

Programación Orientada a Objetos

Trabajo Práctico Especial – 2016

Defina las clases (nombre, superclase, atributos y métodos) para implementar una solución orientada a objetos para el siguiente problema e implemente en Java.

Un sistema de administración de música permite organizar nuestra colección musical en base a cierta información relevante que contienen los archivos o pistas de audio.

Cada pista de la colección posee los siguientes atributos:

- ID
- Título
- Duración (en segundos)
- Artista o Intérprete
- Título del Álbum
- Año
- Género (rock, pop, melódico, etc.)
- Comentarios

Las pistas de música se pueden agregar y eliminar de la colección en todo momento, así como también cambiar cualquiera de los atributos mencionados.

El sistema permite la creación y administración de listas de reproducción o *playlists*. Una lista de reproducción tiene un nombre que la describe, y consiste en un subconjunto ordenado de la colección. Esto incluye el caso de que una *playlist* incluya como uno de sus elementos otra *playlist*. El orden de los elementos de la *playlist* se puede modificar manualmente.

Además de la **funcionalidad** mencionada el Sistema debe proveer los siguientes servicios:

1. Búsquedas El sistema debe contar con el soporte para la búsqueda de pistas que satisfacen un determinado criterio. Los criterios de búsqueda pueden ser simples o compuestos.

Un criterio de búsqueda simple está definido por un solo atributo, por ejemplo:

- * Buscar las pistas de cuyo artista es “Indio Solari”.
- * Buscar las pistas de duración mayor a 300 segundos.

Los criterios de búsqueda compuestos resultan de combinar estos criterios con operaciones lógicas (*and*, *or*, *not*), como por ejemplo:

- * Buscar las pistas cuyo artista es “Maná” y su duración es menor a 350 segundos.
- * Buscar las pistas cuyo nombre contiene la palabra “heart” pero no contienen la palabra “love” en el título del álbum.
- * Buscar las pistas cuyo artista es “Piola Vago”, contienen la palabra “cumbia” en el título del álbum, y el año es posterior a 2005.

Aclaraciones sobre las búsquedas:

- Las búsquedas **no** deben ser “case sensitive”, es decir que si realizo una búsqueda por “fabulosos cadillacs”, las pistas que tienen como artista “Fabulosos Cadillacs”, “FABULOSOS CADILLACS” o “Fabulosos cadillacs” (y similares combinaciones) coinciden con la búsqueda y por lo tanto son parte del resultado.
- Es importante aclarar que las pistas pueden cumplir parcialmente con la búsqueda, y pertenecer igualmente al resultado. Por ejemplo, si en nuestra colección hay pistas de “John Lennon” y de “John Mayer”, una búsqueda por artista que sólo especifique la palabra “John” debe devolver como resultado todas las pistas de ambos artistas.
- Las búsquedas siempre retornan una lista de pistas, y nunca incluyen *playlists* completas. En otras palabras, si se realiza una búsqueda y existen *playlists* anidadas, las búsquedas retornan las pistas que cumplen con el criterio de búsqueda y no la *playlist* que la contiene. Por ejemplo, se tiene una *playlist* “El Indio” que contiene dos *playlists*: “Redondos” y “Solista”. Cada una de estas dos *playlists* contiene a su vez una *playlist* por cada álbum. Si se buscan los temas de más

Programación Orientada a Objetos

Trabajo Práctico Especial – 2016

de 350 segundos, se retornan todos los temas de cada una de las *playlists* que cumpla con el criterio.

2. Operaciones sobre las *playlists*

- Duración total: El sistema debe ser capaz de calcular la duración total de una *playlist* creada por el usuario, en base a la suma de las duraciones de los elementos de la misma.
- Cantidad de elementos: El sistema debe ser capaz de contar la cantidad de pistas almacenadas en la colección completa, o en una *playlist* específica.
- Copia de *playlists*: En determinadas ocasiones, se puede necesitar duplicar una *playlist* completa para poder modificarla, agregándole o quitándole pistas. En ese caso, la nueva *playlist* se crea con el nombre “Copia de NOMBRE_PLAYLIST”, donde NOMBRE_PLAYLIST es el nombre de la lista original. Este nombre puede ser modificado posteriormente. Al duplicar una *playlist*, **no** se duplican los elementos que contiene (sean pistas u otras *playlists*).
- Impresión por pantalla: Se imprime cada uno de los elementos de la misma, en el orden establecido. Cada pista se escribe con el siguiente formato: Título – Artista/Interprete –Álbum (Género, Año)

3. *Playlists* automáticas: Se trata de un caso especial de *playlist* cuyo contenido se actualiza dinámicamente (en cada acceso a la misma) con referencia a las pistas que cumplen con determinado criterio de búsqueda (simple o compuesto). Por ejemplo, una *playlist* automática puede ser creada a partir del criterio “género = melódico”. Las operaciones de la *playlist* automática deben tener en cuenta el **estado actual** de la colección (puede que se agreguen o quiten elementos que coinciden con el criterio de la *playlist*).

Consideraciones generales

- Tener en cuenta los conceptos básicos de la programación orientada a objetos.
- Inscribirse en grupos de dos integrantes mediante el formulario que figura en la página web de la materia. Se admitirán excepcionalmente grupos de tres integrantes; en este caso deben inscribirse dos de los mismos mediante el formulario web y enviar un mail a la cátedra para solicitar la inclusión del tercer integrante al grupo.
- La entrega se realiza por mail al ayudante asignado, y consta de:
 - Un informe con una breve documentación de las decisiones de diseño que han sido tomadas para resolver el problema.
 - Casos de ejemplo (con datos cargados) para probar cada uno de los servicios y funcionalidades requeridas, siguiendo la “Guía para la resolución del TP especial de POO - 2016”
 - Un archivo comprimido con las clases correspondientes a la implementación del ejercicio.