

**Universidad de las Ciencias Informáticas**



**Facultad 2**

**Módulo Procesamiento de fotografía del  
XABAL Arkheia 2.1 para la Oficina de Asuntos  
Históricos del Consejo de Estado.**

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

**Autor:**

**Dariel Batista Donates**

**Tutores:**

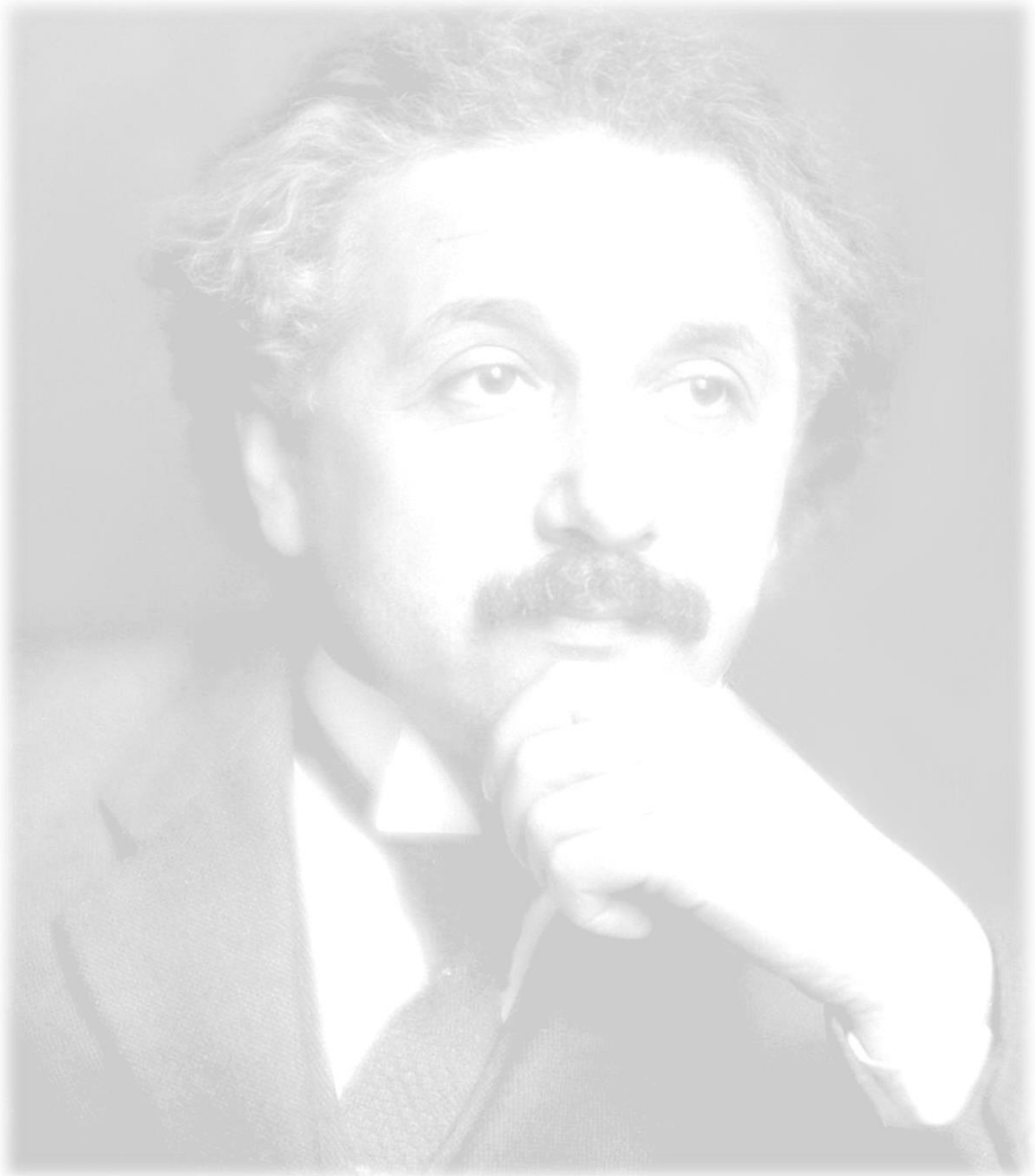
**Ing. Adys Aragón Rodríguez**

**Ing. Javier de León Barral**

**Ing. Elva del Carmen Proenza Orama**

La Habana, junio 2015

"Año 57 de la Revolución"



*'Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad'.*

*Albert Einstein*

## Declaración de autoría

---

### Declaración de autoría

Declaro ser el autor legítimo del trabajo de diploma titulado: “Módulo Procesamiento de fotografía del XABAL Arkheia 2.1 para la Oficina de Asuntos Históricos del Consejo de Estado”, y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste, firmo la presente a los \_\_días del mes de\_\_\_\_\_ del año 2015.

---

Firma del Autor

Dariel Batista Donates

---

Firma del Tutor

Ing. Adys Aragón Rodríguez

---

Firma del Tutor

Ing. Javier de León Barral

---

Firma del Tutor

Ing. Elva del Carmen Proenza Orama

### Agradecimientos

*A mis abuelos por guiar mis primeros pasos y ser ejemplos de respeto y sacrificio.*

*A mis padres por brindarme la mejor educación del mundo y hacer de mí todo lo que soy hoy.*

*A mis tutoras Adys y Elva por brindarme su ayuda en todo momento y todo el tiempo dedicado a mi documento y a responder mis dudas.*

*A Javier y a Leandro por ayudarme cada vez que me enredaba con algo de Grails.*

*A los buenos profesores de los que tuve el privilegio de recibir clases.*

*A mis colegas de la universidad.*

*A todos en general, muchas gracias.*

### Dedicatoria

*A mi familia en general por el apoyo incondicional que me han brindado y en especial:*

*A mis abuelos por todas las enseñanzas que me han dado y por siempre estar ahí cuando los necesité.*

*A mis padres por ser la luz que siempre me ha iluminado el camino, por ayudarme en todo momento y depositar su confianza en mí.*

### Resumen

La archivística es el estudio teórico y práctico de los principios, procedimientos y problemas concernientes al almacenamiento de documentos. Con el fin de salvaguardar el patrimonio documental de una nación y ponerlo al servicio de la comunidad, se ha centrado en la búsqueda de procedimientos para la conservación activa de los documentos.

Cual función exponencial, el establecimiento del orden temporal del siglo XXI por parte de la tecnología informática es estrictamente creciente y los Sistemas de Gestión de Documentos de Archivo (SGDA) representan una eficaz alternativa para preservar y difundir el patrimonio documental de cualquier institución. La Oficina de Asuntos Históricos del Consejo de Estado (OAHCE) posee bajo su jurisdicción información con un gran valor histórico. Para la gestión de la documentación atesorada por esta institución, se desarrolla el SGDA XABAL Arkheia, el sistema actualmente no cuenta con la posibilidad de documentar toda la información necesaria para satisfacer la diversidad de documentos que necesitan ser salvaguardados, dígase fotografías y materiales audiovisuales, además, los documentos son digitalizados a través de herramientas que no forman parte del sistema.

En la presente tesis se realiza una propuesta de solución a dicha problemática con la implementación del módulo Procesamiento de fotografía. Comenzando por el análisis de los principales elementos relacionados con la temática, son valoradas las soluciones similares existentes, son generados los principales artefactos que describen la propuesta y se realizan un conjunto de pruebas para validar su implementación.

**Palabras Clave:** digitalización, descripción, fotografía, procesamiento de fotografía.

## Índice general

<b>INTRODUCCIÓN</b> .....	11
<b>Capítulo 1: Fundamentación teórica</b> .....	15
<b>1.1. Marco conceptual</b> .....	15
<b>1.2. Análisis y selección de un estándar para la adquisición de imágenes desde escáner</b> .....	21
1.2.1. TWAIN .....	21
1.2.2. SANE .....	22
1.2.3. ISIS .....	23
1.2.4. WIA .....	24
1.2.5. Selección de un estándar para la adquisición de imágenes .....	26
<b>1.3. Soluciones similares a nivel internacional</b> .....	26
1.3.1. Addoc.....	26
1.3.2. Kernel Doc .....	27
1.3.3. Adapting.....	27
<b>1.4. Soluciones similares a nivel nacional</b> .....	27
1.4.1. Papiro .....	27
1.4.2. ArchiVenHis .....	28
1.4.3. Análisis y valoración de las soluciones similares.....	28
<b>1.5. Metodología, lenguajes, herramientas y tecnologías</b> .....	29
1.5.1. Metodología .....	29
1.5.2. Lenguajes .....	29
1.5.3. Herramientas .....	32
1.5.4. Tecnologías .....	34
<b>Capítulo 2: Características de la solución</b> .....	37
<b>2.1. Modelo de negocio</b> .....	37
<b>2.2. Requisitos</b> .....	38
2.2.1. Requisitos funcionales .....	38
2.2.2. Requisitos no funcionales .....	40
2.2.3. Técnicas de validación de requisitos .....	42
<b>2.3. Análisis y selección de una librería para el trabajo con el lenguaje Java y los protocolos SANE y TWAIN</b> .....	43

<b>2.4. Propuesta de solución</b> .....	44
<b>2.5. Diagramas de Casos de Uso</b> .....	45
2.5.1. Definición de los actores .....	45
2.5.2. Especificación de casos de uso .....	47
<b>Capítulo 3: Diseño de la solución</b> .....	51
<b>3.1. Patrones de diseño</b> .....	51
3.1.1. Patrones GRASP .....	51
3.1.2. Patrones GoF.....	53
<b>3.2. Arquitectura de Grails</b> .....	53
3.2.1. Patrón de arquitectura Modelo Vista Controlador en Grails.....	54
<b>3.3. Arquitectura del sistema</b> .....	56
<b>3.4. Diagrama de clases de diseño</b> .....	57
<b>3.5. Diagrama de paquetes</b> .....	59
<b>3.6. Diagrama de secuencia</b> .....	59
<b>Capítulo 4: Implementación y prueba</b> .....	61
<b>4.1. Diagrama de componentes</b> .....	61
4.1.1. Descripción de los componentes.....	62
<b>4.2. Diagrama de despliegue</b> .....	64
<b>4.3. Pruebas del sistema</b> .....	65
4.3.1. Métodos de prueba .....	65
4.3.2. Tipos de pruebas .....	66
4.3.3. Estrategias de pruebas .....	66
<b>Conclusiones generales</b> .....	70
<b>Recomendaciones</b> .....	71
<b>Referencias</b> .....	72
<b>Bibliografía</b> .....	76
<b>Glosario de términos</b> .....	81



## Índice de figuras

Figura 1: Elementos de TWAIN. (Dynamsoft Corporation, 2015).....	22
Figura 2: Arquitectura de capas de TWAIN. (Dynamsoft Corporation, 2015) .....	22
Figura 3: Arquitectura de SANE. (Debian Project, 2013) .....	23
Figura 4: Arquitectura de WIA. (Microsoft, 2014) .....	25
Figura 6: Diagrama del proceso de negocio “Procesamiento de fotografía” .....	37
Figura 5: Estructura de la propuesta de solución.....	45
Figura 7: Diagrama de Casos de Uso. Procesamiento de fotografía. ....	46
Figura 8: Estructura de los proyectos Grails. ....	54
Figura 9: Arquitectura de Grails. (Brito, 2009) .....	55
Figura 10: Arquitectura del sistema. ....	56
Figura 11: Diagrama de clases de diseño. Gestionar fotografía. ....	58
Figura 12: Diagrama de paquetes. Módulo Procesamiento de fotografía. ....	59
Figura 13: Diagrama de secuencia. Describir fotografía. ....	60
Figura 14: Diagrama de componentes del sistema.....	61
Figura 15: Paquete de componentes: Vistas. ....	62
Figura 16: Paquete de componentes: Controladores.....	62
Figura 17: Paquete de componentes: Dominio.....	63
Figura 18: Paquete de componentes: Servicios. ....	63
Figura 19: Paquete de componentes: Librerías. ....	64
Figura 20: Diagrama de despliegue.....	65
Figura 21: Informe de resultados de las pruebas de unidad. ....	67
Figura 22: Resultados de las pruebas de Caja negra. ....	68
Figura 23: Caso de prueba Ver detalles de descripción de fotografía.....	69

## Índice de tablas

Tabla 1: Datos de los usuarios finales.....	41
Tabla 2: Requisitos técnicos de los servidores .....	41
Tabla 3: Definición de los actores.....	46
Tabla 4: Descripción del CU Describir fotografía. ....	47

## INTRODUCCIÓN

Asociada a cada estado del proceso biológico evolutivo, la información ha modelado el curso del desarrollo fenotípico en la época moderna. Desde la filogénesis hasta los métodos cuantitativos empresariales más complejos, son completamente dependientes de un conjunto de datos estructurados. Históricamente, las principales fuentes de información sobre los anales de las civilizaciones más arcaicas, revelan, que como resultado de la interacción con el campo perceptivo del entorno, diferentes medios han sido utilizados con el objetivo de lograr fuentes perdurables de información y análogamente al proceso de selección natural, lo que comenzó por un esgrafiado en las rocas, es conocido en la actualidad, como archivo de documentos.

Etimológicamente el término archivo proviene del latín *archivum*, que a su vez procede de la palabra griega *arkheia*, que significa, conjunto de documentos públicos y privados, seriados conforme a criterios definidos y debidamente organizados en base a juicios preestablecidos que deben formar unidades documentales lógicas y congruentes (Fuster Ruiz, 1999). Los archivos tienen como objetivo principal gestionar y conservar el patrimonio documental generado por cualquier entidad con valor administrativo, legal, fiscal, científico, económico, político, cultural o histórico.

Con la aparición de las tecnologías de la información y las comunicaciones, nuevas posibilidades en el acceso, almacenamiento y utilización del conocimiento, son acopladas en función de la era digital, convirtiendo gestores tradicionales de dichos recursos, como bibliotecas o hemerotecas, en centros totalmente accesibles desde cualquier parte del mundo. Esta demanda exponencial en la búsqueda de soluciones informáticas para facilitar el manejo de los archivos, ha dado lugar a la creación de los Sistemas de Gestión de Documentos de Archivo (SGDA). Estos sistemas, se basan en la información digitalizada de los documentos para realizar sus operaciones, simplificando los procesos de almacenamiento, transportación, consulta y manipulación de los registros originales.

El Centro de Informatización de la Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas (UCI) posee dentro de sus productos el Sistema de Gestión de Documentos de Archivo (SGDA) XABAL Arkheia. En la actualidad el proyecto Archivo tiene como objetivo personalizar el XABAL Arkheia para la Oficina de Asuntos Históricos del Consejo de Estado (OAHCE). La personalización del sistema resulta necesaria debido a que, el XABAL Arkheia implementa dentro de sus funcionalidades la incorporación y organización documental

utilizando la Norma Internacional General de Descripción Archivística (ISAD (G), por sus siglas en inglés) pero, la información generada durante el procesamiento de los archivos es insuficiente para satisfacer ciertos requisitos de descripción necesarios para la diversidad de documentos de archivo que atesora esta institución, dígase fotografías, materiales audiovisuales, objetos y documentos.

Por otra parte el XABAL Arkheia permite la incorporación de imágenes digitales de los documentos previamente procesados y escaneados, pero este proceso presenta ciertas dificultades debido a que el sistema no posee la funcionalidad de digitalizar documentos desde escáner y el proceso se realiza con una herramienta externa al XABAL Arkheia. Las imágenes digitales son almacenadas en la computadora del digitalizador y luego, de forma manual, son incorporadas al sistema, generando problemas de seguridad.

Teniendo en cuenta las dificultades que presenta el sistema se identificó el siguiente **problema a resolver**. ¿Cómo incorporar la descripción y digitalización de documentos fotográficos al XABAL Arkheia para la OAHCE de forma funcional? Se define como **objeto de estudio**: la gestión de los documentos fotográficos en los SGDA.

Para guiar el trabajo se plantea como **objetivo general**: Desarrollar el módulo Procesamiento de fotografía del XABAL Arkheia para la OAHCE para describir y digitalizar los documentos fotográficos. Este objetivo se enmarca dentro del **campo de acción**: los procesos de descripción y digitalización de documentos fotográficos en los SGDA.

Para satisfacer el objetivo general quedan definidas las siguientes **tareas de investigación**:

- Estudio de los antecedentes de la gestión y digitalización de documentos de archivo en los SGDA con el objetivo de aportar elementos históricos que fundamenten la investigación
- Caracterización de los protocolos existentes para la interacción con los dispositivos de escaneo
- Caracterización de las soluciones nacionales e internacionales enmarcadas en el objeto de estudio para obtener referencias de las funcionalidades a implementar
- Caracterización de las herramientas, tecnologías, lenguajes y metodologías a utilizar para el desarrollo del módulo Procesamiento de fotografía
- Especificación de los requisitos funcionales y no funcionales del módulo Procesamiento

de fotografía, para conocer las características esenciales que debe poseer

- Elaboración del modelo de diseño para el desarrollo del módulo Procesamiento de fotografía
- Implementación de los casos de uso del módulo Procesamiento de fotografía para cumplir con los requisitos definidos
- Verificación del cumplimiento de los requisitos mediante la realización de pruebas de software al sistema

Durante la investigación se utilizan los siguientes **métodos científicos**:

### **Métodos teóricos:**

- **Análítico-sintético:** se utiliza para el análisis del lenguaje y las herramientas a utilizar, basados en sus características y funcionalidades, en el estudio de los protocolos para la comunicación con el escáner, en la caracterización de las soluciones similares y en gran parte de la investigación.
- **Análisis Histórico – Lógico:** análisis evolutivo de la descripción y digitalización de documentos, el desarrollo de los archivos y la terminología relacionada.
- **Modelación:** se utiliza la modelación para representar mediante diagramas la estructura de los procesos que intervienen en el sistema, facilitando su comprensión.

### **Métodos empíricos:**

- **Entrevista:** se realiza la entrevista al arquitecto del proyecto para obtener información acerca del funcionamiento de los procesos de descripción y digitalización de fotografías, así como su interoperabilidad con el sistema en general.
- **Observación:** se utiliza para la obtención de la información de cada uno de los conceptos relacionados con la gestión y los procesos de descripción y digitalización de fotografías.
- **Análisis estático:** se utiliza en la realización de pruebas al sistema. A través del examen de la estructura de la aplicación, se realiza una inspección y recorrida para detectar errores.

El presente trabajo está estructurado de la siguiente forma:

**Capítulo.1 Fundamentación teórica:** en este capítulo son abordados de forma general los aspectos teóricos más importantes relacionados con el objeto de estudio. Son incluidos los conceptos fundamentales de la gestión de archivos y los principales elementos vinculados en el proceso de digitalización y procesamiento de documentos fotográficos. Se exponen brevemente las herramientas, lenguajes y la metodología a utilizar para la implementación del módulo.

**Capítulo.2 Características de la solución:** se detalla la propuesta de solución y son representados los diagramas de procesos del negocio y de casos de uso, así como, la descripción de los actores y los requisitos funcionales y no funcionales del sistema.

**Capítulo.3 Diseño de la solución:** en este capítulo se realiza el modelado de los diagramas de paquetes, secuencia y clases de diseño, se describen los principales elementos de la arquitectura del sistema, de la solución planteada y los patrones utilizados para el diseño.

**Capítulo.4 Implementación y prueba:** contiene las principales características de la implementación donde son modelados los diagramas de componentes y despliegue. Es validada la solución propuesta mediante pruebas de software realizadas al módulo, comprobando su correcto funcionamiento y calidad.

# Capítulo 1: Fundamentación teórica

---

## Capítulo 1: Fundamentación teórica

### Introducción

En este capítulo se hace referencia a los principales aspectos teóricos que fundamentan la investigación. A través del marco teórico conceptual, se establece un esquema temático, lógico y desglosado que representa la base del proceso investigativo. Se realiza un estudio de los sistemas similares existentes a nivel nacional e internacional, con el objetivo de establecer analogías de arquitecturas, estructuras, lenguajes y demás aspectos que pueden ser de utilidad en el desarrollo de la solución propuesta. Son examinadas las tendencias actuales para el desarrollo de software, con el fin de encontrar herramientas, técnicas y métodos para llevar un adecuado control y desarrollo de la aplicación. Se ofrecen las principales características del entorno de trabajo definido por el proyecto, con el propósito de adquirir información esencial para la selección de un estándar para la digitalización y la implementación general del sistema.

### 1.1. Marco conceptual

Para una consecuente estructuración lógica y análisis crítico de la investigación y con el objetivo de dar unidad, coherencia y consistencia a los postulados existentes en el área de la gestión de documentos fotográficos, se realiza un tratamiento conceptual del tema a partir de los elementos acumulados en el transcurso de la investigación bibliográfica. Para la consecución de dicho objetivo se comienza por la unidad más atómica de la investigación, la fotografía.

### Fotografía

La fotografía es un procedimiento fotoquímico mediante el cual se reproduce una imagen real sobre una superficie sensible a la luz. En la gran mayoría de procesos fotográficos se distingue el soporte físico (por ejemplo el papel si es una postal antigua, o el plástico en un negativo contemporáneo), y las sustancias sensibles a la luz que forman la imagen y que se hallan mezcladas con un aglutinante que las fija al soporte (Foix, 2003). Para que la imagen perdure en el tiempo más allá del instante de exposición a la luz, se somete el negativo o positivo obtenido en la cámara a un proceso químico de revelado y fijado que le da estabilidad y permanencia (Langford, 2004). Cuando estos archivos responden a una cronología y relevancia histórica son considerados documentos fotográficos.

# Capítulo 1: Fundamentación teórica

---

## Documento fotográfico

Los fondos documentales son aquellos que se han formado a lo largo del tiempo por la actividad propia de una persona o entidad. En el caso de los fondos o documentos fotográficos, se refiere a los archivos de un fotógrafo en particular, del fondo de una entidad que recoge las imágenes de los distintos actos y actividades que narran su historia, archivos de prensa de distintas publicaciones, hechos históricos relevantes, etc. El conjunto documental natural al que pertenece una imagen, la dota de la necesaria perspectiva y en muchos casos permite establecer datos de autoría, datación (tópica y crónica), o identificación de lugares, objetos o personas reproducidas en la fotografía. Estos datos, que no siempre podrán ser estudiados en una primera aproximación al fondo o colección, si se mantiene su integridad y procedencia, podrán ser recuperados e interpretados en el momento en que se disponga de recursos para estudiarlos y procesarlos (Foix, 2003). Como los restantes documentos de archivo, las fotografías son susceptibles a un conjunto de condiciones que aceleran su deterioro, evidenciando la necesidad de conservarlas en formato digital. A continuación son presentados los procesos de deterioro que laceran con más frecuencia los documentos fotográficos.

## Procesos de deterioro más usuales

Para comprender cómo y por qué se deteriora una fotografía se debe tener presente su composición estructural: soporte y emulsión (formada por el aglutinante y las sustancias formadoras de la imagen). A partir de sus componentes, a continuación se muestran los procesos de deterioro que afectan habitualmente a las fotografías.

El papel es el soporte más utilizado para las copias positivas. Es frágil ante roturas, dobleces y curvaturas. Si es de mala calidad puede descomponerse, y es además un alimento natural para insectos y microorganismos, especialmente si la humedad y temperatura ambiental son altas. En los primeros papeles contemporáneos multicapa para positivos puede observarse una delaminación, aunque actualmente no tengan este riesgo por su mayor calidad. Los papeles baritados, también propios de la fotografía contemporánea, son muy estables y sólo presentan problemas si el procesado técnico de la imagen no ha sido correcto.

Los daños del soporte por rotura, doblez o abrasión, aceleran otros procesos de deterioro que afectan a la emulsión, ya sea porque se desprende total o parcialmente del soporte volviéndose más frágil, o porque queda más vulnerable frente a la humedad ambiental y al aire. En cuanto a



## Capítulo 1: Fundamentación teórica

---

los deterioros del aglutinante, los aglutinantes más utilizados han sido el colodión, la albúmina y la gelatina. Su mal estado afecta directamente a la imagen, que se halla suspendida en él.

La gelatina es el más habitual, sobre todo en los procesos contemporáneos. En los procesos antiguos hay un amplio uso de la albúmina para los positivos sobre papel. Son muy sensibles a la humedad ambiental. Una humedad relativa alta provoca adhesiones con las superficies adyacentes por reblandecimiento del aglutinante. Si el ambiente es muy seco, tiende a encogerse y perder adherencia, pudiendo darse un desprendimiento de la emulsión respecto del soporte. Tanto la gelatina como la albúmina son muy vulnerables a las infecciones por microorganismos, especialmente en condiciones ambientales de humedad y temperatura altas (Langford, 2004). Un elemento fundamental para la digitalización de los archivos fotográficos, es el almacenamiento de la información que permita su clasificación. Para obtener toda la información asociada a un documento fotográfico es necesaria su descripción.

### **Descripción fotográfica**

En la elaboración de un catálogo se establecen los puntos de acceso al documento a partir de su descripción física y el análisis de su contenido; obteniendo lo que se conoce por ficha descriptiva, que no es más que un documento secundario que sustituye al documento primario o fotografía original. Aunque los catálogos incluyan una reproducción de la fotografía original, el proceso de catalogación es imprescindible para establecer los puntos de acceso que permitan recuperar una determinada imagen, ya que las búsquedas se expresan en lenguaje textual y no gráfico (Olson, 2008). Según Díez muchas imágenes, sin la correcta interpretación e identificación, quedan carentes de sentido o pierden su principal valor, al desconocer los hechos, objetos o personas que reproducen. Por otra parte, cuando se trabaja con originales fotográficos, por su proceso de creación, por su valor histórico, su rareza o su antigüedad, son objetos valiosos por sí solos, y cobra importancia el documento físico. Por ello es importante una buena catalogación descriptiva, donde se identifique el soporte y proceso fotográfico utilizado para la obtención de la imagen. (Díez Carrera, 1998)

La mayoría de fotografías, además de su contenido icónico poseen también un contenido simbólico que es igual o más importante. Especialmente en la fotografía documental, el fotoperiodismo o cualquier reproducción de actos, actividades, o las consecuencias que de ellos se derivan. El contenido simbólico debe deducirse o interpretarse con la ayuda de los datos disponibles del documento (Foix, 2003). Para la descripción y catalogación estandarizada de los documentos fotográficos son utilizadas las normas de descripción.

# Capítulo 1: Fundamentación teórica

---

## Normas de descripción archivística

Las normas para la descripción de documentos de archivos son un grupo de reglas establecidas que sirven como patrón para realizar las descripciones de los documentos. El motivo fundamental para el apoyo y adaptación de normas de descripción bibliográfica, según los propios autores, es el hecho de que pueden de esta forma aprovechar las redes de intercambio de información bibliográfica existente.

Las normas para la descripción del material archivístico aseguran una enorme mejora en cuanto a la creación de descripciones coherentes, apropiadas e inteligibles por sí mismas, facilitan la recuperación y el intercambio de información sobre la documentación de archivo, hacen posible la integración de descripciones de diferentes depósitos en un sistema unificado de información y traen beneficios económicos, organizativos y de mejora del servicio. Además, unas normas descriptivas claras son una premisa obligatoria para cualquier proyecto de automatización serio. (Peis, et al., 2004)

## ISAD (G)

La norma ISAD (G) (Norma Internacional General de Descripción Archivística) es una normativa internacional para la descripción de documentos. Está basada en los modelos estadounidense, británico y canadiense. El principal aporte viene dado por la norma MAD (*Manual of Archival Description*), que brindó una estandarización de la estructura de datos, es decir, determina cuáles son los datos descriptivos (veintiséis elementos) que puede contener una descripción, y los estructura jerárquicamente en las siguientes áreas (Carnicer, et al., 2000):

- Área de mención de identidad: contiene información mínima descriptiva.
- Área de contexto: empleada para recoger los datos provenientes del análisis contextual.
- Área de contenido y estructura: información relativa al objeto y la organización de la unidad de descripción.
- Área de condiciones de acceso y uso: información relativa a la accesibilidad de la unidad de descripción.
- Área de documentación asociada: información relativa a aquellos documentos que tienen una relación significativa con la unidad de descripción.
- Área de notas: información especial y aquella no incluida en ninguna de las demás áreas.

## Capítulo 1: Fundamentación teórica

---

- Área de control de la descripción: información relativa al cómo, cuándo y quién ha elaborado la descripción archivística.

De los veintiséis elementos necesarios para la descripción (datos descriptivos), se consideran de carácter obligatorio seis elementos para la descripción archivística internacional, estos son (Carnicer, et al., 2000):

- Código de Referencia
- Título
- Productor(es)
- Fecha(s)
- Extensión de la Unidad de Descripción
- Nivel de Descripción

La norma ISAD (G) propone determinadas reglas de descripción multinivel. La descripción multinivel consiste en llevar a cabo la descripción de un fondo y todas sus partes componentes (secciones, series, expedientes, documentos y todas las categorías intermedias), utilizando siempre los elementos apropiados para cada unidad de descripción y relacionando las descripciones resultantes de forma jerárquica. (Carnicer, et al., 2000)

A cada nivel de clasificación (agrupación) documental le corresponde un nivel de descripción. Las descripciones resultantes, relacionadas jerárquicamente, suponen la representación de la estructura de clasificación mediante la descripción. La norma ISAD (G) establece un grupo de cinco niveles de descripción que van de lo general a lo específico: nivel de fondo, nivel de sub-fondo, nivel de serie, nivel de expediente y nivel de unidad documental. No obstante, la norma no obliga el nivel de descripción al que se debe llegar. Para una correcta aplicación de la descripción multinivel, la norma ISAD (G) propone cuatro reglas generales: regla de descripción de lo general a lo particular, regla de información pertinente para el nivel de descripción, regla de vinculación de las descripciones y regla de no repetición de la información (Peis, et al., 2004). Para la incorporación de la imagen digitalizada a su descripción archivística es inevitable contar con un proceso de digitalización.

### **Digitalizar**

Digitalizar es la "acción de convertir en digital información analógica. En otras palabras, es convertir cualquier señal de entrada continua en una serie de valores numéricos". Una imagen

# Capítulo 1: Fundamentación teórica

---

en papel puede digitalizarse para que pueda ser procesada en una computadora (u otro dispositivo digital similar). La información digital es la única información que puede procesar una computadora, generalmente en el sistema binario, es decir unos (1) y ceros (0). Digitalizar un documento va a depender del tipo de información. Si es una imagen fotográfica en papel, por ejemplo, puede digitalizarse a través de un escáner. (Map, 2012)

## Imagen digital

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica. Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de bits) o de un gráfico vectorial. El mapa de bits es el formato más utilizado en "la informática". (Bennet, et al., 2010)

Las imágenes digitales se pueden obtener de varias formas:

- Por medio de dispositivos de conversión analógica-digital, como son los escáneres y las cámaras digitales
- Directamente mediante programas informáticos

La imagen digital, obtenida por escaneo, es dividida en una matriz de puntos a modo de cuadrícula, tantos como fotodiodos tenga el sensor. Cada uno de estos puntos recibe el nombre de píxel, que toma el valor binario 1 o 0 dependiendo de la luminosidad y el tono lumínico leído por el escáner. Esta cadena de código binario es enviada al ordenador, donde la imagen queda almacenada en formato digital. A cada píxel se le asigna un valor tonal (negro, blanco, gris, color) que está representado por un código binario (ceros y unos). Estos dígitos binarios (bits) para cada píxel son almacenados en una secuencia y con frecuencia reducidos a una representación matemática (compresión). (Carnicer Arribas, et al., 2011) Para la comunicación con el escáner y la obtención de la imagen digital son necesarios los protocolos o estándares de comunicación que transforman la información binaria a un lenguaje de programación de bajo nivel.

## Protocolo o estándar

Protocolo, en las ciencias de la computación, es un conjunto de reglas o procedimientos para la transmisión de datos entre dispositivos electrónicos, tales como ordenadores, impresoras o escáneres. A fin de que los equipos puedan intercambiar información, debe haber un acuerdo preexistente en cuanto a cómo la información se estructura y cómo cada lado enviará y recibirá.

# Capítulo 1: Fundamentación teórica

---

Sin un protocolo, un equipo de transmisión, por ejemplo, podría estar enviando sus datos en paquetes de 8 bits, mientras que el equipo receptor podría esperar los datos en paquetes de 16 bits. Los protocolos son establecidos por organizaciones internacionales o empresas de la industria. (Encyclopedia Britannica, 2015)

## **1.2. Análisis y selección de un estándar para la adquisición de imágenes desde escáner**

### **1.2.1. TWAIN**

TWAIN es un protocolo de software estándar y una interfaz de programación de aplicaciones (API) que regula la comunicación entre las aplicaciones de software y los dispositivos de adquisición de imágenes, tales como escáneres y cámaras digitales. Fue creado por un pequeño grupo de empresas software y hardware en respuesta a la necesidad de una propuesta de especificación para la industria de la imagen. El objetivo del grupo de trabajo era proporcionar una solución abierta y multiplataforma para interconectar los dispositivos de entrada de trama con el software de aplicación. El grupo de trabajo original estaba compuesto por representantes de cinco empresas: Aldus, Caere, Eastman Kodak, Hewlett-Packard, y Logitech. Otras tres empresas, Adobe, Howtek y arquitectos de software, también contribuyeron de manera significativa.

TWAIN proporciona una metodología simple para la conexión universal entre las aplicaciones de interacción y los dispositivos en escucha. El modelo de iteración de las aplicaciones con la fuente de datos de entrada puede ser descrito a través de un protocolo de cuatro capas (figura 2): la capa de aplicación, la capa de protocolo, la capa de adquisición y la capa de dispositivo (TWAIN Working Group , 2012). Estas capas a su vez están compuestas por un conjunto de elementos, como se aprecia en la figura 1, que describen dónde se establece la interacción directa con los dispositivos de adquisición de imágenes y cómo se establece la comunicación con la aplicación del usuario.

# Capítulo 1: Fundamentación teórica

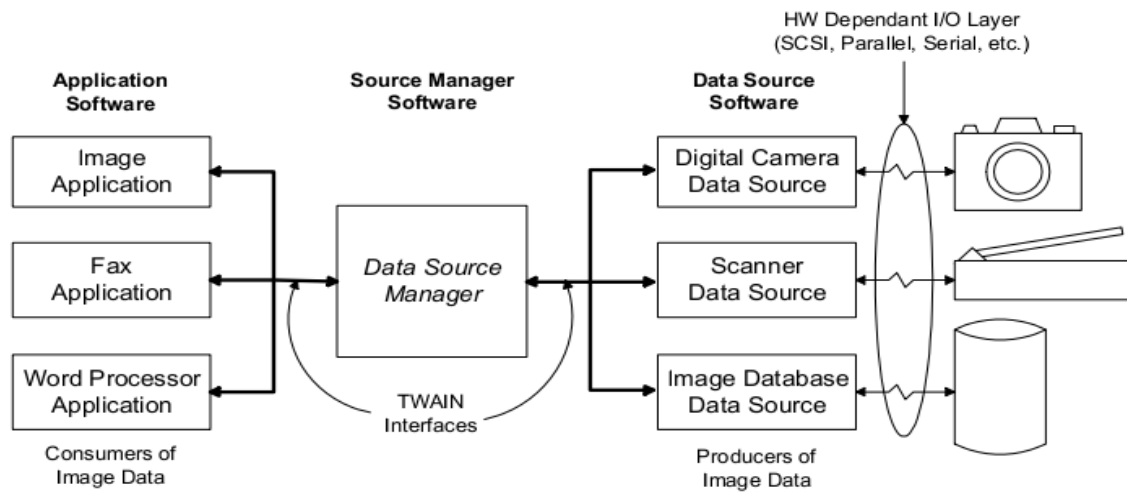


Figura 1: Elementos de TWAIN. (Dynamsoft Corporation, 2015)

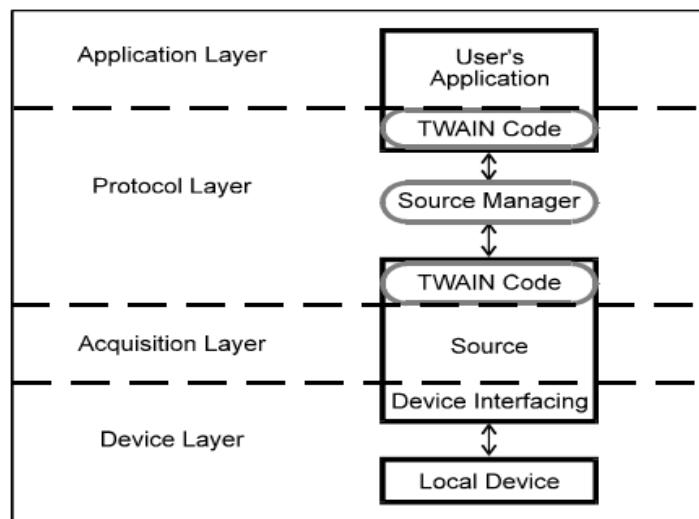


Figura 2: Arquitectura de capas de TWAIN. (Dynamsoft Corporation, 2015)

## 1.2.2. SANE

SANE es el acrónimo de "Scanner Access Now Easy" y es una interfaz de programación de aplicaciones (API, por sus siglas en inglés) que proporciona un acceso estandarizado a cualquier dispositivo de escaneo de imágenes (escáner plano, escáner de mano, etc.). La API de SANE es de dominio público y su discusión y desarrollo está abierto a todo el mundo. El código fuente actual está escrito para UNIX (incluyendo GNU / Linux) y está disponible bajo la Licencia Pública General de GNU (sin embargo, la API de SANE está disponible también para aplicaciones propietarias).

## Capítulo 1: Fundamentación teórica

SANE es una interfaz de escáner universal. El valor de una interfaz tan universal es que permite utilizar un solo controlador por dispositivo de adquisición de imagen en lugar de un controlador para cada dispositivo y aplicación. Así que, si tiene tres aplicaciones y cuatro dispositivos, tradicionalmente hubiese sido necesario escribir 12 programas diferentes. Con SANE, este número se reduce a siete: las tres aplicaciones más los cuatro controladores (ver figura 3). Por supuesto, el ahorro se incrementa a medida que se agregan más controladores y aplicaciones.

SANE no solo reduce el tiempo de desarrollo y la duplicación de código, sino que también eleva el nivel en el que las aplicaciones pueden trabajar. Por lo tanto, habilita aplicaciones que antes eran desconocidas en el mundo UNIX. Mientras SANE está dirigido principalmente a un entorno UNIX, la norma ha sido cuidadosamente diseñada para hacer posible la implementación de la API en prácticamente cualquier hardware o sistema operativo. (Debian Project, 2013)

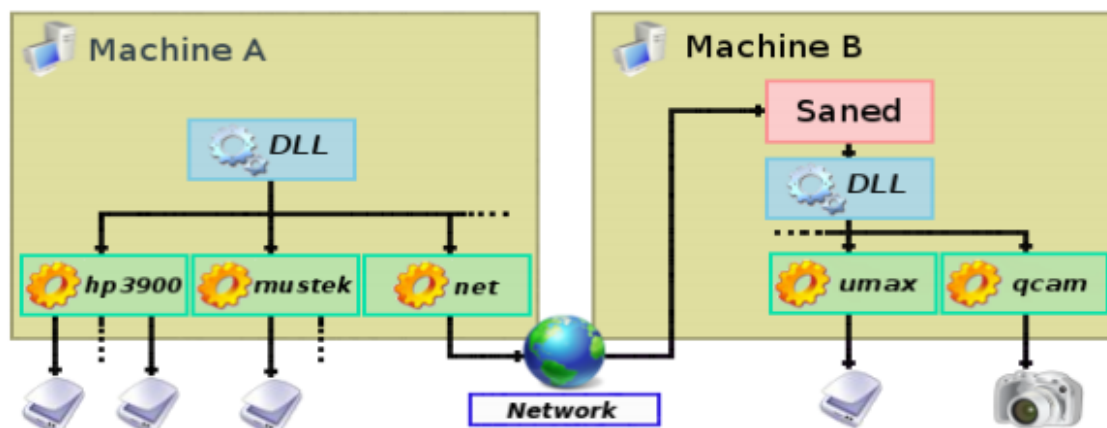


Figura 3: Arquitectura de SANE. (Debian Project, 2013)

### 1.2.3. ISIS

EMC Captiva ISIS Driver (*Image and Scanner Interface Specification*) es la interfaz de controlador de escáner estándar de la industria a nivel empresarial que une escáneres con aplicaciones de software. Más de 300 escáneres diferentes comunican actualmente a las aplicaciones a través de la interfaz del controlador ISIS. ISIS permite a los escáneres funcionar a las velocidades indicadas o superiores y es la única solución de imagen que permite a los usuarios sacar el máximo provecho de la potencia inherente a sus escáneres y otros equipos de imágenes.

# Capítulo 1: Fundamentación teórica

---

La arquitectura ISIS se basa en los componentes de los módulos de software que realizan funciones de imagen específicas (por ejemplo, la adquisición de imágenes, conversión de archivos, extracción de datos y archivo de comandos de lectura / escritura). Esta arquitectura permite que los nuevos módulos se añadan sin realizar cambios en todo el sistema.

Características generales (EMC Corporation, 2009):

- ISIS es modular: permite a las aplicaciones controlar un escáner directamente, o utilizar las rutinas proporcionadas para manejar la mayoría de las situaciones de forma automática.
- ISIS es flexible: usando una interfaz basada en mensajes con etiquetas, puede crecer sin perder compatibilidad. Esto significa que las características, operaciones y formatos no existentes en la actualidad pueden añadirse sin esperar a una nueva versión de la especificación.
- ISIS es una especificación completa: trata todos los temas que una aplicación que usa un escáner debe tratar. Esto incluye tareas como seleccionar, instalar y configurar un escáner nuevo, establecer parámetros específicos del escáner, escanear, leer y escribir archivos, escalado de imágenes, rotación, visualización e impresión. Los controladores ISIS también han sido escritos para preprocesar datos realizando operaciones tales como conversión a escala de grises de forma dinámica.
- ISIS sobresale por la velocidad de procesado. Lo hace enlazando controladores entre sí en una arquitectura en *pipeline* para que los datos fluyan desde el controlador del escáner al controlador de compresión, al controlador de empaquetado, al archivo, al visualizador o impresora en una trama continua, normalmente sin necesidad de procesar más que una pequeña porción de la imagen completa. Debido a que los controladores ISIS se disponen en una *pipeline* cuando son utilizados, cada controlador se especializa en realizar una sola función. Los controladores normalmente son pequeños y modulares, lo que significa que ISIS permite nuevas funcionalidades en una aplicación con modificaciones muy pequeñas.

## 1.2.4. WIA

Adquisición de imágenes de Windows (WIA, por sus siglas en inglés) es la tecnología de imágenes más reciente introducida en *Microsoft Windows Millennium Edition* (Me) que se incluye en las versiones posteriores del sistema operativo de Microsoft Windows. WIA



## Capítulo 1: Fundamentación teórica

proporciona un gran número de funciones de compatibilidad del sistema operativo para permitir que los dispositivos de imágenes, como los escáneres y las cámaras digitales, se comuniquen con aplicaciones de imágenes (por ejemplo, Microsoft Picture It 2000 y Adobe Photoshop).

Antes de WIA, el sistema operativo admitía muy pocos dispositivos de imágenes. El fabricante proporcionaba la mayoría de los componentes de los dispositivos de imágenes, que utilizaban el estándar del sector TWAIN.

Con WIA se proporcionan los componentes del sistema operativo y los fabricantes solo tienen que escribir un pequeño controlador WIA para permitir que el dispositivo funcione (figura 4). Esto es análogo al estándar al que se ha llegado en la impresión, que permite que los fabricantes proporcionen los minicontroladores y que el sistema operativo proporcione los demás componentes de la impresión. (Microsoft, 2014)

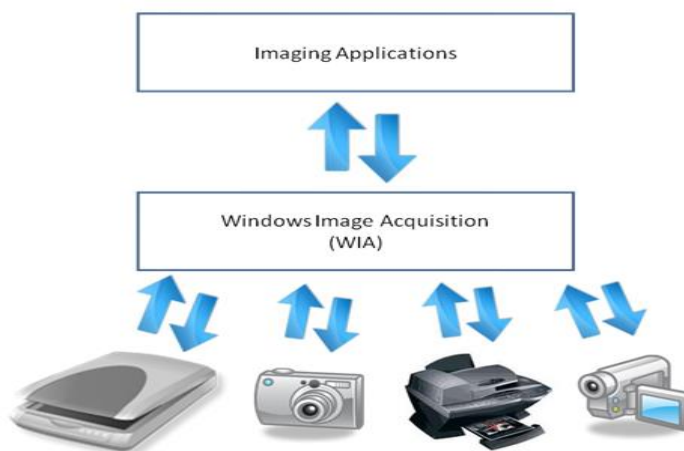


Figura 4: Arquitectura de WIA. (Microsoft, 2014)

Características generales (Microsoft, 2014):

- Se ejecuta en un entorno sólido y estable
- Reduce al mínimo los problemas de interoperabilidad
- Enumera los dispositivos de adquisición de imágenes disponibles
- Crea conexiones a múltiples dispositivos a la vez
- Consulta de propiedades de los dispositivos de una manera estándar y ampliable
- Adquiere los datos del dispositivo utilizando mecanismos estándares y de alto rendimiento de transferencia

## Capítulo 1: Fundamentación teórica

---

- Mantiene las características de la imagen durante toda transferencia de datos
- Notifica de una amplia variedad de eventos de dispositivo

### 1.2.5. Selección de un estándar para la adquisición de imágenes

A partir del análisis realizado y en correspondencia con lo establecido por el proyecto para que el sistema sea multiplataforma, fueron seleccionados los estándares, TWAIN para la implementación en la familia de sistemas operativos Microsoft Windows y SANE para la implementación en las distribuciones Linux, debido a que, son de dominio público, proporcionan acceso estandarizado a cualquier dispositivo de escaneo, son ampliamente conocidos y brindan la posibilidad de interactuar con lenguaje Java. Durante la selección, el protocolo ISIS fue descartado debido a que se debe pagar una licencia por su utilización y el protocolo WIA porque funciona solamente en la plataforma Windows y posee funcionalidades de escaneo más restringidas que los estándares seleccionados.

### Estudio y análisis de las soluciones similares

En proporción a la necesidad exponencial de soluciones eficaces a los problemas de la gestión documental, iniciativas son desarrolladas a nivel nacional e internacional, permitiendo la gestión archivística electrónica altamente eficiente. Bajo la custodia de instituciones especializadas en el manejo de la información, múltiples sistemas de gestión de archivo brindan hoy poderosos servicios de gestión, descripción y digitalización.

A continuación serán analizados los aspectos fundamentales de cada uno de los sistemas que implementan dentro de sus servicios la gestión de archivos fotográficos, con el objetivo de valorar su solución y establecer una comparación con la aplicación en desarrollo para obtener características comunes.

## 1.3. Soluciones similares a nivel internacional

### 1.3.1. Addoc

Addoc es una empresa argentina especializada en la prestación de servicios integrales de administración documental destinados a empresas y organizaciones públicas y privadas. Ofrece servicios de gestión, custodia, protección y digitalización de documentos basados en altos estándares de seguridad a través de un equipo de trabajo conformado por especialistas con experiencia en el mercado nacional e internacional en el análisis, diseño e implementación de

## Capítulo 1: Fundamentación teórica

---

proyectos de gestión documental. Cuenta con servicios de digitalización de documentos donde aplican técnicas como OMR, OCR, ICR e indexación manual de información. (Addoc, 2008)

### 1.3.2. Kernel Doc

Kernel Doc es una empresa española fundada en el 2002. Sus objetivos están centrados en la prestación de servicios en el ámbito de la organización, transformación y gestión de la documentación, independiente de marcas y fabricantes. Sus productos y soluciones van desde la consultoría documental, la digitalización de fondos documentales, procesos de apoyo a la gestión interna, externalización de procesos y recursos así como la custodia y la destrucción de documentos. (Kernel Doc s.l., 2007)

### 1.3.3. Adapting

Adapting es una empresa española creada en 1999. Ofrece soluciones a los problemas de gestión documental y digitalización de las empresas medianas, dentro de sus servicios se encuentra el abox que es una solución de gestión documental profesional que captura, almacena, procesa y distribuye información empresarial no estructurada (documentos). Abox gestiona correspondencia, organiza la documentación, optimiza procesos internos y gestiona virtualmente el archivo físico. Adapting cuenta además con una red internacional de aliados para poder ofrecer proyectos de consultoría, desarrollo o integración “llave en mano”, con las máximas garantías de excelencia y funcionalidad para el cliente. (Adapting sl, 2013)

## 1.4. Soluciones similares a nivel nacional

### 1.4.1. Papiro

Papiro es un sistema que permite la conservación, digitalización, gestión y socialización de información documental de los archivos en Cuba. Este software fue implementado por personal de la empresa DESOFT de la provincia Granma. Es un producto informático de uso libre que emplea herramientas igualmente libres y permite conservar documentación de valor histórico al evitar su manipulación directa. Está concebido para correr bajo la plataforma Microsoft Windows en las versiones 2000 y XP.

El sistema es utilizado en el Archivo Histórico de la Ciudad de Manzanillo, y si bien fue diseñado para su empleo en repositorios históricos, también puede aplicarse en archivos de gestión,

## Capítulo 1: Fundamentación teórica

---

resultando al mismo tiempo un intento pionero en Cuba, al vincular la gestión de bases de datos con las imágenes de documentos originales digitalizados. (Orozco González, et al., 2006)

### 1.4.2. ArchiVenHis

ArchiVenHis es un sistema creado en la Universidad de las Ciencias Informáticas (UCI) en el año 2007, es un sistema web de código abierto, guiado por normas y estándares internacionales como la ISAD (G) y EAD para la gestión archivística. Permite importar representaciones digitales y luego asociarlas a descripciones documentales. Este sistema fue creado para el Archivo General de la Nación de la República Bolivariana de Venezuela.

El sistema fue desarrollado para su uso en una intranet y está compuesto por varios módulos que permiten realizar las actividades comunes en los archivos históricos tales como, descripción de documentos, solicitud de servicios de reproducción y certificación de documentos, solicitud de servicios de transcripción paleográfica, creación de la estructura lógica de organización de la documentación mediante los niveles de descripción, creación de la estructura física de organización de la documentación, solicitud de documentos físicos; siendo entre todas ellas sus funcionalidades esenciales permitir a los historiadores realizar la descripción e incorporar las imágenes digitalizadas de los documentos existentes en los archivos históricos según la norma ISAD(G); y facilitar el acceso a la información mediante búsquedas sobre las descripciones realizadas. (Almaguer, 2008)

### 1.4.3. Análisis y valoración de las soluciones similares

Basados en que el sistema en desarrollo es una personalización de un producto existente, las soluciones similares no fueron consideradas para su utilización, pero a partir del estudio realizado, se obtuvieron un conjunto de características análogas que sirven como referente para la arquitectura, el diseño e implementación del sistema. A partir del análisis de las características semejantes, se obtuvo que los sistemas internacionales estudiados realizan la digitalización directa desde el escáner pero utilizan otra tecnología, A partir de esta funcionalidad, se tomaron referencias con respecto a los elementos de edición y a los parámetros a describir correspondientes a la fotografía, y de los sistemas nacionales ArchiVenHis y Papiro, a pesar de que no cuentan con la funcionalidad de digitalización, se obtuvieron referencias con relación a la descripción archivística de imágenes a través de la norma ISAD(G).

# Capítulo 1: Fundamentación teórica

---

## 1.5. Metodología, lenguajes, herramientas y tecnologías

La metodología, herramientas, lenguajes de desarrollo y tecnologías fueron definidas y establecidas como políticas del proyecto Archivo mediante un estudio realizado por el equipo de arquitectura (Sixto, 2014) con el objetivo de disponer de un entorno de trabajo para el proceso de elaboración del sistema XABAL Arkheia. A continuación se describen con un enfoque conceptual sus principales características relacionadas con la solución a desarrollar.

### 1.5.1. Metodología

#### Relational Unified Process (RUP)

El Proceso Unificado Racional (*Rational Unified Process* en inglés, habitualmente resumido como RUP) es una metodología de desarrollo de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Tiene como objetivo asegurar la calidad en la producción de software, logrando satisfacer en tiempo y presupuesto la necesidad del usuario final (Jacobson, et al., 2000). Es una de las metodologías más utilizadas en la universidad para el desarrollo de software. Las características principales son: guiado por casos de uso, centrado en la arquitectura e iterativo e incremental.

RUP divide el proceso de desarrollo en 4 fases, en las cuales se realizan varias iteraciones de las actividades. Las fases son (Jacobson, et al., 2000):

- **Concepción o Inicio:** se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- **Construcción:** se obtiene un producto documentado y listo para su utilización incluyendo un manual de usuario para facilitar una mejor comprensión de lo desarrollado. Se obtienen una o varias liberaciones del producto que han pasado las pruebas.
- **Transición:** el producto liberado ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

### 1.5.2. Lenguajes

Según lo planteado por la IEEE, un lenguaje puede ser entendido como un recurso que hace posible la comunicación. Esta sección se centra en describir las características de los lenguajes

# Capítulo 1: Fundamentación teórica

---

de programación y de modelado utilizados en la presente investigación (IEEE, 1999). Un lenguaje de programación es aquella estructura que, con una cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora. (Definición. De, 2012)

## Java

Java es un lenguaje de alto nivel basado en el paradigma Orientado a Objetos cumpliendo con características como encapsulación, herencia y polimorfismo. Es un lenguaje que junto a la Máquina Virtual de Java (JVM, por sus siglas en inglés), permite que los proyectos sean ejecutados en diferentes entornos de hardware o software. Sus usos se extienden desde dispositivos móviles, aplicaciones de servidor y clientes, entre otras. Permite la búsqueda y tratamiento de errores en tiempo de ejecución por lo que se considera un lenguaje robusto (Oracle Corporation, 2013). Es utilizado en el módulo en la implementación de la digitalización para interactuar con el “*Data Source Manager*” de TWAIN y con el *frontend* de SANE, para editar la imagen escaneada y enviarla al sistema.

## Groovy 2.0.7

Groovy es un lenguaje de scripting basado en Java Groovy. Groovy tiene mucho en común con lenguajes de scripting populares como Perl, Python y Ruby, pero está escrito en una sintaxis similar a Java. A diferencia de otros lenguajes, Groovy tiene la posibilidad de ejecutarse en el entorno de la Máquina Virtual de Java (JVM). Aplicaciones escritas en Groovy pueden hacer pleno uso de las *Java Application Programmer Interfaces* (APIs). Esto significa que Groovy puede integrarse perfectamente con las aplicaciones escritas en Java, evitando las complejidades del lenguaje Java puro. Sus construcciones más simples y orígenes modernos lo hacen ideal como primer lenguaje para la introducción de principios como la programación orientada a objetos (Barclay, et al., 2006). Es utilizado en Grails para implementar los controladores, las clases de dominio y los servicios.

## JavaScript 1.5

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos (Mozilla, 2015). JavaScript contiene una librería estándar de objetos, tal como *Array*, *Date*, y *Math*, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. El núcleo de JavaScript puede

# Capítulo 1: Fundamentación teórica

---

extenderse para varios propósitos, complementándolo con objetos adicionales, por ejemplo (Mozilla, 2015):

- Client-side JavaScript extiende el núcleo del lenguaje proporcionando objetos para controlar un navegador y su modelo de objetos (o DOM, por las iniciales de *Document Object Model*). Por ejemplo, las extensiones del lado del cliente permiten que una aplicación coloque elementos en un formulario HTML y responda a eventos del usuario, tales como clics del ratón, ingreso de datos al formulario y navegación de páginas.
- Server-side JavaScript extiende el núcleo del lenguaje proporcionando objetos relevantes a la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comuniquen con una base de datos, proporcionar continuidad de la información de una invocación de la aplicación a otra, o efectuar manipulación de archivos en un servidor.

## Lenguaje de modelado

El Leguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje de modelado que se centra en la representación gráfica de un sistema. Tiene una notación gráfica expresiva que permite representar en mayor o menor medida todas las fases de un proyecto de software: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, entre otros, hasta la implementación y configuración con los diagramas de despliegue (Larman, 2003). UML en su versión 2.0 sirve para el modelado completo de sistemas complejos, tanto en el diseño del software como para la arquitectura de hardware donde se ejecuten.

## Modelado de procesos del negocio

Notación para el Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. Es un estándar internacional de modelado de procesos aceptado por la comunidad, independiente de cualquier metodología de modelado de procesos, crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos. (P.Briol, 2012)

## Capítulo 1: Fundamentación teórica

---

Una vez detallados los lenguajes a utilizar, se analizan las herramientas necesarias para ponerlos en práctica.

### 1.5.3. Herramientas

Siguiendo como objetivo lograr un buen resultado en la solución del software, fue necesario el uso de herramientas. A continuación se describen las características del NetBeans 7.2, Visual Paradigm 8.0 y PostgreSQL 9.3.

Un entorno de desarrollo integrado (IDE por sus siglas en inglés) es un programa compuesto por un conjunto de herramientas para programar en un lenguaje de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. (Fergarcia, 2013)

A continuación se describe el IDE de desarrollo utilizado para la implementación de la solución.

**NetBeans** es un IDE disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans es de código abierto y permite a los desarrolladores crear rápidamente aplicaciones para móviles, para la web y de escritorio.

Características (Oracle Corporation, 2015):

- Multiplataforma
- Soporte para diversos lenguajes de programación
- Integración con marcos de trabajos
- Importar y exportar proyectos
- Múltiples idiomas

**Netbeans 7.2** ofrece una mejora significativa del rendimiento y la experiencia de codificación, con nuevas capacidades de análisis de código estático en el editor Java y escaneo de proyecto inteligente.

### Herramienta CASE Visual Paradigm 8.0

Visual Paradigm es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad y con un bajo costo. Permite construir diagramas de



# Capítulo 1: Fundamentación teórica

---

diversos tipos, generar código desde diagramas y generar documentación. Soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Facilita el modelado de UML, pues proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta. Permite generar diversos informes a partir de la información introducida en la herramienta. (Cabrera Gonzalez, y otros, 2012)

## **Sistema de gestión de base de datos**

PostgreSQL es un sistema de gestión de base de datos relacional y de código abierto. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (PostgreSQL, 2015)

## **Servidor de aplicaciones web:**

### **Apache Tomcat 7.0.54**

Apache, también conocido como simplemente Tomcat o Jakarta Tomcat, es un servidor web multiplataforma que funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jackarta perteneciente a la *Apache Software Foundation* bajo la licencia Apache 2.0 y que implementa las especificaciones de los *servlets* y de *JavaServer Pages* o JSP de Sun Microsystem. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos establecidos por la Apache Software Foundation. (Apache Foundation)

# Capítulo 1: Fundamentación teórica

---

## Herramienta de versionado:

### TortoiseSVN 1.7.0

TortoiseSVN es un cliente de *Subversion*, implementado como una extensión de shell de Microsoft Windows, que ayuda a los programadores a manejar diferentes versiones del código fuente de sus programas. Es un software libre publicado bajo la licencia GNU *General Public License*. Lo que significa que es totalmente gratuito para que cualquiera lo use, incluso en un entorno comercial, sin ningún tipo de restricción. (TortoiseSVN team)

### 1.5.4. Tecnologías

**Grails 2.2.1:** Grails es un *framework* libre para desarrollo de aplicaciones web construido sobre cinco fuertes pilares (Brito, 2009):

- **Groovy** para la creación de propiedades y métodos dinámicos en los objetos de la aplicación
- **Spring** para los flujos de trabajo e inyección de dependencia
- **Hibernate** para la persistencia
- **SiteMesh** para la composición de la vista
- **Ant** para la gestión del proceso de desarrollo

Desde el punto de vista del diseño, Grails se basa en dos principios fundamentales:

**Convención sobre configuración:** aunque en una aplicación existen archivos de configuración, es un poco probable que sean editados manualmente, el sistema se basa en convenciones. Por ejemplo todas las clases de la carpeta `grails-app/controllers` serán tratados como controladores, y se mapearán en las direcciones web de la aplicación.

**Don't repeat yourself (“No te repitas”):** la participación de Spring Container en Grails permite inyección de dependencias mediante IoC (Inversión de control), de forma que cada actor en la aplicación debe definirse una única vez, haciéndose visible a todos los demás de forma automática.

### Bootstrap 2.2.2

Bootstrap es un *framework* para el diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos

## Capítulo 1: Fundamentación teórica

---

de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Bootstrap es compatible con la mayoría de los navegadores web. (Cochran, 2012)

### **jQuery 1.9.1**

jQuery es una librería de código abierto del lenguaje JavaScript diseñada para simplificar el scripting del lado del cliente de HTML. jQuery es la librería más popular de JavaScript en uso en la actualidad. La sintaxis de jQuery está diseñada para facilitar la navegabilidad de un documento, seleccionar elementos DOM, crear animaciones, manejar eventos, y desarrollar aplicaciones Ajax. jQuery también proporciona facilidades para los desarrolladores crear plugins. Esto permite a los desarrolladores crear abstracciones para interacciones de bajo nivel, animación y efectos avanzados. El enfoque modular de la librería jQuery permite la creación de potentes páginas web dinámicas y aplicaciones web. (Bibeault, et al., 2008)

### **Librería OpenCV**

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas. Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras y visión robótica (Itseez, 2014). Esta librería es utilizada para auxiliar la edición de las imágenes antes de archivarlas en el sistema.

### **Java Web Start**

Java Web Start es la implementación de referencia de la especificación JNLP (*Java Networking Launching Protocol*) que define cómo ejecutar aplicaciones Java remotamente desde un entorno de red cualquiera. Java Web Start revoluciona el concepto tradicional que se tiene de las aplicaciones. Normalmente cuando se debe ejecutar una aplicación que no se encuentra instalada en un equipo, se descarga del servidor, se instala en dicho equipo y por último se ejecuta. Java Web Start intenta simplificar al máximo todo este proceso de modo que el usuario

# Capítulo 1: Fundamentación teórica

---

lo único que tiene que hacer para lanzar una aplicación sea simplemente dar clic en un enlace de su navegador, a partir de ese momento, todo el proceso relacionado con la descarga, instalación y ejecución del programa se realiza de una manera transparente (Oracle Corporation, 2013). Esta tecnología es utilizada para la comunicación y el envío de las imágenes escaneadas y editadas desde el cliente al sistema en el servidor.

## Conclusiones del capítulo

En este capítulo se elaboró el marco teórico de la investigación, definiendo las tecnologías, metodología y herramientas necesarias para el desarrollo del sistema. De acuerdo al estudio realizado y teniendo en cuenta las especificaciones para la implementación del módulo definidas por el proyecto (Sixto, 2014), se emplearán Java y Groovy como lenguajes de desarrollo con la herramienta Netbeans en su versión 7.2. Se utilizará RUP como metodología de desarrollo y se empleará el Visual Paradigm en su versión 8.0 como herramienta CASE, utilizando el lenguaje de modelado UML 2.0 para la generación de diagramas y BPMN para el modelado del negocio. Como sistema gestor de base de datos se empleará el Postgres SQL en su versión 9.3. A partir del análisis de los principales protocolos para la interacción con el escáner, fueron seleccionados TWAIN y SANE para la implementación en Microsoft Windows y GNU/Linux respectivamente. Para auxiliar el procesamiento de la imagen se seleccionó la librería OpenCV y para la integración final del módulo con el XABAL Arkheia el Grails 2.2.1. Por medio del análisis de las soluciones similares a nivel nacional e internacional, se obtuvieron patrones de referencia acerca de la forma de descripción archivística y el tratamiento de la imagen postdigitalizada con el objetivo de lograr indicadores de competitividad en el diseño e implementación del sistema.

# Capítulo 2: Características de la solución

## Capítulo 2: Características de la solución

### Introducción

En el presente capítulo se desarrolla la descripción detallada del proceso de negocio, estrechamente relacionado con el campo de acción definido *ut supra*. Este proceso es modelado utilizando una herramienta CASE, permitiendo la visualización de las funcionalidades del sistema que se desea implementar. Se describe la propuesta de solución, analizando las necesidades planteadas y tomando referencias de los sistemas similares estudiados, pero demostrando la viabilidad de la solución planteada. Se lleva a cabo la captura de los requisitos funcionales y no funcionales, con el objetivo de guiar el curso de la implementación y por otra parte se realiza la descripción y representación de los casos de uso del sistema, tomando como referencia el caso de uso “Describir fotografía”.

### 2.1. Modelo de negocio

A continuación se describe el proceso de negocio del módulo Procesamiento de fotografía:

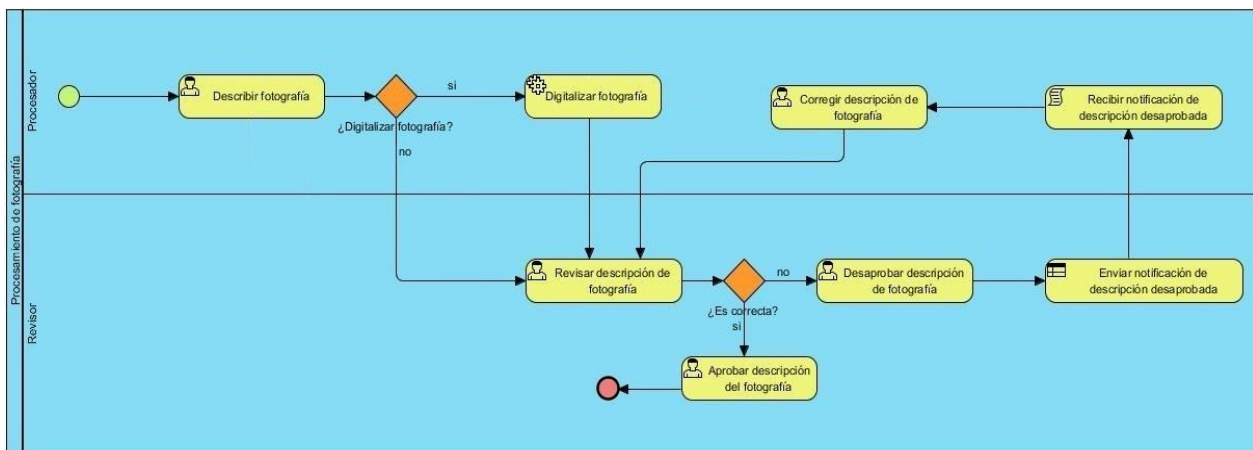


Figura 6: Diagrama del proceso de negocio “Procesamiento de fotografía”.

Este proceso de negocio inicia con la descripción de la fotografía por parte del procesador siguiendo las normas establecidas y opcionalmente su digitalización posterior. A partir de la descripción generada, el revisor comprueba que fue realizada correctamente y si la descripción es correcta, es aprobada finalizando así el proceso. Si la descripción no es correcta, el revisor la desaprueba, enviando una notificación al procesador encargado que, al recibir la notificación, corrige la descripción y posteriormente retorna nuevamente al revisor, creando un ciclo que no culminará hasta su aprobación.

## Capítulo 2: Características de la solución

---

### 2.2. Requisitos

#### 2.2.1. Requisitos funcionales

##### **RF1-** Describir documentos fotográficos

El sistema debe permitir realizar descripciones de documentos fotográficos.

##### **RF2-** Modificar descripción de fotografía

El sistema debe permitir modificar descripciones de documentos fotográficos.

##### **RF3-** Visualizar descripción de fotografía

El sistema debe mostrar la descripción de la fotografía seleccionada.

##### **RF4-** Registrar descriptores de la fotografía

El sistema debe permitir registrar los descriptores de una fotografía (materias, onomásticos, geográficos e institucionales).

##### **RF5-** Revisar descripciones de fotografías

El sistema debe permitir revisar las descripciones de fotografías realizadas, otorgándole la condición de aprobada o desaprobada.

##### **RF6-** Realizar comentarios a un documento desaprobado

El sistema debe permitir que a un documento desaprobado se le puedan añadir comentarios.

##### **RF7-** Notificar descripciones desaprobadas

El sistema debe notificar a un archivero que tiene una descripción desaprobada.

##### **RF8-** Rectificar descripciones desaprobadas

El sistema debe permitir que el archivero pueda rectificar las descripciones que le fueron desaprobadas.

##### **RF9-** Otorgar permisos a ficheros digitalizados

El sistema debe permitir que se le de permisos de acceso a cada fichero digitalizado.

##### **RF10-** Mostrar los ficheros digitales asociados a la posición del documento

El sistema debe mostrar los ficheros digitales asociados a la posición del documento que se está describiendo siempre que estos existan.

## Capítulo 2: Características de la solución

---

### **RF11-** Filtrar descripciones de fotografía

El sistema debe permitir filtrar las descripciones de fotografía por diferentes criterios.

### **RF12-** Reconocer dispositivo de captura de imágenes

El sistema reconoce los dispositivos conectados localmente.

### **RF13-** Seleccionar dispositivo de captura de imágenes

El usuario selecciona el dispositivo de escaneo.

### **RF14-** Ejecutar la aplicación de software para la digitalización

El usuario ejecuta la operación de digitalizar.

### **RF15-** Archivar la copia digitalizada

El usuario inserta la imagen digitalizada en una unidad de conservación.

### **RF16-** Modificar el parámetro “brillo de la imagen”

El sistema permite al usuario modificar el brillo de la imagen escaneada.

### **RF17-** Modificar el parámetro “contraste de la imagen”

El sistema permite al usuario modificar el contraste de la imagen escaneada.

### **RF18-** Modificar el parámetro “saturación de la imagen”

El sistema permite al usuario modificar la saturación de la imagen escaneada.

### **RF19-** Rotar imagen

El sistema permite al usuario rotar la imagen escaneada.

### **RF20-** Seleccionar formato de digitalización de la imagen

El usuario selecciona el formato para escanear la imagen.

### **RF21-** Transformar el formato de la imagen para su ingreso al sistema

Debido a su tamaño el sistema modifica el formato de la imagen para almacenarla en la unidad de conservación seleccionada.

### **RF22-** Modificar el parámetro “resolución de la imagen”

El sistema permite al usuario modificar la resolución de la imagen escaneada.

### **RF23-** Visualizar imagen digitalizada

## Capítulo 2: Características de la solución

---

El sistema muestra la imagen escaneada.

**RF24-** Digitalizar documento sin procesar

El sistema permite digitalizar sin existir previamente una descripción de la imagen a escanear.

### 2.2.2. Requisitos no funcionales

#### 1. Portabilidad

- Se podrá ejecutar la aplicación en Windows y Linux que tengan instalada la Máquina Virtual de Java (JVM, por sus siglas en inglés).

#### 2. Restricciones en el diseño y la implementación

**RnF16.** Lenguaje de programación en el cliente: para la programación del lado del cliente se utiliza HTML y JavaScript.

**RnF17.** Tecnologías en el cliente: las tecnologías utilizadas del lado del cliente son jQuery 1.9.1 y Bootstrap 2.2.2

**RnF18.** Lenguajes de programación en el servidor: se utilizan los lenguajes Java y Groovy para programar la aplicación del lado del servidor.

**RnF19.** Tecnologías en el servidor: se utiliza el *framework* Grails como marco de trabajo para construir la aplicación web.

**RnF20.** Lenguaje de programación de la base de datos: se utiliza el lenguaje SQL y plpgsql en la base de datos.

**RnF22.** Tecnologías en el servidor de base de datos: la base de datos de la aplicación está implementada sobre postgresql. Se utiliza el PGAdminIII para administrar la base de datos.

**RnF23.** Contenedor de aplicación web: para desplegar la aplicación se utiliza Apache Tomcat 7.0.54.

**RnF24.** Entorno de desarrollo integrado: el programador tiene la libertad de elegir un IDE entre los que se relacionan a continuación:

- Netbean 6.5 > \*
- IntelliJIdea 13.0.1 > \*
- STS 2.5.2 > \*
- Editor de texto y consola del sistema operativo

**RnF25.** Librerías que se utilizan en el desarrollo:



## Capítulo 2: Características de la solución

- Lenguaje de programación Java
- Librería OpenCV
- Librería mmscomputing

### 3. Usabilidad

**RnF1.** Tipo de usuario final: la aplicación está destinada a técnicos medio y universitarios en el área de la archivística con las características relacionadas en la siguiente tabla:

*Tabla 1: Datos de los usuarios finales.*

No	Sexo	Edad	Nivel de escolaridad	Ocupación	Experiencia profesional	Experiencia con aplicaciones informáticas	Tipo de discapacidad	Otras
1	M, F	23 – 75	Técnicos medio, universitarios	Archiveros	2 – 50 años	2 años	Ninguna	

**RnF2.** Tipo de aplicación informática: aplicación web.

**RnF4.** Ambiente: para el despliegue de la aplicación se requieren dos servidores, uno para desplegar el producto y otro para ubicar la base de datos. Los datos de estos servidores se muestran en la siguiente tabla:

*Tabla 2: Requisitos técnicos de los servidores*

<b>Hardware (Servidor de aplicación)</b>	
Cantidad	<b>1</b>
CPU	<b>4 x 2.33 GHz</b> (Intel Core2 2.33 GHz)
RAM	<b>4 Gb</b>
HDD	<b>250 Gb</b> (en dependencia del monto de información que la institución desee almacenar)
LAN	<b>1 x NIC</b> (1 Gbit)
SAN	-
Fuentes de Alimentación	<b>2 x 800W</b>
<b>Hardware (Servidor de almacenamiento de información)</b>	
Cantidad	<b>1</b>
CPU	<b>4 x 2.33 GHz</b> (Intel Core2 2.33 GHz)
RAM	<b>2 Gb</b>
HDD	<b>250 Gb</b>
LAN	<b>2 x NIC</b> (1 Gbit)
SAN	<b>1.5 Tb</b>

## Capítulo 2: Características de la solución

Fuentes de Alimentación	2 x 800W
-------------------------	----------

Garantizando estas condiciones mínimas, la aplicación brinda un tiempo de respuesta en el rango de los 700 ms por cada interfaz visualizada, exceptuando las interfaces de los reportes que sí implica un gran cómputo del sistema. Para el usuario final se requiere de un puesto de trabajo con acceso a la red de la institución y con un navegador web instalado.

### 2.2.3. Técnicas de validación de requisitos

Luego de realizar la descripción de los requisitos, estos necesitan ser validados. La validación de requisitos tiene como misión verificar que realmente posean las características que debe tener el sistema y lo que el cliente realmente desea. Las técnicas utilizadas para el desarrollo de este trabajo fueron (Escalona, et al., 2002):

**Revisiones de requisitos:** esta técnica permite verificar cada uno de los requisitos para validar la correcta interpretación de la información transmitida, este proceso fue llevado a cabo por los analistas del proyecto Archivo y fueron detectadas algunas inconveniencias como:

- Presencia de errores ortográficos y de concordancia
- Ocurrencia de palabras ambiguas

Como resultado surgieron algunas recomendaciones de cambios de textos para facilitar la comprensión de los mismos. Todas las anomalías fueron solucionadas y aceptadas, logrando una mejor interpretación de las funcionalidades del sistema.

**Construcción de prototipos:** se muestran los prototipos de interfaz a los usuarios finales, con el objetivo de comprobar si el sistema cumple con los elementos requeridos y lograr una mayor comprensión de la propuesta de requisitos.

Fueron aceptados el 85% de los requisitos propuestos, el resto fue modificado teniendo en cuenta los siguientes aspectos:

- Nombres de los componentes de la interfaz
- Cambios de tipos de componentes
- Eliminación y adición de botones

## Capítulo 2: Características de la solución

---

La utilización de esta técnica facilitó el diseño de las vistas del sistema y el entendimiento de la propuesta de requisitos por parte de los analistas, arquitectos y el jefe de proyecto.

### 2.3. Análisis y selección de una librería para el trabajo con el lenguaje Java y los protocolos SANE y TWAIN

#### Asprise jsane

Asprise jsane es una librería desarrollada por Asprise para la comunicación con dispositivos de adquisición de imágenes mediante el lenguaje Java. Implementa el protocolo TWAIN y es completamente compatible con los sistemas operativos Windows y Mac. La librería muestra una interfaz con múltiples opciones para el manejo y almacenamiento de las imágenes, posee varias clases encargadas de importar la imagen desde el escáner y modificar sus parámetros.

Sus principales características son (Lab Asprise, 2012):

- Simple pero potente adquisición de imágenes: JSANE proporciona un conjunto de API limpio y sencillo para el acceso y el control de las cámaras digitales y escáneres. En su caso más sencillo, tres líneas de código Java pueden hacer el trabajo. Parámetros de adquisición de la imagen se pueden ajustar de una manera elegante.
- Flexibilidad: JSANE se puede utilizar en cualquier aplicación independiente Java, aplicación web, etc.
- Independiente del dispositivo: la mayoría de los escáneres son apoyados por JSANE.
- Producto maduro y estándar de compatibilidad: JSANE v2 v1.0 soporta de SANE en adelante.
- Soporte Carga Web: guardar las imágenes adquiridas y subirlas a cualquier servidor web.
- Soporte TIFF: guardar las imágenes escaneadas en archivos TIFF con Asprise Java TIFF Biblioteca.

#### mmscomputing

Es una librería Java de código abierto creada por Michael Meiwald que implementa los protocolos TWAIN para Windows y SANE para Linux y es totalmente compatible con JRE 1.4.2 o superior. Es una alternativa libre que se ha extendido debido a contribuciones de terceros a sus funcionalidades. (Gustavo Suhit, 2008)

## Capítulo 2: Características de la solución

---

### **Morena**

Morena es una librería de adquisición de imágenes para la plataforma Java, sirviendo como puente entre el dispositivo de adquisición de imágenes (escáneres, cámaras) y el mundo Java. Para comunicarse con el hardware utiliza la interfaz estándar TWAIN para Microsoft Windows y Mac OS X y SANE para un número de plataformas Unix (AIX, BeOS, Darwin, FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS2, Solaris, Unixware). Morena es una librería licenciada. (Gnome Ltd, 2013)

### **JTwain**

JTwain es una librería licenciada que permite el manejo de imágenes TIFF multipágina y que permite la comunicación entre el lenguaje Java y el protocolo TWAIN. Sus funcionalidades fundamentales poseen restricciones para su libre implementación. (Gustavo Suhit, 2008)

### **Selección de una librería para la comunicación entre el lenguaje Java y los protocolos TWAIN y SANE**

A partir del análisis realizado a cada una de las librerías investigadas, fue seleccionada la librería mmscomputing. Los parámetros utilizados en la comparación, estuvieron basados fundamentalmente en la licencia, que en el caso de Jsane, JTwain, Asprise y Morena son librerías propietarias y en la compatibilidad con el sistema, que resultó beneficiar a mmscomputing debido a que sus funcionalidades se acoplan con mayor facilidad al entorno web y su código fuente está disponible para los desarrolladores.

#### **2.4. Propuesta de solución**

Para dar respuesta a la problemática planteada y haciendo uso de las herramientas y metodología seleccionada para la construcción del sistema, se propone la implementación del módulo “Procesamiento de fotografía” perteneciente al subsistema “Incorporación y Organización Documental” del XABAL Arkheia (figura 5). El módulo debe contar con las funcionalidades exclusivas de describir archivos fotográficos utilizando la norma ISAD (G) y asociarla a la digitalización desde el escáner integrado al sistema. Para la comunicación con el dispositivo de adquisición de imágenes se proponen los estándares TWAIN para la familia de sistemas operativos Windows de Microsoft y SANE para los sistemas tipo Unix's como GNU/LINUX, haciendo el acceso a estos dispositivos transparente a la plataforma donde se

## Capítulo 2: Características de la solución

encuentre instalado el software. El especialista debe ser capaz de seleccionar el dispositivo de escaneo deseado y una vez digitalizada la imagen debe ser capaz de modificar los parámetros de brillo, contraste, saturación, cantidad de bits por pixel, tipo de imagen, rotar, cortar, cambiar la resolución y restaurar la imagen a su estado inicial. Una vez editada la imagen digitalizada, el especialista debe contar con la posibilidad de insertarla asociada a la descripción seleccionada, agilizando el proceso y brindando una mayor seguridad a la información.



Figura 5: Estructura de la propuesta de solución.

### 2.5. Diagramas de Casos de Uso

Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un caso de uso es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema. La definición de un caso de uso incluye todo el comportamiento que implica: las líneas principales, las diferentes variaciones sobre el comportamiento normal, y todas las condiciones excepcionales, que pueden ocurrir con tal comportamiento. (James Rumbaugh, Ivar Jacobson, Grady Booch, 2000)

#### 2.5.1. Definición de los actores

Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones

## Capítulo 2: Características de la solución

que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor. (James Rumbaugh, Ivar Jacobson, Grady Booch, 2000)

Tabla 3: Definición de los actores

Actor	Justificación
Procesador	Es el encargado de describir las fotografías de las unidades de conservación a las cuales podrá asociar sus correspondientes imágenes digitales.
Revisor	Es el encargado de revisar si las descripciones de la fotografía se realizaron correctamente.

A continuación se muestra el diagrama de casos de uso perteneciente al módulo Procesamiento de fotografía.

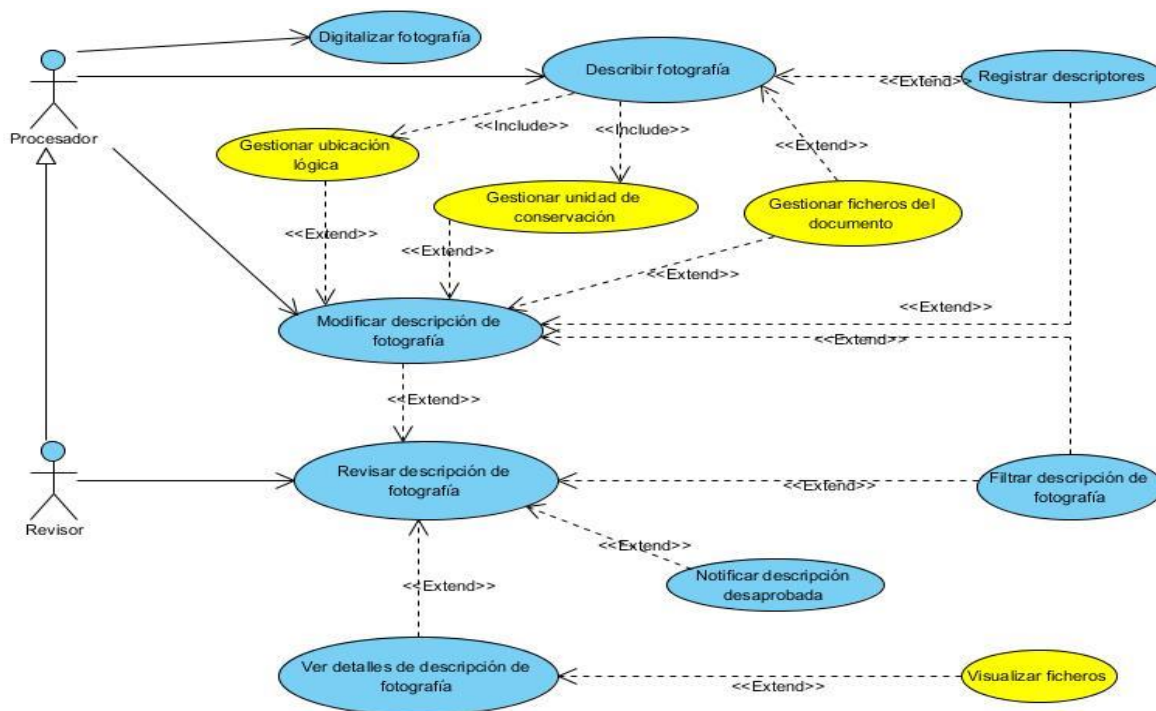


Figura 7: Diagrama de Casos de Uso. Procesamiento de fotografía.

## Capítulo 2: Características de la solución

### 2.5.2. Especificación de casos de uso

#### CU. Describir fotografía

*Tabla 4: Descripción del CU Describir fotografía.*

<b>Objetivo</b>	Realizar descripción de fotografía siguiendo los campos de la norma ISAD (G) para fotografías.	
<b>Actores</b>	Procesador: (Inicia) describe fotografía usando la norma ISAD (G).	
<b>Resumen</b>	El caso de uso se inicia cuando el Procesador selecciona la opción del menú "Describir Fotografía". El sistema muestra la interfaz que permite describir fotografía mediante la norma ISAD (G). El procesador describe la fotografía y guarda los datos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Debe existir una unidad de conservación creada.	
<b>Postcondiciones</b>	Queda registrada la descripción de fotografía.	
<b>Flujo de eventos</b>		
<b>Flujo básico Describir fotografía</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción del menú "Describir Fotografía".	
2.		Verifica que exista una unidad de conservación creada.
3.		<p>Muestra un formulario solicitando la información para registrar la descripción de fotografía.</p> <p><b>Área de Identificación</b></p> <ul style="list-style-type: none"> <li>• Título(obligatorio)</li> <li>• Nivel de descripción(obligatorio)</li> <li>• Número de foto</li> <li>• Fecha inicial</li> <li>• Fecha final</li> <li>• Ubicación lógica <i>Ver CU Gestionar ubicación lógica.</i> (Especificación de CU Módulo Clasificación. Fase 2)</li> <li>• Volumen</li> <li>• Tipo de imagen</li> </ul>

## Capítulo 2: Características de la solución

- Tipo de soporte

### Área de Contexto

- Productores

### Área de Contenido y Estructura

- Alcance y contenido (obligatorio)
- Descriptores
  - Materias
  - Onomásticos
  - Geográficos
  - Institucionales

Muestra la opción para añadir nuevos descriptores. Ver CU Registrar Descriptores.

### Área de Condiciones

- Libre acceso
- Libre reproducción
- Cromaticidad
- Largo
- Ancho
- Estado de conservación
- Dimensiones

### Área de Notas

- Notas
- Valor patrimonial

### Área de Control

- Archivero
- Fecha de la descripción
- Nota del archivero

En todas las áreas muestra los campos

- Unidad de conservación (obligatorio)
- Posición (obligatorio)

### Guardar y asociar

Muestra una tabla con los siguientes datos:

- Folio
- Número de folio
- Permiso

Muestra las opciones que permiten gestionar ficheros. Ver CU Gestionar ficheros del documento.

Muestra las opciones Aceptar y Cancelar.



## Capítulo 2: Características de la solución

1.	Introduce los datos solicitados y selecciona la opción "Aceptar".	
2.		Comprueba que se han introducido los datos obligatorios.
3.		Comprueba que los datos introducidos sean correctos. <b>Ver Modelo Conceptual [1]. Paquete fotografía.</b>
4.		Almacena los datos en la BD y muestra un mensaje indicando que los datos se han guardado correctamente.
5.		Regresa a la página que le dio origen.
6.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>*a Opción "Cancelar"</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción cancelar.	
2.		Retorna a la página que le dio origen.
2a No existe una unidad de conservación creada		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje indicando que no hay unidades de conservación creadas y la opción para crear una unidad de conservación "Crear Unidad de Conservación".
2.	Selecciona la opción "Crear Unidad de Conservación".	
3.		Muestra una interfaz que permite crear la unidad de conservación. <i>CU Gestionar Unidad de Conservación</i> . (Especificación de CU Módulo Procesamiento. Fase 2).
4.		Regresa al paso 3 del flujo básico.
5a No se han introducido todos los datos obligatorios		

## Capítulo 2: Características de la solución

	Actor	Sistema
1.		Muestra un mensaje indicando que existen campos obligatorios que no se han completado, señalando los campos obligatorios vacíos.
2.		Regresa al paso 8 del flujo básico.
6a Los datos introducidos son incorrectos.		
	Actor	Sistema
1.		Muestra un mensaje indicando que existen campos con datos incorrectos, señalando los campos.
2.		Regresa al paso 5 del flujo básico.
<b>Relaciones</b>	<b>CU incluidos</b>	Ubicación lógica en el <u>CU Gestionar ubicación lógica</u> . Módulo Clasificación.  Unidad de conservación en el <u>CU Gestionar Unidad de Conservación</u> . Módulo Procesamiento.
	<b>CU extendidos</b>	Guardar y asociar en el <u>CU Gestionar ficheros del documento</u> . Módulo Digitalización de documentos.  Añadir nuevos descriptores en el <u>CU Registrar Descriptores</u> .
<b>Requisitos no funcionales</b>	No aplica.	
<b>Asuntos pendientes</b>	No aplica.	

### Conclusiones del capítulo

En este capítulo fue elaborada y presentada la propuesta de solución basada en el análisis de la problemática y los sistemas similares estudiados. Se realizó el modelado y la descripción del proceso de negocio y de los actores que interactúan con el sistema, esclareciendo el funcionamiento y la estructura del módulo propuesto. Fueron listados y descritos los requisitos funcionales y no funcionales, contribuyendo a un mayor entendimiento de las funcionalidades a implementar y fue elegida la librería mmscomputing para la comunicación entre los protocolos TWAIN y SANE con el lenguaje Java, dando paso al diseño de la solución.

## Capítulo 3: Diseño de la solución

### Introducción

En este capítulo se presenta el diseño de la solución. Se muestran los diagramas de paquetes, clases de diseño y secuencia, así como la arquitectura del módulo y del sistema en general. Son descritos y ejemplificados además, los patrones utilizados en el diseño, adjuntando un ejemplo de su utilización en el desarrollo del sistema.

### Diseño

Pressman plantea que el diseño es lo que casi cualquier ingeniero quiere hacer. Es el sitio donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. El diseño crea una representación o modelo del software y proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. El diseño permite al ingeniero de software modelar el sistema o producto que se va a construir. Este modelo puede evaluarse en relación con su calidad y mejorarse antes de generar código, de realizar pruebas y de que los usuarios finales se vean involucrados a gran escala. El diseño es el sitio en el que se establece la calidad del software. (Pressman, 2005)

### 3.1. Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. Muchos patrones proporcionan guías sobre el modo en el que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (Larman, 2003). A continuación se explican algunos patrones existentes para asignar responsabilidades.

#### 3.1.1. Patrones GRASP

Patrones de Principios Generales para Asignar Responsabilidades (GRASP, por sus siglas en inglés). Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. (Larman, 2003)

## Capítulo 3: Diseño de la solución

---

- **Experto en información:** es el encargado de asignar las responsabilidades a aquellas clases que contengan la información necesaria para realizar una responsabilidad. Resolviendo el problema de saber a qué clase del sistema le corresponde realizar una o varias operaciones, en dependencia de la información que posea. Este patrón se evidencia dentro del módulo en la clase de dominio `DNormalISADGFotografía` que posee toda la información perteneciente a los elementos de la descripción de las fotografías.
- **Creador:** es el responsable de asignarle a una clase la responsabilidad de crear una instancia de otra clase. Resolviendo así el problema de poder crear una nueva instancia de una clase mediante otras que registren, utilicen o contengan instancias de objetos de esa clase. Este patrón se evidencia en la interacción con el escáner donde, el contenedor general de la digitalización es el encargado de crear el `ImagePanel` que contiene cada imagen.
- **Controlador:** es el encargado de asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema, resolviendo el problema de comunicación de una clase con otra mediante una clase controladora. Este patrón fue utilizado en las clases controladoras de Grails, ejemplificado en la clase `FotografíaController` que se encarga de manejar las acciones y el flujo de información entre las vistas, los servicios y el modelo de datos.
- **Alta cohesión:** asigna una responsabilidad de forma tal que la cohesión siga siendo alta. La dificultad de entender, mantener, reutilizar y las afectaciones debido a los constantes cambios, son resueltas con un alto acoplamiento debido a que las clases tienen la relación necesaria. Este patrón es utilizado en la capa de servicios, colabora con las clases que lo utilizan para llevar a cabo las tareas, de esta forma se delegan responsabilidades (lógica del negocio) a los servicios y se liberan las clases controladoras de tantas operaciones, logrando que no pierdan la cohesión. Un ejemplo de la utilización de este patrón se refleja en la clase `FototecaService`, encargada de mediar entre la controladora `FototecaController` y el dominio.
- **Bajo acoplamiento:** asigna una responsabilidad para conservar bajo acoplamiento. Al aplicar este patrón los problemas relacionados con la reutilización, el entendimiento y los cambios en las clases relacionadas serían resueltos pues existiría la relación necesaria de un elemento con otros. En el módulo este patrón es utilizado en la digitalización,

## Capítulo 3: Diseño de la solución

---

donde, la clase ImageTab es la encargada de crear instancias de la clase ImageTypeConvertOpPanel y esta a su vez es la encargada de crear instancias de la clase JPanel para la edición de las imágenes.

### 3.1.2. Patrones GoF

Los patrones Grupo de los Cuatro (GoF, por sus siglas en inglés) utilizados son:

- **Singleton:** es un patrón utilizado para asegurar la existencia de una instancia de una clase y que esta es accesible, visible y con una referencia permanente. Dándole solución al problema de encontrar un único punto de acceso local. Por defecto, en Grails todos los servicios utilizan este patrón (Brito, 2009). Crea una instancia de un servicio durante todo el ciclo de vida de la aplicación, sería como un mecanismo de visibilidad global de dicha instancia.
- **Fachada:** es un patrón que dado el problema de la necesidad de una interfaz común unificada para un conjunto de implementaciones o interfaces brinda la posibilidad de un único punto de conexión con el subsistema. Este objeto fachada presenta una única interfaz unificada y es responsable de colaborar con los componentes del sistema (Larman, 2003). El patrón Fachada se utiliza para proveer un único punto de acceso de los controladores a las funcionalidades del negocio, permitiendo que sean más reusables y fáciles de adaptar, logrando el desacople entre la capa de presentación y la capa de negocio.

### 3.2. Arquitectura de Grails

Grails es el marco de trabajo para aplicaciones web seleccionado por el equipo de arquitectura (Sixto, 2014) para desarrollar el sistema XABAL Arkheia. Desarrollado sobre el lenguaje de programación Groovy proporciona un entorno de desarrollo estandarizado y oculta gran parte de los detalles de configuración al programador.

En él se definen un conjunto de paquetes ordenados según su arquitectura y tipos de clases a almacenar para que al relacionarse entre sí generen la aplicación final.

La carpeta **grails-app** contiene todos los artefactos que el entorno irá generando y que se modificarán para adaptar el funcionamiento a la aplicación, donde (Brito, 2009):

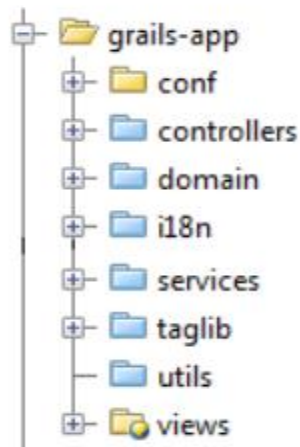


Figura 8: Estructura de los proyectos Grails.

**Conf:** contiene los archivos de configuración de la aplicación, así como las configuraciones personales de la Base de Datos, Spring e Hibernate y si se desea se puede adicionar alguna otra.

**Controllers:** es el mediador entre el modelo y la vista.

**Domain:** se almacenan las clases que Grails mapea como entidades en la base de datos.

**i18n:** contiene los archivos de internacionalización de la aplicación.

**Services:** almacena los servicios de Grails, además son los encargados de implementar la lógica del negocio.

**Taglib:** contiene la configuración de las etiquetas que el desarrollador crea para comodidad al trabajar en las vistas.

**Utils:** contiene las clases útiles para el desarrollo de la aplicación.

**Views:** contiene todas las vistas de la aplicación.

### 3.2.1. Patrón de arquitectura Modelo Vista Controlador en Grails

Este patrón es utilizado por Grails en el diseño y establece que los componentes de software deben organizarse en tres capas distintas (Brito, 2009):

## Capítulo 3: Diseño de la solución

- **Capa de datos (Modelo):** la capa está compuesta por las clases del dominio que representan y gestionan los datos manejados por la aplicación, se encargan de leer y escribir en la base de datos. Estas clases se ubican en la carpeta grail-app/domain.
- **Capa de presentación (Vista):** los componentes de esta capa son responsables de mostrar al usuario el estado actual del modelo de datos y presentarle las distintas acciones disponibles, permitiendo su interacción con el sistema. Estos elementos se ubican en la carpeta grail-app/views.
- **Capas de control (Controlador):** contendrá los componentes que reciben las órdenes del usuario, gestionan la aplicación de la lógica de negocio sobre el modelo de datos y determinan qué vista debe mostrarse a continuación, así como la información que estas contienen.

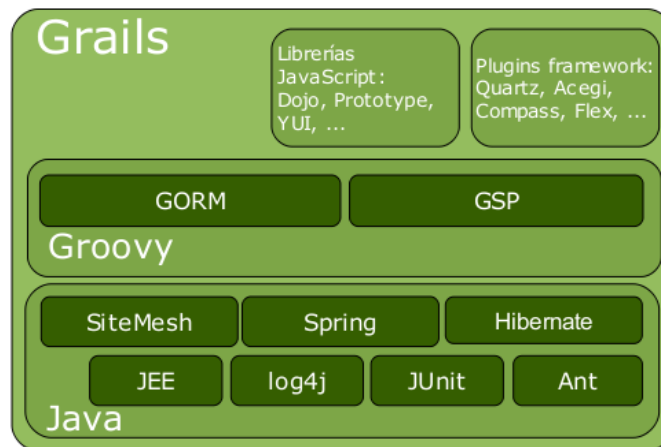


Figura 9: Arquitectura de Grails. (Brito, 2009)

### 3.3. Arquitectura del sistema

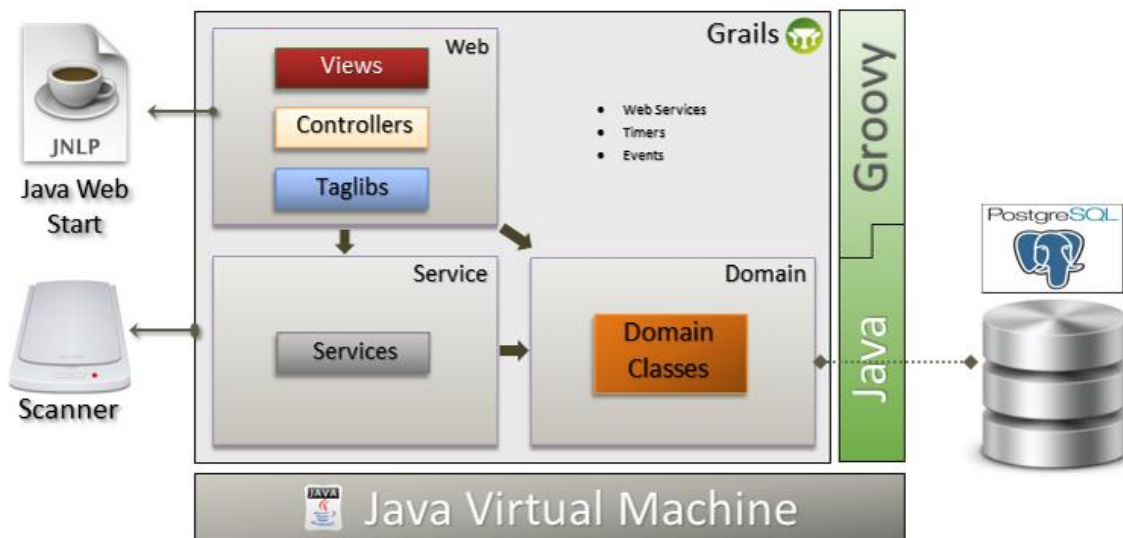


Figura 10: Arquitectura del sistema.

La arquitectura para el módulo Procesamiento de fotografía está conformada de tres capas lógicas principales (figura 10): Capa Web, Capa de Servicios y Capa de Datos, basadas en el patrón **Modelo Vista Controlador (MVC)**. Cada capa brinda sus utilidades a la capa inmediatamente superior y se sirve de las prestaciones que le proporciona la inmediatamente inferior. Esta arquitectura en capas por su diseño proporciona la facilidad de modificar cada capa todo lo posible sin infligir daños o alteraciones a la capa inmediata.

**Capa web:** se encarga de manejar dos partes, las Vistas y las Controladoras del patrón MVC. Esta capa interactúa directamente con el protocolo *Java Network Launching Protocol* (JNLP) utilizado para las aplicaciones de Java Web Start, que es la encargada de cargar la aplicación para la digitalización con el escáner.

- **Vistas:** muestran una interfaz al usuario
- **Controladoras:** gestionan la aplicación mediante la recepción de las acciones del usuario

**Capa de datos:** se encarga de proveer la comunicación con los datos almacenados en la Base de Datos.



## Capítulo 3: Diseño de la solución

---

**Capa de servicios:** contiene los componentes encargados de implementar, manejar y validar la lógica de negocio de la aplicación. Los servicios se ubican en la carpeta grails–app/services.

Cuando se trabaja con Grails se generan componentes en cada una de las capas y es el entorno el que se encarga de conectar unos con otros y garantizar su buen funcionamiento, obteniendo componentes fáciles de manejar, dando paso a la reutilización del código en mayor medida.

### 3.4. Diagrama de clases de diseño

Un diagrama de clases de diseño representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Se ofrece para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad; Relaciones: herencia, composición, agregación, asociación y uso (Egdamar, 2012). A continuación se muestra el diagrama de clases de diseño del Caso de Uso “Gestionar fotografía”.

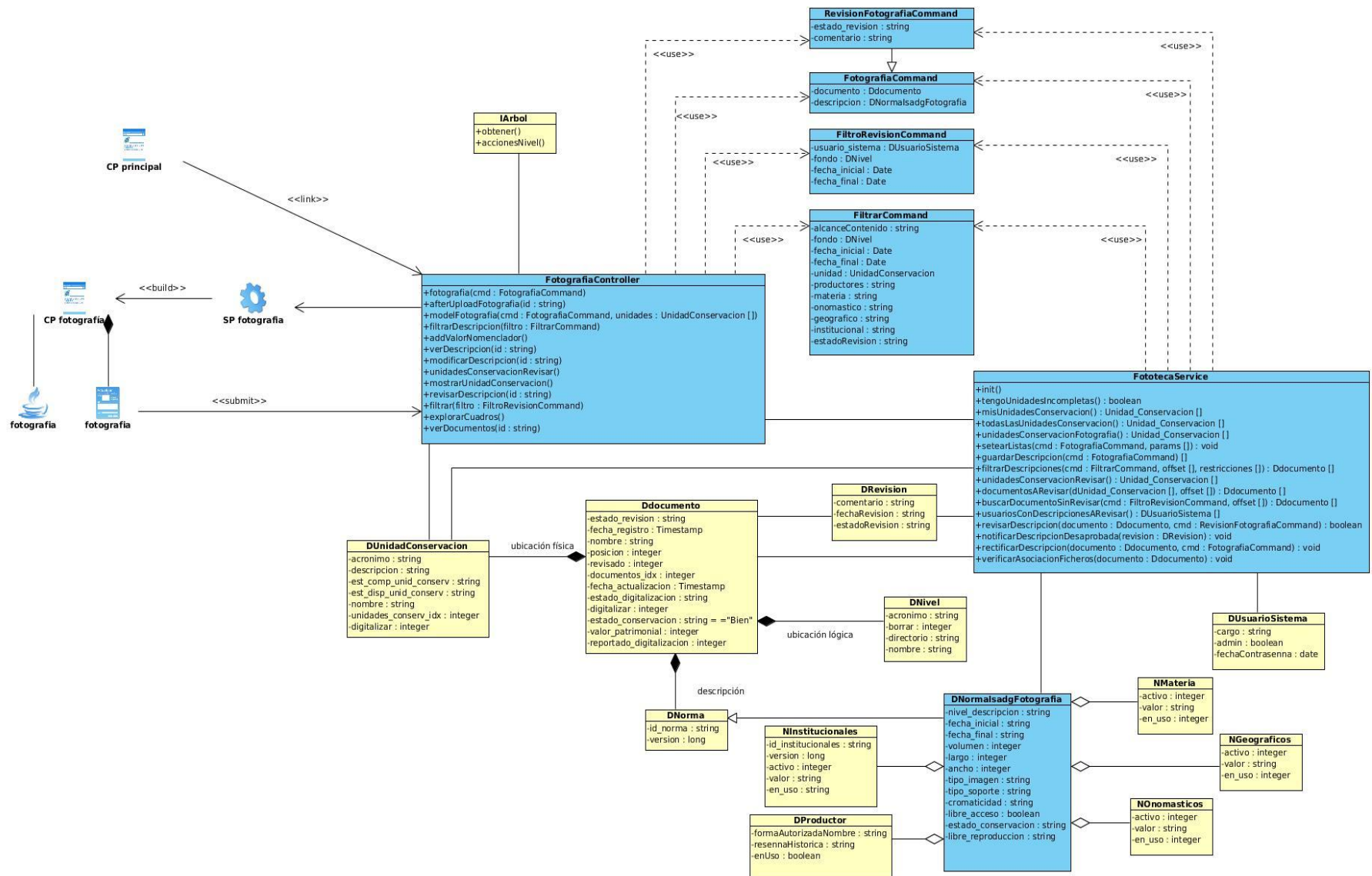


Figura 11: Diagrama de clases de diseño. Gestionar fotografía.

### 3.5. Diagrama de paquetes

Los diagramas de paquetes permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales. Los paquetes pueden estar anidados unos dentro de otros, y unos paquetes pueden depender de otros paquetes. Se pueden utilizar para plantear la arquitectura del sistema a nivel macro. El diagrama muestra cómo está estructurado el sistema, cada paquete puede contener otros paquetes o clases, que tienen interfaces y realizan cierta funcionalidad (Gutierrez, 2009). A continuación se muestra el diagrama de paquetes del módulo Procesamiento de fotografía, facilitando una descomposición de su jerarquía lógica.

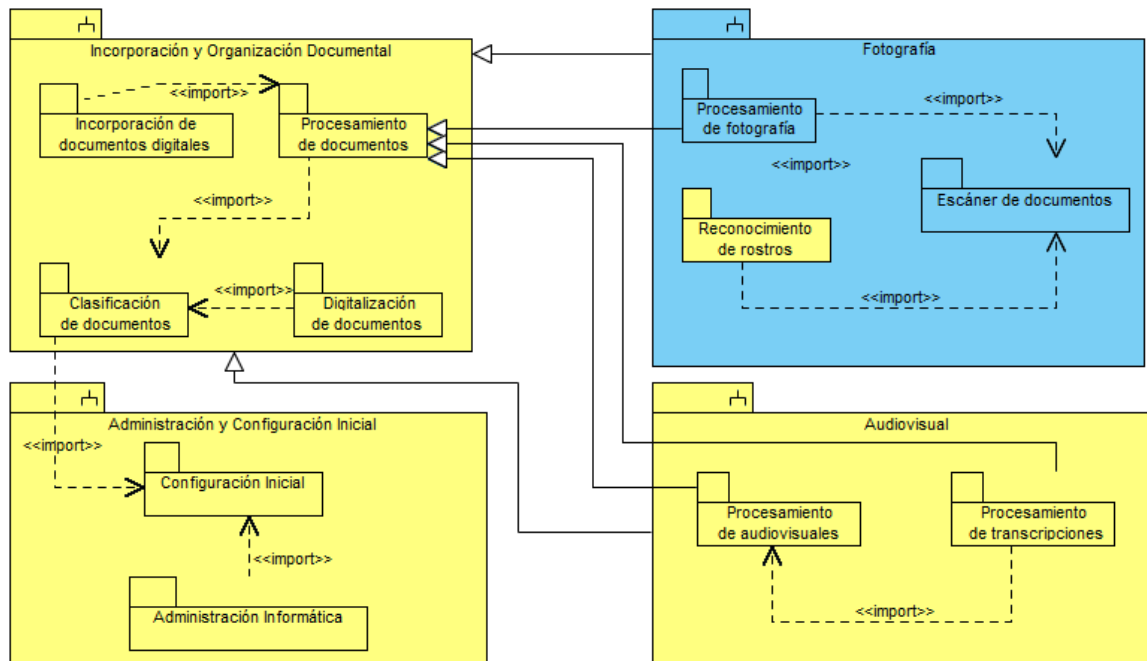


Figura 12: Diagrama de paquetes. Módulo Procesamiento de fotografía.

### 3.6. Diagrama de secuencia

Los diagramas de secuencias muestran la forma en que un grupo de objetos se comunican (interactúan) entre sí a lo largo del tiempo. Consta de objetos, mensajes entre estos objetos y una línea de vida del objeto. A continuación se muestra el diagrama de secuencia del CU Describir fotografía:

## Capítulo 3: Diseño de la solución

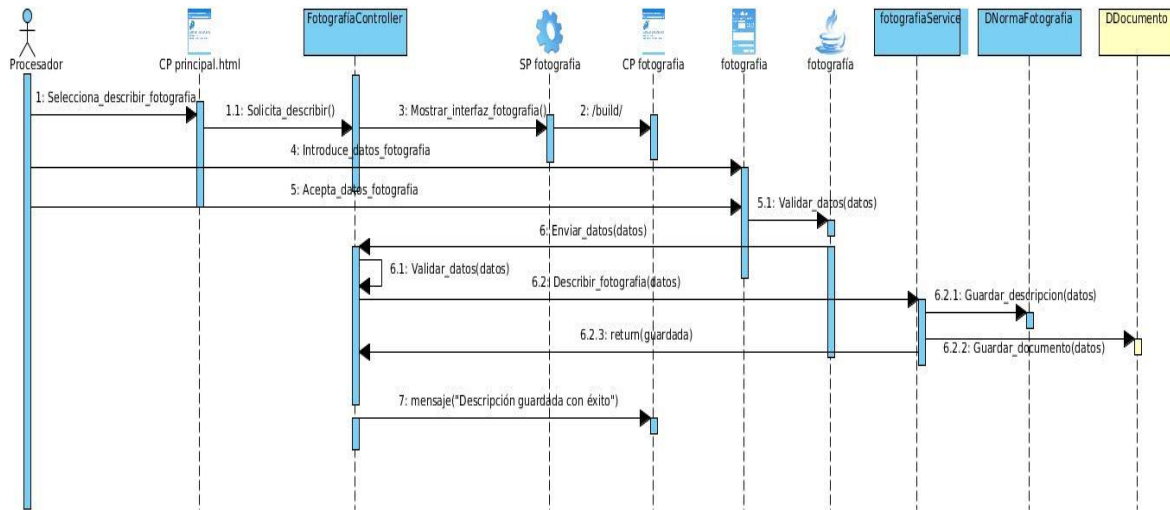


Figura 13: Diagrama de secuencia. Describir fotografía.

### Conclusiones del capítulo

En este capítulo se llevó a cabo el desarrollo de la arquitectura y diseño del sistema. Fueron modelados algunos de los artefactos para los casos de uso más significativos: diagramas de clases de diseño, de secuencia y de paquetes a partir de las especificaciones de los requisitos funcionales para el modelo del diseño. Se realizó una caracterización de las arquitecturas del marco de trabajo Grails y del sistema, así como su relación. Se realizó la selección de los patrones de diseño como estándares de codificación para mejor organización y claridad, ejemplificando su uso en la implementación. Con los artefactos elaborados se obtuvo un mejor enfoque de los elementos que guiarán la implementación de la propuesta de solución.

## Capítulo 4: Implementación y prueba

### Introducción

En el presente capítulo se abordan las fases de Implementación y prueba del sistema. Como parte del desarrollo de la fase de Implementación los diagramas de componentes y despliegue desempeñan un papel fundamental, pues no describen la funcionalidad del software, sino su estructura general con vista a su construcción, ejecución e instalación. Con el objetivo de verificar el 100 % del cumplimiento de los requisitos funcionales acordados, se realizan las pruebas al sistema. El éxito de las pruebas se basa en detectar la presencia de errores para su corrección, mostrando así los resultados obtenidos una vez aplicadas.

#### 4.1. Diagrama de componentes

El diagrama de componentes describe la descomposición física del sistema de software (y, eventualmente, de su entorno organizativo) en componentes, a efectos de construcción y funcionamiento. (Pressman, 2005). La descomposición del diagrama se realiza en términos de componentes y de relaciones entre los mismos. Los componentes identifican objetos físicos que hay en tiempo de ejecución, de compilación o de desarrollo, y tienen identidad propia y una interfaz bien definida (Falgueras, 2005). A continuación se muestra el diagrama de componentes del sistema.

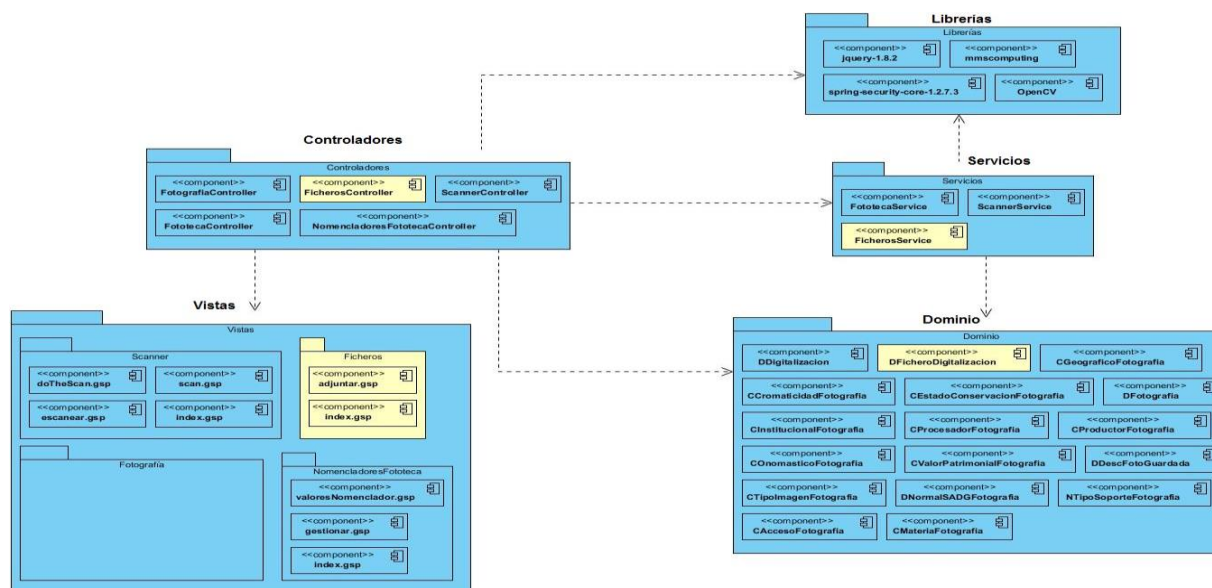


Figura 14: Diagrama de componentes del sistema.

## Capítulo 4: Implementación y prueba

### 4.1.1. Descripción de los componentes

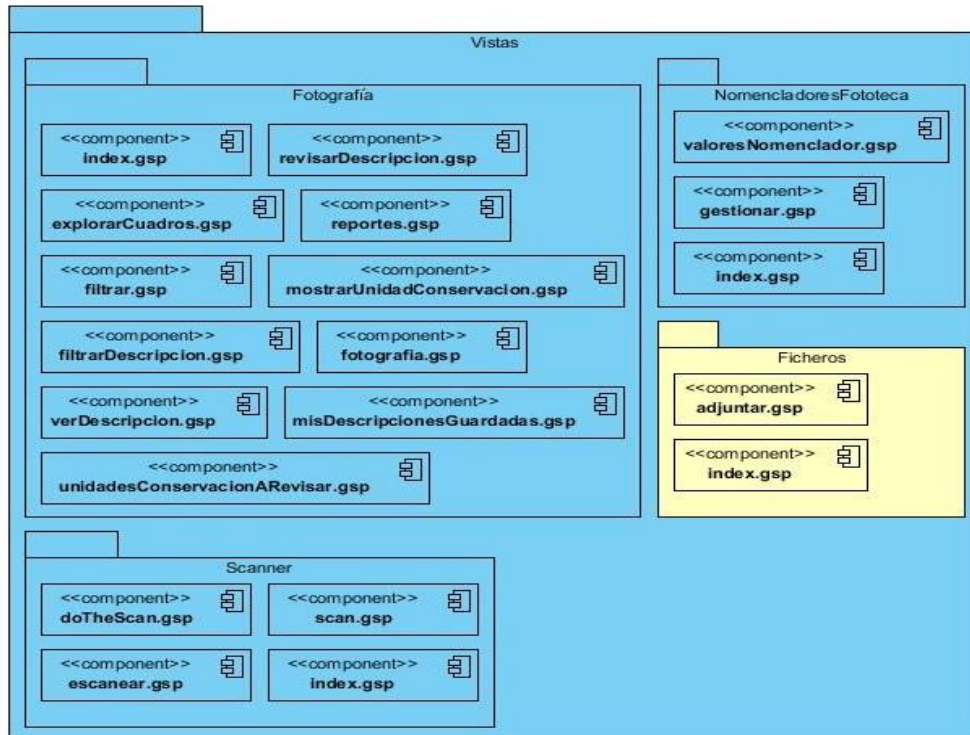


Figura 15: Paquete de componentes: Vistas.

El paquete de componente Vistas es una representación de las páginas .gsp con las cuales el usuario interactúa, permitiendo la entrada de datos al sistema y la realización de consultas a dichos datos.

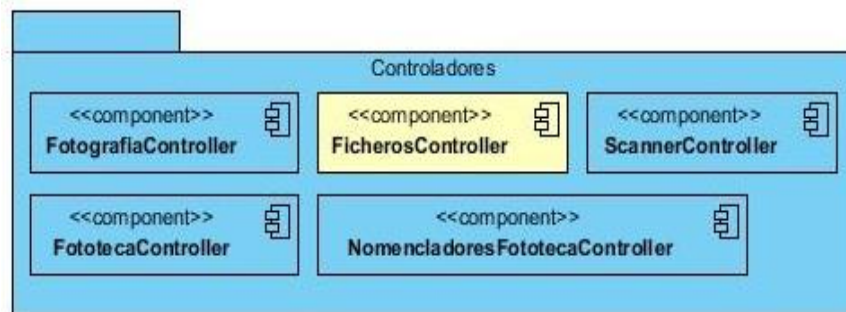


Figura 16: Paquete de componentes: Controladores.

El paquete de componentes Controladores es una representación de las operaciones que permiten el flujo y la comunicación entre las vistas y el dominio, son los encargados de recibir las órdenes por parte del usuario, gestionar la ejecución de la lógica de negocio y



## Capítulo 4: Implementación y prueba

posteriormente actualizar la vista para que el usuario pueda ver cómo ha quedado el modelo de datos tras las actualizaciones.

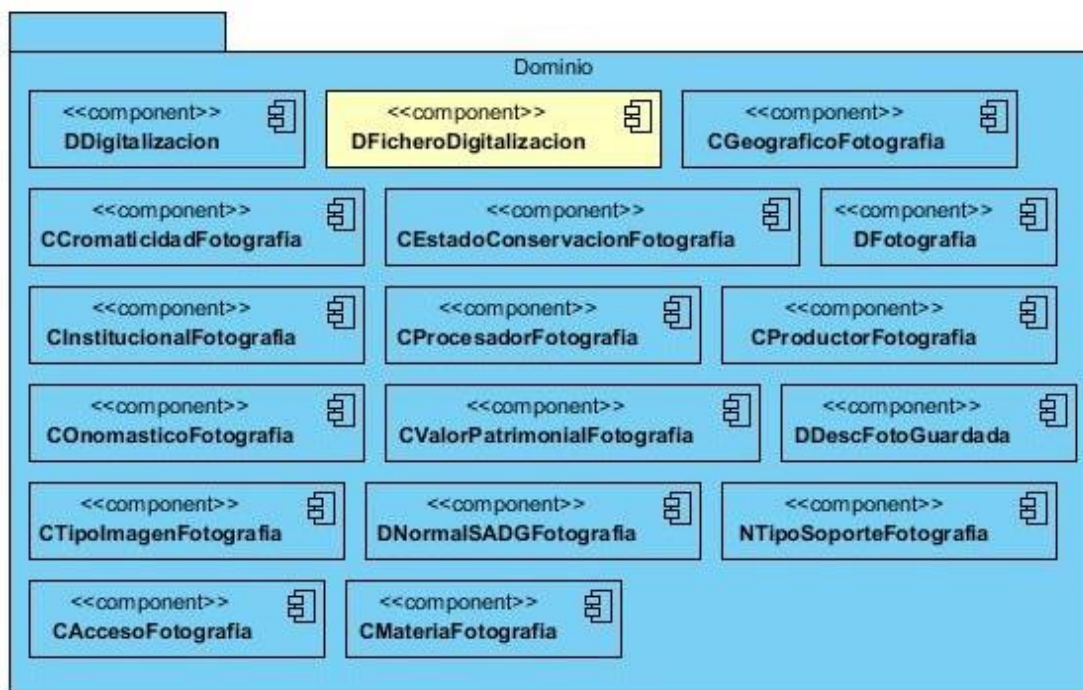


Figura 17: Paquete de componentes: Dominio.

En el paquete Dominio están las clases que representan todo el negocio del sistema. Estas clases del dominio constituyen entidades o tablas que son utilizadas en la base de datos, siendo este un gestor de persistencia para controlar el ciclo de vida de dichas entidades.

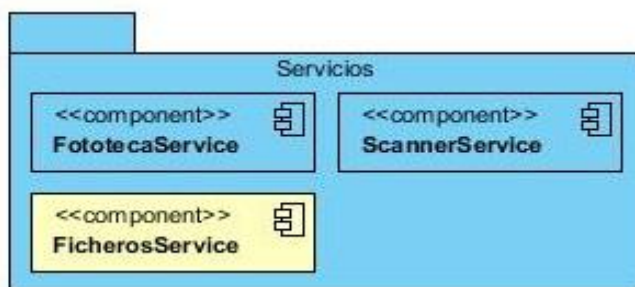


Figura 18: Paquete de componentes: Servicios.

Este paquete contiene todos los servicios utilizados que no son más que las clases que tienen como objetivo implementar la lógica del negocio, dándole respuestas a las peticiones de las clases controladoras.

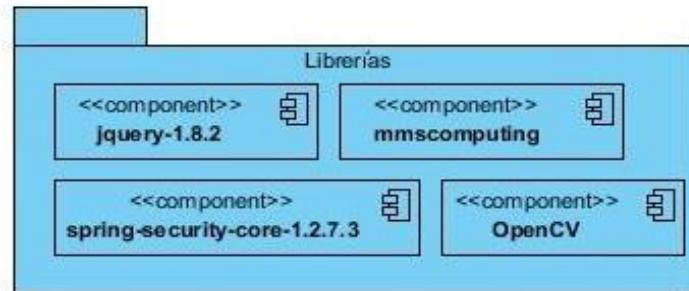


Figura 19: Paquete de componentes: Librerías.

Como librerías para el sistema se utilizaron las pertenecientes a Grails, jQuery que es una librería JavaScript para el manejo del DOM (HTML), y *spring-security-core* que es la que se encarga de gestionar lo referente a la seguridad en el sistema y para la adquisición de imágenes se utilizó *mmscomputing* auxiliado de *OpenCV* para la edición de imágenes.

### 4.2. Diagrama de despliegue

El diagrama de despliegue permite mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y software, se utiliza en el diseño y la implementación. Se pueden distinguir componentes y nodos, así como las relaciones entre todos. Es más limitado que el diagrama de componentes, en el sentido de que se representa la estructura del sistema sólo en tiempo de ejecución, pero no en tiempo de desarrollo o compilación. Sin embargo, resulta más amplio en el sentido de que puede contener más clases de elementos. Se establecen entre los nodos relaciones que significan que existe comunicación entre estos. Se representan mediante líneas continuas, y se puede hacer con un estereotipo que indica el tipo de comunicación (Falgueras, 2005). La arquitectura física del sistema está diseñada por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software.

#### Descripción de los dispositivos de hardware:

**PC\_Cliente:** está destinada a las funciones del usuario.

**Servidor Web:** es el encargado de atender las peticiones de los usuarios y proporcionar la respuesta.

**Servidor de Base de Datos:** almacena toda la información que se gestiona en la institución.

**Escáner:** encargada de realizar el proceso de escaneo de los documentos.



## Capítulo 4: Implementación y prueba

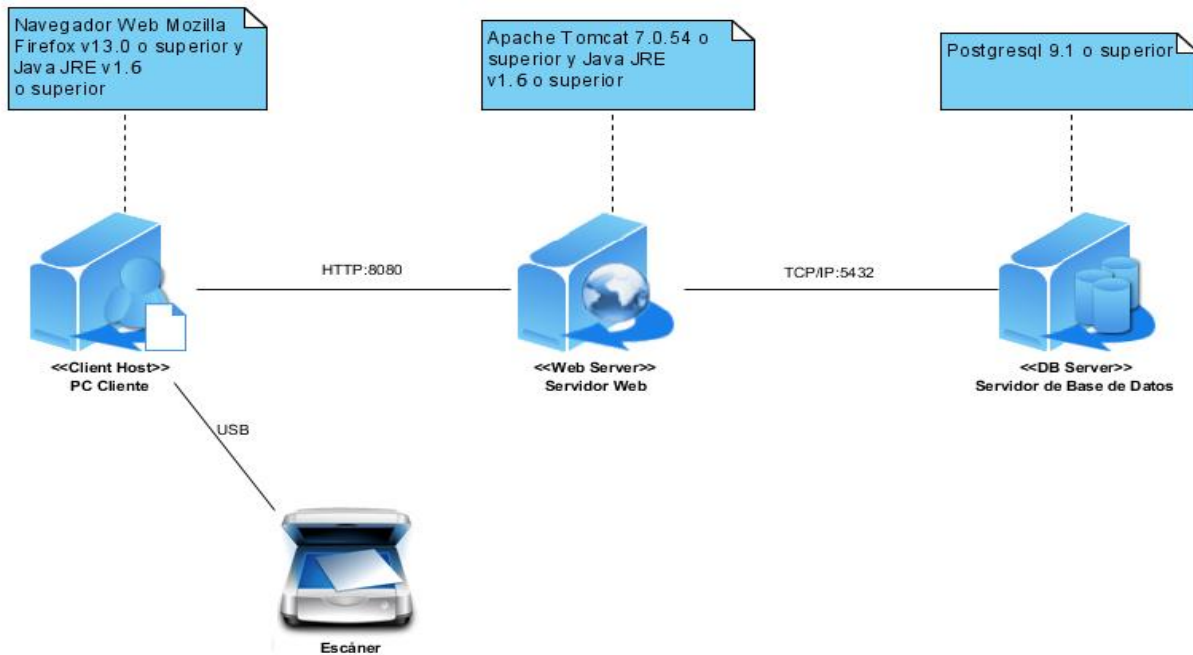


Figura 20: Diagrama de despliegue.

### 4.3. Pruebas del sistema

La prueba del software es un elemento de un tema más amplio que suele denominarse verificación y validación. Verificación es el conjunto de actividades que aseguran que el software implemente correctamente una función específica. Validación es un conjunto diferente de actividades que aseguran que el software construido corresponde con los requisitos del cliente. Las pruebas son el último bastión para la evaluación de la calidad y, de manera más pragmática, el descubrimiento de errores. (Pressman, 2005)

#### 4.3.1. Métodos de prueba

Cualquier producto de ingeniería puede ser probado de una de estas formas:

**Pruebas de Caja Negra:** conociendo la funcionalidad específica para la cual fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y se realizan sobre la interfaz del software. Los artefactos generados en las pruebas de caja negra son los casos de pruebas. Siempre es ejecutada como una unidad independiente, desde el comienzo hasta el final. (Pressman, 2005)

**Pruebas de Caja Blanca:** conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que “todas las piezas encajen”, o sea, que la operación interna se ajusta

## Capítulo 4: Implementación y prueba

---

a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar. (Pressman, 2005)

### 4.3.2. Tipos de pruebas

**Funcionalidad:** validación de las funciones, asegurando que el sistema es accedido sólo por los actores deseados.

- **Función:** pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.
- **Seguridad:** asegurar que los datos o el sistema solamente es accedido por los actores deseados.

**Usabilidad:** prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitivamente al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

**Fiabilidad:** integridad y respuesta bajo condiciones anormales.

- **Integridad:** enfocada a la valoración de la robustez (resistencia a fallos).
- **Stress:** enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponibles). (Pressman, 2005)

### 4.3.3. Estrategias de pruebas

La estrategia de prueba toma un enfoque incremental de las pruebas, inicia con la prueba de unidades individuales del programa, pasa a pruebas diseñadas para facilitar la integración de las unidades, y culmina con pruebas que se realizan sobre el sistema construido. Describe el enfoque y los objetivos generales de las actividades de prueba. Incluye los niveles de prueba (unidad, integración, etc.), el tipo de prueba a ser ejecutada (funcional, stress, etc.) y los casos de prueba diseñados para lograr los objetivos. (Pressman, 2005)

## Capítulo 4: Implementación y prueba

### Pruebas de unidad

Se concentra en el esfuerzo de verificación de la unidad más pequeña de diseño del software: el componente o módulo de software. Tomando como guía la descripción del diseño a nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del módulo. Las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente. (Pressman, 2005). Durante el proceso de desarrollo, es posible probar el estado de la aplicación mediante el comando: **grails test-app**. Este ejecuta las pruebas unitarias y genera un reporte en texto y HTML, al cual será posible recurrir en caso de existir algún fallo en la aplicación.

Para crear un test unitario de un servicio se ejecuta el script:

```
grails create-unit-test <nombre de la prueba>
```

A continuación se muestra el informe de resultados de las pruebas unitarias generado por Grails.

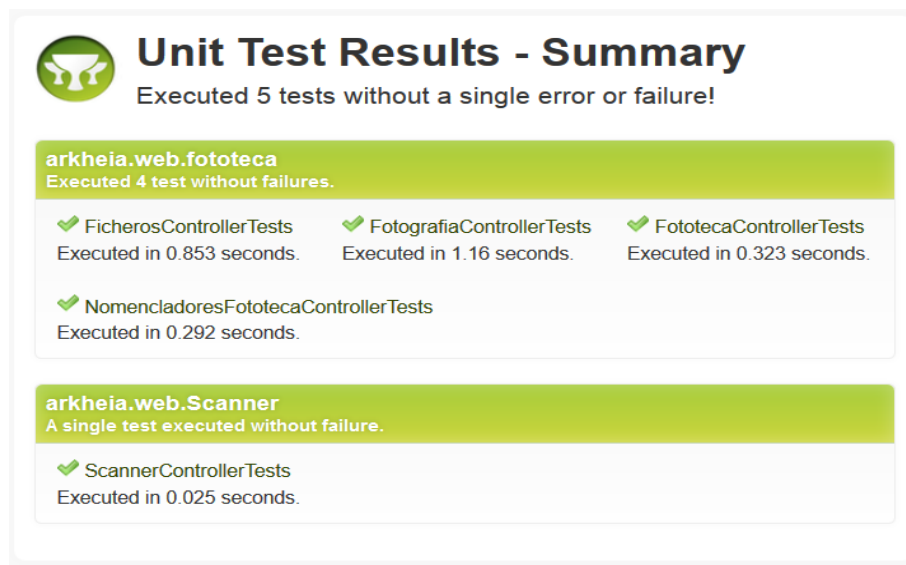


Figura 21: Informe de resultados de las pruebas de unidad.

### Casos de pruebas

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final. A continuación se muestran los resultados de las pruebas de caja negra

## Capítulo 4: Implementación y prueba

realizadas al módulo por el grupo de Calidad UCI mediante un gráfico que relaciona las no conformidades detectadas por cada iteración.

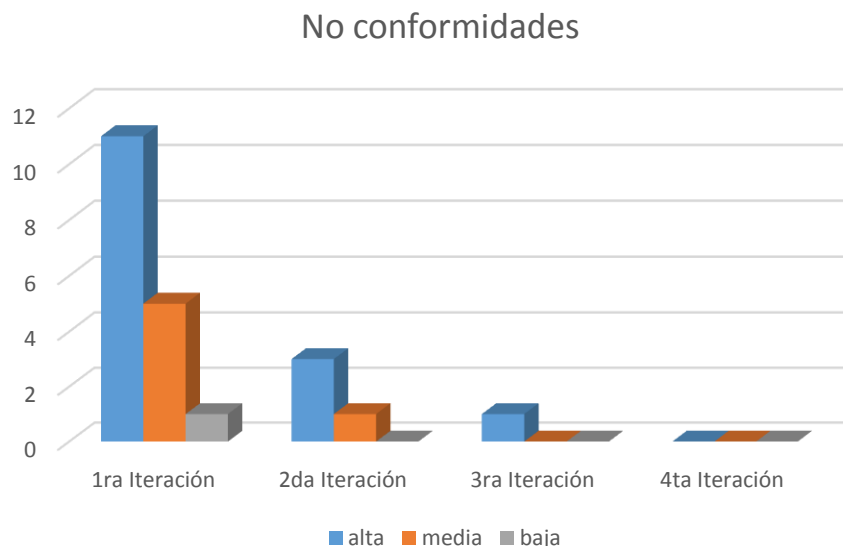


Figura 22: Resultados de las pruebas de Caja negra.

Para la elaboración de los casos de prueba se usó la siguiente estructura:

### Descripción general

[Referencia cruzada al caso de uso o requisito correspondiente]

### Condiciones de ejecución

[Precondiciones del caso de uso o requisito]

### SC <Nombre de la sección>

Escenario	Descripción	Variable 1	Variable 2	Variable n	Respuesta del sistema	Flujo central
EC 1.1 [Nombre del escenario]	[Descripción del escenario de prueba]	V	V	V	[Se escribe el resultado que se espera al realizar la prueba.]	[Pasos a desarrollar para probar la funcionalidad que se indicó.]
		dato	dato	dato		
		V	V	V		
EC 1.n [Nombre del escenario]	[Descripción del escenario de prueba]	NA	NA	NA	[Se escribe el resultado que se espera al realizar la prueba.]	[Pasos a desarrollar para probar la funcionalidad que se indicó.]

[Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.]

## Capítulo 4: Implementación y prueba

A continuación se muestra como ejemplo el caso de prueba Ver detalles de descripción de fotografía:

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar descripción de fotografía	Muestra el listado de las descripciones de fotografía y permite ver los detalles o asociar ficheros.	Muestra el listado de las descripciones de fotografía.	1. Seleccionar la opción "Mostrar descripciones" del menú principal.
EC 1.2 Asociar ficheros	Permite asociar ficheros a una descripción de fotografía.	Asocia los ficheros a la descripción de fotografía.	1. Seleccionar la opción "Mostrar descripciones" del menú principal. 2. Seleccionar la opción "Asociar ficheros". 3. <i>Ver Caso de Prueba CIGED_ARKHEIA-OAHCE-FASE III_DCP_Gestionar Ficheros del Documento SC8 Asociar ficheros de documento de fotografía.</i>
EC 1.3 Filtrar descripción	Permite filtrar las descripciones de fotografía	Muestra en la tabla las descripciones de fotografía que coinciden con los filtros seleccionados.	1. <i>Ver CIGED_ARKHEIA-OAHCE-FASE III_DCP_Filtrar descripción de fotografía.</i>
EC 1.4 Ver detalles	Permite ver los detalles de una descripción de fotografía	Muestra los detalles de la descripción de fotografía seleccionada.	1. <i>Ver CIGED_ARKHEIA-OAHCE-FASE III_DCP_Ver detalles de descripción fotografía</i>

Figura 23: Caso de prueba Ver detalles de descripción de fotografía.

### Conclusiones del capítulo

En este capítulo fueron representados y detallados los artefactos generados en la implementación del módulo. A partir del diagrama de componentes se realizó un recorrido por los elementos del sistema basados en su clasificación en paquetes y se estructuró la distribución física de la arquitectura para la representación del diagrama de despliegue. Para comprobar la eficacia de las tareas de implementación y con el objetivo de garantizar el cumplimiento de los requisitos definidos, se realizaron pruebas al sistema comprobando su funcionalidad y arrojando resultados satisfactorios debido a que fueron resueltas todas las no conformidades existentes y las pruebas de unidad fueron exitosas.

### Conclusiones generales

Con la culminación del presente trabajo de diploma, se desarrolló el módulo Procesamiento de fotografía del XABAL Arkheia 2.1 para la Oficina de Asuntos Históricos del Consejo de Estado, agilizando los procesos de descripción, digitalización y almacenamiento de los archivos fotográficos bajo la jurisdicción de la institución.

Los resultados obtenidos permiten concluir que:

- A través del desarrollo del marco conceptual se establecieron los principales aspectos teóricos para la fundamentación de la investigación.
- El estudio realizado acerca de los protocolos de comunicación con el escáner, permitió seleccionar los estándares TWAIN y SANE por ser los que más se ajustan a las necesidades del sistema.
- Las actividades del diseño del módulo Procesamiento de fotografía fueron ejemplificadas a través de diagramas de clases y de secuencia, cumpliendo así con los objetivos propuestos.
- Se implementó el módulo Procesamiento de fotografía del sistema XABAL Arkheia a partir de los requerimientos de software establecidos con el cliente y haciendo uso de la arquitectura y las tecnologías definidas por el proyecto.
- Las pruebas realizadas demostraron que las funcionalidades del sistema tienen correspondencia con las definidas inicialmente.

## Recomendaciones

Con el propósito de enriquecer la propuesta de solución realizada, se recomienda:

- Realizar un estudio acerca de algoritmos que mejoren el rendimiento del sistema para el trabajo con imágenes de alta resolución.
- Mejorar la interfaz del *Data Source Manager* (DSM) del componente de digitalización.
- Agregar nuevos parámetros de edición de imagen al componente de digitalización.
- Mejorar la seguridad del componente de digitalización.

### Referencias

**Adapting sl. 2013.** Adapting, content software for smart companies. *Adapting, content software for smart companies*. [En línea] Adapting sl, 2013. [Citado el: 3 de 2 de 2015.] <http://www.adapting.com/>.

**Addoc. 2008.** Addoc, administración de documentos S.A. *Addoc, administración de documentos S.A.* [En línea] 2008. [Citado el: 3 de 2 de 15.] <http://li685-159.members.linode.com/>.

**Almaguer, Adrián Cid. 2008.** *Concepción del sistema de gestión de documentos de archivos históricos ArchiVen-His*. La Habana : s.n., 2008.

**Apache Foundation. APACHE SOFTWARE FOUNDATION. APACHE SOFTWARE FOUNDATION.** [En línea] [Citado el: 25 de 04 de 2015.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.

**Barclay, Kenneth y Savage, John . 2006.** *Groovy Programming. An Introduction for Java Developers*. Napier University, Edinburgh, United Kingdom : MORGAN KAUFMANN, 2006. 978-0-12-372507-3.

**Bennet, Michael J y Wheeler, F.Barry. 2010.** *Raw as Archival Still Image Format: A Consideration*. s.l. : UConn Libraries Published Works, 2010.

**Bibeault, Bear y Katz, Yehuda . 2008.** *jQuery in Action*. s.l. : B&W, 2008. 1933988355.

**Brito, Nacho. 2009.** *Manual de desarrollo web con Grails*. Madrid : Safe Creative, 2009. 978-84-613-2651.

**Cabrera Gonzalez, Lianet y Pompa Torres, Enrique Roberto. 2012.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información*. La Habana : s.n., 2012.

**Carnicer Arribas, Dolores, y otros. 2011.** *Recomendaciones para la digitalización de documentos en los archivos*. Valladolid : Archivo General de Castilla y León, 2011.

**Carnicer, María Dolores, Fernández, Blanca Desantes y López, Guadalupe Moreno. 2000.** *ISAD(G): Norma Internacional General de Descripción Archivística*. Madrid : Consejo Intrenacional de Archivos, 2000.

**Cochran, David . 2012.** *Twitter Bootstrap Web Development How-To*. 2012. 9781849518826.

**Debian Project. 2013.** SANE. *SANE*. [En línea] Debian Project, 30 de 9 de 2013. [Citado el: 2 de 2 de 2015.] <http://www.sane-project.org/intro.html>.

**Definición. De. 2012.** Definición. De. *Definición. De*. [En línea] 2012. [Citado el: 20 de 1 de 2015.] <http://definicion.de/lenguaje-de-programacion/>.



## Referencias bibliográficas

---

**Díez Carrera, Carmen. 1998.** *Los materiales especiales en las bibliotecas.* 1998. ISBN: 84-95178-09-5.

**Dynamsoft Corporation. 2015.** TWAIN Scanning. *TWAIN Scanning.* [En línea] 2015. [Citado el: 7 de 5 de 2015.] <http://twainsoft.com/>.

**Egdamar. 2012.** Diseño UML. *Diseño UML.* [En línea] 2012. [Citado el: 21 de 3 de 2014.] <http://egdamar877.blogspot.com/2009/05/expocicion.html>.

**EMC Corporation. 2009.** EMC Captiva. *EMC Captiva.* [En línea] EMC Corporation, 2009. [Citado el: 2 de 2 de 2015.] [http://suppliers.jimtrade.com/147/146878/image\\_scanner\\_interface\\_specificationisis\\_scanner\\_driver.htm](http://suppliers.jimtrade.com/147/146878/image_scanner_interface_specificationisis_scanner_driver.htm).

**Encyclopedia Britannica. 2015.** Encyclopedia Britannica. *Encyclopedia Britannica.* [En línea] Encyclopædia Britannica, Inc., 2015. [Citado el: 5 de 2 de 2015.] <http://www.britannica.com/EBchecked/topic/410357/protocol>.

**Escalona, José María y Koch, Nora . 2002.** *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo.* Sevilla : s.n., 2002.

**Falgueras, Benet Campderrich. 2005.** *Ingeniería del software.* Barcelona : Editorial UOC, de esta edición Aragón, 182, 08011 Barcelona, 2005.

**Fergarcia. 2013.** Entorno de Desarrollo Integrado (IDE). *Entorno de Desarrollo Integrado (IDE).* [En línea] 2013. [Citado el: 21 de 1 de 2015.] <http://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

**Foix, Laia. 2003.** *La gestión de fondos fotográficos en entidades no comerciales.* Barcelona : s.n., 2003.

**Fuster Ruiz, Francisco. 1999.** *Archivística, archivo y archivos de documentos.* Murcia : Servicio de publicaciones, 1999. págs. 103-120. Vol. 2.

**Gnome Ltd. 2013.** Gnome. *Gnome.* [En línea] Gnome Ltd, 3 de noviembre de 2013. [Citado el: 7 de 2 de 2015.] <http://www.gnome.sk/index.html>.

**Gustavo Suhit. 2008.** Informateando. *Informateando.* [En línea] Creative Common, 5 de noviembre de 2008. [Citado el: 5 de 2 de 2015.]

**Gutierrez, Demián. 2009.** *UML diagramas de Paquetes.* Mérida : Universidad de Los Andes, 2009.

**IEEE. 1999.** IEEE Advancing Technology for Humanity. *IEEE Advancing Technology for Humanity.* [En línea] 1999. [Citado el: 20 de 1 de 2015.] <http://www.ieee.org>.

**Itseez. 2014.** OpenCV. *OpenCV.* [En línea] Itseez., 11 de 11 de 2014. [Citado el: 4 de 2 de 2015.] <http://opencv.org/>.

## Referencias bibliográficas

---

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000. 84-7829-036-2.

**James Rumbaugh, Ivar Jacobson, Grady Booch. 2000.** *El lenguaje unificado de modelado*. s.l. : Addison Wesley, 2000.

**Kernel Doc s.l. 2007.** Kernel Doc. *Kernel Doc*. [En línea] Kernel Doc s.l., 2007. [Citado el: 1 de 2 de 2015.] <http://www.kerndoc.es/>.

**Lab Asprise. 2012.** Asprise.com. *Asprise.com*. [En línea] Lab Asprise, 2012. [Citado el: 4 de 2 de 2015.] <http://asprise.com/product/jsane>.

**Langford, Michael. 2004.** *La fotografía paso a paso*. s.l. : Hermann Blume Ediciones, 2004. ISBN 84-87756-01-8.

**Larman, Craig. 2003.** *UML y Patrones*. s.l. : Prentice Hall, 2003. 9788420534381.

**Map, Elvia Leticia William. 2012.** *Propuesta para la Digitalización del Fondo Documental del Dr. Belisario Porras*. Sevilla : Universidad Internacional de Andalucía, 2012.

**Microsoft. 2014.** Windows. *Windows*. [En línea] Microsoft, 2014. [Citado el: 6 de 2 de 2015.] <http://msdn.microsoft.com/en-us/library/windows/hardware/ff542835.aspx>.

**Mozilla. 2015.** MDN. *Mozilla Developer Network*. [En línea] 26 de 04 de 2015. [Citado el: 1 de 05 de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.

**Olson, Nancy B. 2008.** *Cataloging of Audiovisual Materials and Other Special Materials: A Manual Based on AACR2 and MARC 21*. Universidad de Michigan : Libraries Unlimited, 2008. ISBN 9781591586357.

**Oracle Corporation. 2013.** Oracle. *Oracle*. [En línea] Oracle, 2013. [Citado el: 20 de 1 de 2015.] <http://www.oracle.com/technetwork/java>.

**Orozco González, Delio G. y Fernández Bertot, Víctor . 2006.** *Papiro. Un sistema de conservación, digitalización, gestión y socialización de información documental para los*. 2006.

**P.Briol. 2012.** *BPMN, the Business Process Modeling Notation Pocket Handbook*. 2012. 978-1-4092-0299-8.

**Peis, Eduardo y Ruiz-Rodríguez, Antonio A. 2004.** EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones. *HIPERTEXT.NET*. [En línea] 2004. [Citado el: 25 de 2 de 2015.] <http://www.upf.edu/hipertextnet/numero-2/ead.html>.

**PostgreSQL. 2015.** PostgreSQL. *PostgreSQL*. [En línea] 21 de enero de 2015. <http://www.postgresql.org/about/news/1481/>.

**Pressman, Roger S. 2005.** *Ingeniería del software. Un enfoque práctico*. s.l. : MCGRAW-HILL, 2005. 9789701054734.

## Referencias bibliográficas

---

**Sixto, Leandro Roura. 2014.** *Arquitectura de software. Guía base. Proyecto Sistema de Gestión de Archivos Históricos (XABAL-ARKHEIA 2.1) para la OAHCE. Fase III.* Universidad de las Ciencias Informáticas : s.n., 2014.

**TortoiseSVN team.** Tortoise.net. *TortoiseSVN.* [En línea] Tortoise. [Citado el: 25 de 04 de 2015.] <http://tortoisesvn.net/>.

**TWAIN Working Group . 2012.** TWAIN. *TWAIN.* [En línea] 16 de febrero de 2012. [Citado el: 2 de 2 de 2015.] [www.twain.org](http://www.twain.org).

## Bibliografía

**Adapting sl. 2013.** Adapting, content software for smart companies. *Adapting, content software for smart companies*. [En línea] Adapting sl, 2013. [Citado el: 3 de 2 de 2015.] <http://www.adapting.com/>.

**Addoc. 2008.** Addoc, administración de documentos S.A. *Addoc, administración de documentos S.A.* [En línea] 2008. [Citado el: 3 de 2 de 15.] <http://li685-159.members.linode.com/>.

**Almaguer, Adrián Cid. 2008.** *Concepción del sistema de gestión de documentos de archivos históricos ArchiVen-His*. La Habana : s.n., 2008.

**Altova. 2013.** Altova. *Altova*. [En línea] 2013. [Citado el: 22 de 3 de 2015.] <http://www.altova.com/es/umodel/uml-package-diagrams.html>.

**Álvarez de Zayas, Dr. Carlos. 1995.** *Metodología de la investigación científica*. Santiago de Cuba : s.n., 1995.

**Apache Foundation.** APACHE SOFTWARE FOUNDATION. *APACHE SOFTWARE FOUNDATION*. [En línea] [Citado el: 25 de 04 de 2015.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.

**Barclay, Kenneth y Savage, John . 2006.** *Groovy Programming. An Introduction for Java Developers*. Napier University, Edinburgh, United Kingdom : MORGAN KAUFMANN, 2006. 978-0-12-372507-3.

**Bennet, Michael J y Wheeler, F.Barry. 2010.** *Raw as Archival Still Image Format: A Consideration*. s.l. : UConn Libraries Published Works, 2010.

**Bibeault, Bear y Katz, Yehuda . 2008.** *jQuery in Action*. s.l. : B&W, 2008. 1933988355.

**Boadas, Joan, Casellas, Lluís Esteve y Suquet, M. Angels. 2001.** *Manual para la gestión de fondos y colecciones fotográficas*. s.l. : Biblioteca de la imagen, 2001. ISBN:84-95483-11-4.

**Brito, Nacho. 2009.** *Manual de desarrollo web con Grails*. Madrid : Safe Creative, 2009. 978-84-613-2651.

**Cabrera Gonzalez, Lianet y Pompa Torres, Enrique Roberto. 2012.** *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información*. La Habana : s.n., 2012.

**Carlos, Universidad Rey Juan. 2013.** *Arquitecturas Software*. España : s.n., 2013.

**Carnicer Arribas, Dolores, y otros. 2011.** *Recomendaciones para la digitalización de documentos en los archivos*. Valladolid : Archivo General de Castilla y León, 2011.

**Carnicer, María Dolores, Fernández, Blanca Desantes y López, Guadalupe Moreno. 2000.** *ISAD(G): Norma Internacional General de Descripción Archivística*. Madrid : Consejo Intrenacional de Archivos, 2000.

**Cochran, David . 2012.** *Twitter Bootstrap Web Development How-To*. 2012. 9781849518826.

**David Garlan, Mary Shaw. 1994.** *An Introduction to Software Architecture*. 1994.

**Debian Project. 2013.** SANE. *SANE*. [En línea] Debian Project, 30 de 9 de 2013. [Citado el: 2 de 2 de 2015.] <http://www.sane-project.org/intro.html>.

**Definición. De. 2012.** Definición. De. *Definición. De*. [En línea] 2012. [Citado el: 20 de 1 de 2015.] <http://definicion.de/lenguaje-de-programacion/>.

**Díez Carrera, Carmen. 1998.** *Los materiales especiales en las bibliotecas*. 1998. ISBN: 84-95178-09-5.

**Dirección de Archivos Estatales. 1993.** *Diccionario de la Terminología Archivística*. Madrid : Ministerio de Cultura, 1993. 84-8181-066-5.

**Documails. 2007.** Documails Express. *Documails Express*. [En línea] Documails, 2007. [Citado el: 2 de 2 de 2015.] <http://www.documalis.com>.

**Dynamsoft Corporation. 2015.** TWAIN Scanning. *TWAIN Scanning*. [En línea] 2015. [Citado el: 7 de 5 de 2015.] <http://twainscanning.com/>.

**Egdamar. 2012.** Diseño UML. *Diseño UML*. [En línea] 2012. [Citado el: 21 de 3 de 2014.] <http://egdamar877.blogspot.com/2009/05/expocicion.html>.

**EMC Corporation. 2009.** EMC Captiva. *EMC Captiva*. [En línea] EMC Corporation, 2009. [Citado el: 2 de 2 de 2015.] [http://suppliers.jimtrade.com/147/146878/image\\_scanner\\_interface\\_specificationisis\\_scanner\\_driver.htm](http://suppliers.jimtrade.com/147/146878/image_scanner_interface_specificationisis_scanner_driver.htm).

**Encyclopedia Britannica. 2015.** Encyclopedia Britannica. *Encyclopedia Britannica*. [En línea] Encyclopædia Britannica, Inc., 2015. [Citado el: 5 de 2 de 2015.] <http://www.britannica.com/EBchecked/topic/410357/protocol>.

**Escalona, José María y Koch, Nora . 2002.** *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. Sevilla : s.n., 2002.

**Falgueras, Benet Campderrich. 2005.** *Ingeniería del software*. Barcelona : Editorial UOC, de esta edición Aragón, 182, 08011 Barcelona, 2005.

**Fergarcia. 2013.** Entorno de Desarrollo Integrado (IDE). *Entorno de Desarrollo Integrado (IDE)*. [En línea] 2013. [Citado el: 21 de 1 de 2015.] <http://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.

- Foix, Laia. 2003.** *La gestión de fondos fotográficos en entidades no comerciales.* Barcelona : s.n., 2003.
- Fuster Ruiz, Francisco. 1999.** *Archivística, archivo y archivos de documentos.* Murcia : Servicio de publicaciones, 1999. págs. 103-120. Vol. 2.
- Gnome Ltd. 2013.** Gnome. *Gnome.* [En línea] Gnome Ltd, 3 de noviembre de 2013. [Citado el: 7 de 2 de 2015.] <http://www.gnome.sk/index.html>.
- Gustavo Suhit. 2008.** Informateando. *Informateando.* [En línea] Creative Common, 5 de noviembre de 2008. [Citado el: 5 de 2 de 2015.]
- Gutierrez, Demián. 2009.** *UML diagramas de Paquetes.* Mérida : Universidad de Los Andes, 2009.
- Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar. 2006.** *Metodología de la investigación.* s.l. : Mc Graw Hill, 2006. ISBN 970-10-5753-8.
- IEEE. 1999.** IEEE Advancing Technology for Humanity. *IEEE Advancing Technology for Humanity.* [En línea] 1999. [Citado el: 20 de 1 de 2015.] <http://www.ieee.org>.
- Itseez. 2014.** OpenCV. *OpenCV.* [En línea] Itseez., 11 de 11 de 2014. [Citado el: 4 de 2 de 2015.] <http://opencv.org/>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000. 84-7829-036-2.
- Jalón, Javier García de. 2000.** *Aprenda Java como si estuviera en primero.* San Sebastian : UNIVERSIDAD DE NAVARRA, 2000.
- James Rumbaugh, Ivar Jacobson, Grady Booch. 2000.** *El lenguaje unificado de modelado.* s.l. : Addison Wesley, 2000.
- Kernel Doc s.l. 2007.** Kernel Doc. *Kernel Doc.* [En línea] Kernel Doc s.l., 2007. [Citado el: 1 de 2 de 2015.] <http://www.kerneldoc.es/>.
- Koenig, Dierk , y otros. 2007.** *Groovy in Action.* s.l. : B&W, 2007. 9781932394849.
- Lab Asprise. 2012.** Asprise.com. *Asprise.com.* [En línea] Lab Asprise, 2012. [Citado el: 4 de 2 de 2015.] <http://asprise.com/product/jsane>.
- Langford, Michael. 2004.** *La fotografía paso a paso.* s.l. : Hermann Blume Ediciones, 2004. ISBN 84-87756-01-8.
- Larman, Craig. 2003.** *UML y Patrones.* s.l. : Prentice Hall, 2003. 9788420534381.
- Lozano, Rita García. 2010.** El archivo de documentos. El acceso a la información contenida en los archivos públicos. Ciudad Real : Junta de Comunidades de Castilla-La Mancha. Consejería de Administraciones Públicas., 2010.

**Map, Elvia Leticia William. 2012.** *Propuesta para la Digitalización del Fondo Documental del Dr. Belisario Porras.* Sevilla : Universidad Internacional de Andalucía, 2012.

**Merson, Paulo. 2009.** *Data Model as an Architectural View.* 2009.

**Microsoft. 2014.** Windows. *Windows.* [En línea] Microsoft, 2014. [Citado el: 6 de 2 de 2015.] <http://msdn.microsoft.com/en-us/library/windows/hardware/ff542835.aspx>.

**Mozilla. 2015.** MDN. *Mozilla Developer Network.* [En línea] 26 de 04 de 2015. [Citado el: 1 de 05 de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.

**Neyem, Andrés. 2008.** *INSTRUCTIVO PARA EL MODELADO CON CASOS DE USO.* 2008.

**Olson, Nancy B. 2008.** *Cataloging of Audiovisual Materials and Other Special Materials: A Manual Based on AACR2 and MARC 21.* Universidad de Michigan : Libraries Unlimited, 2008. ISBN 9781591586357.

**Oracle Corporation. 2013.** Oracle. *Oracle.* [En línea] Oracle, 2013. [Citado el: 20 de 1 de 2015.] <http://www.oracle.com/technetwork/java>.

**Orozco González, Delio G. y Fernández Bertot, Víctor . 2006.** *Papiro. Un sistema de conservación, digitalización, gestión y socialización de información documental para los.* 2006.

**P.Briol. 2012.** *BPMN, the Business Process Modeling Notation Pocket Handbook.* 2012. 978-1-4092-0299-8.

**Paul Clements, Judith Stafford y otros. 2003.** *Documenting Software Architectures in an Agile World.* 2003.

**Paul Clements, Rick Kazman, Mark Klein. 2001.** *Evaluating a Software Architecture .* 2001.

**Peis, Eduardo y Ruiz-Rodríguez, Antonio A. 2004.** EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones. *HIPERTEXT.NET.* [En línea] 2004. [Citado el: 25 de 2 de 2015.] <http://www.upf.edu/hipertextnet/numero-2/ead.html>.

**PostgreSQL. 2015.** PostgreSQL. *PostgreSQL.* [En línea] 21 de enero de 2015. <http://www.postgresql.org/about/news/1481/>.

**Pressman, Roger S. 2005.** *Ingeniería del software. Un enfoque práctico.* s.l. : MCGRAW-HILL, 2005. 9789701054734.

**Real Academia Española. 2012.** Real Academia Española. *Diccionario de la Lengua Española 22. edición.* [En línea] Real Academia Española, 2012. [Citado el: 28 de Enero de 2015.] <http://buscon.rae.es/drae/>.

**Reynoso, Billy. 2012.** *Architect Academy.* Buenos Aires : s.n., 2012.

**Sheyla García, Felix Gonzalez. 2013.** *Extensión del NetBeans IDE para el diseño de Interfaces. Gráficas de Usuario con Ext JS.* La Habana : s.n., 2013.



## Bibliografía

---

**Sixto, Leandro Roura. 2014.** *Arquitectura de software. Guía base. Proyecto Sistema de Gestión de Archivos Históricos (XABAL-ARKHEIA 2.1) para la OAHCE. Fase III.* Universidad de las Ciencias Informáticas : s.n., 2014.

**Sommerville, Ian. 2005.** *Ingeniería de Software (Séptima edición).* 2005.

**TortoiseSVN team.** Tortoise.net. *TortoiseSVN.* [En línea] Tortoise. [Citado el: 25 de 04 de 2015.] <http://tortoisesvn.net/>.

**TWAIN Working Group . 2012.** TWAIN. *TWAIN.* [En línea] 16 de febrero de 2012. [Citado el: 2 de 2 de 2015.] [www.twain.org](http://www.twain.org).



## Glosario de términos

**Delaminación:** proceso de separación de dos o más capas de cualquier material compuesto.

**Emulsión:** mezcla formada por dos líquidos inmiscibles entre sí que se encuentran uniformemente distribuidos uno en el otro en forma de pequeñas gotas, termodinámicamente inestables.

**Escáner:** dispositivo que permite convertir los caracteres escritos o gráficos impresos en un formato digital, compuesto por un lenguaje binario de 0 y 1 comprensible por la computadora.

**Hemeroteca:** edificio o sala en el que se tiene guardado y ordenado un conjunto de diarios, revistas y otras publicaciones periódicas para que el público pueda leerlos o consultarlos.

**Interoperabilidad:** habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

**Metadatos:** datos altamente estructurados que describen información. Es "información sobre información" o "datos sobre los datos", ya que proporciona información mínima necesaria para identificar un recurso e incluye información descriptiva sobre el contexto, calidad y condición o características de un dato u objeto que tiene la finalidad de facilitar la recuperación, autenticación, evaluación, preservación e interoperabilidad.

**Modularidad:** es la capacidad que tiene un sistema de ser estudiado, visto o entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo. Cada una de esas partes en que se encuentre dividido el sistema recibe el nombre de módulo.

**Papel baritado:** el papel baritado es un tipo de papel para el positivado de blanco y negro, cubierto por una capa blanca de sulfato de bario (también conocido como *barita*) y esta cubierta a su vez por la emulsión sensible a la luz.

**Paleografía:** es la ciencia que se encarga de descifrar las escrituras antiguas y estudiar su evolución, así como datar, localizar y clasificar los diferentes testimonios gráficos.

**Periférico:** nombrado tanto a las unidades o dispositivos a través de los cuales el ordenador se comunica con el mundo exterior, como a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.

**Pixel:** un píxel o pixel, plural píxeles (acrónimo del inglés *picture element*, "elemento de imagen") es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

**Transcripción:** es una adaptación mediante caracteres gráficos de una expresión de otra lengua o de un habla peculiar no estándar usando la ortografía y convenciones gráficas de otra lengua o la lengua estándar.