Universidad de las Ciencias Informáticas

Facultad 10

Título: Módulo para el procesamiento y digitalización de documentos.

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

Autor (es): Katiuska Cost Megret

Jorge Luis Rodríguez Monteagudo.

Tutor (es): Ing. Evelio Maikel Medina Manrique.

Curso 2008-2009

La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida. La libertad no es elegir quien será tu amo, es no tener amo.

Richard Stallman

DECLARACIÓN DE AUTORÍA.

Declaramos que somos los únicos autores del trabajo titulado:

Módulo para el procesamiento y digitalización de documentos y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Katiuska Cost Megret

Autor.

_____

Jorge Luis Rodríguez Monteagudo

Autor.

_____

Ing. Evelio Maikel Medina Manrique

Tutor.

_____

AGRADECIMIENTOS

Katiuska:

Le agradezco a mi madre por brindarme ayuda cuando más lo necesitaba, por su apoyo incondicional, por quererme tanto como lo ha hecho, por cuidarme, por tanta entrega, por estar siempre conmigo, por todo y por tantas cosas…

A mi padre por apoyarme siempre en todo, brindarme ayuda cuando más lo he necesitado, por ser el mejor padre del mundo, por quererme y tenerme siempre presente…

A mi hermana por ayudarme tanto en todos estos años, por su apoyo incondicional, por ser tan buena hermana conmigo, por ayudarme en todo lo que he necesitado, por estar siempre cuidándome, por tantos años de preocupación y entrega…

A mis compañeros de grupo por ser tan buenos conmigo y por los felices y tristes momentos que juntos pasamos durante toda la carrera…

A mis amistades por quererme tanto, por ayudarme cuando más lo necesitaba, por apoyarme siempre en todo y por confiar en mí…

A mis profesoras Susel y Loania por ayudarme siempre que lo he necesitado y por brindarme su amistad…

Jorge:

Primero que todo agradecer a mi familia por todo el apoyo brindado en todo estos 5 años… A mi Mimí (mi mamá) que ha deseado tanto esta graduación…

Agradecer a todas las personas del proyecto de Gestión Documental y Archivística a Salomon, Marcel, Rafa, Michel, Frank y profesores del proyecto…

A los amigos que se hacen en estos cinco años de estudio…

A la Universidad de la Ciencias Informáticas (UCI) específicamente a la Facultad X porque me dio la oportunidad de descubrir y conocer el mundo del Software Libre, abriéndome los ojos en cuanto al gran espectro existente en materia de software, que va mucho mas allá que cualquier nombre de compañía conocido como Microsoft…

DEDICATORIA.

Katiuska

Dedico este trabajo, mi carrera y lo que soy, solamente a tres personas: a mi mamá por ayudarme siempre en todo, a mi papá por sus consejos y confianza en mí y a mi hermana por ayudarme cuando más lo necesitaba, por su apoyo incondicional, por estar siempre a mi lado y levantarme siempre los ánimos. Gracias a los tres por su amor, cariño y dedicación.

Jorge

Dedico este trabajo a mi mamá, a mi familia y a todas las personas que de una u otra manera han contribuido a la realización del presente trabajo.

RESUMEN

Este trabajo tuvo como objetivo el desarrollo de un módulo para el Gestor de Contenido Empresarial (ECM) Alfresco, que fuese capaz de salvar en su repositorio los documentos digitalizados por el software DocDigital a través de su interacción con un escaner. Primeramente se abordó todo lo referente a la digitalización y las ventajas que esta trae al ser aplicada. Posteriormente se realizó un estudio del arte sobre los softwares existentes para digitalizar documentos, contribuyendo con información para la creación del módulo, partiendo del uso de los protocolos estándares TWAIN y SANE para el acceso a un escaner en diferentes sistemas operativos. El desarrollo de este módulo se llevó a cabo por la carencia de una aplicación libre y multiplataforma que permita la obtención de las imágenes de un escaner, y procesarlas para la creación de documentos que serán gestionados por Alfresco, lográndose el objetivo propuesto con la implementación del módulo DocDigital.

Palabras Claves:

Digitalización, Implementación, Alfresco, Documentos, Escaner.

Tabla de contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Con el vertiginoso desarrollo de las tecnologías de la información y el conocimiento, tanto en las empresas como en las instituciones del Estado, las Tecnologías de la Información y las Comunicaciones (TIC) deben ocupar un lugar destacado con el fin de brindar servicios eficientes y eficaces. La información en las instituciones está en los documentos que se producen o se reciben todos los días en el ejercicio de las funciones. Los documentos permiten:

- Conducir los asuntos de una manera ordenada, eficaz y responsable.
- Dar coherencia, continuidad y productividad a las labores de la entidad.
- Mejorar los resultados de las actividades que se realizan en la entidad.
- Proteger los intereses y derechos de los colaboradores y de los usuarios.
- Proporcionar evidencias de las actividades que se realizan.

Los directivos de las instituciones están de acuerdo con la importancia de la gestión de la información, pero en muchos casos no saben cómo abordar los problemas de la información que se dan con soluciones prácticas. En la gran mayoría de los casos se quiere solucionar los problemas de la información desde una perspectiva informática, esto es debido a la publicidad que alguien llega con frases como las siguientes: "*Cualquier día de estos su empresa podría hundirse en un mar de papeles", "Digitalice sus documentos".* Para conseguir una buena gestión de la información ya sea manual o automatizada, se debe contar primero con una buena gestión documental. La gestión documental persigue que el trabajo con los documentos sea más fácil, de manera que una persona sabe qué documentos guardar, cuándo, cómo y dónde, con el fin de localizarlos cuando los necesita. Un proceso que forma parte y ayuda a hacer mucho más fácil el cumplimiento de los propósitos mencionados de la gestión documental es la digitalización de los documentos, su posterior registro en algún medio que permita una fácil búsqueda y consulta del mismo así como la mejoría que trae implícita la digitalización en cuanto a conservación, consulta, búsqueda y almacenamiento, dos ligeros ejemplos; almacenar cientos de miles de archivos de una entidad requiere de cierta cantidad de metros cuadrados disponibles mientras que en formato digital sería suficiente unas decenas de discos compactos (CD), un disco duro o cualquier otro soporte que lo permita; otro ejemplo, un documento original solo puede ser visto por una persona a la vez en tanto uno digitalizado puede ser consultado de forma simultánea por cualquier cantidad de

personas. Pero incluso si se digitalizan los documentos sin un método, sin una descripción precisa para recuperar la información lo que se obtendrá será un caos electrónico costoso y peligroso; siendo incluso más difícil acceder a una información almacenada en un disco óptico que a documentos almacenados en cajas debidamente rotuladas, haciéndose necesario almacenar el archivo en un sistema que sea capaz de conservarlo, organizarlo, etc.

Situación problémica

En el grupo de proyecto Gestión Documental y Archivística (GDA) perteneciente al polo Gestión de la Información y el Conocimiento, uno de los procesos fundamentales es la digitalización de documentos, proceso que es tenido en cuenta por el proyecto a la hora de ofrecer sus soluciones, pero no se cuenta con un software para digitalizar archivos que se integre con la solución Alfresco que propone el grupo. En el mercado existen gran variedad de herramientas propietarias pero no herramientas de Software Libre (**SWL**), por lo que es difícil encontrar una herramienta informática que permita desde varios sistemas operativos el manejo de un escaner determinado, así como la gestión del proceso integro referente a la digitalización de documentos.

Problema científico

¿Cómo resolver la necesidad de una aplicación para la digitalización de documentos que se integre al sistema Alfresco que propone el grupo de proyecto Gestión Documental y Archivística de la UCI?

Objeto de estudio

El objeto de estudio de este trabajo de diploma son las aplicaciones para la digitalización de documentos y su interoperabilidad con algún sistema gestor de contenidos.

Campo de acción

El campo de acción es una aplicación para digitalizar archivos que tenga interoperabilidad con el repositorio del **ECM** Alfresco.

Objetivo General

Desarrollar una aplicación para digitalizar y procesar documentos, que permita almacenarlos en el repositorio del ECM Alfresco.

Objetivos específicos

- Analizar las aplicaciones para digitalizar documentos.
- Aplicar los estándares SANE (Scanner Access Now Easy) y TWAIN (Technology Without An Interesting Name) para el trabajo con el escaner en varios Sistemas Operativos utilizando la librería mmscomputing.
- Digitalizar las imágenes a los formatos **PDF** (Portable Document Format) o **RTF** (Rich Text Format) a través de la librería iText.
- Realizar el análisis y diseño para el desarrollo de la aplicación.
- Implementar la aplicación que sea multiplataforma.

Tareas

Con vistas al alcance de los objetivos propuestos se hace necesario realizar las siguientes tareas:

- Estudio en Internet de aplicaciones para digitalizar documentos y de los estándares SANE y TWAIN.
- Estudio y uso de la librería libre en Java mmscomputing.
- Estudio y trabajo con la capa de servicios web que brinda el ECM Alfresco.

Los métodos empleados para desarrollar la investigación fueron:

Métodos teóricos.

Análisis-Síntesis.

Este método ha servido para analizar y comprender la teoría y documentación relacionada con el tema de investigación logrando la asimilación de los elementos más importantes relacionados con el objeto de estudio.

Métodos Empíricos.

Observación

Se ha utilizado este método sobre otros trabajos que están estrechamente relacionados con la digitalización de documentos o tienen elementos en común con la investigación. También se ha observado la situación real que se está investigando, permitiendo acercarse al objetivo final (implementación de la

aplicación para la digitalización de documentos).

Estructuración del contenido.

El trabajo de diploma consta de cuatro capítulos que incluirán todo lo relacionado con el trabajo investigativo a realizar, así como el análisis y diseño de la herramienta que se propone.

El capítulo 1: Fundamentación teórica. Conceptualizar la funcionalidad de las aplicaciones de digitalización de documentos así como la investigación acerca de este tipo de herramientas, una breve descripción, definiciones y las ventajas que tiene la digitalización de documentos. Analizar ejemplos de estas aplicaciones en el mundo. Describir la metodología, las tecnologías, las herramientas, los estándares, las librerías y el lenguaje a utilizar para la implementación de la aplicación.

El capítulo 2: Propuesta del sistema. Explicar detalladamente el proceso que se encuentra involucrado en el dominio, representación mediante una herramienta **CASE**, descripción por pasos lógicos para comprender y organizar el proceso. Realizar la representación y descripción de los Casos de Uso del Sistema (**CUS)** y de la propuesta que se quiere implementar. Llevar a cabo la captura de los Requerimientos no Funcionales (**RNF**) y Funcionales (**RF**) los que darán apoyo para comenzar a desarrollar el software.

El capítulo 3: Diseño del sistema. Presentar el diseño del sistema que se propone usando la metodología **RUP**. Describir y mostrar el diagrama de clases del diseño con la descripción de cada una de sus clases para una mejor organización, sus diagramas de interacción y el diagrama de despliegue, el cual describe cómo y dónde el sistema será puesto en funcionamiento y donde se verá bien detallado la ubicación y funcionamiento del escaner.

El capítulo 4: Implementación y Prueba. Abordar todo lo relacionado con el flujo de trabajo de implementación, desarrollar el diagrama de implementación o componentes para dar una visión de cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel así como las conexiones entre ellos, además de la descripción detallada de los paquetes de implementación. Realizarlas pruebas de caja negra, las cuales aseguran el correcto funcionamiento de la aplicación según sus requerimientos funcionales.

CAPÍTULO 1

Fundamentación teórica

1.1 Introducción

El avanzado desarrollo de la ciencia, la tecnología y otras ramas en el mundo, actualmente da lugar a que las empresas e instituciones, independientemente de su tamaño produzcan, manejen y reciban diariamente una gran cantidad de documentos de muy diversa tipología. Ante tal situación, para que se maneje de forma organizada y digitalizada estos documentos, surgió la necesidad de utilizar herramientas para crear aplicaciones para digitalizar documentos.

En el presente capítulo se realiza una investigación acerca de aplicaciones para digitalizar documentos, una breve descripción, definiciones, estado del arte de los aplicaciones para la digitalización, ventajas de la digitalización de documentos, así como un estudio de las tecnologías, metodologías, lenguajes, herramientas de desarrollo de software utilizadas en la actualidad y la selección de las mismas para llevar a cabo el desarrollo de nuestro trabajo de tesis, teniendo en cuenta que las que se utilicen deben garantizar el cumplimiento de nuestro objetivo.

1.2 Aplicaciones para la digitalización de documentos. Breve descripción y Definiciones

Cuando se habla de aplicaciones para digitalizar documentos se hace referencia a una aplicación donde los documentos puedan ser transformados de forma escrita o impresa, es decir formato papel en forma digital o electrónica, y es aquí donde la digitalización ejerce su papel, la cual es la operación mediante la cual se convierte una imagen en una serie de códigos binarios que representan cada uno de los puntos de su estructura y que, de esta forma, puede ser almacenada en el ordenador. Así pues, se trata de la conversión de una imagen en un conjunto de valores numéricos digitales. Los dispositivos característicos para esta operación son: los escaneres y las videocámaras. En todos estos casos, la información puntual recibida llega a la memoria del ordenador y se va utilizando después para el tratamiento y salida subsiguiente. Estos documentos una vez digitalizados pueden ser gestionados por algún Sistema de Manejo de Contenido Empresarial (ECM). Para este proceso se requiere de varios mecanismos que optimicen la velocidad y validez para propiciar intervenciones mínimas del usuario, estos documentos posteriormente deben ser almacenados en un repositorio para que usuarios autorizados accedan

fácilmente a los mismos de forma instantánea y simultánea. Si se dispone de la información en un formato único, esto facilita su gestión y organización de la misma de forma centralizada. La documentación en formato digital es mucho más sencilla de manejar y brinda la posibilidad de disponer de ella cuando se necesite. La disponibilidad inmediata de la información no es la única ventaja de la digitalización de documentos. La ausencia de archivos en formato papel permite un importante ahorro de espacio además de ser una medida más ecológica.

1.3 Estado del arte de las aplicaciones para la digitalización de documentos

Realizando investigaciones de herramientas o aplicaciones que digitalizan documentos se han encontrado varios software que resuelven el gran problema de digitalizar los documentos, los cuales están enfocados mayormente a las necesidades de empresas e instituciones que trabajan con gran cantidad de papeles.

- Ejemplos en el mundo.

Abbyy FineReader v9.0.724 Corporate Edition Multilenguaje

Abbyy FineReader digitaliza los documentos de forma rápida. FineReader es una aplicación **OCR** (Reconocimiento Óptico de Caracteres) de varios idiomas, capaz de construir texto con formato a través de imágenes procedentes de un escaner o de una cámara digital. De esta forma, evita tener que volver a introducir por teclado el contenido de un texto impreso, del que no se ha guardado una copia digital. El funcionamiento es sencillo: se escanea el documento, se 'lee' y se guarda en el ordenador. (1)

El programa hace gala de un alto nivel de precisión en el reconocimiento de caracteres y retención de formato, lo que permite disponer en unos pocos minutos de una copia electrónica de papeles, revistas, faxes, periódicos, libros, etc. y a partir de aquí poder hacer lo que se quiera con dichos documentos: editarlos, enviarlos por e-mail, guardarlos en diversos formatos (doc, pdf, xml, html, etc.) FineReader es una herramienta muy potente que reconocerá el texto, las imágenes o las tablas que haya en un documento. Resulta además muy fácil de usar; dispone de una interfaz intuitiva y de diseño sencillo, con un editor de textos integrado, soporte para guardar imágenes, utilidad de búsqueda y mucho más. Muchas aplicaciones han intentado eliminar definitivamente la tortuosa tarea de copiar texto, pero pocas alcanzan la eficiencia de Abbyy FineReader, considerado uno de los mejores de su tipo por sus funcionalidades, facilidad de uso y consumo de recursos. (1) Para muchos, el mejor reconocedor de texto del mercado: sencillo, potente y seguro, solo se ejecuta en sistemas Windows; su precio es de 169.00 € y

no tiene integración con ningún sistema gestor de contenidos.

.

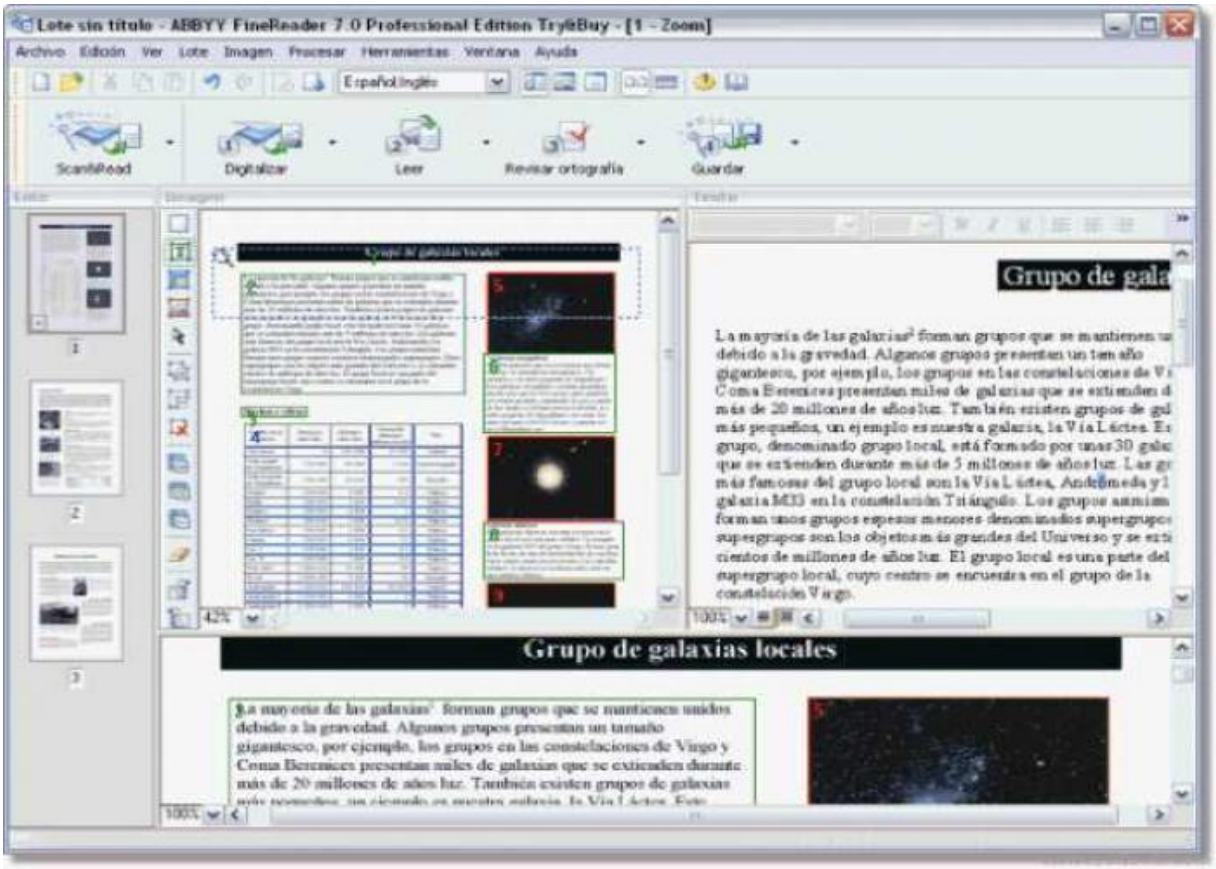


Figura 1: Aplicación Abbyy FineReader.

QuickScan Pro de Captiva de EMC (Emerge Merger Corporation)

QuickScan Pro de Captiva de EMC es una solución de captura de documentos en una única estación de trabajo, lista para usar. QuickScan Pro permite escanear documentos en batches (lotes), visualizar automáticamente documentos separados y manipular archivos antes de exportarlos a formatos de archivos más populares. Esta solución de escritorio se puede integrar fácilmente a sistemas de administración de documentos y funcionar como módulos de escaneo para entregar imágenes escaneadas y mejoradas al sistema de administración de contenido. (2)

Capítulo 1: Fundamentación teórica

Este software de escaneo de documentos es ideal para pequeñas y medianas empresas, y permite eliminar el papeleo manual y automatizar el manejo de miles de documentos. Podrá procesar más documentos de manera más eficaz y reducir o eliminar los costos de archiving y almacenamiento de documentos impresos. (2) Esta aplicación se caracteriza por ser propietaria, solo puede ejecutarse en Sistemas Windows, su precio oscila entre $475/escaner y $3075/escaner en dependencia del nivel de funcionalidades y se puede integrar con Documentum AplicationXtender que almacena, organiza y administra todos los tipos de información del negocio y proporciona acceso instantáneo, con control de seguridad a través de Microsoft Windows o de navegadores Web estándar.

QuickScan Pro de Captiva de EMC ofrece: (2)

Características	Beneficios
Integración con escaneres	Aproveche la compatibilidad con más de 300 tipos diferentes de escaneres e implemente QuickScan Pro de manera rápida y fácil.
Administrador de batches	Administre fácilmente el estado de varios batches y determine el momento en que se escaneó, indexó y exportó cada batch.
Filtros de mejora de imagen	Mejore la claridad de imagen, reconozca códigos de barras para separación e identificación de documentos, y elimine páginas en blanco.
Asignación automática de nombres de archivos	Asigne nombres a archivos y organícelos en función de sus reglas de negocio.
Configuración de documentos de múltiples páginas	Anexe documentos de múltiples páginas y elimine o inserte imágenes en documentos de múltiples páginas.
Comentarios de imágenes	Mejore las imágenes con resaltadores de múltiples colores, dibujos de múltiples líneas y a mano alzada, redacciones, notas autoadhesivas y otras herramientas.

Modo administrado	Permite a múltiples usuarios compartir perfiles en una red, lo que simplifica la implementación de grupos de trabajo.

Tabla 1: Características y beneficios de QuickScan Pro de Captiva.

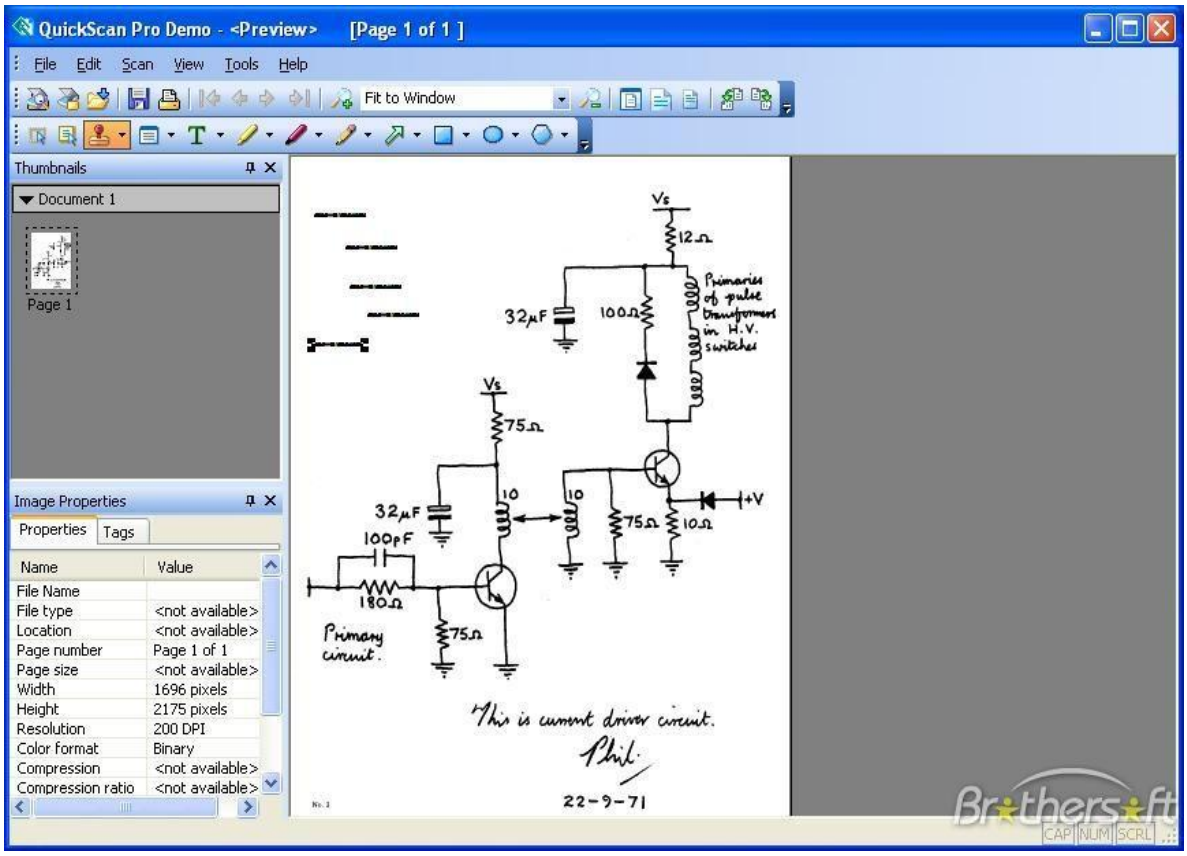


Figura 2: Software QuickScan Pro.

Documalis Free Scanner 1.0.0.1

Documalis Free Scanner es un software gratuito para digitalización de documentos y las plataformas corren sobre Windows. Posee una interfaz simple y amigable tipo Microsoft Office 2007 y es capaz de manejar cualquier escaner compatible con TWAIN o **WIA**. Los documentos digitalizados pueden ser pre-

visualizados en la pantalla, y luego convertidos directamente a formatos pdf, jpeg, bmp, png o tiff. Usando el Escaner Gratuito Documalis se puede manejar cualquier escaner con o sin alimentador y configurar la resolución y la cantidad de colores para su digitalización. Esta solución también administra digitalización frente/dorso, provee una función para detección y eliminado de páginas en blanco en forma manual o automática, para rotar o reemplazar páginas digitalizadas. (3) Se ejecuta solamente en Sistemas Windows y no tiene integración directa con ningún gestor de contenidos.

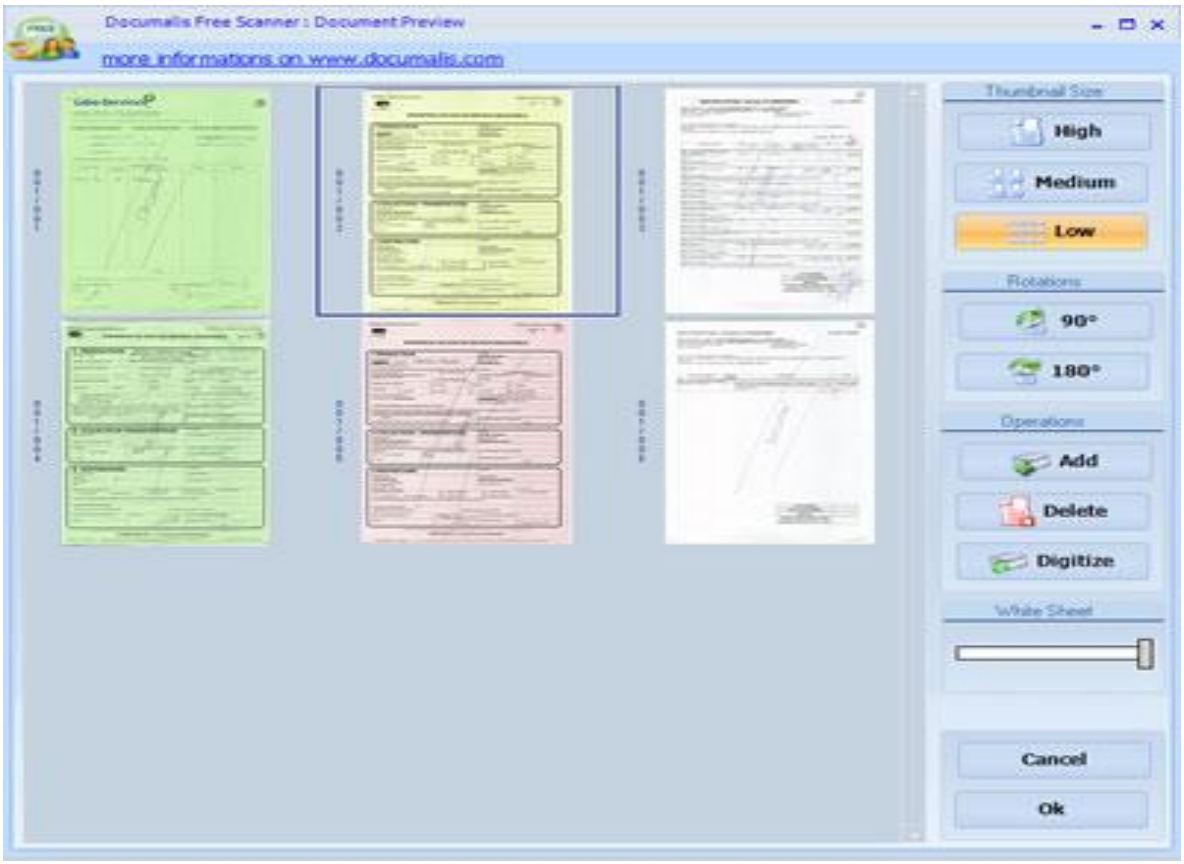


Figura 3: Software Documalis Free Scanner.

Kooka

Es un magnifico programa que convierte una imagen con tan solo pulsar 2 botones. Su instalación es bastante simple, funciona en **KDE** y Gnome.

Kooka es un software con licencia **GPL** (General Public Licence, Licencia Pública General) que ayuda a manejar los más importantes parámetros de escaneo, encuentra el formato correcto de imagen para salvar y manejar las imágenes escaneadas. También ofrece soporte para diferentes módulos OCR (Reconocimiento Óptico de Caracteres). Libkscan, una parte autónoma de Kooka, provee servicios de escaneo para un fácil y consistente uso. (4)

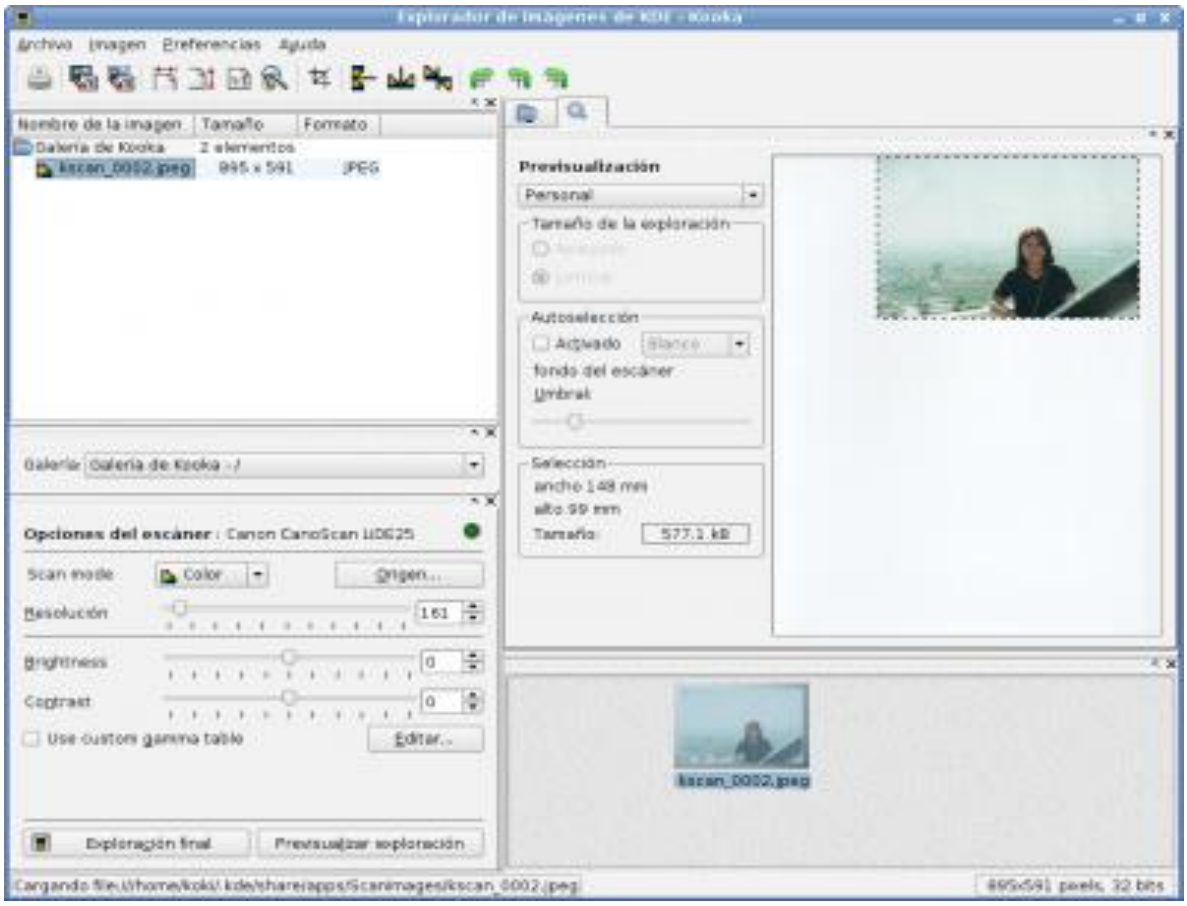


Figura 4: Software Kooka.

ScanToPDF 3.2

Con ScanToPDF se podrá pasar todos los documentos de papel a soporte digital de forma cómoda y rápida. Lo único que se necesita es un escaner y este programa. ScanToPDF permite pasar los documentos que se escanean directamente a formato pdf, el formato estándar para la producción y gestión de documentos compatible con todos los sistemas operativos y todas las plataformas. Sólo se

tiene que insertar el papel en el escaner y haciendo clic en el botón correspondiente de la interfaz de ScanToPDF, el documento será escaneado, convertido en pdf y guardado en el disco duro. Desde el propio programa se podrá enviar si es necesario el documento directamente por correo electrónico. También funciona con cámaras digitales, permitiendo pasar imágenes a formato pdf. (5) Este software funciona en Sistemas Windows.

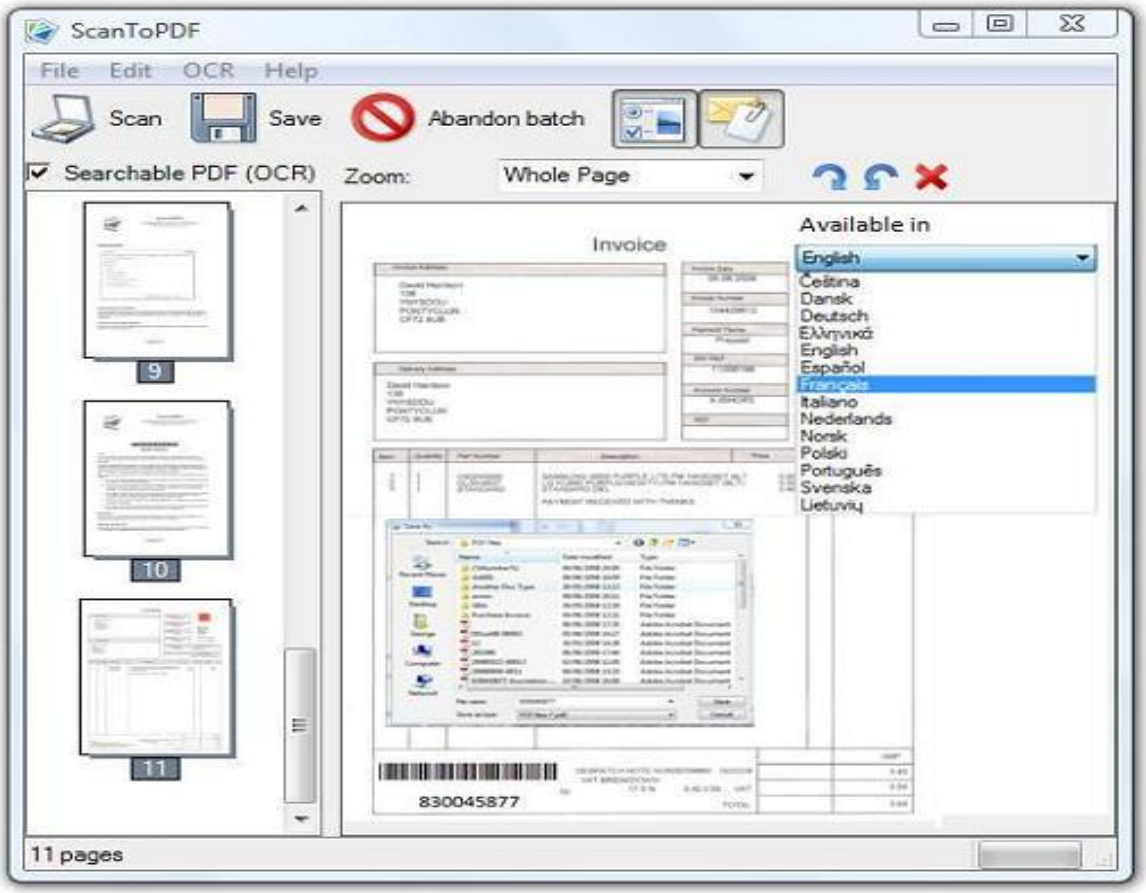


Figura 5: Programa ScanToPDF 3.2.

Kofax Ascent Capture

Obtiene la información electrónica proveniente de múltiples fuentes (escaner, fax, e-formas, XML y más), luego transforma .Puede distribuir los datos directamente en plataformas de administración de contenido.

Kofax Ascent Capture ofrece también herramientas tales como la indización automática, identificación de forma automática, precisa de zona de Reconocimiento Óptico de Caracteres (OCR) y Reconocimiento Inteligente de Caracteres (**ICR**). (6)

Gracias a la perfecta integración con gestores de contenido, Kofax Ascent Capture forma una única plataforma para el escaneo de documentos e imágenes. El resultado: menos silos de información, como datos de diferentes sitios y departamentos se unifican en una sola solución. (6) Este programa se ejecuta en Sistemas Windows y su precio oscila entre $995.00 y $11714.00.

Beneficios

Estrecha integración con varios sistemas gestores de contenidos como Documentum, **IBM** Content Manager, Alfresco, entre otros más.

- A través de Plug-in permite la extensión de sus funcionalidades, estos Plug-in pueden ser desarrollados por terceros; dando la posibilidad de interactuar con otras aplicaciones y transferir la información capturada hacia estas.
- Captura de datos eficientemente.
- Fácil instalación y uso.

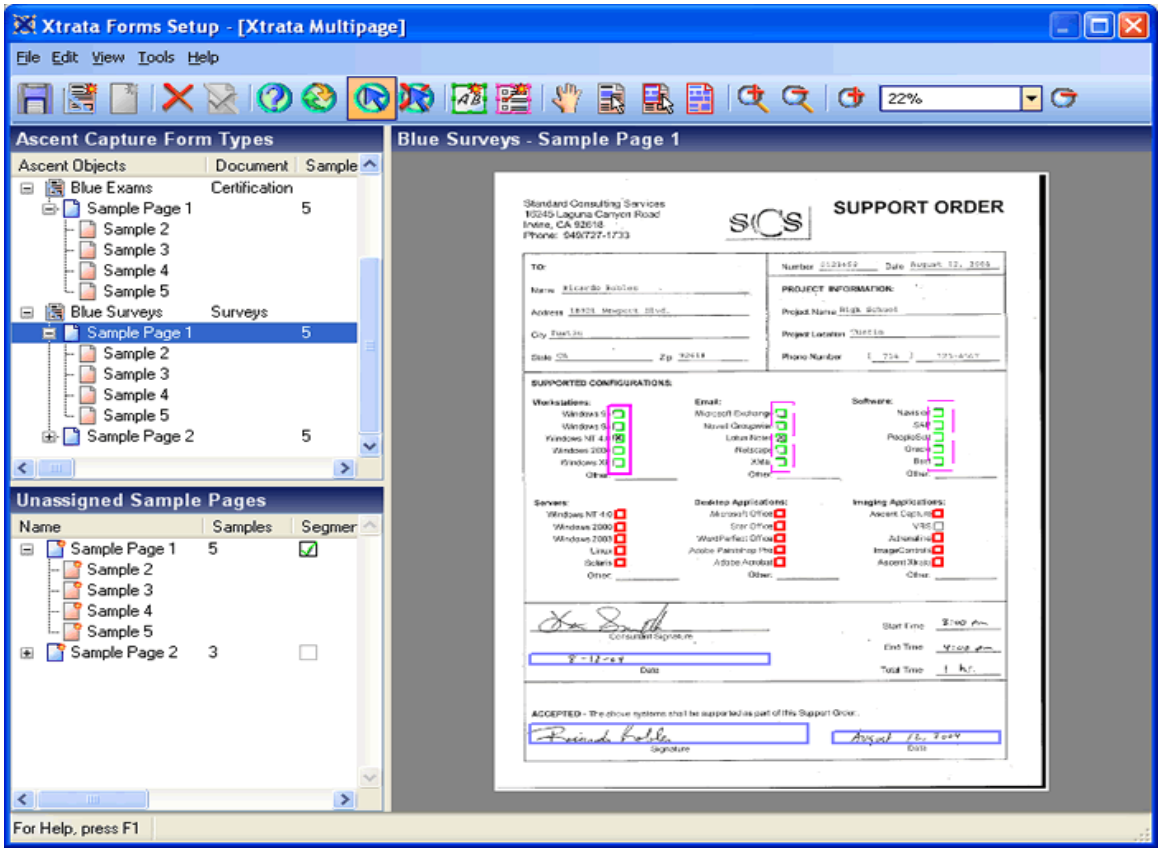


Figura 6: Kofax Ascent Capture.

Xsane

Es una aplicación que hace posible el funcionamiento de un escaner. Con Xsane puedes escanear un archivo, hacer una fotocopia, crear un fax, crear y enviar un correo electrónico y todo ello de forma muy sencilla desde el entorno grafico. El propio programa detecta los escaneres que estén conectados a la PC automáticamente, se selecciona el que se desee y se podrá obtener imágenes del escaner. Esta aplicación tiene licencia GNU GPL.

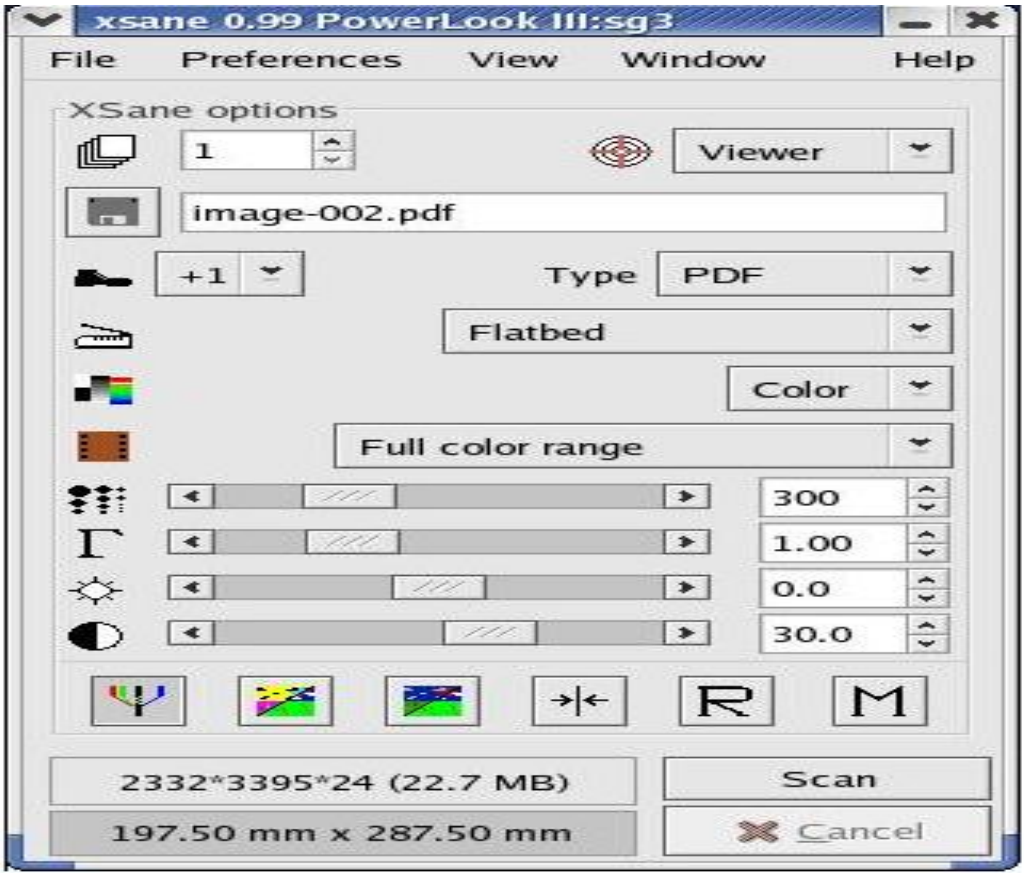


Figura 7: Aplicación Xsane.

1.4 Carencia de aplicaciones para la digitalización de documentos en Software libre

En la actualidad existe un gran avance de las tecnologías a nivel mundial y con ello viene consigo que empresas e instituciones produzcan una gran cantidad de papeles y documentos para el desarrollo de sus

actividades, pues archivar estos documentos en forma digital es una de las mayores necesidades de toda institución o empresa para alcanzar un alto nivel de organización y nivel de desarrollo; y si a esto se le añade el uso de computadoras y posteriormente el escaner como herramientas y dispositivos para la realización de aplicaciones que sean capaces de digitalizar documentos en un entorno libre, entonces se estará más cerca de alcanzar el objetivo, es decir el éxito.

Cuando las instituciones son grandes, sus fondos y documentos más valiosos suelen estar impresos en papel, muchos de estos documentos son antiguos, algunos estropeados por el tiempo y el uso o con problemas de conservación, por ello se requiere de una digitalización para evitar estos problemas existentes.

En estas instituciones con la existencia de grandes volúmenes de archivos para facilitar el análisis y proporcionar una buena planificación, es necesario la realización de aplicaciones para digitalizar documentos que sean eficaces y eficientes.

1.5 Ventajas de la digitalización de documentos

Los documentos digitalizados tienen diversas ventajas sobre los documentos impresos, entre ellas se destacan las siguientes:

1. Se evita el deterioro, pérdida o rotura de documentos originales.

2. Se requiere de inmediata localización además de búsqueda rápida y precisa de los documentos.

3. No existe el riesgo de robo o destrucción de papeles, es decir hay un total control y seguridad de los documentos.

4. Se tiene beneficio de comodidad ya que al tener los documentos en la PC se pueden mandar fácilmente por correo o por fax.

5. Acceso de forma simultánea por múltiples usuarios a la misma o diferente documentación.

6. Se ahorra espacio físico destinado al almacenaje de archivos.

7. Preservación y resguardo de los documentos con el paso del tiempo.

1.6 Metodología, tecnología, lenguajes y herramientas a usar

Para la realización de este epígrafe se realizó un estudio de las metodologías, las tecnologías,

herramientas y lenguajes que posibilitarán el desarrollo de una aplicación libre para la digitalización de documentos en los sistemas que propone el grupo de proyecto Gestión Documental y Archivística de la UCI, pero solo se hace referencia a las que fueron empleadas.

1.6.1 Servicio Web

Por las funcionalidades y avanzada tecnología se decidió utilizar un Servicio Web para salvar los documentos en un repositorio (repositorio del ECM Alfresco), ya que es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar. (7)

El mundo actual está cada vez más interconectado, donde una gran cantidad de aplicaciones tanto empresariales como institucionales y de servicios corren a través de Internet, las empresas necesitan aplicaciones y sistemas que no sean simples, por lo que los Servicios Web toman cada vez más importancia y protagonismo, así como para el manejo interno de una organización o empresa, también para su relación con clientes y proveedores. Estos Servicios Web han venido revolucionando en el mundo de la programación, ofreciendo una inmensa cantidad de ventajas mediante el uso de SOAP.

1.6.2 Lenguaje de programación

Se seleccionó el lenguaje de programación Java, ya que el mismo es un lenguaje avanzado con el que se puede realizar cualquier tipo de programa, desde una aplicación web o desktop hasta aplicaciones para sistemas embebidos. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia en la informática. Es desarrollado por la empresa Sun Microsystems enfocado a cubrir las necesidades tecnológicas más punteras.

Características de Java

1. Metodología de la programación orientada a objetos.

2. Ejecución de un mismo programa en múltiples sistemas operativos.
3. Ejecutar código en sistemas remotos de forma segura.
4. Fácil de usar y tomando lo mejor de otros lenguajes orientados a objetos, como C++.

Una de las principales características por las que Java se ha hecho muy famoso, es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. Actualmente Java se utiliza en un amplio abanico de posibilidades, ya que es posible programar páginas web dinámicas, con accesos a bases de datos y utilizando **XML** con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee hacer con acceso a través de la web se puede hacer utilizando Java.

1.6.3 UML (Lenguaje de Modelamiento Unificado)

Se seleccionó este lenguaje ya que es el más utilizado y conocido en la actualidad por sus características propias. El Lenguaje de Modelamiento Unificado (**UML**) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de funcionalidades concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

1.6.4 Metodología a utilizar

Como metodología a utilizar se escogió **RUP**, es una de las más usadas en el desarrollo del software, por su eficiencia y calidad obtenida a lo largo del ciclo de vida del software, traza una mejor y completa línea de trabajo, proporciona una guía en el orden de las actividades de un equipo, dirige las tareas individuales de los desarrolladores, especifica que productos deberían ser desarrollados y ofrece criterios para monitorear y medir los productos y actividades de un proyecto, así como usar casos de uso en forma efectiva; facilita una interacción continua y clara con el cliente, evitando construir Sistemas de Información que no están acorde a las expectativas finales. Unifica los mejores elementos creados por algunas

metodologías existentes en el desarrollo del software, preparados para desarrollar grandes y complejos proyectos de envergaduras, orientado a objetos, utiliza el UML como lenguaje de representación visual, se caracteriza por ser guiado por casos de uso, iterativo e incremental, y centrada en la arquitectura.

1.6.5 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

1.6.6 Eclipse

Eclipse es una comunidad de fuente abierta, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles y herramientas para crear, desplegar y gestionar software en todo el ciclo de vida. Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Fue desarrollado originalmente por IBM y es ahora desarrollado por la **Fundación Eclipse**, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

1.6.7 Solución tecnológica ECM Alfresco. Gestión de Contenidos Empresariales

Alfresco es una solución 100% Software Libre de Gestión de Contenido Empresarial (ECM), que extiende la Gestión Documental para dar respuesta al ciclo de vida de forma íntegra no solo de los documentos, sino de todos los contenidos de una organización, independientemente de su naturaleza. Alfresco ha sido desarrollado utilizando las últimas tecnologías Java, incluyendo JBoss Application Server 4.0, JBoss Portal 2.0, Spring 1.2, Hibernate 3.0, MyFaces 1.0, Lucene 1.4 y Java 1.5. La versión lanzada por Alfresco ofrece capacidades de gestión de contenidos empresariales en portales compatibles con JSR-168 y JBoss Portal 2.0. Alfresco se estructura en torno a un repositorio de contenidos único, gestionando el almacenamiento de la información en cualquiera de sus formatos nativos, indexando y categorizando los contenidos para su rápida búsqueda y localización, almacenando los metadatos en sistemas de Gestión de Bases de Datos (DBMS).

1.7 Protocolos software estándar y librerías a utilizar

Para el desarrollo de la aplicación se utilizarán varios estándares y librerías que harán posible el diseño del sistema.

1.7.1 SANE

Scanner Access Now Easy (SANE) es un estándar en sistemas UNIX's para la adquisición de imágenes. Proporciona una Interfaz de Programación de Aplicaciones (**API**) que proporciona acceso estandarizado a cualquier dispositivo de escaneo, dígase escaner de sobremesa, escaner de mano, cámaras y videocámaras. El API de SANE es de dominio público y su discusión y desarrollo está abierto a todo el mundo. El controlador de SANE sólo proporciona una interfaz con el dispositivo y describe un número de opciones que controla cada escaneo. (8)

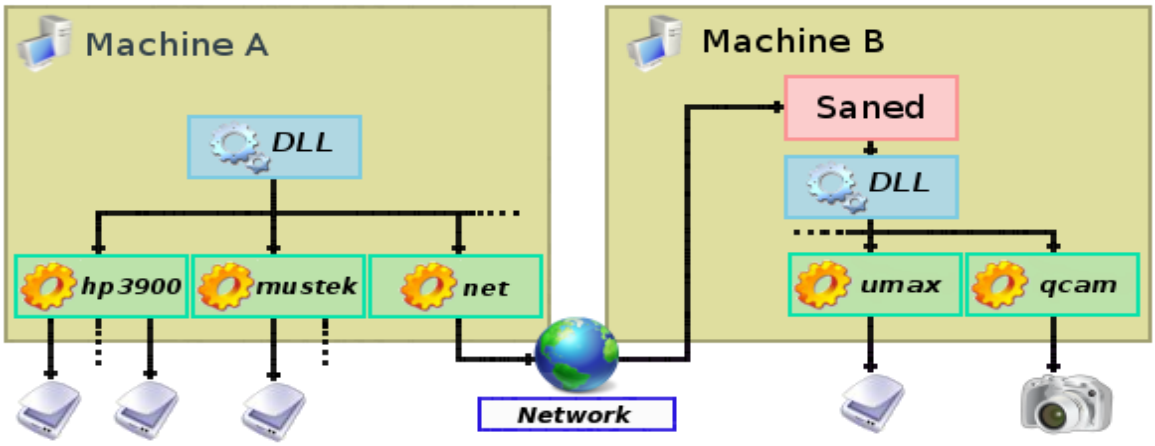


Figura 8: Estructura de SANE.

- Estructura de SANE.

Backends de SANE: Los backends son los controladores que implementan las funciones que proporciona la capa de abstracción de SANE para la comunicación de las aplicaciones de usuario con los diferentes dispositivos de escaneo soportados.

Interfaz dll: Por encima de los backends se encuentra la interfaz dll, una librería de enlace dinámico que puede entenderse como un backend más, ya que implementa las mismas funciones que estos, pero proporciona la posibilidad de utilizar más de un backend.

Frontends de SANE: Los frontends son aplicaciones que hacen uso de la capa de abstracción proporcionada por SANE para acceder a los dispositivos de escaneo. Estos programas no tienen por qué conocer el funcionamiento interno de cada dispositivo ya que de ese trabajo se encargan los backends.

1.7.2 SOAP

SOAP (Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos). Especificación XML para la formación de los mensajes intercambiados entre los sistemas distribuidos y la red. Los mensajes deben tener un formato determinado empleando XML para encapsular los parámetros de la petición. El mensaje está compuesto de tres partes: un sobre, encabezado y el cuerpo. El sobre envuelve al mensaje y contiene el encabezado y el cuerpo; el encabezado es un elemento opcional que provee información para el enrutamiento del mensaje; el cuerpo contiene datos etiquetados como XML, este protocolo es estandarizado por el consorcio **W3C**, que especifica todas las reglas necesarias para ubicar Servicios Web basados con XML. SOAP posee independencia del modo de transporte ya que puede funcionar sobre múltiples protocolos de transporte de textos, como por ejemplo **HTTP**, **HTTPS**, **SMTP**, **FTP**, etc.

Este protocolo facilita la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web. En un principio, ese protocolo se utilizaba para realizar **RPC**, es decir, se podía realizar peticiones mediante HTTP a un servidor web. En el núcleo de los Servicios Web se encuentra el protocolo simple de acceso a datos proporcionando un mecanismo estándar de empaquetar mensajes. Este protocolo ha recibido gran atención debido a que facilita una comunicación del estilo RPC entre un cliente y un servidor remoto.

Algunas ventajas.

1. No está asociado con ningún lenguaje:

Si bien tiene como parámetro XML, los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista, pero los desarrolladores responsables de mantener antiguas aflicciones heredadas, podrían no hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la misma se deja al lenguaje de programación.

2. No se encuentra fuertemente asociado a ningún protocolo de transporte:

Su especificación no describe cómo se deberían asociar los mensajes de SOAP con HTTP. Un mensaje

de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.

3. Aprovecha los estándares existentes en la industria:

Los principales desarrolladores de este protocolo evitaron intencionadamente reinventar las cosas, optaron por extender los estándares existentes para que coincidieran con sus necesidades, por ejemplo, aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema, que ya están definidas en la especificación de esquemas de XML.

4. Permite la interoperabilidad entre múltiples entornos:

Se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una computadora, puede comunicarse con una aplicación, ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP. (9)

1.7.3 TWAIN.

TWAIN fue una iniciativa lanzada en 1992 por los líderes de la industria de la imagen, que reconocieron la necesidad de lograr un protocolo software estándar y una interfaz de programación de aplicaciones (API) que reglamentara la comunicación entre las aplicaciones y los dispositivos de adquisición de imágenes; TWAIN define ese estándar para los sistemas operativos Microsoft Windows y Apple Macintosh, es promovido por el Grupo TWAIN, organización sin ánimo de lucro, el grupo no provee ni distribuye ningún escaner o controlador de dispositivo, solo provee la especificación para que los vendedores puedan escribir los controladores de sus dispositivos basados en estas. Entre los afiliados a la organización se encuentran Kodak (Eastman Kodak Company), Fujitso (Fujitso Computer Products of America), Avision, Epson (Epson), HP (Hewlett-Packard), Xerox (Xerox), entre otras. Algunos de los objetivos de grupo son promover la adopción de la especificación de TWAIN, asegurar compatibilidad entre el software de adquisición de imágenes y el hardware. (10)

Actualmente está ratificado en la versión 2.0 del 28 de noviembre de 2005, en esta versión empieza a soportar arquitecturas de 64 bits y los sistemas operativos tipo Unix y Linux, pero todavía es muy pobre el número de controladores TWAIN para estos sistemas, hasta ahora solo Kodak ha proporcionado

controladores para varios de sus dispositivos.

La desventaja de TWAIN como implementación para una aplicación típica de escaneo, es que no siempre separa la interfaz de usuario del controlador de dispositivo (al contrario que SANE). Esto hace difícil proveer servicios TWAIN a programas ajenos al fabricante del dispositivo. Cada vez que una aplicación carga un controlador TWAIN, no se puede separar de la **GUI**.

- Estructura de TWAIN.

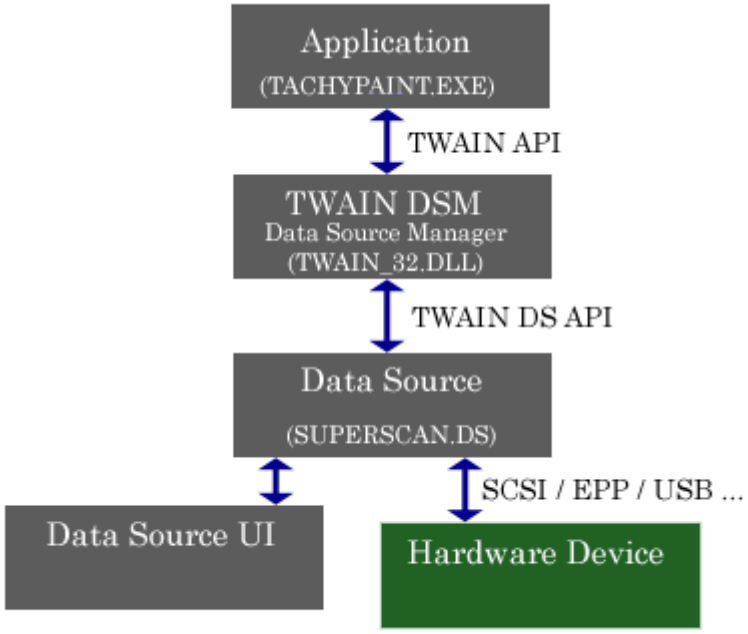


Figura 9: Estructura de TWAIN.

1.7.4 iTEXT.

La librería iText es una biblioteca Open Source para crear y manipular archivos pdf, rtf y html en Java. Está distribuida bajo la Mozilla Public License con la **LGPL** como licencia alternativa. Ha sido extendida a una biblioteca PDF de propósito general, capaz de rellenar formularios, mover páginas de un PDF a otro, y otras cosas. Estas extensiones son a menudo mutuamente excluyentes. Una clase te permite rellenar en formularios, mientras una clase diferente e incompatible hace posible copiar páginas de un PDF a otro.

El soporte de PDF de iText es, sin embargo, bastante extensivo. Esto soporta firmas basadas en **PKI** de PDF, cifrado de 40-bit y 128-bit, corrección de colores, **PDF/X**, gestión de colores por **perfiles ICC**, y es anfitriona de otras características.

1.7.5 Librería mmscomputing.

Es una librería JAVA de código abierto creada por Michael Meiwald y que se ha extendido un poco más por contribuciones de terceros interesados en agregarle más funcionalidades; el objetivo es crear una alternativa libre en el mundo JAVA para el acceso a dispositivos de imágenes. Permite el manejo de escaner para Windows mediante el estándar TWAIN y en sistemas Unix´s a través de Sane. Funciona perfectamente con **JRE** 1.4.2 o superior.

1.7.6 Librería Jsane.

Es un conjunto de ficheros **JAR** que permite que un programa JAVA se comunique mediante el protocolo de red de sane con el demonio saned, se centra en el protocolo de red a través de una implementación pura de JAVA, eliminando cualquier dependencia nativa y logrando una comunicación con el servidor desde cualquier sistema operativo que pueda ejecutar la Máquina Virtual de JAVA (**JVM**).

1.8 Conclusiones

A modo de conclusiones de este capítulo se puede decir que la realización de un estudio profundo de las aplicaciones que digitalizan documentos tanto para software libre como propietario, ayudó grandemente en la comprensión de las ventajas que proporcionaría tener este tipo de aplicaciones en instituciones, empresas y también en nuestra Universidad. Un análisis detallado de las tecnologías facilitó la comprensión necesaria para el desarrollo de la propuesta de solución.

CAPÍTULO 2

Propuesta del Sistema

2.1 Introducción

En este capítulo entre las actividades fundamentales que se desarrollan se encuentra la descripción detallada del proceso que se encuentra involucrado en el dominio, el cual está estrechamente relacionado con la digitalización de documentos y fuertemente asociado a nuestro campo de acción. De dicho proceso se realiza una representación mediante una herramienta, permitiendo dar una visión hacia el sistema que se desea implementar. En este capítulo se realiza una propuesta del sistema que se quiere implementar, luego de haber analizado otros sistemas, pero demostrando lo necesario que se hace implementar la propuesta de solución planteada a continuación. Se lleva a cabo la captura de los requisitos funcionales y no funcionales, dando el punto de partida para empezar a desarrollar el sistema con las características y funcionalidades requeridas, por otra parte se realiza la descripción y representación de los casos de uso del sistema.

2.2 Propuesta del sistema

En el anterior capítulo se realizó un profundo análisis a los distintos software ejemplificados en el estudio del arte de las aplicaciones para digitalizar documentos, y se pudo ver que existen varias aplicaciones muy completas en cuanto a las funcionalidades que requiere este tipo de aplicaciones como OCR, buen tratamiento de imágenes e integración con algún sistema de contenidos , algunos de estos software son por ejemplo, QuickScan Pro, Abbyy FineReader y Kofax, pero los mismos son propietarios y dependientes del Sistema Operativo (SO) Windows, imposibilitando ejecutarse en otros sistemas. Existen además muchas pequeñas soluciones freeware (gratis), que no implica que se tenga acceso al código, pero las mismas también están fuertemente ligadas al SO Windows, por otra parte existen soluciones libres como Kooka y Xsane, estas se centran principalmente en el acceso a dispositivos por el estándar sane y no pueden ser ejecutadas en otra plataforma. A excepción de Kofax y QuickScan, las demás no ofrecen integración con ningún gestor de contenidos. De acuerdo a lo expuesto anteriormente se ha concebido como propuesta de solución al problema científico que se plantea, crear una aplicación que garantice la digitalización de los documentos. La propuesta de solución DocDigital es un módulo para Alfresco a través

de una aplicación Desktop, módulo que pretende satisfacer la necesidad de una aplicación para digitalizar documentos que se integre con la solución de gestión documental que propone el grupo de proyecto Gestión Documental y Archivística.

Se habla de una aplicación libre y multiplataforma que se pueda ejecutar en el sistema operativo Windows de Microsoft y sistemas tipo Unix's como por ejemplo GNU/LINUX, que sea capaz de interactuar con dispositivos de adquisición de imágenes que puedan ser gestionadas por los protocolos software estándar TWAIN en Windows y SANE en Unix's, y a través de estos protocolos mencionados anteriormente se pueda obtener desde el escaner la imagen o imágenes del documento que se desea digitalizar y una vez obtenidas esas imágenes se les pueda hacer una edición mínima si se desea, en este caso la reducción de color, además de poder salvar las imágenes en formatos como png, gif, jpeg y otros, también poder crear con estas imágenes documentos PDF y RTF, y estos archivos generados puedan ser enviados hacia el repositorio de Alfresco mediante el consumo de un servicio web que este provee.

Se propone una aplicación implementada sobre estándares bien definidos como TWAIN y SANE para el acceso a los dispositivos de adquisición de imágenes haciendo el acceso a estos dispositivos transparente al sistema operativo donde se encuentre hospedado el software. El consumo de un servicio web para la comunicación de la aplicación con el Alfresco aprovechando así todas la ventajas que ofrecen el uso de estos servicios.

2.3 Modelo de Dominio

Luego de realizar un profundo análisis se determinó que no es necesario realizar un modelo completo del negocio, por lo que se decide enfocar nuestro proceso a un modelamiento del dominio o modelo conceptual, el cual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software. Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Se considera en RUP un subconjunto del llamado modelo de objetos del negocio y se representa en UML con un Diagrama de Clases en los que se muestra:

- conceptos u objetos del dominio del problema: clases conceptuales

- asociaciones entre las clases conceptuales
- atributos de la clase conceptuales

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. Cualquiera sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo contempla cualquiera de las soluciones analizadas. Es global, es decir se realiza para todos los casos de uso y no para uno en particular.

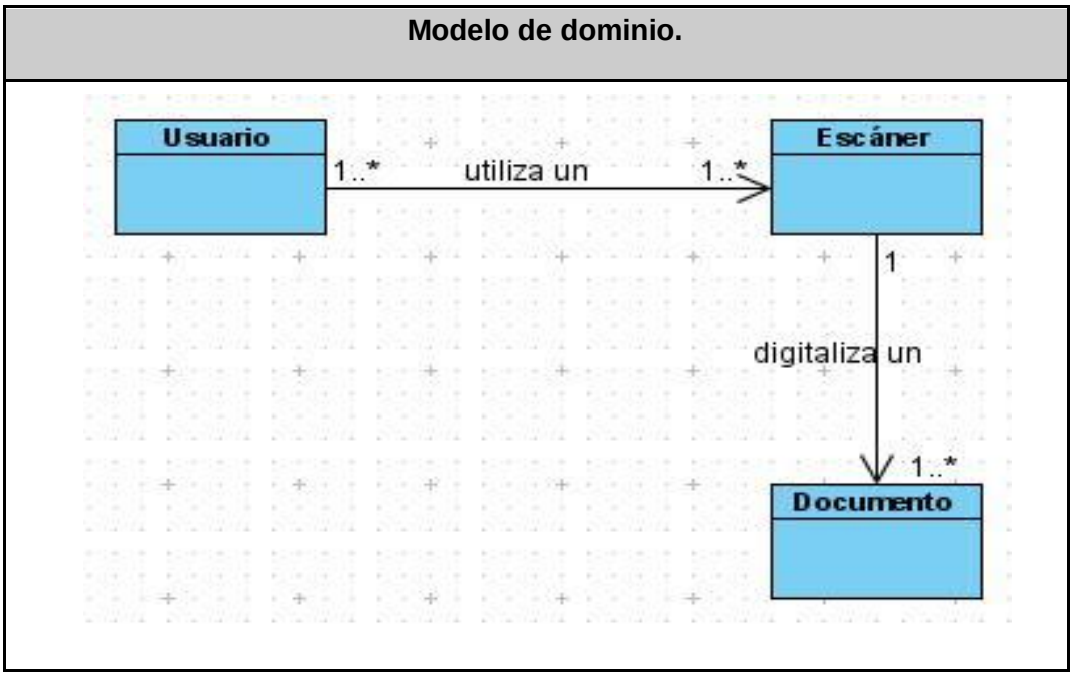


Figura 10: Modelo de dominio.

2.4 Glosario de términos empleados en el desarrollo del modelo de dominio

Usuario: El usuario es la persona interesada en la funcionalidad del sistema para acceder al servicio de digitalización de un documento y luego para asegurar la información lo guarde en un repositorio.

Documento: Un documento es un escrito que contiene información. Es el género documental fijado en un

soporte. Las personas registran su información en cualquier tipo de soporte (papel, cintas, discos magnéticos, películas, fotografías, etcétera) a través de un lenguaje natural o convencional. La información recogida en ese soporte es la fuente documental, término con el que también se designa al origen de esa información.

Escaner: Es un dispositivo externo conectado a una computadora, que hace posible la conversión a formato digital de cualquier documento impreso o escrito en forma de imagen.

2.5 Modelo del sistema

Especificación de requerimientos.

Para una mejor comprensión a continuación se enumeran los requerimientos funcionales, los cuales se modelan en término de casos de uso del sistema para definir las funciones que éste será capaz de realizar, así como los no funcionales que describen de una forma u otra las características que limitan el sistema. Además, se hará un análisis del los principales casos de uso del sistema.

2.5.1 Requerimientos funcionales

RF1- Escanear imagen.

RF1.2- Introducir las opciones de escaneo.

RF2- Seleccionar el escaner.

RF3- Adquirir un escaner ubicado en otra PC.

RF3.1- Configurar opciones de acceso.

RF3.2- Seleccionar escaner.

RF4- Salvar documento.

RF4.1- Indicar destino del documento.

RF5- Guardar imagen

RF6- Salvar en repositorio.

RF7- Cargar una imagen desde el disco.

RF7.1- Seleccionar archivo deseado.

RF8- Rotar imagen.

RF9- Reducir color.

RF10- Deshacer.

2.5.2 Requerimientos no funcionales

1. **Software**
   - ✓ La PC requiere tener instalada la máquina virtual de Java (JRE) 1.5 igual o superior para poder ejecutar la aplicación.
   - ✓ Se necesita la instalación de libsane en Sistemas Unix para el acceso al escaner.
2. **Portabilidad**
   - ✓ Se podrá ejecutar la aplicación en Windows y Linux que tengan instalado la JRE.
   - ✓ La aplicación presenta un mecanismo para salvar el documento en el repositorio, basado en servicio web, haciéndolo independiente del lenguaje y del Sistema Operativo (OS).
3. **Hardware**
   - ✓ Requiere como mínimo de RAM 1 GB.
   - ✓ El disco duro requiere como mínimo 5 GB para almacenar cierta cantidad de documentos.
4. **Restricciones en el diseño y la implementación**
   - ✓ Lenguaje de programación Java.
   - ✓ Librería iText.
   - ✓ Librería Jsane.
   - ✓ Librería libsane.

- ✓ Librería mmscomputing.

5. **Usabilidad**

   - ✓ La aplicación podrá ser ejecutada desde diferentes plataformas de desarrollo.

6. **Legales**

   - ✓ La aplicación y toda la documentación generada pertenecen al grupo de proyecto Gestión Documental y Archivística y a la Universidad de las Ciencias Informáticas.

2.5.3 Actores del sistema

El actor de nuestro sistema es el usuario, el cual es el que interactúa con la aplicación, accediendo al servicio de digitalización de documentos que brinda nuestro sistema. A continuación se muestra en la siguiente tabla su justificación.

Actores del sistema	Justificación
Usuario	Es el que accede a la aplicación para digitalizar un documento.

Tabla 2: Actor del sistema.

2.5.4 Diagrama de casos de uso del sistema

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación. En el diagrama de casos de uso de este sistema (Ver Figura 2-1. Diagrama de CUS), el usuario se relaciona con varios casos de uso, los cuales son ejecutados por el mismo en el sistema.

Diagrama general de caso de uso del sistema.

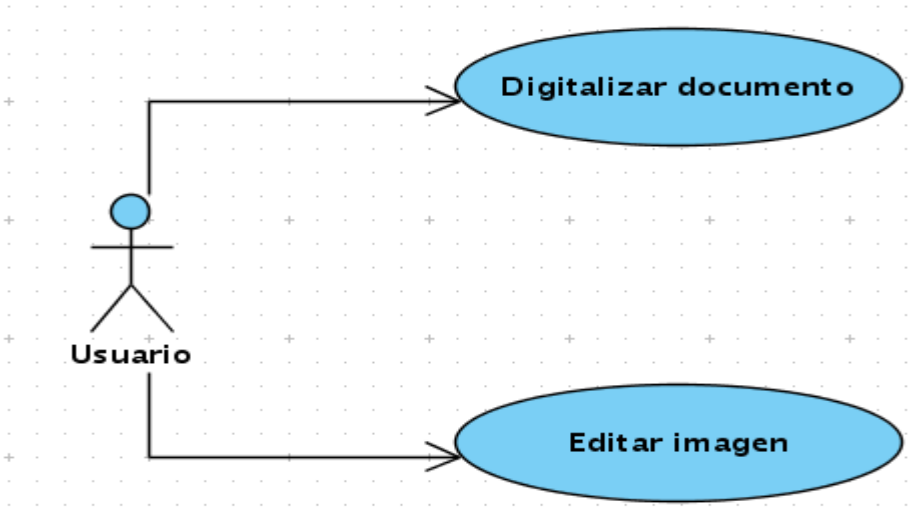


Figura 11: Diagrama de CUS.

2.5.5 Casos de uso del sistema (CUS)

CU-1	Digitalizar documento.
Propósito	Permite digitalizar cualquier documento.
Actor(es)	Usuario

Resumen:	El mismo se inicia cuando el actor accede a la aplicación para escanear una imagen, el mismo puede adquirir un escaner ubicado en otra PC que esté compartido mediante el demonio saned, configurando las opciones de acceso o simplemente seleccionar uno de los escaneres que se encuentra ubicado en la maquina localmente e introducir las opciones de escaneo. Este servicio al realizarlo brinda la posibilidad de salvar la imagen, luego salvarla en un repositorio para su posterior gestión y de esta forma finaliza el caso de uso.
Referencias	RF1, RF2, RF3, RF4, RF6.

Flujo normal de eventos.	
Acción del actor.	**Respuesta del sistema**
1. El usuario accede a la aplicación y selecciona la opción de escanear una imagen. 3. El usuario introduce o configura las opciones de escaneo. 4. Selecciona escanear imagen. 5. Salva el documento en el directorio. 6. Salva el documento en el repositorio.	2. El sistema le brinda la posibilidad de adquirir un escaner ubicado en otra PC que esté compartido mediante el demonio saned o Seleccionar el escaner que se encuentra instalado localmente en la PC. 7. Finaliza el caso de uso

Curso Alternativo1.	
Acción del actor.	**Respuesta del sistema**

Viene del paso 2 del Flujo normal de eventos. 1. Configura las opciones de acceso.	2. Se retorna un mensaje indicando que no hay conexión al servidor.
Curso Alternativo2.	
Acción del actor.	**Respuesta del sistema**
Viene del paso 2 del Flujo normal de eventos. 1. Introduce las opciones de escaneo.	2. Devuelve un mensaje de error en caso de que no haya un escaner instalado.

Tabla 3: Descripción expandida del CUS Digitalizar documento.

CU-2	Editar imagen.
Propósito	Permite cargar una imagen desde el disco y realizarle varios cambios si se desea.
Actor(es)	Usuario
Resumen:	Este caso de uso se inicia cuando el usuario decide desde la aplicación cargar una imagen desde el disco, luego puede o no realizar una serie de opciones con la imagen (reducir color, rotar imagen y deshacer), para luego salvar el documento, salvarlo en el repositorio para gestionarlo si se desea, finalizando el caso de uso.
Referencias	RF5, RF6, RF7, RF8, RF9, RF10.
Acción del actor.	**Respuesta del sistema**

1. El usuario accede a la aplicación y selecciona el servicio que le permite editar una imagen. 3. El usuario selecciona la imagen deseada. 5. Realiza cambios a la imagen si desea. 6. Salva el documento en el repositorio.	2. El sistema le muestra una ventana de selección de archivos. 4. El sistema carga la imagen y la muestra en la aplicación. 7. Finaliza el caso de uso.
Curso Alternativo.	
Acción del actor.	**Respuesta del sistema**
	4. Se retorna un mensaje en caso de que la imagen tenga un error de formato no predeterminado.

Tabla 4: Descripción expandida del CUS Editar imagen.

Capítulo 3: Diseño del Sistema

CAPÍTULO 3

Diseño del Sistema

3.1 Introducción

En el presente capítulo se expone el diagrama de clases de diseño, el cual se decidió que fuese general, donde agrupara todas las clases que intervienen en lo realización de los casos de usos pertenecientes al ciclo de desarrollo de nuestro sistema, con el objetivo de obtener un mejor entendimiento. Además se representan los diagramas de secuencia por escenario de cada caso de uso, además se expone el diagrama de despliegue.

3.2 Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación.

Lo principal de esta etapa es la elaboración del diagrama de clases de diseño, donde se muestra las clases participantes en la ejecución de un caso de uso. A continuación una breve descripción del Diagrama general de clases del diseño (Figura 3-1. Diagrama general de clases del diseño).

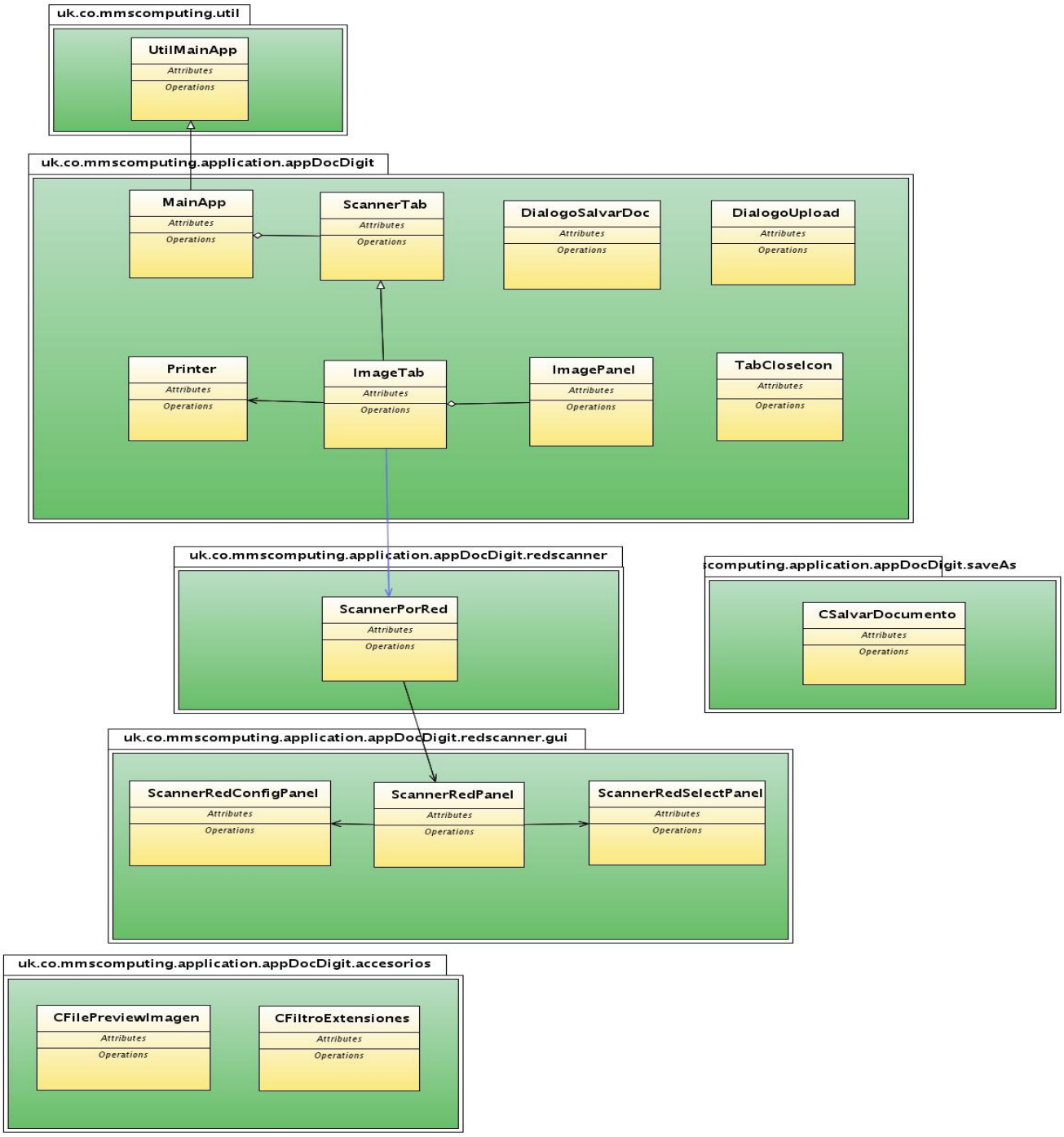


Figura 12: Diagrama general de clases del diseño.

Capítulo 3: Diseño del Sistema

3.3 Estructuración del modelo de diseño

La aplicación cuenta con 6 paquetes y 16 clases lo que permite un mejor entendimiento, funcionamiento y organización del código.

A continuación se representa el paquete **uk.co.mmscomputing.util** que está compuesto por la clase UtilMainApp que es la encargada de construir la clase que recibe el título del Frame y los argumentos pasados a la aplicación.

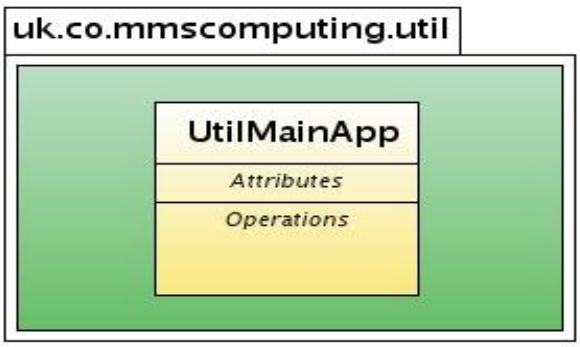


Figura 13: Paquete uk.co.mmscomputing.util.

El próximo paquete representado es **uk.co.mmscomputing.application.appDocDigit** que está formado por las clases MainApp, ImageTab, ScannerTab, TabCloseIcon, DialogoSalvarDoc, Printer, ImagePanel y DialogoUpload.

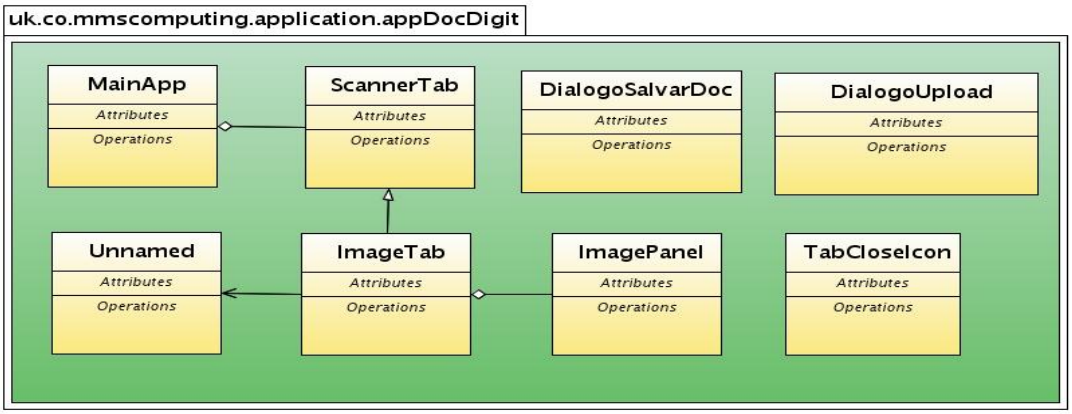


Figura 14: Paquete uk.co.mmscomputing.application.appDocDigit.

A continuación se representa el paquete **uk.co.mmscomputing.application.appDocDigit.saveAs** que está compuesto por la clase CSalvarDocumento.

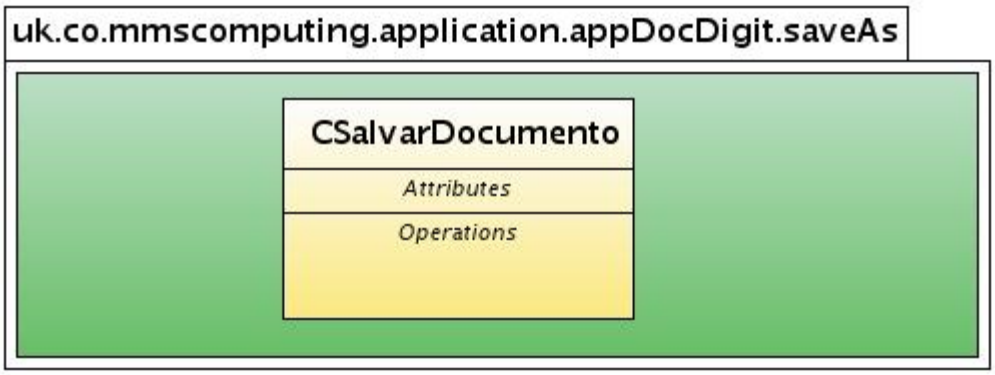


Figura 15: Paqueteuk.co.mmscomputing.application.appDocDigit.saveAs.

El siguiente diagrama representa el paquete **uk.co.mmscomputing.application.appDocDigit.accesorios** que está formado por las clases CfilePreviewImagen y CFiltroExtensiones.

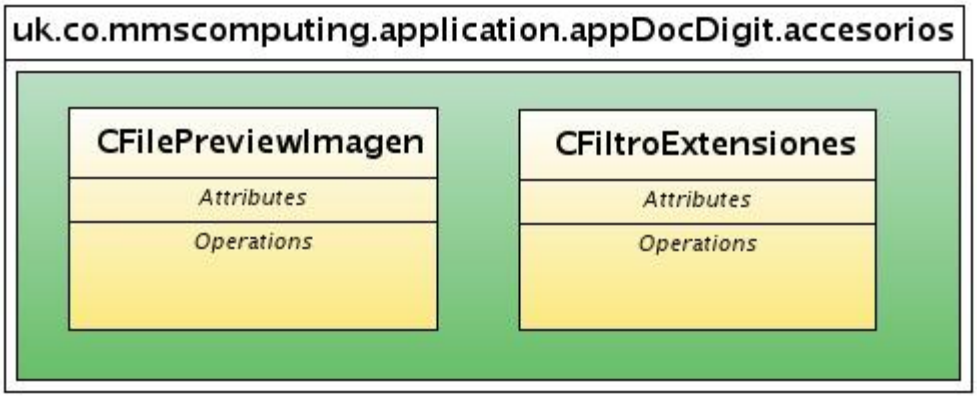


Figura 16: Paquete uk.co.mmscomputing.application.appDocDigit.accesorios.

El paquete **uk.co.mmscomputing.application.appDocDigit.redscanner** está formado por la clase ScannerPorRed.

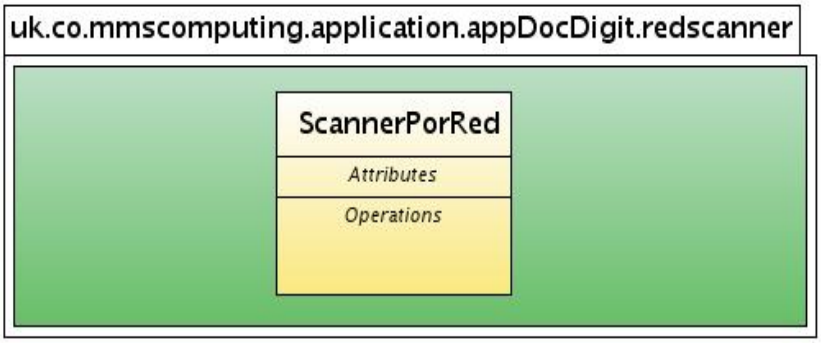


Figura 17: Paquete uk.co.mmscomputing.application.appDocDigit.redscanner.

El siguiente paquete **uk.co.mmscomputing.appDocDigit.redscanner.gui** contiene las clases ScannerRedConfigPanel, ScannerRedPanel y ScannerRedSelectPanel.

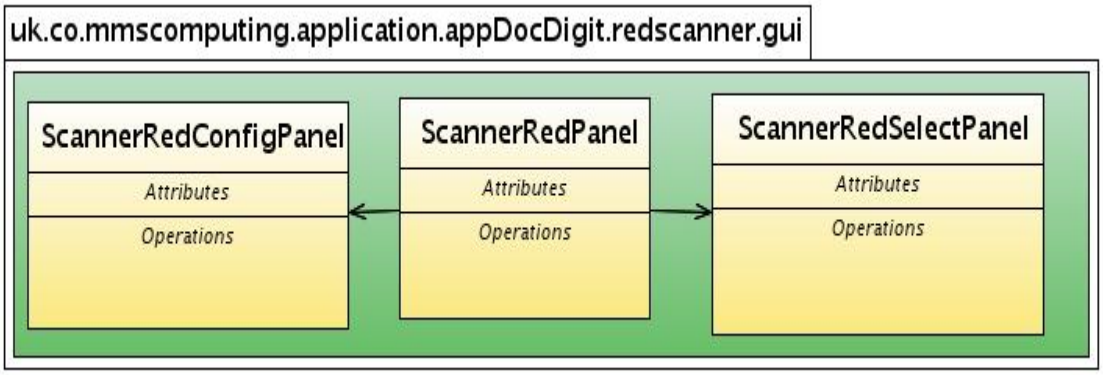


Figura 18: Paquete uk.co.mmscomputing.appDocDigit.redscanner.gui.

3.4 Descripción de las clases del diseño

Nombre: UtilMainApp	
Tipo de clase: Interfaz	
Atributo	Tipo
Properties propertiesFile log menuBar scannerTab	Properties File LogBook JmenuBar JPanel
Para cada responsabilidad:	
Nombre:	UtilMainApp (String title, String[ ] argv)
Descripción:	Constructor de la clase que recibe el título del Frame y los argumentos pasados a la aplicación.

Nombre:	setFrameSize (JFrame frame, Rectangle bounds)
Descripción:	Le da las dimensiones al Frame con las dimensiones del rectángulo entrado como argumento.
Nombre:	getCenterPanel (Properties properties)throws Exception
Descripción:	Método abstracto que retorna una Jpanel.
Nombre:	CreateGUI( )
Descripción:	Método que inicializa los componentes visuales a mostrar en la ventana.
Nombre:	Stop( )
Descripción:	Método que salva las preferencias de la aplicación para futuras ejecuciones de la aplicación.

Tabla 5: Descripción de la clase UtilMainApp.

Nombre: MainApp	
Tipo de clase: Interfaz	
Para cada responsabilidad:	
Nombre:	MainApp(String title, String[ ] argv)
Descripción:	Constructor que crea una ventana con el título pasado por parámetros y recibe los parámetros pasados al programa en su ejecución.
Nombre:	getCenterPanel(Properties properties) throws Exception
Descripción:	Retorna el Panel que será puesto en el centro de la ventana
Nombre:	main(final String[ ] argv)

Descripción:	Método estático de entrada de ejecución de la aplicación.

Tabla 6: Descripción de la clase MainApp.

Nombre: ImageTab	
Tipo de clase: Controladora	
Atributo	Tipo
FileOpenID	String
fileSaveID	String
NEEDIMAGE	String
NONEEDIMAGE	String
EXISTIMAGE	String
TABCHANGE	String
properties	Properties
images	JtabbedPane
openfc, savefc	JfileChooser
toolBar, toolBarEdicion	JtoolBar
progressBar	JprogressBar
applyTodasImagenes, scanfromnet	JtoggleButton
guiNet	Jpanel
acciones	Hashtable<String, Vector<Action>>
menuBarAccions	JmenuBar
netScanner	ScannerPorRed

Para cada responsabilidad:	
Nombre:	ImageTab(Properties properties, JMenuBar toolMenu)
Descripción:	Crea una instancia de ImagePanel, en la barra de menú se insertarán las acciones a las que se podrán acceder, también por la barra de herramientas.
Nombre:	setOpenDir(String path)
Descripción:	Inicializa y configura el selector de ficheros con la ruta al directorio que se quiera abrir por defecto.
Nombre:	setSaveDir(String path)
Descripción:	Inicializa y configura el selector de ficheros con la ruta al directorio que se quiera abrir por defecto.
Nombre:	setButtonPanel(JPanel gui)
Descripción:	Inicializa los componentes gráficos que se mostraran.
Nombre:	getDeshacerConvertAction()
Descripción:	Devuelve la acción de deshacer cambio.
Nombre:	getNewAction()
Descripción:	Devuelve la acción que cerrará todas las imágenes abiertas.
Nombre:	getOpenAction()
Descripción:	Devuelve la acción para abrir una imagen desde el disco duro.
Nombre:	getSaveAction()
Descripción:	Devuelve la acción para salvar la imagen en el disco duro.
Nombre:	getSaveAction()

Descripción:	Devuelve la acción para crear un documento.
Nombre:	getPrintAction()
Descripción:	Devuelve la acción para imprimir las imágenes abiertas en la aplicación.
Nombre:	getConvertAction()
Descripción:	Devuelve la acción para reducir el color a las imágenes.
Nombre:	getRotateAction()
Descripción:	Devuelve la acción que rotará la imagen activa en ese momento.
Nombre:	addImage(String fn, BufferedImage img)
Descripción:	Adiciona una Imagen.
Nombre:	open(String filename)
Descripción:	Abre una imagen dado el nombre del fichero.
Nombre:	getIIOImage(ImageWriter writer, ImageWriteParam iwp, BufferedImage image)
Descripción:	Devuelve una imagen que contendrá los datos de la imagen pasada por parámetro.
Nombre:	save(String filename)
Descripción:	Salva la imagen en el fichero con el nombre y el formato que tenga el parámetro filename.
Nombre:	convertImage(final boolean applySelected)
Descripción:	Le aplica una reducción de color a todas las imágenes si el parámetro es true, de lo contrario solo a la imagen activa.

Nombre:	propertyChange(PropertyChangeEvent evt)
Descripción:	Método para escuchar cambio de propiedades de interés durante la ejecución de la aplicación.
Nombre:	stateChanged(ChangeEvent e)
Descripción:	Método para saber el cambio de la imagen activa.

Tabla 7: Descripción de la clase ImageTab.

Nombre: ScannerTab	
Tipo de clase: Controladora	
Atributo	Tipo
No guiScan scanner	Int Jcomponent Scanner
Para cada responsabilidad:	
Nombre:	ScannerTab(java.util.Properties properties, JMenuBar toolB)
Descripción:	Crea una instancia de ImagePanel, en la barra de menú se insertarán las acciones que se pueden acceder también por la barra de herramientas.
Nombre:	getActionRedScanner()
Descripción:	Retorna la acción que permite seleccionar un escaner por la red que este accesible a través del demonio saned.

Nombre:	setButtonPanel(JPanel gui)
Descripción:	Inicializa los componentes gráficos que se mostraran.
Nombre:	update(ScannerIOMetadata.Type type, final ScannerIOMetadata metadata)
Descripción:	Método de la interfaz ScannerListener, para obtener la imagen escaneada, opciones y excepciones que ocurran durante el proceso de escaneo.
Nombre:	negotiate(ScannerIOMetadata metadata)
Descripción:	Método para negociar y preguntar por las opciones de escaneo.

Tabla 8: Descripción de la clase ScannerTab.

Nombre: TabCloseIcon	
Tipo de clase: Entidad	
Atributo	Tipo
Micon mTabbedPane mPosition closed	Icon JtabbedPane Rectangle Boolean
Para cada responsabilidad:	
Nombre:	TabCloseIcon(Icon icon)

Descripción:	Crea una instancia de TabCloseIcon con el icono que se mostrará, el cual dará la opción para poder cerrar una imagen.
Nombre:	TabCloseIcon( )
Descripción:	Crea una instancia de TabCloseIcon con un icono por defecto.
Nombre:	paintIcon(Component c, Graphics g, int x, int y)
Descripción:	Método encargado de dibujar el icono y donde se pregunta además sise ha hecho click en él para cerrar el tab activo con la imagen.
Nombre:	getIconWidth()
Descripción:	Retorna el ancho del icono.
Nombre:	getIconHeight()
Descripción:	Retorna la altura del icono.

Tabla 9: Descripción de la clase TabCloseIcon.

Nombre: DialogoSalvarDoc	
Tipo de clase: Controladora	
Atributo	Tipo
Imagenes jlistImagenes datosjList imagePanel	JtabbedPane Jlist DefaultListModel ImagePanel

salvarAs formatos ordenOriginalImag	String[] JcomboBox HashMap<String, Integer>
Para cada responsabilidad:	

Nombre:	DialogoSalvarDoc(JTabbedPane jtImagenes)
Descripción:	Crea una instancia de DialogoSalvarDoc, en tbImagenes estarán las imágenes que se quieren formen parte del documento.
Nombre:	getActionAceptar()
Descripción:	Retorna la acción Aceptar que creará el documento.
Nombre:	setNombreImagenesPorPosicion(JTabbedPane jtPanel)
Descripción:	Inicializa al atributo ordenoriginalImag.
Nombre:	getActionCancelar()
Descripción:	Retorna la acción de Cancelar, que cerrará el Dialogo y no se construirá el Documento.
Nombre:	getActionUp()
Descripción:	Retorna la acción Up que subirá un nivel en el orden que tiene la imagen seleccionada.
Nombre:	getActionDown()
Descripción:	Retorna la acción Down que bajará un nivel en el orden que tiene la imagen seleccionada.
Nombre:	valueChanged(ListSelectionEvent e)
Descripción:	Método para escuchar la selección de las imágenes en el

	Jlist y mostrarla.

Tabla 10: Descripción de la clase DialogoSalvarDoc.

Nombre: ScannerPorRed	
Tipo de clase: Entidad	
Atributo	Tipo
PropertyHost propertyHostPort metadata listener preferencias scannerHost, puertoHost coneccion device	String String ScannerIOMetadata Vector<ScannerListener> Properties String Jsane_Net_Connection JSane_Base_Device
Para cada responsabilidad:	

Nombre:	ScannerPorRed(Properties pref)
Descripción:	Crea una instancia de ScannerPorRed, recibe un Properties para que pueda recordar algunas configuraciones de interés.
Nombre:	addScannerlistener(ScannerListener sc)
Descripción:	Adiciona un oyente de ScannerListener sc al listado.
Nombre:	removeListener(ScannerListener sc)
Descripción:	Elimina el oyente del listado.

Nombre:	fireListenerUpdate(ScannerIOMetadata.Type type)
Descripción:	Notifica a los oyentes que ocurrió un evento.
Nombre:	fireExceptionUpdate(Exception ee)
Descripción:	Notifica a los oyentes que ocurrió una excepción.
Nombre:	setImagen(BufferedImage ima)
Descripción:	Inserta la imagen en la metadata que sea utilizada para notificar a los oyentes.
Nombre:	getScanGui()
Descripción:	Retorna el panel gráfico que da acceso a las acciones de acceder a un escaner por la red.
Nombre:	getScannerHost()
Descripción:	Retorna el IP del host.
Nombre:	getScannerPort()
Descripción:	Retorna el puerto del servidor.
Nombre:	GetMetadata()
Descripción:	Retorna la metadata utilizada para notificar a los oyentes.
Nombre:	GetConeccion()
Descripción:	Retorna la conexión con el servidor.
Nombre:	GetDevice()
Descripción:	Retorna el dispositivo seleccionado en el servidor.
Nombre:	setHostAndPort(String host, String port)

Descripción:	Introduce la dirección IP y el puerto donde se ejecuta el demonio saned.
Nombre:	exitConeccion()
Descripción:	Cierra la conexión con el dispositivo en el servidor.
Nombre:	InitConeccion()
Descripción:	Inicia la conexión con el servidor donde se ejecuta el demonio saned.
Nombre:	setDevice(JSane_Base_Device dev)
Descripción:	Inserta el dispositivo seleccionado en el servidor.
Nombre:	scanImage()
Descripción:	Solicita al dispositivo que escanee la imagen.

Tabla 11: Descripción de la clase ScannerPorRed.

Nombre: ScannerRedConfigPanel	
Tipo de clase: Entidad	
Atributo	Tipo
DemonioSaned frame jtHost jtPort	ScannerPorRed Jdialog JtextField JTextField
Para cada responsabilidad:	

Nombre:	ScannerRedConfigPanel(ScannerPorRed scanner)
Descripción:	Crea una instancia de ScannerRedConfigPanel para configurar los parámetros de acceso al escaner.
Nombre:	GetActionCancelar()
Descripción:	Retorna la acción que cierra el diálogo.
Nombre:	showDialogo()
Descripción:	Muestra el diálogo de configuración.
Nombre:	getActionAceptar()
Descripción:	Retorna la acción que inicia la conexión con los parámetros recogidos en el diálogo.

Tabla 12: Descripción de la clase ScannerRedConfigPanel.

Nombre: ScannerRedPanel	
Tipo de clase: Contenedora	
Atributo	Tipo
DemonioSaned configurar seleccionar scan	ScannerPorRed Jbutton Jbutton JButton
Para cada responsabilidad:	
Nombre:	ScannerRedPanel(ScannerPorRed scanner)
Descripción:	Crea una instancia de ScannerRedPanel, que contendrá las

	acciones de acceder a un escaner por la red.
Nombre:	getActionConfigurar()
Descripción:	Retorna la acción que muestra el diálogo de configuración de parámetros para establecer conexión con el demonio saned.
Nombre:	GetActionSelectDevice()
Descripción:	Retorna la acción que muestra el diálogo de seleccionar el dispositivo.
Nombre:	GetActionScanImage()
Descripción:	Retorna la acción que hace el pedido de escaneo al dispositivo.

Tabla 13: Descripción de la clase ScannerRedPanel.

Nombre: ScannerRedSelectPanel	
Tipo de clase: Entidad	
Atributo	Tipo
ListDevices dialogo demonioSaned	Jlist JDialog ScannerPorRed
Para cada responsabilidad:	
Nombre:	ScannerRedSelectPanel(ScannerPorRed sc)
Descripción:	Crea una instancia de ScannerRedSelectPanel que contendrá el listado para seleccionar el dispositivo en el servidor.

Nombre:	GetActionSelect()
Descripción:	Retorna la acción que selecciona y establece la conexión con el dispositivo seleccionado.
Nombre:	GetActionCancelar()
Descripción:	Retorna la acción que cierra el diálogo.
Nombre:	ShowDialogo()
Descripción:	Muestra el diálogo para seleccionar el dispositivo.

Tabla 14: Descripción de la clase ScannerRedSelectPanel.

Nombre: CSalvarDocumento	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	guardarComoPDF(com.lowagie.text.Rectangle sizePagina, JTabbedPane imagenes, HashMap<String, Integer> orden, File out, DefaultListModel dataList)
Descripción:	Crea un documento PDF con las imágenes.
Nombre:	void guardarComoRTF(com.lowagie.text.Rectangle sizePagina,JTabbedPane imagenes, HashMap<String, Integer> orden, File out, DefaultListModel dataList)
Descripción:	Crea un documento RTF con las imágenes

Tabla 15: Descripción de la clase CSalvarDocumento.

Nombre: CfilePreviewImagen	
Tipo de clase: Entidad	
Atributo	Tipo
VistaMiniatura File	BufferedImage File
Para cada responsabilidad:	

Nombre:	CFilePreviewImagen(JFileChooser fc)
Descripción:	Crea una instancia de la clase, fc será el selector de ficheros que mostrará la vista en miniatura de la imagen que se tenga seleccionada en ese momento.
Nombre:	LoadImagen()
Descripción:	Carga la imagen que tenga seleccionada el selector de ficheros y crea su vista en miniatura.
Nombre:	paintComponent(Graphics g)
Descripción:	Pinta la vista en miniatura de la imagen seleccionada.
Nombre:	propertyChange(PropertyChangeEvent evt)
Descripción:	Método a la escucha de la propiedad cambio de fichero del Selector de ficheros.
Nombre:	BufferedImage getScaledInstance(BufferedImage img, int targetWidth, int targetHeight, Object hint, boolean higherQuality)
Descripción:	Retorna una imagen escalada, de la imagen pasada por parámetro, con las dimensiones especificadas.

Tabla 16: Descripción de la clase CfilePreviewImagen.

Nombre: Printer	
Tipo de clase: Entidad	
Atributo	Tipo
Pj pf bk	PrinterJob PageFormat Book
Para cada responsabilidad:	
Nombre:	Printer()
Descripción:	Crea una tarea de impresión.
Nombre:	append(ImagePanel image)
Descripción:	Adiciona la imagen que será impresa en la tarea de impresión creada.
Nombre:	Print()
Descripción:	Manda a imprimir las imágenes adicionadas al trabajo de impresión.

Tabla 17: Descripción de la clase Printer.

Nombre: CFiltroExtensiones	
Tipo de clase: Entidad	
Atributo	Tipo

Descripcion	String
extensiones	String[]
all	CfiltroExtensiones[]
AllFormat	CfiltroExtensiones
jpgFormat	CfiltroExtensiones
tifFormat	CfiltroExtensiones
pngFormat	CfiltroExtensiones
gifFormat	CfiltroExtensiones
bmpFormat	CfiltroExtensiones
ppmFormat	CfiltroExtensiones
wbmpFormat	CfiltroExtensiones
pdfFormat	CfiltroExtensiones
rtfFormat	CFiltroExtensiones
Para cada responsabilidad:	

Nombre:	CFiltroExtensiones(String descripcion, String[ ] ext)
Descripción:	Crea un filtro con su descripción y las extensiones que acepta.
Nombre:	CFiltroExtensiones(String descripcion, String ext)
Descripción:	Crea un filtro con la extensión que acepta.
Nombre:	GetTodosLosFiltros()
Nombre:	GetDescripcion()
Descripción:	Retorna la descripción del o los formatos aceptados por el filtro.
Nombre:	GetExtensiones()

Descripción:	Retorna las extensiones aceptadas por el filtro.
Nombre:	accept(File f)
Descripción:	Retorna verdadero en caso que el fichero sea aceptado por el filtro.

Tabla 18: Descripción de la clase CFiltroExtensiones.

Nombre: DialogoUpload	
Tipo de clase: Interfaz	
Atributo	Tipo
_URL,_PORT,_USER ,_PASS jtURL, jtPORT, jtUSER jtPASS next, cancel properties jfupload jprogressBar card panelCental user credentials	String Jlabel JpasswordField Jbutton Properties JfileChooser JprogressBar CardLayout Jpanel User Credentials
Para cada responsabilidad:	

Nombre:	JDialogUploadConfig(JFrame owner, boolean modal)
Descripción:	Constructor de la clase que recibe el título del propietario o padre del Dialogo y el modo en que debe aparecer el Dialogo.
Nombre:	JDialogUploadConfig(Properties properties)
Descripción:	Constructor de la clase que recibe un Properties para que use y salve algunas preferencias y configuraciones.
Nombre:	Action getOkAction()
Descripción:	Acción para el botón Aceptar. Es el que realiza la llamada de subir los ficheros al repositorio.
Nombre:	Action getCancelAction()
Descripción:	Para cancelar la acción de que no se desea subir los ficheros.
Nombre:	void uploadFiles(File[ ] files)
Descripción:	Función para Sincronizar con el repositorio (Subir ficheros).
Nombre:	boolean validateFields()
Descripción	Función para la validación de los campos de texto.
Nombre:	int isOpenSession()
Descripción	Función para comprobar el inicio de sesión en el repositorio y validarla.
Nombre:	String[ ] filesToFilesnames(File[ ] files)
Descripción	Función que convierte un arreglo de ficheros en un arreglo con los nombres de los mismos.

Tabla 19: Descripción de la clase DialogoUpload.

Nombre: ImagePanel	
Tipo de clase: Interfaz	
Atributo	Tipo
CHANGED image cambios	String BufferedImage Stack<BufferedImage>
Para cada responsabilidad:	

Nombre:	ImagePanel()
Descripción:	Constructor por defecto de la clase.
Nombre:	ImagePanel(BufferedImage ima)
Descripción:	Constructor de la clase que recibe la imagen que se pintará en la pantalla.
Nombre:	BufferedImage getImage()
Descripción:	Retorna la imagen que contiene para mostrar en la pantalla.
Nombre:	setImage(BufferedImage image)
Descripción:	Introduce o modifica la imagen a mostrar en la pantalla y recuerda el cambio hecho.
Nombre:	void atrasCambios()
Descripción:	Método de ayuda para deshacer un cambio que se le haya hecho a la imagen.
Nombre:	Stack<BufferedImage> getStackCambios()
Descripción	Retorna el historial de cambios.

Nombre:	paintComponent(Graphics gc)
Descripción	Método que dibuja la imagen en pantalla.
Nombre:	Dimension getPreferredSize()
Descripción	Retorna el tamaño preferido del componente que será calculado a partir de las dimensiones de la imagen que esté mostrando.
Nombre:	BufferedImage newImage(int w, int h, int t)
Descripción	Retorna una nueva Imagen con el tamaño y tipo especificado.
Nombre:	void rotate()
Descripción	Aplica una rotación a la imagen de 90 grados.
Nombre:	int print(Graphics gc, PageFormat pf, int page)
Descripción	Método que implementa la interfaz Printable para imprimir la imagen.

Tabla 20: Descripción de la clase ImagePanel.

3.5 Representación UML de los diagramas de interacción

Otro objetivo de este capítulo es la elaboración de los diagramas de interacción que muestran gráficamente como lo objetos se comunican entre ellos con el objetivo de cumplir los requerimientos. Esta interacción se puede expresar en diagramas de colaboración y de secuencia. Estos últimos son los que se han desarrollado, los mismos detallan las secuencias de interacciones ordenadas en el tiempo.

Diagrama de secuencia. Digitalizar documento.

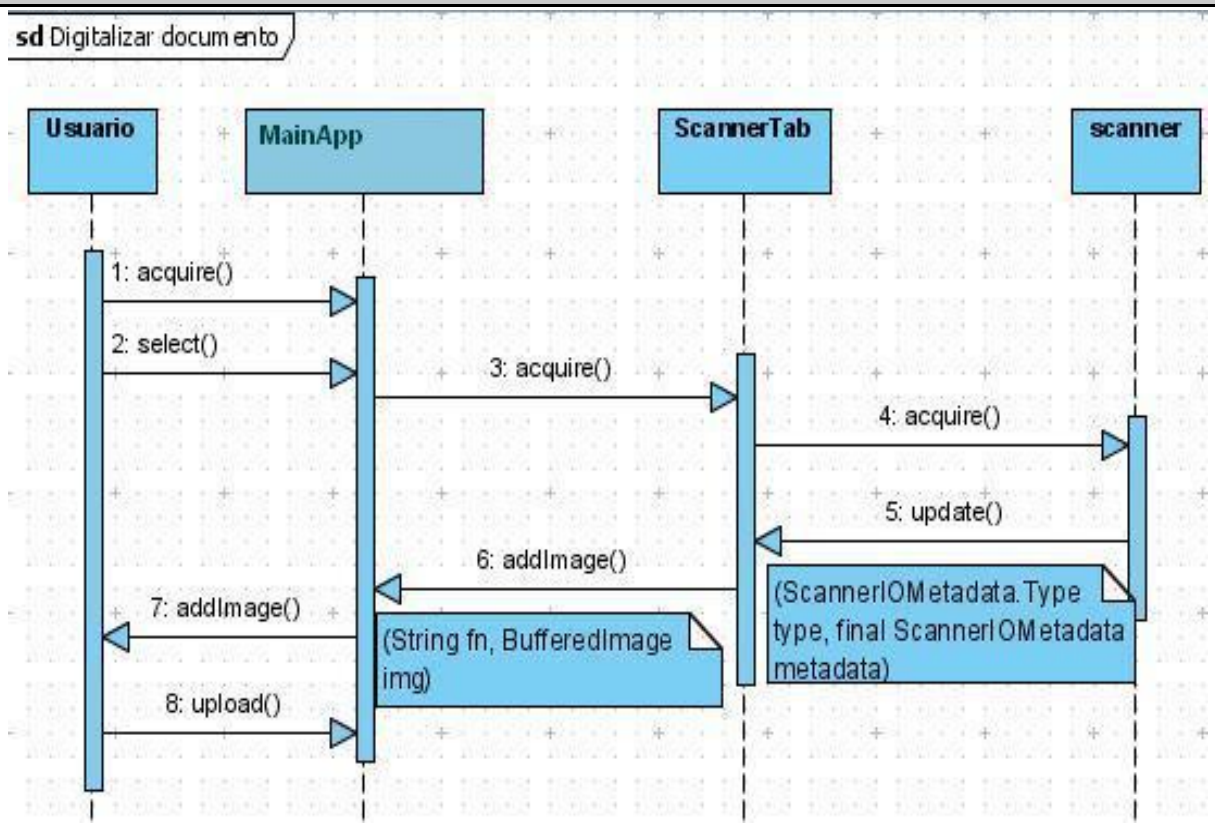


Figura 19: DS. CU_Digitalizar documento.

Diagrama de secuencia. Editar imagen.

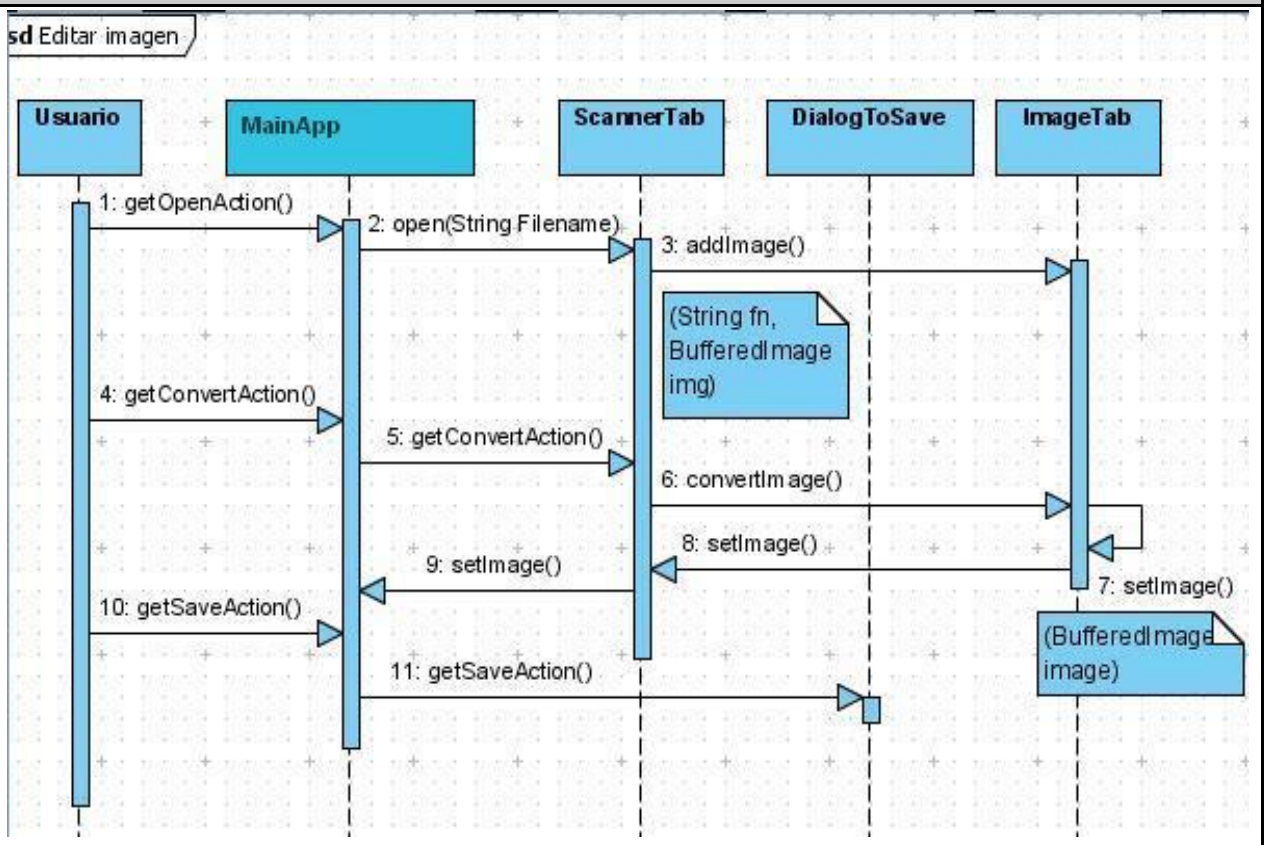


Figura 20: DS. CU_Editar imagen.

3.6 Diagrama de despliegue

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. Las estaciones de trabajo, dispositivos y procesadores son reflejados como nodos. (Figura 4-1.Diagrama de despliegue.)

3.6.1 Representación UML del diagrama de despliegue

En el diagrama de despliegue se representan varios nodos, los cuales son los ordenadores utilizables en el entrono de trabajo, la aplicación DocDigital y el repositorio están ubicados en PCs diferentes y conectados mediante el protocolo SOAP, pues este repositorio garantiza la seguridad de la información creada. La aplicación también está conectada con otra PC cliente mediante el protocolo de comunicación TCP/IP y ambas aplicaciones están conectadas a un escaner mediante USB. Esta PC cliente es un host remoto que tiene instalado el demonio saned que permite el acceso al escaner desde la aplicación..

Diagrama de despliegue.

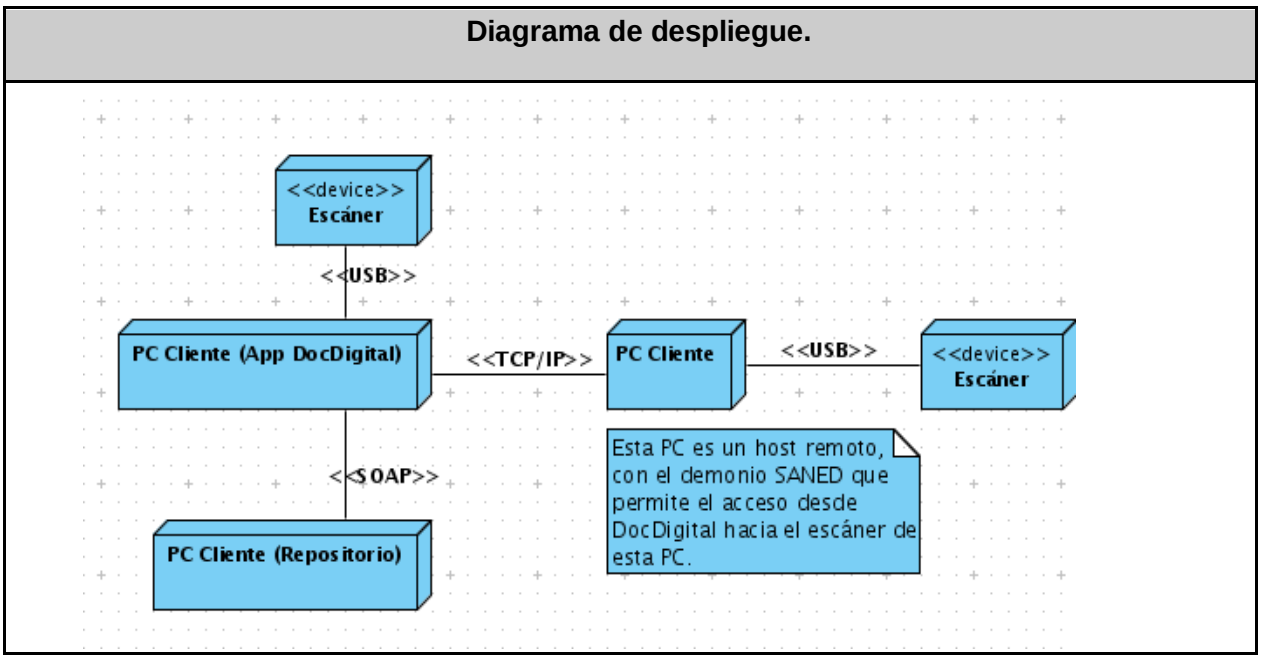


Figura 21: Diagrama de despliegue.

CAPÍTULO 4

Implementación y Prueba

4.1 Introducción

En este capítulo se expone el diagrama de despliegue según nuestra propuesta y el de componentes, del último se realiza un diagrama general, y luego se detallan cada uno de los paquetes por los que está compuesto, esto se hizo para lograr un buen entendimiento y organización del diagrama.

4.2 Diagrama de componentes

El diagrama de componentes define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos. Los componentes son artefactos de software compilados que trabajan acoplados para brindar el comportamiento requerido dentro de las restricciones definidas en el proceso de captura de requisitos. En la (Figura 4-1. Diagrama general de componentes), se puede observar una vista general del diagrama de componentes, así como una vista detallada de cada uno de los paquetes en que se han dividido los mismos, con vistas a lograr una mayor comprensión del modelo.

4.3 Representación UML del diagrama de componentes

Diagrama general de componentes.

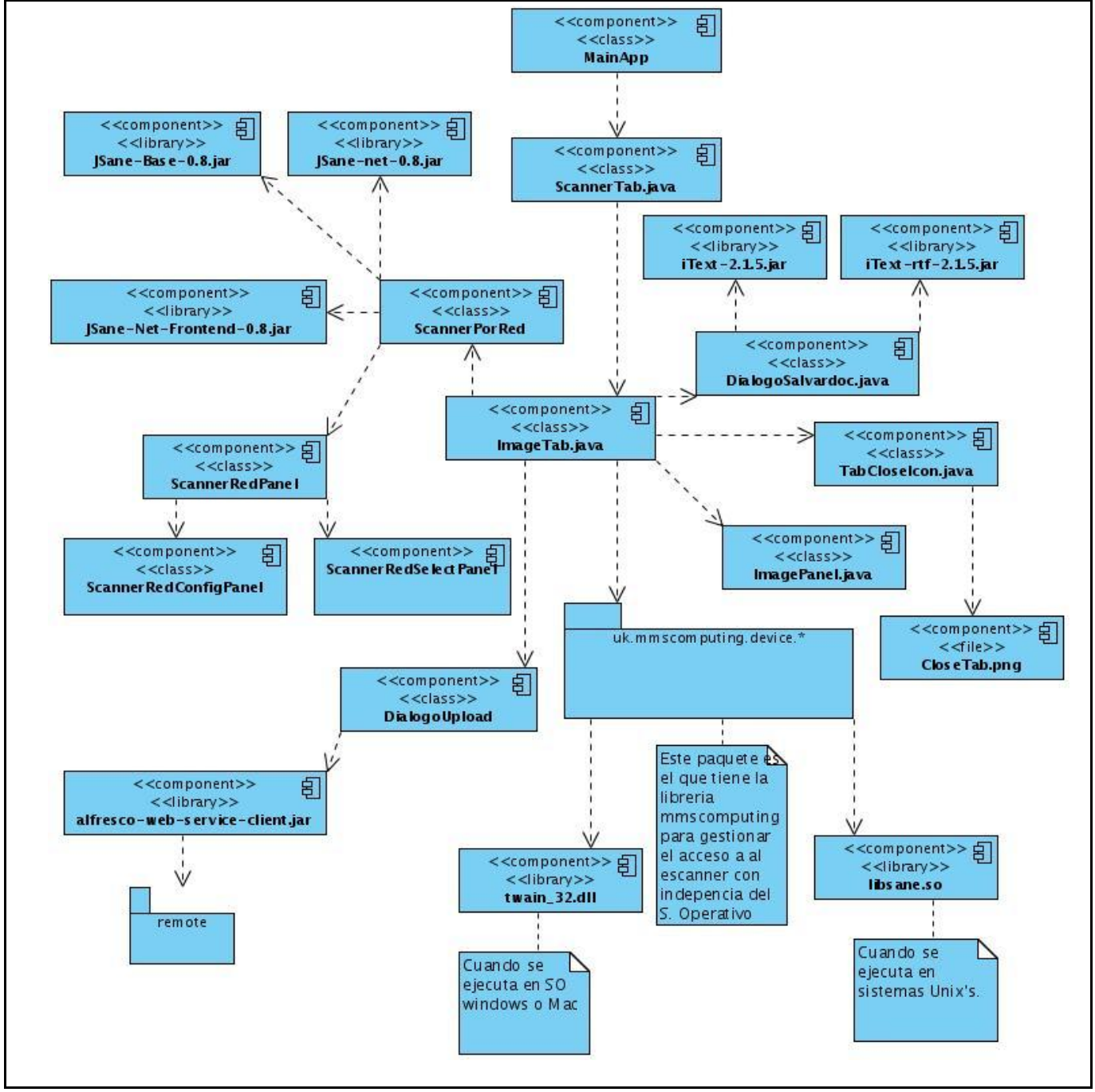


Figura 22: Diagrama de componentes.

4.4 Estructuración del paquete remote del diagrama de componentes

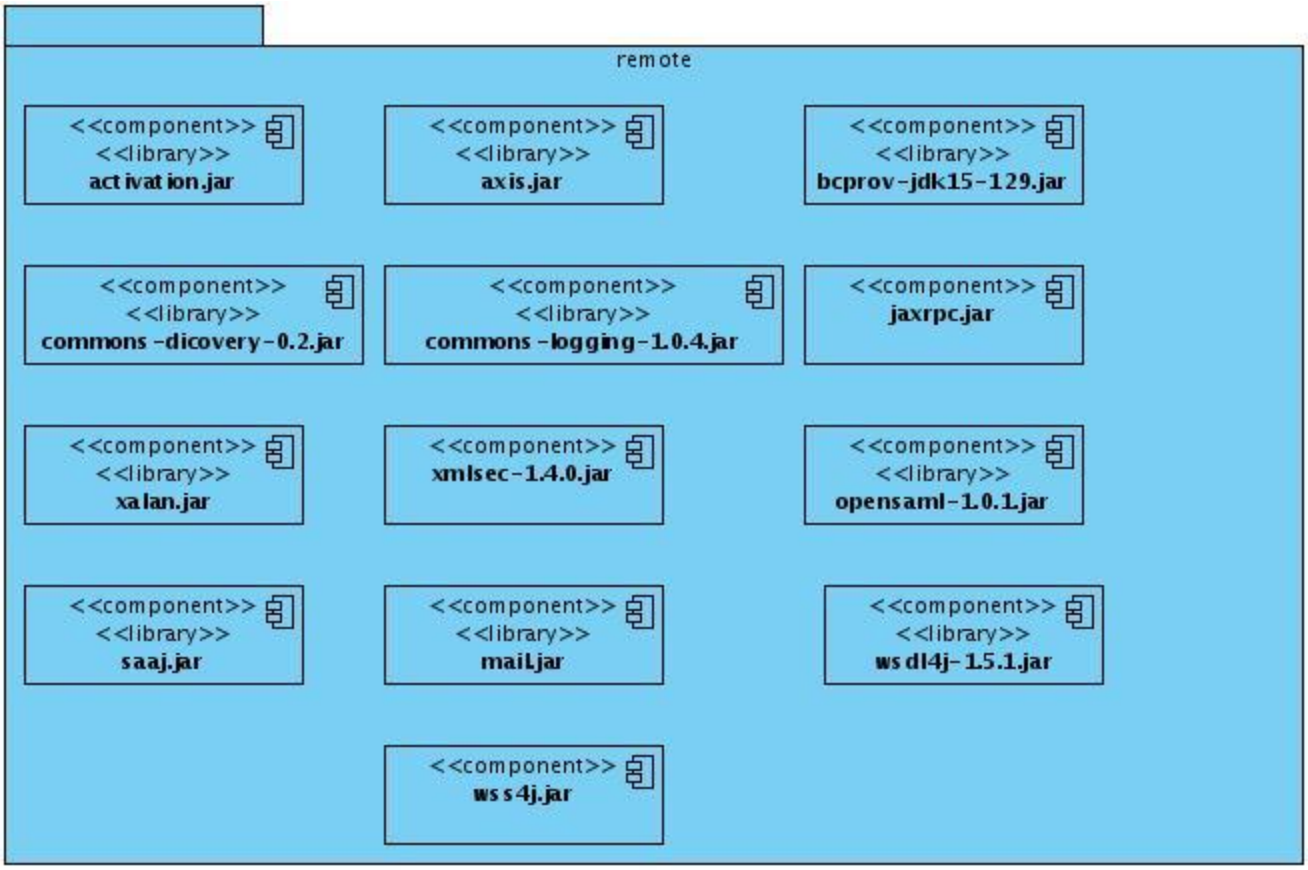


Figura 23: Paquete remote.

4.5 Validación de la propuesta de solución

En el desarrollo del software, las posibilidades de error son innumerables. Pueden darse por una mala especificación de los Requisitos Funcionales (RF), uso indebido de las estructuras de datos, errores al enlazar módulos, etc. El desarrollo del software ha de ir acompañado de alguna actividad que garantice la calidad; la prueba es un elemento crítico para la garantía de la calidad del software. La prueba y validación de los resultados no es un proceso que se realiza una vez desarrollado el software, sino que debe efectuarse en cada una de las etapas de desarrollo. Es fundamental medir la cobertura de las pruebas, es decir, la determinación de cuando se han realizado las suficientes pruebas. Si se siguen encontrando

errores cada vez que se procesa el programa, las pruebas deben continuar. Si se producen modificaciones en el programa, habrá que probar de nuevo todas las partes del programa afectadas por las modificaciones. Para ello se realizarán pruebas de caja negra, que se basan en un minucioso examen de los requerimientos funcionales.

4.5.1 Prueba de caja negra

La prueba de Caja Negra se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.

Muchos autores consideran que estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien. Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

1. Técnica de Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

2. Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

3. Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

4.5.1.1 Descripción de las pruebas de caja negra realizadas

Para la aplicación de este tipo de prueba se usarán los casos de uso “Digitalizar documento” y “Editar imagen”. Se realizarán en conjunto debido a que existe una estrecha relación entre ambos a la hora de interactuar con la aplicación.

Clases Válidas	Clases no Válidas	Resultados	Evaluación	Observaciones
El usuario da clic en el botón “Seleccionar escaner” y se le muestra una ventana para seleccionar el escaner que le permitirá escanear un documento.	EL usuario al dar clic en el botón “Seleccionar escaner” no se le muestre la ventana de selección.	Luego de dar clic se le mostró la ventana de selección de escaner	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en el botón “Escanear imagen” y se le muestra la ventana para la adquisición de imagen y al dar clic en el botón escanear se obtiene la imagen.	Al dar clic en el botón “Escanear imagen” no se le muestre la ventana para adquirir la imagen.	El usuario pudo obtener la imagen.	Satisfactorio	La operación se realizó correctamente.

El usuario da clic en la opción “Acceder a escaner por la red” y se le muestran las opciones de configuración y acceso al escaner remoto.	Al dar clic en la opción “Acceder a escaner por la red” no se le muestren las opciones.	Al dar clic en la Opción configura las opciones.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Configurar” y se le muestra una ventana para configurar el IP y el puerto para el acceso al escaner remoto.	Al dar clic en la opción “Configurar” no se le muestre la ventana para la configuración.	El usuario al dar clic en la opción pudo configurar el IP y el puerto.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Seleccionar” y se le muestra una ventana para seleccionar el escaner remoto.	Al dar clic en la opción “Seleccionar” no se le muestre la ventana para seleccionar el escaner.	El usuario pudo seleccionar el escaner.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en el botón “Escanear imagen” y se escanea la imagen.	Al dar clic en el botón “Escanear imagen” no escanee la imagen.	Al dar clic en el botón se escanea la imager	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la	El usuario al dar clic en	Al dar clic en la	Satisfactorio	La operación se

opción “Rotar imagen” y se rota la imagen en un ángulo de 90°.	la opción “Rotar imagen no se rote la Imagen.	opción se rota la imagen.		realizó correctamente.
El usuario da clic en la opción “Reducir color” y se le muestra una ventana con los parámetros para reducir color y al dar clic en “OK” se le reduce el color a la imagen.	Al dar clic en la opción “Reducir color” no se le reduzca el color a la imagen.	El usuario pudo reducirle el color a la imagen.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Deshacer cambios” y la imagen vuelve al estado en que se encontraba anteriormente.	El usuario al dar clic en la opción “Deshacer cambios” no cambie la imagen.	Al dar clic en la opción se deshace el cambio en la imagen.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Nuevo” y se cierran todas las imágenes abiertas.	Al dar clic en la opción “Nuevo” no se cierren las imágenes.	El usuario al dar clic en la opción se cierran las imágenes.	Satisfactorio	La operación se realizó correctamente.

El usuario da clic en la opción “Abrir” y se le muestra una ventana para seleccionar una imagen en el disco y luego se muestra en la aplicación.	El usuario al dar clic en la opción “Abrir” no se muestre la imagen en la pantalla.	Al dar clic en la opción se muestra la imagen.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Guardar” y se le muestra una ventana para introducir el nombre con el que desea salvar la imagen.	Al dar clic en la opción “Guardar” no se le muestre la ventana para guardar la imagen.	El usuario al dar clic en la opción se guarda la Imagen.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Crear documento” y se le muestra una ventana para crear el documento ya sea PDF o RFT y al dar clic en “Aceptar” se crea el documento.	El usuario al dar clic en la opción “Crear documento” no se le muestre la ventana.	Al dar clic en la opción se crea el documento.	Satisfactorio	La operación se realizó correctamente.
El usuario da clic en la opción “Subir documento al repositorio” y se le	Al dar clic en la opción “Subir documento al repositorio” no se le muestre la ventana para	El usuario al dar clic en la opción pudo subir el o los documentos al	Satisfactorio	La operación se realizó correctamente.

muestra una ventana para introducir los datos de conexión y al dar “Aceptar” selecciona los ficheros para subir al repositorio.	introducir los datos para subir el o los documentos al repositorio.	Repositorio.		

Tabla 21: Descripción de las pruebas de caja negra.

CONCLUSIONES

En la realización de este trabajo de diploma se demostró la necesidad de desarrollar e implementar una aplicación libre para el procesamiento y digitalización de documentos para su posterior gestión en un repositorio del ECM Alfresco, la información que se gestiona es documentación generada por usuarios en los sistemas que propone el grupo de proyecto Gestión Documental y Archivística (GDA), y por cualquier empresa o institución que utilice la aplicación, los principales procesos que brinda la aplicación son: digitalizar un documento y editar una imagen. Para la creación de esta aplicación se elaboró la propuesta de utilizar el lenguaje de programación Java, los estándares y librerías de java para asegurar la portabilidad y un Servicio Web para salvar los documentos en el repositorio Alfresco, además de realizar una estudio detallado de aplicaciones para digitalizar documentos, logrando amplios conocimientos para desarrollar la investigación.

Para llevar a cabo el proceso de desarrollo, se utilizó la metodología RUP y lenguaje de modelado UML a través de la herramienta CASE Visual Paradigm.

Se modeló el proceso de dominio, el cual captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema

Para llevar a cabo el comienzo de la implementación se definieron los requisitos funcionales y no funcionales, así como la modelación y descripción de los casos de uso del sistema los cuales dieron el punto de partida y apoyo para lograr el objetivo final: poner en funcionamiento la aplicación.

En el diseño se elaboró el modelado de los diagramas de clases del diseño y sus paquetes, además de los diagramas de secuencia.

Por otra parte en la implementación se propuso la realización del diagrama de despliegue en el cual se expone la ubicación física de cada componente hardware y software, y el diagrama de componentes para describir los componentes físicos del sistema y sus relaciones. También se realizó el modelo de prueba, utilizando la prueba del camino básico que es una técnica de prueba de caja blanca para la validación de errores en el código fuente.

Luego de realizado todo este trabajo se puede concluir que la aplicación brinda solución a la situación problémica planteada, la cual posibilitó el origen a esta investigación. Su uso y explotación significará una mejora considerable en la calidad y eficiencia de la gestión de la información en el grupo de proyecto

Gestión Documental y Archivística de la Universidad de las Ciencias Informáticas.

Recomendaciones

RECOMENDACIONES

Se recomienda luego de la realización del trabajo:

A la dirección de la facultad y del proyecto:

- *Continuar promoviendo la vinculación temprana, por parte de los estudiantes de la facultad, a proyectos productivos que ayuden en la gestión de la información.*
- *Implementar nuevas funcionalidades para asegurar la calidad de la aplicación.*
- *Presentar el trabajo de diploma en eventos científicos para dar a conocer su importancia.*

REFERENCIAS BIBLIOGRÁFICAS

1. Softonic. [En línea] [Citado el: 7 de diciembre de 2008.] http://abbyy-finereader.softonic.com/.

2. EMC. [En línea] [Citado el: 20 de noviembre de 2008.] http://spain.emc.com/products/detail/software/quickscan-pro.htm.

3. Sitio de descargas de software. [En línea] [Citado el: 20 de noviembre de 2008.] http://www.freedownloadmanager.org/es/downloads/Documalis_Explorador_Libre_47382_p/.

4. **Marcos Dacosta Balboa.** LinWind. [En línea] [Citado el: 20 de noviembre de 2008.] http://www.dacostabalboa.com/es/digitalizar-imgenes-en-linux/527.

5. Softnic. [En línea] [Citado el: 20 de noviembre de 2008.] http://scantopdf.softonic.com/.

6. Kofax. [En línea] [Citado el: 10 de diciembre de 2008.] http://www.kofax.com/.

7. SANE. [En línea] [Citado el: 20 de noviembre de 2008.] http://www.sane-project.org/.

8. **Benjamín González C.** DesarrolloWeb.com. [En línea] [Citado el: 21 de noviembre de 2008.] http://www.desarrolloweb.com/articulos/1557.php.

9. TWAIN. [En línea] [Citado el: 8 de diciembre de 2008.] http://www.twain.org/abouttwain.shtm.

10. **Bruno Lowagie, Paulo Soares.** iText. [En línea] [Citado el: 20 de enero de 2009.] http://www.lowagie.com/iText/.

Bibliografía

BIBLIOGRAFÍA

Softonic. [En línea] [Citado el: 7 de diciembre de 2008.] http://abbyy-finereader.softonic.com/.

EMC. [En línea] [Citado el: 20 de noviembre de 2008.]
http://spain.emc.com/products/detail/software/quickscan-pro.htm.

Sitio de descargas de software. [En línea] [Citado el: 20 de noviembre de 2008.]
http://www.freedownloadmanager.org/es/downloads/Documalis_Explorador_Libre_47382_p/.

Softnic. [En línea] [Citado el: 20 de noviembre de 2008.] http://scantopdf.softonic.com/.

Kofax. [En línea] [Citado el: 10 de diciembre de 2008.] http://www.kofax.com/.

W3C. [En línea] [Citado el: 8 de diciembre de 2008.]
http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb.

SANE. [En línea] [Citado el: 20 de noviembre de 2008.] http://www.sane-project.org/.

TWAIN. [En línea] [Citado el: 8 de diciembre de 2008.] http://www.twain.org/abouttwain.shtm.

Blog Oficial Estudiored. [En línea] [Citado el: 15 de noviembre de 2008.]
http://www.estudiored.es/definiciones-artes-graficas-v/.

Jorge Pozo Alonso. Gestiberia. [En línea] [Citado el: 16 de enero de 2009.]
http://www.informacionydocumentacion.com/digitalizacion.php.

W3Schools.com. [En línea] [Citado el: 18 de enero de 2009.]
http://www.w3schools.com/soap/soap_intro.asp.

Miguel Angel Alvarez. DesarrolloWeb.com. [En línea] [Citado el: 18 de enero de 2009.]
http://www.desarrolloweb.com/articulos/497.php.

Patricio Salinas Caro, Nancy Histchfeld K. *Tutorial de UML.* [En línea] [Citado el: 20 de enero de 2009.]
http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html.

Sitio de descargas de software. [En línea] [Citado el: 20 de enero de 2009.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/.

Eclipse. [En línea] [Citado el: 21 de enero de 2009.] http://www.eclipse.org/.

Bibliografía

Bruno Lowagie, Paulo Soares. iText. [En línea] [Citado el: 20 de enero de 2009.] http://www.lowagie.com/iText/.

Eclipse. [En línea] [Citado el: 20 de enero de 2009.] http://www.eclipse.org/.

Bruno Lowagie, Paulo Soares. iText. [En línea] [Citado el: 21 de enero de 2009.] http://www.lowagie.com/iText/.

Gustavo Suhit. Informateando. [En línea] [Citado el: 23 de enero de 2009.] http://gsuhit.blogspot.com/2008/11/jtwain-utilizando-el-scanner-desde-java.html.

Sourceforge.net. [En línea] [Citado el: 5 de febrero de 2009.] http://sourceforge.net/projects/jsane-net.

Expendedora: Modelo de Dominio. [En línea] [Citado el: 10 de marzo de 2009.] http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf.

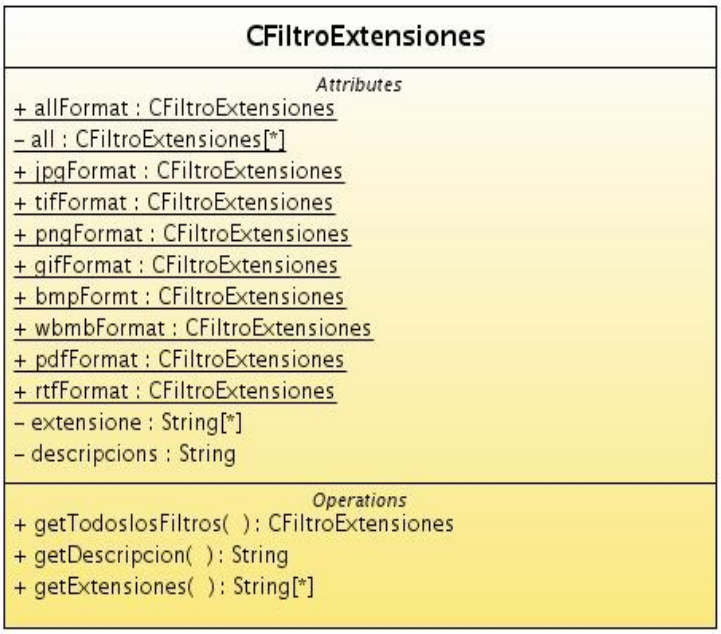
Saga Soluciones Tecnológicas. [En línea] [Citado el: 25 de abril de 2009.] http://www.sagasoluciones.com/nav/soluciones/gestiondocumental/.

José A. Mañas . *Prueba de Programas.* [En línea] [Citado el: 5 de mayo de 2009.] http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm.

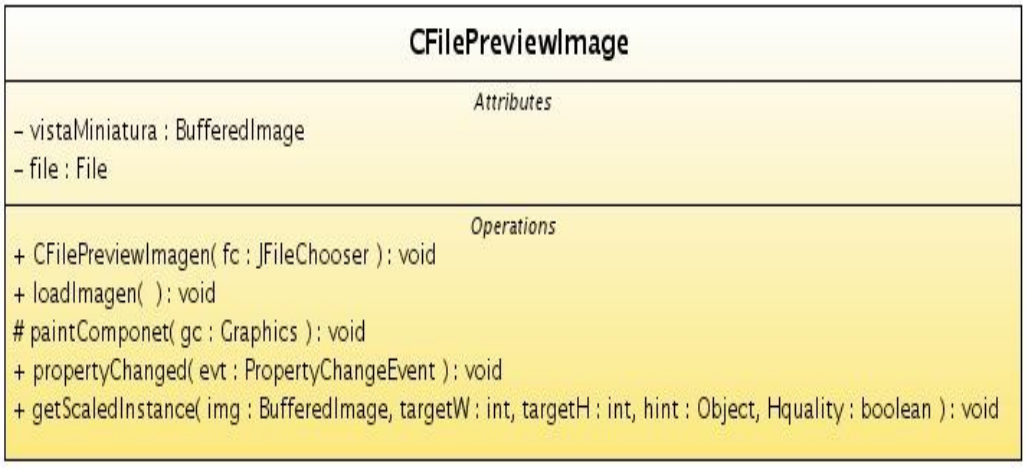
Informatizate. [En línea] [Citado el: 10 de diciembre de 2008.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Zukowski, John. 2003. *Programación Java 2 J2SE 1.4.* Madrid: Anaya Multimedia, 2003. 84-415-1559-X.

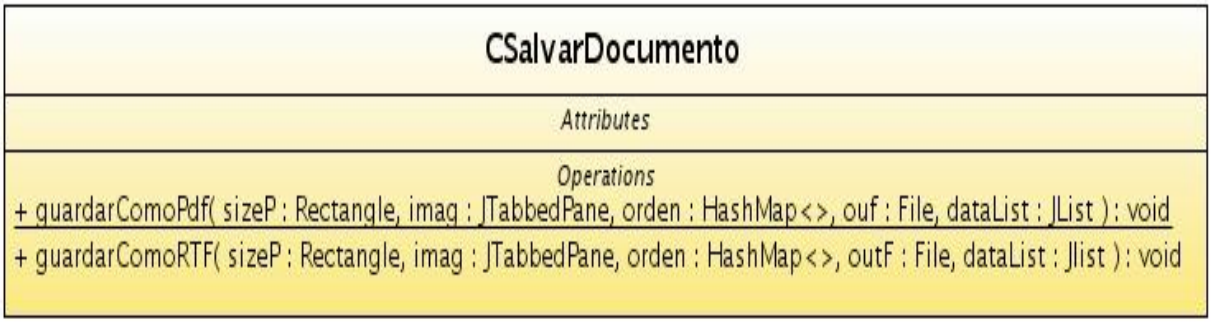
ANEXOS.



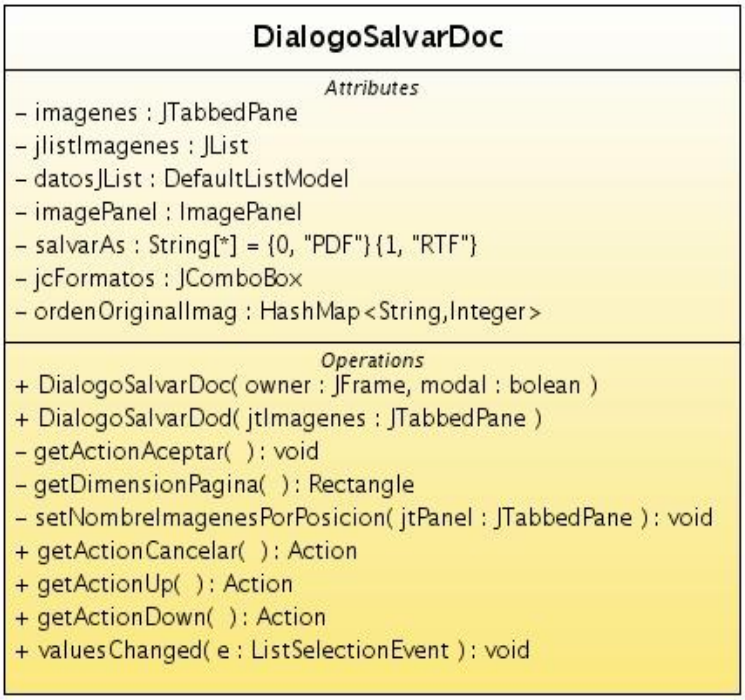
Anexo 1: Clase CFiltroExtensiones.



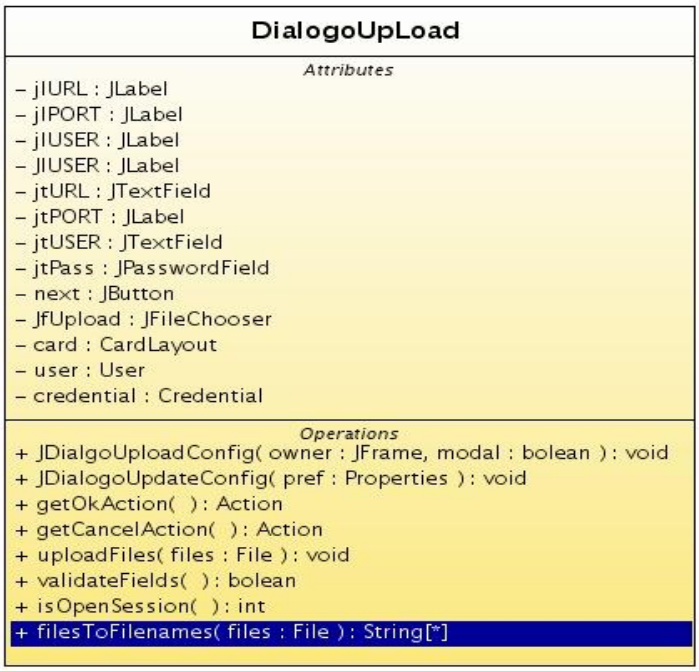
Anexo 2: Clase CFilePreviewImage.



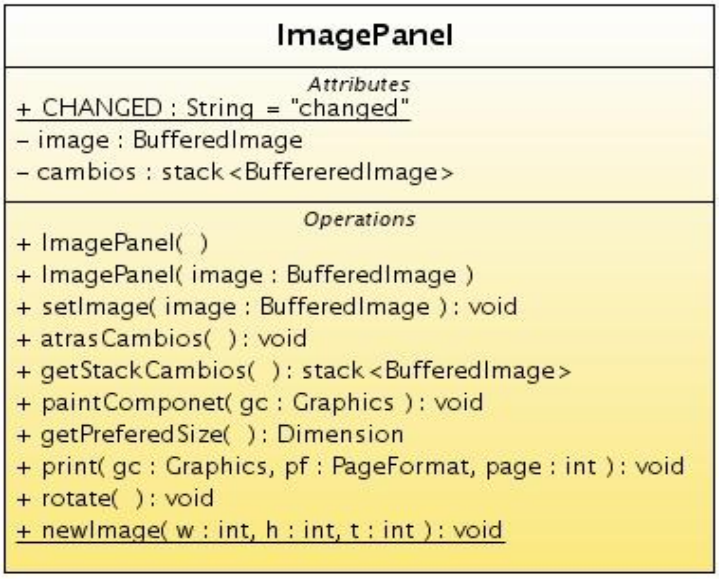
Anexo 3: Clase CSalvarDocumento.



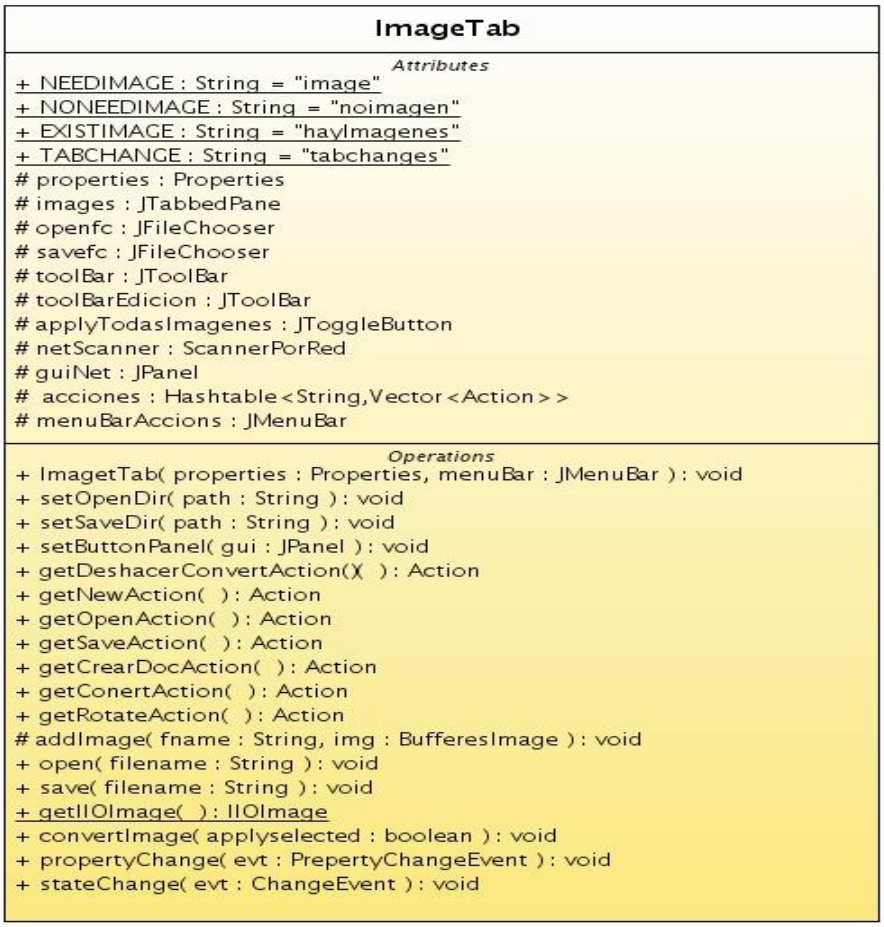
Anexo 4: Clase DialogoSalvarDoc.



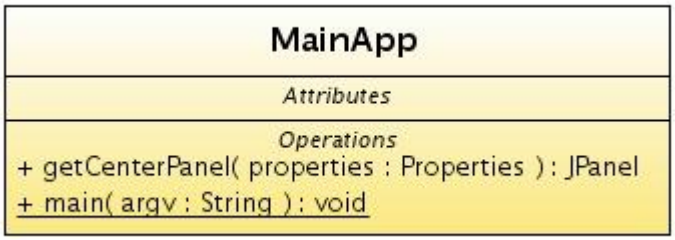
Anexo 5: Clase DialogoUpload.



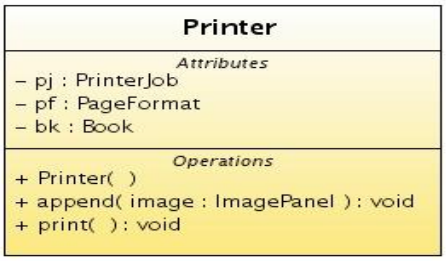
Anexo 6: Clase ImagePanel.



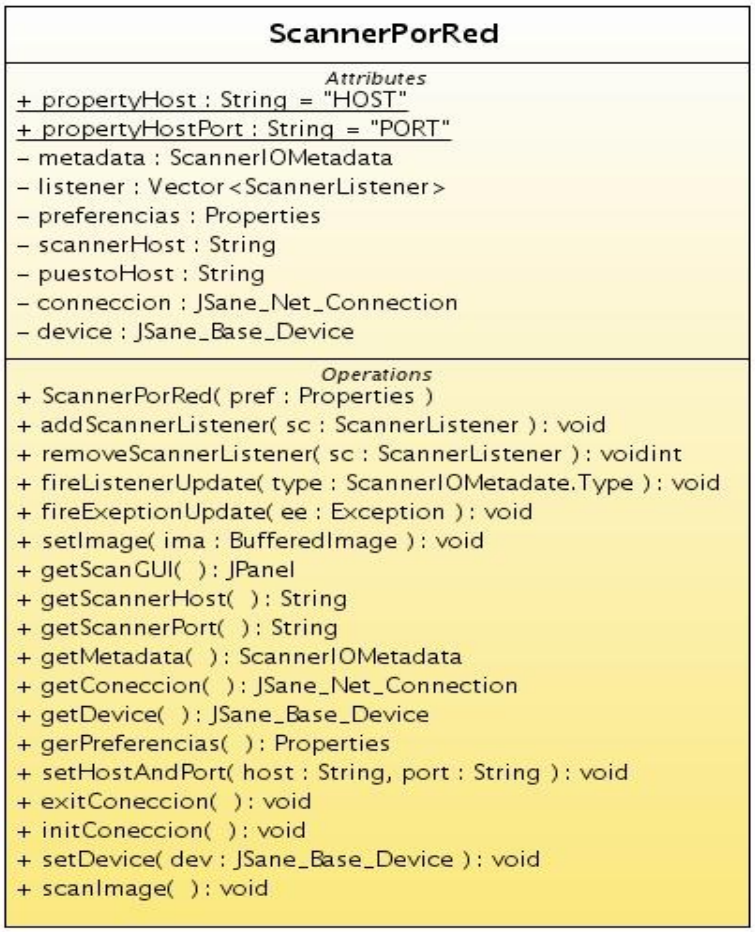
Anexo 7: Clase ImageTab.



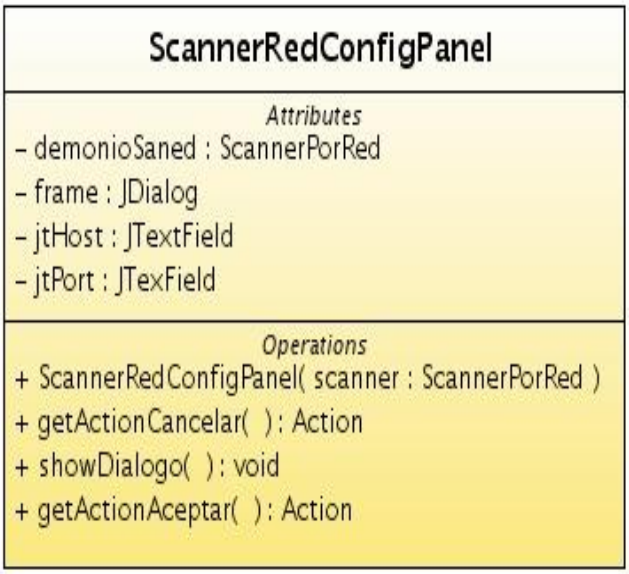
Anexo 8: Clase MainApp.



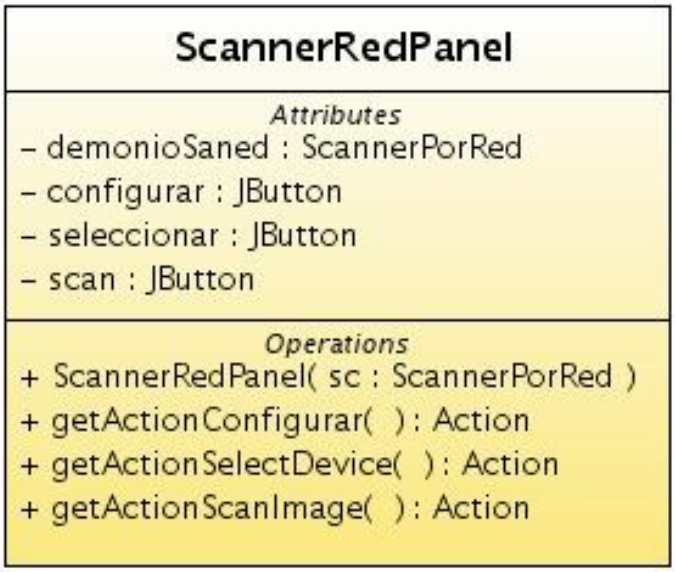
Anexo 9: Clase Printer.



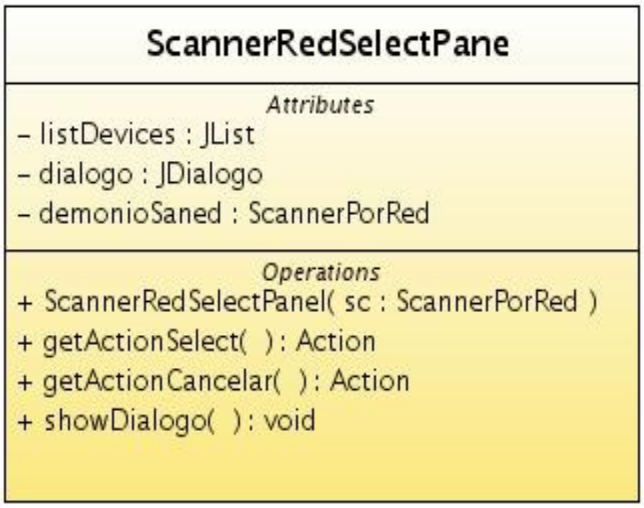
Anexo 10: Clase ScannerPorRed.



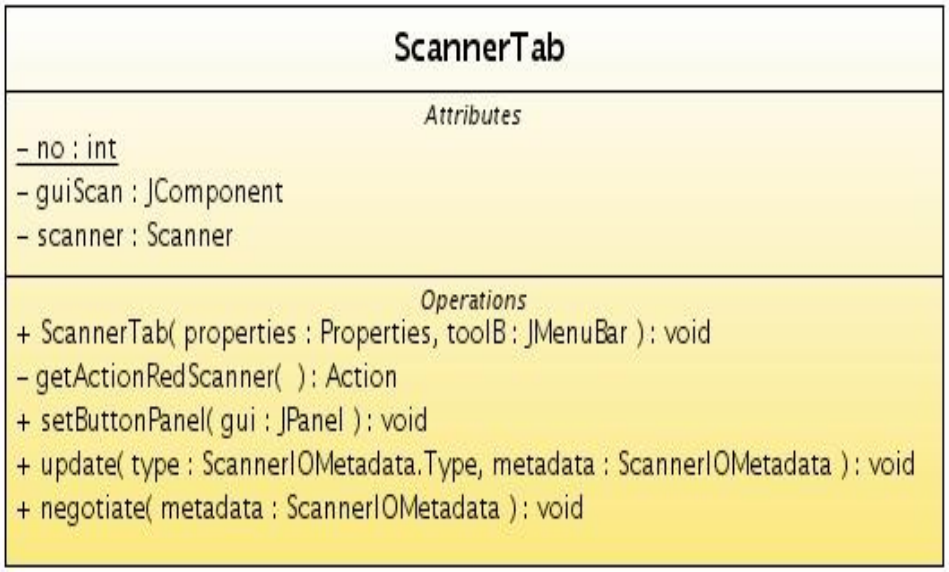
Anexo 11: Clase ScannerRedConfigPanel.



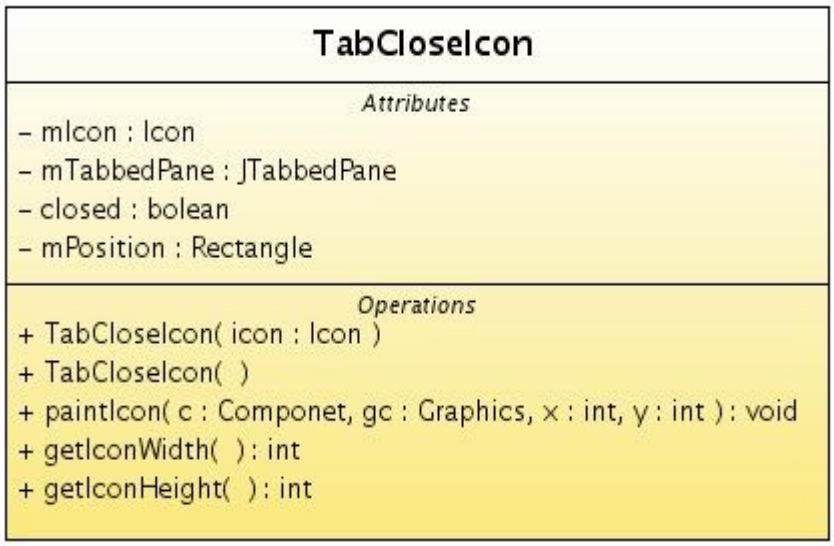
Anexo 12: Clase ScannerRedPanel.



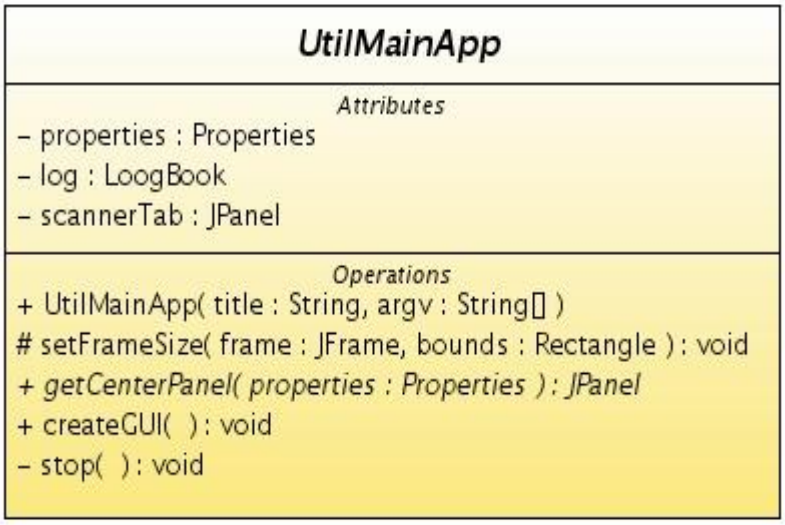
Anexo 13: Clase ScannerRedSelectPanel.



Anexo 14: Clase ScannerTab.



Anexo 15: Clase TabCloseIcon.



Anexo 16: Clase UtilMainApp.

GLOSARIO DE TERMINOS

Término	Definición
API	API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
CASE	Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.
CUS	Caso de uso del sistema.
ECM	ECM (Manejo de Contenido empresarial). Plataforma que incluye un repositorio de contenidos a escala empresarial
FTP	Protocolo de Transferencia de Archivos. Es el protocolo de comunicaciones que permite la interconexión entre ordenadores y la transferencia de ficheros.
Fundación Eclipse	La Fundación Eclipse es una organización sin fines de lucro, lidera el desarrollo de Eclipse, el IDE y plataforma de código abierto para el desarrollo de aplicaciones en Java.
GPL	GPL (General Public License). Es una licencia creada por la Free Software

	Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.
GUI	GUI (Grafical User Interface o Interfaz Gráfica de Usuario).
HTTP	Protocolo de Transferencia de Hipertexto. Protocolo para transferir archivos o documentos hipertexto a través de la red.
HTTPS	Protocolo Seguro de Transferencia de Hipertexto. Garantiza la seguridad de las comunicaciones entre el usuario y el servidor web al que este se conecta.
ICR	ICR (Reconocimiento Inteligente de Caracteres). Proporciona a los sistemas de reproducción por escaner y sistemas de imágenes la habilidad de convertir caracteres en letra manuscrita (no cursiva) en caracteres capaces de ser interpretados o reconocidos por una computadora.
IBM	IBM (International Business Machines). Conocida coloquialmente como el Gigante Azul, es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.
JRE	JRE (Java Runtime Environment o Entorno en Tiempo de Ejecución Java). Conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.
JAR	JAR (Archivo JAR o Java ARchive). Es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java.
JVM	JVM (Java Virtual Machine o Máquina Virtual Java). Es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y

	ejecutar instrucciones expresadas en un código binario especial
KDE	KDE (K Desktop Environment o el Entorno de Escritorio K). Es una red transparente contemporáneo entorno de escritorio de trabajo UNIX.
LGPL	LGPL (Lesser General Public License o Licencia Pública General Reducida de GNU). Es una licencia de software creada por la Free Software Foundation. Pretende garantizar su libertad de compartir y modificar el software "libre", esto es para asegurar que el software es libre para todos sus usuarios.
OCR	OCR (Reconocimiento Óptico de Caracteres). Extrae de una imagen los caracteres que componen un texto para almacenarlos en un formato con el cual puedan interactuar programas de edición de texto.
PDF	PDF (Portable Document Format o Formato de Documento Portátil). Es un formato de almacenamiento de documentos
PDF/X	Subconjunto del formato PDF especialmente destinado a artes gráficas.
Perfiles ICC	ICC (International Color Consortium). Permite adaptar los periféricos utilizados para las imágenes (impresoras, escaneres, monitores, cámaras fotográficas digitales) para que los colores de una imagen, una vez impresa, sigan siendo fiel a los del original
PKI	PKI (Infraestructura de Clave Pública o Public Key Infrastructure). Es una combinación de hardware y software, políticas y procedimientos de seguridad que permiten la ejecución con garantías de operaciones criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas.

Glosario de términos

RF	Requisito Funcional.
RNF	Requisito no Funcional.
RPC	Llamada a Procedimiento Remoto. Permite que un programa ejecute una subrutina o procedimiento en un espacio de direcciones diferente (comúnmente otro ordenador) sin que el programador tenga que codificar explícitamente los detalles de la interacción remota.
RTF	RTF (Rich Text Format).
RUP	El Proceso Unificado de Racional es un proceso de desarrollo de software
SMTP	SMTP (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo electrónico). Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos.
SOAP	Protocolo Simple de Acceso a Objeto. Protocolo para el intercambio de mensajes XML entre redes, comúnmente utilizado sobre el protocolo HTTP.
SWL	Software Libre (SWL) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.
UML	Lenguaje Unificado de Modelado, lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
W3C	W3C (World Wide Web Consortium). Desarrolla tecnologías ínter-operables

	(especificaciones, guías, software y herramientas) para maximizar el potencial de la web.
WIA	WIA (Windows Image Acquisition) es un modelo de driver e Interfaz de Programación de Aplicaciones (API) para los sistemas operativos más modernos de Microsoft Windows que permite a las aplicaciones de gráficos comunicarse con dispositivos de imagen tales como escaneres, cámaras digitales y equipos de video digital.
XML	XML (Extensible Markup Language o Lenguaje de Marcas Extensible) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).