro, y los días que atiende el profesional seleccionado.(1)
- **Buscar turno:** Abre el formulario de búsqueda de turnos, donde se puede buscar por documento del paciente, apellido del profesional, id del turno, o todos. Al mismo tiempo que se pueden buscar futuros, pasados o todos. Donde se puede modificar, exportar o eliminar dicho turno. (2)
- **Ver turnos:** Se pueden ver los turnos de una forma más enfocada, por fecha o todas las fechas, por profesional o todos los profesionales y por documento del paciente o todos los pacientes, en todas sus combinaciones. Pudiendo exportar todas las coincidencias. (3)

Nuevo turno para Leonel Ledesma

Seleccionar por:

Seleccione profesional:

Especialidades:

Seleccione Fecha:

Seleccione Horario:

Confirmar Turno

(1)

Buscar por: Apellido
 Documento paciente Futuros

Coincidencias:

	Id	Fecha	Paciente	Profesional	Especialidad	Asistió
▶	4	06/06/2022 11:00	Exequiel Ledesma	Leonel Ledesma	Cirugia	<input type="checkbox"/>

Modificar

Eliminar

Exportar

(2)

Ver turnos

Fecha: ☒ Todos
 Profesional: Todos
 Documento paciente: Todos Buscar

Exportar coincidencias

	Fecha	Paciente	Profesional	Especialidad	Id	PacienteAsistio
▶	06/06/2022 10:00	Leonel Ledesma	Leonel Ledesma	Cirugia	1	<input type="checkbox"/>
	06/06/2022 10:30	Leonel Ledesma	Leonel Ledesma	Medicina Familiar	2	<input type="checkbox"/>
	13/06/2022 10:00	Leonel Ledesma	Leonel Ledesma	Medicina Familiar	3	<input type="checkbox"/>
	06/06/2022 11:00	Exequiel Ledesma	Leonel Ledesma	Cirugia	4	<input type="checkbox"/>

(3)

Archivos:

Se pueden exportar las siguientes listas a .json o .xml.

- **Importar pacientes**
- **Importar profesionales**
- **importar centro**
- **Exportar pacientes**
- **Exportar profesionales**
- **Exportar centro**



Archivos

Importar pacientes
 Importar profesionales
 Importar centro
 Exportar pacientes
 Exportar profesionales
 Exportar centro

Temas a aplicar:

Los temas a aplicar son:

1. **Excepciones:** Fueron aplicadas varias veces en el proyecto para poder controlar los posibles errores en el tiempo de ejecución. Algunos de los ejemplos son:

```
public CentroMedico DeserializarFromXML(string path)
{
    StreamReader streamReader = null;
    try
    {
        streamReader = new StreamReader(path);

        XmlSerializer xml = new XmlSerializer(typeof(CentroMedicoData));
        CentroMedicoData centroMedico = xml.Deserialize(streamReader) as CentroMedicoData;
        return (CentroMedico)centroMedico;
    }
    catch (FileNotFoundException e)
    {
        throw new DeserializarException("No se ha encontrado el archivo del centro medico.", e);
    }
    catch (Exception e)
    {
        throw new DeserializarException("Error al deserializar del centro medico", e);
    }
    finally
    {
        if (streamReader != null)
            streamReader.Close();
    }
}
```

Donde se creó un nuevo tipo de excepción con el fin de controlar este tipo de errores al deserializar las entidades.

2. **Pruebas unitarias:** Se implementaron para confirmar el correcto funcionamiento de algunas implementaciones.

```
public class TestCentroMedico
{
    [TestMethod]
    // 0 referencias
    public void EliminarPaciente_CuandoNoExisteEnLaListaDePacientes_DeberiaRetornarFalse()
    {
        // Arrange
        CentroMedico centroMedico = CentroMedico.Instanciar("centro");
        Paciente paciente = new Paciente("Leonel", "Ledesma", new System.DateTime(2000, 02, 10),
            Persona.eGenero.Masculino, Persona.eNacionalidad.Argentina, "41458745", "1145858574", "", "", "");
        centroMedico.AgregarPersona(paciente);
        Paciente paciente2 = new Paciente("Juan", "Perez", new System.DateTime(1960, 10, 05),
            Persona.eGenero.Masculino, Persona.eNacionalidad.Argentina, "18458969", "458599*/", "", "", "");

        bool result;
        //act
        result = centroMedico.RemoverPersona(paciente2);

        //Assert
        Assert.IsFalse(result);
    }
}
```

3. **Tipos genéricos:**
4. **Interfaces:**
5. **Serialización:** Tanto las interfaces como tipos genéricos fueron utilizados en diferentes ámbitos, con el fin de facilitar ciertas implementaciones de la serialización. Creando así `ISerializable` e `IDeserializable`.

```

public interface IDeserializable<T>
{
    /// <summary> Deserializa la entidad desde un archivo .XML, devolviendo un objet ...
    2 referencias
    public T DeserializarFromXML(string path);

    /// <summary> Deserializa la entidad desde un archivo .JSON, devolviendo un obje ...
    2 referencias
    public T DeserializarFromJson(string path);
}

```

```

/// <summary> Serializa una lista del centro medico a un archivo XML.
3 referencias
public void SerializarListaToXML<T>(string fullPath, List<T> lista)
{
    StreamWriter streamWriter = null;
    try
    {
        streamWriter = new StreamWriter(fullPath);

        XmlSerializer xml = new XmlSerializer(typeof(List<T>));
        xml.Serialize(streamWriter, lista);
    }
    catch (Exception e)
    {
        throw new SerializarException("Error al serializar la lista", e);
    }
    finally
    {
        if (streamWriter != null)
            streamWriter.Close();
    }
}

```

Permitiendo así poder serializar cualquier tipo de lista del centro médico.

6. **Archivos:** Además de utilizarse para la serialización y deserialización, lo utilizo para guardar la configuración de autoguardado.

```

/// <summary> Guarda el autoguardado y la direccion de este en el directorio don ...
1 referencia
private void GuardarConfiguracion()
{
    GestorDeArchivos.GuardarArchivo($"{Directory.GetCurrentDirectory()}.cds", $"{this.AutoGuardado},{this.PathAutoGuardado}");
}

```

```

/// <summary> Guarda en el path recibido la cadena de texto de datos.
2 referencias
public static bool GuardarArchivo(string path, string datos)
{
    try
    {
        using (StreamWriter sw = new StreamWriter(path))
        {
            sw.WriteLine(datos);
        }
        return true;
    }
    catch (Exception)
    {
        throw new Exception("Ocurrio un error al guardar el archivo.");
    }
}

```