

## Trabajo Práctico Grupal. Desarrollo en Java

### Año 2023 1er. cuatrimestre

#### Enunciado del problema

---

Se desea implementar un sistema que gestione **estadísticas de publicaciones** de un perfil de *Instagram*.

##### Se informa que:

La **información** que se desea administrar de todos los **tipos de publicación** es la siguiente: nombre, fecha de subida a Instagram, cantidad de “me gusta”, las etiquetas o *hashtags* contenidas y los comentarios que otros usuarios escribieron.

Para cada tipo de publicación se agregan datos y operaciones específicas.

- Para los **videos**: duración, resolución, cantidad de cuadros. Permitir avanzar, detener, y aplicar filtros.
- Para las **imágenes**: resolución, ancho, alto. Permitir aplicar filtros.
- Para los **audios**: duración, velocidad de bits. Permitir avanzar, detener.

El perfil puede poseer una estructura de agrupaciones lógicas (**álbumes**) para organizar las publicaciones en base a múltiples criterios. Para cada álbum se pretende conocer su cantidad de me gusta acumulada, y cantidad de publicaciones por tipo y totales

- Una publicación puede pertenecer a cero o más álbumes
- Un álbum puede contener cero o más sub-álbumes, por lo que se genera una estructura de árbol de álbumes.

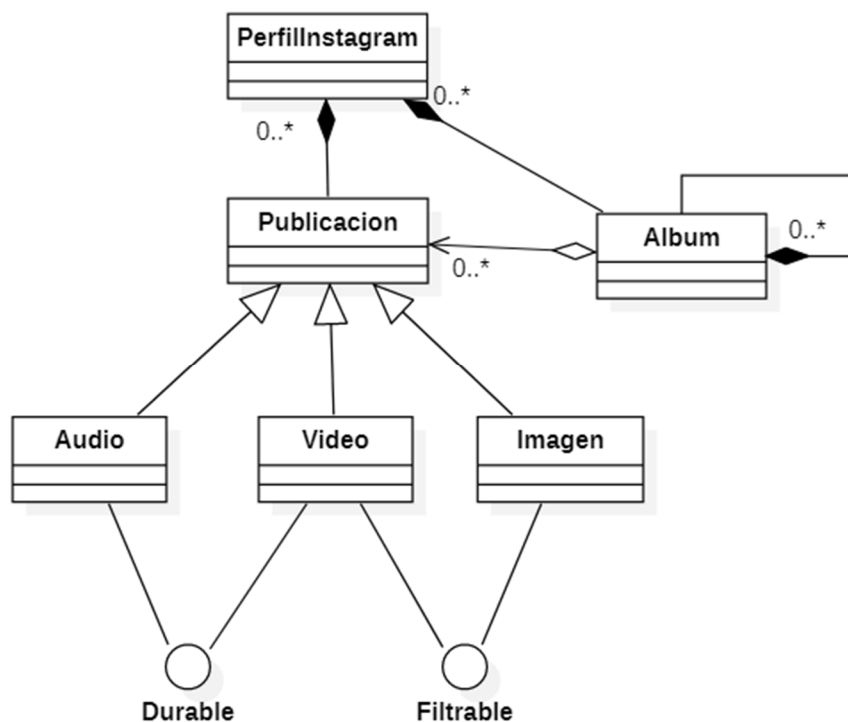
##### Se solicita que el sistema:

- Permita cargar los datos de las publicaciones desde un archivo XML, o un archivo de texto delimitado por comas u otro separador.
  - La lista de publicaciones deberá estar ordenada alfabéticamente por su nombre.
- Permita crear, gestionar y eliminar álbumes.
  - Al eliminar un álbum, deben eliminarse todos sus sub-álbumes. Las publicaciones no deben perderse, sino solamente ser sacadas de esos álbumes.
  - Se debe permitir agregar y sacar publicaciones de los álbumes.
- Permita la consulta y reproducción de un grupo de publicaciones seleccionadas de acuerdo a filtros flexibles aplicados a los atributos que se crean más relevantes. El orden de reproducción puede ser configurable de acuerdo a algún o algunos atributos.
  - Cada publicación a reproducir puede ser configurada previamente a la reproducción, aplicando algún filtro, adelantando el inicio del mismo a determinada posición, y estableciendo un punto de detención o finalización. Para las imágenes se puede parametrizar la duración de la reproducción genéricamente.
  - Antes del inicio de la reproducción se debe mostrar el tiempo total de reproducción.
  - La reproducción debe implementarse de forma simulada, mediante un listado detallado del proceso, indicando para cada publicación inicio y fin, configuraciones aplicadas.
- Genere estadísticas resumidas de las publicaciones, utilizando gráficos, tablas, histogramas, u otras herramientas que permiten ver claramente información relacionada al perfil y sus publicaciones.

- Genere los siguientes reportes (por pantalla y en archivos de texto):
  - Listado completo de publicaciones agrupadas por tipo (audio, foto, video). Para cada grupo, ordenar por cantidad de “me gusta” descendente, mostrar cantidad de publicaciones y cantidad de “me gusta” promedio de cada uno de los 3 tipos.
  - Listado alfabético de Álbumes, detallando para cada uno cantidad de publicaciones subidas en un rango de fechas solicitado al operador. Incluir la cantidad de comentarios correspondientes a esas publicaciones.
- Realice las validaciones necesarias para que la información sea consistente (ej. completitud de datos obligatorios (ej. nombre), validez de las fechas ingresadas, valores numéricos positivos donde corresponda, etc.)

### Sugerencias y comentarios

- Diagrama de clases sugerido



- La **interfaz de usuario** puede ser elegida por el grupo (caracter, gráfica (Swing), web (html + Servlets ó JSP) u otra dominada por los integrantes del grupo.
- La **persistencia** debe implementarse mediante serialización (clásica o XML)
- Considerar:
  - el uso de las clases **Containers** provistas por Java (TreeSet, ArrayList, LinkedList, TreeMap) para administrar listas y conjuntos.
  - el uso de árboles (TreeSet) para almacenar la **lista ordenada** de publicaciones.
  - el uso de clases específicas para el manejo de los **reportes**.
  - el uso de **Enum** para los valores discretos.
  - Aplicar el lanzamiento de **excepciones** en las validaciones de las clases del dominio con el objetivo de desacoplarla de la Interfaz de Usuario.
  - el uso de la interface **Filtrable** como determinante para publicaciones sobre cuyo contenido se puede aplicar filtros, y de la interface **Durable** para las publicaciones que los que se puede avanzar o detener.

## Condiciones de Aprobación Trabajo Práctico Java

- Conformar un **grupo** de 3 personas, de ser posible. Según el número de alumnos del curso se permitirán 2 grupos de 2 alumnos o un grupo de 4 alumnos.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Aplicar indefectiblemente en la solución los siguientes conceptos de la Programación orientada a Objetos: **encapsulamiento, polimorfismo, herencia**.
- Cumplir estrictamente con el **cronograma de entregas**, vía tarea del Aula Virtual, según el siguiente detalle:

	Fecha
Entrega Parcial de avance	27/04/2023
Entrega Final – 1ra. Fecha	28/05/2023
Entrega Final – 2da. Fecha	14/06/2023

- En la Entrega Parcial de avance, se deberá mostrar el avance del desarrollo del TP teniendo al menos la declaración de las clases centrales del sistema con sus atributos y cabeceras de los métodos más relevantes. La calidad de esta entrega, o no haberla realizado, será tenido en cuenta en la nota final del TP.
- En la Entrega Final:
  - presentar copia digital (por mail, tarea aula virtual, etc.) de los **archivos fuentes** (.java) y la **documentación** en **HTML** generada automáticamente con *javadoc*. Todos los integrantes del grupo serán evaluados oralmente de manera individual posteriormente a la entrega grupal. No se permitirán modificaciones posteriores a la entrega.
  - los docentes podrán proponer la implementación del sistema con un **lote de datos** propuesto por la cátedra.
  - en la evaluación oral deberán estar presentes **todos** los integrantes del grupo. El integrante que no lo estuviera se considerará fuera del grupo y deberá realizar su propio TP.
- Otros **conceptos** que incidirán en la **aprobación** del trabajo práctico son:
  - Reutilización adecuada del código.
  - Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
  - Bajo acoplamiento entre interfaz y lógica de dominio
  - Empaquetamiento criterioso de las clases.
  - Utilización de operadores, métodos y técnicas propias del lenguaje Java (ej: manejo de errores con excepciones propias, métodos `compareTo`, `toString`)
  - Validaciones de ingresos de datos y consistencia de la información.
  - Código prolijo, claro y correctamente comentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos)
  - Amigabilidad de las interfaces de usuario
- La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la evaluación oral.