



Universidad de las Ciencias
Informáticas

Facultad 1

Herramienta para la migración de los archivos ofimáticos

*Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas*

Autoras:

Madeleinis Morales Rodríguez

María Leisy González Carrera

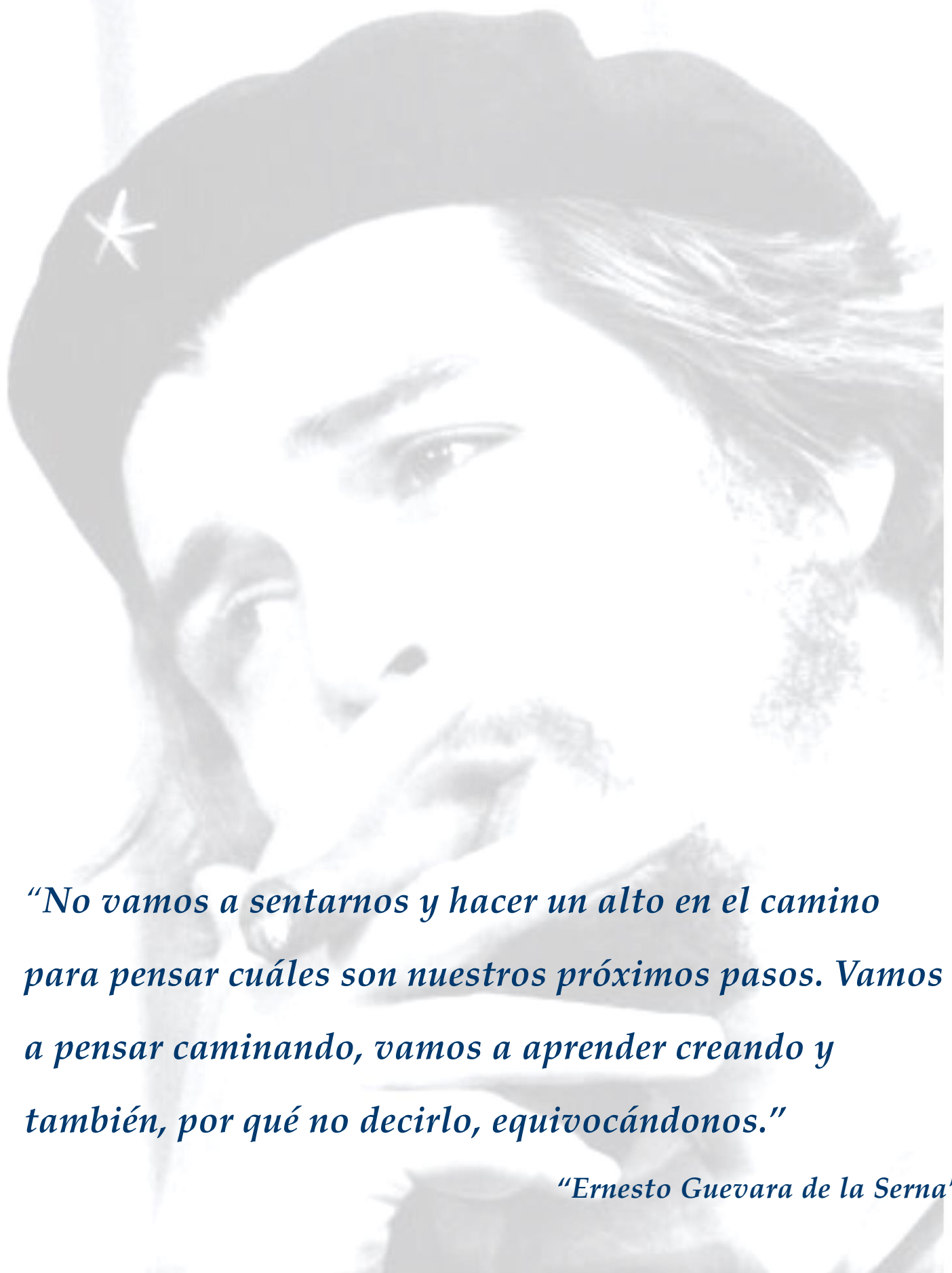
Tutores:

Msc. Yenisleydi Cariaga Cristo

Ing. Yoandy Pérez Villazón

La Habana, Junio 2013

"Año 55 de la Revolución"



“No vamos a sentarnos y hacer un alto en el camino para pensar cuáles son nuestros próximos pasos. Vamos a pensar caminando, vamos a aprender creando y también, por qué no decirlo, equivocándonos.”

“Ernesto Guevara de la Serna”

Declaración de autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Madeleinis Morales Rodríguez

Msc. Yenisleydi Cariaga Cristo

María Leisy González Carrera

Ing. Yoandy Pérez Villazón



Agradecimientos

A nuestro Comandante en Jefe Fidel Castro Ruz, por darnos la oportunidad de estudiar en la Universidad de las Ciencias Informáticas.

A nuestros tutores Yeni y Yoandy, que nos brindaron su apoyo incondicional, guiándonos en todo momento y confiando plenamente en nuestras decisiones.

A los profesores que han contribuido a nuestra formación en todos estos años.

A nuestros compañeros por su invaluable ayuda y las siempre bien recibidas enseñanzas.

A todas aquellas personas que con su esfuerzo han hecho posible la realización del presente trabajo, en especial:

Al profesor Migue, a Jailen, Abelito, Luis Daniel, Osiris y a Nelio que jugaron un papel fundamental en el desarrollo de nuestra tesis

A todos Muchas Gracias!!!



Dedicatoria

Made:

A mis padres y a mi hermana que han sido mi fuerza, mi espíritu y mis ganas de seguir adelante en la vida, a ustedes les dedico todos mis éxitos.

A mi novio que ha ocupado un papel fundamental en mi vida y en los últimos meses de mi carrera.

A mi pequeña sobrina que viene en camino y que desde su arcano rinconcito, se ha convertido en la alegría de la familia.

A mis seres queridos y a todos los que hicieron de mí, la mujer que soy hoy.

Mary:

Quisiera dedicar este éxito a mis padres que los amo y los adoro con la vida y porque si no fuera por ellos no existiera, a mi hermana por aconsejarme mucho, a mi abuela porque desde que tengo uso y razón me ha inculcado el respeto, la humildad y el cariño.

A mi sobrinito Alenjandrito y a mi primito Yosva para que me tengan presente en cada paso de su vida.

Y en especial a mi tía Noe que fue mi otra mami, y aunque no pueda estar físicamente aquí conmigo lo estará espiritualmente hoy y siempre, porque a ella le debo gran parte de lo que soy, una muchacha alegre, fuerte y sobre todo luchadora.



Resumen

Luego de la experiencia en la ejecución de diversos procesos de migración ofimática en Cuba por parte de los especialistas del departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS), fueron detectadas un conjunto de irregularidades en el proceso de cambio de formato de los archivos de extensión docx, xlsx, pptx y mdb hacia odt, ods, odp y odb respectivamente, las cuales afectan el tiempo de duración de la migración ofimática y en ocasiones se incurre en errores humanos por la realización manual de la tarea. Para resolver los problemas antes mencionados se desarrolló una solución informática cuyo objetivo es proveer a los especialistas de una herramienta que automatice el proceso. Fueron estudiadas las principales soluciones existentes de código abierto, distinguiéndose dos grupos: las orientadas a migrar archivos de documentos, presentaciones y hojas de cálculo, y las destinadas a la migración de bases de datos personales. Fueron empleadas las herramientas JODConverter, Odf-converter y mdbtools para la construcción del sistema. Otro conjunto de librerías empleadas favorecieron la reutilización, entre ellas jodconverter.jar, hsqldb.jar, sqllitejdbc.jar. Se construyó además un componente reutilizable llamado mdbaOdb.jar para la migración de los datos de bases de datos de Microsoft Access hacia LibreOffice Base. El desarrollo del sistema estuvo guiado por la metodología de desarrollo de *software* SXP. Los requisitos del cliente fueron cumplidos satisfactoriamente, el mismo fue partícipe del proceso de pruebas al producto, realizando las pruebas de aceptación. Al culminar la tercera iteración se obtuvo un producto libre de errores y adecuado a las exigencias iniciales.

Palabras Claves: archivos ofimáticos, conversión, componente reutilizable, gestión, migración de archivos.

Índice de Contenido

| | |
|--|---|
| Índice de Figuras..... | |
| Índice de Tablas..... | |
| Introducción | 1 |
| Capítulo 1: Fundamentación teórica | 5 |
| 1.1 Conceptos Fundamentales | 5 |
| Extensión | 5 |
| Formatos..... | 5 |
| Archivos | 5 |
| Formatos libres o abiertos | 6 |
| Formatos no libres, privativos o cerrados | 6 |
| Suite Ofimática..... | 6 |
| 1.2 Estándares para el intercambio de información ofimática..... | 6 |
| 1.3 Herramientas para la conversión de archivos ofimáticos..... | 7 |
| 1.3.1 Odf-convert | 7 |
| Observaciones: | 8 |
| 1.3.2 Unoconv..... | 8 |
| Observaciones: | 8 |
| 1.3.3 JODConverter | 8 |
| Observaciones: | 8 |
| 1.3.4 Convertidor de documentos de LibreOffice y Apache OpenOffice | 8 |
| Observaciones: | 9 |
| Conversor de bases de datos mdbtools | 9 |



| | |
|---|----|
| 1.3.6 Comparación de las herramientas de conversión..... | 9 |
| 1.3. 7 Elección de las herramientas | 10 |
| 1.4 Migración de las base de datos de Access hacia LibreOffice Base | 11 |
| 1.4.1 Análisis de la estructura de una base de datos odb..... | 12 |
| 1.5 Metodología de Desarrollo | 12 |
| 1.6 Tecnologías empleadas en el desarrollo de la solución | 13 |
| 1.6.1 Lenguaje de programación..... | 13 |
| 1.6.2 Lenguaje de modelado..... | 13 |
| 1.6.3 Base de datos | 14 |
| SQLite | 14 |
| 1.6 4 Librerías y aplicaciones externas | 14 |
| Hsqldb.jar | 14 |
| Sqlitejdbc.jar..... | 14 |
| JODConverter.jar..... | 14 |
| Aplicaciones mdbtools y Odf-converter | 15 |
| 1.6.4 Herramientas para el desarrollo del sistema | 15 |
| Netbeans:..... | 15 |
| Visual Paradigm para UML: | 15 |
| Subversion: | 15 |
| RapidSVN | 16 |
| Conclusiones del Capítulo | 16 |
| Capítulo 2: Propuesta de solución | 17 |
| 2.1 Descripción de la solución..... | 17 |
| 2.2 Requisitos del sistema. | 17 |

| | |
|---|----|
| 2.2.1 Lista de Reserva del Producto | 17 |
| 2.3 Historias de Usuario..... | 19 |
| 2.4 Consideraciones de diseño | 24 |
| 2.4.1 Arquitectura de software..... | 24 |
| 2.4.2 Patrón Arquitectónico | 25 |
| 2.4.3 Modelo de Diseño | 26 |
| 2.4.4 Patrones GRASP | 28 |
| Experto:..... | 28 |
| Creador: | 29 |
| Controlador: | 29 |
| Bajo Acoplamiento:..... | 29 |
| Alta Cohesión:..... | 30 |
| 2.2.5 Patrones GOF | 30 |
| 2.4.6 Exigencias mínimas para el funcionamiento del sistema | 31 |
| Conclusiones del Capítulo | 32 |
| Capítulo 3: Implementación y prueba..... | 33 |
| 3.1 Plan de Release..... | 33 |
| 3.2 Tareas de Ingeniería | 34 |
| 3.3 Estándar de Codificación | 37 |
| 3.4 Componentes definidos para el sistema..... | 38 |
| 3.5 Descripción de los métodos del sistema | 40 |
| 3.6 Pruebas | 41 |
| 3.6.1 Casos de pruebas de Aceptación..... | 41 |
| Observaciones: | 45 |

| | |
|--|----|
| 3.7 Impacto y aporte de la solución..... | 46 |
| Conclusiones de Capítulo: | 47 |
| Conclusiones | 48 |
| Recomendaciones | 49 |
| Referencias Bibliográficas..... | 50 |
| Bibliografía..... | 52 |
| Glosario de Términos..... | 53 |
| Anexos..... | 55 |
| Anexo 1..... | 66 |
| Anexo 2..... | 57 |
| Anexo 3..... | 58 |
| Anexo 4..... | 59 |
| Anexo 5..... | 62 |

Índice de Figuras

| | |
|---|----|
| Ilustración 1: Estructura de las bases de datos odb | 12 |
| Ilustración 2: Diagrama de componentes | 27 |
| Ilustración 3: patrón Experto | 28 |
| Ilustración 4: Patrón Creador | 29 |
| Ilustración 5: Patrón Controlador..... | 29 |
| Ilustración 6: patrón Bajo Acoplamiento | 30 |
| Ilustración 7: patrón Alta Cohesión..... | 30 |
| Ilustración 8: patrón Singleton..... | 31 |
| Ilustración 10: Diagrama de componentes | 39 |
| Ilustración 11: Proceso de pruebas | 46 |
| Ilustración 12: JODConverter | 55 |
| Ilustración 13: Unoconv..... | 55 |
| Ilustración 14: Odf-converter..... | 56 |
| Ilustración 15: Ejemplo de la implementación del VBA y el OOO Basic para macros de un Excel | 57 |
| Ilustración 16: Modelo del dominio | 58 |
| Ilustración 17: Interfaz de la funcionalidad Convertir | 59 |
| Ilustración 18: Prototipo de la funcionalidad ubicar archivo en la ruta especificada..... | 60 |
| Ilustración 19: Prototipo de la interfaz Añadir archivo..... | 60 |
| Ilustración 20: Prototipo de interfaz de la funcionalidad Eliminar archivo | 61 |
| Ilustración 21: Prototipo de interfaz de la funcionalidad Limpiar lista..... | 61 |

Índice de Tablas

| | |
|--|----|
| Tabla 1: Comparación de las herramientas..... | 10 |
| Tabla 2: Requisitos del sistema | 34 |
| Tabla 3: HU_1..... | 20 |
| Tabla 4: HU_2..... | 21 |
| Tabla 5: HU_3..... | 21 |
| Tabla 6: HU_4..... | 22 |
| Tabla 7: HU_5..... | 22 |
| Tabla 8: HU_6..... | 23 |
| Tabla 9: HU_7..... | 23 |
| Tabla 10: HU_8..... | 24 |
| Tabla 11: Plan de Release..... | 33 |
| Tabla 12: Tarea de Ingeniería_1 | 34 |
| Tabla 13: Tarea de Ingeniería_2..... | 35 |
| Tabla 14: Tarea de Ingeniería_6..... | 35 |
| Tabla 15: Tarea de Ingeniería_7..... | 35 |
| Tabla 16: Tarea de Ingeniería_8..... | 36 |
| Tabla 17: Tarea de Ingeniería_9..... | 36 |
| Tabla 18: Tarea de Ingeniería_17..... | 37 |
| Tabla 19: Tarea de Ingeniería_18..... | 37 |
| Tabla 20: Tarea de Ingeniería_19..... | 37 |
| Tabla 21: Descripción de los métodos de la clase_Convertir | 40 |
| Tabla 22: Descripción de los métodos de clase_Archivo | 41 |
| Tabla 23: Descripción de los métodos de clase_AccesoADatos | 41 |



| | |
|--|----|
| Tabla 24: Caso de Prueba de Aceptación_1 | 42 |
| Tabla 25: Caso de Prueba de Aceptación_2 | 43 |
| Tabla 26: Caso de Prueba de Aceptación_3 | 43 |
| Tabla 27: Caso de Prueba de Aceptación_4 | 43 |
| Tabla 28: Caso de Prueba de Aceptación_5 | 44 |
| Tabla 29: Caso de Prueba de Aceptación_6 | 44 |
| Tabla 30: Caso de Prueba de Aceptación_7 | 45 |
| Tabla 31: Caso de Prueba de Aceptación_8 | 45 |
| Tabla 32: Tarea de Ingeniería_3 | 62 |
| Tabla 33: Tarea de Ingeniería_4 | 62 |
| Tabla 34: Tarea de Ingeniería_5 | 63 |
| Tabla 35: Tarea de Ingeniería_10 | 63 |
| Tabla 36: Tarea de Ingeniería_11 | 63 |
| Tabla 37: Tarea de Ingeniería_12 | 64 |
| Tabla 38: Tarea de Ingeniería_13 | 64 |
| Tabla 39: Tarea de Ingeniería_14 | 64 |
| Tabla 40: Tarea de Ingeniería 15 | 65 |
| Tabla 41: Tarea de Ingeniería 16 | 65 |
| Tabla 42: Tarea de Ingeniería 20 | 65 |

Introducción

El desarrollo científico existente en la actualidad, ha impulsado cambios significativos en el campo de las Tecnologías de la Información y las Comunicaciones (TIC), provocando un avance progresivo en todas las esferas de la humanidad. Su evolución ha dado lugar a lo que se denomina sociedad de la información, caracterizada por manejar grandes volúmenes de datos a través de distintos medios.

El surgimiento del Software Libre (SWL) constituye un cambio contundente dentro del ciclo evolutivo de las TIC y una alternativa para los países subdesarrollados. Su filosofía se basa en la autonomía que poseen los usuarios de copiar, distribuir, estudiar, modificar y mejorar el *software*, permitiendo, una vez que el usuario lo adquiera, adaptarlo a los requerimientos tecnológicos de su entorno, además de distribuirlo libremente a instituciones u otros usuarios que requieran del programa (1). Por las ventajas socio-económicas que implica la utilización del SWL, muchos países hacen uso de él, modificándolo de manera tal, que su tecnología no se vea afectada por cuestiones de cambios.

El uso del SWL representa para Cuba una herramienta fundamental para llevar a cabo la informatización de la sociedad, pues el bloqueo económico, financiero, comercial y tecnológico impuesto por el gobierno de Estados Unidos impide el desarrollo progresivo de la misma, resultando una utopía para el país el uso o acceso legal a las aplicaciones tecnológicas o productos desarrollados por empresas norteamericanas (2).

El gobierno está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos, convencido de que para los países subdesarrollados resulta imprescindible el logro de este propósito, pues su objetivo fundamental es lograr el crecimiento económico de sus pueblos.

A partir del año 2004 se inicia en el país un proceso paulatino de introducción en el programa de informatización de tecnologías de SWL, logrando avanzar en la migración de varias entidades nacionales, entre las que se destacan la Aduana General de la Republica, la red de datos Infomed, entre otras.

Con el objetivo de ordenar el programa de migración nacional, se creó el Grupo Ejecutivo Nacional, al cual se subordinan cuatro subgrupos, entre los que se encuentra el Grupo Técnico Nacional, encargado de desarrollar el sistema operativo cubano y definir los lineamientos, políticas y estrategias para el desarrollo exitoso de la migración en los Organismos de la Administración Central del Estado (OACE). Este último grupo posee su principal fuerza de trabajo en el Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI). El departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS) pertenece a CESOL y asesora a las entidades nacionales en su cambio tecnológico hacia SWL.

Durante el proceso de elecciones generales convocado para el domingo 21 de octubre del 2012 la Comisión Electoral Nacional (CEN) y la Oficina Nacional para la Informatización (ONI) decidieron impulsar el uso de aplicaciones de SWL como soporte tecnológico para tan importante proceso que involucra a toda la nación. Para poder lograr este objetivo, al departamento SIMAYS se le indicó la migración ofimática de aproximadamente 800 documentos usados sobre Microsoft Windows en pasadas elecciones.

Entre los archivos se encontraban documentos de Microsoft Word, Excel y PowerPoint. El proceso de conversión se realizó de forma manual usando la herramienta ofimática LibreOffice, involucrando un grupo grande de especialistas. El trabajo de conversión culminó satisfactoriamente y los documentos fueron entregados a la CEN, sin embargo fueron detectados un conjunto de dificultades, entre las que se encuentran:

- La tarea tardó más tiempo del planificado debido a la realización manual del proceso por parte de los especialistas.
- Se incurrió en errores en los documentos migrados; por ejemplo: introduciendo estilos de fuentes diferentes en los archivos.
- Se dedicaron un conjunto importante de recursos humanos para la actividad, teniendo que realizar una fase de pruebas de calidad al terminar la migración de los documentos para validar su correcto formato y estructura.

Como parte del asesoramiento que ofrece SIMAYS a los OACE, se ha trabajado en la migración de diversas entidades entre las que se encuentran la empresa Telemar y el Centro de Cibernética Aplicada a la Medicina (CECAM), en ambas instituciones se culminó el paso de tecnologías privativas hacia aplicaciones de SWL. Una de las tareas fundamentales desarrolladas fue la migración ofimática de los documentos. Esta actividad trajo consigo problemas similares a los ocurridos en la migración de los documentos de la CEN. A estos casos se le adicionó la necesidad de portar hacia la herramienta LibreOffice Base las bases de datos de Microsoft Access con las que contaban ambas instituciones. El proceso tuvo que realizarse usando como intermediario una base de datos MySQL, elemento que complejizó la tarea. Teniendo en cuenta que SIMAYS tiene como objetivo ofrecer soluciones de apoyo a la migración nacional, que sean lo más sencilla posible de utilizar, este método empleado para la migración de los datos del gestor privativo hacia su similar libre no es el más adecuado.

La migración que tuvo lugar con los documentos de la CEN y la transferencia de los datos de Microsoft Access hacia LibreOffice Base son actividades que están presentes en cualquier proceso de migración nacional.

A partir de las dificultades detectadas en la migración ofimática en los casos anteriores se plantea como **problema científico**: ¿Cómo migrar los formatos de archivos ofimáticos provenientes de Microsoft Office hacia LibreOffice 3.6 garantizando la conversión de los archivos de extensiones docx, xlsx, pptx y .mdb hacia odt, ods, odp y odb respectivamente?

Se plantea como **objetivo general**: Desarrollar una solución informática para la migración de los formatos de archivos provenientes de Microsoft Office con extensiones docx, xlsx, pptx y mdb hacia odt, ods, odp y odb respectivamente, de forma tal que se garantice la integración de varias aplicaciones en una solución única y se facilite el proceso de migración ofimática a los especialistas.

En la presente investigación se define como **objeto de estudio**: Herramientas para la conversión de formatos de archivos ofimáticos, enmarcado en el **campo de acción**: Herramientas de código abierto para la conversión de los formatos de archivos ofimáticos de extensiones docx, xlsx, pptx y mdb hacia odt, ods, odp y odb.

Para darle cumplimiento al objetivo general se plantean los siguientes objetivos:

- Sistematizar sobre las herramientas de conversión de formatos de archivos ofimáticos.
- Analizar y diseñar la herramienta de migración de los formatos de archivos.
- Implementar el sistema diseñado.
- Realizar pruebas al sistema.

Para dar cumplimiento a los objetivos de la investigación, se definieron una serie de tareas a desarrollar, tales como:

- Definición de los conceptos asociados al dominio de la investigación y estudio de las soluciones de código abierto existentes para la conversión de los formatos de archivos ofimáticos.
- Definición de los requisitos y arquitectura del sistema.
- Construcción de la herramienta para la migración de archivos ofimáticos.
- Análisis de los resultados.

Se **defiende la idea** que: Una herramienta informática única que permita la conversión de los formatos de archivos de Microsoft Office con extensión docx, xlsx, pptx y mdb hacia odt, ods, odp y odb respectivamente facilitará el proceso de migración ofimática a los especialistas.

Entre los **Métodos Teóricos** utilizados se encuentran:

- Método inducción-deducción:

La deducción es una forma de razonamiento, mediante el cual se pasa de un conocimiento general a otro de menos nivel de generalidad. El método Inducción-deducción se llevó a la práctica a través de estudios realizados a las herramientas de conversión de archivos ofimáticos a lo largo de la investigación, que aportaron el conocimiento necesario para llevar a cabo la elaboración de la propuesta de solución.

Técnicas de Recopilación:

➤ Observación científica:

Se realizó una observación selectiva, sistémica y objetiva acerca de la evolución de las herramientas de conversión de archivos ofimáticos, teniendo en cuenta sus características y funcionalidades, con el fin de definir la estructura de la propuesta de solución.

➤ Entrevista no estructurada:

Se realizaron entrevistas abiertas a especialistas en el tema para obtener criterios acerca de la estructura y funcionamiento del sistema a implementar.

El documento consta de 3 capítulos bien estructurados y de forma organizada:

Capítulo 1: Definición de los conceptos asociados al dominio de la investigación, sistematización de las herramientas libres de migración de formatos de archivos ofimáticos. Descripción y caracterización de la metodología, las herramientas y tecnologías empleadas en el desarrollo del sistema.

Capítulo 2: Diseño de la solución propuesta guiado por la metodología de desarrollo definida, identificación de funcionalidades, descripción de historias de usuario y consideraciones del diseño del sistema.

Capítulo 3: Definición del plan de release, descripción de las tareas de ingeniería, las clases del sistema y los casos de pruebas, además del análisis del impacto de la solución propuesta.

Capítulo 1: Fundamentación teórica

Dentro del proceso de migración se encuentran implícitas diversas tareas que de una forma u otra complementan su ejecución. La migración de los formatos se convierte en un elemento decisivo dentro de la institución a migrar, garantizando la accesibilidad y perpetuidad de los datos existentes.

En el presente capítulo se precisan términos importantes que facilitan la comprensión del objeto de estudio y el campo de acción donde se enmarca la investigación. Se exponen elementos relacionados con los tipos de formatos actuales, haciendo énfasis en el análisis de las herramientas que permiten convertir diferentes formatos de archivos y los avances que se han originado hasta la actualidad en cuanto al tema, además de vincular y especificar las herramientas y lenguajes de programación necesarios para la elaboración de la solución.

1.1 Conceptos Fundamentales

El desarrollo de la herramienta para la migración de archivos ofimáticos, toma como punto de partida algunos núcleos conceptuales que delimitan el estado actual en el que se encuentra el progreso de la misma, así como precisa los términos que se emplearán para elucidar las bases teóricas de la investigación.

Extensión

Una extensión de archivo o extensión de fichero, es una cadena de caracteres anexada al nombre de un archivo, usualmente precedida por un punto. Su función principal es diferenciar el contenido del archivo, de modo que el sistema operativo disponga el procedimiento necesario para ejecutarlo o interpretarlo. Sin embargo, la extensión es solamente parte del nombre del archivo y no representa ningún tipo de obligación respecto a su contenido. Algunos sistemas operativos utilizan las extensiones de archivo para reconocer su formato, incluyendo el de archivos ejecutables (3).

Formatos

Un formato de archivo es una manera particular de codificar información para almacenarla en un archivo dentro una computadora. Es necesario conocer el formato utilizado al almacenarla para recuperarla correctamente. La misma información codificada de manera diferente, se guarda distinta en la memoria (4).

Archivos

La información de una computadora está almacenada en lo que se llaman archivos, o ficheros (*files* en inglés), que suelen estar formados por un nombre, un punto y una extensión que normalmente son tres o cuatro caracteres (Ej. HISTLABORAL.EXE). El nombre sirve para identificar y diferenciar unos archivos de otros. La

extensión sirve para atribuirle propiedades como "tipo de archivo" (datos, programa e imagen.) (5).

Los archivos en una empresa son un instrumento muy importante, en ellos se recopilan de forma organizada los registros y documentos de interés para la empresa. El procedimiento de clasificar, organizar y conservar los registros de forma sistemática se le conoce como archivar, actividad que se realiza con la finalidad de lograr rapidez en la localización de los documentos.

Formatos libres o abiertos

Son aquellos que poseen una especificación de referencia bajo una licencia libre y pueden ser implementados por cualquiera sin restricciones legales de uso. Habitualmente son publicados y patrocinados por organizaciones de estándares abiertos, aunque muchos son desarrollados por empresas (6).

Formatos no libres, privativos o cerrados

Son aquellos que tienen restricciones legales de uso, no pueden ser implementados por cualquiera ya que sus especificaciones no son públicas, están sujetos al pago de licencias y son controlados y definidos por intereses privados (7).

Suite Ofimática

Una suite está integrada por programas informáticos diseñados para el trabajo de oficina. Estos se pueden distribuir en conjunto o individualmente. Los programas de una misma suite ofimática permiten la interacción mutua con un sistema operativo y normalmente presentan una interface similar (8).

1.2 Estándares para el intercambio de información ofimática

El 1 de mayo del 2006, la Organización Internacional de Estandarización (ISO) aprobó, por unanimidad y de forma inédita hasta la fecha, la estandarización como norma ISO 26300¹ del formato ofimático ODF, respaldada por prácticamente todas de las empresas e instituciones del sector informático, salvo *Microsoft Corporation*, la única entidad que decidió no apoyar ni usar el formato ISO 26300, conocida por ser la distribuidora de la aplicación ofimática con posición dominante en el mercado actual: MS-Office. Hasta la versión de Microsoft Office 2003, MS-Office ha generado los documentos en unos formatos exclusivos y secretos (.doc, .xls y .ppt) que sólo habían podido ser descifrados parcialmente por sus competidores. Sin embargo, ante la presión de las

¹ISO / IEC 26300:2006 describe el contenido de la tabla de un documento en formato OpenDocument, su contenido gráfico y el contenido del formulario. También se describe el contenido que es común en todos los documentos. Se describe la integración de animación SMIL marcado en el esquema OpenDocument. Explica contenido de la información de estilo y especifica las propiedades de formato que se pueden utilizar dentro de los estilos. Se describen los tipos de datos utilizados por el esquema OpenDocument. El formato OpenDocument hace uso de un concepto de paquete; estos paquetes se describen.

aplicaciones basadas en el formato abierto ODF, Microsoft, en vez de competir con su aplicación usando el estándar ISO 26300 existente, decide iniciar una guerra de formatos creando uno nuevo alternativo y totalmente distinto a los anteriores, llamado originalmente *Microsoft Office OpenXML*, OpenXML u OOXML que, tras su paso por *European Computer Manufacturers Association (ECMA)*, actualmente conocida como ECMA International, comenzó a ser llamado OOXML. Siendo interés de Microsoft que este formato sea estandarizado como ISO 29500² pese a que sólo su aplicación MS-Office sea la única que los implementa en una única plataforma (9).

Aunque el formato OOXML es estándar (EMCA), la ISO rechazó por votación en el 2007 la solicitud de *Microsoft* para registrarlo como estándar internacional, pues contradice los estándares ISO preexistentes, pasando por alto la interoperabilidad, ya que no se documentaron partes del estándar propuesto necesarias para una implementación independiente.

Aun cuando las suites ofimáticas libres puedan ejecutar el formato OOXML, este no ha sido estandarizado de forma correcta, pues la compañía *Microsoft* es la que implementa y modifica las definiciones del mismo para después ponerlo en el mercado impidiendo que sea modificado por otras entidades, lo cual constituye una competencia desleal. El proceso de estandarización debe ser llevado a cabo por una empresa imparcial y no por la que implemente el formato, debido a que pese a su estandarización, *Microsoft* seguirá siendo restrictivo en cuanto a su aplicación en plataformas diferentes de *Microsoft Office* y *Windows*. Por tales razones se define dentro del proceso de migración y respaldado por la ISO/26300 como estándar abierto a ODF, por ser además, el formato por defecto de las suites ofimáticas libres Apache OpenOffice y LibreOffice.

1.3 Herramientas para la conversión de archivos ofimáticos

Debido a la contrariedad vigente en el tópico de los formatos OOXML y ODF en cuanto a su estandarización y uso, han surgido herramientas destinadas a la conversión de archivos ofimáticos, que contribuyen a que los usuarios puedan eludir las consecuencias del empleo de los formatos restrictivos. El estudio realizado se basa en la profundización de las características más significativas y funcionalidades de las herramientas de código abierto para la conversión de archivos ofimáticos, tomando como punto de partida, la conversión al formato ODF.

1.3.1 Odf-converter

La herramienta *Odf-converter* está orientada a la apertura de archivos generados con *Microsoft Office* 2007/2010, que incluye los formatos *Word*, *Power Point* y *Excel*, generando una conversión hacia el formato

² **ISO / IEC 29500** es un estándar para documentos de procesamiento de textos, presentaciones y hojas de cálculo que están destinadas a ser implementadas por múltiples aplicaciones en múltiples plataformas.

ODF. Utiliza el código de *Odf-converter*, que convierte los archivos OOXML a .odt, .ods y .odp. Es multiplataforma y se encuentra bajo la licencia GPL (10).

Observaciones:

Para comprobar el correcto funcionamiento de la herramienta, se realizaron pruebas a sus funcionalidades, que permitieron percibir que a pesar de brindar la opción de seleccionar la ruta de salida, constituye una desventaja para su uso la ausencia de una interfaz gráfica que permita al usuario un mejor entendimiento del sistema.

1.3.2 Unoconv

Esta herramienta convierte entre cualquier formato de documento soportado por Apache OpenOffice, como los documentos, hojas de cálculos, presentaciones y gráficos. Su licencia es GPL, es compatible con Apache OpenOffice en GNU/Linux y *Windows* y su interfaz es a través de líneas de comando, por su acrónimo en inglés de *Command Line Interface* (CLI, por sus siglas en inglés) (11).

Observaciones:

Luego de probar las funcionalidades que ofrece la herramienta se pudo constatar que no posee la opción de seleccionar la dirección de salida del documento. Carece además, de interfaz gráfica y posee funcionalidades mínimas para efectuar la conversión.

1.3.3 JODConverter

Es el convertidor de Java *OpenDocument*, automatiza todas las conversiones desde los formatos de *Microsoft Office* a ODF, incluyendo conversiones a otros formatos. Se puede utilizar como biblioteca de Java. Se ejecuta en cualquier sistema operativo entre ellos *Windows* y GNU/Linux donde Java y Apache OpenOffice estén disponibles. La herramienta *JODConverter* posee interfaz de tipo CLI y su licencia es GPL (12).

Observaciones:

Para evidenciar el correcto funcionamiento de la herramienta se realizaron pruebas a sus funcionalidades, en las que se corroboró que al igual que el *Unoconv*, el *JODConverter* carece de interfaz gráfica, no obstante, se puede utilizar como librería de Java incrustada en la propia aplicación.

1.3.4 Convertidor de documentos de LibreOffice y Apache OpenOffice

El asistente convierte documentos de *Microsoft Word*, *Excel* y *Power Point* al formato ODF de LibreOffice. Tiene varias páginas agrupadas en un asistente para convertir los documentos. En la primera, se especifica si se van a convertir documentos de *Microsoft Office*, así como el tipo de documento a convertir, en la segunda se define para cada tipo de plantilla y de documento, el directorio del que se debe leer y el directorio en el que se debe escribir y en la tercera se muestra un resumen en el que se especifica lo que se va a convertir al pulsar el

botón Convertir (13).

El convertidor de Apache OpenOffice, posee las mismas funcionalidades que el de LibreOffice, efectuando la conversión a través de los pasos anteriormente descritos.

Observaciones:

Luego de haber estudiado detenidamente las características y funcionalidades de los Convertidores de Documentos de Apache OpenOffice y LibreOffice. Se pudo comprobar que a pesar de ser los más completos, eficientes y poseer interfaz gráfica, para hacer uso de ellos se debe ejecutar previamente la suite ofimática a la que pertenecen, además, convierten todos los documentos de la carpeta, aunque no sea necesario.

Conversor de bases de datos mdbtools

La herramienta mdbtools permite la exportación de los esquemas de las bases de datos MS Access hacia archivos de texto plano (.sql), así como los campos y los valores hacia el archivo de texto plano (.csv). Su ejecución es a través de líneas de comando y se encuentra bajo la licencia GPL. La herramienta solo permite la exportación de .mdb a .csv y .sql.

1.3.6 Comparación de las herramientas de conversión

En aras de lograr una mejor visión de las potencialidades que ofrece cada solución estudiada, la Tabla 1 muestra una comparativa entre las mismas teniendo en cuenta las funcionalidades principales que se necesitan en la solución final.

| Funcionalidad | JODConvert er | Unoconv | ODF- Converter | Convertidor de documentos de OpenOffice | Convertidor de documentos de LibreOffice | mdbtools |
|---|------------------|---------|-------------------|---|--|----------|
| Selección del formato de salida | X | X | X | X | X | 0 |
| Selección de la dirección de salida | 0 | 0 | X | X | X | 0 |
| Conversión del archivo de extensión docx a odt | X | X | X | X | X | 0 |
| Conversión del archivo de extensión xlsx a ods | X | X | X | X | X | 0 |
| Conversión del archivo de extensión pptx a odp | X | X | X | X | X | 0 |
| Conversión del archivo de extensión mdb a odb | 0 | 0 | 0 | 0 | 0 | X |
| Elegir Librería para la conversión | 0 | 0 | 0 | 0 | 0 | 0 |
| Detener la conversión | 0 | 0 | 0 | 0 | 0 | 0 |
| Realizar la conversión integral | 0 | 0 | 0 | X | X | 0 |

Tabla 1: Comparación de las funcionalidades de las herramientas

1.3. 7 Elección de las herramientas

Al efectuar la comparación entre las herramientas de conversión anteriormente analizadas, con el objetivo de

definir cuál usar para el desarrollo de la solución, se descarta el uso de los asistentes de Conversión de Apache OpenOffice y LibreOffice, debido a que poseen interfaz gráfica y se encuentran incrustados a las suites ofimáticas; de las restantes herramientas se opta por el uso del *JODConverter* ya que a pesar de ejecutarse por líneas de comando al igual que el *Unoconv* y el *Odf-converter*, puede ser usada como librería Java incrustada en la propia aplicación y el *Odf-converter* a partir de pruebas realizadas que emitieron como resultados conversiones con una calidad superior a la de *JODConverter* y el *Unoconv*, (Anexo 1). En dicho anexo se ilustran conversiones realizadas con las tres herramientas a un documento *Word* con maquetado, donde se observa como resultado para *JODConverter* y el *Unoconv*, la pérdida total del maquetado del documento, mientras que para *Odf-converter* la conversión se realiza de manera exitosa y por último se selecciona la herramienta *mdbtools* para la exportación de las bases de datos de extensiones *.mdb*.

1.4 Migración de las base de datos de Access hacia LibreOffice Base

El proceso de migrar las bases de datos resulta complejo, pues existen distintos tipos y es difícil crear una solución informática que resulte efectiva para todos los casos (14), por tal razón el estudio realizado se centra en la migración de las bases de datos de *Microsoft Access 2003* a LibreOffice Base.

LibreOffice Base es el componente de LibreOffice para la administración de bases de datos. Su motor es *HSQLDB*³ y está escrito en el lenguaje de programación Java, al igual que la Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés), por lo que se necesita tener correctamente instalado un *Java Runtime Environment* (JRE).

A través de un estudio realizado, se pudo confirmar la inexistencia de una solución informática que permita la migración completa de las bases de datos de *Microsoft Access* hacia LibreOffice Base. Los formularios, las consultas y los informes se encuentran programados en *Visual Basic* para Aplicaciones (VBA) definidos por un conjunto de macros destinadas a mejorar sus funcionalidades, proporcionando un subconjunto de comandos disponibles en VBA (15). El problema al migrar estos componentes, viene dado por la incompatibilidad de las API del editor de diálogos de OOO Basic con VBA, esto hace imposible la importación directa de las macros VBA (Anexo 2). Como se puede observar en la imagen del Anexo 2, la estructura de un *IF* o de un bucle *FOR* es la misma, en cuanto al *Basic* son compatibles, pues VBA y OOO Basic pertenecen a la familia de lenguajes de tipo *Basic*, sin embargo, los objetos, el acceso a los mismos, sus propiedades, eventos y métodos, son completamente diferentes. Por tal motivo la solución centrará sus resultados en la migración de las tablas y los datos de las bases de datos de *Access 2003* hacia LibreOffice Base, aprovechando su uso en gran parte de las empresas del país.

³**HSQLDB**: *HSQLDB* (HyperSQL base de datos) es el motor de base de datos relacional SQL escrito en Java. Dispone de un pequeño y rápido motor de base de datos, multihilo y transaccional con tablas en memoria y basadas en disco, es compatible con los modos de servidor incrustados.

1.4.1 Análisis de la estructura de una base de datos odb

Los documentos pertenecientes al formato ODF, constituyen básicamente, un conjunto de archivos que se integran en una carpeta comprimida. Las bases de datos .odb propias de dicho formato, se encuentran estructuradas por ficheros que cumplen una función única. Para visualizar el contenido de los mismos, se muestran a continuación los elementos más importantes presentes dentro del directorio Database, incluyendo los ficheros Log y Manifest.xml (ver figura 2).

Properties: contiene configuraciones relacionadas con la base de datos.

Script: almacena la definición de las tareas y otros objetos de la base de datos.

Log: contiene los cambios recientes realizados en la base de datos.

Data: almacena toda la información de las tablas.

Backup: es un comprimido con el último estado consistente del fichero data

Manifest.xml: describe la estructura del archivo odb.

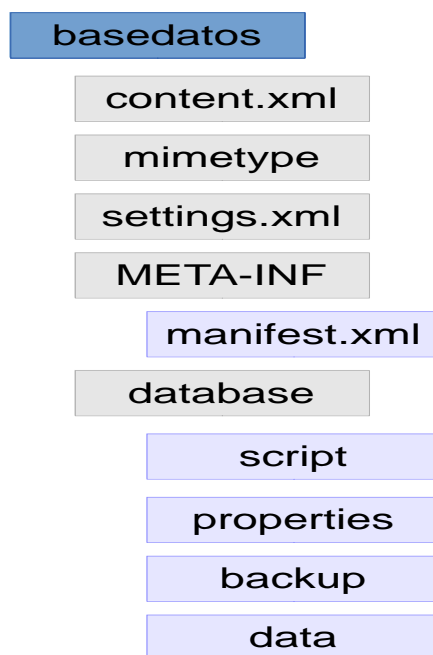


Ilustración 1: Estructura de las bases de datos odb

1.5 Metodología de Desarrollo

La metodología definida para la elaboración de la herramienta para la conversión de archivos ofimáticos es SXP, ya que se aplica en el proyecto SIMAYS del cual forma parte la solución de la investigación realizada, además, su propósito consiste en la programación rápida partiendo de iteraciones incrementales con ciclos

cortos, fomentando el desarrollo de la creatividad y responsabilidad entre los miembros del equipo.

SXP consta de cuatro fases principales:

Planificación-Definición: se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

Desarrollo: se realiza la implementación del sistema hasta que esté listo para ser entregado.

Entrega: se despliega y se pone en marcha.

Mantenimiento: se realiza el soporte para el cliente.

SXP es la metodología más apropiada para proyectos de corta duración con requisitos cambiantes o no bien definidos, donde prevalezca la retroalimentación entre el cliente y el equipo de trabajo. El desarrollo con SXP se realiza en iteraciones cortas (sprints) a lo largo de tres fases, dándole cumplimiento a un grupo de actividades, de las que se generan una serie de artefactos, que documentan el proceso de desarrollo, obteniendo un release del producto con nuevas funcionalidades (16).

1.6 Tecnologías empleadas en el desarrollo de la solución

El desarrollo de una solución informática que cumpla con los requisitos especificados para su elaboración, precisa la elección de las tecnologías necesarias para dar cumplimiento a dicho propósito. En el presente epígrafe se dará a conocer una breve síntesis de las características y ventajas que poseen los lenguajes, herramientas y tecnologías investigadas para llevar a cabo la implementación de la solución, además de exponer el motivo por el cual fueron utilizadas.

1.6.1 Lenguaje de programación

Para la elaboración de la herramienta de conversión de archivos ofimáticos, se define Java como lenguaje de programación, debido a que la herramienta *JODConverter* que será empleada en la solución, está desarrollada en el mismo, permitiendo una integración más sencilla a la aplicación a través de una librería. Otro punto a favor es que la documentación de desarrollo existente de LibreOffice se encuentra mejor fundamentada en este lenguaje, además, el componente de manejo de base de datos de LibreOffice Base (HSQLDB) empleado para la inserción de datos en la base de datos .odb está escrito en Java proporcionando una librería de fácil manejo (17).

1.6.2 Lenguaje de modelado

Para el modelado del sistema se define el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), con el fin de simplificar el complejo proceso de análisis de tal manera que facilite el plano para la construcción

de la propuesta de solución (18).

1.6.3 Base de datos

Luego de definir los lenguajes de programación y modelado que tributarán a la implementación de la aplicación, es necesario contar con un motor de bases de datos que permita la persistencia de la información asociada a los archivos que están en proceso de conversión, con el objetivo de poder pausar y reiniciar el proceso o reanudar la tarea de conversión ante fallas eléctricas o interrupción inesperada o deseada.

SQLite

SQLite es una pequeña librería programada en lenguaje C que implementa un completo motor de base de datos multiplataforma que no precisa configuración. Se distribuye bajo licencia de dominio público. Es muy rápido y la ventaja fundamental es que permite utilizar un amplio subconjunto del lenguaje estándar SQL (19). Se emplea en el desarrollo de la aplicación, ya que la base de datos perteneciente a la misma estará compuesta por una tabla donde se guardarán las direcciones de los archivos, por tanto es preciso un manejo rápido de los datos.

1.6 4 Librerías y aplicaciones externas

El uso de librerías constituye un elemento fundamental en aras de agilizar el desarrollo de las aplicaciones. Contienen código y datos que brindan servicios a programas independientes, permiten además, la reutilización de código, facilitando el perfeccionamiento de la solución. Para la elaboración de la solución se utilizaron un conjunto de librerías que facilitaron su implementación.

Hsqldb.jar

Hsqldb.jar es la librería de acceso al motor de bases de datos HSQLDB. Constituye un elemento fundamental para este, ya que es el JDBC y a su vez el motor de la base de datos. Los componentes de HSQLDB - el motor de la base de datos, los procesos del servidor, el controlador de JDBC y la documentación - son proporcionados en el paquete de *.jar, instalado en lib/hsqldb.jar* (20). La librería hsqldb.jar se empleó dentro de la solución para la inserción de datos en la base de datos .odb

Sqlitejdbc.jar

Sqlitejdbc.jar es una biblioteca destinada para el acceso y la creación de archivos de bases de datos SQLite en Java. Para el desarrollo fue necesario su uso, ya que se empleó una base de datos SQLite para la persistencia de los archivos.

JODConverter.jar

JODConverter es una biblioteca Java destinada a la conversión de documentos de oficina en los diferentes

formatos que usan LibreOffice y Apache OpenOffice. Es la librería que utiliza el *JODConverter* para establecer la conexión con la capa de servicios de LibreOffice, y se encuentra bajo la licencia LGPL (21).

Aplicaciones mdbtools y Odf-converter

Las soluciones *mdbtools* y *Odf-converter* definidas en las páginas 23 y 24 respectivamente, serán empleadas como librerías para el desarrollo de la solución.

1.6.4 Herramientas para el desarrollo del sistema

Una vez seleccionados los lenguajes y tecnologías destinadas al desarrollo del sistema, es preciso definir las herramientas necesarias que servirán de base para su construcción.

Netbeans:

Netbeans IDE (*Integrated Development Environment*) es un entorno de desarrollo distribuido bajo la doble Licencia CDDL (*Common Development and Distribution License*) y GPL versión 2 con *Classpath Exception*. Brinda soporte para PHP, JavaScript y características visuales para el desarrollo web. Es un producto libre, multiplataforma y gratuito sin restricciones de uso (22). Entre sus características principales se encuentran, la integración nativa con sistemas gestores de bases de datos, la integración con la herramienta de modelado UML *Visual Paradigm* y el soporte para la comunicación con el sistema de control de versiones *Subversion*, funcionalidad que fue usada para el versionado del código fuente (22).

Visual Paradigm para UML:

Visual Paradigm for UML (VP-UML) es una herramienta *CASE UML* diseñada para la ayuda al desarrollo de *software*. Soporta los principales estándares de la industria, entre ellos Lenguaje de Modelado Unificado (UML, por sus siglas en inglés). Su licencia es de dominio público y ofrece un conjunto de herramientas a los equipos de desarrollo de *software* necesario para la captura de los requisitos, *software* de planificación, la planificación de controles, el modelado de clases y modelado de datos. Proporciona un ambiente de modelado visual para satisfacer la tecnología del *software* y necesidades de comunicación. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario (23).

Subversion:

Subversion es un *software* de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es SWL bajo una licencia de tipo *Apache/BSD* y se le conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos. Todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. La capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración.

RapidSVN

RapidSVN es un cliente con interfaz gráfica para la comunicación con servidores *Subversion*. Está escrito en C++ y distribuido bajo licencia GPL. Este facilita el mantenimiento de las diferentes versiones de ficheros, desde una interfaz sencilla e intuitiva, y se encuentra disponible para plataformas Windows, GNU/Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente.

Conclusiones del Capítulo

LibreOffice y Apache OpenOffice son las soluciones informáticas de oficina de código abierto más empleadas en la actualidad. Ambas utilizan el formato de archivos ODF para la persistencia de los documentos ofimáticos. Entre las soluciones seleccionadas para la conversión ofimática están *Odf-converter* y *JODConverter*, este último desarrollado en Java, razón –entre otras- por la que este lenguaje de programación es el seleccionado para el desarrollo del sistema. La migración de los datos provenientes de archivos mdb hacia LibreOffice Base se apoyará en el uso de la herramienta *mdbtools* y la librería *hsqldb.jar*. La migración de formularios, consultas e informes no está soportada por la incompatibilidad entre la implementación de *Visual Basic* para *Microsoft Office* y LibreOffice.

Capítulo 2: Propuesta de solución

En el presente capítulo se concreta la estructura de la propuesta de solución, a partir de un conjunto de requisitos técnicos y operacionales definidos. Además, se precisan puntos claves dentro de la arquitectura del sistema como son: la lista de reserva del producto, las historias de usuario, los patrones arquitectónicos, las consideraciones de diseño y la distribución de componentes.

2.1 Descripción de la solución

La migración de los documentos a la nueva plataforma instalada, constituye uno de los procesos fundamentales dentro de la migración a SWL. La herramienta de conversión de archivos ofimáticos deberá automatizar las conversiones de los documentos de extensiones docx, xlsx, pptx y mdb hacia odt, ods, odp y odb respectivamente, de manera fácil e intuitiva, garantizando que al ejecutar la aplicación el usuario pueda añadir los documentos del formato a convertir así como eliminarlos, elegir la librería para efectuar la conversión, además de iniciarla o pausarla. El proceso comienza una vez ejecutada la herramienta, el mismo tiene como entrada los documentos a convertir y como salida los archivos convertidos al formato respectivo. Para la concepción del sistema se toma como punto de partida el modelo de dominio, con el objetivo de proporcionar una primera visión del sistema a desarrollar (Anexo 3).

Partiendo de la descripción de la solución propuesta y con una visión general del sistema a implementar, es necesario definir, mediante las necesidades del cliente, cuáles serán las características y cualidades a cumplir, por la propuesta de solución.

2.2 Requisitos del sistema.

Los requisitos del sistema incluyen el conjunto de tareas que conducen a comprender cuál será el impacto del *software* sobre el negocio, qué es lo que el cliente quiere y cómo interactúan los usuarios finales con el *software* (24). Como resultado de esta actividad se obtiene el artefacto Lista de Reserva del Producto que a continuación se detalla.

2.2.1 Lista de Reserva del Producto

La Lista de Reserva del Producto está compuesta por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Esta lista puede crecer y editarse a medida que se obtienen más conocimientos acerca del producto, con la restricción de que sólo puede cambiarse entre iteraciones, para lo cual se definen los requisitos funcionales y los requisitos no funcionales.

Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir. Se emplean para

especificar los principales servicios que el sistema proporciona. Determinan el comportamiento de la aplicación ante entradas y salidas específicas sin alterar las funcionalidades. En la siguiente tabla se recogen en cada una de las filas los requisitos identificados del producto. Se muestra el número identificativo del requisito, la descripción, el tiempo estimado para su cumplimiento y el nombre del estimador.

Los requisitos no funcionales responden a propiedades o cualidades que el *software* debe tener para que se pueda comportar de manera atractiva, confiable y segura. Suelen estar vinculados con los requisitos funcionales y son esenciales para que el producto alcance el éxito deseado (25).

Para la confección de la Lista de Reserva del Producto, el equipo de desarrollo definió la prioridad de las funcionalidades según las necesidades del cliente, teniendo en cuenta la complejidad de su implementación. A continuación se muestra una tabla donde se describen las características y cualidades presentes en el sistema, divididas en 8 requisitos funcionales y 7 no funcionales, para un total de 14 requisitos.

| Prioridad | Ítem * | Descripción | Estimación | Estimado por |
|-----------------|--------|--|------------|--------------|
| Muy Alta | | Convertor de documentos | | |
| | 1 | Convertir los archivos de formato <i>Microsoft Office Access</i> (mdb) a <i>Open Document Base</i> (.odb). | 2 semanas | María Leisy |
| Alta | | | | |
| | 2 | Convertir los archivos de formato <i>Microsoft Office Word</i> (docx) a <i>Open Document Text</i> (.odt). | 1 semana | Madeleinis |
| | 3 | Convertir los archivos de formato <i>Microsoft Office Power Point</i> (pptx) a <i>Open Document Presentation</i> (.odp). | 1 semana | María Leisy |
| | 4 | Convertir los archivos de formato <i>Microsoft Office Excel</i> (xlsx) a <i>Open Document Spreadsheet</i> (.ods). | 1 semana | Madeleinis |
| | 5 | Ubicar el archivo convertido en la ruta especificada. | 1 semana | María Leisy |

| Media | | | | |
|--------|--|--|------------|-------------|
| | 6 | Añadir archivos a la lista de archivos a convertir | 0,5 semana | María Leisy |
| | 7 | Eliminar archivos de la lista de archivo | 0,5 semana | Madeleinis |
| Baja | | | | |
| | 8 | Listar archivos a convertir. | 0,5 semana | Madeleinis |
| Ítem * | Requisitos no funcionales | | | |
| 9 | La herramienta deberá tener una interfaz sencilla, de fácil manejo y con un diseño agradable a la vista del usuario. | | | |
| 10 | La herramienta deberá ser intuitiva, los iconos serán representativos y deberá estar equilibrada en cuanto a colores y organización. | | | |
| 11 | Deben utilizarse tecnologías libres para el desarrollo del sistema. | | | |
| 12 | Usar como lenguaje de programación JAVA. | | | |
| 13 | Los desarrolladores del departamento SIMAYS le proporcionarán soporte y seguimiento a la herramienta. | | | |
| 14 | Utilizar como servidor de bases de datos SQLite. | | | |

Tabla 2: Requisitos del sistema

2.3 Historias de Usuario

Una vez definidas las características y cualidades que han de estar presentes en el sistema, se prosigue con la elaboración de las Historias de Usuario (HU) pertenecientes a las funcionalidades existentes en la aplicación. Las HU constituyen la técnica utilizada en SXP para especificar los requisitos del *software*, son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto (26).

Para dar cumplimiento al desarrollo de la aplicación se elaboraron las HU que a continuación se ilustran, clasificadas y organizadas según la prioridad que el cliente le atribuye a su implementación, se definió 1 HU de prioridad “Muy Alta”, 4 de “Alta” prioridad, 2 de prioridad “Media” y 1 de “Baja” prioridad, para un total de 8 HU.

Convertir los archivos de formato *Microsoft Office Access (mdb)* a *Open Document Database (.odb)*

| Historia de Usuario | |
|--|--|
| Número: 1 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Access(mdb)</i> a <i>Open Document Database(.odb)</i> |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: María Leisy | Iteración Asignada: 3 |
| Prioridad en Negocio: Muy Alta | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Muy Alto | Puntos Reales: 2 |
| Descripción: Deberá efectuar la importación de las tablas y los datos de las bases de datos de extensiones mdb para su exportación a odb. | |
| Observaciones: Solo se podrán convertir los archivos con extensiones mdb. | |
| Prototipo de Interfaz: (Ver anexo 4, Ilustración 16) | |

Tabla 3: HU_1

Convertir los archivos de formato *Microsoft Office Word (docx)* a *Open Document Text (odt)*

| Historia de Usuario | |
|--|---|
| Número: 2 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Word(docx)</i> a <i>Open Document Text(.odt)</i> |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: María Leisy | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: Deberá efectuar la importación de los documentos de extensiones docx, permitiendo la elección de la librería más idónea para su exportación a la extensión de archivo odt. | |
| Observaciones: El sistema posee la funcionalidad Conversión integral, que efectuará la importación de todas las extensiones de archivos incluyendo docx, permitiendo elegir la librería para su exportación a sus respectivos | |

| |
|--|
| equivalentes. |
| Prototipo de interfaz: (Ver anexo 4,Ilustración 16) |

Tabla 4: HU_2

Convertir los archivos de formato *Microsoft Office Power Point (pptx)* a *Open Document Presentation (.odp)*

| Historia de Usuario | |
|---|---|
| Número: 3 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Power Point(pptx)</i> a <i>Open Document Presentation (.odp)</i> |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Madeleinis | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: Deberá efectuar la importación de los documentos de extensiones pptx, permitiendo la elección de la librería más idónea para su exportación a la extensión de archivo odp. | |
| Observaciones: El sistema posee la funcionalidad Conversión integral, que efectuará la importación de todas las extensiones de archivos incluyendo pptx, permitiendo elegir la librería, para su exportación a sus respectivos equivalentes. | |
| Prototipo de interfaz: (Anexo 4,Ilustración 16) | |

Tabla 5: HU_3

Convertir los archivos de formato *Microsoft Office Excel (xlsx)* a *Open Document Spreadsheet (.ods)*

| Historia de Usuario | |
|--|---|
| Número: 4 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Excel(xlsx)</i> a <i>Open Document Spreadsheet(.ods)</i> |
| Modificación de Historia de Usuario Número: 1 | |
| Modificación de Historia de Usuario Número: 1 | Modificación de Historia de Usuario Número: 1 |
| Usuario: María Leisy | Iteración Asignada: 2 |

| | |
|--|----------------------------|
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | |
| Descripción: Deberá efectuar la importación de los documentos de extensiones xlsx, permitiendo la elección de la librería más idónea para su exportación a la extensión de archivo ods. | |
| Observaciones: La funcionalidad Conversión integral, que permitirá la conversión de todos la extensiones de archivos incluyendo la conversión de la extensión de archivo xlsx a ods. | |
| Prototipo de interfaz: (Anexo 4,Ilustración 16) | |

Tabla 6: HU_4

Ubicar el archivo convertido en la ruta especificada.

| Historia de Usuario | |
|---|--|
| Número: 5 | Nombre Historia de Usuario: Ubicar el archivo convertido en la ruta especificada. |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: Madeleinis | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: Permitirá al usuario, una vez añadido el archivo, especificar la ruta de destino a la que desee enviarlo luego de terminada la conversión | |
| Observaciones: Al añadir los archivos la herramienta adoptará la dirección de origen de estos por defecto. | |
| Prototipo de interfaz: (Anexo 4,Ilustración 17) | |

Tabla 7: HU_5

Añadir Archivos

| Historia de Usuario | |
|--|--|
| Número: 6 | Nombre Historia de Usuario: Añadir Archivos |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: María Leisy | Iteración Asignada: 1 |

| | |
|---|----------------------------|
| Prioridad en Negocio: Media | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Medio | Puntos Reales: 1 |
| Descripción: Una vez elegida la opción Añadir Archivos, aparecerá una ventana que permitirá añadir uno o varios archivos al sistema, registrándose automáticamente en la base de datos del sistema, finalizando el proceso una vez aceptada la elección. | |
| Observaciones: El sistema permitirá añadir un archivo o una carpeta de archivos, la acción se completará exitosamente solo si las extensiones de los archivos a adicionar pertenecen a las extensiones docx, pptx, xlsx y mdb, de no ser así, la aplicación lanzará una excepción. | |
| Interfaz de Usuario: (Anexo 4,Ilustración 18) | |

Tabla 8: HU_6

Eliminar Archivo

| Historia de Usuario | |
|--|---|
| Número: 7 | Nombre Historia de Usuario: Eliminar Archivo |
| Modificación de Historia de Usuario Número: 1 | |
| Usuario: María Leisy | Iteración Asignada: 1 |
| Prioridad en Negocio : Media | Puntos Estimados: 0,5 |
| Riesgo en Desarrollo : Medio | Puntos Reales: 0,5 |
| Descripción: Permitirá eliminar uno o varios archivos, incluyendo su registro en la base de datos, para ello el usuario deberá seleccionarlos previamente, la acción se completará una vez se confirme la eliminación de los documentos seleccionados. | |
| Observaciones: Para completar la acción se debe confirmar la eliminación. Se dará la opción de limpiar la lista de archivos, una vez ejecutada automáticamente se borrarán todos los archivos existentes en la aplicación, incluyendo su registro en la base de datos. | |
| Prototipo de interfaz: (Anexo 4,Ilustración 19) | |

Tabla 9: HU_7

Listar archivos a convertir

| Historia de Usuario | |
|--|--|
| Número: 8 | Nombre Historia de Usuario: Listar archivos a convertir. |
| Modificación de Historia de Usuario Número:1 | |
| Usuario: Madeleinis | Iteración Asignada:1 |
| Prioridad en Negocio: Baja | Puntos Estimados:0,5 |
| Riesgo en Desarrollo: Bajo | Puntos Reales:0,5 |
| Descripción: Mostrará un resumen de las propiedades de los archivos, dentro de las que se encontrarán el origen, tamaño y estado, este último podrá encontrarse en tres fases, convirtiendo, convertido y en espera. | |
| Observaciones: En caso de ser pausada la conversión, añadirá el estado pausado | |
| Prototipo de interfaz:(Anexo 4,Ilustración 19) | |

Tabla 10: HU_8

2.4 Consideraciones de diseño

Para lograr que el sistema cumpla con todas las funcionalidades previstas, es imprescindible definir un diseño que incluya una solución simple pero a la vez efectiva, que refleje a través de principios y buenas prácticas la intención de planificación del equipo de desarrollo. Para ello es necesario puntualizar la arquitectura de *software* que adoptará el sistema, la distribución de las clases y paquetes definidos para la misma y por último los patrones de diseño GRASP (*General Responsibility Assignment Software Patterns*, en español Patrones Generales de *Software* para la Asignación de Responsabilidades) y GOF (*Gang of Four*, en español Pandilla de Cuatro) que se emplearán en la estructura de la solución.

2.4.1 Arquitectura de software

El diseño arquitectónico representa la estructura de los datos y componentes del programa necesarios para construir un sistema computacional. Según Roger Pressman en la 6ta edición de su libro "Ingeniería de *software*. Un enfoque práctico", capítulo 10 epígrafe 10.1.1; la arquitectura es la forma en que se integran los diversos componentes del sistema para formar un todo cohesionado. Es la manera en que la solución se

amolda y combina con aplicaciones externas. Resumiendo, la arquitectura de *software* de un programa es la estructura del sistema que incluyen los componentes de *software*, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Permite a un ingeniero de *software* analizar la efectividad del diseño para cumplir con los requisitos establecidos, considerar opciones arquitectónicas en una etapa en la que aún resulta relativamente fácil hacer cambios al diseño y reducir los riesgos asociados a la construcción del *software* (27).

Para el desarrollo de la solución se adoptará una arquitectura en Capas, dada las potencialidades que ofrece para la reutilización y el aprovechamiento de los beneficios de una clara separación entre las funciones desplegadas en capas. Además, permite la estandarización y proporciona una distribución clara del trabajo entre los miembros de un equipo de desarrollo.

2.4.2 Patrón Arquitectónico

Una arquitectura de *software* diseñada en capas consiste en la definición de niveles de abstracción, los cuales tienen una función específica permitiendo un diseño modular. Esto implica la creación de sistemas con un bajo acoplamiento entre sus módulos o componentes. Una variante de este patrón muy utilizada es la de Tres Capas, mediante la cual se fracciona al sistema informático en la capa de Presentación, la capa de Lógica del Negocio y la capa de Acceso a Datos.

“La capa de Presentación es la que se encarga de interactuar con el usuario mediante la interfaz de usuario. La capa Lógica del Negocio, llamada también como Lógica de la Aplicación, se encarga de realizar las tareas para las cuales está concebido el sistema. Es implementada utilizando un modelo orientado a objetos del dominio de la aplicación. Se encarga de controlar las operaciones de acuerdo con las reglas del negocio. La capa de Acceso a Datos es la que gestiona el almacenamiento de los datos, ya sea en una base de datos o en un fichero, así como la consulta a los mismos” (28).

Una restricción de este patrón consiste en que las capas inferiores no deben de conocer ni hacer llamadas a procedimientos implementados en capas superiores, sino que las funcionalidades que ellas ofrecen son accedidas desde niveles mayores.

Características del patrón:

- Descomposición de los servicios, de manera que la mayoría de las interacciones se realizan sólo entre capas vecinas.
- Los componentes de cada capa se comunican con los componentes de otras capas, a través de interfaces bien conocidas.
- Cada nivel agrega las responsabilidades y abstracciones del nivel inferior.

- Muestra una vista completa del modelo y al mismo tiempo brinda suficientes detalles para entender las relaciones entre capas.
- Separa claramente la funcionalidad de cada capa (29).

Al definirse la arquitectura global de la aplicación, se procede a la confección del diagrama de paquetes correspondiente a la misma, a partir de una explicación detallada de las relaciones existentes entre las clases que lo conforman.

2.4.3 Modelo de Diseño

El modelo de diseño proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes que son necesarios para implementar el sistema, además de estabilidad, control y organización de las actividades que se producen dentro del ciclo de vida del *software*. El modelo de diseño representa los requisitos funcionales en un lenguaje técnico lo más cercano posible al lenguaje de programación y sirve como una abstracción del código fuente (30).

Para una mejor comprensión de la solución y de acuerdo con la arquitectura de tres capas se han organizado las clases en tres paquetes principales: Presentación, Lógica del Negocio y Acceso a Datos. Cada uno de los paquetes se ha dividido a la vez en sub-paquetes de acuerdo con las funcionalidades de las clases contenidas. Cada paquete representa una capa de la arquitectura y está construido sobre su predecesor. Las clases pertenecientes a una capa están relacionadas con las clases de la capa inmediata inferior y desconocen a las clases de las capas superiores. A continuación se muestra cómo fue diseñado el sistema en términos de clases y paquetes:

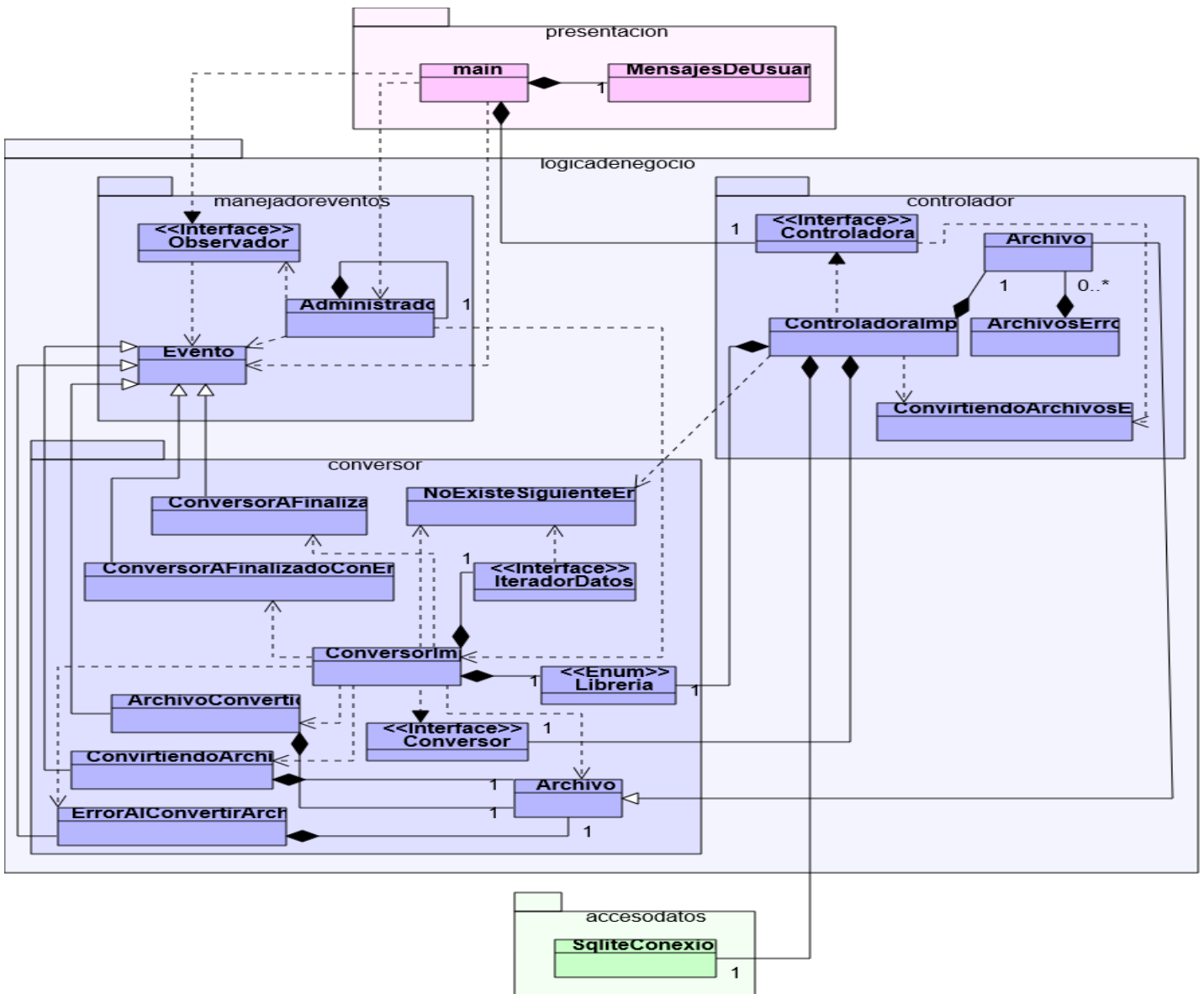


Ilustración 2: Diagrama de componentes

Los elementos del modelo representados en el diagrama de clases se agrupan en cinco paquetes. Tres de ellos: presentación, controlador y acceso a datos, constituyen capas; los dos restantes: conversor y manejador de eventos, son elementos reutilizables. A continuación se describen dichos paquetes.

- **Presentación:** Sus elementos implementan los requerimientos para garantizar la interacción con el usuario (persona). Se relaciona con la capa controladora y con el paquete manejador de eventos, a través del cual es informado cuando ocurre un error de conversión o se convierte un archivo.
- **Controlador:** Coordina la conversión de archivos, implementa la lógica de negocio, atiende las solicitudes de la capa de presentación e interactúa con la capa de acceso a datos.
- **Acceso a datos:** Implementa el acceso a los datos, transforma el esquema de la base de datos a

objetos de la lógica de negocio.

- **Conversor:** Contiene los elementos para realizar la conversión de archivos, se puede ejecutar desde varios hilos, notifica los sucesos relacionados con la conversión y tiene las implementaciones para realizar conversiones con varias librerías.
- **Manejador de eventos:** Posibilita que interactúen de manera desacoplada la presentación con el conversor de archivos.

Para lograr una asignación correcta de las responsabilidades en las clases representadas en el diagrama de paquetes y un diseño adecuado de las relaciones entre sus objetos, es necesario tener en cuenta el uso de los patrones de diseño, garantizando organización y uniformidad en la implementación del sistema. A continuación de describen los patrones GRASP y GOF empleados en la solución.

2.4.4 Patrones GRASP

Los patrones GRAP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Las decisiones poco acertadas dan origen a sistemas y componentes frágiles, difíciles de mantener, entender, reutilizar o extender (31). Por tanto, el uso de los patrones GRASP se hace imprescindible para la elaboración de la solución. En el desarrollo de la misma se aplicaron los patrones GRASP: Experto, Creador, Controlador, Bajo Acoplamiento y Alta Cohesión.

Experto:

El patrón Experto soluciona el problema de asignar una responsabilidad de forma general, al recomendar para esta función a la clase que maneje la información necesaria dentro del sistema. A continuación se muestra un ejemplo del uso del mismo dentro de la aplicación

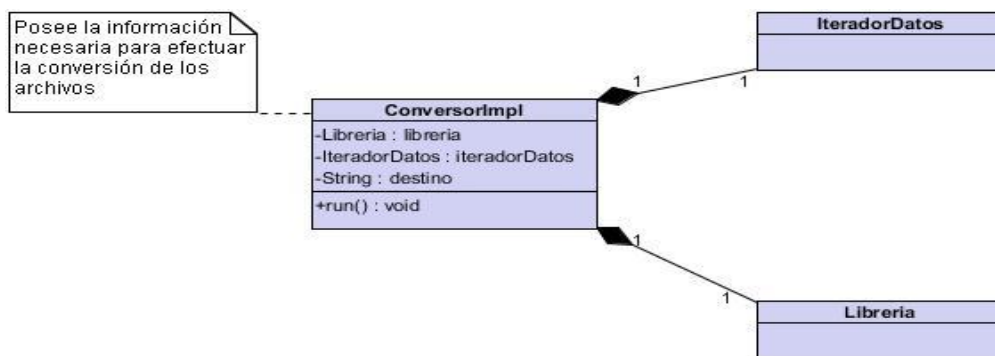


Ilustración 3: patrón Experto

En el diagrama anterior se muestra el método *run* implementado por la clase *ConversorImpl*, debido a que para ejecutar la acción es necesario conocer la librería, el destino y el archivo a convertir, información que solo

posee dicha clase.

Creador:

El patrón Creador trata el problema de qué clase debe ser la encargada de instanciar a otra. Seguidamente se muestra un ejemplo de su uso en la aplicación.

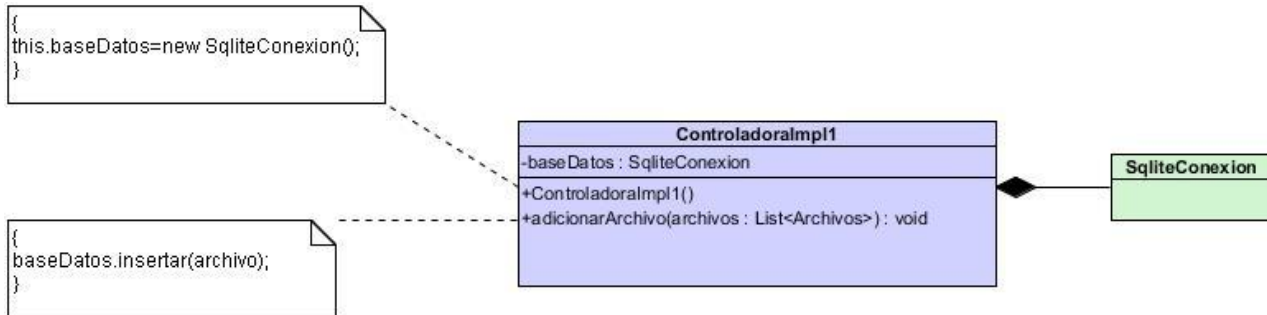


Ilustración 4: Patrón Creador

En el ejemplo se muestra que la clase *ControladorImpl1* tiene una relación de composición con *Sqlite*. Por tanto, *ControladorImpl1* se encarga de crear una instancia de *Sqlite* asignándole valores. En este caso el patrón se fundamenta en la relación A contiene a B.

Controlador:

El patrón Controlador ayuda a decidir quién se encarga de administrar un evento del sistema. En el diseño de la aplicación se pone de manifiesto a través de la clase *Main*. Esta es la fachada global del sistema (ventana principal). Basándose en el patrón controlador se le dio la responsabilidad de implementar las respuestas a todos los eventos que el usuario emita a través de la interfaz gráfica.

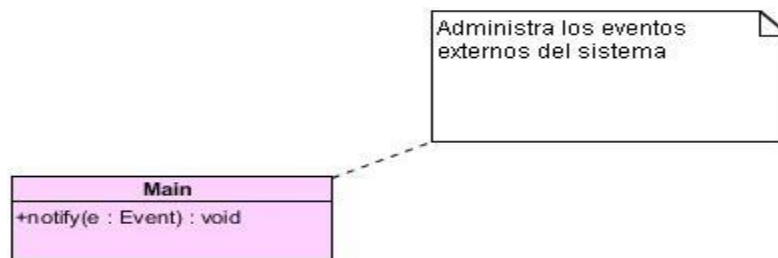


Ilustración 5: Patrón Controlador

Bajo Acoplamiento:

El Bajo Acoplamiento recomienda asignar las responsabilidades de tal forma que se le dé soporte a una mayor reutilización y poca dependencia. Para promover el patrón en el diseño de la aplicación, se emplea el uso de interfaces que rijan la interacción entre los componentes. A continuación se ilustra un ejemplo del uso del patrón a través de un diagrama de clases.

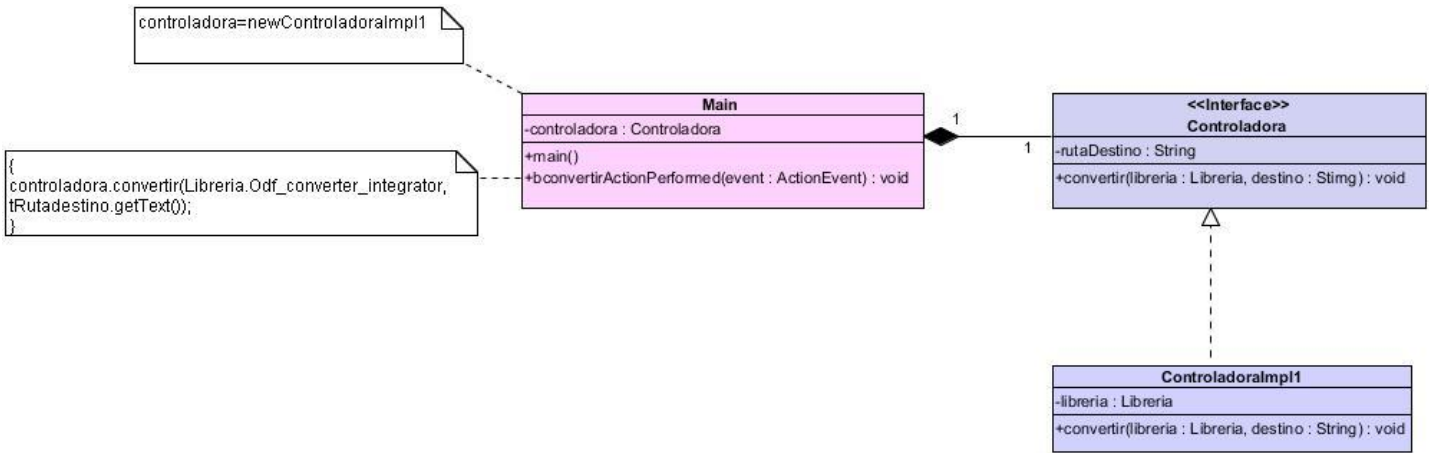


Ilustración 6: patrón Bajo Acoplamiento

Alta Cohesión:

El patrón Alta Cohesión aconseja mantener la clase lo más cohesionada posible para controlar la complejidad de la misma. A continuación se muestra un ejemplo del uso del patrón dentro de la aplicación.

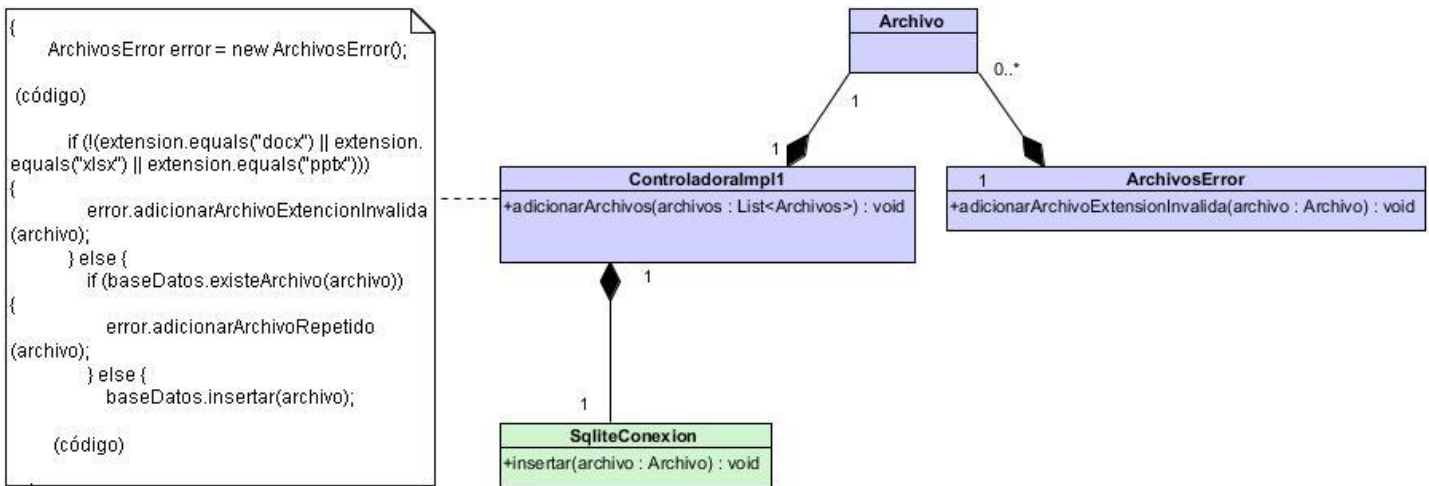


Ilustración 7: patrón Alta Cohesión

En el ejemplo anterior se muestra el método *adicionarArchivo(archivos:List<Archivos>)* de la clase *ControladoraImpl1*. El mismo hace llamadas a los métodos *AdicionarArchivoExtensionInvalida(archivo:Archivo)* y al *insertar(archivo:Archivo)* que se encuentran en *ArchivosError* y *SqliteConexion*, con el fin de evitar el exceso de complejidad en la misma.

2.2.5 Patrones GOF

Los patrones de diseño GOF constituyen un conjunto de reglas que describen cómo afrontar tareas y

solucionar problemas que surgen durante el desarrollo del *software*. Para la implementación de la solución se empleó el patrón Singleton.

Singleton pertenece al conjunto de patrones creacionales, encargados de crear objetos, de manera que los desarrolladores no tengan que instanciar directamente, proporcionando a los programas una mayor flexibilidad para decidir qué objetos usar. Garantiza que solo exista una instancia y que se pueda acceder a ella desde cualquier clase. Para ello, en lugar de tener una variable global, la instancia se almacena en un atributo estático de la clase y se accede a ella por el método *getInstance* (32). El patrón se pone de manifiesto dentro la aplicación a través de la clase *Administrador*, que define el método estático *getInstance* (), lo que proporciona visibilidad global a esta única instancia, de manera tal que en cualquier punto del código, se podrá escribir *Administrador.getInstance* para obtener acceso a la instancia. A continuación se muestra un ejemplo del uso del patrón dentro de la solución.

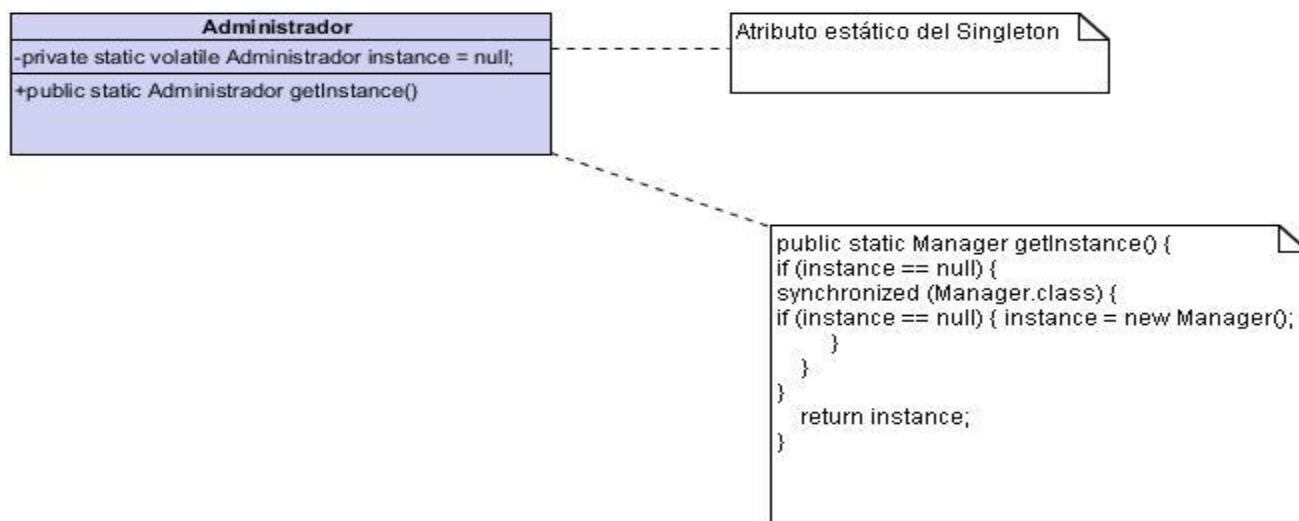


Ilustración 8: patrón Singleton

2.4.6 Exigencias mínimas para el funcionamiento del sistema

Las exigencias mínimas de hardware y *software* para la instalación, configuración y ejecución del sistema son:

- Para lograr un mejor rendimiento de la aplicación se recomienda que la máquina tenga como mínimo un 1GB de memoria RAM.
- Se deberá tener instalada la Máquina Virtual de Java 7.0.
- Se deberá instalar previamente la herramienta *Odf-converter_strawberry_0.2.3-2_i386.deb*.
- Es necesaria la instalación previa del LibreOffice (para poder convertir usando la librería *JODConverter*).
- Se deberá instalar previamente el paquete *mdbtools*.

Conclusiones del Capítulo

Luego de analizar y diseñar el sistema, teniendo en cuenta las características y cualidades que debe cumplir el mismo según las necesidades del cliente, se cuenta con una arquitectura en capas en la que se organiza el código de la aplicación. Para el diseño del producto se modelaron 22 clases, aplicando buenas prácticas de programación haciendo uso de los patrones GRASP y GOF.

Capítulo 3: Implementación y prueba

Durante el desarrollo del capítulo se abordará la implementación del sistema haciendo énfasis en la planificación de las iteraciones del ciclo de desarrollo. Para ello se argumentarán aspectos tales como: las tareas de ingeniería y el plan de release, que forman parte del grupo de artefactos resultantes de las actividades que se llevan a cabo durante el ciclo de desarrollo del *software*. Se presentarán los métodos más significativos que posee el sistema, dando una descripción de cada uno de ellos, además de definir los componentes reutilizables.

Un elemento fundamental para lograr la organización durante el proceso de desarrollo del *software* lo constituye la planificación, proporcionando un marco de trabajo que permite estimar los recursos, costos y planificación temporal. Dentro de la metodología que se utiliza, la planificación se manifiesta mediante iteraciones, planeando el trabajo a realizar en cada una de ellas.

3.1 Plan de Release

De acuerdo con las funcionalidades descritas en las HU y en concordancia con la prioridad asignada para su realización, se planificaron 3 iteraciones que abarcan el desarrollo de la aplicación.

| Release | Descripción de la iteración | Orden de la HU a implementar | Duración total |
|---------|---|------------------------------|----------------|
| 1 | Desarrollo de las HU de media y baja prioridad. | HU_6, HU_7, HU_8 | 4 semanas |
| 2 | Desarrollo de las HU de alta prioridad. | HU_2, HU_3, HU_4, HU_5 | 4 semanas |
| 3 | Desarrollo de las HU de Muy alta prioridad | HU_1 | 2 semanas |

Tabla 11: Plan de Release

A partir de las iteraciones definidas para la implementación de las HU que se muestran en el Plan de Release, es imprescindible precisar las tareas que se desempeñarán para lograr el objetivo de las mismas.

3.2 Tareas de Ingeniería

Las tareas de ingeniería constituyen una definición detallada de las HU, facilitando el entendimiento del proceso de implementación. Cada HU puede contener una o más tareas en caso de necesitarla, explicando paso a paso las acciones que se realizan en la misma. Los puntos de estimación asignados a cada tarea son medidos por días según las características y la complejidad que posea la misma. Permiten además organizar el proceso de implementación, así como conocer el grado de complejidad de cada HU, teniendo en cuenta la cantidad de tareas asociadas (33).

A continuación se exponen 7 Tareas de Ingeniería pertenecientes a las HU más significativas, las tareas restantes se encuentran detalladas en el (Anexo 5), para un total de 20.

Tareas de Ingeniería para las: HU_2, HU_3, HU_4

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 1 | Número Historia de Usuario: HU_2, HU_3, HU_4 |
| Nombre Tarea: Estudio de la librería <i>JODConverter</i> . | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 1/12/2012 | Fecha Fin: 1/01/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: La librería <i>JODConverter</i> permite una conversión rápida de los archivos existentes en el sistema. Se instala la librería y se prueban las funcionalidades que brinda. | |

Tabla 12: Tarea de Ingeniería_1

Tareas de Ingeniería para las: HU_2, HU_3, HU_4

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 2 | Número Historia de Usuario: HU_2, HU_3, HU_4 |
| Nombre Tarea: Estudio de la librería <i>Odf-converter</i> . | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio:1/12/2012 | Fecha Fin:1/01/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |

Descripción: La herramienta *Odf-converter* será utilizada como librería por defecto para la conversión de los archivos, ya que realiza el proceso con mayor calidad.

Tabla 13: Tarea de Ingeniería_2

Tareas de Ingeniería para la HU_Conversión de docx a odt

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 6 | Número Historia de Usuario: HU_2 |
| Nombre Tarea: Diseño de la interfaz para la conversión de archivos de formato <i>Microsoft Office Word(docx)</i> a <i>Open Document Text(.odt)</i> | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 05/02/2013 | Fecha Fin: 10/02/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: El usuario podrá seleccionar el tipo de librería a través del <i>jbutton</i> Configuración y ejecutar la conversión de las extensiones de archivo docx, xlsx, pptx y mdb a odt, ods, odp y odb, con el <i>jbutton</i> Convertir. | |

Tabla 14: Tarea de Ingeniería_6

Tareas de Ingeniería para la HU_Conversión de pptx a odp

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 7 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Diseño de la interfaz para la conversión de archivos de formato <i>Microsoft Office Power Point(pptx)</i> a <i>Open Document Presentation(.odp)</i> | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 10/02/2013 | Fecha Fin: 15/02/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: El usuario podrá seleccionar el tipo de librería a través del <i>jbutton</i> Configuración y ejecutar la conversión de las extensiones de archivo docx, xlsx, pptx y mdb a odt, ods, odp y odb, con el <i>jbutton</i> Convertir. | |

Tabla 15: Tarea de Ingeniería_7

Tareas de Ingeniería para la HU_Conversión de xlsx a ods:

| Tarea de Ingeniería | |
|------------------------|---|
| Número Tarea: 8 | Número Historia de Usuario: HU_4 |

| | |
|--|------------------------------|
| Nombre Tarea: Diseño de la interfaz para la conversión de archivos de formato <i>Microsoft Office Excel(xlsx)</i> a <i>Open Document Spreadsheet(.ods)</i> | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 15/02/2013 | Fecha Fin: 20/02/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: El usuario podrá seleccionar el tipo de librería a través del <i>jbutton</i> Configuración y ejecutar la conversión de las extensiones de archivo docx, xlsx, pptx y mdb a odt, ods, odp y odb, con el <i>jbutton</i> Convertir. | |

Tabla 16: Tarea de Ingeniería_8

Tareas de Ingeniería para la HU_Ubicar archivo en la ruta especificada

| | |
|---|---|
| Tarea de Ingeniería | |
| Número Tarea: 9 | Número Historia de Usuario: HU_5 |
| Nombre Tarea: Diseño de la interfaz para la funcionalidad ubicar el archivo convertido en la ruta especificada | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 05/04/2013 | Fecha Fin: 10/04/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: La interfaz permitirá al usuario a través de un <i>jbutton</i> definir la dirección de salida del archivo convertido. | |

Tabla 17: Tarea de Ingeniería_9

Tareas de Ingeniería para la HU-Conversion de mdb a odb

| | |
|---|---|
| Tarea de Ingeniería | |
| Número Tarea: 17 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Estudio de las base de datos mdb | |
| Tipo de Tarea: Investigación | Puntos Estimados: 1 |
| Fecha Inicio: 15/12/2012 | Fecha Fin: 22/05/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: Se analizan los diferentes métodos y herramientas existentes para la conversión de bases de datos del formato <i>Microsoft Office Access(mdb)</i> a <i>Open Document Database(.odb)</i> | |

Tabla 18: Tarea de Ingeniería_17

Tareas de Ingeniería para la HU-Conversion de mdb a odb

| | |
|---|---|
| Tarea de Ingeniería | |
| Número Tarea: 18 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Diseño de la interfaz para la conversión de archivos de formato <i>Microsoft Office Access(mdb)</i> a <i>Open Document Database(.odb)</i> | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.4 |
| Fecha Inicio: 25/05/2013 | Fecha Fin: 27/06/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: Al ejecutar la interfaz añadir archivo permitirá al usuario a través de un <i>jbutton</i> , añadir los archivos necesarios para ejecutar la conversión de los esquemas y los datos de las bases de datos mdb hacia odb. | |

Tabla 19: Tarea de Ingeniería_18

Tareas de Ingeniería para la HU-Conversion de mdb a odb

| | |
|--|---|
| Tarea de Ingeniería | |
| Número Tarea: 20 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Desarrollo de la librería <i>mdbaOdb</i> | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.4 |
| Fecha Inicio: 29/05/2013 | Fecha Fin: 1/06/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: Se implementó una librería que permite la exportación de los datos y los esquemas de las bases de datos de extensiones mdb hacia bases de datos odb. | |

Tabla 20: Tarea de Ingeniería_19

3.3 Estándar de Codificación

Para el cumplimiento de las Tareas de Ingenierías trazadas por el equipo de desarrollo, es de vital importancia que exista un entendimiento entre los programadores y el código de la aplicación. Por tal motivo, se definen los estándares de codificación, que constituyen una de las buenas prácticas propuesta por la metodología SXP,

asegurando que el código exprese claramente el propósito del mismo y agilice el proceso de refactorización, que sea legible, entendible y refleje un estilo armonioso (34). Para la implementación de la solución propuesta se utilizó el estándar de código CamelCase.

3.4 Componentes definidos para el sistema

Otro eslabón importante dentro del desarrollo de la aplicación, consiste en lograr una correcta organización de los componentes definidos para la solución, detallados a través de los diagramas de componentes, que escriben los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de *software* que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros (35). A continuación se presenta el diagrama de componentes del sistema

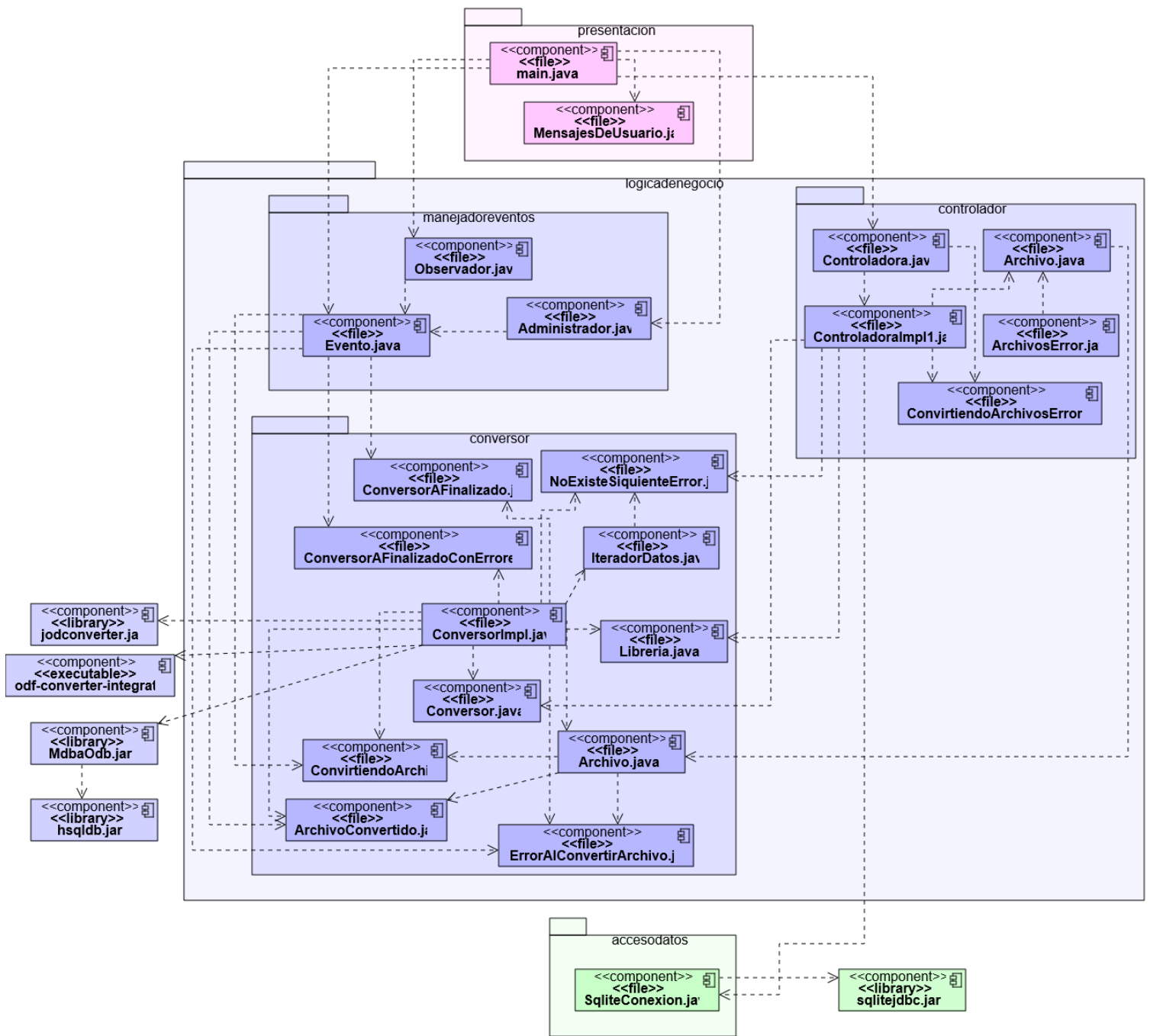


Ilustración 9: Diagrama de componentes

Para la conversión de los archivos de base de datos de extensión .mdb se implementó la librería *MdbaOdb*. El proceso de conversión se realiza de la siguiente forma:

- Se exporta la estructura de la base de datos .mdb a un fichero .sql.
- Se exporta el contenido de cada tabla de la base de datos .mdb a un fichero .csv.

- Se crea una nueva base de datos .odb.
- Para replicar la estructura del fichero .sql antes mencionado y la información recogida en el .csv se realiza una conexión a la nueva base de datos creada utilizando la librería *hsqldb.jar*, seguidamente se crea la estructura de la base de datos leyendo el fichero .sql que contiene el esquema y se insertan los datos contenidos en el fichero .csv.

3.5 Descripción de los métodos del sistema

Las clases definidas para la aplicación se encuentran agrupadas por asignación de responsabilidades, partiendo de la arquitectura y los patrones de diseño que presenta el sistema, lo cual posibilita la implementación de los métodos de manera tal que cada cual cumpla con su responsabilidad. A continuación se presenta una descripción de los métodos más significativos presentes en la aplicación.

ConversorImpl del paquete conversor

| Método | Descripción | Parámetros | Valor de retorno |
|--|--|-----------------|------------------|
| Run | De acuerdo a la librería seleccionada ordena la conversión invocando a los métodos <code>convertirJODConverter</code> , <code>convertirOdfConverterIntegrator</code> o <code>convertirMdbAODB</code> | - | Void |
| <code>convertirJODConverter</code> | Realiza la conversión con la librería <code>JODConverter</code> | Archivo archivo | Void |
| <code>convertirOdfConverterIntegrator</code> | Realiza la conversión con la librería <code>ODFConverterIntegrator</code> | Archivo archivo | Void |
| <code>convertirMdbAODB</code> | Realiza la conversión con la librería <code>MdbaOdb</code> | Archivo archivo | Void |

Tabla 21: Descripción de los métodos de la clase_Convertir

Archivo del paquete Conversor

| Método | Descripción | Parámetros | Valor de retorno |
|----------------------------------|---|------------|------------------|
| <code>obtenerExtensionODF</code> | Reconoce el formato OOXML y se los asigna a su derivado en ODF, | | String |

además de la extensión
mdb.

Tabla 22: Descripción de los métodos de clase_Archivo

SqliteConexión del paquete accesodatos

| Método | Descripción | Parámetros | Valor de retorno |
|----------------------|---|-----------------|------------------|
| SqliteConexion | | - | Void |
| Desconectar | | - | Void |
| Insertar | Inserta el origen y la longitud de los archivos en la base de datos | Archivo archivo | Void |
| Obtener | Obtiene el origen y la longitud de los archivos en la base de datos | - | Archivo archivo |
| Eliminar | Elimina los archivos de la base de datos | -String camino | Void |
| existeArchivo | Verifica si existe el camino del archivo dado. | Archivo archivo | Boolean |
| obtenerPrimerArchivo | Obtiene el primer archivo de la tabla. | - | Archivo archivo |
| Limpiar | Borra los elementos existentes en la base de datos | - | Void |

Tabla 23: Descripción de los métodos de clase_AccesoADatos

3.6 Pruebas

Una vez implementadas las HU definidas para el sistema, se transita hacia la fase de pruebas con el propósito de obtener una solución libre de errores. Durante su desarrollo, la aplicación fue sometida a un conjunto de casos de pruebas, a partir de las iteraciones definidas, con el objetivo de garantizar su buen funcionamiento de acuerdo con las HU presentes en el sistema.

3.6.1 Casos de pruebas de Aceptación

El objetivo de las pruebas de aceptación es demostrar que un sistema cumple con el funcionamiento esperado

y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente (36).

Para llevar a cabo la ejecución de las pruebas de aceptación definidas para el sistema se planificó un encuentro con el cliente, para garantizar la conformidad del mismo con las funcionalidades del sistema. En dicho proceso se describieron los criterios de evaluación para las pruebas, como clasificación de las No Conformidades, Pedidos de cambios y listas de chequeo. A continuación se muestran los casos de Pruebas de Aceptación definidos para las HU existentes en la aplicación.

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: 1 | Nombre Historia de Usuario: Añadir Archivos |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad añadir archivo | |
| Condiciones de Ejecución: El archivo a insertar deberá ser un documento de texto. | |
| Entrada / Pasos de ejecución: Para añadir archivos a la aplicación, el usuario deberá seleccionar la opción de añadir archivos, en la cual podrá añadir uno o más archivos, que posean solo las extensiones docx, xlsx, pptx y mdb. | |
| Esperado: De no ser un documento de texto el archivo no se insertará y la aplicación lanzará una excepción informando el error. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 24: Caso de Prueba de Aceptación_1

| Caso de Prueba de Aceptación | |
|--|---|
| Código Caso de Prueba: 2 | Nombre Historia de Usuario: Listar archivos a convertir. |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad listar archivos a convertir. | |
| Condiciones de Ejecución: Se mostrarán en una lista todos los archivos que se encuentren dentro de la aplicación. | |
| Entrada / Pasos de ejecución: Listar los archivos presentes dentro de la aplicación dará información al usuario de la cantidad de archivos a convertir, el tipo de conversión que se efectuará y el estado de conversión del documento. | |

| |
|--|
| Esperado: En la lista se reflejará el estado del documento: en espera, convirtiendo, convertido y pausado |
| Evaluación de la Prueba: Satisfactoria |

Tabla 25: Caso de Prueba de Aceptación_2

| Caso de Prueba de Aceptación | |
|---|--|
| Código Caso de Prueba: 3 | Nombre Historia de Usuario: Eliminar Archivos |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad eliminar archivo | |
| Condiciones de Ejecución: El documento deberá encontrarse en la lista de archivos de la aplicación. | |
| Entrada / Pasos de ejecución: Para eliminar archivos de la aplicación, el usuario deberá seleccionar la opción de eliminar archivos, en la cual podrá eliminar uno limpiar la lista de archivos. | |
| Esperado: Si el archivo a eliminar se encuentra en el estado convirtiendo, aparecerá una notificación preguntando al usuario si está seguro de realizar la acción. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 26: Caso de Prueba de Aceptación_3

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: 4 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Word(docx)</i> a <i>Open Document Text(.odt)</i> |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad convertir los archivos de formato Microsoft Office Word(docx) a Open Document Text(.odt) | |
| Condiciones de Ejecución: El archivo a convertir deberá tener la extensión docx. | |
| Entrada / Pasos de ejecución: El usuario deberá elegir la opción de añadir archivo y luego iniciar la conversión. | |
| Esperado: Al finalizar la conversión la aplicación enviará una notificación. Si ocurre algún error durante la conversión se lanzará un mensaje de error. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 27: Caso de Prueba de Aceptación_4

| Caso de Prueba de Aceptación | |
|---------------------------------|---|
| Código Caso de Prueba: 5 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Power Point(pptx)</i> a <i>Open Document</i> |

| | |
|--|--------------------|
| | Presentation(.odp) |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad convertir los archivos de formato Microsoft Office Power Point(pptx) a Open Document Presentation(.odp) | |
| Condiciones de Ejecución: El archivo a convertir deberá tener la extensión pptx. | |
| Entrada / Pasos de ejecución: El usuario deberá elegir la opción de añadir archivo y luego iniciar la conversión. | |
| Esperado: Al finalizar la conversión la aplicación enviará una notificación. Si ocurre algún error durante la conversión se lanzará un mensaje de error. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 28: Caso de Prueba de Aceptación_5

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: 6 | Nombre Historia de Usuario: Convertir los archivos de formato <i>Microsoft Office Excel(xlsx)</i> a <i>Open Document Spreadsheet(.ods)</i> |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad convertir los archivos de formato Microsoft Office Excel(xlsx) a Open Document Spreadsheet(.ods) | |
| Condiciones de Ejecución: El archivo a convertir deberá tener la extensión xlsx. | |
| Entrada / Pasos de ejecución: El usuario deberá elegir la opción de añadir archivo y luego iniciar la conversión. | |
| Esperado: Al finalizar la conversión la aplicación enviará una notificación. Si ocurre algún error durante la conversión se lanzará un mensaje de error. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 29: Caso de Prueba de Aceptación_6

| Caso de Prueba de Aceptación | |
|---|--|
| Código Caso de Prueba: 7 | Nombre Historia de Usuario: Ubicar el archivo convertido en la ruta especificada. |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Prueba a la funcionalidad ubicar el archivo convertido en la ubicación especificada. | |
| Condiciones de Ejecución: La dirección a ubicar el archivo debe existir, si no aparecerá una notificación preguntando al usuario si desea crearla. | |
| Entrada / Pasos de ejecución: El usuario deberá elegir en la opción cambiar carpeta de salida y elegir la nueva | |

| |
|--|
| ubicación que tendrá el archivo convertido. |
| Esperado: Al finalizar la conversión la aplicación enviará el archivo convertido hacia la nueva ubicación proporcionada por el usuario. |
| Evaluación de la Prueba: Satisfactoria |

Tabla 30: Caso de Prueba de Aceptación_7

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: 8 | Nombre Historia de Usuario: Convertir los archivos de formato Microsoft Office Access(mdb) a Open Document Database(.odb) |
| Nombre de la persona que realiza la prueba: Yoandy Pérez Villazón | |
| Descripción de la Prueba: Convertir los archivos de formato Microsoft Office Access(mdb) a Open Document Database(.odb) | |
| Entrada / Pasos de ejecución: El usuario deberá elegir la opción de añadir archivo y luego iniciar la conversión. | |
| Esperado: Al finalizar la conversión la aplicación enviará una notificación. Si ocurre algún error durante la conversión se lanzará un mensaje de error, la conversión se efectuará solo para las tablas y los datos de las bases de datos. | |
| Entrada / Pasos de ejecución: El usuario deberá elegir la opción de añadir archivo y luego iniciar la conversión. | |
| Evaluación de la Prueba: Satisfactoria | |

Tabla 31: Caso de Prueba de Aceptación_8

Observaciones:

Se realizaron un total de 12 pruebas, divididas en 3 iteraciones según los resultados obtenidos en cada caso de prueba de aceptación. En la primera iteración se realizaron 8 casos de prueba arrojando 3 no conformidades para las HU: 1, 2, 3, 4 y 5 siendo corregidas posteriormente. En la segunda iteración se elaboraron un total de 5 pruebas donde se obtuvo como resultado 1 no conformidad para la HU: 1 corregida de manera inmediata y para la tercera se efectuó 1 prueba arrojando resultados satisfactorios, comprobando que la herramienta se encuentra lista para ser usada en proyectos reales. A continuación se ilustra un gráfico que muestra el proceso detalladamente.

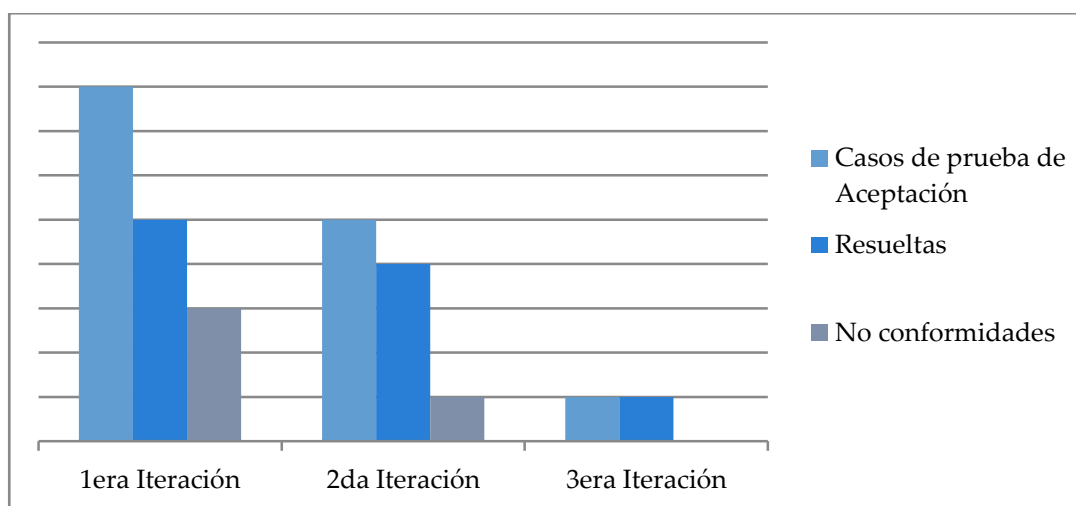


Ilustración 10: Proceso de pruebas

No conformidades detectadas en la 1era iteración:

- Las funcionalidades definidas en las HU_2, HU_3 y HU_4 no reconocen la conversión de los archivos que contengan espacios en el nombre.
- La funcionalidad definida para la HU_5 permite añadir cualquier tipo de extensión.
- La funcionalidad definida para la HU_1, para las bases de datos muy grandes no realiza la migración de los datos y las tablas.

No conformidades detectadas en la 2da iteración:

- La funcionalidad definida para la HU_1, cuando las bases de datos son muy grandes tarda en migrar los datos y las tablas.

3.7 Impacto y aporte de la solución

El impacto de la solución se materializa al efectuar la migración de los datos ofimáticos hacia la nueva plataforma libre instalada. El sistema permite efectuar conversiones de documentos de forma fácil e intuitiva. Su uso facilita el trabajo de los especialistas en migración, evitando que incurran en errores humanos durante la conversión.

Dentro de los aportes que ofrece la solución propuesta se encuentran la inclusión de un gestor de bases de datos que posibilita mantener la dirección de los archivos que aún no se hayan convertido dentro del sistema, listos para continuar con el proceso una vez ejecutada la aplicación. Esto permite que al abrirla, luego de algún error o fallo en el fluido eléctrico, el usuario no tenga que cargar los archivos pendientes por convertir, el

sistema los cargará automáticamente desde la base de datos. Además de la conversión de los archivos del formato OOXML al formato ODF, posibilita la migración de las tablas y los datos de las bases de datos de Access 2003 a LibreOffice Base. Para los archivos de extensión docx, pptx y xlsx, brinda la posibilidad de elegir la librería que desee para ejecutar la conversión pudiendo seleccionar *JODConverter* u *Odf-converter*, si una vez terminado el proceso, el usuario cree que no tiene la calidad requerida.

La propuesta de solución constituye un sistema con interfaz gráfica, que brinda varias opciones como: “Añadir archivos”, que permitirá al usuario añadir uno o más archivos al sistema; también se encuentra la funcionalidad “Eliminar archivos”, que suprimirá uno o varios archivos de la aplicación, dando la posibilidad además de “Limpiar la lista de archivos”. Si el usuario necesita realizar otras tareas que requieran los recursos de la computadora (CPU, RAM) la conversión podrá ser pausada sin necesidad de cerrar el programa o cancelar la ejecución. La herramienta de conversión de archivos ofimáticos, constituye un sistema multitarea, permitiendo la ejecución de varios procesos al mismo tiempo. Brinda una solución encaminada a mejorar el proceso de migración del país.

Conclusiones de Capítulo:

Con el apoyo del plan de release, el producto se fue desarrollando de manera escalonada, implementando las HU definidas en el diseño. Para la correcta construcción del sistema fueron trazadas 20 tareas de ingeniería y las clases y librerías fueron agrupadas en el diagrama de componentes del sistema. Finalmente el cliente aprobó el producto realizando pruebas de aceptación, ejecutadas en tres iteraciones, que permitieron obtener una aplicación libre de errores y ajustada a las necesidades del mismo.

Conclusiones

Al finalizar la investigación se concluye que:

1. Las herramientas *JODConverter* y *Odf-converter* permiten la correcta conversión de los documentos de extensiones docx, xlsx y pptx hacia odt, ods, odp y odb respectivamente. La aplicación *mdbtools* permite exportar a ficheros de texto planos los datos y el esquema de una base de datos *Microsoft Access*.
2. Los archivos odb contienen una base de datos HSQLDB.
3. Las librerías *jodbconverter.jar* y *hsqldb.jar* se pueden integrar a una aplicación Java para convertir archivos a odt, odp, ods y gestionar una base de datos HSQLDB respectivamente.
4. La migración de los formularios, consultas e informes de *Access* a Base no es posible debido a la incompatibilidad de las implementaciones de los lenguajes OOo Basic de LibreOffice Base y VBA de *Microsoft Access*.
5. El desarrollo de una interfaz gráfica, intuitiva y fácil de utilizar, hizo que el sistema se adecuara a las necesidades de los especialistas, facilitando el trabajo de los mismos.
6. Mediante la implementación de la propuesta de solución se obtuvo un sistema capaz de realizar conversiones de archivos ofimáticos destinadas a mejorar la calidad del proceso de migración a las empresas.
7. A partir de las pruebas a las funcionalidades implementadas, se comprobó que el sistema propuesto se encuentra listo para ser usado en procesos de migración reales.

Recomendaciones

A partir del estudio realizado en la presente investigación se hacen las siguientes recomendaciones:

- Validar el uso de la herramienta de conversión dentro de la migración real.
- Agregar al sistema la funcionalidad que permita la migración de las bases de datos de *Access* 2007/2010 a *odb*.
- Adicionarle al sistema funcionalidades que permitan la conversión hacia otras extensiones de archivo.
- Implementar un conversor de código *Visual Basic* para Aplicaciones hacia *OpenOffice Basic*.

Referencias Bibliográficas

1. *Software Libre*. [En línea] [Citado el: 10 de marzo de 2013.] <http://www.vialibre.org.ar>.
2. Villazón, Yoandy Pérez. *GUÍA CUBANA PARA LA MIGRACIÓN A SWL*. Universidad de las Ciencias Informáticas.
3. Tenzer, Mario Simón. Archivos, formatos y extensiones. *Archivos, formatos y extensiones*. [En línea] <http://www.ccee.edu.uy>.
4. Idem a la referencia. (3) [En línea]
5. Idem a la referencia. (3) [En línea]
6. Escuelas Libres. *Alternativas libres a formatos que no lo son*. [En línea] [Citado el: 4 de febrero de 2013.] <http://www.escuelaslibres.org.ar/2011/04/alternativas-libres-a-formatos-que-no-lo-son/>.
7. (6), Idem a la referencia. [En línea]
8. liveManual. [En línea] [Citado el: 4 de febrero de 2013.] 7. <http://www.google.com/cu/search?q=3.+http%3A%2F%2Fwww.livemanager.info%2Fchapter.php%3Fid%3D356&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a&channel=fflb>.
9. OpenXML no debe ser ISO 29500. *Introducción a OpenDocument y Office OpenXML*. [En línea] [Citado el: 1 de febrero de 2013.] 8. http://www.openxml.info/index.php?option=com_content&task=view&id=1&Itemid=7.
10. odf-converter-integrator. [En línea] [Citado el: 11 de marzo de 2013.] 9. http://www.downloadplex.com/Linux/Business/Office-Suites-Tools/odf-converter-integrator_234709.html.
11. JODConverter. [En línea] [Citado el: 11 de marzo de 2013.] 10. <http://www.artofsolving.com/opensource/jodconverter>.
12. Unoconv Art of Solving. [En línea] [Citado el: 7 de febrero de 2013.] 11. <http://www.artofsolving.com/opensource/Unoconv>.
13. Anon. Convertidor de documentos. [En línea] [Citado el: 11 de marzo de 2013.] <http://www.openoffice.org>.
14. —. Migración de bases de datos Microsoft Access a OpenOffice Base y MySQL. *Tutorial:Migración de bases de datos Microsoft Access a OpenOffice Base y MySQL*. [En línea] [Citado el: abril de 25 de 2013.] http://www.ehowenespanol.com/tutorial-macros-ms-access-manera_109051/.
15. —. Conceptos básicos de las macros en Access. [En línea] [Citado el: 1 de mayo de 2013.] 15. <http://office.microsoft.com/es-es/access-help/conceptos-basicos-de-las-macros-en-access-2007-HA001214202.aspx>.
16. Peñalver, Gladys Marsi. *SXP, metodología ágil para proyectos de Software Libre*. La Habana: Universidad de las Ciencias Informáticas : s.n., 2008.
17. Anon. Características de JAVA. [En línea] [Citado el: 4 de febrero de 2013.] 17. <http://tikal.cifn.unam.mx/~jsegura/LCGII/java3.htm>.

18. Mora, Farnscisco. *Lenguaje Unificado de Modelado*. s.l. : DCCIA, Universidad de Alicante, 2002, 2002.
19. Servicios a los usuarios. [En línea] [Citado el: 23 de febrero de 2013.] <http://recursostic.educacion.es/usuarios/web/es/preguntas-frecuentes/45-sqlite-> .
20. HyperQSL - 100% Database. *HyperSQL*. [En línea] [Citado el: 2 de junio de 2013.] <http://hqslodb.org>.
21. JODConverter. [En línea] [Citado el: 11 de marzo de 2013.] <http://www.artofsolving.com/opensource/jodconverter>.
22. Anon. Tutorial de Netbeans 7.2 parte 1. [En línea] [Citado el: 2 de mayo de 2013.] <http://hjcode.wordpress.com/tutorial-de-netbeans-7-2>.
23. (Paradigma Visual para UML (ME)) por Visual Paradigm International Ltd. [En línea] [Citado el: 1 de abril de 2013.]. <http://www.freedownloadmanager.org/es/>.
24. Idem a la referencia.(16) página 7
25. Idem a la referencia.(16) página 8
26. Idem a la referencia.(16) página 17
27. Preesman, Roger. *Ingeniería de software. Un enfoque práctico*.
28. Flower, Martin. *Patterns of Enterprise Application Architecture*. 2003.
29. Torriente, LLorente De La. *Una Guia_Arquitectura_N-Capas_DDD_NET_4_(Borrador_Marzo_2010)*. 2010.
30. Larman, Craig. *UML y Patrones*. . s.l. : Patience Hall, 1999. .
31. Idem a la referencia.(30) página 259
32. Larman, Craig. *Patrones del Gang of Four*. Madrid : s.n.
33. Idem a la referencia.(16) página 10
- 34 Heredia, Juan Carlos. *Dime cómo programas y te diré quién eres*.Latinoamérica msdn.[En línea] [Citado el 4 de mayo de 2013.]. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvsngen/html/hunganotat.asp>
35. Idem a la referencia. (27) página
36. Idem a la referencia.(16) página 26

Bibliografía

1. Anon. Tutorial de Netbeans 7.2 parte 1. [2 mayo 2013]. Disponible en la web: <<http://hjcode.wordpress.com/tutorial-de-netbeans-7-2>>.
2. DE LA TORRIENTE LLORENTE, Cesar, ZORRILLA CASTRO,Unai.Guia_Arquitectura_N-Capas_DDD_NET_4_(Borrador_Marzo_2010).
3. FERNÁNDEZ CÉSPEDES Raycel, PINO GARCÍA Susel, PEÑALVER ROMERO Gladys Marsi. Propuesta de un expediente, para los proyectos productivos del Polo de *Software* Libre, de la Facultad 10. Habana: Universidad de las Ciencias Informáticas, 2008.
4. Flower, Martin. 2003. Patterns of Enterprise Application Architecture. s.l. : Addison Wesley, 2003.
5. Larman, Craig,1999. UML y Patrones. s.l. :Patience Hall, 1999. 970-17-0261-1.
6. Larman, Craig,1999. Patronde del Gang of Four. s.l. :Patience Hall,
7. MORA, Farncisco. UML: Lenguaje Unificado de Modelado. DCCIA, Universidad de Alicante, 2002.
8. Anon (Paradigma Visual para UML (ME)) por Visual Paradigm International Ltd. [1 Abril 2013]. Disponible en la Web: <http://www.freedownloadmanager.org/es/>.
9. PEÑALVER, Gladys Marsi ROMERO. SXP, metodología ágil para proyectos de *Software* Libre. Habana: Universidad de las Ciencias Informáticas, 2008.
10. PEREIRA DE LA ROSA, Adisleydis. Propuesta de solución para la conversión de formatos de archivos a estándares abiertos. Habana: Universidad de las Ciencias Informáticas, 2012.
11. PÉREZ VILLAZÓN, Yoandy. GUÍA CUBANA PARA LA MIGRACIÓN A SWL versión 0.32.
12. PRESSMAN Roger. Ingeniería de *software*. Un enfoque práctico. Sexta parte.

Glosario de Términos

Código abierto: Es el término con el que se conoce al *software* distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado SWL.

GPL: Es una licencia creada por la *Free Software Foundation* en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de *software*. Su propósito es declarar que el *software* cubierto por esta licencia es SWL y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Historia de Usuario: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Multiplataforma: Es un término usado para referirse a los programas, sistemas, lenguajes de programación, u otra clase de *software*, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en *Windows* en un procesador x86, en GNU/Linux en un procesador x86, y en MAC OS X en uno x86 (solo para equipos Apple) o en un PowerPC.

ODF: *Open Document Format* (ODF), es un formato de archivo abierto y estándar para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, textos, gráficas y presentaciones.

OOXML: Office OpenXML (también llamado OOXML u OpenXML) es un formato de archivo abierto y estándar cuyas extensiones más comunes son .docx, .xlsx y .pptx. Se le utiliza para representar y almacenar hojas de cálculo, gráficas, presentaciones y documentos de texto.

SXP: Metodología compuesta por las metodologías SCRUM y XP, especialmente indicada para proyectos pequeños, rápido cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.

Release: Se refiere a una versión funcional de un producto *software*.

Software Libre: Es la denominación del *software* que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuirlo libremente. Según la *Free Software Foundation* (FSF), el SWL se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el *software* y distribuirlo modificado.

URL: El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos

de información, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etc. También es usado como dirección de los recursos de información disponibles en la Internet.

Anexos

Anexo 1

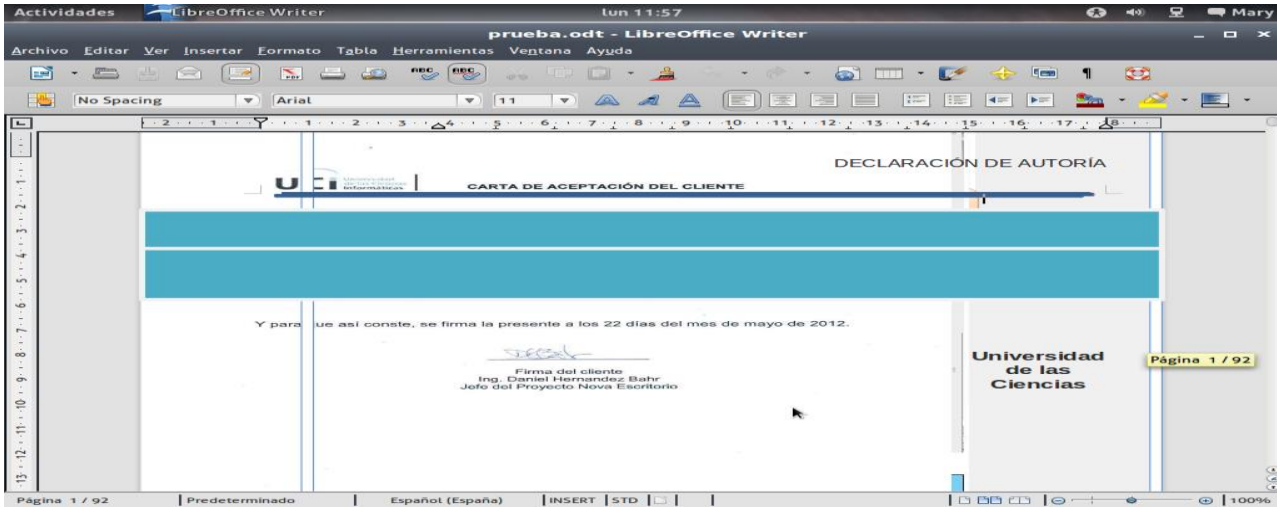


Ilustración 11: JODConverter

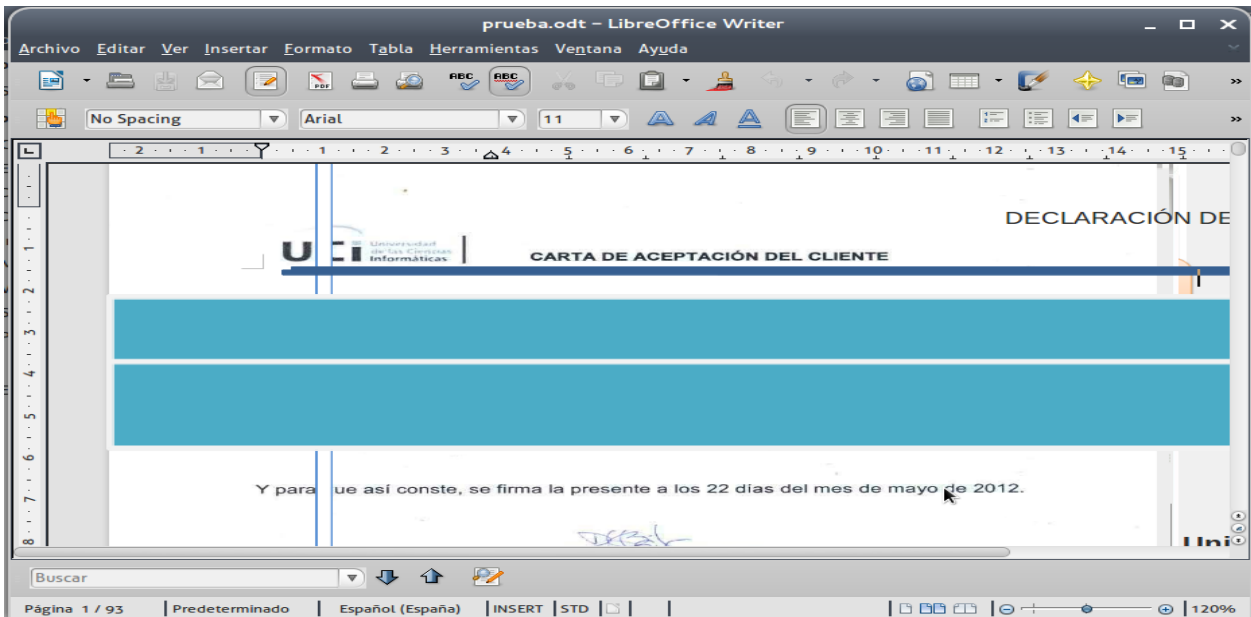


Ilustración 12: Unoconv

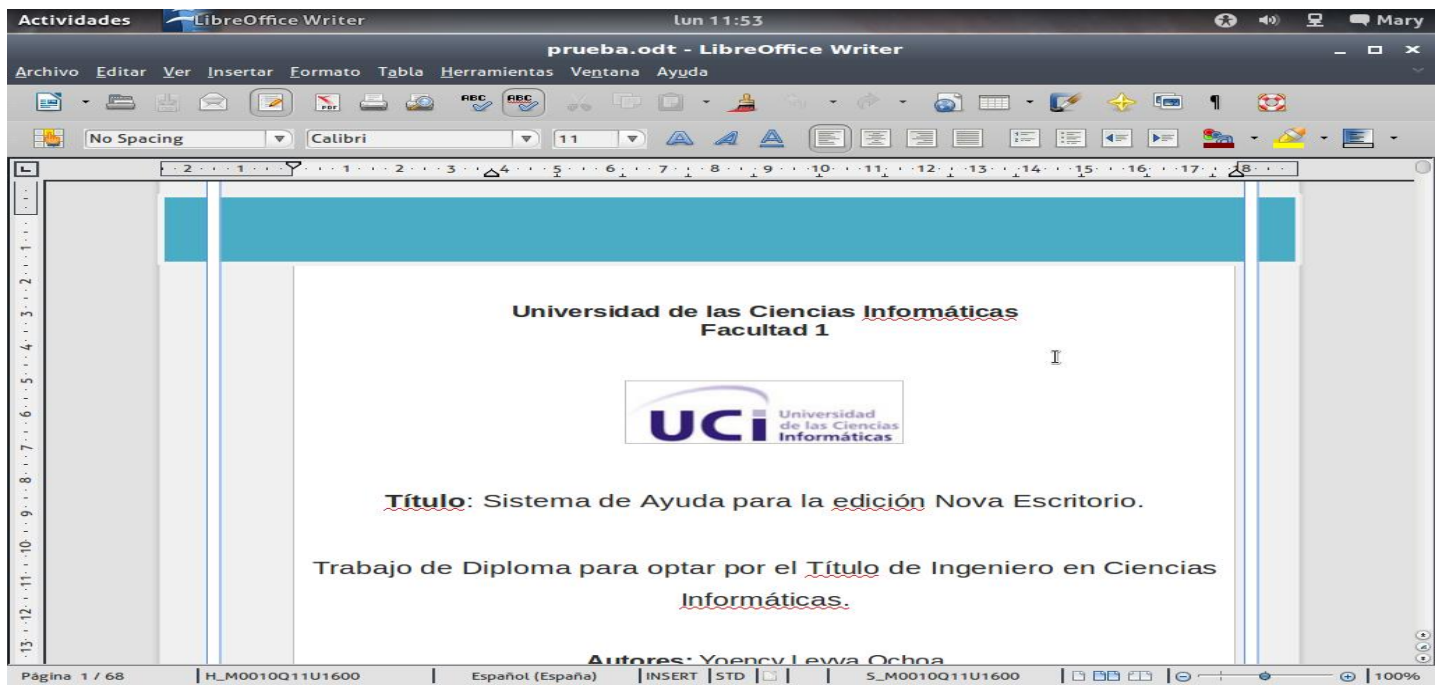


Ilustración 13: Odf-converter

Anexo 2

| VBA [2] | OOo Basic [2] |
|--|---|
| <pre>If x = 0 Then MsgBox "x es igual a cero", vbInformation Else MsgBox "x es distinto de cero", vbCritical End If</pre> | <pre>if x=0 then msgbox "x es igual a cero", 64 else msgbox "x es distinto de cero", 16 end if</pre> |
| <pre>For k = 0 To 3000 m = m + 1 Next MsgBox "k contiene " & k & " y m contiene " & _ Format(m, "#,##0.00"), vbQuestion</pre> | <pre>for k=0 to 3000 m=m+1 next msgbox "k contiene " & k & " y m contiene " & _ format(m,"#,##0.00"), 32</pre> |
| <pre>Sub EjemploVBA() ' Desde un documento Excel abre otro documento Excel ' existente y lo cierra. Dim oHoja As Workbook ' Abro el libro donde quiero buscar el valor oHoja = Workbooks.Open Filename:= _ "NombreyRutaDelDocumento" ' Cierro el libro que acabo de abrir. oHoja.Close End Sub</pre> | <pre>Sub EjemploOOoBasic() ' Desde un documento Calc abre otro documento Calc ' existente y lo cierra. Dim sRuta As String, oHoja As Object, mArg() ' Abro el libro donde quiero buscar el valor sRuta = ConvertToUrl("NombreyRutaDelDocumento") oHoja = StarDesktop.loadComponentFromURL(_ sRuta, "blank", 0, mArg()) ' Cierro el libro que acabo de abrir oHoja.close(True) End Sub</pre> |

Ilustración 14: Ejemplo de la implementación del VBA y el OOo Basic para macros de un Excel

Anexo 3

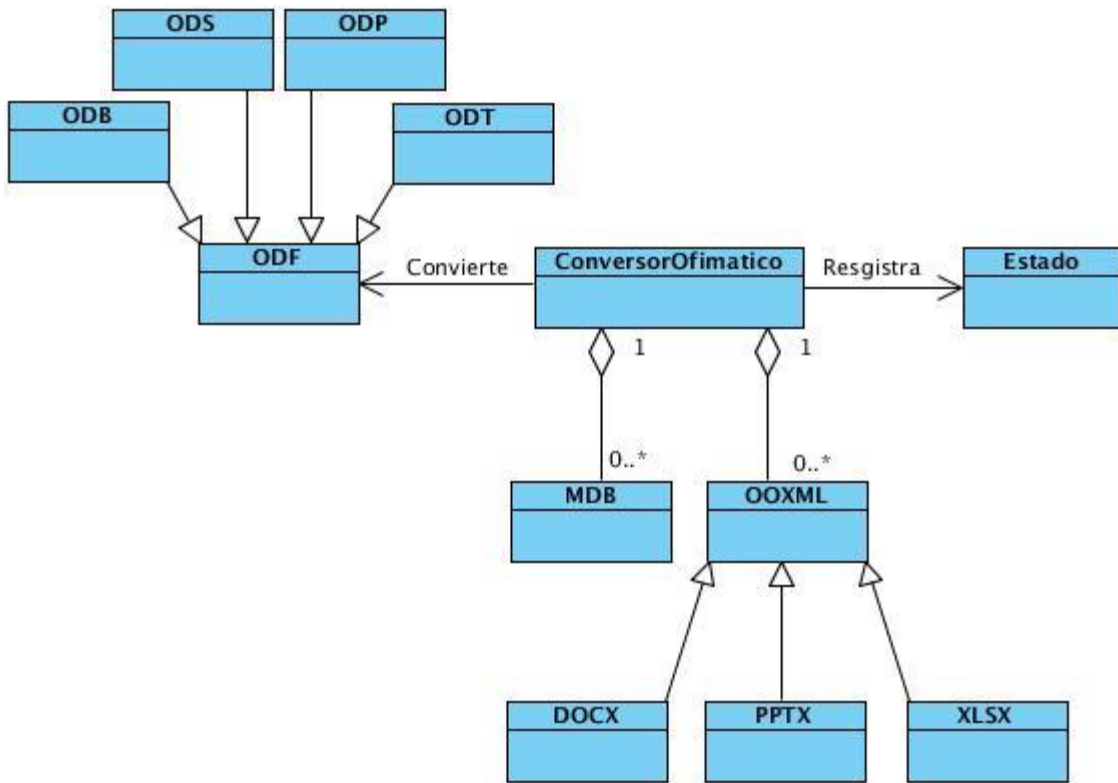


Ilustración 15: Modelo del dominio

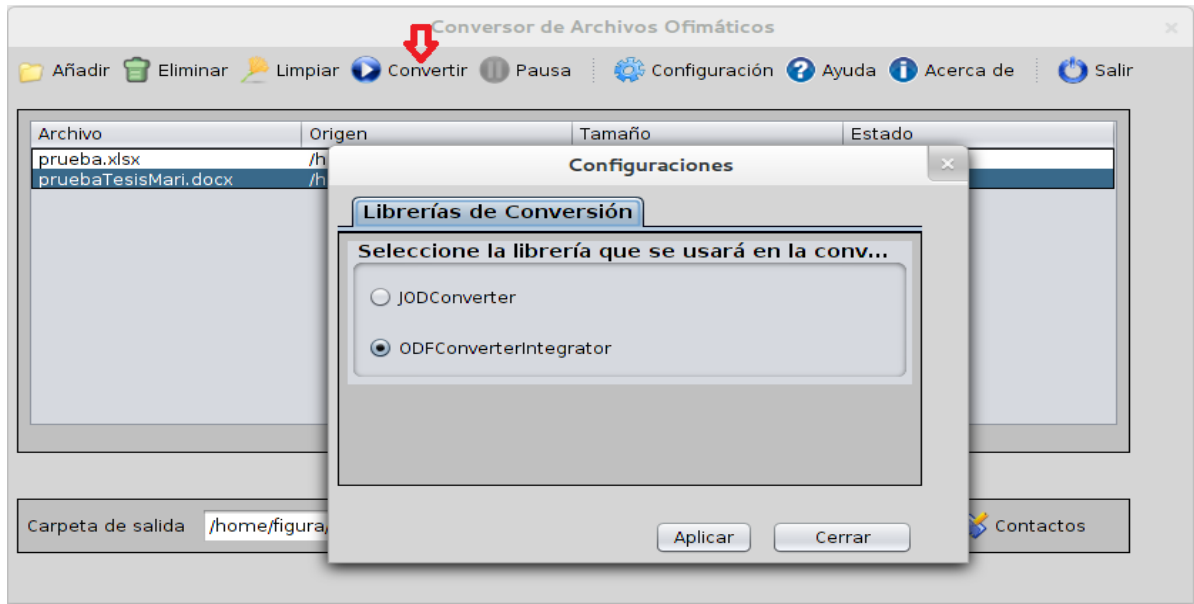


Ilustración 16: Interfaz de la funcionalidad Convertir

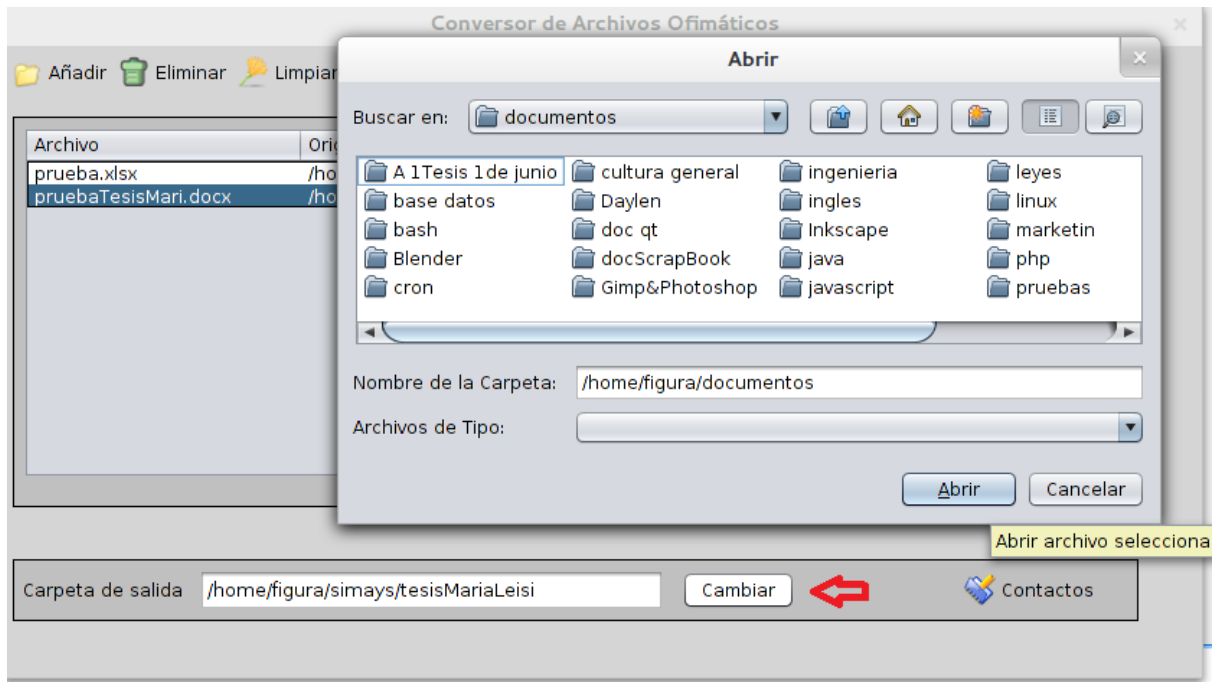


Ilustración 17: Prototipo de la funcionalidad ubicar archivo en la ruta especificada

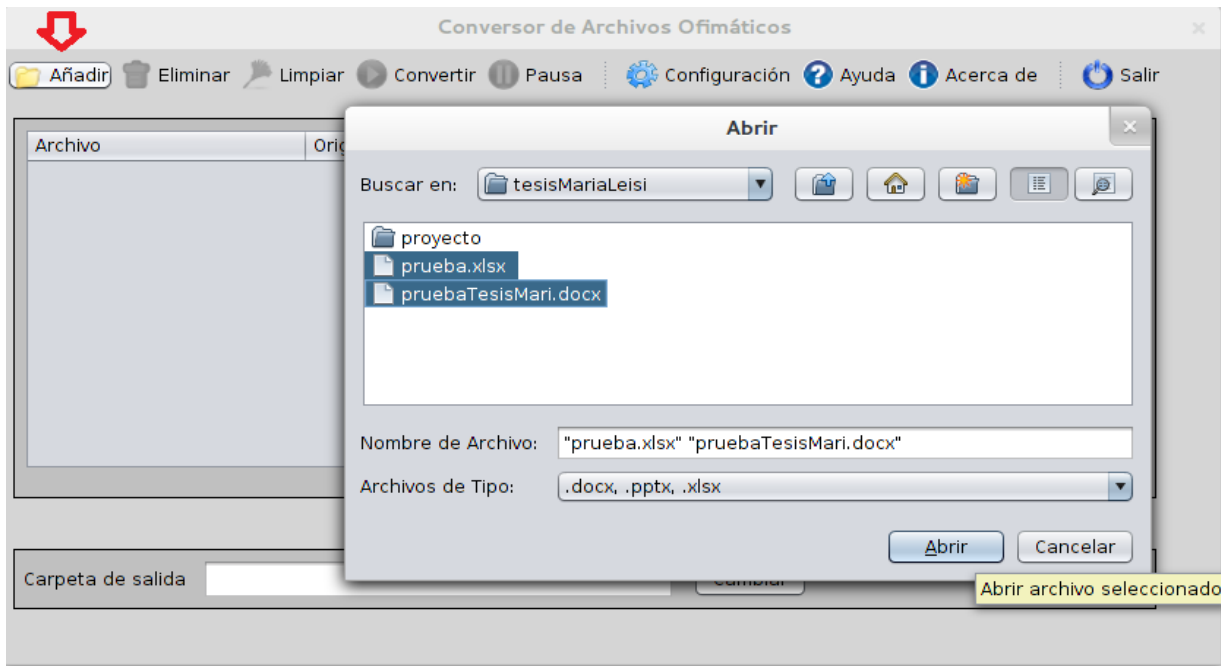


Ilustración 18: Prototipo de la interfaz Añadir archivo

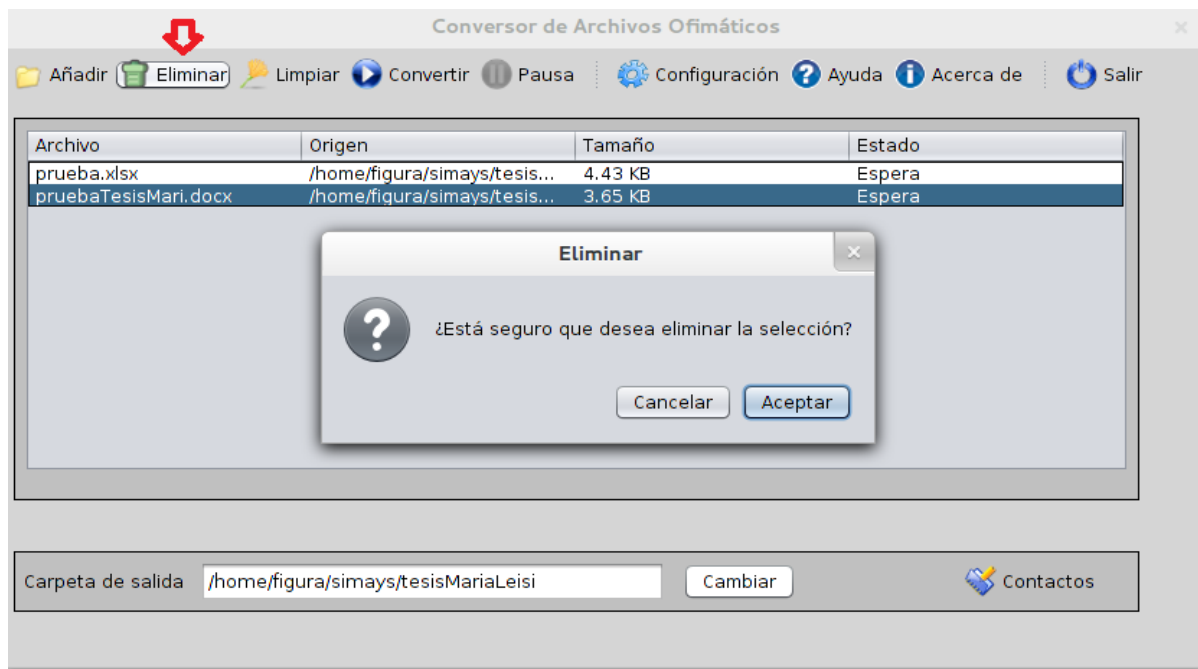


Ilustración 19: Prototipo de interfaz de la funcionalidad Eliminar archivo

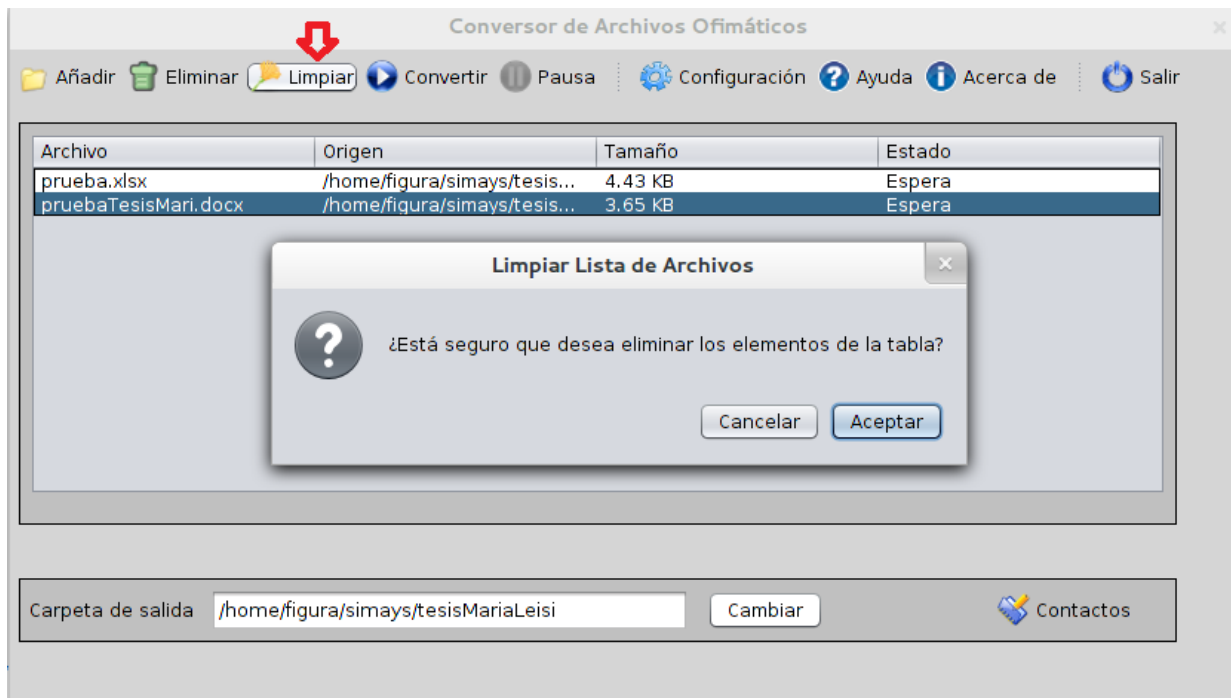


Ilustración 20: Prototipo de interfaz de la funcionalidad Limpiar lista

Anexo 5

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 3 | Número Historia de Usuario: HU_6 |
| Nombre Tarea: Diseño de la interfaz para añadir el archivo a la lista. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.2 |
| Fecha Inicio: 10/01/2013 | Fecha Fin: 15/01/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: Se diseña la interfaz para añadir los archivos a la lista para la conversión, a través del jbutton Añadir. | |

Tabla 32: Tarea de Ingeniería 3

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 4 | Número Historia de Usuario: HU_7 |
| Nombre Tarea: Diseño de la interfaz eliminar el archivo de la lista. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.4 |
| Fecha Inicio: 20/01/2013 | Fecha Fin: 25/01/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: Se diseña la interfaz para la eliminación de los archivos a través del jbutton Eliminar, se podrá eliminar uno o varios archivos a la vez. | |

Tabla 33: Tarea de Ingeniería 4

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 5 | Número Historia de Usuario: HU_8 |
| Nombre Tarea: Diseño de la interfaz para listar archivos a convertir. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 30/01/2013 | Fecha Fin: 05/02/2013 |
| Programador Responsable: María Leisy González Carrera | |

Descripción: Se diseña la interfaz para mostrar las propiedades de los archivos tales como tamaño, tipo y nombre.

Tabla 34: Tarea de Ingeniería 5

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 10 | Número Historia de Usuario: HU_6 |
| Nombre Tarea: Desarrollo de la funcionalidad añadir archivo a la lista. | |
| Tipo de Tarea: Desarrollo | Puntos Estimados: 0.4 |
| Fecha Inicio: 01/03/2013 | Fecha Fin: 05/03/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: La funcionalidad permitirá añadir uno o más archivos a la vez. | |

Tabla 35: Tarea de Ingeniería 10

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 11 | Número Historia de Usuario: HU_7 |
| Nombre Tarea: Desarrollo de la funcionalidad eliminar el archivo de la lista. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 0.4 |
| Fecha Inicio: 06/03/2013 | Fecha Fin: 11/03/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: Se seleccionan los archivos de la lista para su eliminación, se podrá eliminar uno o varios archivos a la vez. | |

Tabla 36: Tarea de Ingeniería 11

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 12 | Número Historia de Usuario: HU_8 |
| Nombre Tarea: Desarrollo de la funcionalidad para listar archivos a convertir. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 12/03/2013 | Fecha Fin: 18/03/2013 |

| |
|---|
| Programador Responsable: María Leisy González Carrera |
| Descripción: Se mostrarán al usuario las propiedades del archivo tales como, tamaño, tipo, nombre. |

Tabla 37: Tarea de Ingeniería 12

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 13 | Número Historia de Usuario: HU_2 |
| Nombre Tarea: Desarrollo de la conversión de archivos de formato <i>Microsoft Office Word(docx)</i> a <i>Open Document Text(.odt)</i> | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 20/03/2013 | Fecha Fin: 25/03/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: La funcionalidad convertirá las extensiones de documentos de docx a odt, a través de las librerías <i>JODConverter</i> o <i>Odf-converter</i> . | |

Tabla 38: Tarea de Ingeniería 13

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 14 | Número Historia de Usuario: HU_3 |
| Nombre Tarea: Desarrollo de la conversión de archivos de formato <i>Microsoft Office Power Point(pptx)</i> a <i>Open Document Presentation(.odp)</i> | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 25/03/2013 | Fecha Fin: 30/03/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: La funcionalidad convertirá las extensiones de documentos de pptx a odp, a través de las librerías <i>JODConverter</i> o <i>Odf-converter</i> . | |

Tabla 39: Tarea de Ingeniería 14

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 15 | Número Historia de Usuario: HU_4 |
| Nombre Tarea: Desarrollo de la conversión de archivos de formato <i>Microsoft Office Excel(xlsx)</i> a <i>Open Document Spreadsheet(.ods)</i> | |

| | |
|---|------------------------------|
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 01/04/2013 | Fecha Fin: 05/04/2013 |
| Programador Responsable: María Leisy González Carrera | |
| Descripción: La funcionalidad convertirá las extensiones de documentos de xlsx a ods, a través de las librerías <i>JODConverter</i> o <i>Odf-converter</i> . | |

Tabla 40: Tarea de Ingeniería 15

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 16 | Número Historia de Usuario: HU_5 |
| Nombre Tarea: Desarrollo de la funcionalidad ubicar el archivo convertido en la ruta especificada | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 |
| Fecha Inicio: 05/04/2013 | Fecha Fin: 10/04/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: La funcionalidad permitirá la elección de la carpeta de salida que el usuario escoja. | |

Tabla 41: Tarea de Ingeniería 16

| Tarea de Ingeniería | |
|---|---|
| Número Tarea: 19 | Número Historia de Usuario: HU_1 |
| Nombre Tarea: Desarrollo de la conversión de archivos de formato <i>Microsoft Office Access</i> (mdb) a <i>Open Document Database</i> (.odb) | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 2 |
| Fecha Inicio: 20/05/2013 | Fecha Fin: 01/06/2013 |
| Programador Responsable: Madeleinis Morales Rodríguez | |
| Descripción: La funcionalidad importará los esquemas y los datos de las bases de datos de extensiones de .mdb a .odb, haciendo uso de la librería <i>mdbaOdb.jar</i> . | |

Tabla 42: Tarea de Ingeniería_20