

Fecha tope de entrega: **22 de MAYO de 2020 a las 23:55**

La nota será de máximo **5 puntos**.

Los elementos marcados como **AMPLIACION** valdrán **2 puntos** más.

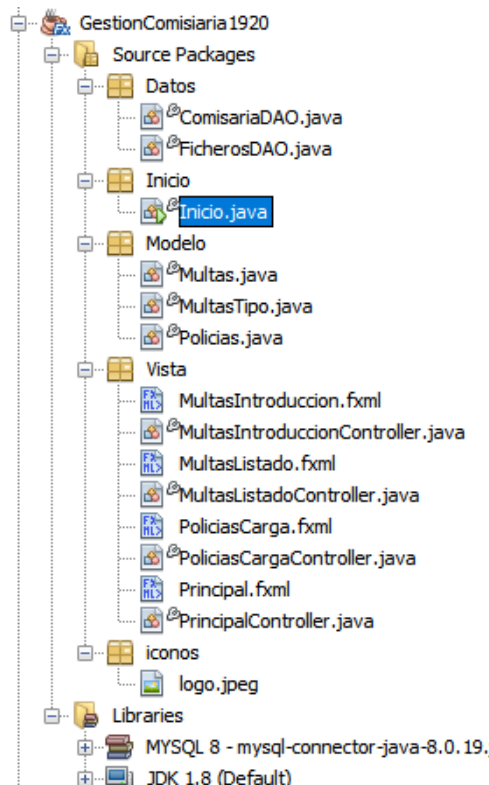
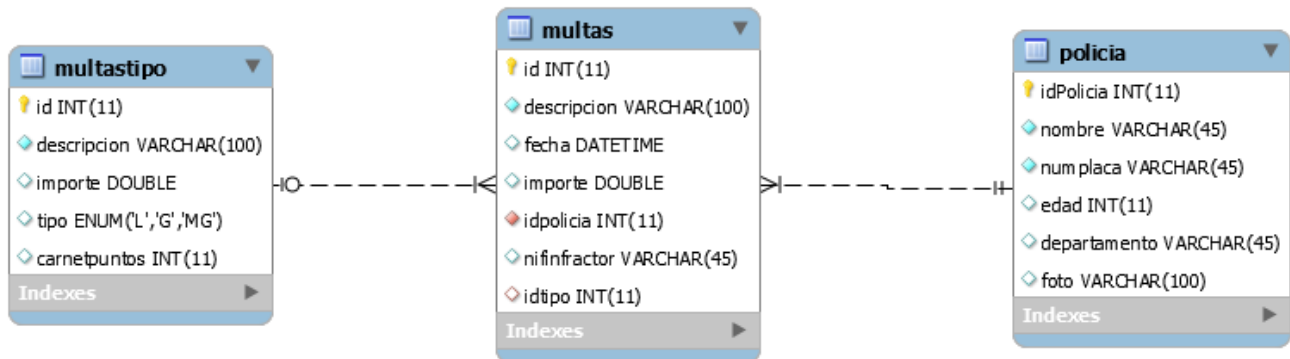
La presentación ante la profesora, por videoconferencia, será del 25 de mayo al 29 de mayo, valdrá **3 puntos** más si se contesta de forma correcta a las preguntas. **Si no se saca mínimo 1 punto en este apartado la práctica se considerará suspendida.**

Se valorarán criterios de calidad de código: eficiencia temporal y espacial, control de excepciones y aviso al usuario de forma gráfica, código formateado y diseño de ventanas trabajado y usable.

Se pueden añadir campos a las tablas, ventanas y cualquier componente gráfico, siempre que se cumplan primero los requisitos aquí detallados.

COMISARIA DE POLICÍA

Basándote en el esquema siguiente implementado en el archivo **comisaria.sql**, crea un aplicación gráfica mediante un proyecto de **JAVA-FX**, para la gestión de policías de una comisaria y las multas que estos han interpuesto.



El proyecto estará estructurado en 3 capas, por tanto trabajaremos como mínimo con tres packages:

- **MODELO**, se almacenarán las clases base que representan a policías, multas y tipos de multa.
- **VISTA**, que alberga las clases que representan a la interacción con el usuario (ventanas de cada una las opciones que se piden en el ejercicio y sus controllers). Por ejemplo principal, introducción de multas,.....
- **DATOS**, que tendrá clases encargadas de las operaciones de base de datos y archivos. Deben recibir y/o enviar objetos de las clases base.
 - **ComisariaDAO.java**: todo lo relativo a bd/jdbc
 - **ArchivosDAO.java**: todo lo relativo a ficheros.

REQUISITOS DE LA APLICACIÓN

PRINCIPAL

Ventana que permitirá acceso (mediante botones, utilizando menús....) al resto de opciones.

INTRODUCCION DE MULTAS

- Permitirá **insertar datos** en la tabla multas comprobando que los campos not null de la tabla tienen valor.
- Para seleccionar el policía, se mostrarán como mínimo nombre y nº de placa de todos los existentes. Utilizar objetos Policia
- Permitir filtrar por departamento los policías mostrados
- Tipo de multa: visualizar los valores desde la tabla multas-tipo
- Importe (Spinner): se deberá calcular según el tipo de multa, actualizando su valor cada vez que cambie el tipo de multa, pero permitiendo cambiar su valor
- Visualización de la foto (componente *ImageView*)
- Se permitirá seleccionar o no descuento del 20%. Deberá actualizar el importe automáticamente
- Fecha: usar componente *DatePicker* y comprobar que no puedan introducirse fechas posteriores al día de hoy en la bda.
- **AMPLIACION** Añadir al menos dos nuevos componentes de una librería externa. (ejemplo controlsfx, JFXtras, JFoenix... hay muchas en este [link se comentan muchas mas](#))

CARGA DE DATOS DE POLICIAS

Esta ventana permitirá seleccionar un archivo mediante FileChooser, relleno de datos de policías cuya estructura será:

- o primera fila nombre de los datos, resto de filas un policía por fila
- o todos los campos están separados por el carácter ","

Se pasará el fichero a un método de la clase *ArchivosDAO*, éste lo recorrerá y devolverá una colección de objetos Policia. Esta colección será pasada a otro método de la clase *ComisariaBD* que los insertará en la tabla policías.

AMPLIACION El método de la clase *ComisariaBD* deberá informar del número total de policías almacenados y del nombre de los que NO ha podido insertar. Esta información deberá ser mostrada de forma visual mediante al menos una ventana de Alert.

LISTADO DE MULTAS POR POLICIA

- Se debe permitir buscar por policía (placa ó nombre) y mostrar todas sus multas.
- Las multas encontradas se cargaran en un TableView.
- Se deberá calcular y mostrar el total a recaudar de las multas visualizadas
- **AMPLIACION:** Se debe permitir la selección de varios policías a la vez, y por tanto visualizar todas sus multas juntas pero ordenadas por policía.

REQUISITOS TÉCNICOS

COMPONENTES

Deben aparecer como **mínimo un control de cada tipo**: Combobox, ListView, TableView . Cargando sus datos mediante objetos de las clases Modelo

CONEXIÓN A BASE DE DATOS

La conexión debe ser creada una sola vez en la clase *ComisariaDAO*.

LOGO/ICONO

La aplicación debe tener un logo que se visualizará como icono de la aplicación en la barra de tareas.