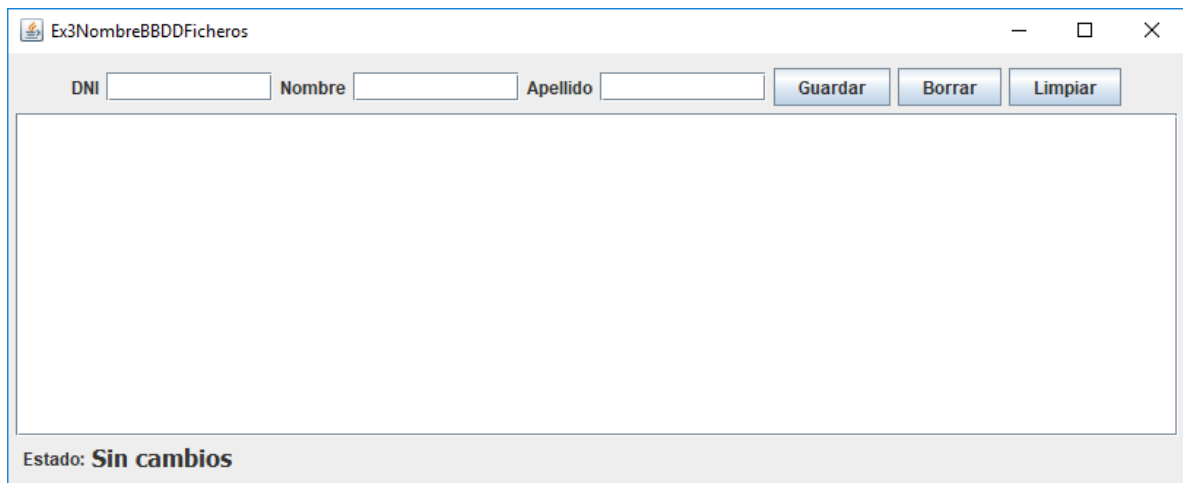
	Curso / Kurtsoa	Fecha / Data	Nivel / Maila	Eval. / Ebal.
	2018/2019	27/05/2019	1	3
Módulo / Modulua	Kodea / Código	U.Didak /Unid didác	Tipo / Mota	Calificación/Kalifikazioa
Programación	PROG	1-12	P	
Nombre / Izena				

Crea un nuevo proyecto Java. Dentro de ese proyecto crea un nuevo paquete de nombre **examen3XXX** (donde XXX es tu nombre) y guarda dentro de él todas las clases que necesites.

Cada parte del examen tiene un valor y algunos ejercicios se subdividen en partes que a su vez tienen un valor determinado. Si no se consigue cada subparte completamente, el valor del mismo será 0.

Para crear la clase **Ev3NombresBBDDFicheros**, tienes disponible la clase **Persona** y puedes utilizar el interfaz **Ev3NombresBBDDFicheros_Interface**



Ev3NombresBBDDFicheros es una clase que tiene un contenedor de nombre **contenedor** con varios componentes.

En la parte central, hay un JList de nombre **lstPersonas**, que permite la selección múltiple y que contiene elementos de la clase **Persona**. Los datos de la lista estarán en un **DefaultListModel** de nombre **dlnPersonas**.

1ª Parte (1 punto)

lblDni. JLabel con el texto “DNI”.

txtDni. Al pulsar Enter, se ejecutará el código del botón **btnGuardar**. Al recibir el foco, se seleccionará el texto del textfield.

lblNombre. JLabel con el texto “Nombre”.

txtNombre. Al pulsar Enter, se ejecutará el código del botón **btnGuardar**. Al recibir el foco, se seleccionará el texto del textfield.

lblApellido. JLabel con el texto “Apellido”.

txtApellido. Al pulsar Enter, se ejecutará el código del botón **btnGuardar**. Al recibir el foco, se seleccionará el texto del textfield.

2ª Parte (2 puntos)

btnGuardar. JButton con el texto “Guardar”. Al hacer clic en este botón, se creará un nuevo objeto de la clase Persona con los datos de los campos DNI, Nombre y Apellido, y si esa persona no está en la lista (si el DNI no está en la lista), lo añadirá a la lista. Para añadir una Persona a la lista no puede estar vacío ninguno de los textfields. **(1 punto)**

btnBorrar. JButton con el texto “Borrar”. Al hacer clic, se borrarán todos los elementos de la lista que estén seleccionados. **(0,5 puntos)**

btnLimpiar. JButton con el texto “Limpiar”. Al hacer clic, se eliminarán todos los elementos de la lista. **(0,5 puntos)**

3ª Parte (1 punto)

En la parte inferior, hay un JPanel de nombre panelEstado que tiene un JLabel de nombre lblEstado y con el texto “Estado”. Existe otro JLabel de nombre **lblEstadoActual** con el texto “**Sin cambios**”. Cuando los datos cambian, aparecerá el mensaje “**Datos modificados**”

4ª Parte (4 puntos)

Al pulsar la X de sistema, saldremos de la aplicación. Si los datos han sido modificados, se lanzarán dos procesos:

Guardar datos en el fichero. Las personas que están en la lista, se guardarán en el fichero **personas.dat**. Para ello, se debe programar un método llamado **GuardarDatos**.

Guardar datos en la Base de datos. Las personas de la lista se guardarán en la tabla personas de la **base de datos ev3**, eliminando los datos que existen anteriormente.

5ª Parte (2 puntos)

Al iniciar la aplicación, las personas que existen en la tabla personas de la base de datos ev3 se introducirán en el datamodel y en el JList. Para ello utilizaremos el metodo **Cargar Datos**.

Criterios de Corrección

Al finalizar el tiempo se recoge el examen y se guarda el paquete con las clases dentro de una carpeta de nombre **Apellido, Nombre** donde Apellido hace referencia tu primer apellido y nombre a tu nombre. Por ejemplo, *De Miguel, Txema*.

No se permitirá ningún tipo de comunicación durante el examen. Los cables de red permanecerán desconectados, las conexiones inalámbricas (si las hubiera) se deshabilitarán y los **móviles** deberán estar **apagados**. Si un alumno incumple alguna de estas normas entregará su examen y tendrá un 0 como nota.