

## **Capítulo V. Pruebas de Software**

Este capítulo es un reporte de las pruebas realizadas al software de este proyecto de tesis. En primer lugar se presenta un reporte de las pruebas formativas de usabilidad a las que se sometió la interfaz de usuario del sistema, y en segundo lugar se presenta un reporte de las pruebas finales de funcionalidad realizadas.

### **5.1. Pruebas formativas de usabilidad**

#### **5.1.1. Planeación de pruebas**

##### ***5.1.1.1. Propósito de la prueba***

El propósito de realizar pruebas formativas de usabilidad al software de este proyecto fue el de obtener retroalimentación sobre la experiencia de uso del sistema por parte de potenciales usuarios finales, y con ella encontrar posibles áreas de mejora de las funcionalidades y la interfaz gráfica de usuario.

##### ***5.1.1.2. Lugar, fecha y duración***

Las pruebas se llevaron a cabo en el Laboratorio de Tecnologías Interactivas y Cooperativas (ICT), ubicado en el Centro Interactivo de Recursos de Información y Aprendizaje (CIRIA), dentro del campus universitario. Inicialmente se estimó la duración de las pruebas en 60 minutos, considerando que la temática que aborda este proyecto no es muy conocida, pero todos los participantes la terminaron en un máximo de 30 minutos.

Todo el proceso de la realización de las pruebas se llevó a cabo en dos semanas. Durante la semana comprendida del 17 al 21 de abril de 2006 se llevó a cabo la etapa de planeación y una prueba piloto. En la siguiente semana, del 24 al 28 de abril de 2006, se llevaron a cabo las pruebas definitivas, se analizaron los resultados, se elaboró el reporte de las pruebas y se desarrolló un plan de mejora a las funcionalidades y la interfaz gráfica de usuario del sistema.

#### ***5.1.1.3. Hardware y software necesarios***

Las pruebas se llevaron a cabo en el equipo de desarrollo del proyecto, debido a que éste contaba con los requerimientos de hardware y software para realizarlas. El equipo ejecutó a la vez el cliente y el servidor del sistema: tenía previamente instalado el Servidor HTTP de Apache con el sistema cargado, el navegador *Internet Explorer* y la versión 1.5 de la máquina virtual de Java, que incluye *Java Web Start*. Además del software necesario para ejecutar el sistema, se instaló en el equipo un demo del software *Camtasia Recorder*, que permitió grabar en video digital la actividad del monitor del equipo durante la realización de las pruebas.

#### ***5.1.1.4. Estado del sistema al iniciar la prueba***

Antes del inicio de cada prueba se prepararon el equipo y el sistema a probar de la siguiente manera:

- Se iniciaron el equipo y el servidor HTTP de Apache.
- Se mostró el navegador con la página Web que inicia la ejecución del sistema.
- Se preparó el software *Camptasia Recorder* para grabar la actividad del monitor.

#### ***5.1.1.5. Requerimientos de red y tráfico***

Al ejecutarse el cliente y el servidor del sistema en un solo equipo, no fue necesario contar con conexión a Internet o a redes de área local, ni con condiciones específicas de tráfico o factores relacionados.

#### ***5.1.1.6. Facilitadores y nivel de participación***

El autor del proyecto asumió el rol de facilitador de las pruebas. Su nivel de participación se limitó a recibir dudas y comentarios de los participantes relacionadas con el uso del sistema. Sólo se indicaron instrucciones cuando se pudo encontrar la manera de realizar tareas específicas de la hoja de actividades de la prueba.

#### ***5.1.1.7. Usuarios participantes***

Se realizó una prueba piloto y tres pruebas definitivas o finales, contando con la participación de un total de 4 personas. Los participantes de las pruebas fueron miembros de la comunidad universitaria:

- una estudiante de Doctorado en Ciencias de la Computación con un proyecto relacionado con objetos de aprendizaje, en vistas de la reutilización de este proyecto;
- un estudiante de licenciatura recién recibido, en cuyo proyecto de tesis fueron de gran importancia las pruebas de usabilidad, y cuyas recomendaciones fueron valiosas;
- un profesor de la facultad de Ciencias de la Información, visto como un potencial usuario final del sistema;
- un empleado del área de Innovación y Servicios de Información del CIRIA, visto también como un potencial usuario final del sistema.

#### ***5.1.1.8. Tareas a desarrollar por los usuarios***

Se definió un conjunto de pruebas que consta de los siguientes elementos:

- una hoja de derechos de los participantes;
- una hoja con la descripción básica de la temática de objetos de aprendizaje;
- un formato de carta de consentimiento;
- una hoja de actividades de evaluación;
- un cuestionario de retroalimentación.

Dicho conjunto de pruebas está disponible en el Anexo F de este documento. En la hoja de actividades de evaluación se establecieron las tareas a desarrollar por los participantes en las pruebas.

#### ***5.1.1.9. Criterios de terminación de tareas***

Se estableció un tiempo máximo de dos minutos y medio para la realización de cada actividad, sin embargo, también se dio por terminado el tiempo en situaciones cuando los participantes no podían realizar las actividades, preguntaron al facilitador cómo realizarlas y éste les indicó la manera de llevarlas a cabo.

#### ***5.1.1.10. Materiales de apoyo para usuarios***

Se desarrolló como parte del conjunto de pruebas (ver Anexo F) una hoja con la descripción básica de la temática de los objetos de aprendizaje, que fue especialmente útil en los casos en los que los participantes no estaban familiarizados con el tema. Adicionalmente el sistema contaba para el momento de la prueba con una barra de *tips* de uso del sistema, con indicaciones para llevar a cabo tareas concretas.

#### **5.1.1.11. Métodos de recolección y análisis de datos**

Se contó con tres fuentes de recolección y análisis de datos en estas pruebas:

- las hojas de actividades de evaluación de los participantes, en las que anotaron comentarios sobre su experiencia con el uso del sistema;
- los cuestionarios de retroalimentación de los participantes, en los que se pudieron analizar tanto preguntas específicas como comentarios adicionales;
- las grabaciones de la actividad del monitor con *Camtasia Recorder* durante el desarrollo de las pruebas, que permitieron analizar más a detalle las experiencias de los participantes al realizar las pruebas.

#### **5.1.2. Resultados obtenidos**

El anexo F de este documento contiene los documentos involucrados en la realización de estas pruebas de usabilidad, como las hojas de actividades y cuestionarios de retroalimentación de los participantes, así como gráficas de los resultados obtenidos.

A partir de la observación de los participantes en las pruebas, así como del posterior análisis de los documentos incluidos en el anexo F, y de los videos de actividad del monitor, se llegó a las conclusiones sobre la usabilidad del sistema que se presentan a continuación. Se presentan la medición de la satisfacción del usuario, los hallazgos importantes observados durante las pruebas y el análisis de datos, las recomendaciones de cambios a la interfaz de usuario, y finalmente una conclusión sobre las pruebas.

### ***5.1.2.1. Medición de la satisfacción del usuario***

Para medir la satisfacción del usuario, se realizó un promedio ponderado de las respuestas de los participantes en los cuestionarios de retroalimentación. Como se puede ver en el anexo F, cada pregunta en este cuestionario era de opción múltiple, con opciones numéricas del 1 al 5. Siendo un total de 4 participantes, la satisfacción total del uso del sistema se definiría como una frecuencia de 4 respuestas de la opción 5 en las 5 preguntas, esto es:

$$4 \times 5 \times 4 = 100 \%$$

El porcentaje real de satisfacción del usuario se obtuvo entonces de la siguiente manera:

- Para cada pregunta, se multiplicó el número de cada opción (1 a 5) por el número de participantes que la eligieron como respuesta (0 a 4). El valor mínimo de estas operaciones por pregunta puede ser 4 (1 X 4), y el valor máximo puede ser 20 (5 x 4).
- Se obtuvo un porcentaje de satisfacción de uso por cada pregunta, que no fue más que una regla de tres del resultado obtenido, tomando en cuenta que 20 como resultado indica un 100% de satisfacción. El valor mínimo posible es de 20%, y el máximo 100%.
- Se calculó la media de los cinco porcentajes de satisfacción obtenidos en el paso anterior, y el número obtenido se tomó como porcentaje de satisfacción de uso del sistema por parte de los participantes de las pruebas. El valor mínimo de este indicador es 16%, que se logra obteniendo 20% de satisfacción del usuario en cada pregunta, y el valor máximo es 100%, que se logra obteniendo un 100% de satisfacción en cada pregunta.

El resultado del cálculo descrito anteriormente dio un indicador global de satisfacción del usuario de 83%. La figura 5.1 es una gráfica de los porcentajes de satisfacción de uso del sistema separados por pregunta. En el anexo F se pueden encontrar más gráficas relacionadas con la obtención de este porcentaje.

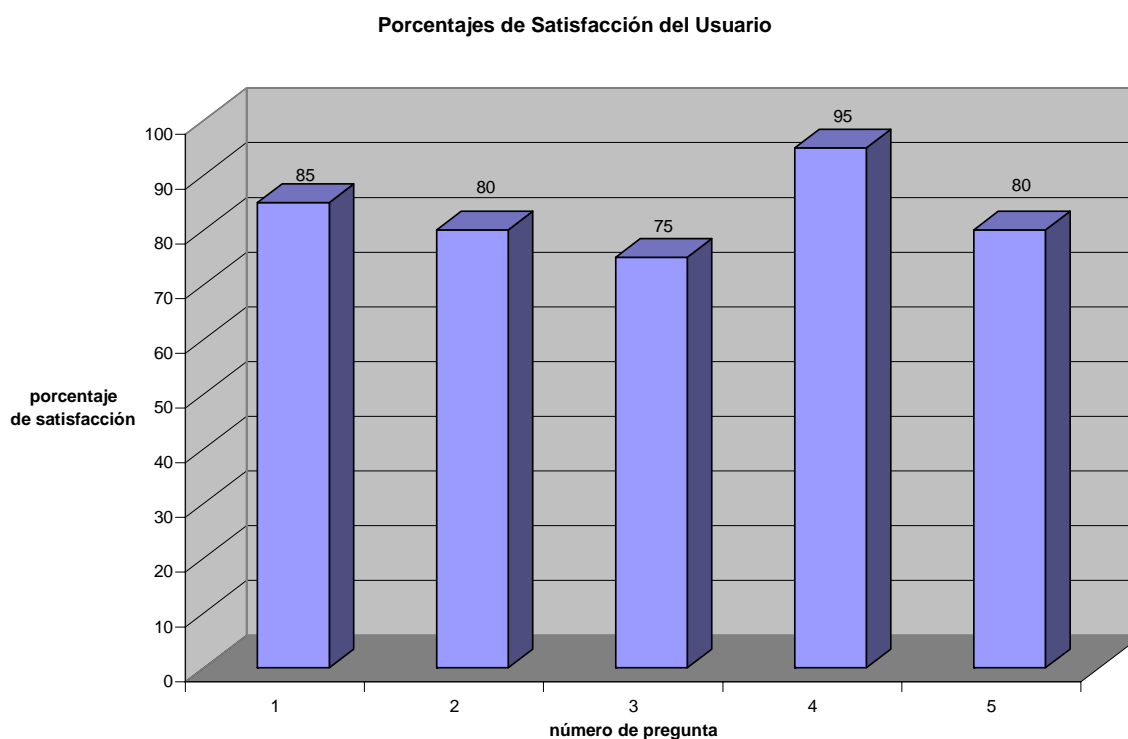


Figura 5.1. Gráfica de porcentajes de satisfacción por pregunta

### ***5.1.2.2. Hallazgos importantes***

A continuación se exponen los hallazgos más importantes de las pruebas realizadas. En primer lugar se mencionan los puntos fuertes encontrados y en segundo lugar se mencionan las áreas de mejora. Las imágenes de la interfaz gráfica de usuario del sistema VELOAT al momento de estas pruebas pueden encontrarse en el anexo F de este documento.

### Puntos fuertes del sistema

- El sistema es intuitivo. La interacción con el sistema para los participantes no fue sencilla al empezar, pero después de realizar pocas actividades con el mismo, se acostumbraron a él, lo encontraron más amigable y encontraron la manera de realizar las actividades de la prueba más rápido. Esta observación se confirmó a través de la retroalimentación tanto oral como escrita de los participantes, y se notó en los videos de actividad del monitor del equipo de pruebas.
- El sistema es fácil de aprender a usar. Para los participantes no familiarizados con la temática de objetos de aprendizaje fue más difícil la interacción con los árboles en la GUI, pero después de un tiempo y con la ayuda del facilitador, lograron entender que cada nodo en las vistas de árboles del sistema representa un nivel de granularidad distinto.
- El sistema es poco propenso a errores. Hubo casos en los que los participantes no realizaron acciones que desplegaran mensajes de error, y para los que sí los vieron, pudieron encontrar la manera adecuada de realizar las actividades en poco tiempo.

### Áreas de mejora

- La mayoría de los participantes expresaron su confusión, poca preferencia o falta de familiaridad con el uso del botón derecho del Mouse para visualizar menús emergentes en árboles.
- Algunos textos en los componentes de interacción como menús o botones no dan una idea clara de la funcionalidad que permiten hacer, como las opciones del menú “Archivo”.



- La opción de generación de metadatos no fue encontrada fácilmente por la mayoría de los usuarios. Algunos inclusive no lograron habilitarla. Otros la buscaron dentro de la interfaz de edición de metadatos y no en la interfaz de la ventana principal del sistema.
- La mayoría de los participantes tuvieron dificultades para llevar a cabo las actividades de establecimiento de puntos de entrada y cambio de visibilidad al usuario de recursos. En este sentido, no sólo es cuestión de los componentes de interacción, sino de la familiaridad con el tema. Independientemente de qué tan sencillo o usable pueda hacerse el sistema, se considera necesaria cierta capacitación inicial antes de usarlo.

#### ***5.1.2.3. Recomendaciones de cambios en la interfaz de usuario del sistema***

- Se sugiere la creación de barras de herramientas o íconos de acceso rápido para la realización de tareas comunes, como las que se realizan con los menús emergentes de los árboles de recursos y organizaciones de contenido.
- A pesar de que el sistema cuenta con una barra de *tips* de uso, y de que los participantes no la ocultaron durante todas las pruebas, la información en la barra no fue leída, con la consecuente dificultad en la realización de algunas tareas. Algunos participantes en su retroalimentación hicieron comentarios al respecto. Se recomienda reubicar la barra de *tips* y usar colores llamativos en la misma, o sustituirla por *tooltip texts*.
- Para los casos como las rutas de archivos de contenido o el ID y la versión del paquete, donde había campos de texto junto con un botón con la leyenda “Cambiar”, muchos usuarios teclearon en los campos de texto pero no hicieron clic en el botón. Se concluye que el botón debe eliminarse y se debe encontrar otro modo de actualizar el estado del sistema.

- Es necesario habilitar y deshabilitar los componentes de la GUI de acuerdo al estado del sistema para evitar inconsistencias y posibles fuentes de errores.
- Es importante que quede claro cuáles campos de la interfaz de usuario son obligatorios y cuáles no, y que la aplicación es *Key Sensitive*. En el caso de los campos obligatorios se sugiere el uso de colores o negritas.

#### ***5.1.2.4. Conclusiones y comentarios finales***

En resumen, la tendencia en las pruebas para los participantes en general fue de dificultad al principio y facilidad de uso al final. En algunos casos esta tendencia se debió a la falta de familiaridad con la temática del sistema, y otras porque los componentes de interacción necesitan estar más visibles al usuario.

Hubo casos en los que sorprendió la rapidez con la que algunos participantes encontraron la manera de realizar ciertas tareas. Uno de ellos inclusive se sintió con tanta confianza que encontró la manera de realizar tareas adicionales a las establecidas en la hoja de actividades, como eliminar recursos o agregar descripciones en el editor de LOM. En contraste a la situación anterior, hubo otros participantes que omitieron la realización de ciertas actividades o las llevaron a cabo de manera incorrecta sin que se dieran cuenta.

Todo lo anterior indica que el sistema cuenta con una interfaz fácil de aprender a usar, pero que puede mejorar en muchos sentidos todavía. Algunos cambios se llevaron a cabo antes del fin de este proyecto de tesis, como se puede apreciar en las figuras de la sección 4.1.2, pero como trabajo a futuro queda un estudio más exhaustivo de usabilidad.

## 5.2. Pruebas de funcionalidad

Las pruebas de funcionalidad realizadas al sistema desarrollado se llevaron a cabo primero por módulos, y después se buscó la integración de todos los módulos y tecnologías usadas, como *XMLBeans*, *Java Media Framework* y *Java Web Start*. En este capítulo se presentan los aspectos más relevantes en las distintas pruebas de funcionalidad realizadas.

Debido a la gran cantidad de archivos de código creados y a la extensión de los mismos, el código fuente de VELOAT es muy extenso, por lo que al final de estas pruebas el sistema aún queda en calidad de prototipo. Son tantas las funcionalidades que se podían probar detalladamente por separado, que una prueba más exhaustiva de la correcta funcionalidad de todo el sistema queda como trabajo a futuro.

### 5.2.1. Pruebas del módulo central

Las pruebas realizadas en este módulo fueron de dos tipos principales. En primer lugar, se verificó que el sistema creara paquetes de contenido educativo que fueran realmente compatibles con el estándar usado, y que pudieran ser usados en otros sistemas basados en el mismo estándar. Para esta finalidad, se hizo uso del software *SCORM Sample Runtime Environment* (SCORM SRTE), disponible en la Página Web de *Advanced Distributed Learning* (ver sección 2.6.4). En este software se cargaron los paquetes de contenido educativo generados con el sistema desarrollado.

Se realizaron varios cambios antes de que los paquetes creados por VELOAT pudieran ser visualizados correctamente en el SCORM SRTE. Los principales errores fueron originados por dos factores principales, que fueron los siguientes:

- El estándar requiere que los archivos de validación del manifiesto estén dentro de cada paquete creado, en la carpeta raíz del mismo, junto con este archivo. Inicialmente los *schemas* del estándar de Metadatos (LOM) eran guardados en una subcarpeta de los paquetes creados. Los archivos de esta subcarpeta se tuvieron que subir a la carpeta raíz y hacer los cambios necesarios en el resto de los *schemas* para actualizar la localización de los archivos movidos.
- Algunas veces, antes de usar *XMLBeans*, los manifiestos creados por VELOAT no podían ser validados por sus *schemas*, y el SCORM SRTE no podría cargarlos. Esta situación se mantuvo en algunos casos aún después de usar *XMLBeans*, que se supone que genera documentos validados, pero se redujo considerablemente. Algunas veces el SCORM SRTE no puede cargar paquetes creados con RELOAD si éstos tienen metadatos. Se encontró que la razón es que *XMLBeans* no siempre genera documentos LOM válidos. Queda como trabajo a futuro resolver esta situación.

El segundo tipo de prueba a la que se sometió este módulo fue a la correcta visualización y edición de paquetes de contenido educativo creados por otros sistemas. Para esta prueba se utilizaron 30 paquetes IMS creados con la herramienta RELOAD (ver sección 2.6.5) disponibles en la Colección de Objetos de Aprendizaje de la Universidad de Guadalajara Virtual, disponible en línea a través de la siguiente dirección: <http://www.udgvirtual.udg.mx/dspace/handle/123456789/72>. Las siguientes fueron las dos principales fuentes de error encontradas:

- La mayoría de los paquetes en el repositorio consultado tienen un archivo XML separado del manifiesto principal para los metadatos LOM. VELOAT inicialmente estaba diseñado para obtener estos metadatos del archivo manifiesto principal de los paquetes, pero la situación se encontró tan comúnmente que se implementó el soporte para leer y guardar metadatos LOM en archivos separados del manifiesto principal. Aún así, se encontró que RELOAD no sigue el estándar de metadatos recomendado por SCORM, y por lo tanto no se pudieron visualizar ni editar metadatos de los paquetes.
- Todos los paquetes usados estaban en formato ZIP. VELOAT inicialmente sólo podría leer archivos XML correspondientes a manifiestos, pero se implementó el soporte para visualizar y editar paquetes de contenido educativo a partir de archivos ZIP.

La tabla 5.1 muestra un resumen de estas pruebas de funcionalidad.

<b>Prueba realizada</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>
Cargar paquetes creados con VELOAT en el sistema SCORM SRTE	Compatibilidad de los paquetes creados con VELOAT en otros sistemas	Después de realizar correcciones, la mayoría de los paquetes generados con VELOAT pueden ser cargados y visualizados en el SCORM RTE sin mayores problemas.
Abrir paquetes creados con RELOAD en el sistema desarrollado (VELOAT)	Compatibilidad de los paquetes creados con otras herramientas con VELOAT	Se logró visualizar y editar la información de recursos, organizaciones de contenido y empaquetado. Los metadatos no se lograron visualizar ni editar debido a que RELOAD no sigue el estándar de metadatos recomendado por SCORM.

Tabla 5.1. Resumen de pruebas de funcionalidad del módulo central de VELOAT

### 5.2.2. Pruebas del módulo de metadatos

Este módulo se probó de dos formas. En primer lugar, se probó su funcionamiento como parte del módulo central de VELOAT, y en segundo lugar como módulo independiente. La tabla 5.2 muestra un resumen de las pruebas de funcionalidad de este módulo. En ambos casos, se corrigieron las siguientes situaciones de error identificadas a la hora de editar y generar metadatos:

- Inconsistencias entre vistas y modelos, que se presentaban como falta de actualización de modelos a partir de los datos en las vistas o como falta de notificación a las vistas de cambios en los modelos.
- Modificación no deseada de los metadatos previamente existentes al cancelar los cambios realizados en el editor por el usuario.
- Falta de actualización de los metadatos originales una vez aceptados los cambios realizados por el usuario en el editor.
- Eliminación de metadatos previamente existentes al activar la generación semiautomática de metadatos.
- Repetición de valores en la generación de metadatos. Se optó por evitar el llamado a los métodos de generación de metadatos en categorías y etiquetas que ya tuvieran valores.

<b>Prueba realizada</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>
Edición y generación de valores de metadatos	Consistencia entre modelo y vista del módulo de metadatos	Identificación y corrección de errores de consistencia entre modelo y vista.

Tabla 5.2. Resumen de pruebas de funcionalidad del módulo de metadatos

### 5.2.3. Pruebas del módulo de tecnologías de voz

Al igual que el módulo anterior, este se probó tanto tomándolo en cuenta como parte del módulo principal y como en su capacidad de módulo independiente. La tabla 5.3 muestra un resumen de las pruebas de funcionalidad de este módulo. Los errores encontrados y corregidos a la hora de realizar grabaciones de voz en este módulo fueron los siguientes:

- Errores y excepciones surgidas del mal uso de la librería *Java Media Framework*, al tratar de reproducir grabaciones recién realizadas, o al tratar de sobrescribirlas.
- Modificación no intencionada de archivos de audio previamente existentes al cancelar los cambios realizados en el grabador de voz.
- Falta de actualización de archivos de audio previamente existentes al aceptar los cambios realizados en el grabador de voz.
- “Congelamientos” del sistema debidos a falta de control de eventos o control de los estados de los objetos de la librería *Java Media Framework*.

<b>Prueba realizada</b>	<b>Resultados esperados</b>	<b>Resultados obtenidos</b>
Crear un nuevo archivo de grabación de voz	Se esperaba como salida un nuevo archivo de audio en el sistema del cliente.	Se lograron los resultados esperados una vez manejando adecuadamente la librería <i>Java Media Framework</i> .
Crear una grabación de voz sobre-escribiendo un archivo existente.	Recuperación del archivo original si el usuario salía del módulo con la opción “Cancelar”. Sobre-escritura del archivo original si el usuario salía del módulo con la opción “Aceptar”.	Al principio se daban modificaciones no intencionadas de los archivos originales, así como falta de sobre-escritura. Estas situaciones fueron corregidas.

Tabla 5.3. Resumen de pruebas de funcionalidad del módulo de grabación de voz

#### 5.2.4. Pruebas de integración de las tecnologías usadas

Surgieron tres problemas principales a la hora de tratar de integrar los módulos del sistema y las tecnologías usadas:

- En primer lugar, parecía no ser posible el acceso al sistema de archivos del cliente desde *Java Web Start*. Esta situación se corrigió al especificar en el archivo JNLP de la aplicación que dicha funcionalidad se permitiera, así como firmando los archivos JAR que conformaron el sistema con certificados de seguridad temporales.
- *XMLBeans* parecía no funcionar sobre *Java Web Start*, que lanzaba una excepción antes de inicializar VELOAT. Fue necesario descargar el código del proyecto *XMLBeans* para corregir esta situación. Se hicieron cambios en tres archivos y se recompiló el proyecto para poder corregir esta situación.
- *Java Media Framework* lanzaba excepciones de seguridad sobre *Java Web Start*, que indicaban falta de permisos para crear archivos de audio o para leer librerías del sistema del lado del cliente. Se encontraron varias opciones de solución a este problema, como la definición o modificación de archivos de políticas de seguridad del cliente, la modificación de variables de entorno en el sistema del cliente, la implementación de una clase gestora de seguridad en el sistema y la instalación local de *Java Media Framework* del lado del cliente como requerimiento para poder realizar grabaciones de voz. Se probaron todas las posibles opciones de solución hasta que finalmente se eligió la última de ellas, debido a que instalando *Java Media Framework* del lado del cliente, el usuario es quien configura esta herramienta y decide si otorga los privilegios necesarios para que el módulo de grabación de voz de VELOAT funcione.