

Trabajo Práctico Nº 4

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.


A screenshot of a modal form titled 'Alta de paciente'. The form is light grey and 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' with 'Cardiología' selected. At the bottom, there are two buttons: 'Aceptar' and 'Cancelar'.

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 four buttons: "Serializar Pacientes XML", "Deserializar Pacientes XML", "Guardar archivo de texto", and "Leer archivo de texto".

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
	5	Marcia	Gonzalez	34	Clínico	33097123

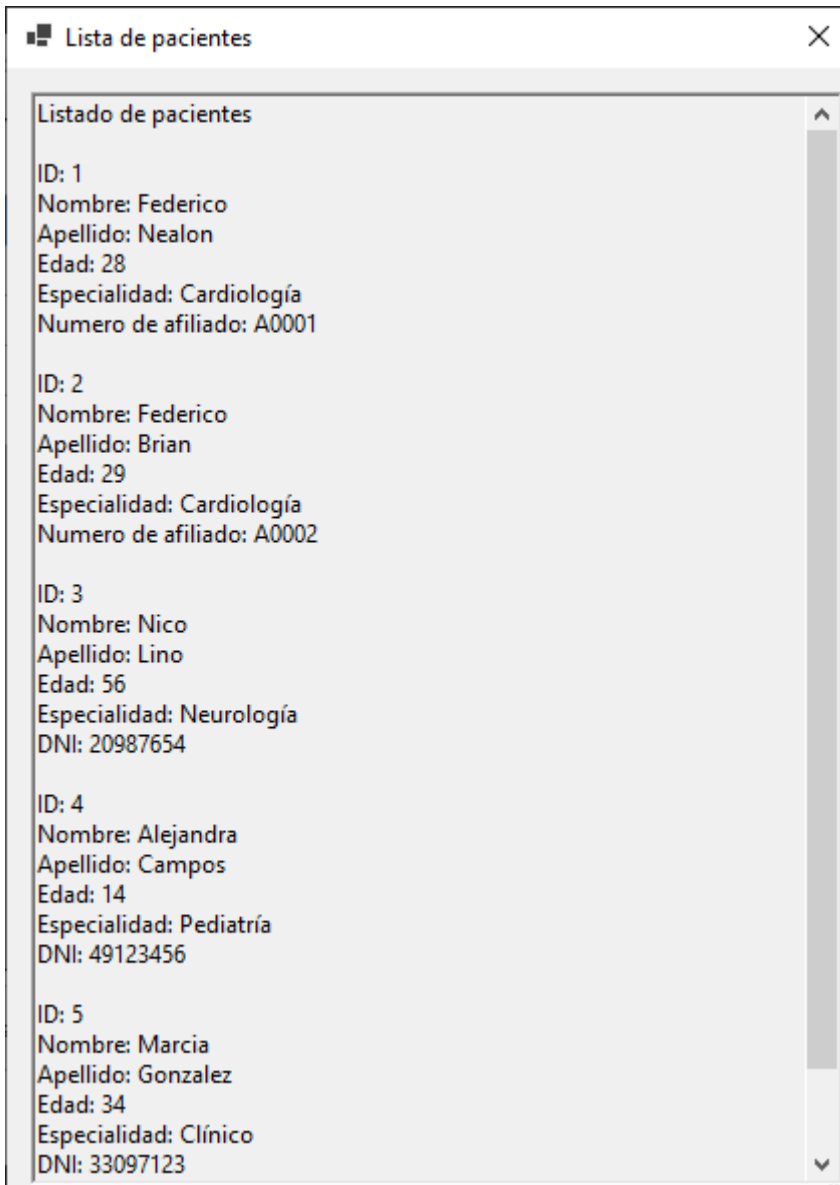
Serializar Pacientes XML

Deserializar Pacientes XML

Guardar archivo de texto

Leer archivo de texto

Los botones de este formulario realizan las acciones indicadas en su texto. En el caso del botón “Leer archivo de texto”, genera otro formulario con un Rich Text Box que no puede editarse, con el listado de pacientes que se encuentran guardados en el archivo de texto.



Lista de pacientes

Listado de pacientes

ID: 1
Nombre: Federico
Apellido: Nealon
Edad: 28
Especialidad: Cardiología
Numero de afiliado: A0001

ID: 2
Nombre: Federico
Apellido: Brian
Edad: 29
Especialidad: Cardiología
Numero de afiliado: A0002

ID: 3
Nombre: Nico
Apellido: Lino
Edad: 56
Especialidad: Neurología
DNI: 20987654

ID: 4
Nombre: Alejandra
Apellido: Campos
Edad: 14
Especialidad: Pediatría
DNI: 49123456

ID: 5
Nombre: Marcia
Apellido: Gonzalez
Edad: 34
Especialidad: Clínico
DNI: 33097123

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.

Correcciones de TP3

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
	5	Marcia	Gonzalez	34	Clínico	33097123

Serializar Pacientes XML

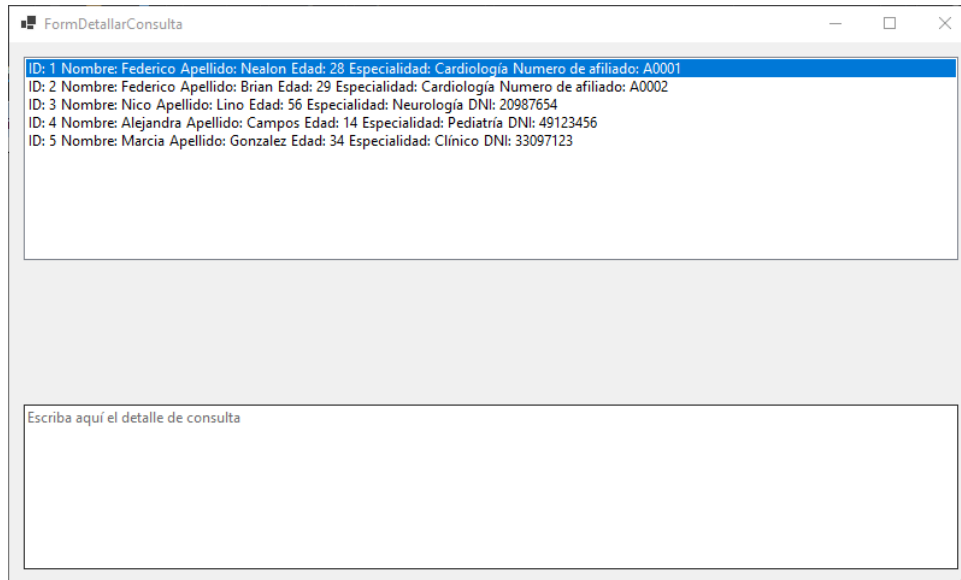
Deserializar Pacientes XML

Guardar archivo de texto

Leer archivo de texto

Detallar Consulta

Se agrega el botón de Detallar Consulta, con el que se abre un nuevo formulario con un List Box conteniendo todos los pacientes y un Text Box en el que se solicita el detalle de la consulta.

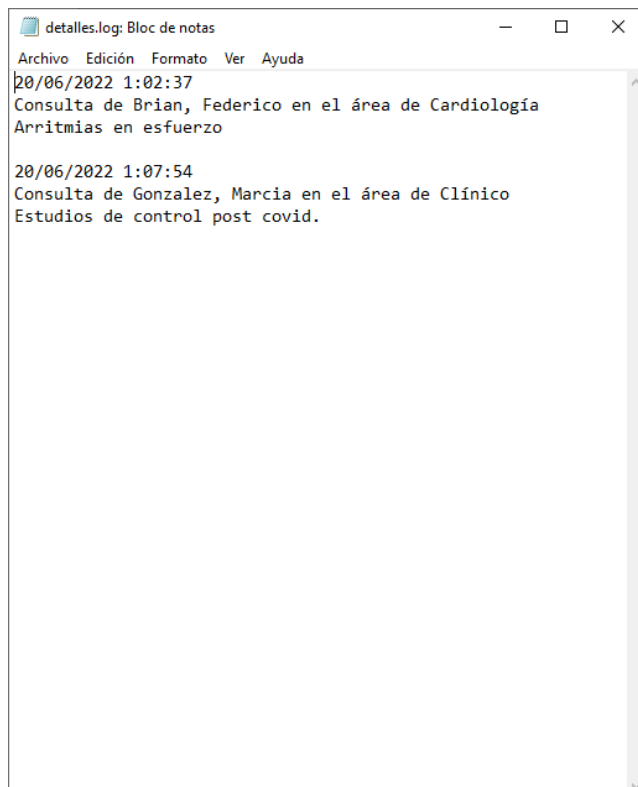


FormDetallarConsulta

ID: 1 Nombre: Federico Apellido: Nealon Edad: 28 Especialidad: Cardiología Numero de afiliado: A0001
ID: 2 Nombre: Federico Apellido: Brian Edad: 29 Especialidad: Cardiología Numero de afiliado: A0002
ID: 3 Nombre: Nico Apellido: Lino Edad: 56 Especialidad: Neurología DNI: 20987654
ID: 4 Nombre: Alejandra Apellido: Campos Edad: 14 Especialidad: Pediatría DNI: 49123456
ID: 5 Nombre: Marcia Apellido: Gonzalez Edad: 34 Especialidad: Clínico DNI: 33097123

Escriba aquí el detalle de consulta

Una vez escrito el detalle, al hacer doble click en alguno de los pacientes se presenta un Message Box que solicita la confirmación del detalle a cargar. En caso de aceptar y no haber anotado nada, se lanza una excepción que es atrapada para poder escribir lo necesario. Si todo es correcto, se guarda en la ubicación de la app el archivo detalles.log, en donde se cargan la hora actual, la especialidad en la que se realizó la consulta y el nombre y apellido del paciente.



detalles.log: Bloc de notas

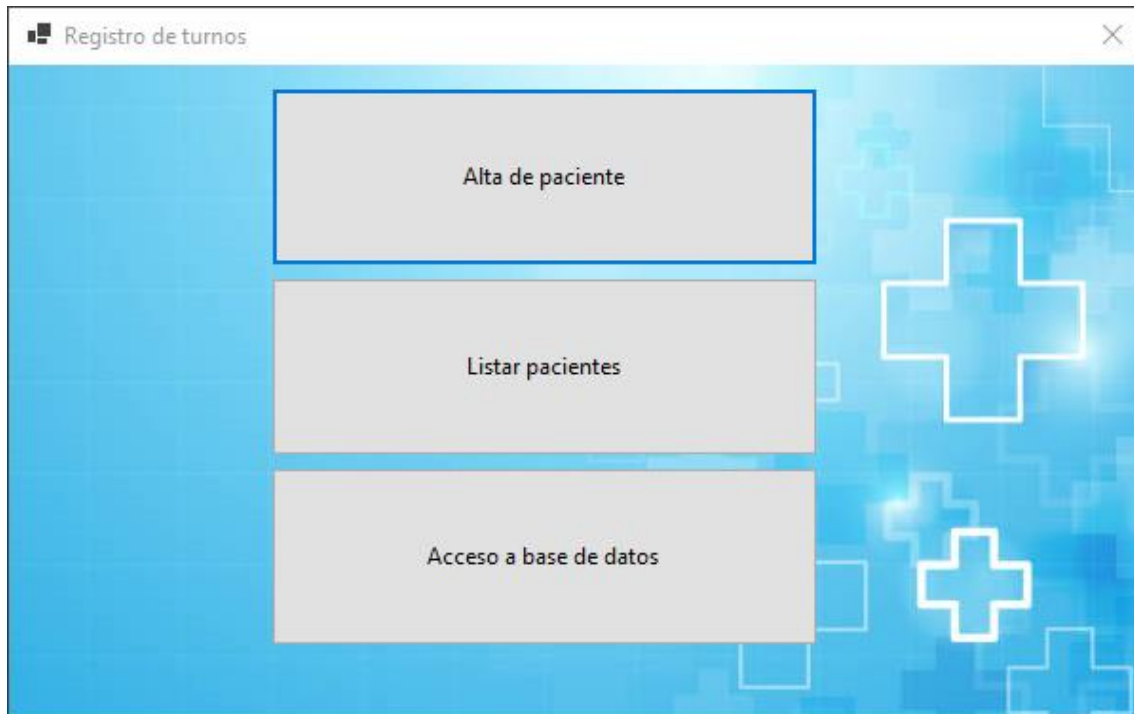
Archivo Edición Formato Ver Ayuda

20/06/2022 1:02:37
Consulta de Brian, Federico en el área de Cardiología
Arritmias en esfuerzo

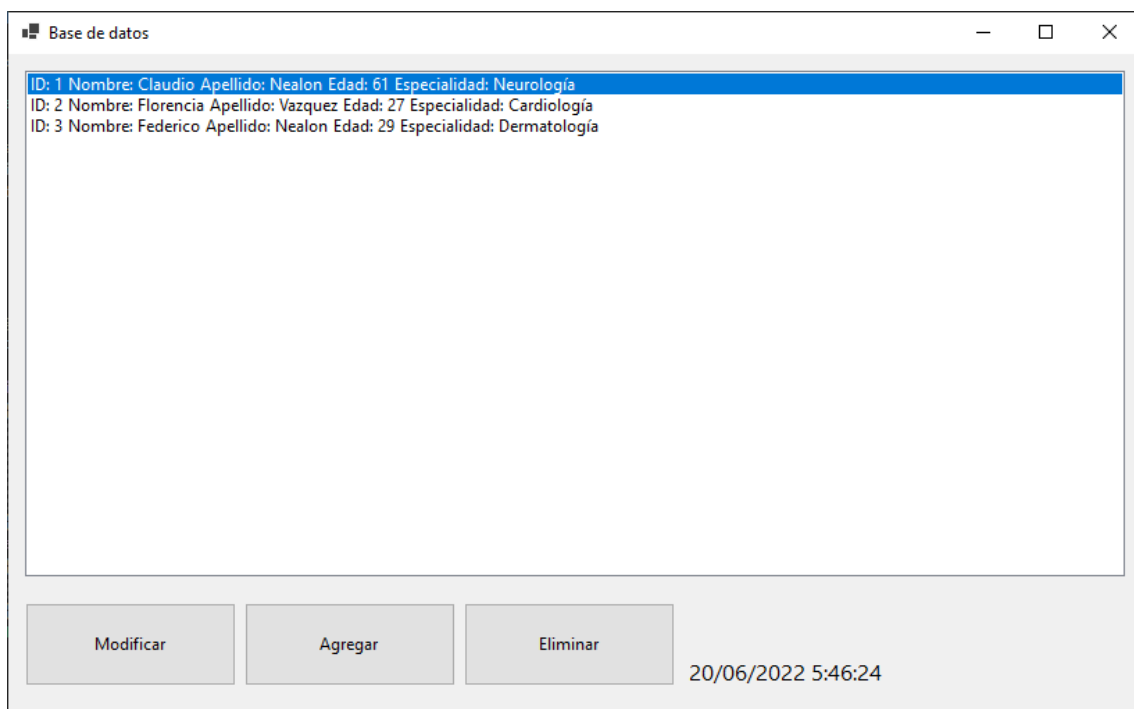
20/06/2022 1:07:54
Consulta de Gonzalez, Marcia en el área de Clínico
Estudios de control post covid.

Con el fin de facilitar la corrección se generan la clase DAO y un formulario dedicado a la base de datos. Para generar la base de datos será necesario que el punto de conexión se llame localhost, o modificar el Connection String ubicado en la clase DAO. La base de datos a crear se llamará *pacientes_db*, y el script se encuentra dentro de la carpeta del trabajo práctico con el nombre *archivoScript.sql*.

En el formulario de inicio se agrega un botón para el acceso a la base de datos.



Una vez seleccionada dicha opción, se abre el formulario dedicado a la BBDD con el siguiente formato:



En él se observan 3 botones que realizarán las acciones indicadas. En el caso de los botones de Modificar y Eliminar, se cargarán los datos del ítem seleccionado en el List Box en un formulario FormAltaPaciente, con la posibilidad de modificar los valores.

- **Clases 16 y 17, Bases de datos, SQL:** Se declara en el proyecto de Biblioteca de clases Entidades la clase DAO (Data Access Object), con métodos para obtener los pacientes de la base de datos, agregarlos, modificarlos y eliminarlos.
- **Clase 18 y 20, Delegados y Eventos:** Se declara el delegado DelegadoConsultaOldRepetido para ser utilizado con el evento ConsultaOldRepetido, que notifica al usuario que se está repitiendo una consulta al mismo paciente en la misma especialidad o se está intentando cargar una consulta nueva con un ID ya utilizado. Si alguno de estos casos se presenta, la consulta no es cargada en la base de datos y se le notifica al usuario a través de un Message Box.
- **Clase 19, Hilos:** Se utiliza un hilo para mostrar en el formulario de base de datos la fecha y hora actual, siendo actualizada cada segundo.