



Enunciado General

Proyecto de Curso

1. Introducción

Durante el semestre se realizará un proyecto grupal, que estará dividido en entregas parciales y una entrega final. Esta última incluirá una presentación del proyecto al curso, por parte del grupo.

1.1. Grupos

El proyecto será realizado en grupos de **3** alumnos, los cuales serán definidos por los mismos estudiantes al comenzar el semestre. Si bien la inscripción de los grupos es permanente para todo el semestre, el profesor se reserva el derecho a redefinir los grupos en pos del cumplimiento de los objetivos del curso. Por supuesto, esto es una medida absolutamente excepcional, que se espera no sea utilizada.

Para procurar una buena coordinación, cada grupo deberá definir entre sus miembros los siguientes roles:

- **Encargado de repositorio:** este alumno estará a cargo de administrar el repositorio de entrega, asegurándose que los avances realizados se encuentren debidamente subidos y compilen correctamente¹
- **Encargado de trabajo interno:** este alumno será el encargado de registrar la división de tareas asignadas para cada entrega, y asegurar el cumplimiento de estas por parte de sus compañeros.
- **Encargado de relaciones exteriores:** este alumno estará a cargo de las relaciones con el equipo docente, será quien publique las dudas en el foro de SAF y envíe mails por consultas particulares. También será quien reciba las noticias referentes al grupo, en caso de que el equipo docente necesite informarles.

1.2. Tutor

Los ayudantes de proyecto del curso ejercerán la labor de **tutor** de los grupos. El trabajo del **tutor** será resolver las dudas que surjan en el grupo respecto a los contenidos del curso, y ayudarlos en el seguimiento y desarrollo del proyecto. En particular, el **tutor** pondrá especial énfasis en temas de organización y diseño de la solución.

¹En aquellas entregas que además (o en vez) de código se requiera documentación, este encargado debe asegurarse que el formato de entrega sea el solicitado



El **tutor**, además, se preocupará de velar por una correcta distribución del trabajo dentro de los grupos, pudiendo redefinir la labor de cada integrante para una entrega si detecta imparcialidades en este ámbito.

Los grupos podrán reunirse con el **tutor** en las horas de taller de proyecto establecidas en el calendario del curso, que por lo general serán unos días antes de la entrega. En estas reuniones se revisará el avance y el cumplimiento de las tareas definidas para cada integrante, además de ser una instancia para resolver las dudas técnicas que tengan².

Por supuesto el **tutor** participará activamente en los foros del curso, y además el encargado de comunicaciones exteriores del grupo podrá contactarlo a través de su correo electrónico. Les recomendamos aprovechar estas instancias, ya que el **tutor** está para resolverles sus dudas y apoyarlos en su proceso de aprendizaje. Es importante notar que, a diferencia de lo que ocurre en una ayudantía de cátedra común, en este modelo se espera una actitud pro-activa de parte de los grupos.

2. Descripción del Proyecto

Todos los grupos deberán desarrollar un programa de reproducción multimedia, que será una mezcla similar a Netflix con Spotify.

2.1. Descripción general

En términos generales, se le pedirá implementar un software de reproducción multimedia con características comparables a aquellas presentes en las aplicaciones actuales. Esto implica que su aplicación deberá ser capaz de manejar un sin número de canciones y videos, y realizar diversas operaciones con ellas.

Lo principal de esta aplicación es que es un repositorio de canciones y videos que debe permitir organizarlos y encontrarlos de manera fácil e intuitiva. Dado lo anterior, es necesario que los archivos multimedia cuenten con información adicional, tales como, género, categorías, autores, actores, directores, estudio, entre otros. De esta manera los archivos pueden ser organizados y agrupados según una o múltiples características. En otras palabras, esto debe permitir búsquedas simples o complejas, por ejemplo, buscar todas las películas donde aparece un determinado actor o encontrar todas las canciones donde el cantante y compositor son mujeres. Además, debe ser posible asignar un “ranking” (o valor de calidad) a cada canción o video, para poder después buscar por los mejores.

Otras características que debe considerar son los datos intrínsecos de los archivos multimedia, es decir, aquellos que no necesitan ser agregados manualmente. Por ejemplo, la duración, dimensión del video o de las imágenes agregadas, calidad de canción o video, tamaño en memoria, entre otros.

Todas las características mencionadas anteriormente son a modo de ejemplo, es decir, queda abierto y es necesario que ustedes definan cuáles agregarán.

²Es importante que lleguen a los talleres con un avance considerable de la entrega, para que sean sesiones de trabajo provechosas.



Por supuesto que el realizar búsquedas es solo un primer (e importante) paso, pero a la vez la aplicación debe ser capaz de realizar otras acciones, como por ejemplo permitir construir **playlists** con canciones/vídeos seleccionados por el usuario.

Finalmente, debe considerar (al igual que las aplicaciones reales) que deben existir distintos tipos de usuarios, con características diferentes, por ejemplo, un administrador, un usuario gratuito y uno pagado, y cualquier otro que ustedes estimen necesario.

La creatividad e innovación fundamentada será premiada, es decir, innovar y ser creativos tendrá puntaje siempre y cuando esté argumentado.

2.2. Descripción Específica

A continuación detallaremos cada una de las funcionalidades mínimas que debe ser capaz de realizar su aplicación:

1. Importar canciones y vídeos → Agregar nuevos archivos multimedia de forma sencilla. Soportar al menos los formatos más comunes (mp3, wav, mp4, etc).
2. Información de las canciones (y vídeos)
 - a) Metadata → Cada canción debe tener información como el género, cantante, compositor, discografía, estudio, año de publicación, letra, etc. Por otra parte, los vídeos deben contener información como género, categoría, actores, director, estudio, año de publicación, descripción, calificación, etc.
 - b) Metadata al importar → Al cargar canciones (y vídeos) agregar información para todas las canciones que se importen.
 - c) Información intrínseca de cada canción (y vídeo) → Información que puede ser obtenida como resultado de operaciones simples como la duración, etc.
 - d) Información en la plataforma → Cada canción y vídeo debe tener información que se genere en la plataforma, por ejemplo, número de reproducciones, cantidad de usuarios que les gusta, calificación (nota que ponen los usuarios), etc.
3. Búsquedas → buscar canciones (y videos) definiendo distintos filtros.
 - a) Filtros para búsqueda → cada filtro es una combinación entre alguna propiedad y uno o más valores determinados.
 - 1) Por palabras clave → definir una o más palabras que son buscadas entre la información de las canciones (y videos).
 - 2) Por persona → definir cuales personas deben estar (autores/actores) en la canción (o video).
 - 3) Por características de personas → definir valores para las distintas características de personas que deben estar en la canción (o video) (sexo, edad, ...).



- 4) Por resolución \rightarrow mayor, menor o igual a algún tamaño.
 - 5) Por evaluación \rightarrow mayor, menor o igual a alguna nota definida.
 - 6) Por categoría
 - b) Múltiples filtros a la vez \rightarrow para cada búsqueda se deben poder combinar los filtros utilizados. La combinación de los filtros puede ser tanto como un *and*, es decir, se buscan aquellas que cumplan con todos los filtros, o como un *or*, se buscan las canciones o videos que cumplan con al menos un filtro. Para los múltiples filtros se pueden combinar el uso de *and* y *or*.
 - c) Listas inteligentes \rightarrow poder almacenar una búsqueda arbitraria como una lista, de manera que si en el futuro se agregan a la biblioteca de canciones o videos algunas que cumplen con el criterio de búsqueda, se muestren automáticamente como parte de la lista.
4. Playlists
- a) Playlist \rightarrow crear playlists con canciones (o videos). Un playlist no puede tener canciones y videos, los archivos multimedia deben ser el mismo tipo. Estas pueden ser privadas o públicas. Si un usuario es privado, no puede tener playlist públicas.
- 5. En cola \rightarrow Se debe poder crear una cola de reproducción automática (lista que se reproduzca automáticamente) tanto para video como para canciones
 - 6. Playlist específica de favoritos \rightarrow Se debe poder agregar canciones y videos a una lista de favoritos. Canciones y videos deben estar en listas diferentes.
 - 7. Buscar y seguir: Usuarios, playlist, discos, cantantes, actores, etc.
 - 8. Agregar imágenes, por ejemplo, una imagen para representar un disco o un película.
 - 9. Administración de usuarios \rightarrow Crear perfiles de usuarios (con información y gustos por ejemplo), modificarlos, definir si son privados o públicos, etc.
 - 10. Diferenciar usuarios pagos de no pagos: Esto es abierto, pero una buena base es ver como lo hace Spotify.
 - 11. Las canciones pueden o no ser descargadas, pero no las películas.
 - 12. Si el usuario cierra la aplicación y en dicho momento estaba reproduciendo algún archivo, entonces se debe guardar el tiempo de reproducción para que cuando vuelva a abrirla pueda continuar desde ahí.

Recordar que estas son las funcionalidades mínimas exigidas, pero se espera que ustedes realicen la aplicación de la manera más completa posible



3. Evaluación

Para cada entrega se especificará los ítemes a evaluar. Por ser éste un curso de programación **avanzada**, no sólo se evaluará el cumplimiento de lo pedido, sino que también se evaluará la calidad de su solución, ya que se espera que ustedes apliquen lo aprendido de la mejor forma posible.

Además, para el diseño de su interfaz de usuario deberán poner mucha atención en la usabilidad de su software. Esto significa que su software debe ser lo más simple e intuitivo de usar posible. Se espera que ustedes realicen soluciones creativas a los problemas que les vayan surgiendo, las cuales permitan al usuario tener una mejor interacción con su programa. Recuerde que no siempre las soluciones disponibles en los programas comerciales son las mejores, por lo que antes de copiarlas, piense si existe una mejor forma de solucionar el problema.

En cada entrega deberán completar una evaluación par a sus compañeros de grupo con el fin de evaluar el trabajo que realizó cada uno. A la vez, cada entrega tendrá una nota grupal y una individual.

Por último, la nota final será de acuerdo al trabajo de todos los grupos, esto quiere decir que el mejor o mejores definirán la exigencia para la nota máxima.

3.1. Descripción General

Las entregas estarán divididas de la siguiente manera:

- Entrega 1 (21/04/2020): se deberán modelar todas las clases de la aplicación y las relaciones entre ellas.
- Entrega 2 (12/05/2020): se deberá programar la aplicación para que funcione solamente en consola.
- Entrega 3 (02/06/2020): se deberá programar la aplicación para que funcione con una interfaz gráfica.
- Entrega 4 (16/06/2020, 18/06/2020 y 21/06/2020): se deberá terminar de programar todos los detalles de la aplicación y además hacer una presentación en que se mostrará su funcionamiento y como fue desarrollada.

La interfaz del software quedará a criterio de los alumnos, pero deberá ser relativamente amigable e intuitiva para un usuario.