

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



Componente para la asociación dinámica de metadatos de la plataforma SuiteDoc.

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

Autora: Yudier Pereira Salazar

Tutora: Ing. Yadira Lizama Mué

Ciudad de La Habana, 2012

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

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

Yudier Pereira Salazar

(Autor)

Ing. Yadira Lizama Mué

(Tutora)

AGRADECIMIENTOS

A mi mamá y a mi papá porque a ellos le debo mi vida y les debo todo lo que soy, por su confianza y por siempre hacerme sentir que puedo lograr todo lo que me proponga en la vida.

A mi tutora Yadira, por toda la ayuda que me dio desde el momento en que nos pusieron a trabajar juntos en el proyecto RAP hasta que culmine mi tesis y por llevarme a Venezuela.

A mi tío Omar por toda la ayuda que me ha brindado durante estos 5 años y por atenderme como si fuera un hijo más para él.

Al tribunal por la confianza que depositaron en mí, y por siempre darme una oportunidad para seguir adelante cuando las cosas estaban más complicadas.

Al oponente por la atención que me prestó y por responderme todos mis correos, por la paciencia y por la confianza.

A Dayami por ser mi amiga incondicional durante estos últimos años y por toda la ayuda que me brindó con la tesis, sin ella no lo hubiera logrado.

Al Cuadra y a Daniel, por toda la ayuda que me brindaron en la parte de la implementación y por la paciencia que tuvieron con respecto a la cantidad de veces que los molesté, que sé que estuve a punto de volverlos locos.

A Leyandry por ser mi gran amigo durante estos 5 años, por cumplir años el mismo día que yo, y por escucharme cada vez que iba a llorar penas a su cuarto.

A Raúl y Amílcar, que a pesar de que no me ayudaron en nada, mucho apoyo moral sí me dieron, además de ser 2 muy buenos amigos y eso sí es realmente lo importante.

A Sojo, que a pesar de pasarnos las 24 horas del día fajados, ha sido mi amigo y me ha ayudado bastante desde que comenzamos a estudiar juntos.

A la gente de mi vieja aula: Luis Ángel, Edriel, Alejandro, El Chiki, Jorge, Dayana y Amalia, que a pesar de que nos separaron de grupo siempre supimos mantuvieron conmigo muy buenas relaciones y me ayudaron siempre que lo necesité.

A la gente de mi aula nueva, no voy a mencionar los nombres porque son muchos pero todos supieron ser muy buenos amigos conmigo y los tendré siempre presentes.

A mis amigos de 4ta: El titi, Duvergel, Josué, Miguel Ángel, el Bati y Ricardo, porque se han vuelto casi como mi familia aquí en la UCI y porque pase muy buenos momentos junto a ellos.

A Ailyn, que a pesar de que a cada rato me volvía loco y la quería matar, con los años llegue hasta a cogerle cariño porque muy pero muy en el fondo, ella es muy buena persona.

A José Ángel y Sergio por la ayuda que me dieron en un momento que bastante la necesitaba, ellos fueron el motor que me impulsó y me permitió realizar mi tesis.

AGRADECIMIENTOS

A los profesores que estuvieron conmigo en Venezuela, por toda la ayuda que me dieron, y por enseñarme a comprar que yo realmente no sabía, y por los buenos momentos que pasamos juntos allá, que ojala y se repita.

A mi piquete del dotaaaaa, que para mí son los mejores, quitando a Raúl que es malísimo, y que gracias a eso realmente pasamos mucho tiempo juntos, y nos permitió crear grandes lazos de amistad.

Y a mí, definitivamente a mí, que he realizado un esfuerzo bien grande en estos últimos meses jajaja.

RESUMEN

El desarrollo de sistemas que contribuyen a la conservación y preservación de archivos documentales ha alcanzado gran auge en la actualidad. El presente trabajo tiene su aplicación práctica en la plataforma de desarrollo de Sistemas para la Digitalización del Centro CEGEL (Centro de Gobierno Electrónico), adscrito a la Universidad de las Ciencias Informáticas y dedicado a la producción de software para el gobierno electrónico en Cuba. Dicha plataforma permite la integración de componentes de software en función de la automatización de los procesos del Centro de Digitalización para garantizar la obtención de imágenes digitales con valor legal a partir de la digitalización de fondos documentales. La presente investigación tiene como objetivo principal desarrollar un componente que permita la identificación y asociación dinámica de metadatos a partir de documentos digitales con valor legal. Se analizan los conceptos teóricos relacionados a la investigación, las principales técnicas de asociación de metadatos y las herramientas utilizadas para el desarrollo del sistema. Se realizan las actividades correspondientes a la metodología XP para el desarrollo del software, generando los artefactos necesarios. Se valida el componente de software obtenido a partir de la realización de pruebas que certifican la calidad del mismo. Como resultado se obtiene el desarrollo de un componente de asociación dinámica de metadatos que será incorporado a la plataforma de desarrollo de sistemas para la digitalización del Centro CEGEL que agilice el proceso de configuración de metadatos durante la creación de sistemas de este tipo.

Palabras claves: componente, digitalización, metadatos.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.4 Metadatos.....	8
La asociación de metadatos en sistemas de digitalización de documentos.....	11
1.5.1 QuickScan Pro.....	11
1.5.2 Sistema de Digitalización de Documentos (SDD)	12
1.6 Soluciones nacionales de referencia	14
1.6.1 Papiro	14
1.6.2 Digipyrus	15
1.6.3 Alfabéticas.....	15
1.6.4 DigiDAP.....	15
1.6.5 Conclusiones sobre el análisis de las soluciones.....	16
1.7 Herramientas y tecnologías de desarrollo.....	16
1.7.1 Metodología de desarrollo Extreme Programming (XP).....	16
1.7.2 Lenguaje de programación Java.....	17
1.7.3 Tecnología Enterprise Java Bean (EJB).....	19
1.7.4 Lenguaje Unificado de Modelado 2.0.....	20
1.7.5 Herramienta CASE. Visual Paradigm 3.4.....	21
1.7.6 Entorno Integrado de Desarrollo. NetBeans 7.1.....	21
1.7.7 Sistema Gestor de Base de Datos. PostgreSQL 8.4.....	22
1.8 Conclusiones Parciales	23
CAPÍTULO 2: REQUISITOS Y DISEÑO	24
2.1 Introducción.....	24
2.2 Propuesta de sistema	24
2.2.1 Descripción de Procesos	24
2.2.2 Configurar Tipo Documental	24
2.2.3 Asociación de Metadatos.....	25
2.2.4 Personas relacionadas con el sistema.....	26
2.2.5 Especificación de requisitos no funcionales del sistema.	27
2.3 Fase de exploración y planificación.	27
2.3.1 Exploración.....	27
2.3.1.1 Historias de Usuario.....	28
2.3.2 Planificación.	34
2.3.2.1 Estimación de esfuerzo por Historias de usuarios.....	34
2.3.2.2 Plan de iteraciones.....	34

ÍNDICE

2.3.2.3	Plan de duración de las iteraciones.....	35
2.3.3	Plan de entregas	36
2.4	Estilo arquitectónico cliente – servidor.....	36
2.5	Arquitectura del sistema.	36
2.6	Patrones de diseño.....	39
2.7	Tarjetas Colaborador - Responsabilidad – Clase (CRC).....	41
2.8	Diagrama de Paquetes.....	44
2.9	Diseño de la base de datos.....	45
2.10	Conclusiones parciales.....	46
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....		48
3.1	Introducción.....	48
3.2	Fase de implementación.....	48
3.2.1	Iteraciones.....	48
3.2.2	Primera iteración	49
3.2.3	Segunda iteración.....	49
3.2.4	Tareas de la ingeniería	50
3.3	Diagrama de componentes del sistema.....	52
3.4	Diagrama de despliegue.....	54
3.5	Pruebas.....	55
3.6	Pruebas Unitarias	55
3.7	Casos de prueba.....	57
3.8	Análisis de los resultados.....	58
3.9	Pruebas de aceptación.....	58
3.10	Conclusiones parciales	63
CONCLUSIONES GENERALES		64
RECOMENDACIONES.....		65
BIBLIOGRAFÍA.....		66
ANEXOS.....		69
GLOSARIO DE TÉRMINOS		74

ÍNDICE DE FIGURAS

Figura 1: Atributos de los metadatos.....	10
Figura 2: Diagrama de Proceso de configuración de un tipo documental.	25
Figura 3: Diagrama de proceso de asociación de metadatos.	26
Figura 4: Vista de la arquitectura (Cliente).	38
Figura 5: Vista de la arquitectura (Servidor).	39
Figura 6: Diagrama de Paquetes.....	44
Figura 7: Modelo de datos.....	46
Figura 8: Diagrama de componentes Capa de Presentación.	52
Figura 9: Diagrama de Componentes Capa de Negocio.	53
Figura 10: Diagrama de Componentes Capa de Acceso a Datos.	53
Figura 11: Diagrama de despliegue.	54
Figura 12: Código asociar metadatos.....	56
Figura 13: Grafo de flujo asociado.	56
Figura 14: Código de prueba CU Asociar metadatos.	57
Figura 15: Resultados de las pruebas del sistema.....	58
Figura 16: Diagrama de proceso Rectificar Tipo Documental.	69
Figura 17: Diagrama Cancelar Tipo Documental.	69
Figura 18: Diagrama Consultar Documento Digital.	70
Figura 19: Pruebas unitarias.	71
Figura 20: Pruebas GestorNCValoresMetadatos.	71
Figura 21: Pruebas GestorNCTipoDocumental.	72
Figura 22: Prueba GestorNCMetadatos.	72
Figura 23: Pruebas GestorNCNota.	73
Figura 24: Pruebas GestorNCTipoDato.	73

ÍNDICE DE TABLAS

Tabla 1: Personas relacionadas con el sistema.	26
Tabla 2: HU Configurar Tipo Documental.	29
Tabla 3: HU Configurar Metadatos.	30
Tabla 4: HU Buscar Coincidencias Documentales.	30
Tabla 5: Modificar Tipo Documental.	31
Tabla 6: Consultar documento digital.	32
Tabla 7: Rectificar tipo documental.	32
Tabla 8: Cancelar tipo documental.	33
Tabla 9: Anexar tipo documental a expedientes.	34
Tabla 10: Estimación de esfuerzo por HU.	34
Tabla 11: Plan de duración de las iteraciones.	35
Tabla 12: Plan de duración de entregas.	36
Tabla 13: Tarjetas CRC Configurar metadatos de tipo documental.	42
Tabla 14: Tarjetas CRC Configurar metadatos.	42
Tabla 15: Tarjetas CRC Asociar metadatos.	42
Tabla 16: Tarjetas CRC Modificar metadatos.	43
Tabla 17: Tarjetas CRC Rectificar tipo documental.	43
Tabla 18: Tarjetas CRC Cancelar tipo documental.	43
Tabla 19: Tarjetas CRC Buscar coincidencias documentales.	44
Tabla 20: Tarjetas CRC Modificar tipo documental.	44
Tabla 21: Componentes abordados en la primera iteración.	49
Tabla 22: Componentes abordados en la segunda iteración.	49
Tabla 23: Tareas de la ingeniería de la primera iteración, Configurar metadatos de tipo documental.	50
Tabla 24: Tareas de la ingeniería de la primera iteración, Rectificar tipo documental.	50
Tabla 25: Tareas de la ingeniería de la primera iteración, Cancelar tipo documental.	50
Tabla 26: Tareas de la ingeniería de la primera iteración, Modificar tipo documental.	51
Tabla 27: Tareas de la ingeniería de la primera iteración, Configurar metadatos.	51
Tabla 28: Tareas de la ingeniería de la primera iteración, Asociar metadatos.	51
Tabla 29: Tareas de la ingeniería de la primera iteración, Modificar metadatos.	52

ÍNDICE DE TABLAS

Tabla 30: Caso de prueba de aceptación.....	59
Tabla 31: Caso de prueba de aceptación: Comprobar Configurar Tipo Documental.	60
Tabla 32: Caso de prueba de aceptación: Comprobar Configurar Metadatos.	60
Tabla 33: Caso de prueba de aceptación: Comprobar Asociar Metadatos.	60
Tabla 34: Caso de prueba de aceptación: Comprobar buscar coincidencias documentales.	61
Tabla 35: Caso de prueba de aceptación: Comprobar Rectificar Tipo Documental.	61
Tabla 36: Caso de prueba de aceptación: Comprobar Cancelar Tipo Documental.	61
Tabla 37: Caso de prueba de aceptación: Comprobar Modificar Tipo Documental.	62
Tabla 38: Caso de prueba de aceptación: Comprobar Modificar Metadatos.....	62
Tabla 39: Caso de prueba de aceptación: Comprobar consultar documento digital.	63
Tabla 40: Caso de prueba de aceptación: Comprobar anexar tipos documentales a expedientes.	63

INTRODUCCIÓN

La salvaguarda del patrimonio cultural es uno de los temas que, en los últimos años, se ha perfilado como prioritario, ya que constituye uno de los elementos fundamentales en la formación y conservación de la identidad de los pueblos. En este siglo, considerado como la centuria de la información y el conocimiento, las Tecnologías de la Información y de las Telecomunicaciones (TIC) ocupan un lugar destacado en la sociedad con el fin de brindar servicios eficientes y eficaces. Hasta el momento la actual evolución tecnológica ha provocado que las instituciones hayan sido obligadas a elaborar estrategias de conservación acordes a las necesidades del entorno.

El uso de las TIC representa sin duda una oportunidad, cuando de la salvaguarda del patrimonio cultural se trata, ya que su utilización contribuye no sólo a la catalogación estandarizada del patrimonio sino a su difusión y apropiación por la sociedad en su conjunto. (1)

Un aspecto fundamental en cuanto a la utilización de las TIC en favor de la preservación y conservación de la información es la digitalización de documentos. La digitalización es el proceso de convertir información analógica en formato digital. Esta viene dada por la necesidad de resolver problemas entre los que se encuentran la ampliación del acceso, la preservación y conservación, la reducción de costos, optimización del espacio de almacenamiento físico, la transformación de servicios o la recuperación de la información, y lo más importante es que no se altere la información que contiene el material, lo que significa que la representación digital contenga la misma información y datos que la representación analógica. El método más efectivo para garantizar la legalidad de este proceso es el otorgamiento de la firma digital. (2) (3)

Luego de efectuar el proceso de digitalización de documentos, se debe realizar la asociación de metadatos con el objetivo de poder gestionar la información contenida en dichos documentos sin tener que utilizar el recurso. La asociación de metadatos no es más que registros de información de diferentes tipos que pueden obtenerse de forma manual, automática, mediante instrumentos, o a través de procesamiento realizado por computadores. Cuando se hace referencia a los metadatos, realmente se está haciendo referencia a la existencia de datos sobre los datos. La interpretación global de este concepto permite visualizar todo aquello que se puede decir sobre un objeto, que puede ser manipulado como una entidad discreta. Las ventajas asociadas a los metadatos hacen que su utilización en sistemas para la digitalización de documentos sea imprescindible. (4) (5)

Existe gran necesidad de que el proceso de asociación de metadatos se realice de una forma genérica y dinámica, debido a un conjunto de factores como son, la existencia de

INTRODUCCIÓN

diferentes tipos de archivos que almacenan gran variedad de información: primarios, relacionados a actividades administrativas, y secundarios, destinados a la investigación que se corresponden al patrimonio cultural e histórico de los pueblos, así como además, de la existencia de una gran cantidad de tipos documentales diferentes en su contenido, estructura y diseño que relacionan diferentes tipos de información. Por otra parte en la actualidad, en el mundo se manejan grandes volúmenes de información, almacenados en archivos físicos, lo que conlleva al deterioro de los documentos que almacenan esta información por la constante manipulación a la que son sometidos. También, existen casos en los que dicho proceso se realiza de forma automática, pero reducido solamente a los archivos específicos de la entidad que utiliza el sistema implementado. Actualmente existen sistemas que realizan este proceso de forma genérica, pero estos sistemas son basados en tecnologías de desarrollo privadas, lo cual implica un aumento considerable en el costo de las soluciones debido al pago de licencias y patentes, también existen otras soluciones que se llevan a cabo según las necesidades del cliente, por lo que no existe flexibilidad ante los cambios que puedan surgir luego de la puesta en marcha de la solución. Todo esto da como resultado, el surgimiento de la necesidad de crear un sistema que pueda agrupar en su implementación todos los tipos documentales existentes, el cual permita realizar un proceso de asociación de metadatos dinámico y genérico para así poder ser aplicado en cualquier sistema de gestión documental y archivística de acuerdo a las necesidades de dicho sistema.

El sistema implementado deberá ser integrado a una plataforma de digitalización llamada SuiteDoc, la cual tendrá principalmente tres diferentes líneas de desarrollo: solución de software, transformación organizacional y gestión archivística. En la línea de solución de software se distinguen tres directrices fundamentales: desarrollo de plataforma para la creación y personalización de soluciones de digitalización de documentos, creación de un repositorio de componentes para la digitalización desde sistemas externos e interacción con dispositivos de hardware. Garantizar la configuración de metadatos en tiempo real durante la construcción del sistema de digitalización y el almacenamiento y gestión de los mismos es una de las primeras necesidades identificadas en la primera directriz, debido a la importancia que se le confiere a los metadatos en este tipo de sistema, problemática que da paso a la identificación del siguiente **problema a resolver**: ¿Cómo garantizar la identificación y asociación dinámica de metadatos a partir de documentos digitales en la plataforma SuiteDoc?

Por lo que se propone como **objeto de estudio** el proceso de desarrollo de software para la gestión documental, enmarcado en el procesamiento de datos de documentos digitales con valor legal como **campo de acción**. En consecuencia el **objetivo general** es

INTRODUCCIÓN

Desarrollar un componente que contribuya a la identificación y asociación dinámica de metadatos a partir de documentos digitales en la plataforma SuiteDoc. Se utiliza como referencia el objetivo general y el problema a resolver, obteniendo como resultado la siguiente **idea a defender**: Con el desarrollo de un componente se contribuirá a la identificación y asociación dinámica de metadatos a partir de documentos digitales en la plataforma SuiteDoc.

Los **objetivos específicos** que persigue esta investigación son los siguientes:

Elaborar el marco teórico de la investigación.

Especificar los requisitos no funcionales y las historias de usuario del componente de asociación de metadatos.

Elaborar artefactos del diseño y realizar la implementación del componente a partir de los elementos del diseño obtenidos.

Validar los artefactos generados y la solución obtenida a partir del desarrollo de pruebas unitarias y pruebas de aceptación para obtener el nivel de calidad requerido.

Para dar cumplimiento a los objetivos específicos se trazaron las siguientes **tareas de la investigación**:

- Estudio de las soluciones existentes para la asociación de metadatos en sistemas para la digitalización de documentos en Cuba y el mundo.
- Selección de las principales herramientas, metodologías y tecnologías para la solución del problema.
- Especificación de requisitos no funcionales de software e historias de usuario.
- Elaboración de los artefactos de la etapa de diseño correspondientes a la metodología de desarrollo XP.
- Implementación del componente de metadatos de acuerdo a las normas y estándares establecidos para la codificación.
- Aplicación de pruebas de aceptación, pruebas unitarias y análisis de sus resultados para validar la implementación del componente.

Métodos de investigación

Métodos Teóricos:

Análisis y Síntesis: Es utilizado para analizar y comprender la información relacionada con la gestión documental, la digitalización de documentos y la asociación de metadatos, logrando la asimilación de los elementos más importantes relacionados con el objeto de estudio.

Histórico – Lógico: Es utilizado para estudiar la evolución de las técnicas de asociación de metadatos que se enmarcan en la digitalización de documentos realizando énfasis en

INTRODUCCIÓN

las características más importantes que han surgido y tienen un nivel de impacto en la solución a desarrollar.

Estructura de la tesis

El contenido a desarrollar en la investigación está estructurado en 3 capítulos:

Capítulo 1: Fundamentación Teórica: En este capítulo se abordan aspectos sobre el proceso de asociación de metadatos a documentos digitales. Se realiza un estudio sobre cómo se realiza este proceso en soluciones existentes en el mundo, en Cuba y en la Universidad de las Ciencias Informáticas, y se realiza un análisis de las tecnologías, metodologías y herramientas a utilizar en el desarrollo del presente trabajo.

Capítulo 2: Características y Diseño del sistema: En este capítulo se describe la propuesta del sistema mostrando como artefactos principales el Modelo del sistema, la especificación de los requisitos no funcionales, las historias de usuarios, la estimación de esfuerzo por cada una de ellas; así como los procesos que serán objetos de automatización.

Capítulo 3: Implementación y prueba: En este capítulo se muestra los diagramas de componentes por capas: Presentación, Negocio, Acceso a Datos que representan la implementación del sistema. Se exponen las normas y estándares de codificación y de diseño de interfaz utilizadas y definidas para el desarrollo. Se evalúa el grado de calidad y fiabilidad de la implementación del sistema realizando un análisis de los resultados alcanzados en la aplicación de pruebas unitarias y de caja negra.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

El presente capítulo enmarca la fundamentación teórica de la investigación, se abordan aspectos sobre el proceso de asociación de metadatos a documentos digitales. Posteriormente se hace un estudio del estado actual de soluciones existentes en el mundo, en Cuba y en la universidad que desarrollan esta rama y las características de las tecnologías, metodología y herramientas a utilizar en el desarrollo del presente trabajo.

1.2 LA CONSERVACIÓN Y PRESERVACIÓN DEL PATRIMONIO DOCUMENTAL

La conservación y preservación del patrimonio documental de los pueblos durante los últimos años se ha convertido en una tarea de gran importancia en todo el mundo. En la actualidad existe una enorme cantidad de información registrada en documentos físicos, información que posee una importancia vital, ya que tributa a la difusión y apropiación del patrimonio documental por la sociedad. Con el objetivo de fomentar el cuidado de los grandes volúmenes de información almacenada en los documentos surge la necesidad de crear estrategias encaminadas a la preservación y conservación del patrimonio documental, estas medidas estarán destinadas a proteger adecuadamente los documentos, con el fin de prolongar su utilización en condiciones óptimas durante el mayor tiempo posible.

La preservación es el acto de preservar, que significa “proteger, resguardar anticipadamente a una persona, animal o cosa, de algún daño o peligro.” Y la conservación se refiere al acto de conservar que es “mantener algo o cuidar de su permanencia.” (6) (7)

Al igual que en otros ámbitos de la sociedad, las TIC han encontrado su propio camino en el ámbito de la preservación y conservación de la información y, concretamente, en el mundo del patrimonio cultural se están buscando las vías de aplicación propias de estas tecnologías. (8)

Para lograr una eficiente aplicación de las TIC a un sistema de administración de documentos se debe considerar los siguientes elementos para diseñar una estrategia tecnológica en el proceso de gestión documental: Organización, captura, administración, almacenamiento y preservación. (8)

Es importante aclarar que el uso de las TIC no debe ser dirigido a sustituir o desechar los documentos originales, por el contrario, siempre debe estar encaminado a la protección de estos. La tecnología debe ser utilizada para disminuir la frecuencia de manipulación de los documentos físicos que tanto contribuye al deterioro, ampliar las alternativas de acceso a la información, mantener la organización de los archivos en función de agilizar

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

la localización de la información, garantizar las condiciones de locales, almacenamiento, entre otros.

1.3 LA DIGITALIZACIÓN DE DOCUMENTOS EN FUNCIÓN DE LA PRESERVACIÓN Y CONSERVACIÓN DE DOCUMENTOS

La digitalización, reconocida internacionalmente como una solución para contribuir a la preservación y conservación de fondos documentales es, más que una técnica, todo un proceso que involucra recursos, actividades, trabajadores, seguridad legal y protección de los documentos. La digitalización de documentos permite el acceso de forma más sencilla y rápida a la información, además de contribuir en gran medida a la conservación y preservación de la información almacenada en formato duro. (9)

El uso de la digitalización de documentos proporciona a las empresas diversas ventajas, principalmente en productividad y seguridad al permitir el rápido y eficiente acceso y evitar pérdidas de documentos y expedientes.

¿Qué es digitalizar?

Según el diccionario de la Real Academia de la Lengua Española digitalizar es expresar los datos de un tipo documental o imagen en forma digital”, también se plantea que digitalizar es convertir cualquier señal de entrada como una imagen o una señal de sonido, en una serie de valores numéricos. (10)

Otro concepto plantea que la digitalización no es más que la conversión de documentos en papel a imágenes electrónicas por medio de un scanner, las cuales son almacenadas en una computadora. (11)

Varios autores plantean numerosas medidas que deben tenerse en cuenta para el proceso de digitalización, estas medidas estarán encaminadas a fomentar el cuidado y la preservación del patrimonio documental que se necesite digitalizar, entre las consideraciones necesarias aparece principalmente que:

- Se deben utilizar guantes y batas y se evitará la excesiva manipulación de los documentos en dependencia del estado en el que se encuentren los mismos.
- De modo general, se optará por capturar los documentos dentro de medidas fijas, de modo que se facilite la operación de recortado de la imagen, automatizando ésta si es posible.
- No se desencuadernarán los documentos ni se adoptarán medidas que vayan contra de su integridad física, aun cuando la finalidad sea obtener una mejor reproducción. Sin embargo, en función de aplicar técnicas de encuadernación que favorezcan la conservación de los documentos, estas deberán ser aplicadas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- La digitalización se realizará preferentemente con un escáner aéreo o cenital. Otros dispositivos de captura de la imagen posibles son la cámara de fotos digital y los escáneres de sobremesa y tracto masivo.
- En ambos casos, sería aconsejable el uso de cartas de escala de grises y/o color, de cara a obtener un perfil de color que sea fiel a los originales capturados. (12) (13)

Digitalizar no es solo una actividad, sino que en la actualidad se considera como un macroproceso que involucra diferentes procesos que garantizan la calidad de la obtención de documentos digitales.

El proceso de digitalización de documentos evita el deterioro por el uso, conservación del documento para su disponibilidad al público, incremento en la investigación y los estudios académicos, refuerza la identidad nacional y patrimonio cultural, mejora los niveles de gestión y accesibilidad a la información, y agiliza los trámites del servicio administrativo. (3)

Se plantea que todo proyecto de digitalización debe contemplar tres aspectos fundamentales:

- Los objetivos que se pretenden conseguir: La definición de estos objetivos dependerá de lo que cada institución pretenda realizar.
- El ámbito de aplicación del proyecto: Implica integrar los archivos en papel, los documentos electrónicos y las bases de datos, en el caso de que esté trabajando en esas tres áreas por separado.
- La responsabilidad de las partes involucradas en el proyecto: Un proyecto de digitalización requiere la participación de administradores, archivistas e informáticos, de manera que se debe establecer claramente, desde el principio del proyecto, la responsabilidad de cada uno de los participantes. (3) (13) (9)

La digitalización tiene una gran importancia en la actualidad por la necesidad de conservar el patrimonio documental de los pueblos, no obstante es de igual importancia que a las imágenes digitalizadas se le asocien metadatos, ya que durante el proceso de asociación de metadatos se registran todos los datos asociados al documento facilitando así la obtención de la información presente en el documento. La obtención del documento digital a partir de la digitalización de tipos documentales no el fin del proceso. Existe una actividad muy importante relacionada a los datos contenidos en el documento que constituye punto de partida para la generación de información.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4 METADATOS

Los metadatos considerados por la mayoría como los datos sobre los datos, pueden casi siempre contener toda la información necesaria para producir el conocimiento.

¿Qué son los metadatos?

Se ha podido conceptualizar a los metadatos de distintas formas aunque todas en general plantean elementos estrechamente relacionados. Algunas de las definiciones más universales en cuanto al tema son:

- Según el colectivo de autores de la Real Academia Española los metadatos son datos que describen otros datos.
- Son datos sobre datos.
- Son un conjunto de elementos que pueden ser usados para describir y representar objetos de información.
- Datos que describen los atributos de un recurso de información y apoyan su localización identificación, evaluación y selección, entre otras funciones, las cuales pueden ser llevadas a cabo por un usuario final o por sus agentes (otras personas o sistemas automatizados).
- Datos asociados con objetos de información los cuales le evitan a sus usuarios potenciales el tener un conocimiento previo de su existencia o sus características.
- Datos con sentido propio que proporcionan información o documentación acerca de otros datos manejados dentro de una aplicación o ambiente. Los metadatos pueden incluir información descriptiva acerca del contexto, calidad y condiciones o características de los datos. (14) (15) (16) (10)

Luego de detallar alguno de los conceptos más conocidos de los metadatos se puede concluir que un metadato no es más que un dato estructurado sobre la información, o sea, información sobre información, o de forma más simple, datos sobre datos.

Una de las características más importantes de los metadatos es su capacidad de relación o de establecer enlaces, de esta forma se han hecho imprescindibles en la recuperación global de la información, puesto que se trata de indizar y clasificar inconmensurables cantidades de información de diversos tipos. (17)

Los metadatos pueden ser, entonces, construcciones complejas que se necesitan crear y mantener y que se refieren a un conjunto específico de objetos. Se considera a los metadatos, en las áreas de telecomunicaciones e informática, como información no relevante para el usuario final pero sí de suma importancia para el sistema que maneja la data. Los metadatos se envían junto a la información cuando se realiza alguna petición o

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

actualización de la misma. El uso de los metadatos mencionado más frecuentemente es la refinación de consultas a buscadores. Usando informaciones adicionales los resultados son más precisos, y el usuario se ahorra filtraciones manuales complementarias. Los metadatos facilitan el flujo de trabajo convirtiendo datos automáticamente de un formato a otro. Para eso es necesario que los metadatos describan contenido y estructura de los datos. (18)

Los metadatos tienen tres funciones básicas:

- Proporcionar una descripción de una entidad de información junto con otra información necesaria para su manejo y preservación.
- Proporcionar los puntos de acceso a esa descripción.
- Codificar esa descripción. (19)

En conjunto con las funciones básicas de los metadatos existen, además, tres criterios en los que estos pueden clasificarse:

- Contenido: Subdividir metadatos por su contenido es lo más común. Se puede separar los metadatos que describen el recurso mismo de los que describen el contenido del recurso. Es posible subdividir estos dos grupos más veces.
- Variabilidad: Según la variabilidad se puede distinguir metadatos mutables e inmutables. Los inmutables no cambian, no importa qué parte del recurso se vea. Los mutables difieren de parte a parte.
- Función: Los datos pueden ser parte de una de las tres capas de funciones: subsimbólicos, simbólicos o lógicos. Los datos subsimbólicos no contienen información sobre su significado. Los simbólicos describen datos sub simbólicos, es decir añaden sentido. Los datos lógicos describen cómo los datos simbólicos pueden ser usados para deducir conclusiones lógicas, es decir añaden comprensión. (20)

Los metadatos están compuestos por atributos, cada atributo, tiene diferentes aspectos que los caracterizan. Ejemplo de estos atributos son detallados en el siguiente cuadro de texto (17):

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Atributo	Características	Ejemplos
Fuente	Metadatos internos generados por el agente creador con el propósito de informar sobre el momento de su creación	Nombres de ficheros Estructuras de directorios Formatos de ficheros y algoritmos de compresión
	Metadatos externos relativos a una información que se modifica después de su creación	Registros catalográficos Información sobre derechos de autor
Método de creación	Metadato generado automáticamente por un ordenador	Índices de palabras clave Logs Weblogs y bitácoras
	Metadatos creados manualmente	Herramientas descriptivas
Naturaleza	Creados por el autor del documento objeto	Los utilizados en páginas HTML
	Generados por profesionales de la información, independientemente de quién sea el autor del documento objeto	Registros MARC Encabezamientos de materia
Estado	Estático: no cambian desde su creación	Título, fecha de creación
	Dinámico: varía con el uso del documento objeto	Estructuras de directorios Logs
	A largo plazo: necesario para asegurarse de que el documento objeto será accesible en todo momento	Información de los derechos (de autor, de uso, de difusión...)
	A corto plazo: con clara vocación transaccional	Información sobre el uso
Estructura	Con estructura basada en estándares	MARC TEI AACR2
	Sin estructura predecible	Metadatos ad hoc (la mayoría de los generados en y para bibliotecas digitales)
Semántica	Normalizados por medio de un vocabulario controlado	MARC AACR2
	No controlados	Etiquetas HTML
Nivel	Colecciones de metadatos relativos a colecciones de documentos objeto	MARC Índices especializados
	Un metadato relativo a un documento objeto individual, fuera de cualquier colección	Información sobre el formato Leyenda de una imagen

Figura 1: Atributos de los metadatos.

Los metadatos poseen una gran cantidad de ventajas que hacen que su utilización sea totalmente necesaria en los sistemas informáticos que se necesite hacer gestión de la información y a la vez preservar el patrimonio documental asociado a esa información, a continuación se detallan varias de las ventajas más significativas que se pueden lograr a través de la asociación de metadatos:

- Permiten establecer relaciones entre registros y ficheros diferentes.
- Son más flexibles en la indización y presentación del texto.
- Son menos complejos, permiten la realización de diferentes búsquedas.
- Permiten la automatización de procesos en su creación incluso su adicción automática.
- Permiten el uso conjunto de más de un esquema. (17)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La asociación de metadatos en sistemas de digitalización de documentos

Los metadatos son las herramientas de las que disponemos para especificar la información contextual asociada a cada documento: su contenido, el historial de las transformaciones sufridas por cada objeto digital, los formatos de cada fichero, etc.

Cada tipo de documento puede definirse por un conjunto de metadatos. De este modo, cada documento tendrá metadatos propios y se le podrán asociar reglas de gestión basadas en dichos metadatos.

Cabe destacar aquí que puede ser interesante indexar varios archivos con un único conjunto de metadatos. La relación entre la reseña descriptiva y los archivos almacenados debería ser flexible para permitir tener desde cero a «n» archivos adjuntos a una misma reseña. (21)

Estructura de reseñas.

Se llama reseña al conjunto de metadatos asociados a un documento. El contenido de esta reseña suele introducirse mediante un formulario. (22)

Metadatos descriptivos.

De cada una de las obras digitalizadas se generan metadatos descriptivos en formato.

Marc XML. Los ficheros “.mrc” de las obras a digitalizar se obtienen desde Unicorn (ISO2709). Estos ficheros se desglosan en dos partes:

- .mrc que correspondan a objetos simples (documentos compuestos por una sola imagen). Se generará un fichero XML que englobará a todos los objetos simples.
- .mrc que correspondan a objetos complejos (documentos compuestos por varias imágenes). Se generará un fichero XML por cada objeto complejo que vaya a cargarse. (23)

1.5 SOLUCIONES INTERNACIONALES DE REFERENCIA

1.5.1 QuickScan Pro

QuickScan Pro (QSP) es una solución de captura de documento estandarizada para una única estación de trabajo de EMC Captiva®, diseñada dentro de la arquitectura ISIS, la interfaz estándar que une escáneres con aplicaciones de software. Con QuickScan Pro, los usuarios pueden escanear documentos por lotes y, a continuación, ver documentos separados automáticamente, con la posibilidad de manipular los archivos antes de exportarlos a formatos de archivo habituales. (24)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

QuickScan Pro es ideal para empresas pequeñas y medianas, así como para departamentos de organizaciones de mayor tamaño, para eliminar el papeleo manual y automatizar el manejo de miles de documentos.

Escaneado, presentación e indexación

Las funciones de escaneado, indexación, presentación y visualización de QSP permiten a los usuarios agilizar el proceso de escaneado por lotes. Además de ser compatible con escáneres estándar binarios y con posibilidad de color, QSP también soporta funciones avanzadas de escaneado de documentos, incluyendo MultiStream y detección automática del color. El reconocimiento óptico de caracteres (OCR) de QSP convierte documentos electrónicos en documentos de texto utilizables tales como texto .PDF, XML, Word y ASCII, HTML y otros formatos de archivos de texto configurables para Web. El OCR zonal captura los datos en zonas previamente configuradas de una imagen, como, por ejemplo, los números de factura o de recibos de ventas. Para ayudar a automatizar el proceso de captura de datos, usando la detección de códigos de barras para capturar datos de códigos de barras 1D y 2D, QSP puede rellenar automáticamente campos de índice, lotes independientes sin necesidad de páginas separadoras y asignar nombres automáticamente a los documentos. Las imágenes de los documentos escaneados se pueden visualizar y manipular mediante funciones como el dimensionamiento rápido (ampliación y reducción), la rotación y la escala de grises. (24)

Digitalización

Usa las librerías ISIS® (Especificación de imágenes e interfaz de escáner), que lo hacen compatible con más de 300 escáneres de diversos fabricantes y le permiten ajustar los valores de configuración de su escáner a través de la propia aplicación QuickScan. Los perfiles de digitalización le permiten almacenar la configuración de las tareas de digitalización más habituales con el fin de simplificar el proceso. (24)

Reconocimiento óptico de caracteres (OCR)

Permite a los usuarios realizar un procesamiento de OCR de toda la página o de una zona y posteriormente exportar los resultados del procesamiento.

1.5.2 Sistema de Digitalización de Documentos (SDD)

El Sistema de Digitalización de Documentos (SDD) tiene como objetivo: la conversión y almacenamiento de documentación en papel a archivos digitales almacenables en bases de datos. Se ubica como una aplicación intermedia entre los documentos en papel y los

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

sistemas de gestión documental. Su función es la de convertir e ingresar los documentos físicos en los sistemas de gestión documental. (25)

Ventajas:

- Flexibilidad para ser adaptado a cualquier organización laboral.
- Economía de recursos (espacio y dinero).
- Posibilita la recuperación de espacio libre en archivos.
- Ofrece consulta e integración de los documentos en redes informáticas y gestores documentales y/o flujos de trabajo.
- Además de las funciones de digitalización, permite consultas por web, por visualizadores, integrando un sistema de gestión en red o todas simultáneamente.
- Transforma en información exacta, recuperable y lista para ser exportada a sus aplicaciones de gestión y bases de datos corporativas.
- También puede exportarse a CD con visualizador propio.
- Fácil de utilizar, permite una máxima escalabilidad y flexibilidad en su configuración.
- Operación Flexible.
- Incorporación de documentos mediante archivos.
- Facilidad de búsqueda combinada.
- Selección de varios documentos. (25)

Algunas características:

- Permite la identificación de caja, estante, sector de archivo de la documentación original, formando parte de un sistema de archivo.
- Soporte para cualquier tipo de scanner con interface TWAIN.
- Importación de Imágenes TIF, JPG, GIF, BMP. • Creación de CDS auto ejecutables.
- Lectura de códigos de barra.
- Envío de documentos por correo electrónico.
- Visualización de los documentos en PDF para evitar la adulteración.
- Creación de índices o claves de búsquedas.
- Reconocimiento óptico de caracteres (OCR)
- Soporte para MySQL e incluye otras bases de datos con interfaces ODBC.
- Permite exportar los documentos a bases de datos compatibles con SQL.
- Almacenamiento de los documentos en cualquier medio de almacenamiento.
- Almacenamiento final del documento.

Cómo funciona:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El documento pasa por las etapas de una línea de trabajo hasta que queda listo para salir del sistema. Un documento puede ser: una carilla, una hoja, un grupo de hojas, un cuadro de microfilm, una o varias tiras de microfilm, negativos fotográficos y cualquier tipo de documento que un scanner pueda entregar como archivo o grupo de archivos de imagen mediante la interface TWAIN. El SDD también permite el ingreso de documentos digitales como archivos propietarios (Word, Excel, CAD, etc.).

Los documentos ingresados al SDD, recorren una línea de trabajo que está dividida en etapas. Cada etapa de estas líneas de trabajo es ejecutada por los operadores, hasta llegar a la última etapa, en la que el documento se da por finalizado. Los documentos finalizados están listos para abandonar el sistema e ingresar al sistema de gestión documental, ser almacenados en medios físicos o ser exportados como PDF.

A las etapas se les asignan niveles que deben ser ejecutados en orden descendente para un mismo documento. Una etapa de mayor nivel no podrá ser ejecutada hasta que la etapa de nivel inferior no haya sido finalizada.

La aplicación de OCR en la asociación de metadatos, evidenciada en ambas soluciones es, sin dudas una buena técnica, aunque las librerías que utiliza son privativas, y presentan altos costos en la adquisición de licencias, además el OCR depende en alto grado de la calidad visual de la imagen digital obtenida por lo que de igual forma el usuario debe revisar y corregir errores en el dato. Esto hace que la técnica sea menos factible a medida que los documentos a escanear presenten mayor nivel de deterioro. Aunque el aspecto más significativo de esta solución en cuanto a la asociación de metadatos es la utilización del OCR, la misma no ofrece la posibilidad a los usuarios de establecer y gestionar aquellos datos que considere necesarios de forma genérica para cada tipo de documento.

1.6 SOLUCIONES NACIONALES DE REFERENCIA

1.6.1 Papiro

Papiro es un producto informático desarrollado en Cuba hace varios años que permite la digitalización de documentos. Es de uso libre y emplea herramientas igualmente libres, aspecto significativo porque no presenta costos por concepto de adquisición de licencias. Incluye la gestión de publicaciones periódicas, especialmente revistas y medios de publicación similares en formato electrónico. Sin embargo el proceso de asociación de metadatos es muy básico pues solo tributa a la consulta electrónica de documentos. Los metadatos que permite recopilar son descriptivos del documento, es decir solo sirven para su indexación y localización. (26)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.6.2 Digipyrus

Es un sistema, desarrollado en el marco del Convenio Cuba-Venezuela, con el objetivo de brindar una solución para la digitalización de los Tomos de los Registros y Notarías de esta nación, contribuyendo a agilizar el acceso a la información que a diario se tramita en cada una de las Oficinas de los Registros y Notarías Públicas. Este sistema surge debido a que se demanda de una gestión más rápida, automática y segura de los trámites que se realizan con los documentos archivados en las Oficinas de los Registros y Notarías Públicas. De ahí, que se haya propuesto presentar una solución que cumpla con el objetivo de dotar a los Registros y Notarías Públicas de un sistema que contenga las funciones, para realizar el proceso de digitalización de todos los documentos archivados y de esta forma obtener el fondo digital de cada una de estas.

1.6.3 Alfabéticas

Sistema para la digitalización de tarjetas de identificación de ciudadanos venezolanos. Está centrado en el diseño organizacional, instalación y transferencia de ocho (8) Centros de Digitalización para la obtención de objetos digitales con valor legal, generados a partir de la información contenida en la Tarjetas de Alfabética, además del diseño y desarrollo de un sistema informático que será desplegado en cada centro. La información almacenada será utilizada en los procesos de gestión que realiza actualmente el Servicio Administrativo de Identificación, Migración y Extranjería de Venezuela (SAIME). (27)

1.6.4 DigiDAP

Sistema para la digitalización del fondo documental para la división de antecedentes penales de la república bolivariana de Venezuela, desarrollado por el centro CEGEL adscrito a la Universidad de las Ciencias Informáticas en el marco de cooperación internacional entre Venezuela y Cuba. Esta solución soporta todos los procesos del centro para la digitalización del fondo documental de la División de Antecedentes Penales, tiene como objetivo fundamental la obtención de imágenes con valor legal para favorecer la preservación y conservación del fondo documental de la institución. Permite la recepción de documentos, mejora de la calidad visual de imágenes escaneadas, posee asociación de metadatos, búsqueda de coincidencias documentales, agrupación de unidades documentales, revisión de trámites, otorgamiento y firma digital e identificación de documentos por código de barra, tiene soporte para reestructuración del fondo documental, seguridad y control de los documentos durante el proceso, realización de trámites de inscripción, rectificación y cancelación de unidades documentales, sistema de Administración y Configuración integrado, sistema de reportes estadísticos y de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

seguridad (trazas, excepciones, etc.) integrado, flexible a los cambios de flujo del proceso, garantizando la adaptación de la misma a cambios de estado de los trámites u omisión de etapas del proceso por variaciones del negocio. Permite el procesamiento de unidades documentales genéricas, aunque no llega a un nivel de genericidad completo puesto que solo gestiona para todos los tipos documentales algunos metadatos que permiten la indexación y búsqueda de documentos. (28)

1.6.5 Conclusiones sobre el análisis de las soluciones

Luego de realizar un análisis tanto de las soluciones internacionales de referencia como de las nacionales, se puede concluir que a pesar de que estas tienen muchas características las cuales se podrían utilizar en favor del sistema que se desea implementar, tienen como principal inconveniente que no están implementados de forma genérica, solo pueden ser utilizados por la entidad para la cual fueron diseñados, además, en la mayoría de las soluciones realizadas tanto los entornos de desarrollo como la mayoría de las principales librerías utilizadas son privativas y se necesita el pago de una licencia para su utilización, mientras que el sistema que se desea realizar debe estar basado en tecnologías libres, por otra parte las que utilizan tecnologías de desarrollo libres realizan un proceso de asociación de metadatos muy básico y sencillo que no garantiza los elementos requeridos.

1.7 HERAMIENTAS Y TECNOLOGÍAS DE DESARROLLO

1.7.1 Metodología de desarrollo Extreme Programming (XP)

Las metodologías ágiles son sin duda uno de los temas en ingeniería de software que acaparan gran interés. Prueba de ello es que se están haciendo un espacio destacado en la mayoría de conferencias celebradas en los últimos años. Además ya es un área con cabida en prestigiosas revistas internacionales. En la comunidad de la ingeniería del software, se está viviendo con intensidad un debate abierto entre los partidarios de las metodologías tradicionales y aquellos que apoyan las ideas emanadas del "Manifiesto Ágil".

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (29)

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. (29)

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo. (29)

XP nace en busca de simplificar el desarrollo del software y que se lograra reducir el costo del proyecto, es un método ligero que no causa mucha sobrecarga de actividades de desarrollo, no impide el avance de nuestros proyectos y genera una disminución en la documentación, mientras que RUP utiliza principalmente los métodos pesados que intentan conseguir el objetivo común por medio del orden y documentación. XP es muy eficaz para proyectos de pequeño alcance, mientras que RUP es ideal para proyectos de gran alcance. XP tiene un desarrollo iterativo incremental mientras que RUP tiene un desarrollo en etapas interactivas.

1.7.2 Lenguaje de programación Java

Java es un lenguaje que permite tener desarrollos bastante completos y es de propósito general con el que se puede programar desde una aplicación Web o distribuida hasta una aplicación independiente sin ninguna conexión a Internet. Al ser un lenguaje libre dispone

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de una gran cantidad de características que lo convierte en la herramienta ideal para la creación aplicaciones distribuidas.

- **Simple:** El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Java es más complejo que un lenguaje simple, pero más sencillo que cualquier otro entorno de programación. El único obstáculo que se puede presentar es conseguir comprender la programación orientada a objetos, aspecto que, al ser independiente del lenguaje, se presenta como insalvable.
- **Orientado a objetos:** Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos.
- **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Este lenguaje obliga a la declaración explícita de métodos reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.
- **Distribuido:** Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- **Indiferente a la arquitectura:** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variados, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.
- **Portable:** Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.
- **Multithreaded:** Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

- Dinámico: El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red. (30)

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado. (30)

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para el desarrollo de todo tipo de aplicaciones informáticas. (31)

1.7.3 Tecnología Enterprise Java Bean (EJB)

EJB es una tecnología que permite desarrollar soluciones basadas en componentes facilitando así que los mismos se distribuyan sin restricción alguna en servidores de aplicaciones. Los componentes EJB son manejados por un contenedor EJB, comportándose como una aplicación que gestiona los mismos. A estos componentes se les conoce como Bean.

El contenedor EJB proporciona dos formas de interacción entre los componentes, una es a través de Interfaces Remotas a métodos (RMI, por sus siglas en inglés) y la otra es a través de Servicios Web (WS, por sus siglas en inglés). (32)

Los Enterprise Beans son componentes Java que implementan la tecnología EJB. Este tipo de componentes se ejecutan dentro de un contenedor EJB (entorno de ejecución dentro de un servidor de aplicaciones). (32)

Un Enterprise Bean encapsula la lógica del negocio de una aplicación. Esta lógica del negocio es el código que satisface el propósito de la aplicación. Su mayor beneficio es la construcción de aplicaciones de negocio portables, reusables y escalables, además de lograr un desarrollo modular de una solución empresarial. El contenedor EJB proporciona servicios a nivel de sistema a los Enterprise Beans, por lo que el desarrollador de estos componentes puede concentrarse solamente en resolver los problemas de la lógica del negocio. La tecnología EJB define tres tipos de Beans: Beans de sesión, Beans de entidad y Beans dirigidos por mensajes. (30)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.7.4 Lenguaje Unificado de Modelado 2.0

Una exigencia de la gran mayoría de instituciones dentro de su Plan Informático estratégico, es que los desarrollos de software bajo una arquitectura en Capas, se formalicen con un lenguaje estándar y unificado. Se necesita un lenguaje que sea gráfico, a fin de especificar y documentar un sistema de software, de un modo estándar incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema.

El lenguaje unificado que cumple con estos requerimientos, es ciertamente UML, el cual cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos.

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos. UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Además puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. Es un lenguaje que ayuda a interpretar grandes sistemas mediante gráficos o texto, obteniendo modelos explícitos que contribuyen a la comunicación durante el desarrollo, ya que al ser estándar, pueden ser interpretados por personas que no participaron en su diseño (33)

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
 - Permite realizar una verificación y validación del modelo realizado.
 - Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.
- (34)

UML propone diagramas con la finalidad de presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.7.5 Herramienta CASE. Visual Paradigm 3.4

La herramienta CASE (de las siglas en inglés: *ComputerAided Software Engineering*) seleccionada por el equipo de desarrollo es Visual Paradigm en su versión 3.4, proporciona numerosas facilidades para el trabajo colaborativo, la integración, el trabajo con modelos relacionales de Bases de Datos, modelado de procesos de negocio a través de BPMN (Business Process Modeling and Notation) así como el modelado de todos los artefactos propuestos por la metodología a través de UML, en una sola herramienta. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es una herramienta multiplataforma, fácil de instalar y actualizar. Posibilita un entorno de creación de diagramas para UML 2.x. (35)

1.7.6 Entorno Integrado de Desarrollo. NetBeans 7.1

Como entorno integrado de desarrollo se escoge el NetBeans en su versión 7.1, puesto que el NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java, además de ser un producto libre y gratuito sin restricciones de uso, también es un proyecto de código abierto de gran éxito con una gran base de usuarios. Soporta el desarrollo de todos los tipos de aplicación Java. Las funciones de este IDE son provistas por módulos donde cada uno provee una función bien definida: soporte de Java, edición o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, facilitándole al usuario comenzar a trabajar inmediatamente. La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Ofrece servicios comunes permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. (36)

Entre las características de la plataforma se encuentran:

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Marco de trabajo basado en asistentes. (36)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.7.7 Sistema Gestor de Base de Datos. PostgreSQL 8.4

PostgreSQL en su versión 8.4 es un potente sistema de gestión de bases de datos objeto-relacional (O-RDBMS), multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y que está liberado bajo la licencia Berkeley Software Distribution (BSD). Es ampliamente considerado como el sistema gestor de bases de datos de código abierto más potente del mundo; fue pionero en muchos conceptos que estuvieron disponibles en algunos sistemas de bases de datos comerciales de alto calibre como por ejemplo: control de acceso simultáneo, gestión de transacciones, puntos de seguridad; fue uno de los primeros intentos en implementar un motor de bases de datos relacional. (37)

PostgreSQL ofrece sofisticadas características como:

- Organiza los datos mediante un modelo objeto-relacional.
- Capaz de manejar procedimientos, rutinas complejas y reglas.
- Soporta tablespaces, transacciones anidadas, copias de seguridad en línea y soporte para parte de los estándares SQL 92, 99, 2003 y 2008.
- Cuenta con una API sumamente flexible propia para el trabajo con varios lenguajes de programación y procedurales como C, C++, .NET, Bash, Delphi, PL/Java, PL/Perl, PL/Tcl, PL/pgSQL, PL/Ruby, PL/PHP, PL/Python, PL/Scheme y PL/R.
- Ofrece transacciones que permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos declarados por el usuario; soporta además sobrecarga de operadores, sobrecarga de procedimientos, vistas materializables, particionamiento de tablas y datos.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información dentro de las bases de datos. (37)

Entre sus principales ventajas se encuentran:

- Instalación ilimitada: no hay costo asociado a la licencia del software.
- Soporte: existe una gran comunidad de profesionales y empresas que ofrecen soporte a PostgreSQL, de la cual el Grupo Global de Desarrollo de PostgreSQL es la principal.
- Estabilidad y confiabilidad legendaria: miles de compañías reportan que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- Extensible: el código fuente está disponible para todos sin costo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Multiplataforma: soporta alrededor de 34 plataformas incluyendo Linux y Unix en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.
- Posee herramientas gráficas de diseño y administración de bases de datos: existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Data Architect). (37)

1.8 CONCLUSIONES PARCIALES

En este capítulo se realizó un análisis sobre el desarrollo y utilización de las tecnologías de la información y las comunicaciones, en función de la preservación y conservación de documentos, la digitalización de documentos, y la asociación de metadatos. Se presentó además un estudio de algunas soluciones de digitalización significativas existentes en Cuba y el mundo haciendo énfasis en cómo se realiza el proceso de asociación de metadatos. Quedó definida como metodología de desarrollo XP por ser una metodología ágil, muy eficiente en proyectos de corto alcance y corta duración. Se seleccionó como lenguaje de programación Java por ser un lenguaje independiente de la plataforma y de código abierto. Como entorno de desarrollo se seleccionó NetBeans en su versión 7.0.1, por ser multiplataforma y por tener soporte para el lenguaje de programación Java. Como gestor de base de datos se seleccionó PostgreSQL en su versión 8.4 por ser el gestor de código abierto más potente en la actualidad. Como herramienta Case se seleccionó el Visual Paradigm en su versión 3.4, ya que es multiplataforma y permite elaborar todos los artefactos necesarios en el trabajo de investigación. El lenguaje de modelado UML en su versión 2.0 fue el seleccionado para la elaboración de los diagramas. Estas conclusiones servirán de guía para continuar con el desarrollo de la solución al problema planteado.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

2.1 INTRODUCCIÓN

En este capítulo se presenta la propuesta del sistema, planteándose el modelo del mismo, los requisitos no funcionales que debe cumplir y las historias de usuario con la estimación de esfuerzo necesario para implementar cada una de ellas. Además se describe la fase de diseño propia de la metodología de desarrollo XP, donde el diseño ofrece una guía de implementación para una historia de usuario, se define el patrón de arquitectura, el estilo arquitectónico y los patrones de diseño a utilizar para la implementación del sistema. Se procederá al diseño y descripción de la base de datos. Quedan definidas las tarjetas CRC (Colaborador - Responsabilidad – Clase) como técnica de diseño.

2.2 PROPUESTA DE SISTEMA

2.2.1 Descripción de Procesos

En el epígrafe que se muestra a continuación se describirán los diagramas de procesos de la solución propuesta. Para esto se seleccionarán los dos principales procesos que aparecen en el sistema, el proceso de configurar tipos documentales y el proceso de asociar metadatos.

2.2.2 Configurar Tipo Documental

El proceso de configuración de un tipo documental comienza cuando el usuario selecciona la opción inscribir tipo documental, luego introduce los datos que va a tener la unidad documental (nombre, descripción), y por último el sistema almacena los datos registrados.

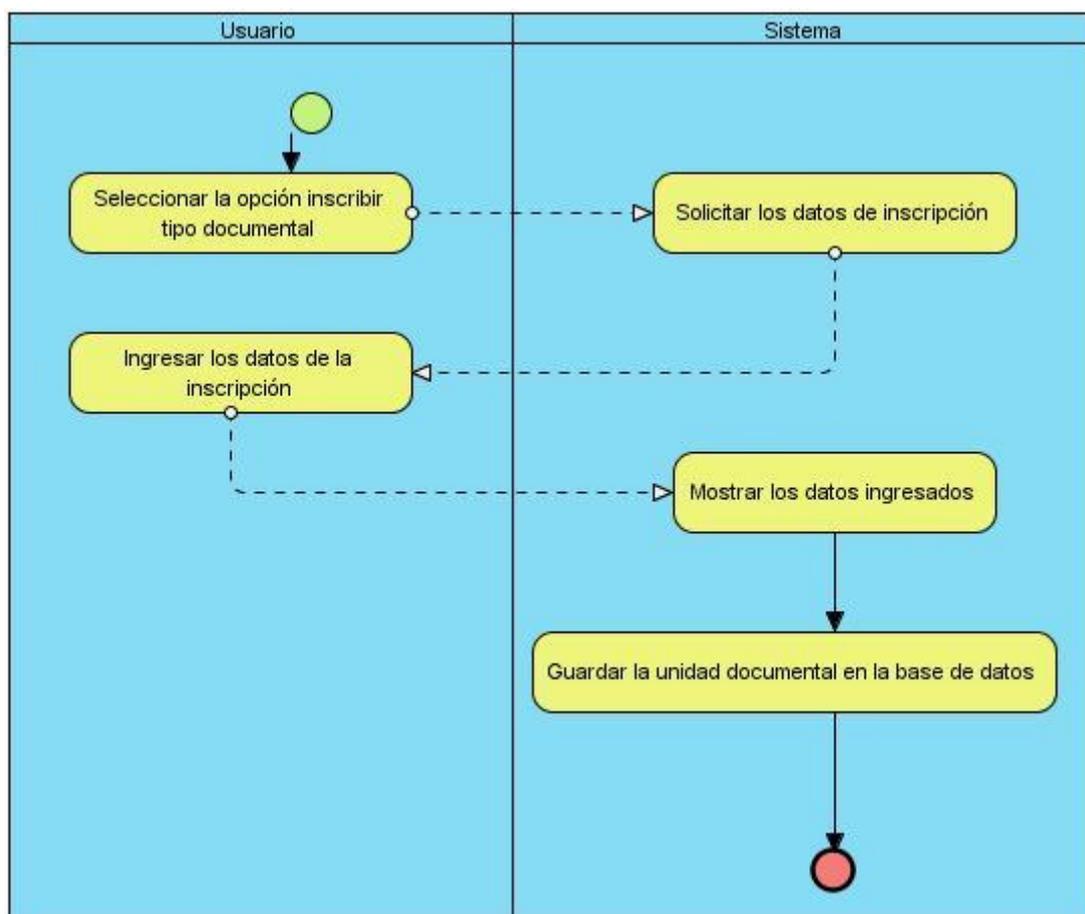


Figura 2: Diagrama de Proceso de configuración de un tipo documental.

Para consultar las tareas de los procesos Asociar metadatos, Rectificar tipo documental, Cancelar tipo documental consultar, ver **Anexo 1**.

2.2.3 Asociación de Metadatos

El proceso de asociación de metadatos comienza cuando el usuario selecciona la opción asociar metadatos, luego el sistema muestra un listado de los tipos documentales a los que se le pueden asociar metadatos, por último el usuario registra los valores de los campos correspondientes a los tipos documentales.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

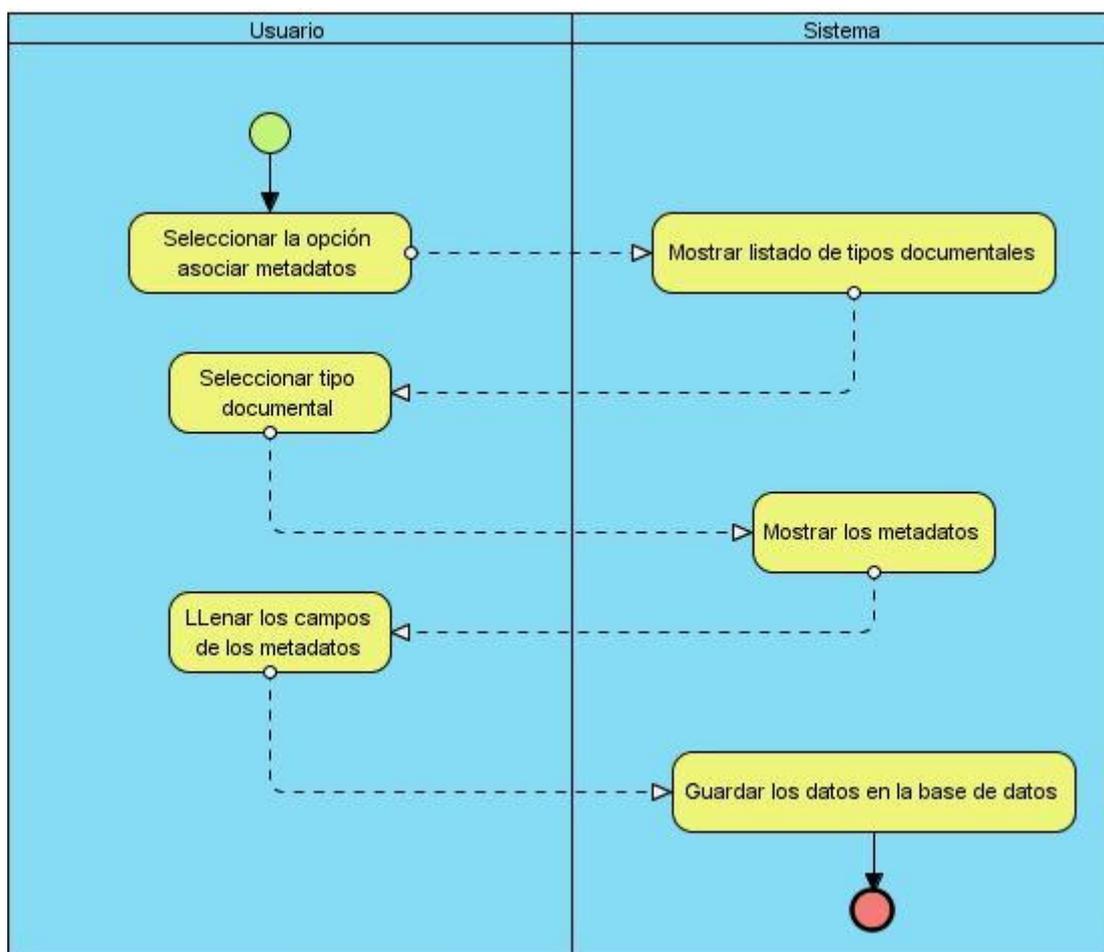


Figura 3: Diagrama de proceso de asociación de metadatos.

2.2.4 Personas relacionadas con el sistema

Se define como persona relacionada con el sistema aquel que obtiene un resultado de la ejecución de uno o varios procesos en el mismo y aquellos que se encuentran involucrados en la administración de dichos procesos. En este caso las personas relacionadas con el sistema son:

Personas relacionadas con el sistema	Justificación
Usuario	Es la persona que interactúa con el sistema teniendo acceso a todas las diferentes funcionalidades que el mismo brinda. Se encarga de Asociar metadatos, Gestionar tipo documental, Consultar documento digital y Anexar documento a expediente.

Tabla 1: Personas relacionadas con el sistema.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

2.2.5 Especificación de requisitos no funcionales del sistema

Los requisitos no funcionales son cualidades o propiedades que debe tener el producto con el fin de hacerlo confiable, usable, rápido y atractivo. Los presentes en este sistema son:

Apariencia o interfaz externa

La aplicación debe tener una interfaz sencilla y amigable, fácil de utilizar por el usuario, que le permita al mismo trabajar de forma cómoda y ágil.

Usabilidad

La aplicación gracias al ambiente sencillo y fácil de usar que posee podrá ser utilizada por usuarios con conocimientos básicos en computación o sistemas informáticos.

Soporte

El software deberá ser actualizado por los administradores y se debe garantizar el mantenimiento de la aplicación.

Software

El sistema requiere para su funcionamiento:

- Por parte del cliente: Sistema operativo Windows (98 o superior) o Linux.
- Por parte del servidor: Servidor de aplicaciones GlassFish Server 3.1. Servidor de base de datos PostgreSQL.

Hardware

- Estaciones de trabajo clientes: Las PC clientes deberán contar como mínimo con 128 MB de memoria RAM, con velocidad de procesamiento 450MHz o superior.
- Servidor: La PC servidora deberá contar como mínimo con 512 MB de memoria RAM, con velocidad de procesamiento 2GHz o superior.

Rendimiento

- El tiempo de respuesta brindado por el sistema será menor de 3 segundos.

Portabilidad

- El sistema deberá ser independiente de la plataforma.

2.3 FASE DE EXPLORACIÓN Y PLANIFICACIÓN

2.3.1 Exploración

La metodología XP define como primera la fase de exploración. Esta es la fase donde los clientes plantean a grandes rasgos su propuesta de trabajo, al mismo tiempo el equipo de desarrollo se familiarizará con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Es donde queda definido el alcance real del proyecto y sobre todo se

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

definen las historias de usuario (HU), las cuales describen brevemente las características que el sistema debe poseer y el cliente establece la prioridad a cada HU.

2.3.1.1 Historias de Usuario

Las historias de usuario (en lo adelante HU) son la técnica utilizada por XP para especificar las características y funcionalidades requeridas para el sistema que se desarrollará, las mismas son redactadas por el cliente, el mismo le asigna una prioridad o valor a cada HU, éstas no deben ser descritas en más de tres líneas, el contenido que ellas abarcan debe ser concreto y sencillo, escritas en el lenguaje del usuario. El equipo de desarrollo evalúa cada HU y le asigna un costo, el cual se mide en semanas de desarrollo. A medida que avance el desarrollo del sistema, el cliente puede agregar HU, modificar el valor de historias existentes, dividir las o eliminarlas.

Clasificación de la prioridad de las HU

- **Alta:** Se le otorga esta prioridad a las HU que el cliente define como funcionalidades principales en el desarrollo del sistema y fundamentales para el control general del sistema.
- **Media:** Se le otorga a las HU que el cliente define como funcionalidades menos significativas, sin que estas afecten el desarrollo del sistema.
- **Baja:** Se le otorga a las HU que se definen como funcionalidades de ayuda a la estructura y control de elementos asociados al desarrollo del sistema.

Las HU se representarán mediante tablas divididas por las siguientes secciones:

- Número de la HU.
- Nombre de la HU.
- Usuario involucrado en el desarrollo de la HU.
- Prioridad de la HU.
- Complejidad de desarrollo.
- Estimación del tiempo de desarrollo de la HU.
- Iteración en la que será desarrollada la HU.
- Descripción de la HU.
- Observaciones realizadas a la HU.
- Prototipo de interfaz si se aplica.

A continuación se muestran las tablas de HU con sus respectivas descripciones e interfaces.

No: 1	Nombre: Configurar Metadatos de Tipo Documental
--------------	--

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Usuarios	Dayami Darias Dávila
Prioridad: Alta	Complejidad: alta
Estimación: 2 días.	Iteración: 1
Descripción	Permite al usuario configurar los diferentes tipos documental.
Observaciones	Da cumplimiento al requisito funcional 1.
Prototipo de Interfaz	

Tabla 2: HU Configurar Tipo Documental.

No: 2	Nombre: Configurar Metadatos
Usuarios	Dayami Darias Dávila
Prioridad: Alta	Complejidad: Alta
Estimación: 2 días.	Iteración: 1
Descripción	Permite al usuario configurar los metadatos que va a tener un tipo documental determinado.
Observaciones	Da cumplimiento al requisito funcional 2.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Prototipo de Interfaz	
------------------------------	--

Tabla 3: HU Configurar Metadatos.

No: 3	Nombre: Buscar Coincidencias Documentales
Usuarios	Dayami Darias Dávila
Prioridad: baja	Complejidad: alta
Estimación: 3 días.	Iteración: 1
Descripción	Permite al usuario buscar los tipos documentales que poseen coincidencias en cuanto al valor de sus metadatos.
Observaciones	Da cumplimiento al requisito funcional 3.
Prototipo de Interfaz	

Tabla 4: HU Buscar Coincidencias Documentales.

No: 4	Nombre: Modificar Tipo Documental
Usuarios	Dayami Darias Dávila

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

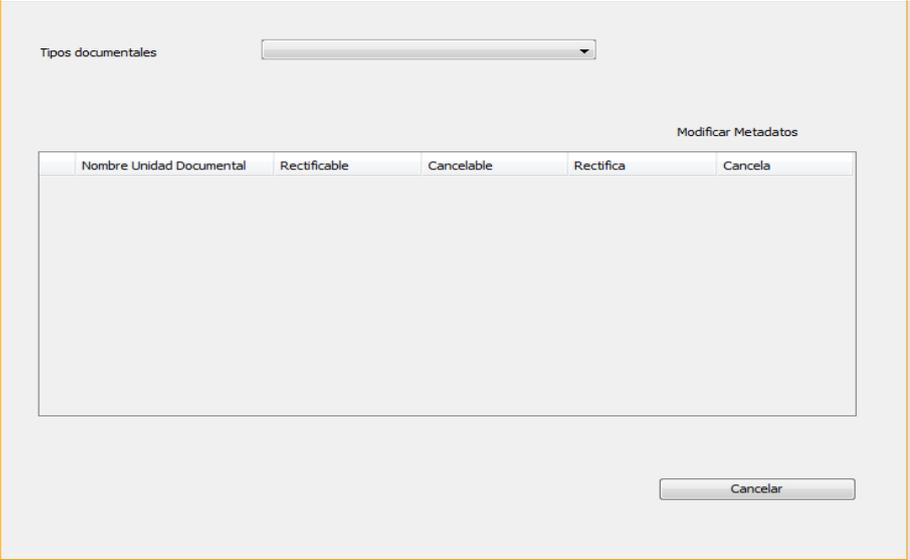
Prioridad: Alta	Complejidad: Alta
Estimación: 2 días.	Iteración: 1
Descripción	Permite al usuario modificar el valor de los metadatos de un tipo documental determinado.
Observaciones	Da cumplimiento al requisito funcional 4.
Prototipo de Interfaz	

Tabla 5: Modificar Tipo Documental.

No: 5	Nombre: Consultar documento digital.
Usuarios	Dayami Darias Dávila
Prioridad: media	Complejidad: media
Estimación: 2 días.	Iteración: 1
Descripción	Permite al usuario consultar una determinada imagen digital.
Observaciones	Da cumplimiento al requisito funcional 5.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Prototipo de Interfaz	<p>Nombre del documento: <input type="text"/></p> <table border="1"> <thead> <tr> <th>Id Documento</th> <th>Nombre Documento</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Mostrar"/></p>	Id Documento	Nombre Documento		
Id Documento	Nombre Documento				

Tabla 6: Consultar documento digital.

No: 6	Nombre: Rectificar tipo documental.														
Usuarios	Dayami Darias Dávila														
Prioridad: Alta	Complejidad: alta														
Estimación: 3 días.	Iteración: 1														
Descripción	Permite al usuario rectificar un tipo documental determinado.														
Observaciones	Da cumplimiento al requisito funcional 6.														
Prototipo de Interfaz	<p>Tipo Documental: <input type="text"/></p> <table border="1"> <thead> <tr> <th>Metadatos</th> <th>Valores</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Buscar Coincidencias"/></p> <table border="1"> <thead> <tr> <th>Nombre Tipo Documental</th> <th>Puede ser rectificada</th> <th>Puede ser cancelada</th> <th>Rectifica</th> <th>Cancela</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="Rectificar"/> <input type="button" value="Cancelar"/> </p>	Metadatos	Valores			Nombre Tipo Documental	Puede ser rectificada	Puede ser cancelada	Rectifica	Cancela					
Metadatos	Valores														
Nombre Tipo Documental	Puede ser rectificada	Puede ser cancelada	Rectifica	Cancela											

Tabla 7: Rectificar tipo documental.

No: 7	Nombre: Cancelar tipo documental.
Usuarios	Dayami Darias Dávila

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

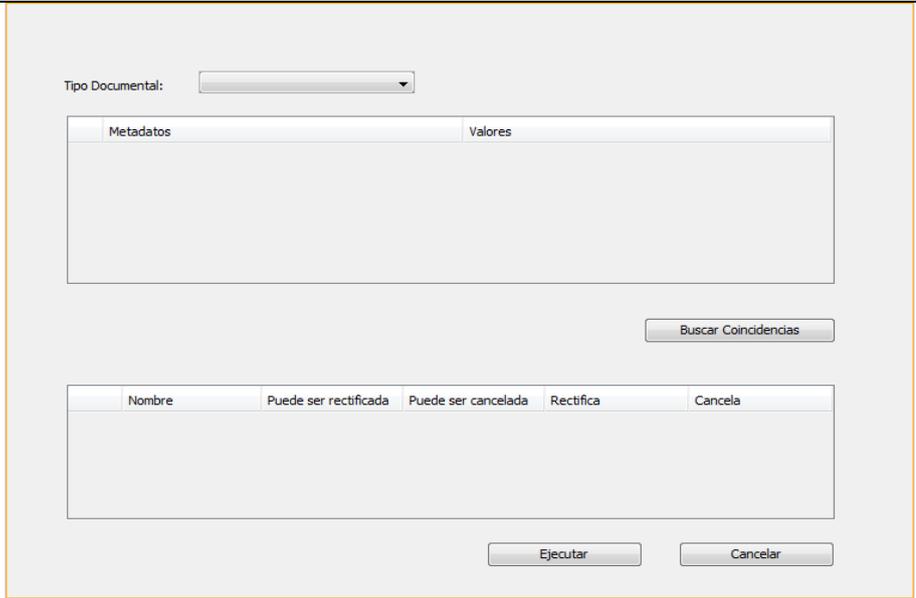
Prioridad: Alta	Complejidad: alta
Estimación: 3 días.	Iteración: 1
Descripción	Permite al usuario cancelar un tipo documental determinado.
Observaciones	Da cumplimiento al requisito funcional 7.
Prototipo de Interfaz	

Tabla 8: Cancelar tipo documental.

No: 8	Nombre: Anexar tipo documental a expedientes.
Usuarios	Dayami Darías Dávila
Prioridad: baja	Complejidad: alta
Estimación: 3 días.	Iteración: 1
Descripción	Permite al usuario anexar un expediente a un tipo documental determinado.
Observaciones	Da cumplimiento al requisito funcional 8.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Prototipo de Interfaz

Código Expediente: Número Expediente:

Código Expediente Número Expediente

Tipo Documental:

Nombre	Puede ser rectificada	Puede ser cancelada	Rectifica	Cancela
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Tabla 9: Anexar tipo documental a expedientes.

2.3.2 Planificación

La planificación es la fase que plantea XP como un permanente diálogo entre el cliente y los desarrolladores del proyecto. En esta fase los desarrolladores realizan una estimación del esfuerzo que costará implementar de cada una de las HU. Luego de realizar la estimación, se organizan las HU en las iteraciones correspondiente, según la prioridad establecida por el cliente y el tiempo de desarrollo de cada una.

2.3.2.1 Estimación de esfuerzo por Historias de usuarios

Historias de Usuario	Estimación (días)
Configurar Metadatos de Tipo Documental.	2
Configurar Metadatos.	2
Buscar Coincidencias Documentales.	3
Modificar Tipo Documental.	2
Consultar documento digital.	2
Rectificar tipo documental.	3
Cancelar tipo documental.	3
Anexar tipo documental a expedientes.	3

Tabla 10: Estimación de esfuerzo por HU.

2.3.2.2 Plan de iteraciones

Luego de ser identificadas las HU y haber realizado la estimación de esfuerzo de cada una de ellas, se procede a realizar la planificación de la etapa de implementación del

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

sistema, elaborando el plan de iteraciones. En este plan se especifica la prioridad con que se irán desarrollar las HU organizadas por iteraciones.

Una iteración no es más que un mini-proyecto donde se obtiene un resultado del software con valor para el cliente. Es preciso aclarar que no se obtiene la versión completa del producto, ya que este concluye al finalizar la última iteración.

Iteración 1

Esta iteración tiene como objetivo la implementación de las HU de prioridad alta y prioridad media. Propiciando al finalizar tener implementadas las peticiones del cliente descritas en las historias, contando con las funcionalidades principales para el funcionamiento del sistema.

Iteración 2

En esta iteración serán implementadas las HU de prioridad baja, las mismas tienen como propósito brindarle comodidad al cliente en la gestión de otras tareas asociadas a las de prioridad alta. Al terminar esta iteración se obtendrá una versión 1.0 del producto final, listo para poner en funcionamiento y ser evaluado.

2.3.2.3 Plan de duración de las iteraciones

Continuando con el desarrollo de la metodología XP, se pasa a crear el plan de duración de las iteraciones. Este plan se encarga de detallar el orden en que se implementarán las HU dentro de cada iteración, así como la estimación completa de la misma.

Iteraciones	Orden de las HU a implementar	Duración de las iteraciones
Iteración 1	Configurar Metadatos de Tipo Documental. Configurar Metadatos. Modificar Tipo Documental. Consultar Documento Digital. Rectificar Tipo Documental. Cancelar Tipo Documental.	3 semanas
Iteración 2	Buscar Coincidencias Documentales. Anexar Tipo Documental a Expedientes.	1 semana

Tabla 11: Plan de duración de las iteraciones.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

2.3.3 Plan de entregas

El plan de entregas establece que HU serán agrupadas para conformar una entrega, y el orden de las mismas, así como la fecha fin de cada iteración. Este plan acopla las funcionalidades referentes a un mismo tema en el módulo que serán implementadas.

Historia de Usuario	Fin de la 1ra iteración 3ra semana de Abril	Fin de la 2ra iteración 1ra semana de Mayo
Configurar Metadatos de Tipo Documental	V 0.1	Finalizado
Configurar Metadatos	V 0.1	Finalizado
Modificar Tipo Documental	V 0.1	Finalizado
Consultar Documento Digital	V 0.1	Finalizado
Rectificar Tipo Documental.	V 0.1	Finalizado
Cancelar Tipo Documental	V 0.1	Finalizado

Tabla 12: Plan de duración de entregas.

2.4 ESTILO ARQUITECTÓNICO CLIENTE – SERVIDOR

La arquitectura cliente-servidor es un modelo en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Admite al usuario en una máquina llamada cliente, requerir algún tipo de servicio de una máquina a la que está conectado, llamado servidor, mediante una red de área local o una red de área mundial.

Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los servicios solicitados pueden ser peticiones de datos de una base de datos, de información contenida en archivos o los archivos en sí mismos. Este tipo de arquitectura es la base para todas las aplicaciones que hoy se pueden encontrar en internet y además, en la mayoría de la intranet corporativa que hoy existen en muchas de las empresas. (38)

2.5 ARQUITECTURA DEL SISTEMA

La arquitectura de software alude a la estructura general del software y las formas en que la estructura proporciona una integridad conceptual para un sistema. En su forma más

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

simple, la arquitectura es la estructura u organización de los componentes del programa, la manera en que estos componentes interactúan, y la estructura de datos que utilizan los componentes. En sentido más amplio, sin embargo, los componentes pueden generalizarse para representar elementos importantes del sistema y sus interacciones. (39)

La arquitectura del sistema tiene como principales características:

- Cliente/Servidor
- Distribuida en n capas

La arquitectura Cliente/Servidor proporciona considerables mejoras desde el punto de vista de la portabilidad de la aplicación, escalabilidad, robustez y reutilización de código. Asimismo facilita las tareas de migración o cambios en el sistema gestor de base de datos.

Las principales ventajas son:

- **Recursos centralizados:** Debido a que el servidor es el centro de la red, puede administrar los recursos que son comunes a todos los usuarios, por ejemplo: una base de datos centralizada se utilizaría para evitar problemas provocados por datos contradictorios y redundantes.
- **Seguridad mejorada:** Ya que la cantidad de puntos de entrada que permite el acceso a los datos no es importante, por lo que se puede ocultar el código.
- **Administración al nivel del servidor:** Ya que los clientes no juegan un papel importante en este modelo, requieren menos administración.
- **Red escalable:** Gracias a esta arquitectura, es posible quitar o agregar clientes sin afecta el funcionamiento de la red y sin la necesidad de realizar mayores modificaciones.

También tiene las siguientes desventajas:

- Costo elevado: debido a la complejidad técnica del servidor.
- Un eslabón débil: el servidor es el único eslabón débil en la red de cliente/servidor, debido a que toda la red está construida en torno a él. Afortunadamente, el servidor es altamente tolerante a los fallos.

El sistema está compuesto principalmente por dos aplicaciones cliente y servidor, la cliente (componente-metadatos) se adiciona como componente a las aplicaciones que lo utilizan, interactúa con un .jar desplegado en el servidor (componente-metadatos-ejb) que

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

proporciona los servicios adecuados para la realización de la lógica de negocio de la aplicación. La comunicación entre ambas se garantiza a través de EJB.

Según Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (39)



Figura 4: Vista de la arquitectura (Cliente).

La capa de presentación contiene los formularios con los que interactúan los usuarios finales del sistema y permiten la visualización de toda la información que estos necesitan y las acciones que gestionan las funcionalidades de los formularios y garantizan la comunicación con los gestores de negocio en el cliente para la invocación de la lógica de negocio.

La capa de negocio contiene los gestores de negocio del cliente que utilizan las acciones, además la FachadaGestoresRemotos que sirve de fachada para la comunicación entre los gestores clientes y servidor, es donde se procesan los datos del cliente.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA



Figura 5: Vista de la arquitectura (Servidor).

Existe además la capa de negocio en el servidor que contiene los gestores de negocios del servidor donde se contiene la mayor parte de la implementación de la lógica de negocio invocada por los gestores de negocio en el cliente. Responsable de las tareas propias de la aplicación, implementa la lógica de la aplicación y aplica las reglas de negocio y las entradas del usuario.

La capa de acceso a datos contiene los gestores de acceso a datos y las clases persistentes para la gestión de los datos almacenados en la capa de datos. Es responsable de la gestión y almacenamiento permanente de los datos.

La capa de datos contiene los datos gestionados por la aplicación.

Los componentes lo constituyen los gestores de negocio en el servidor que son Beans de sesión remotos sin estado, los gestores de acceso a datos que son Beans de sesión locales sin estado y las clases persistentes que son Beans de entidades. Los Beans dirigidos por mensajes existen también para el paso de mensajes entre las aplicaciones cliente y servidor.

2.6 PATRONES DE DISEÑO

Brad Appleton define un patrón de diseño de la siguiente manera: Un patrón es una semilla de conocimiento, la cual tiene un nombre y transporta la esencia de una solución probada a un problema recurrente dentro de cierto contexto en medio de intereses y competencias. Dicho de otro modo un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño en particular dentro de un contexto específico y en medio de fuerzas que pueden tener un impacto en la manera que se aplica y utiliza el patrón. (39)

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

La finalidad de cada patrón de diseño es proporcionar una descripción que le permita determinar al diseñador:

- Si el patrón es aplicable al trabajo actual.
- Si el patrón se puede reutilizar.
- Si el patrón puede servir como guía para desarrollar un patrón similar, pero diferente en cuanto a la funcionalidad o estructura.

Los patrones más importantes a usar en el desarrollo son los siguientes:

Según el libro GOF (Gang-Of-Four Book) existen 3 clasificaciones de patrones:

De Creación: abstraen el proceso de creación de instancias.

- **Solitario / Singleton** para la creación de una instancia de las clases que sean necesarias, como los gestores del negocio y en las clases internacionalización.

Estructurales: se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.

- **Fachada / Facade** para proporcionar una interfaz para la creación de familias de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. El patrón se utilizó para la creación de la clase FachadaGestoresRemotos para la facilitar la conexión de los gestores cliente y servidor.

De Comportamiento: atañen a los algoritmos y a la asignación de responsabilidades entre objetos.

- **Iterador / Iterator** para proporcionar una forma coherente de acceder secuencialmente a los elementos de una colección, independientemente del tipo de colección subyacente. El patrón se utilizó para recorrer las estructuras en las clases factorías del servidor.

Los **patrones GRASP** (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Los mismos se utilizan en todas las clases gestores: cliente, servidor y acceso a datos.

- **Bajo Acoplamiento** para estimular la asignación de responsabilidades de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. El Bajo Acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables. Este patrón se utiliza en la comunicación entre los gestores de negocio del servidor y los gestores de acceso a datos, la cual se realiza a través de las interfaces remotas y locales.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- **Experto** para que cada objeto realice la funcionalidad de acuerdo a la información que domina, la cuestión a la hora de diseñar es asignar responsabilidades a la clase que mayor información posee para cumplir con dicha tarea. Este patrón se utiliza fundamentalmente en las clases de acceso a datos que son clases que contienen los datos para realizar una funcionalidad determinada, o sea que contienen la información para cumplir con una responsabilidad que otra clase no podría.
- **Creador** para encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Brinda soporte a un bajo acoplamiento lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Este patrón se utiliza en las factorías y en las clases acciones del cliente, siempre que se necesite crear una instancia de una determinada clase.
- **Alta cohesión** para brindar una alta cohesión funcional cuando los elementos de un componente (clase, por ejemplo) "colaboran para producir algún comportamiento bien definido. En el sistema se tiene bien definido las responsabilidades de cada clase, por ejemplo, se definen clases del negocio por cada caso de uso, para evitar la sobrecarga de funcionalidades en una clase, de esta forma se evita el exceso de carga de trabajo de una determinada clase, luego de la anterior explicación se puede llegar a la conclusión de que este patrón es utilizado en todas las clases de la aplicación.
- **Controlador** para asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:
 - El "sistema" global (controlador de fachada).
 - La empresa u organización global (controlador de fachada).
 - Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).

Este patrón se utiliza fundamentalmente en los gestores del negocio, tanto del cliente como del servidor, como en los gestores de acceso de datos.

2.7 TARJETAS COLABORADOR - RESPONSABILIDAD – CLASE (CRC)

La fase de diseño en la metodología XP sigue de manera rigurosa el principio de mantenerlo simple, como parte de la misma se propone el modelado de la tarjetas CRC, proporcionando un medio sencillo para identificar y organizar las clases relevantes para los requisitos del sistema. Donde el objetivo de estas tarjetas es desarrollar una representación organizada de las clases.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Las responsabilidades en estas son los atributos y las operaciones relevantes para la clase, de manera más simple una responsabilidad es cualquier cosa que una clase sabe o hace. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad, este implica en general ya sea una solicitud de información o de alguna acción. (39)

Las tarjetas CRC están divididas en tres secciones:

- ✓ Nombre del componente.
- ✓ Responsabilidades.
- ✓ Colaboradores.

Componente Configurar Metadatos de Tipo Documental	
Responsabilidad	Colaborador
adicionarTipoDocumental()	GestorNCTipoDocumental GestorMarcoTrabajo GestorNSTipoDocumental
persistirTiposDocumentales()	GestorADTipoDocumental EDTipoDocumental

Tabla 13: Tarjetas CRC Configurar metadatos de tipo documental.

Componente Configurar Metadatos	
Responsabilidad	Colaborador
llenarComboTiposDocumentales()	GestorNCMetadatos
llenarComboTiposDatos()	GestorMarcoTrabajo
adicionarMetadatosTipoDocumental()	GestorNSMetadatos
salvarMetadatos()	GestorADMetadatos EDMetadatos

Tabla 14: Tarjetas CRC Configurar metadatos.

Componente Asociar Metadatos	
Responsabilidad	Colaborador
cargarTiposDocumentales()	GestorNCMetadatos
llenarGridMetadatos()	GestorMarcoTrabajo
llenarCamposMetadatos()	GestorNSMetadatos GestorADMetadatos
onSiguienteResultado()	EDMetadatos
onAnteriorResultado()	

Tabla 15: Tarjetas CRC Asociar metadatos.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Componente Modificar Metadatos	
Responsabilidad	Colaborador
modificarMetadatos()	GestorNCMetadatos
onSiguienteResultado()	GestorMarcoTrabajo
onAnteriorResultado()	GestorNSMetadatos GestorADMetadatos EDMetadatos

Tabla 16: Tarjetas CRC Modificar metadatos.

Componente Rectificar Tipo Documental	
Responsabilidad	Colaborador
llenarComboTiposDocumentales()	GestorNCTipoDocumental
llenarMetadatosCoincidencias()	GestorMarcoTrabajo GestorNSTipoDocumental
llenarGridTipoDocumentales()	GestorADTipoDocumental
rectificarTipoDocumental()	EDTipoDocumental
onSiguienteResultado()	
onAnteriorResultado()	

Tabla 17: Tarjetas CRC Rectificar tipo documental.

Componente Cancelar Tipo Documental	
Responsabilidad	Colaborador
llenarComboTiposDocumentales()	GestorNCTipoDocumental
cancelarTipoDocumental()	GestorMarcoTrabajo GestorNSTipoDocumental
onSiguienteResultado()	GestorADTipoDocumental
onAnteriorResultado()	EDTipoDocumental

Tabla 18: Tarjetas CRC Cancelar tipo documental.

Componente Buscar Coincidencias Documentales	
Responsabilidad	Colaborador
CargarTiposDocumentales()	GestorNCTipoDocumental
LlenarMetadatosCoincidencias()	GestorMarcoTrabajo
LlenarGridTipoDocumentales()	GestorNSTipoDocumental
onSiguienteResultado()	GestorADTipoDocumental

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

onAnteriorResultado()	EDTipoDocumental
-----------------------	------------------

Tabla 19: Tarjetas CRC Buscar coincidencias documentales.

Componente Modificar Tipo Documental	
Responsabilidad	Colaborador
CargarTiposDocumentales()	GestorNCTipoDocumental GestorMarcoTrabajo
CancelarTipoDocumental()	GestorNSTipoDocumental GestorADTipoDocumental
onSiguienteResultado()	EDTipoDocumental
onAnteriorResultado()	acciónModificarMetadatos

Tabla 20: Tarjetas CRC Modificar tipo documental.

2.8 DIAGRAMA DE PAQUETES

El diagrama de paquetes muestra cómo el sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Los Paquetes están organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los mismos. (40)

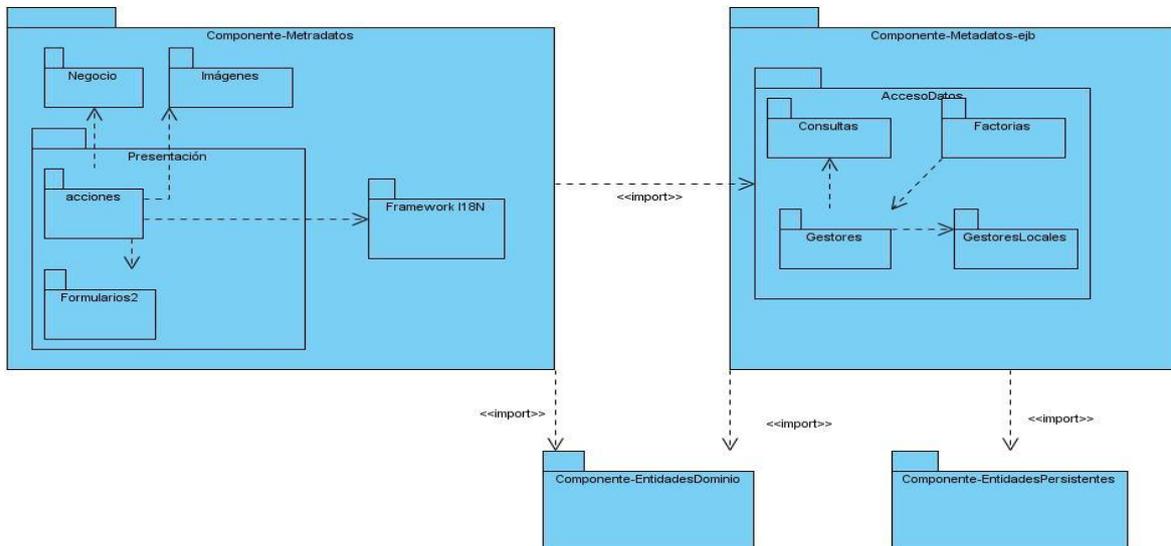


Figura 6: Diagrama de Paquetes.

Componente – Metadatos

- **frameworkI18N:** Contiene las clases que se encargan de gestionar la internacionalización del sistema software de manera que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- **presentación:** Contiene las clases y formularios que se encargan de presentar los datos al usuario.
- **negocio:** Contiene las clases que gestionan la lógica de negocio en el cliente. Utilizan los gestores de negocio en el servidor para completar las acciones del sistema.
- **imágenes:** Contiene los iconos e imágenes utilizados en la aplicación.

Componente – Metadatos – ejb

- **acceso a datos:** Almacena todos los paquetes que contienen las clases que gestionan el acceso a datos de la aplicación.
- **consultas:** Contiene los ficheros de consulta .xml utilizadas para la gestión de los datos.
- **factoría:** Contiene clases de factorías que convierten las entidades persistentes a entidades de dominio para el paso de información entre las capas.
- **gestores locales:** Contiene las clases interfaces que implementan los gestores de acceso a datos.
- **gestores:** Contienen las clases que gestionan el acceso a datos.
- **fachada:** Contiene las clases interfaces que implementan los gestores de negocio publicados en el servidor.
- **gestores:** Contiene las clases que implementan la lógica de negocio en el servidor.
- **Componentes - entidades persistentes:** Contiene las clases entidades para el acceso a datos del servidor.
- **Componente - entidades dominio:** Contiene las clases entidades de dominio de la aplicación.

2.9 DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos es una de las tareas más importantes en la construcción de un sistema que la utilice. Este es fundamental, puesto que tiene como objetivo brindar la persistencia al modelado descrito anteriormente, elaborándose con el fin de acceder a la información de manera eficiente. Quedando definido el modelo de datos siguiente:

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

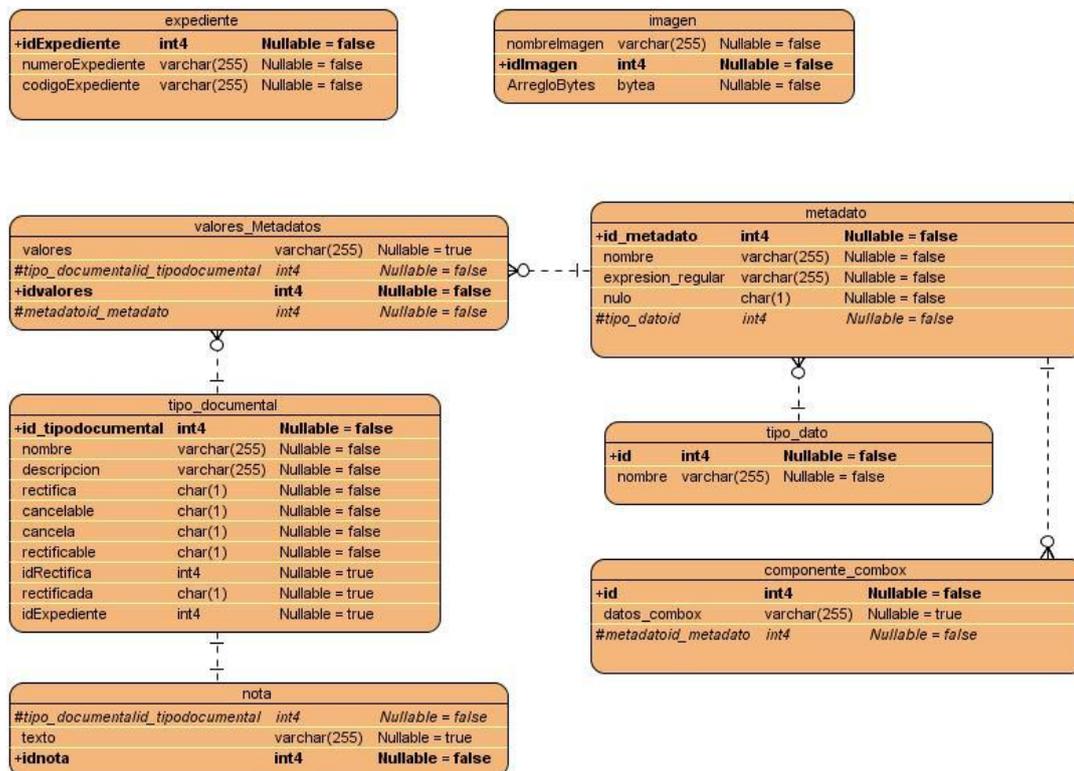


Figura 7: Modelo de datos.

- La entidad **tipo_documental** se encarga de almacenar los datos del tipo documental registrado.
- La entidad **metadato** se encarga de almacenar la configuración de los metadatos que se asocien.
- La entidad **valores_metadatos** se encarga de almacenar los valores de los metadatos gestionados previamente.
- La entidad **tipo_dato** se encarga de almacenar todos los distintos tipos de datos existentes, además de relacionarse con la entidad metadatos, de forma tal que un tipo de dato tenga muchos metadatos.
- La entidad **componente_combobox** almacena los valores de los metadatos que están previamente definidos por el usuario, y que a su vez puede ser seleccionado más de un valor para dicho metadato.
- La entidad **nota** se encarga de guardar la información que se describe en la nota.

2.10 CONCLUSIONES PARCIALES

En este capítulo quedó detallada la propuesta del sistema y los requisitos no funcionales del mismo. Se describieron las HU que son los artefactos generados en la fase exploración y planificación de la metodología XP, se realizó una estimación de esfuerzo por cada una de las historias, y quedó definido el plan de duración de entregas. Además se definió la arquitectura y estilo arquitectónico presente para el desarrollo del sistema,

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

detallándose los patrones de diseños aplicados. También quedaron definidos los componentes del sistema en cuestión, así como las tarjetas CRC que son los artefactos generados en la fase de diseño de la metodología XP. Realizándose además el diseño y descripción de la base de datos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.1 INTRODUCCIÓN

En el presente capítulo se describen las tareas de la ingeniería pertenecientes a las historias de usuarios de prioridad alta definidas en el capítulo anterior. Además se muestra el desarrollo de la implementación del componente durante las diferentes iteraciones definidas por la metodología de desarrollo XP. También se realiza el diagrama de componentes y el diagrama de despliegue. Por último para validar la solución propuesta se realizaran las pruebas correspondientes a todas las funciones implementadas durante el desarrollo de la aplicación.

3.2 FASE DE IMPLEMENTACIÓN

La metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al finalizar cada iteración un producto funcional que debe ser probado y mostrado al cliente. En esta fase se realiza la implementación de cada una de los HU durante el transcurso de las iteraciones, creándose las tareas de la ingeniería para ayudar a organizar la implementación exitosa de la HU, estas tareas son asignadas a los programadores que son los responsables de la implementación. Estas tareas pueden escribirse en un lenguaje técnico, al contrario de las HU que son escritas en el lenguaje del cliente. (39)

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- Número de la tarea.
- Número de la historia de usuario a la que pertenece la tarea.
- Nombre de la tarea.
- Tipo de la tarea, puede ser de Desarrollo, Corrección, Mejora u Otra (especificar).
- Puntos de estimación, que son el tiempo estimado en semanas que se le asignará para su desarrollo.
- Fecha de inicio del desarrollo de la tarea.
- Fecha de fin del desarrollo de la tarea.
- Programador responsable.
- Descripción de la tarea.

3.2.1 Iteraciones

Teniendo en cuenta la planificación realizada anteriormente, quedaron definidas 2 iteraciones de desarrollo sobre el sistema, con el objetivo de obtener un producto con

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

todas las características deseadas para ser utilizado. Detallándose cada una de las iteraciones a continuación.

3.2.2 Primera iteración

En la 1ra iteración se implementaron las historias de usuario de prioridad alta y media, con el fin de obtener una versión del producto que cuente con las funcionalidades críticas para el funcionamiento del sistema. A continuación se detallan los componentes abordados en la primera iteración.

Componente.	Historia de usuario.	Tiempo de implementación.	
		Estimación.	Real.
Gestión de Tipo Documental	Configurar Metadatos de Tipo Documental	2	2
	Rectificar Tipo Documental.	2	2
	Cancelar Tipo Documental.	1	2
	Modificar Tipo Documental.	2	3
Gestión de Metadatos.	Configurar Metadatos.	1	2
	Asociar Metadatos.	3	3
	Modificar Metadatos	2	2

Tabla 21: Componentes abordados en la primera iteración.

3.2.3 Segunda iteración

En la 2da iteración se implementaron las historias de usuario de prioridad baja, con el fin de brindarle comodidad al cliente en la gestión de otras tareas asociadas a las de prioridad alta. A continuación se detallan los componentes abordados en la primera iteración.

Componente.	Historia de usuario.	Tiempo de implementación.	
		Estimación.	Real.
Gestión de Tipo Documental.	Buscar coincidencias documentales	1	1
	Anexar tipos documentales a expedientes	1	2
	Consultar documento digital	1	1

Tabla 22: Componentes abordados en la segunda iteración.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.2.4 Tareas de la ingeniería

Se describirán las tareas de la ingeniería pertenecientes a las HU de prioridad alta, efectuadas para cada uno de estos componentes implementados.

Tareas	
No. Tarea: 1	No. HU: 1
Nombre de la tarea: Configurar Metadatos de Tipo Documental.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se configurarán los campos del tipo documental, validando que se llenen todos los campos del tipo documental, luego el sistema guardará los valores de los campos del tipo documental.	

Tabla 23: Tareas de la ingeniería de la primera iteración, Configurar metadatos de tipo documental.

Tareas	
No. Tarea: 2	No. HU: 2
Nombre de la tarea: Rectificar tipo documental.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se realizará una búsqueda de coincidencias de tipos documentales, luego de que el sistema realice la búsqueda se seleccionará el tipo documental que se desee rectificar, para luego rectificarlo.	

Tabla 24: Tareas de la ingeniería de la primera iteración, Rectificar tipo documental.

Tareas	
No. Tarea: 3	No. HU: 3
Nombre de la tarea: Cancelar tipo documental.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se realizará una búsqueda de coincidencias de tipos documentales, luego de que el sistema realice la búsqueda se seleccionará el tipo documental que se desee cancelar, para luego cancelarlo.	

Tabla 25: Tareas de la ingeniería de la primera iteración, Cancelar tipo documental.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Tareas	
No. Tarea: 4	No. HU: 4
Nombre de la tarea: Modificar tipo documental.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se realizará una búsqueda de todos los tipos documentales existentes, luego de obtener el listado de los tipos documentales se seleccionará uno para modificarle los metadatos deseados.	

Tabla 26: Tareas de la ingeniería de la primera iteración, Modificar tipo documental.

Tareas	
No. Tarea: 5	No. HU: 5
Nombre de la tarea: Configurar Metadatos.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se configuraran los metadatos asociado a un tipo documental determinado, verificando que no existan campos vacíos, para luego salvar la configuración en la base de datos.	

Tabla 27: Tareas de la ingeniería de la primera iteración, Configurar metadatos.

Tareas	
No. Tarea: 6	No. HU: 6
Nombre de la tarea: Asociar Metadatos.	
Tipo de tarea: Desarrollo.	Puntos de estimación:
Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se selecciona el tipo documental a la que se le va a asociar metadatos, luego se introducen los valores de los metadatos, verificando que no se dejen campos en blanco para luego salvar los valores en la base de datos.	

Tabla 28: Tareas de la ingeniería de la primera iteración, Asociar metadatos.

Tareas	
No. Tarea: 7	No. HU: 7
Nombre de la tarea: Modificar Metadatos.	
Tipo de tarea: Desarrollo.	Puntos de estimación:

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Fecha inicio:	Fecha fin:
Programador responsable: Yudier Pereira Salazar	
Descripción: se selecciona el tipo documental a la que se le va a modificar metadatos, luego se modifican los valores de los metadatos deseados, para luego salvar los valores en la base de datos.	

Tabla 29: Tareas de la ingeniería de la primera iteración, Modificar metadatos.

3.3 DIAGRAMA DE COMPONENTES DEL SISTEMA

El diagrama de componentes representa como está dividido el sistema y la dependencia entre sus componentes, se agrupan en subsistemas organizados en capas y jerarquías. A continuación se muestran los diagramas correspondientes a las capas de presentación, negocio y acceso a datos.

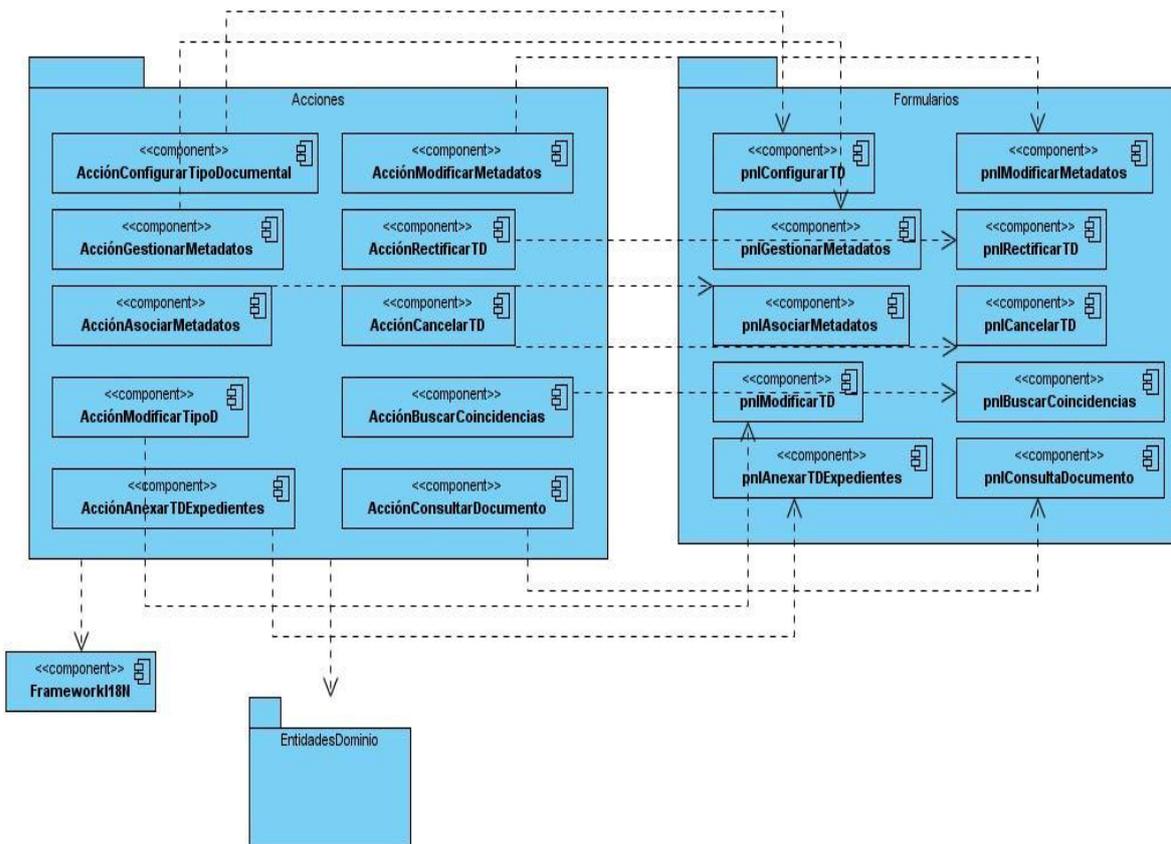


Figura 8: Diagrama de componentes Capa de Presentación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

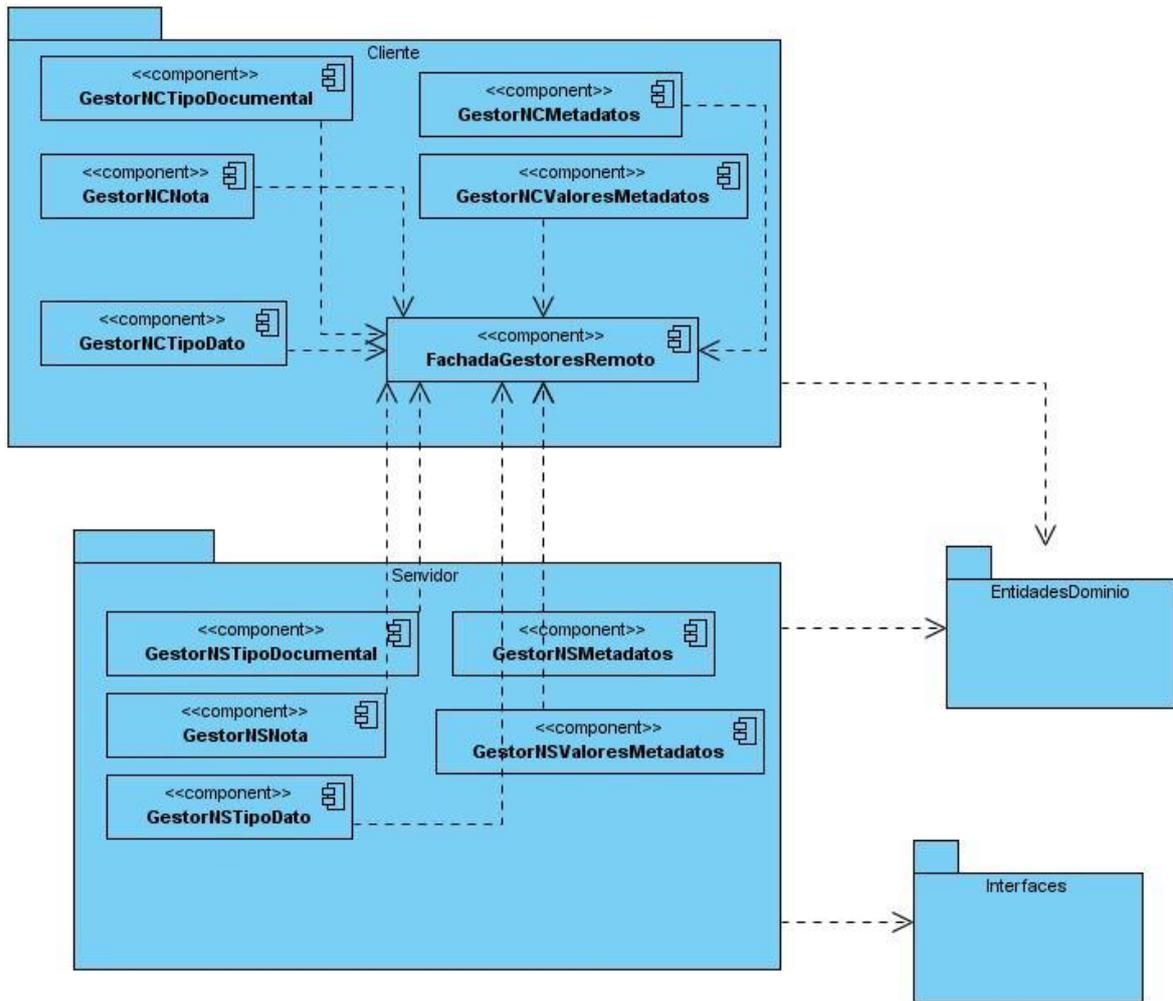


Figura 9: Diagrama de Componentes Capa de Negocio.

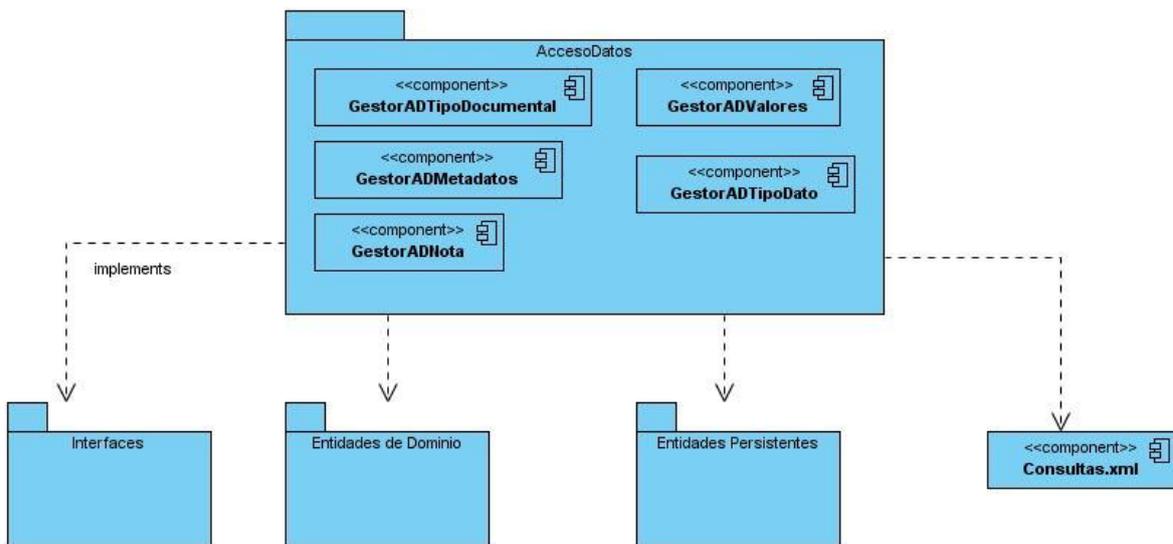


Figura 10: Diagrama de Componentes Capa de Acceso a Datos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.4 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue muestra las relaciones físicas entre componentes de hardware y software en el sistema, es decir permite apreciar de forma visual como se encuentran relacionados físicamente los componentes del sistema (40). La metodología XP no plantea el modelado de este artefacto, pero para un mejor entendimiento de donde se desplegará el sistema se decide su realización.

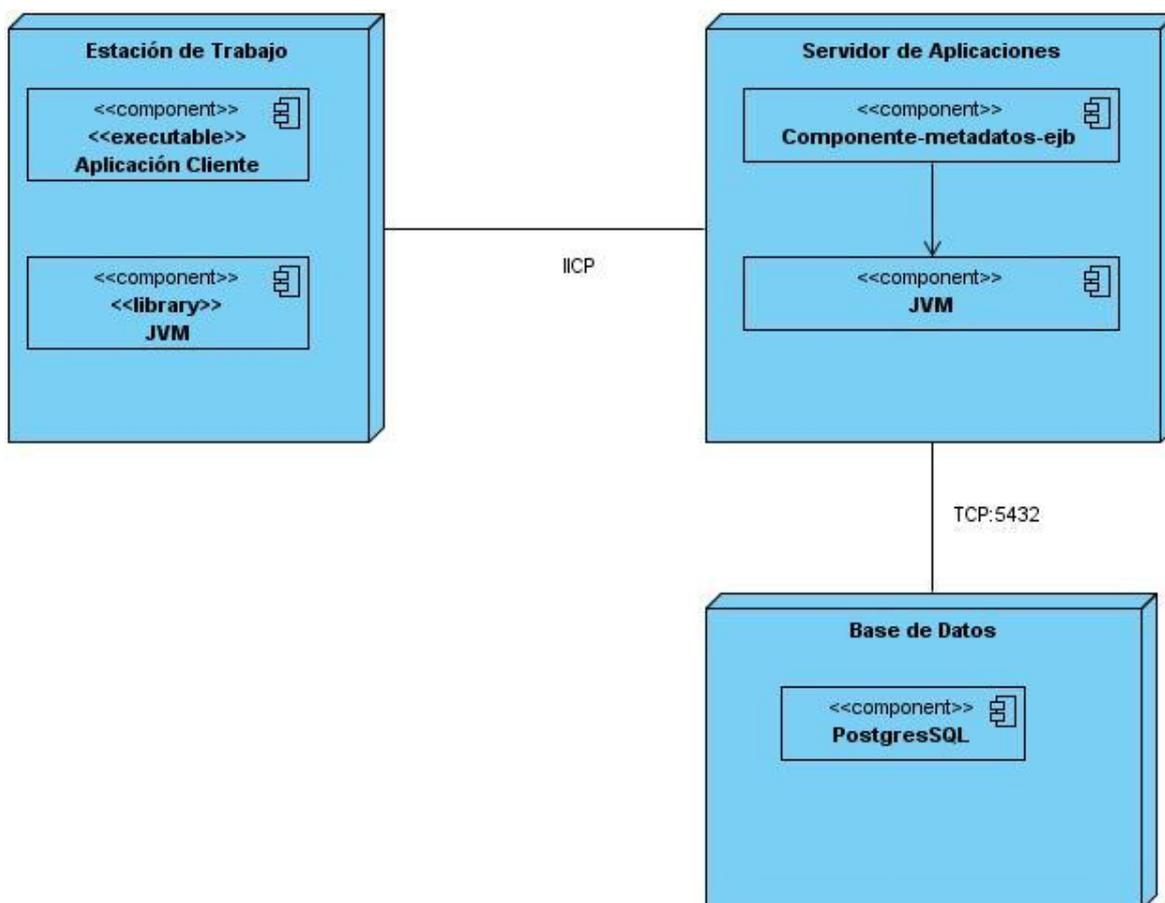


Figura 11: Diagrama de despliegue.

Estación de Trabajo: Es la computadora que se conecta a la aplicación para visualizar los componentes desarrollados.

Servidor de Aplicaciones: Es el servidor que contiene la aplicación y software del sistema, a los cuales se accederá desde las pc clientes, aplicación que se comunica con un sistema de gestión de base de datos, empleándose como protocolo de comunicación entre el cliente y el servidor el IICP debido a que está basado en el modelo cliente-servidor.

Base de Datos: Es el servidor que almacena todos los datos relacionados con el sistema, el cual se comunica con el servidor mediante el protocolo TCP/ IP (del inglés:

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Transmission Control Protocol/ Internet Protocol), ya que este permite establecer una conexión e intercambiar datos entre estos, garantizando la entrega de los datos y que estos no se pierdan durante la transmisión.

3.5 PRUEBAS

Uno de los principios fundamentales de XP es el proceso de prueba, donde los desarrolladores realizan pruebas constantemente tanto como sea posible; de esta manera se reduce el número de errores no detectados, así como el tiempo entre la introducción de éste en el sistema y su detección. Todo esto contribuye a elevar la calidad del producto desarrollado.

XP divide las pruebas en dos grupos:

- Pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática.
- Pruebas de aceptación, las cuales son destinadas a evaluar bloques más grandes de la funcionalidad del sistema, tales como las historias de usuarios, verificando que al final de una iteración se obtuvo la funcionalidad requerida, además de que dicha funcionalidad sea la esperada por el cliente.

3.6 PRUEBAS UNITARIAS

Las pruebas unitarias son la forma de probar el correcto funcionamiento en código de un sistema informático, lo que es de vital importancia a la hora de obtener un sistema de calidad. Seguidamente se describirá la realización de las pruebas unitarias realizadas, desglosándola en sus elementos fundamentales: confección del grafo de flujo, cálculo de la complejidad ciclomática, extracción de los caminos independientes, realización de los casos de pruebas y análisis de los resultados obtenidos.

A continuación se muestra el código del método **asociarMetadatos** de la clase **acciónAsociarMetadatos**, al cual se le aplicó la métrica de complejidad ciclomática debido a que es uno de los métodos relevantes en la solución informática brindada.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

```
192 public void AsociarMetadatos() throws Exception {
193
194     List<EDMetadatos> listaVal = (List<EDMetadatos>) this.getFormulario().tblMetadatos.getFuenteDatos();
195     accionLlenarMetadatosComboBox accionNueva = new accionLlenarMetadatosComboBox();
196     listaVal.addAll(accionLlenarMetadatosComboBox.listaMetadatos);
197     accionLlenarMetadatosComboBox.listaMetadatos.clear();
198     List<EDValoresMetadatos> valoresComp = ValoresMetadatosparaMensaje();
199     if (valoresComp.size() != listaVal.size()) {
200         GestorMensajes.MensajeAviso(FrameworkI18n.getInstancia().getMensaje("MensajeNoAsocioTodos"));
201         return false;
202     }
203     return;
204 }
205 for (int i = 0; i < listaVal.size(); i++) {
206     if (!"".equals(listaVal.get(i).getValorMetadato())) {
207         GestorMensajes.MensajeAviso(FrameworkI18n.getInstancia().getMensaje("MensajeNoAsocioTodos"));
208         return false;
209     }
210 }
211 }
212
213 return true;
214 GestorNCMetadatos.LlenarCampoMetadatos(listaVal, (EDTipoDocumentales) this.getFormulario().cmbTiposDocumentales.getObjetoSeleccionado());
215
216
217 }
```

Figura 12: Código asociar metadatos.

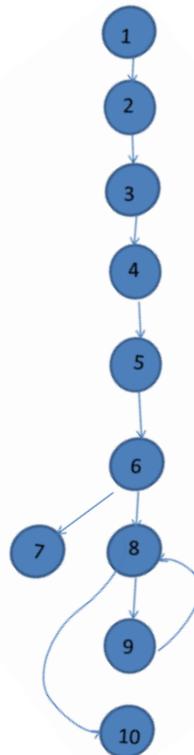


Figura 13: Grafo de flujo asociado.

Cálculo de la complejidad ciclomática

$$V(G) = R = 2$$

$$V(G) = A - N + 2 = 10 - 10 + 2 = 2$$

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

$$V(G) = P + 1 = 1 + 1 = 2$$

Caminos independientes determinados

Camino 1: 1-2-3-4-5-6-7

Camino 2: 1-2-3-4-5-6-8-9-10

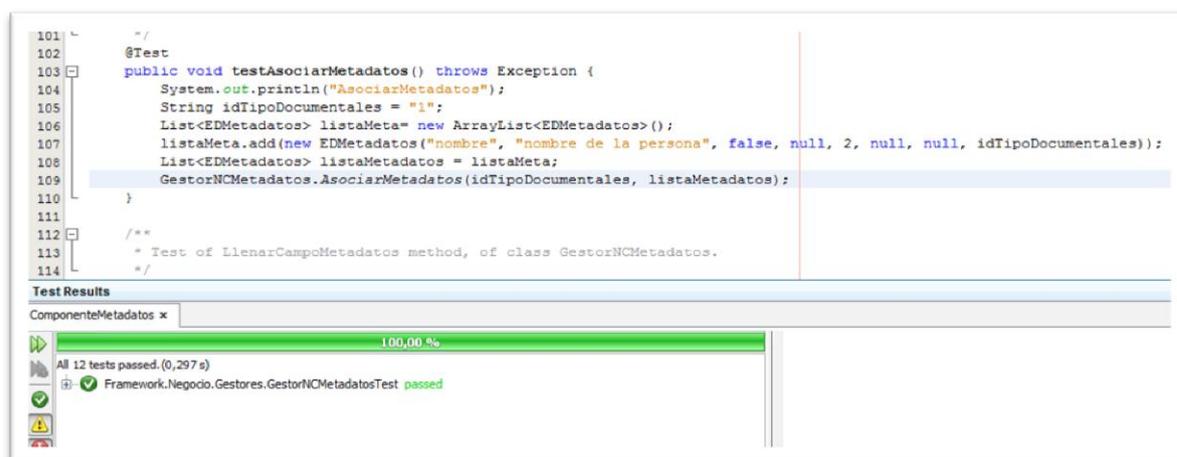
Camino 2: 1-2-3-4-5-6-8-10

Para consultar los resultados de las pruebas realizadas al código del sistema, ver **Anexo 2**.

3.7 CASOS DE PRUEBA

Para llevar a cabo las pruebas unitarias se utilizó el framework de pruebas JUnit. Al elaborar un caso de prueba en JUnit es necesario hacer una “clase de test”, que se llama igual a la clase original pero añadiendo el sufijo “Test” al final, además se puede elegir sobre qué método específico se desea realizar dicha prueba o si es a toda la clase. Para realizar una prueba el test utiliza el método assertEquals para comprobar si el resultado de llamar al método coincide con el esperado. Otras posibles funciones son assertTrue, assertFalse, etc. Si este comando falla cuando se ejecuta el caso de prueba, JUnit dará el caso por fallido, de lo contrario el resultado es correcto.

Se realizaron las pruebas, por cada uno de los caminos independientes determinados, obteniendo resultados satisfactorios. A continuación se muestran las pruebas realizadas al método asociarMetadatos de la clase acciónAsociarMetadatos.



```
101  //
102  @Test
103  public void testAsociarMetadatos() throws Exception {
104      System.out.println("AsociarMetadatos");
105      String idTipoDocumentales = "1";
106      List<EDMetadatos> listaMeta= new ArrayList<EDMetadatos>();
107      listaMeta.add(new EDMetadatos("nombre", "nombre de la persona", false, null, 2, null, null, idTipoDocumentales));
108      List<EDMetadatos> listaMetadatos = listaMeta;
109      GestorNCMetadatos.asociarMetadatos(idTipoDocumentales, listaMetadatos);
110  }
111
112  /**
113   * Test of LlenarCampoMetadatos method, of class GestorNCMetadatos.
114   */
```

Test Results
ComponenteMetadatos x
100,00 %
All 12 tests passed. (0,297 s)
Framework.Negocio.Gestores.GestorNCMetadatosTest passed

Figura 14: Código de prueba CU Asociar metadatos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
GestorNCExpedienteTest	2	0	0	0.113	2012-06-08T14:36:17	NERUBIAN-PC
GestorNCImagenTest	2	0	0	0.074	2012-06-08T14:39:35	NERUBIAN-PC
GestorNCMetadatosTest	12	0	0	0.280	2012-06-08T14:24:35	NERUBIAN-PC
GestorNCNotaTest	1	0	0	0.070	2012-06-08T14:38:01	NERUBIAN-PC
GestorNCTipoDatoTest	2	0	0	0.076	2012-06-08T14:30:09	NERUBIAN-PC
GestorNCTipoDocumentoTest	13	0	0	0.219	2012-06-08T14:24:36	NERUBIAN-PC
GestorNCValoresMetadatosTest	3	0	0	0.095	2012-06-08T14:27:33	NERUBIAN-PC

Figura 15: Resultados de las pruebas del sistema.

3.8 ANÁLISIS DE LOS RESULTADOS

Para la validación del código generado en el desarrollo de la herramienta se seleccionaron todos los métodos de la aplicación, a los cuales se le realizaron las pruebas, para poder evaluar si el funcionamiento de cada uno de estos métodos se comportó de la manera esperada. Se realizaron un total de 35 pruebas a todas las funcionalidades implementadas, de las cuales 35 resultaron satisfactorias, comprobando la estabilidad de la lógica aplicada en el código generado en el desarrollo de la herramienta informática.

3.9 PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación son las pruebas de caja negra que se crean a partir de las historias de usuarios. Durante las iteraciones las HU seleccionadas serán sometidas a pruebas de aceptación. En estas pruebas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Una HU no se considera completa hasta que no ha pasado por las pruebas de aceptación. Siendo el objetivo final de estas, garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

Caso de Prueba de Aceptación	
Código	Historia de Usuario (Nro. y Nombre):
Nombre	
Descripción	
Condiciones de Ejecución	
Entrada / Pasos de Ejecución	
Resultado Esperado	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Evaluación de Prueba

Tabla 30: Caso de prueba de aceptación.

Campos de la tarjeta Caso de Prueba Número historia de usuario: Número de la historia de usuario a la que corresponde. Índice de la historia de usuario a la que se le desea comprobar este aspecto.

Número: Índice del caso de prueba. Es un número único que se le asigna a cada caso de prueba que pertenece a una historia de usuario determinada con el fin de lograr una mejor organización de estos.

Nombre: Nombre del caso de prueba. Debe ser descriptivo, en la medida de las posibilidades, de lo que se comprobará y no muy extenso.

Descripción: Se describe qué es lo que se desea probar. La descripción debe ser corta y precisa.

Fecha: Fecha en que se realiza el caso de prueba. En caso de realizarse varias veces siempre se actualiza con la última.

Tipo de prueba: Informa si la prueba que se realiza es funcional o unitaria. **Condiciones**

de ejecución: Condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

Entradas: Entradas al caso de prueba en caso de necesitarlas.

Resultado esperado: Resultado que se desea tenga el caso de prueba. Descripción breve de lo que debe suceder.

Evaluación: Se evalúa si el caso de prueba tuvo éxito o no. En caso de ser exitoso se asigna un resultado de satisfactorio, en caso contrario insatisfactorio.

Caso de Prueba de Aceptación	
Código: 1	Historia de Usuario(Nro. y Nombre): HU #Configurar Tipo Documental
Nombre: Comprobar Configurar Tipo Documental	
Descripción: Prueba para la funcionalidad de configurar tipos documentales	
Condiciones de Ejecución: Entrar al sistema los datos de la configuración del tipo documental.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos del tipo documental	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Tabla 31: Caso de prueba de aceptación: Comprobar Configurar Tipo Documental.

Caso de Prueba de Aceptación	
Código: 2	Historia de Usuario(Nro. y Nombre): HU# Configurar Metadatos.
Nombre: Comprobar Configurar Metadatos	
Descripción: Prueba para la funcionalidad de configurar metadatos.	
Condiciones de Ejecución: Entrar al sistema los datos de la configuración de los metadatos.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos de los metadatos	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 32: Caso de prueba de aceptación: Comprobar Configurar Metadatos.

Caso de Prueba de Aceptación	
Código: 3	Historia de Usuario (Nro. y Nombre): HU #Asociar Metadatos.
Nombre: Comprobar Asociar Metadatos	
Descripción: Prueba para la funcionalidad de asociar metadatos.	
Condiciones de Ejecución: Entrar al sistema los valores para los metadatos asociados a un tipo documental determinado.	
Entrada / Pasos de Ejecución: Se intenta introducir los valores de los metadatos	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 33: Caso de prueba de aceptación: Comprobar Asociar Metadatos.

Caso de Prueba de Aceptación	
Código: 4	Historia de Usuario(Nro. y Nombre): HU #Buscar Coincidencias Documentales
Nombre: Comprobar buscar coincidencias documentales	
Descripción: Prueba para la funcionalidad buscar coincidencias documentales.	
Condiciones de Ejecución: Entrar al sistema los datos de la búsqueda de coincidencias	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Entrada / Pasos de Ejecución: Se intenta introducir los datos de la búsqueda de coincidencias
Resultado Esperado: El sistema muestra un listado con los tipos documentales esperados.
Evaluación de Prueba: Prueba satisfactoria.

Tabla 34: Caso de prueba de aceptación: Comprobar buscar coincidencias documentales.

Caso de Prueba de Aceptación	
Código: 5	Historia de Usuario (Nro. y Nombre): HU #Rectificar Tipo Documental.
Nombre: Comprobar Rectificar Tipo Documental	
Descripción: Prueba para la funcionalidad de rectificar tipos documentales	
Condiciones de Ejecución: Entrar al sistema los datos de la rectificación del tipo documental.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos de la rectificación del tipo documental.	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 35: Caso de prueba de aceptación: Comprobar Rectificar Tipo Documental.

Caso de Prueba de Aceptación	
Código: 6	Historia de Usuario (Nro. y Nombre): HU #Cancelar Tipo Documental.
Nombre: Comprobar Cancelar Tipo Documental	
Descripción: Prueba para la funcionalidad de cancelar tipos documentales	
Condiciones de Ejecución: Entrar al sistema los datos de la cancelación del tipo documental.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos de la cancelación del tipo documental.	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 36: Caso de prueba de aceptación: Comprobar Cancelar Tipo Documental.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Caso de Prueba de Aceptación	
Código: 7	Historia de Usuario (Nro. y Nombre): HU #Modificar Tipo Documental.
Nombre: Comprobar Modificar Tipo Documental	
Descripción: Prueba para la funcionalidad de modificar tipos documentales	
Condiciones de Ejecución: Entrar al sistema los datos de la modificación del tipo documental.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos de la modificación del tipo documental.	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 37: Caso de prueba de aceptación: Comprobar Modificar Tipo Documental.

Caso de Prueba de Aceptación	
Código: 8	Historia de Usuario (Nro. y Nombre): HU #Modificar Metadatos.
Nombre: Comprobar Modificar Metadatos	
Descripción: Prueba para la funcionalidad de modificar metadatos.	
Condiciones de Ejecución: Entrar al sistema los datos de la modificación de los metadatos.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos de la modificación de los metadatos.	
Resultado Esperado: El sistema guarda los datos de forma satisfactoria.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 38: Caso de prueba de aceptación: Comprobar Modificar Metadatos.

Caso de Prueba de Aceptación	
Código: 9	Historia de Usuario (Nro. y Nombre): HU #Consultar Documento Digital.
Nombre: Comprobar consultar documento digital.	
Descripción: Prueba para la funcionalidad de consultar documento digital.	
Condiciones de Ejecución: Entrar al sistema los datos del tipo documental del cual se desea mostrar la imagen	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Entrada / Pasos de Ejecución: Se intenta introducir los datos del tipo documental del cual se desea mostrar la imagen.
Resultado Esperado: El sistema muestra la imagen del tipo documental.
Evaluación de Prueba: Prueba satisfactoria.

Tabla 39: Caso de prueba de aceptación: Comprobar consultar documento digital.

Caso de Prueba de Aceptación	
Código: 10	Historia de Usuario (Nro. y Nombre): HU #Anexar Tipos Documentales a Expedientes.
Nombre: Comprobar anexar tipos documentales a expedientes.	
Descripción: Prueba para la funcionalidad de anexar tipos documentales a expedientes.	
Condiciones de Ejecución: Entrar al sistema los datos del tipo documental el cual se va a anexar al expediente.	
Entrada / Pasos de Ejecución: Se intenta introducir los datos del tipo documental el cual se va a anexar al expediente.	
Resultado Esperado: El sistema adiciona el tipo documental al expediente.	
Evaluación de Prueba: Prueba satisfactoria.	

Tabla 40: Caso de prueba de aceptación: Comprobar anexar tipos documentales a expedientes.

3.10 CONCLUSIONES PARCIALES

- Al culminar el capítulo “Implementación y Prueba”, se le ha dado cumplimiento a las últimas tareas trazadas para el desarrollo del sistema, quedando totalmente implementada y validada la aplicación. Se realizaron los diagramas de componentes para modelar la vista estática del sistema, y el diagrama de despliegue para indicar la situación física de los componentes lógicos desarrollados. Se describieron las tareas de ingeniería de las historias de usuario de prioridad alta, y se realizaron las pruebas correspondientes para validar la implementación realizada.

CONCLUSIONES GENERALES

Una vez concluido el desarrollo del presente trabajo de diploma se puede llegar a las siguientes conclusiones:

- La correcta selección de la metodología y tecnologías utilizadas para el desarrollo del componente se realizó a partir de la elaboración del marco teórico de la investigación.
- Los artefactos que sirvieron para mostrar la primera visión de la implementación del sistema donde se definió una estrategia para la construcción flexible y escalable con la utilización de patrones, arquitectura y estándares se generaron con la realización del diseño.
- El componente de asociación de metadatos que convirtió los elementos del diseño en términos de implementación aplicando los estándares de interfaz y codificación definidos por el equipo de desarrollo se obtuvo con la implementación de todos los requerimientos identificados.
- El nivel de calidad y fiabilidad que presenta el diseño, la implementación y las pruebas realizadas se evaluó con la aplicación de pruebas unitarias y la corrección de los errores detectados durante el proceso.
- La solución alcanzada logró satisfacer los requisitos de la plataforma SuiteDoc para la asociación de metadatos con la implementación del componente de asociación dinámica de metadatos.

RECOMENDACIONES

Luego de haber dado cumplimiento a los objetivos de este trabajo de diploma y teniendo en cuenta las experiencias adquiridas durante el desarrollo del mismo, se recomienda:

1. Integrar el componente desarrollado a la plataforma SuiteDoc.
2. Incorporar la captura de datos por OCR como otra técnica de asociación dinámica de metadatos que pueda ser aplicada en sistemas que lo requieran
3. Trabajar en nuevas versiones del componente que permitan la mejora continua del mismo.

BIBLIOGRAFÍA

1. Uso de las Tecnologías de la Información y la Comunicación (TIC) para la catalogación y difusión del patrimonio cultural. **Concepción Pérez de Celis Herrero, Gustavo Cossio Aguilar y Jaime Lara Álvarez.** 2010. 1850-0013.
2. **Colectivo de autores.** buenas tareas. [En línea] 2012. [Citado el: 3 de Enero de 2012.] <http://www.buenastareas.com/ensayos/Proyecto-De-Fabrica-De-Pastas/1232629.html>.
3. La Importancia de la Digitalización de Archivos para la Biblioteca. **Cam, Celso Gonzáles.** Peru : s.n., 2007.
4. Manejo de información básica de datos y metadatos. **Geóg. Zulay Méndez, Ing. Rodrigo Torrén.** 1.0, 2003.
5. **Alejandra Sachez Maganto, Javier Noguerras Iso, Daniela Ballari.** www.academia.edu. Normas sobre metadatos. [En línea] [Citado el: 15 de febrero de 2012.] http://wu.academia.edu/DanielaBallari/Papers/272386/Normas_Sobre_Metadatos_ISO19115_ISO19115-2_ISO19139_ISO15836_..
6. **Márquez, Juan Voutssás.** www.interpares.org. Preservación del patrimonio documental digital en Mexico. [En línea] 2009. [Citado el: 20 de diciembre de 2011.] http://www.interpares.org/ip2/display_file.cfm?doc=ip3_mexico_dissemination_b_voutssas_unam_2009.pdf.
7. **Nelson Gustavo A Cossari, Daniel Germán Luna.** www.aaba.org.ar. El principio de prevención y la evaluación de impacto ambiental. [En línea] 4 de octubre de 2002. [Citado el: 15 de diciembre de 2011.] <http://www.aaba.org.ar/bi20op27.htm>.
8. **Real, Ismael González.** ebam.gesbi.com.ar. Gestión documental y gobierno electrónico. [En línea] 10 de septiembre de 2010. [Citado el: 10 de enero de 2012.] <http://ebam.gesbi.com.ar/reservorio10/ponencias2EBAM/2EBAM-E1-p3a.pdf>.
9. **Elsy Patricia Muños Gomez, Carmen Paola Reyes, Leonardo Calderon Mejía .** www.slideshare.net. Gestion de Documentos Digitales. [En línea] junio de 2010. [Citado el: 14 de marzo de 2012.] <http://www.slideshare.net/ilypatico/eje-temtico-4-digitalizacin-definitivo>.
10. **Colectivo de Autores.** Diccionario de la Real Academia de Española. 2001.
11. www.hispanoargentinos.com. www.hispanoargentinos.com. [En línea] 2002. [Citado el: 15 de febrero de 2012.] <http://www.hispanoargentinos.com/editoriales/Matozza/Digitalizacion-Documentos-2.htm>.
12. Recomendaciones Técnicas para la digitalización de documentos. **Juntas de Andalucía.** Andalucía : s.n., 2010.

BIBLIOGRAFÍA

13. **Muñoz, Maria Teresa Bermúdez.** www.archivonacional.go.cr. Guia para digitalizar documentos. [En línea] 23 de marzo de 2006. [Citado el: 14 de marzo de 2012.] http://www.archivonacional.go.cr/pdf/guia_digitalizar_documentos.pdf.
14. Qué son los metadatos y como es la organización de los recursos electrónicos. **Adrián, Luis.** 2007.
15. **Peiya, Liu.** www.dsc.ufcg.edu.br. Metadata Standards for Web-Based Resources. [En línea] Marzo de 2001. [Citado el: 15 de marzo de 2012.] <http://www.dsc.ufcg.edu.br/~garcia/cursos/TEICOPIN/metadataWE.pdf>.
16. **Silvia Marcela Angelozzi.** www.bn.gov.ar. Análisis y comparación de metadatos para la descripción de recursos electrónicos en línea. [En línea] 15 de diciembre de 2007. [Citado el: 15 de marzo de 2012.] http://www.bn.gov.ar/descargas/catalogadores/ponencia_angelozzi.pdf.
17. Metadatos:Necesidad e importancia de integrar estandares. **Garcia, Nelida Elba.** Buenos Aires : s.n., 2009.
18. www.metadatos-xmlrdf.com. METADATOS CATALOGANDO Y DESCRIBIENDO PARA FACILITAR LA BUSQUEDA Y LA RECUPERACIÓN DE LA INFORMACIÓN. [En línea] 2010. [Citado el: 15 de marzo de 2012.] <http://www.metadatos-xmlrdf.com/metadatos>.
19. **Swiatecka, Agnieszka.** slideshare. [En línea] 10 de febrero de 2008. [Citado el: 19 de marzo de 2012.] <http://www.slideshare.net/anansi/introduccion-a-metadatos>.
20. Metadatos y documentos XML/RDF para la recuperación de información. **Gonzalo.**
21. Gestión documental open source. Metadatos, Las Funcionalidades. [En línea] 2010. http://www.smile-iberia.com/ca/libros_blanco/gestion_documental_open_source/las_funcionalidades/metadatos.
22. **Colectivo de autores.** Smile. [En línea] 2010. [Citado el: 20 de Marzo de 2012.] http://www.smile-iberia.com/es/libros_blanco/gestion_documental_open_source/las_funcionalidades/metadatos.
23. PROCESO DE DIGITALIZACIÓN EN LA BIBLIOTECA NACIONAL DE ESPAÑA. **BIBLIOTECA DIGITAL HISPÁNICA.** España : s.n., 2011.
24. QuickScan Pro™ Solución estandarizada para la captura de documentos. **EMC Captiva.** San Diego : EMC, 2006.
25. www.novatec.com.ve. [En línea] 8 de agosto de 2010. [Citado el: 16 de mayo de 2012.] <http://www.novatec.com.ve/index.php/sdd>.

BIBLIOGRAFÍA

26. **Fernández Bertot, Víctor and Orozco González, Delio G.** Un sistema de conservación, digitalización, gestión y socialización de información documental para los archivos en Cuba. Granma - Manzanillo : s.n. sn: 200.
27. **Proyecto CDA.** Presentación de productos de Digitalización, Mercadotecnia. Habana : s.n., 2011.
28. **Mué, Yadira Lizama.** CENTRO DE DIGITALIZACIÓN PARA LA DIVISIÓN DE ANTECEDENTES PENALES DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. Ciudad de La Habana : CENTRO DE DIGITALIZACIÓN PARA LA DIVISIÓN DE ANTECEDENTES PENALES DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA., 2011.
29. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). **Letelier, Patricio y Penadés, M^a Carmen.** Valencia : s.n.
30. ¿Qué es Java? **Alvarez, Miguel Angel.** 2001.
31. **Wilson, Leslie B.** Comparative Programing Languages. Addison-Wesley : s.n., 1993.
32. Especificación e implementación de software basado en componentes con la tecnología EJB 3.0. **Gómez, Omar.** 2008.
33. ¿Qué es UML? **Cornejo, José Enrique González.**
34. El Lenguaje Unificado de Modelado (UML). **Orallo, Enrique Hernández.** España : s.n.
35. **Colectivo de Autores.** Sitio Web oficial Visual-Paradigm. [En línea] 2011. <http://www.visual-paradigm.com/>.
36. **Corporación Oracle.** NetBeans. [En línea] Oracle Corporation, 2012. [Citado el: 4 de Diciembre de 2011.] http://netbeans.org/community/releases/71/index_pt_BR.html.
37. Introducción a Base de Datos con PostgreSQL. **Medrano, Jesus Rafael Sanchez.** 2009.
38. **Ecured.** Arquitectura Cliente/Servidor. [En línea] <http://www.ecured.cu/index.php/Cliente-Servidor>.
39. **Pressman, Roger S.** Ingeniería del Software:Un enfoque práctico. es.scribd.com. [En línea] [Citado el: 14 de mayo de 2012.] <http://es.scribd.com/doc/7978336/Ingenieria-de-Software-Un-Enfoque-Practico-Pressman-5th-Ed>.
40. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** El Proceso Unificado de Desarrollo de Software. s.l. : Addison Wesley, 2000.
41. **Camacho, Erika, Cardeso, Fabio y Núñez, Gabriel.** Arquitecturas de Software. 2004.

ANEXOS

ANEXO 1: Diagrama de Procesos.

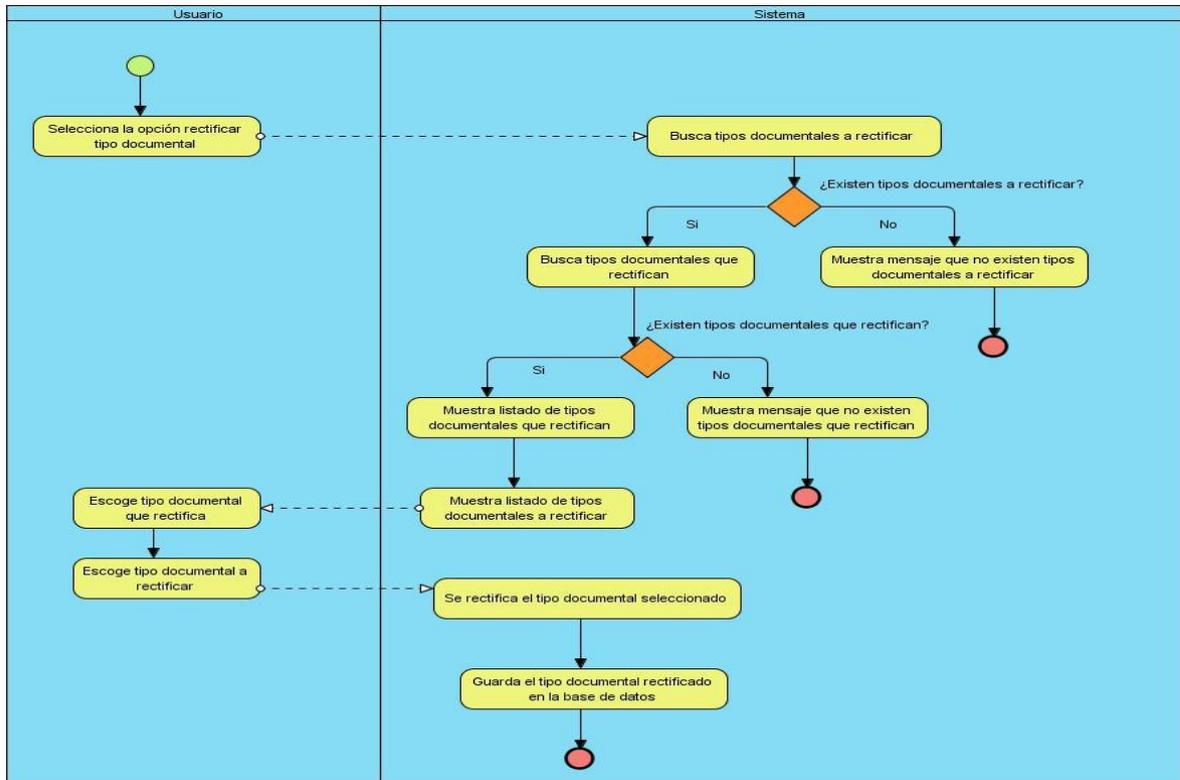


Figura 16: Diagrama de proceso Rectificar Tipo Documental.

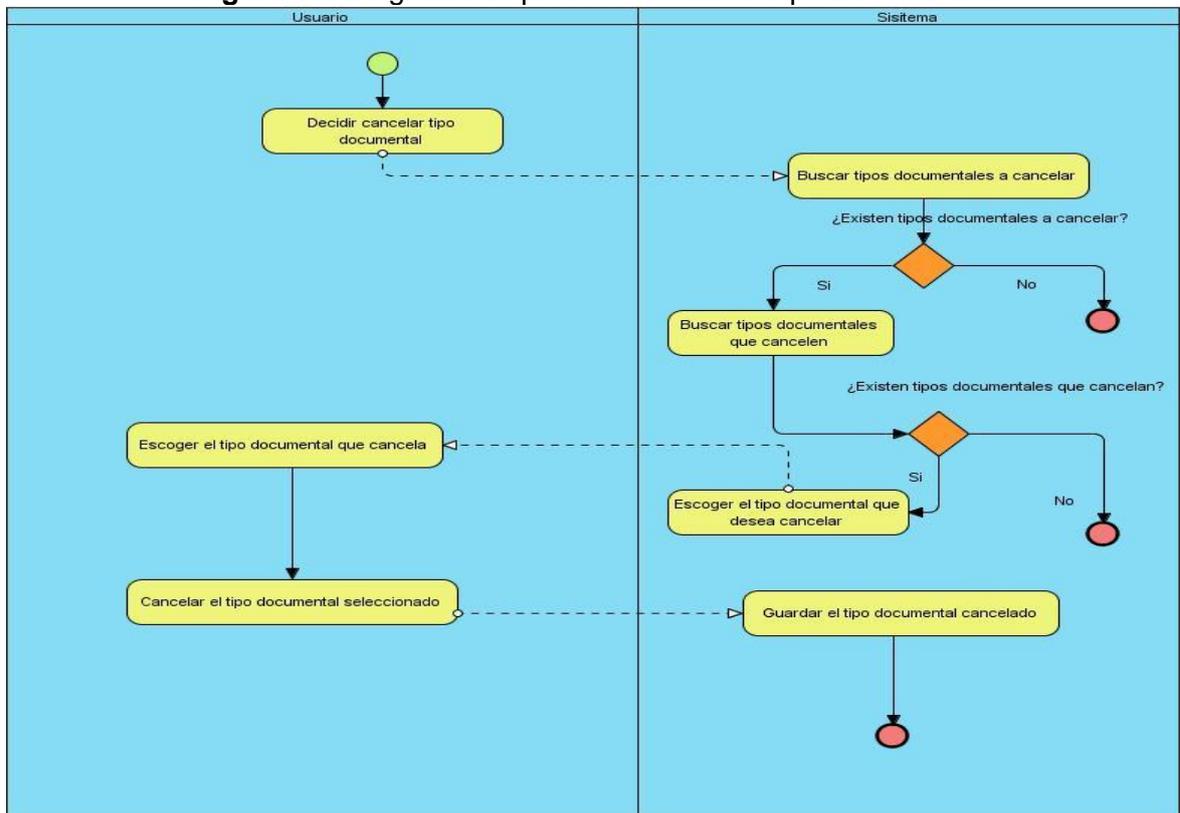


Figura 17: Diagrama Cancelar Tipo Documental.

ANEXOS

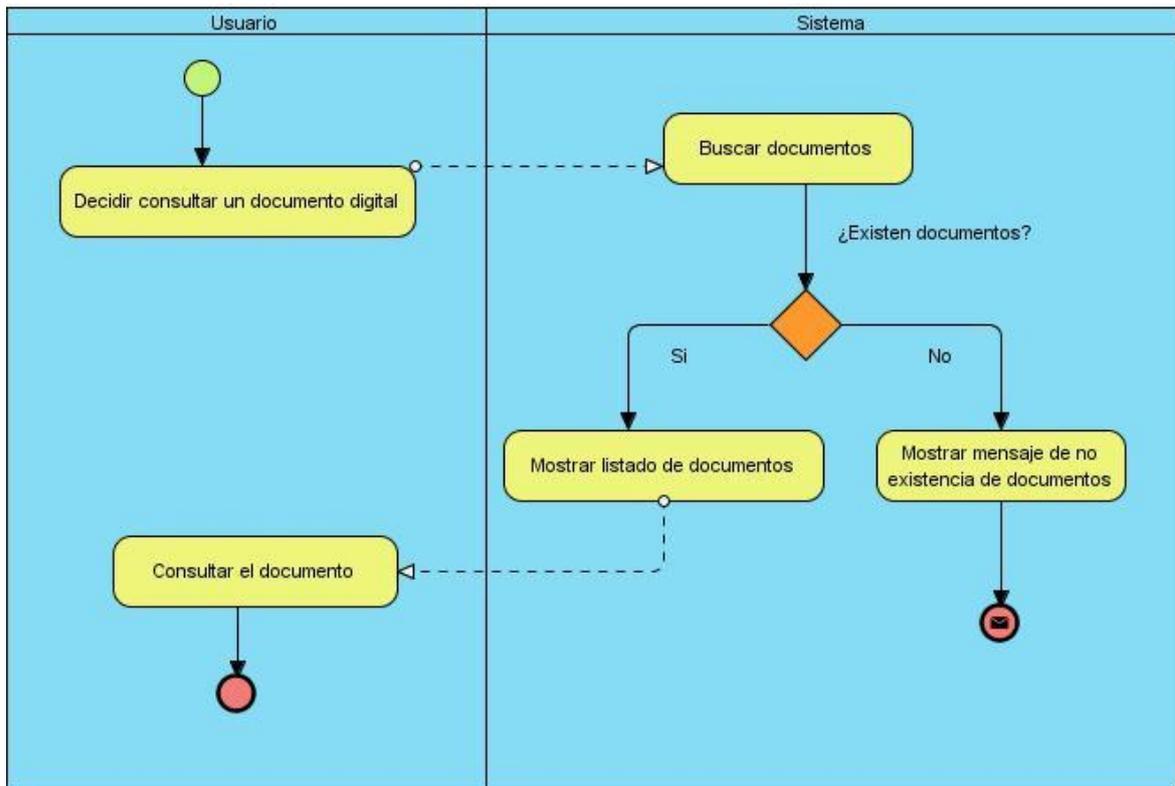


Figura 18: Diagrama Consultar Documento Digital.

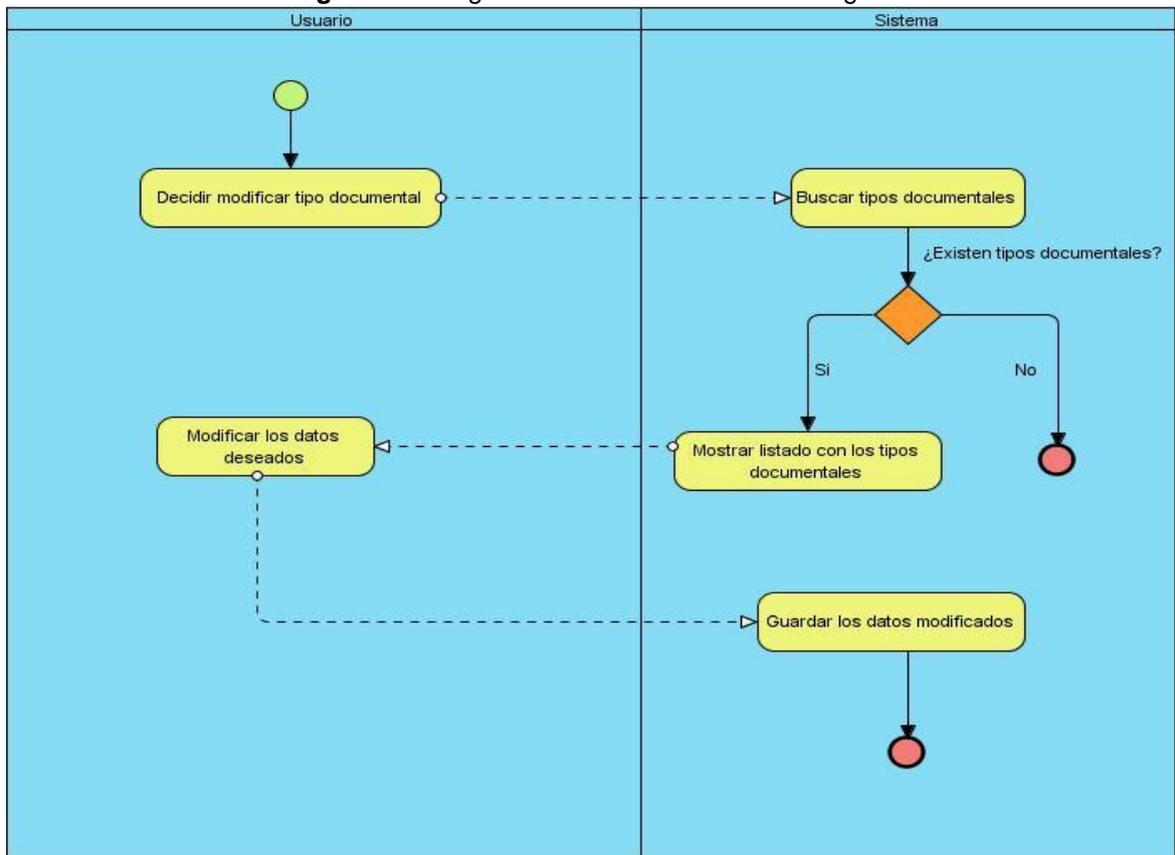


Figura 19: Diagrama Modificar Tipos Documentales.

ANEXOS

ANEXO 2: Resultados de las pruebas.

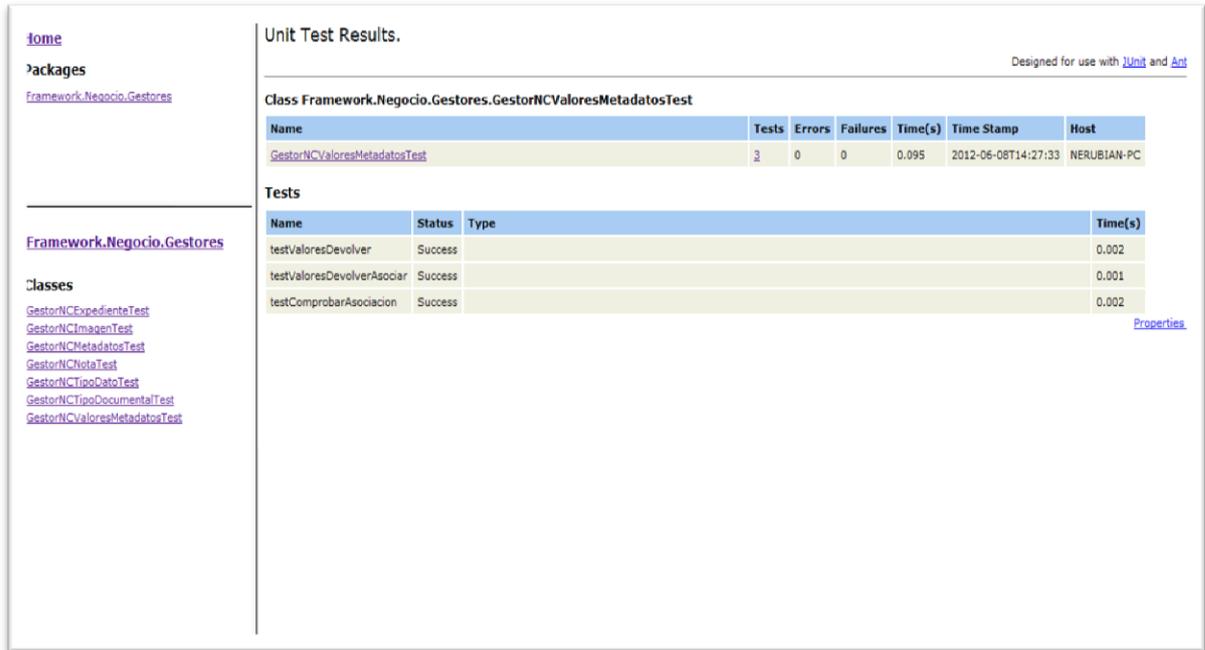


Figura 20: Pruebas unitarias.

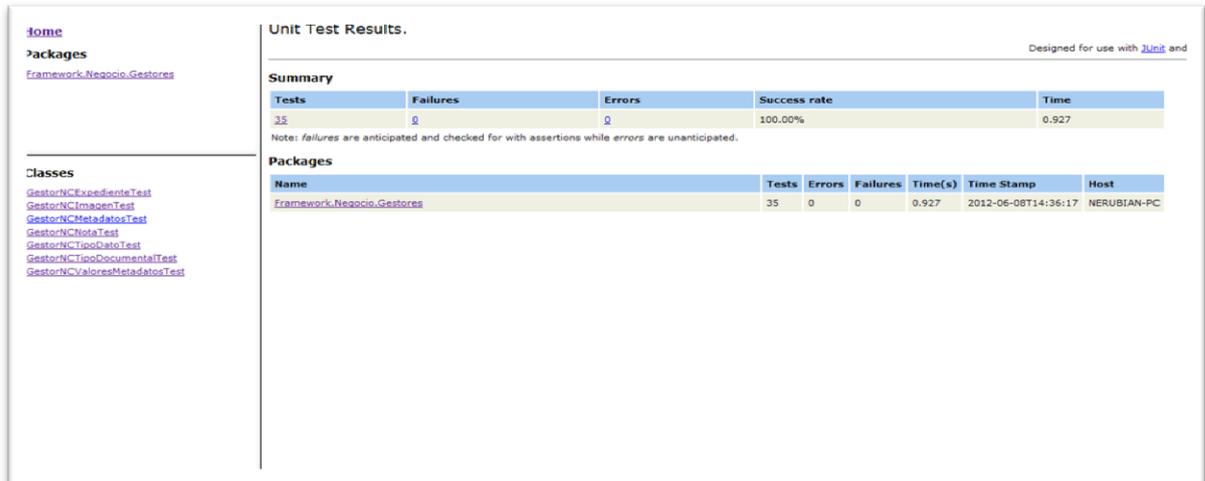


Figura 21: Pruebas GestorNCValoresMetadatos.

ANEXOS

Unit Test Results. Designed for use with [JUnit](#) and [Ant](#)

Class Framework.Negocio.Gestores.GestorNCTipoDocumentalTest

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
GestorNCTipoDocumentalTest	13	0	0	0.219	2012-06-08T14:24:36	NERUBIAN-PC

Tests

Name	Status	Type	Time(s)
testSalvarTiposDocumentales	Success		0.002
testObtenerTiposDocumentales	Success		0.001
testObtenerTiposDocumentalesSinMetadatos	Success		0.005
testLlenarGrid	Success		0.000
testObtenerTiposDocumentalesRectificables	Success		0.001
testObtenerTiposDocumentalesCancelables	Success		0.001
testCancelarTipoDocumental	Success		0.007
testLlenarGridTiposDocumentales	Success		0.000
testBuscarDocumentalesCanceladas	Success		0.000
testObtenerTiposDocumentalesModificar	Success		0.000
testRectificarTipoDocumental	Success		0.003
testAnexarTDExpedientes	Success		0.002
testLlenarGridNoAnexadas	Success		0.001

[Properties](#)

Figura 22: Pruebas GestorNCTipoDocumental.

Unit Test Results. Designed for use with [JUnit](#) and [Ant](#)

Class Framework.Negocio.Gestores.GestorNCMetadatosTest

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
GestorNCMetadatosTest	12	0	0	0.280	2012-06-08T14:24:35	NERUBIAN-PC

Tests

Name	Status	Type	Time(s)
testSalvarMetadatos	Success		0.002
testLlenarGridMetadatos	Success		0.001
testModificarMetadatos	Success		0.001
testLlenarMetadatosAsociar	Success		0.071
testAsociarMetadatos	Success		0.001
testLlenarCampoMetadatos	Success		0.001
testElementosComboBox	Success		0.001
testLlenarComboMetadato	Success		0.001
testLlenarValoresComboMetadatos	Success		0.001
testActualizarValorCombo	Success		0.002
testLlenarComboMetadatosModificar	Success		0.002
testLlenarMetadatosCoincidencias	Success		0.001

[Properties](#)

Figura 23: Prueba GestorNCMetadatos.

ANEXOS

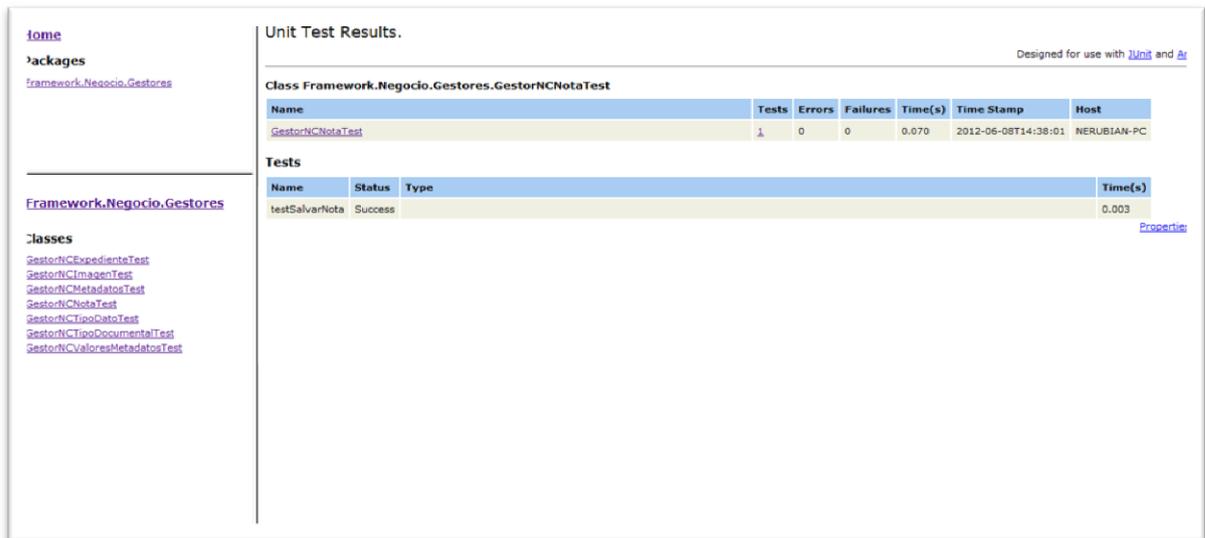


Figura 24: Pruebas GestorNCNota.

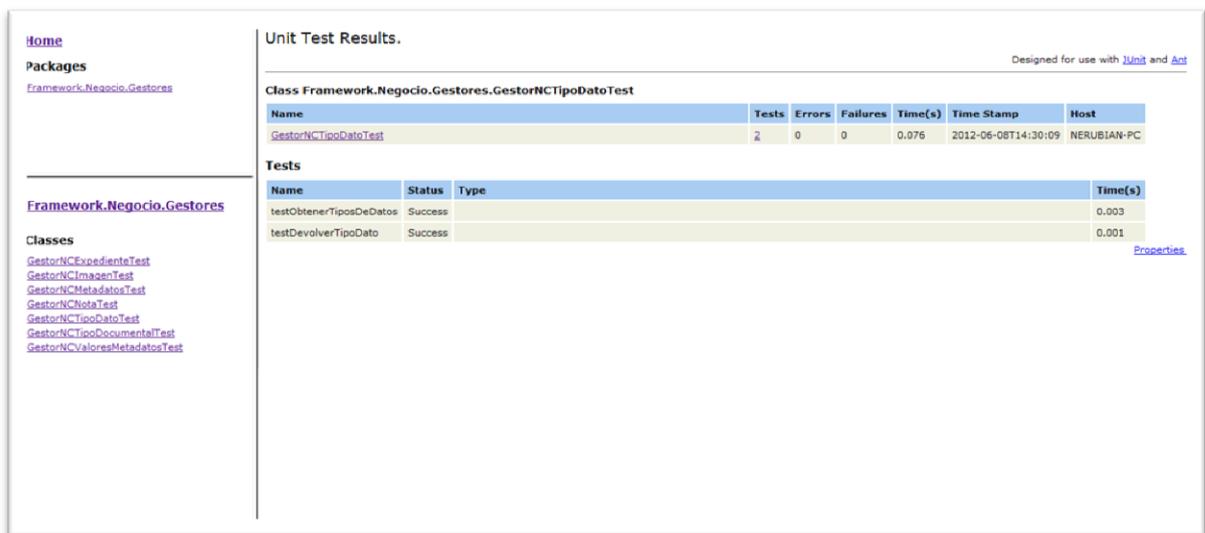


Figura 25: Pruebas GestorNCTipoDato.

GLOSARIO DE TÉRMINOS

BPM: (Business ProcessMangement), Gestión de Procesos de negocio.

Case: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador).

Cliente: Rol que se encarga de escribir las historias de usuario, asignarle prioridad y escribir las pruebas funcionales.

Desarrollador: Miembro del equipo de desarrollo que puede encarar cualquiera de los roles de XP. Programador: Rol encargado de diseñar, hacer pruebas, programar e integrar el sistema.

Equipo de desarrollo: Grupo de desarrolladores que implementa un software determinado, en el que se pueden encontrar diferentes roles.

Historia de usuario (HU): Tarjetas de papel donde se especifican los requisitos del software a implementar.

IDE: Un entorno de desarrollo integrado o en inglés *Integrated Development Environment* ('IDE') es un programa compuesto por un conjunto de herramientas para un programador.

JDBC (*Java Database Connectivity*): Es un API para trabajar con bases de datos desde Java, independientemente de la base de datos a la que accedemos.

JEE (*Java Platform, Enterprise Edition o Java EE*): Es una plataforma de programación (parte de la Plataforma Java) para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

Metodologías: Se refiere a los métodos de investigación en una ciencia. Se entiende como la parte del proceso de investigación que permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Define “Quién” debe hacer, “Qué”, “Cuándo” y “Cómo” debe hacerlo.

Metodologías de Desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Programación Extrema (XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

GLOSARIO DE TÉRMINOS

Requisitos: Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

RNF: Requerimiento no funcional.

RUP (*Rational Unified Process*, en español Proceso Unificado de Racional): Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

TIC: Tecnologías de la Información y las Comunicaciones.

TCP/IP (*Transmission Control Protocol / Internet Protocol*, en español Protocolo de Control de Transmisión / Protocolo de Internet) es uno de los protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

XML (*Extensible Markup Language*, en español Lenguaje de Marcas Extensible): Es un metalenguaje extensible de etiquetas. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).