

## Capítulo IV: Pruebas y Conclusiones

En este capítulo se describen las distintas pruebas que se realizaron para comprobar la robustez de nuestro sistema, y se describen los resultados obtenidos de dichas pruebas. También se describe el trabajo a futuro por realizar, es decir, el trabajo que se puede continuar dentro de ésta misma línea de investigación. Finalmente, pero sin restar importancia, se describen las distintas conclusiones a las que se llegaron durante la realización de éste trabajo de tesis.

### 4.1 Pruebas

Uno de los fines más importantes de nuestro sistema es que cuente con un alto grado de robustez, es decir, que sea altamente tolerante a fallas. Realmente nos interesa que los usuarios puedan utilizar nuestro sistema sin tenerse que preocupar por las fallas que éste pueda producir. Es de gran importancia contar con un sistema de recuperación a fallas, es decir, en caso de que el sistema falle por alguna razón, es importante que éste cuente con la capacidad de recuperarse sin que se caiga por completo y deje de funcionar. Es importante también prevenir dichas fallas para poder identificarlas al momento que se produzcan, y comunicarle al usuario el tipo de falla que se produjo de una manera amigable para que el dicho usuario pueda entender el estado en el que se encuentra el sistema. No es de interés para el usuario recibir como respuesta a una petición, una página la cual describe la ruta completa producida por alguna falla, o sea, la página que describe el árbol completo de alguna excepción de Java atrapada por el sistema. Nos interesa que en caso de que algún error se produzca, el usuario pueda recibir un mensaje el cual describa dicho error y sugiera alguna solución; en el peor de los casos, nos interesa que el usuario reciba un mensaje de error de la manera más amigable posible sin que nuestro sistema se caiga por completo.

Para aumentar el grado de robustez de nuestro sistema, nos hemos apoyado en las distintas herramientas y funciones que el software *Tomcat* nos ofrece. Dichas funciones y herramientas implementadas, hacen uso de las mismas excepciones de Java para poder identificar el error y manejarlo dependiendo la configuración de respuesta para cada error. *Tomcat* nos da la capacidad de configurar una respuesta distinta (documento

*HTML*) para cada tipo de error distinto que se presente. Esto también se aplica para los códigos de errores dentro del protocolo *HTTP*, es decir, dependiendo el código de error que genere una cierta petición, (404 - archivo no encontrado, 500 - error interno del servidor, etc.) es la página que se le envía de regreso al usuario.

Realmente, nuestro sistema y el software *Tomcat* trabajan en conjunto para brindar un alto grado de robustez al sistema.

### 4.1.1 Tipos de Pruebas

Las pruebas que se realizaron, se enfocaron a cada una de las funciones que se pueden realizar dentro del sistema. Evidentemente, cada tipo de usuario puede realizar distintas operaciones. Se realizaron pruebas para cada una de las operaciones de cada uno de los usuarios. Para cada operación, se realizaron dos tipos de pruebas:

- **Caso Optimo**

Caso dentro del cual todos los parámetros son introducidos correctamente, donde se envían los archivos con el formato correcto. Es el caso ideal donde suponemos que todo funciona a la perfección y que el usuario no comete ningún error.

- **Caso Crítico**

Es el caso donde se introducen incorrectamente los parámetros o que se dejen en blanco ciertos campos por llenar, y se envían archivos con el formato incorrecto. Es aquí donde suponemos que los errores pueden ocurrir y dónde realmente podemos probar la robustez de nuestro sistema.

Para facilitar la exposición de las pruebas realizadas, hemos decidido agrupar los distintos casos de uso de nuestro sistema en conjuntos comunes. Tenemos tres grandes conjuntos: llenado de campos de texto, subir archivos al servidor (imágenes y audio), y las búsquedas.

### **4.1.2 Pruebas Sobre Campos de Texto**

En este conjunto, abarcamos todas las situaciones donde cualquiera de los usuarios (administrador o proveedor) deben de llenar algunos campos de texto con cierta información. Los casos que requieren llenado de campos de texto son: agregar un usuario al sistema, agregar un artículo, modificar el perfil del proveedor, modificar la descripción del proveedor y al confirmar los atributos del archivo de audio al ser subido al servidor.

Evidentemente el caso óptimo es cuando todos los campos de texto son llenados correctamente. De ser así, el usuario recibe por parte del sistema una verificación de que los datos fueron admitidos correctamente por el sistema y no se reporta ningún tipo de error.

En caso de que no se llenen todos los campos de texto necesarios o que algunos no cuenten con el formato necesario, el sistema le regresa al usuario la forma previamente llenada con los datos introducidos y con un mensaje indicándole el error. Esto se repite hasta que el usuario ingrese todos los datos con el formato correcto.

### **4.1.3 Pruebas al Subir Archivos al Servidor**

Dentro de éste conjunto, se encuentran todas las situaciones donde cualquiera de los usuarios (administrador o proveedor) deben de subir algún tipo de archivo al servidor; ya sea de imagen o de audio, dependiendo el caso. Nuestro sistema solamente maneja archivos de audio MP3, pero admite diferentes formatos de imágenes: GIF, JPG, JPE, TIF, BMP y PNG.

Los casos donde los usuarios deben de subir archivos al servidor son: al agregar un nuevo archivo de audio MP3 al sistema, al agregar un artículo y al modificar la descripción del proveedor.

El caso óptimo para este conjunto, es cuando el usuario sube al servidor el archivo con el formato correcto, en éste caso, el sistema le indica al usuario que dicho archivo ya se encuentra en el sistema y que no se registró ningún problema.

En caso de que el archivo por subir no tenga el formato correcto, el sistema le regresa al usuario la página con la forma para subir el archivo junto con un mensaje indicándole el error.

#### **4.1.4 Pruebas Sobre el Buscador**

En este conjunto, abarcamos todas las situaciones posibles que se puedan presentar dentro del rango de las búsquedas de archivos de audio de nuestro sistema. Como se describió anteriormente, se pueden realizar cuatro tipos de búsquedas: sobre el título de la canción, sobre el artista que interpreta dicha canción, sobre el álbum al que tal canción pertenece y sobre los tres atributos anteriores. Estas pruebas se realizaron en base al tiempo de ejecución del buscador, es decir, se analizó el tiempo en el que el sistema realiza los cuatro tipos de búsquedas. Para realizar dichas mediciones, se dividió el proceso de búsquedas en tres partes: el tiempo de búsqueda sobre el índice, el tiempo de recuperación de información sobre la base de datos relacional y el tiempo que tarda el sistema en ordenar los resultados dependiendo su grado de relevancia. Una vez obtenidos los tiempos por separado, cada uno de ellos fue comparado contra el tiempo total de la búsqueda, de ésta manera, obtuvimos un porcentaje representativo de cada una de las tres partes en comparación con el tiempo total, y así pudimos definir cual de los tres procesos es el más pesado, el medianamente pesado y el menos pesado. Las pruebas se realizaron con aproximadamente 1000 archivos de audio registrados dentro del sistema. Todos los tiempos presentados están dados en milisegundos, la tabla **4.1** representa los tiempos de búsqueda sobre los tres atributos: título, artista y álbum.

Tabla 4.1. Pruebas de tiempos de corridas del buscador sobre los tres atributos.

Prueba	Consulta	Tipo	Índice (ms)	BD (ms)	Ordenamiento (ms)	Total (ms)
1	pink floyd	all	1693	1001	160	2854
2	bob	all	521	520	10	1031
3	au	all	1101	471	10	1582
4	track	all	351	170	20	541
5	go	all	440	441	0	881

<b>Promedio</b>	821.2	520.6	40	1377.8
<b>Porcentaje</b>	<b>59.60226448</b>	<b>37.78487444</b>	<b>2.903178981</b>	<b>100</b>

Para los cuatro tipos de búsqueda se realizaron cinco pruebas. La primera columna indica el número de la prueba, la segunda indica la consulta introducida al sistema, la tercera indica el tipo de búsqueda, la cuarta indica el tiempo (en milisegundos) en el que se realizó la búsqueda sobre el índice, la quinta indica el tiempo en el que el sistema recuperó la información de los archivos resultado de la base de datos relacional, la sexta columna indica el tiempo en el que el sistema ordenó los resultados y la última columna indica el tiempo total de la búsqueda. Abajo se indica el promedio (en milisegundos) de cada una de las tres partes de la búsqueda, junto con el promedio del tiempo total, y el porcentaje que cada una de las tres parte representa en comparación con el tiempo total. Como podemos observar, la búsqueda sobre el índice es el proceso más tardado mientras que el ordenamiento de los resultados es el menos tardado y la recuperación de la información de la base de datos relacional es el proceso medianamente pesado. Las tablas 4.2, 4.3, 4.4, representa los tiempos de búsqueda sobre cada uno de los tres atributos por separado.

Tabla 4.2. Pruebas de tiempos de corridas del buscador el nombre del artista.

Prueba	Consulta	Tipo	Índice (ms)	BD (ms)	Ordenamiento (ms)	Total (ms)
1	ai	artist	891	941	0	1832
2	pink	artist	2383	1582	20	3995
3	dav	artist	491	220	20	731
4	j	artist	781	461	10	1252
5	peter gabriel	artist	821	952	10	1783

<b>Promedio</b>	1073.4	831.2	12	1918.6
<b>Porcentaje</b>	55.94704472	43.32325654	0.625456062	100

Tabla 4.3. Pruebas de tiempos de corridas del buscador sobre el título de la canción.

Prueba	Consulta	Tipo	Índice (ms)	BD (ms)	Ordenamiento (ms)	Total (ms)
1	global underground	title	511	170	0	681
2	wall	title	210	80	0	290
3	da	title	581	310	10	901
4	we	title	300	160	10	470
5	sky	title	220	80	0	300

<b>Promedio</b>	364.4	160	4	528.4
<b>Porcentaje</b>	68.96290689	30.28009084	0.757002271	100

Tabla 4.4. Pruebas de tiempos de corridas del buscador sobre el nombre del álbum.

Prueba	Consulta	Tipo	Índice (ms)	BD (ms)	Ordenamiento (ms)	Total (ms)
1	of the	album	651	701	10	1362
2	get	album	190	60	0	250
3	of	album	1402	1032	10	2444
4	the	album	2744	2073	100	4917
5	ea	album	240	120	0	360

<b>Promedio</b>	1045.4	797.2	24	1866.6
<b>Porcentaje</b>	56.00557163	42.70866817	1.285760206	100

Finalmente, la tabla 4.5 nos muestra los resultados finales de las distintas pruebas. Como podemos observar, la búsqueda sobre el índice abarca aproximadamente el 58% del tiempo total de la búsqueda, la recuperación de información de la base de datos relacional abarca aproximadamente el 40% y el ordenamiento de los resultados abarca apenas el 1.4% del tiempo total. Entonces podemos concluir que el proceso más pesado, es el de realizar las búsquedas. Estos resultados comprueban lo esperado. El proceso de búsqueda había sido considerado como el más pesado ya que el motor de búsqueda debe realizar las consultas sobre el índice completo, y la recuperación de la información de los resultados está limitada solamente a los atributos de los archivos obtenidos en los resultados de la consulta.

Se había considerado que el proceso de ordenamiento de los resultados sería más pesado de lo que en realidad es, ya que éste solo representa un 1.4% del tiempo total, eso indica que el algoritmo adoptado para dicho proceso es bastante bueno.

Tabla 4.5. Estadísticas finales de las pruebas realizadas sobre el buscador.

	<b>Índice (ms)</b>	<b>BD (ms)</b>	<b>Ordenamiento (ms)</b>	<b>Total (ms)</b>
<b>Promedio Total</b>	826.1	577.25	20	1422.85
<b>Porcentaje Final</b>	<b>58.0595284</b>	<b>40.5699828</b>	<b>1.405629546</b>	<b>100</b>

#### 4.1.5 Pruebas Sobre la Base de Datos, el Índice y el Servidor Web

Finalmente, una de las pruebas más importante que se realizaron, fue simular la situación en la que la base de datos relacional, el índice o el servidor web no funcionaran, o sea, que cualquiera de los tres se encontrara dañado o simplemente apagado temporalmente. Se realizó una prueba para cada caso: base de datos relacional abajo, servidor web abajo e índice abajo. En la situación de que el servidor web se encontrara abajo, el navegador web se encarga de reportar dicho error al usuario.

En el caso de que la base de datos relacional se encuentre suspendida temporalmente, el sistema le envía al usuario un mensaje de error interno en el servidor, sugiriéndole al usuario que intente realizar su petición más tarde, pero en ningún momento el sistema cae por completo.

Al igual que con la base de datos relacional, al bajar el administrador de los índices, el usuario recibe el mismo mensaje de error y la misma sugerencia sin que el sistema se caiga por completo.

## 4.2 Trabajo a Futuro

Este trabajo de tesis tomó alrededor de un año en realizarse. Evidentemente existen muchas cosas que se dejaron afuera por falta de tiempo o de un equipo de trabajo para realizarse. En esta sección mencionaremos cuáles son algunas de las funciones más importantes que no se pudieron implementar dentro de ésta tesis y quizás en un futuro puedan ser implementadas. A continuación una descripción breve de dichas funciones:

- **Búsqueda a Texto Abierto Sobre las Letras de las Canciones**

Dentro del marco teórico de este documento de tesis, se describieron las etiquetas ID3 las cuales pueden contener hasta 256 MB de información dentro de ellas. Al contar con gran espacio de almacenamiento, es posible almacenar la letra completa de dicha canción dentro de una de las etiquetas ID3. Una de las funciones que no se pudieron implementar, fue introducir la letra completa de cada uno de las canciones dentro de sus respectivos archivos, y realizar búsquedas tipo texto abierto sobre ellas. Es decir, que el usuario podría realizar búsquedas no solo sobre el nombre de la canción o el nombre del título del artista, sino también sobre la letra de las distintas canciones que se encuentran registradas en nuestro sistema. De esta manera, quizá el usuario no conoce el nombre de la canción o el artista que la interpreta, pero quizá sí conoce parte de la letra de dicha canción; y de ésta manera, el sistema proporcionaría una función para realizar búsquedas introduciendo una frase o palabra contenida dentro de la letra de alguna canción.



- **Soportar Varios Formatos de Archivos de Audio**

Como se explicó anteriormente, nuestro sistema solamente maneja archivos de audio MP3. Dicho sistema, se podría extender de manera que soportara varios formatos distintos de archivos de audio al mismo tiempo.

- **Implementar un Reproductor de Archivos de Audio**

Una de las funciones más interesantes a desarrollar, sería un reproductor de archivos de audio que pueda reproducir los archivos contenidos dentro de la colección sin necesidad de bajarlos, es decir, que el usuario pueda reproducir algún archivo de audio que desee directamente en el portal sin necesidad de bajar dicho archivo y reproducirlo localmente.

- **Mostrar Lista de Proveedores con sus Atributos**

Quizá una función importante que no se incluyó dentro del sistema, es el generar una lista de todos los proveedores con sus atributos principales para que el administrador del sistema pueda contactarlos o realizar cualquier tarea que dependa de dichos datos.

- **Dar de Baja a Proveedores**

Esta es quizá una de las funciones más críticas que no se incluyeron dentro del sistema. Que sucedería si alguno de los proveedores decidiera retirarse por completo del servicio que nuestro sistema provee. En ese caso, se necesitaría dar de baja del sistema todos los archivos de audio que dicho proveedor agregó, así como dar de baja total a dicho usuario. Nuestro sistema hoy en día no provee de dicha opción que el administrador necesita realizar.

- **Dar de Baja Archivos de Audio**

Parte de la función anteriormente mencionada, es la capacidad de que cada proveedor pueda dar de baja voluntariamente cualquiera de los archivos de audio que dicho proveedor haya agregado al sistema. De ser así, el índice y la base de datos relacional tendrían que ser actualizadas automáticamente.

### 4.3 Conclusiones

Como elemento final de éste documento de tesis, mencionaremos las conclusiones principales a las que se llegaron durante el desarrollo de dicho documento. Las conclusiones que a continuación mencionaremos, son una combinación del análisis teórico y de la experiencia práctica adquirida durante todo el desarrollo de éste trabajo de tesis. No se introducirá ninguna idea nueva sobre los distintos temas previamente estudiados, más bien, se aterrizarán los conceptos y experiencias como una sola idea la cual abarca y sintetiza los puntos más importantes de nuestro trabajo.

- Los archivos de audio MP3 son un formato de compresión de audio de alta fidelidad los cuales cuentan con una serie de características que los hacen únicos en su clase. Estos archivos se encuentran fragmentados en miles de pequeños bloques los cuales facilitan que cualquier sistema pueda comenzar a decodificarlos (reproducirlos) en cualquier momento de su transmisión. Es decir, si alguno de estos archivos se está transmitiendo al aire o por Internet, no es necesario esperar a que termine la transmisión de dicho archivo y esperar a que comience uno nuevo para poder ser escuchados. Durante cualquier momento de la transmisión, cualquier reproductor puede comenzar a reproducir dicho archivo ya que al comienzo de cada uno de los bloques contenidos dentro de los archivos MP3 encontramos un encabezado el cual le indica al reproductor toda la información que necesite para reproducir dicho fragmento de audio contenido dentro del archivo.

Las etiquetas ID3 son una de las características más importantes, ya que a través del uso de los metadatos podemos introducir información referente a los atributos de dicho archivo. Es posible introducir hasta 250 MB de información dentro de dichas etiquetas; algunos de los atributos que se pueden utilizar son: el título del archivo de audio, el artista intérprete, el nombre del álbum al que pertenece, la letra completa de dicha canción, la imagen del álbum, entre otras; esto nos ayuda a poder realizar búsquedas sobre un tipo de dato que hasta hace poco fue considerado como cien por ciento no estructurado.

- La utilización de patrones de diseño, mejora considerablemente el diseño de cualquier aplicación que nosotros deseemos desarrollar. En específico, el patrón de diseño MVC es, a nuestra manera de ver, el patrón de diseño que mejor encaja al desarrollar aplicaciones web. Nos ayuda a delimitar cada parte de nuestro sistema dándole un alto grado de reutilización a nuestras clases.
- La combinación de metadatos e índices, es una de las herramientas más poderosas en lo que a métodos de búsquedas se refiere. Especialmente cuando sobre lo que deseamos buscar son datos no estructurados. Los metadatos nos ayudan a definir atributos sobre los cuales realizaremos las búsquedas y de ésta manera reduciremos considerablemente el dominio y el rango de nuestras consultas.

Los índices reducen altamente el tiempo de búsqueda, esto debido a que no se tiene que recorrer la colección completa de datos para obtener resultados. Los índices representan una estructura y organización de los datos sobre los que deseamos realizar consultas, de tal manera que nuestro motor de búsquedas sabrá perfectamente donde buscar y omitirá el resto de la información.

- Existen varias técnicas y modelos de recuperación de información. Cada uno diseñado para ser utilizado en situaciones específicas. Pero entonces ¿por qué limitarnos a una sola técnica? En nuestro caso, hemos combinado varios modelos y técnicas para desarrollar un sistema de recuperación de información sumamente rápido y eficaz. Hemos tomado lo mejor de cada uno de los modelos y los hemos enfocado a tareas específicas mejorando su desempeño.
- Consideramos que el uso de las distintas técnicas y herramientas de análisis y diseño de sistemas es de vital importancia al desarrollar cualquier sistema. Es de suma importancia contar con un buen análisis de requerimientos de manera que podamos definir y delimitar cada una de las funciones de nuestro sistema. Esto nos ayudará a diseñar un sistema mucho más limpio y eficiente.