

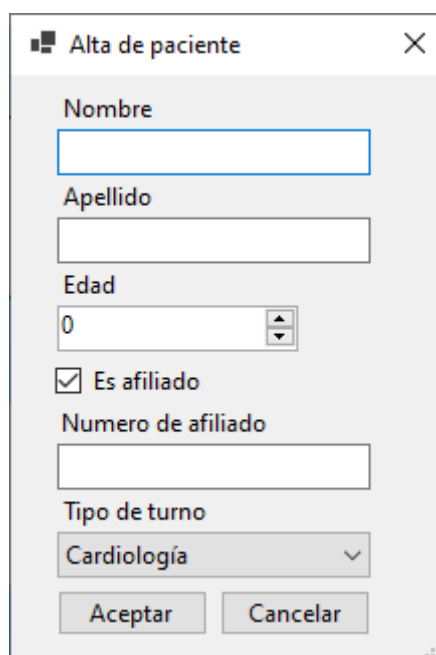
Trabajo Práctico Nº 3

Laboratorio de computación II

Se desarrolla un sistema de registro de turnos para un hospital, en el que pueden encontrarse pacientes afiliados y no afiliados. El formulario principal consiste en dos botones, uno para cargar un turno nuevo a mano y el otro para listar los turnos disponibles.



Al seleccionar la opción de alta de paciente se abre un formulario nuevo en forma modal para realizar la carga de datos correspondiente.


The image shows a modal window titled 'Alta de paciente'. It contains the following fields and controls: a text input for 'Nombre', a text input for 'Apellido', a spin box for 'Edad' with the value '0', a checked checkbox for 'Es afiliado', a text input for 'Numero de afiliado', and a dropdown menu for 'Tipo de turno' currently showing 'Cardiología'. At the bottom are 'Aceptar' and 'Cancelar' buttons.

En el formulario de alta de paciente se cargan todos los datos que se encuentran validados por excepciones:

- Si en nombre/apellido se ingresan caracteres diferentes a letras, al elegir aceptar se muestra un MessageBox. Al destildar el CheckBox de afiliado, el Label siguiente cambia automáticamente de “Numero de afiliado” a “DNI”.
- Si es afiliado, el “Número de afiliado” debe comenzar con A. Caso contrario, el DNI debe constar de sólo 8 números.

Si los datos son correctos y el paciente NO se encuentra ya en la lista, al aceptar se agrega el paciente a la lista de pacientes y se vuelve al formulario de inicio.

Al seleccionar la siguiente opción, “Listar pacientes”, se genera otro formulario con un data grid en el que se cargarán todos los pacientes.



The screenshot shows a window titled "Listado de pacientes" with a close button (X) in the top right corner. Inside the window is a data grid with the following columns: ID, Nombre, Apellido, Edad, Derivar a, and DNI/Nº afiliado. The grid is currently empty. Below the grid are two buttons: "Serializar Pacientes XML" and "Deserializar Pacientes XML".

Si no se cargan manualmente, la vista inicial del formulario será esta, sin filas en el data grid.

Al cargar pacientes se irán completando por filas, asignándole un ID autoincremental a cada uno. El usuario tendrá permitido ordenar de acuerdo con lo que le convenga.

Listado de pacientes

	ID	Nombre	Apellido	Edad	Derivar a	DNI/Nº afiliado
▶	1	Federico	Nealon	28	Cardiología	A0001
	2	Federico	Brian	29	Cardiología	A0002
	3	Nico	Lino	56	Neurología	20987654
	4	Alejandra	Campos	14	Pediatría	49123456

Serializar Pacientes XML

Deserializar Pacientes XML

Los botones de este formulario realizan las acciones indicadas en su texto.

Temas e implementación

- Clase 10, Excepciones:** La solución contiene una Biblioteca de clases dedicada a las Excepciones, con el mismo nombre. Son utilizadas en el formulario FormAltaPaciente para validar los datos del paciente a cargar, y en el manejo de archivos en la clase SerializacionXml.
- Clase 11, Pruebas unitarias:** La solución contiene un proyecto de tipo Pruebas de MSTest, llamado UnitTesting en el que se prueban las excepciones NumeroAfiliadoInvalidoException y DniInvalidoException, así como que el operador + de la clase Pacientes no agregue a su lista un paciente ya ingresado.
- Clase 12, Tipos Genéricos:** Se declara en el proyecto de Biblioteca de clases Entidades la clase Pacientes, de tipo genérica, con la restricción de que T debe ser de tipo Persona o derivadas.

- **Clase 13, Interfaces:** Se hace uso de la interfaz IArchivo, en la que se declaran los métodos Guardar y Leer, que serán implementados por las clases SerializacionXml y Pacientes.
- **Clase 14, Archivos:** Junto con la interfaz descrita en el punto anterior, se hace manejo de archivos desde el formulario FormListadoDePacientes y en la clase Pacientes.
- **Clase 15, Serialización:** Se utilizan en la clase SerializacionXml.