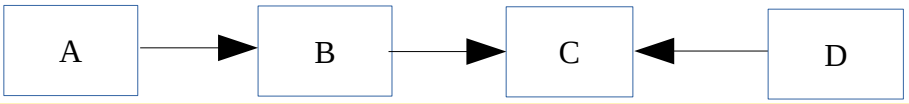
PRÁCTICA 6 de Java: Final

Fecha tope entrega	viernes, 10 diciembre.	**Fecha tope defensa**	viernes, 10 diciembre.
Tipo	Individual	**Asunto en email**	JAVA06
Formato fichero	ApellidoNombreJAVA06 (por invitación al repositorio privado de Github)		

Requisitos:

1.- Base de datos: desarrolla una base datos libre que deberá ser tener el visto buenos del profesor. El sistema de BDs será Derby y tendrá las siguientes características:

1.1- Crea 4 tablas con una relación 1→N y otra 1→N←1, como las del siguiente ejemplo :



1.2.- La tabla A sería algo como socio, cliente, usuario, etc. Para validarse en la aplicación con los campos usuario y contraseña. Tendrá obligatoriamente: NIF, foto y fecha.

* 1.3.- A, B y D tendrán sus PK tipo entero. En C la PK estará formada por 3 campos: las dos FKs procedentes de B y D y un contador relativo a la FK de B o D.

1.4.- Habrá un campo calculado en C que dependerá de valores de B y/o D.

1.5.- Existirán campos de tipo entero, real, alfanumérico y fecha.

2.- Vista. Desarrolla una aplicación gráfica con las siguientes características:

2.1.- Los JFrame y los JPanel deberán ser clases en ficheros independientes.

2.2,. Inicialmente no se podrá acceder a la aplicación o los diferentes opciones estarán deshabilitadas, salvo para iniciar sesión y salir..

2.3.- Inicio de sesión: consistirá en validar una fila de la tabla A (socio, cliente, etc.) mediante el nombre y la contraseña. Esto es independiente de la conexión a la BDs con el usuario administrador, aunque se aconsejan que se hagan juntas

"Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/empresa","empresa","empresa");
No confundir con validación que es una consulta"

2.2.1.- Permitirá acceder a la aplicación o habilitará todas las funciones de la aplicación (menús).

2.4.- Cerrar sesión. Deshabilita el uso de la aplicación.

2.5.- Panel ver filas B una a una. Solo podrá ver las filas de B **relacionadas** con elemento de A validado.

2.5.1.- Tendrá los botones: avanzar, retroceder, primero y último.

"Probar los métodos isFirst() e isLast().
https://docs.oracle.com/javase/7/docs/api/java/sql/ResultSet.html"

* 2.5.2- Ver filas de C. En un Jlist se irán mostrando aquellas relacionadas con el elemento de B mostrado. Deberán estar en el mismo panel.

2.6.- Panel nueva fila C. Podrá crearse una nueva fila en C.

2.6.1.- Permitirá seleccionar el elemento de la tabla B y D con los que se relaciona.

2.6.2.- Formará su PK con las PKs de B y D seleccionadas y por un número correlativo a una de ellas

2.6.3.- El campo calculado se hallará automáticamente.

2.7.- Panel propiedades de la tabla A: mostrará todos los campos de la tabla A.

2.7.1.- Permitirá modificar la imagen, una fecha y el NIF. Utilizad un botón guardar:

2.7.2.- Las imágenes estarán en una carpeta pero se podrán buscar en cualquier lugar mediante un filechooser, para luego copiarla a dicha carpeta.

2.7.3.- La fecha, se podrá modificar usando DatePicker o DataChooser. Deberá haber la posibilidad de errores en la fecha controlados por el sistemas de errores.

2.7.4.- El NIF será validado mediante la librería adjunta ValNif08201.zip de la Agencia Tributaria.

2.8.- Panel Acerca de.

3.- Modelo:

3.1.- Habrá una clase por cada tabla de la base de datos.

3.2.- (Voluntario) Existirá una o varias clases para todas las consultas. Sus métodos llevarán la consulta y los campos (PreparedStatement) por parámetro. Devolverán una collection de objetos o un objeto, según el tipo de consulta.

4.- Controlador. Estará compuesto de varias clases:

4.1.- Clase para la gestión de la conexión y validación.

4.2.- En la validación usará un PreparedStatement.

4.3.- Clase o clases (si se hace por tablas de la BDs) para la gestión de las operaciones select, insert y update. Sus métodos reciben una cadena y devuelven un objeto o una collection de objetos.En los insert o update, reciben una cadena y devuelven un entero con el número de filas afectadas.

4.4.- Clase especial para mostrar uno a uno los registros B. Tendrá, al menos, los métodos lógicos: iniciar(sql), avanzar(), retroceder(), irPrimero(), irUltimo() y finalizar(). Además del método “objeto leer()”.

* 4.5.- Para las consultas sobre la tabla C necesarias en el apartado 2.5.2 se usará un PreparedStatement.

Nota PreparedStatement:

https://www.arquitecturajava.com/jdbc-prepared-statement-y-su-manejo/

https://docs.oracle.com/javase/7/docs/api/java/sql/PreparedStatement.html

5.- Sistemas de Errores (dentro del controlador)

5.1.- Habrá una clase de errores unificados: será estática, gestionará todos (o casi todos) los errores de forma preestablecida y se usará para informar al usuario de la aplicación.

- Consistirá en una estructura de pares u objetos: numero de error y mensaje de error.

- Tendrá un método con un número por parámetro y devolverá el mensaje asociado.

5.2.- Habrá una clase excepción personalizada para propagar o subir los errores/excepciones del sistema, y así gestionarlos con el sistema de errores unificados (apartado 5.1). Hay que añadirle el atributo número de error.

5.3.- Funcionamiento: en los catch del sistema se realizarán dos acciones:

- Se aconseja en la fase de desarrollo, imprimir en consola la fecha, hora y el mensaje del sistema (excepción Java real)

- Guardará esos mismos datos en un fichero de texto a modo de log.

- Finalmente: lanzará una excepción personalizada (5.2) con el número de error dado (5.1).

5.4.- Si el error está controlado por un “if” u otra instrucción, también se lanzará la excepción personaliza o se obtendrá el mensaje de error tras buscarlo por su número.

6.- Exigencias del MVC:

6.1.- Será en la vista donde muestren los mensaje (JOptionPane, JDialog, etc) informativos o de error, con la ayuda de la Clase de errores unificados (5.1). No se podrán llamar desde otros lugar.

6.2.- Independencia de clases, sobre todo por capas del MVC. Las clases se comunicarán con otras clase a través de métodos y no podrán acceder directamente a los atributos. PE: No se puede usar directamente un ResultSet en un JPanel.

6.3.- Cada parte del MVC tendrá su propio paquete.

7.- Se usará un repositorio git.