

# Proyecto Bases de Datos 2011

Durante el corriente año se va a implementar, como prueba piloto, un único proyecto integrador de las tres materias de computación del tercer año de las carreras de computación. El proyecto se desarrollará sobre un único dominio, aportando cada materia a la realización del mismo. A continuación se presenta una narrativa donde se describe el dominio.

## Narrativa

**Plarpebu** es un reproductor de archivos de música, que permite reproducir temas musicales a partir de una lista que puede ser creada y modificada por el usuario. Las canciones que forman esta lista deben estar almacenadas en algún dispositivo como pen drive; cd o disco rígido, etc.

Plarpebu posee un menú llamado **Plugins** que permite activar o desactivar las funcionalidades del reproductor. Dentro de ellas podemos mencionar el Karaoke el cual muestra un visualizador en donde la aplicación reproducirá la letra del tema que se esté escuchando (si es que el archivo la incluye).

El reproductor cuenta con un ecualizador que nos permitirá ajustar el sonido a nuestro gusto para lograr la mejor experiencia auditiva. Para adaptarse a los gustos de los diferentes usuarios, **Plarpebu** ofrece una funcionalidad singular llamada “skins” que permite cambiar el “look and feel” de la aplicación, seleccionando entre una serie de configuraciones preestablecidas y pudiendo además personalizarlas.

Algunos plugins no están implementados aún.

El plugin más interesante es el de la “PlayList” que permite crear y cargar las listas de reproducción de archivos de audio a partir de archivos de texto, en donde se almacena la lista de los archivos de audio a reproducir junto con su PATH. Cabe mencionar que la única información persistente que manipula esta aplicación es la almacenada en dichas listas de reproducción.

El trabajo consiste en diseñar una base de datos relacional que permita almacenar información persistente. La base de datos debe poder almacenar las listas de reproducción y además poder agrupar las listas creadas en carpetas y subcarpetas. Se debe almacenar la siguiente información sobre los temas: título, artista, álbum, año y un género en particular (pop, jazz, latino, etc.). Cada tema en una lista tiene un Orden determinado.

También es necesario que la base de datos soporte toda la configuración( *look and feel, ecualizador, etc*) de la herramienta, de esta forma la herramienta podrá ser personalizada, con lo cual no se perderá cuando se cierra la aplicación. Además cada usuario debe poder crear sus propios perfiles. Es necesario incorporar el manejo de usuarios que acceden al sistema, no por una cuestión de seguridad, sino para el manejo de perfiles. Cada usuario tiene definido el perfil por defecto con el cual iniciará la aplicación.

El ecualizador se compone de un conjunto de bandas (en nuestro caso son 32), las cuales corresponden a distintas frecuencias. Las frecuencias soportadas por nuestro plugin de EQ son: Banda1,..., Banda32. Cada banda, corresponde a una frecuencia en particular, se puede ajustar en el rango de -12db. a 12db. (decibeles), por defecto el ecualizador es plano, todas las bandas están ajustada en 0db.

Cada tema, de acuerdo a la lista de reproducción, puede tener una ecualización por defecto.

### **Consideración a tener en cuenta en la implementación de la base de datos en un motor de base de datos:**

- Utilice códigos cuando lo crea conveniente.
- Los diferentes códigos deben ser generados automáticamente.
- No puede haber dos temas con el mismo nombre en una lista de reproducción.

- Dentro de una carpeta no pueden existir dos listas con el mismo nombre.
- La implementación de la base de datos deberá permitir generar información de auditoría automáticamente. Se deberá agregar información en una tabla sobre los eliminación de las listas de reproducción, esta información deberá contener la lista eliminada, fecha de eliminación y el usuario que la realizó.

#### Tareas a Realizar:

1. Diseñar el diagrama de Entidades y Relaciones.
2. Realizar el pasaje del modelo E-R a Relacional (con claves foráneas).

Utilizando el lenguaje SQL:

3. Definir el diccionario de datos creando:
  - La base de datos.
  - Las tablas componentes con todas las restricciones que el problema requiere (claves primarias, claves foráneas indicando que hacer al borrar o actualizar, etc.). Implementar las restricciones de tipos.
4. Generar un script SQL para la carga de información en la base de datos (archivo de texto con el código SQL para la inserción de datos).
5. Realizar un Programa Java que permita:
  - Insertar temas en una lista.
  - Eliminar temas de una lista.
  - Listar todas las listas que hay en una carpeta determinada.
6. Resolver las siguientes consultas:
  - a) Devolver un listado con las listas de reproducción y la cantidad de temas que poseen.
  - b) Devolver la lista de temas que tenga más tiempo de reproducción (sumando la duración de los temas incluidos).
  - c) Definir consultas propias (no menos de dos), donde por lo menos una utilice subconsultas.

#### Características del proyecto

- La base de datos debe ser implementada en algunos de los siguientes motores de base de datos: MySQL versión 5.0, Firebird 2.0(o superior), Postgres 8.3(o superior) o el motor que el grupo seleccione con acuerdo de la cátedra.
- Los grupos de trabajo estarán conformados por 3 alumnos.

#### Evaluación:

- La evaluación constará de una corrección teórica y una prueba práctica de funcionamiento.
- En todas las entregas se evaluará el informe presentado.

#### **Se deberá cumplir con el siguiente cronograma de entrega:**

- 27/05/2011: Puntos 1, 2. **Horario 9:30hs.**
- 10/06/2011: Puntos 3 y 4. Además del informe, entregar un CD o Pen conteniendo la base de datos con datos cargados (archivo de la base de datos). **Horario 09:30hs.**
- 15/06/2011: Punto 5 y 6. Además del informe, en un CD o Pen conteniendo un archivo .txt con las consultas realizadas, el archivo con la base de datos y el programa java. **Horario a confirmar.**