

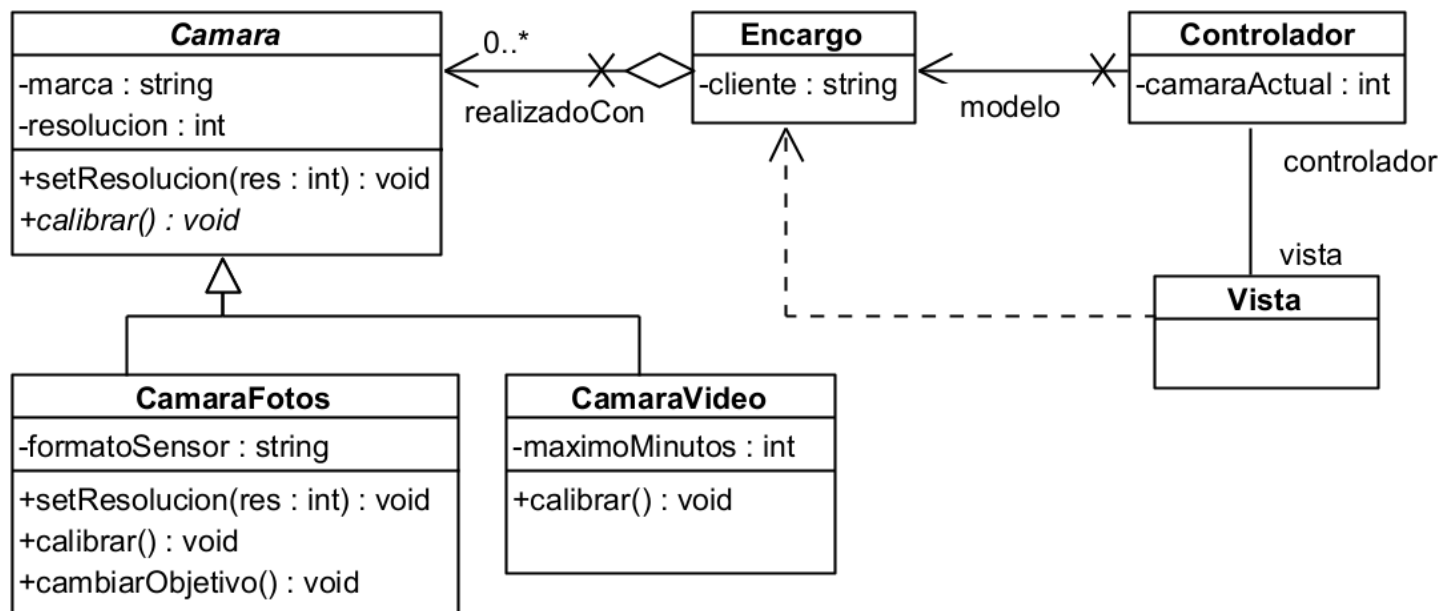
Programación y Diseño Orientado a Objetos – 2021/2022

Examen de prácticas P4, P5 – Incidencias

DNI: _____ Apellidos: _____ Nombre: _____

Descripción de la actividad

Se desea implementar el siguiente diseño UML:



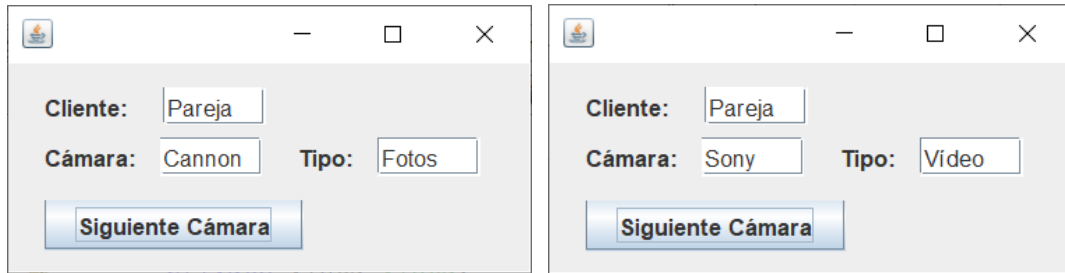
correspondiente a una aplicación para la gestión de las cámaras utilizadas al hacer encargos a un estudio fotográfico. Para ello, siga los pasos indicados en los siguientes **ejercicios**:

1.- (3.5 puntos) Implemente las clases **Encargo** (1 punto), **Camara** (1 punto), **CamaraFotos** (1 punto) y **CamaraVideo** (0.5 puntos) tal y como se indica en el diagrama, teniendo en cuenta que **Camara** y **calibrar** están en cursiva en la clase **Camara** y siguiendo las siguientes indicaciones:

- Implemente el **constructor parametrizado** de cada clase para que se inicialicen todos sus atributos.
- El método **setResolucion** en **CamaraFotos** sólo establece la resolución si el parámetro es menor o igual a 1024. El método **cambiarObjetivo** en **CamaraFotos** muestra por consola "Objetivo cambiado".
- El método **calibrar** en cada clase muestra por consola "Calibrando XX" donde XX es el nombre de la clase.
- Implemente en Encargo un método **addCamara** para añadir una cámara a un encargo.
- Implemente en Encargo un método **calibrar** para calibrar todas las cámaras de un encargo.

2.- (2 puntos) Implemente el **constructor copia** de la clase **Encargo** de manera que realice una copia profunda. Añada todos los métodos que necesite para poder implementar correctamente este constructor.

3.- (1 punto) Implemente la clase **Vista** (del tipo **JFrame**) declarando únicamente los atributos indicados en el diagrama de clases y los necesarios para los controles gráficos. El diseño gráfico de esta clase debe ser como se ve en las siguientes figuras:



The figure shows two screenshots of a Java Swing window titled "Vista". The window contains three text input fields: "Cliente:", "Cámara:", and "Tipo:". Below these fields is a button labeled "Siguiete Cámara". In the left screenshot, the "Cliente" field contains "Pareja", the "Cámara" field contains "Cannon", and the "Tipo" field contains "Fotos". In the right screenshot, the "Cliente" field contains "Pareja", the "Cámara" field contains "Sony", and the "Tipo" field contains "Video".

4.- (2.5 puntos) Implemente la clase **Controlador** y añada los métodos necesarios en las clases que crea conveniente de manera que cuando se pulse el botón **Siguiete Cámara** se actualice el atributo **camaraActual** del controlador y se muestre en la vista la información adecuada para la cámara correspondiente a ese valor. En la vista, después de "Tipo: " se mostrará "Fotos" o "Video" dependiendo del tipo de cámara.

5.- (1 punto) Complete el **programa principal** para que realice las siguientes operaciones:

- (0.2 puntos) Declare un array, de nombre **camaras**, conteniendo: 1) una cámara de fotos de sensor "SLR", marca "Cannon" y resolución 512, y 2) una cámara de vídeo con máximo de minutos 1000, marca "Sony" y resolución 2048. Cambie a 2048 la resolución (método **setResolucion**) de la cámara que se encuentre en la primera posición del array **camaras**.
- (0.2 puntos) Cambie el objetivo (método **cambiarObjetivo**) de la cámara que se encuentre en la primera posición del array **camaras**.
- (0.2 puntos) Declare un encargo, de nombre **boda** y cliente "Pareja", y añádale todas las cámaras del array **camaras**. Haga una copia del encargo anterior, en la variable **bautizo**, utilizando el constructor copia de **Encargo**.
- (0.2 puntos) Calibre (método **calibrar** de **Encargo**) todas las cámaras del encargo **bautizo**.
- (0.2 puntos) Realice las operaciones necesarias para que se **visualice la vista gráfica** y se pueda visualizar la información del encargo **bautizo** utilizando el botón **Siguiete Cámara**, como se ve en las figuras de más arriba.