

Universidad de las Ciencias Informáticas



Facultad 3

Título: Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

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

Autores: Alejandro Hernández Toledo

Liliana Borges Piedra

Tutor: Ing. Yadira Lizama Mue

Ciudad de la Habana, Junio 2011

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Liliana Borges Piedra

Alejandro Hernández Toledo

Firma del Autor

Firma del Autor

Yadira Lizama Mue

Firma del Tutor

Dedicatoria

Liliana

Dedico este trabajo a mis padres que desde niña me han guiado por el buen camino, por seguirme y confiar siempre en mí, por ser ejemplos de sacrificio y entrega durante todos estos años de vida, por todo ese amor y cariño que me han brindado, a ustedes que hicieron que este sueño se hiciera realidad. Todo lo que soy hoy, es gracias ustedes.

Alejandro

Dedico este trabajo a mi madre y a mi padre que aunque no esté aquí me siento orgulloso por el hecho de darme la vida y es lo menos que puedo hacer por él , a mi madre por enseñarme desde niño el sacrificio de estudiar y de graduarme algún día ...de verdad muchas gracias mamá ...este es mi mayor regalo.

Agradecimientos

Liliana

A mis padres por ser los mejores del mundo y ejemplos de sacrificio y entrega:

A mi mamita linda, por siempre confiar en mí, por estar con los brazos abiertos siempre que te necesito, por aconsejarme y apoyarme en mis decisiones, por todo el amor y cariño que me has dado.

A mi papito querido desde niña, por ser mi guía, por quererme y cuidarme, por tenerme siempre presente, por todo el amor y cariño que me has dado.

A mi familia por todo su apoyo en estos 5 años en especial:

A mi hermano por quererme y ayudarme con su apoyo incondicional.

A mi abuelita Isabel y mi tío Silfredo por quererme tanto, cuidarme y tenerme siempre presente.

A mis tías Bertha y Gela por brindarme siempre su apoyo, por sus consejos y por confiar siempre en mí.

A mis primas Mailín y Mileidis las que siempre están cuando más las necesito, por apoyarme siempre, por todos los consejos y por quererme como a una hermanita.

A mi segunda mamita linda Cary, por preocuparse siempre por mí, por apoyarme y aconsejarme en mis decisiones, por ser la madre de una de las personas que más quiero en mi vida y por quererme tanto.

A Argilagos por ser tan especial, por tener un enorme corazón, por ser mi amigo y novio, por apoyarme siempre, por sus consejos, por complacerme mis caprichos, porque sé que aunque estés lejos siempre estaré presente en ti.

A mis amig@s por estar siempre presentes en mi corazón, sin ustedes la vida en la UCI no hubiese sido la misma:

A Pedro mi amigo por quererme tanto, por tu apoyo incondicional, por tus consejos, por estar siempre presente cuando más te he necesitado y por tu preocupación por mí.

A Maye por ser tan especial, por ser esa amiga que no deja que estés triste por nada ni por nadie, por quererme como una hermana, por estar a mi lado en los momentos buenos y malos en estos 5 años, por esos días felices en tu casa junto a tus padres que los adoro y quiero muchísimo: Reina y Roly.

A Mairobys mi niña, por ser esa amiga que te hace sonreír cuando tienes deseos de llorar, por quereme como una hermana, por estar siempre a mi lado en los momentos buenos y malos desde que nos conocimos.

A Mary por sus consejos y su apoyo siempre que lo necesité.

A mis compañeras de apartamento a Lisbet, Arisday, Nany, Leanet, Danae y Yanila por su ayuda, por su preocupación y apoyo.

A mi compañero de tesis: por ayudarme en la realización de este trabajo.

A mi tutora por su comprensión y apoyo, guiándome y orientándome en todo momento.

A los compañer@s del proyecto a Sandra, Rene, Lisday y a los profes Eliober.

A todos los que de una forma u otra aportaron su granito de arena y me ayudaron en la realización de este trabajo.

Alejandro

A mi madre por ser una persona muy especial por su ejemplo y sacrificio de madre y padre durante todos estos años.

*A mi hermana por apoyarme siempre y confiar en mí durante todos estos años de la carrera
A mi novia Sheila por confiar en mí y ser tan paciente en estos últimos meses.
A toda mi familia por su incondicional apoyo y por darme siempre una mano donde apoyarme cuando la necesitaba...*

A mis amigos por soportarme todos estos años en la uci...y por todas las cosas buenas por las que pasamos a Ever L, José Alejandro, Manuel Alejandro, Aliandro, Aurelio, René, Angel Gabriel, Yudier...y al piquete de movemos la uci...

*A mi compañera de tesis: por su paciencia en estos meses de realización en la tesis y su apoyo
A mi tutora Yadira por su comprensión y ayuda en todo momento.*

A los compañer@s del proyecto a Sandra, René, Lisday y al profe Eliober.

A todos los que de una forma u otra aportaron su granito de arena y me ayudaron en la realización de este trabajo.

RESUMEN

El presente proyecto de tesis se desarrolla para el Centro de Digitalización de la División de Antecedentes Penales, institución adscrita a la Dirección General de Justicia, Instituciones Religiosas y Cultos, perteneciente al Viceministerio de Política Interior y Seguridad Jurídica que registra y emite las certificaciones de antecedentes penales de los ciudadanos venezolanos. Como parte de la Solución Tecnológica Integral para la Automatización y Modernización de la institución, el centro tiene como objetivo garantizar la obtención de documentos digitales con valor legal y metadatos asociados, a partir de la digitalización de su Fondo Documental Histórico. Vencida la etapa de Especificación de requisitos, es necesario desarrollar el sistema que apoyará la realización de los procesos del centro. El objetivo de este trabajo es realizar el diseño e implementación de los módulos Recepción y Devolución de Documentos, Control de Calidad y Otorgamiento del Centro de Digitalización que permita la modernización del fondo documental de la División de Antecedentes Penales. Se utilizaron diferentes herramientas y tecnologías para el desarrollo, definidas previamente por el equipo de proyecto. Los principales resultados esperados son el diseño, implementación del subsistema correspondiente y validación de la solución propuesta a partir de la aplicación de métricas para diseño y pruebas de caja negra para la implementación.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación Teórica	6
Introducción.....	6
1.1 La Gestión Documental	6
1.2 Digitalización de documentos	7
1.3 Proceso de digitalización de documentos	7
1.4 Centros y aplicaciones para la digitalización. Estrategias y tendencias mundiales y nacionales.....	10
1.4.1 En el mundo	10
1.4.2 En Cuba	14
1.4.3 En la Universidad	15
1.5 Valoración crítica.....	16
1.6 Tecnologías utilizadas para el desarrollo.....	16
Conclusiones.....	24
CAPÍTULO 2: Características y Diseño del sistema	25
Introducción.....	25
2.1 Propuesta de sistema.....	25
2.1.1 Especificación de los requisitos de software.....	25
2.1.2 Definición de casos de uso.....	26
2.2 Arquitectura	29
2.3 Patrones de diseño.....	31
2.4 Paquetes del subsistema DigiDAP	33
2.5 Modelo de diseño	34
2.5.1 DCD- Recepcionar expediente	35
2.5.2 Descripción de clases.....	35
2.5.3 Estándares del diseño	39
2.6 Diseño de la base de datos	40
2.6.1 Modelo de datos.....	40
2.6.2 Descripción de las tablas.....	40
Conclusiones.....	43

CAPÍTULO 3: Implementación y validación de la solución propuesta.....	45
Introducción.....	45
3.1 Diagrama de componentes del sistema.....	45
3.2 Arquitectura de información.....	46
3.3 Estándares de codificación.....	47
3.4 Validación de la solución propuesta.....	49
3.4.1 Métricas para el modelo de diseño.....	49
3.5.2 Validación de la implementación.....	63
Casos de Prueba.....	63
Conclusiones.....	65
CONCLUSIONES.....	66
RECOMENDACIONES.....	67
BIBLIOGRAFÍA.....	68
Anexo 1.....	¡Error! Marcador no definido.
Descripción de clases del diseño.....	¡Error! Marcador no definido.
DCD- Dar entrada a expediente.....	¡Error! Marcador no definido.
DCD- Devolver expediente.....	¡Error! Marcador no definido.
DCD- Revisar Trámite.....	¡Error! Marcador no definido.
DCD- Revisar Unidad Documental.....	¡Error! Marcador no definido.
DCD- Otorgar Trámite.....	¡Error! Marcador no definido.
DCD- Buscar Trámite.....	¡Error! Marcador no definido.
DCD- Firmar Trámite Digitalmente.....	¡Error! Marcador no definido.
Anexo 2.....	¡Error! Marcador no definido.
Casos de Prueba.....	¡Error! Marcador no definido.
CP_RecepcionarExpedientes.....	¡Error! Marcador no definido.
CP_Dar Entrada a Expedientes.....	¡Error! Marcador no definido.
CP_Devolver expedientes.....	¡Error! Marcador no definido.
CP_Revisar Trámite.....	¡Error! Marcador no definido.
CP_Revisar Unidad Documental.....	¡Error! Marcador no definido.
CP_Buscar Trámite.....	¡Error! Marcador no definido.

CP_Otorgar Trámite de Digitalización ¡Error! Marcador no definido.
CP_Firmar Trámites ¡Error! Marcador no definido.
No conformidades ¡Error! Marcador no definido.
Glosario de términos70

INTRODUCCIÓN

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana. Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva. Es evidente que para los países subdesarrollados resulta un reto el logro de este propósito. Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones y ha hecho extensiva nuestra experiencia hacia otros países. Como expresó nuestro Comandante en Jefe Fidel:

“La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación...” (1)

El **Convenio Integral de Cooperación Cuba - Venezuela** se firmó en Caracas en el año 2000, mediante el cual los gobiernos de ambas naciones conscientes de su interés común por promover y fomentar el progreso de sus respectivas economías y las ventajas recíprocas que resultan de una cooperación que tenga resultados efectivos en el avance económico y social de ambos países y la integración de América Latina y el Caribe, se comprometieron a elaborar de común acuerdo, programas y proyectos de cooperación.

Como resultado del convenio se decide realizar el proyecto de desarrollo **Solución Tecnológica Integral para la Automatización y Modernización de la División de Antecedentes Penales** que tiene como objetivo garantizar la calidad, confiabilidad y rapidez en la gestión y digitalización de la información contenida en su archivo histórico.

La Solución de software, subproyecto firmado en el contrato, está dividida en dos subsistemas. Sistema para la Gestión de Antecedentes Penales (SIGESAP), que mejorará los procesos de inscripción y certificación de antecedentes penales y Centro de Digitalización (DigiDAP), que garantizará la incorporación de los documentos del archivo a SIGESAP.

La División de Antecedentes Penales (DAP) es una institución adscrita a la Dirección General de Justicia, Instituciones Religiosas y Cultos, perteneciente al Viceministerio de Política Interior y Seguridad Jurídica de Venezuela. Registra los antecedentes penales de los ciudadanos venezolanos y emite las

certificaciones de antecedentes penales a los distintos entes tanto nacionales como internacionales autorizados por la legislación venezolana vigente.

En el archivo de la DAP existe una gran cantidad (no apreciada) de unidades documentales almacenadas alfanuméricamente de diversos tipos. La técnica utilizada para relacionar dichos documentos hace que su localización sea bastante engorrosa. Cuando llega al archivo un expediente resultado de un proceso de inscripción que guarda relación con otros expedientes ya almacenados se escribe en la carátula del mismo el identificador alfanumérico del nuevo es decir los documentos relacionados a un mismo proceso penal están dispersos. Este proceso no es efectivo en todos los casos debido a la gran dispersión de la información, por lo que es muy engorroso encontrar todos los datos referentes a un mismo proceso con los mecanismos de recuperación de la información que tienen en la actualidad.

Utilizan un sistema informático que permite la inscripción y certificación de antecedentes penales llamado SIRCAP. Este sistema gestiona la información referente a pocos años de trabajo de la oficina por lo que es difícil la localización de la documentación con independencia del mismo, lo que provoca una vinculación parcial entre lo real existente en el archivo y además no contempla alternativas de manipulación y conservación digital de la información que se consulta manualmente cada vez que es necesaria mediante solicitudes de préstamo al Archivo. El nivel de deterioro de los expedientes aumenta con el tiempo que llevan archivados debido a las condiciones del local y la constante manipulación de los mismos. Algunos ejemplos que evidencian el deterioro son: papel rasgado, tinta corrida, escritura borrosa. DigiDAP tiene como objetivo garantizar la obtención de documentos digitales con valor legal y metadatos asociados a partir de la digitalización del Fondo Documental Histórico de la institución para contribuir a la mejora de los procesos y la calidad de los servicios de la misma.

Está concebido en 6 áreas de procesos fundamentales:

Recepción y devolución de documentos donde se da entrada a los expedientes que serán digitalizados y salida a los que hayan culminado satisfactoriamente el proceso.

Preparación de documentos donde se garantiza las condiciones necesarias para proceder a la digitalización de los documentos. Se realiza la limpieza de expediente; guillotinado de las páginas; estiramiento de folios arrugadas.

Digitalización de documentos donde se crea el documento digital multipágina a partir de los folios contenidos dentro del expediente.

Asociación de metadatos donde se registran los metadatos asociados a las unidades documentales de un expediente digital. Ocurre la validación de los datos insertados y la identificación de posibles errores contenidos en los documentos.

Control de calidad donde se verifica la calidad del trámite realizado, enfatizando en la calidad del documento digital y los metadatos asociados. En caso de errores el expediente regresa al área donde se produjo el mismo para su corrección.

Otorgamiento donde se certifica a través de la firma digital y manuscrita la legalidad del proceso.

Desarrollada la etapa de análisis, donde se obtuvieron los artefactos que describen las funcionalidades que debe cumplir el sistema se da paso a la próxima etapa de desarrollo donde se identifica el siguiente **problema de investigación** a resolver:

¿Cómo satisfacer los requisitos acordados con los clientes para la realización de los procesos de Recepción y Devolución de documentos, Control de Calidad y Otorgamiento del Centro de Digitalización?

El **objeto de estudio** de la investigación es el Proceso de desarrollo de Software para la gestión de documentos con valor legal.

El **objetivo general** del trabajo es realizar el diseño e implementación de los módulos de Recepción y Devolución de Documentos, Control de Calidad y Otorgamiento del Centro de Digitalización para obtener la solución informática que logrará satisfacer los requisitos acordados con los clientes.

El **campo de acción** son los procesos de digitalización de documentos para la División de Antecedentes Penales.

Idea a defender

Con el diseño e implementación de los módulos de Recepción y Devolución de Documentos, Control de Calidad y Otorgamiento del Centro de Digitalización se logrará satisfacer los requisitos acordados con los clientes para la realización de los procesos correspondientes.

Objetivos Específicos

1. Elaborar el marco teórico de la investigación para lograr la correcta selección de la metodología y las tecnologías a utilizar en el desarrollo de la solución.
2. Realizar el diseño de los módulos correspondientes a partir de análisis del modelo del sistema para generar los artefactos que permitan obtener la primera visión de la implementación.

3. Realizar la implementación de los elementos del diseño para garantizar la obtención de un producto informático listo.
4. Validar la solución propuesta para evaluar el grado de calidad y fiabilidad de la misma.

Para dar cumplimiento a los objetivos planteados se planificaron las siguientes **tareas**:

- Análisis de las principales herramientas, metodologías y tecnologías para la solución del problema.
- Análisis del estado del arte sobre soluciones existentes para la digitalización de documentos.
- Estudio de los artefactos Modelo de Procesos de Negocio, Especificación de requisitos de software y Modelo de sistema del Centro de Digitalización.
- Análisis de la arquitectura definida por el equipo de desarrollo.
- Realización del Modelo de Diseño para los módulos del sistema correspondientes.
- Implementación de los módulos correspondientes de acuerdo a las normas y estándares establecidos por el equipo de proyecto.
- Aplicación de métricas orientadas a objetos para validar el diseño.
- Diseño y aplicación de pruebas de caja negra a los módulos implementados para validar la solución propuesta.
- Redacción de la investigación realizada.

Métodos de investigación

Métodos Teóricos:

Análisis y Síntesis: Ha servido para analizar y comprender la información relacionada con la gestión documental, la digitalización de documentos con valor legal y herramientas para el desarrollo, logrando la asimilación de los elementos más importantes relacionados con el objeto de estudio.

Histórico – Lógico: Se ha utilizado para estudiar la evolución de los sistemas informáticos para la gestión documental que se enmarcan en la digitalización de documentos realizando énfasis en las características más importantes que han surgido.

El contenido a desarrollar en la investigación está estructurado en 4 capítulos:

Capítulo 1: Fundamentación Teórica: se abordan aspectos sobre la digitalización de documentos, una breve descripción, definición y ventajas. Posteriormente se hace un estudio del estado actual de centros y aplicaciones existentes en el mundo, en Cuba y nuestra universidad que desarrollan esta rama y las

características de las tecnologías, metodología y herramientas a utilizar en el desarrollo de nuestro trabajo.

Capítulo 2: Características y Diseño del sistema: se propone el diseño de la solución a partir de la especificación de los requisitos de software y los diagramas de casos de uso del sistema. Se exponen los patrones de diseño utilizados y se realiza una breve descripción de la arquitectura para lograr el entendimiento del diseño. Se muestran como resultado del modelo de diseño, los diagramas de clases y la descripción de cada una de ellas por casos de uso, además las normas de diseño, el diagrama de paquetes y el modelo de datos utilizado en el sistema.

Capítulo 3: Implementación y validación de la solución propuesta: se muestran los diagramas de componentes por capas: Presentación, Negocio, Acceso a Datos. Se exponen las normas y estándares de codificación y de diseño de interfaz utilizadas y definidas por el equipo de desarrollo. Se evalúa el grado de calidad y fiabilidad de los datos de los resultados obtenidos para el diseño con métricas orientadas a objetos y la implementación con pruebas de caja negra.

CAPÍTULO 1: Fundamentación Teórica

Introducción

El presente capítulo enmarca la fundamentación teórica de la investigación se abordan aspectos sobre la digitalización de documentos, una breve descripción, definición y ventajas. Posteriormente se hace un estudio del estado actual de centros y aplicaciones existentes en el mundo, en Cuba y nuestra universidad que desarrollan esta rama y las características de las tecnologías, metodología y herramientas a utilizar en el desarrollo de nuestro trabajo.

1.1 La Gestión Documental

Para dar inicio se dará una serie de conceptos relacionados a la gestión documental con lo que se pretende ubicarnos en el contexto del tema que estaremos tratando a lo largo de este trabajo.

¿Qué es un documento?

Un documento es aquello que “documenta” o “consigna” algo con un propósito intelectual deliberado. Aunque el concepto de documento es universal, se reconoce que algunas culturas son más “documentales” que otras. (2)

Podemos decir entonces que, “La gestión documental abarca el ciclo de vida completo de los documentos, es decir, el tratamiento secuencial y coherente que se da a los documentos desde que se producen o reciben en las distintas unidades hasta el momento en que son eliminados o conservados, en función de su valor testimonial o histórico como fuente para el conocimiento de la trayectoria de la Universidad.” (Universidad de Málaga. Archivo General.2006). Por otro lado se comenta que la gestión documental según Fernández Gil, Paloma (1999) “pretende abarcar desde la elaboración de los documentos hasta su servicio, pasando por su organización y descripción.” Según el planteamiento que se hace en la Ley General de Archivos de Colombia, la Gestión documental es un “Conjunto de actividades administrativas y técnicas tendientes a la planificación, manejo y organización de la documentación producida y recibida por las entidades, desde su origen hasta su destino final, con el objeto de facilitar su utilización y conservación”. Ahora podemos decir, tomando las ideas anteriores, que la gestión documental consiste, en el tratamiento y conservación que se les da a los documentos, desde el principio de su ciclo de vida, es decir, la producción del mismo, hasta su eliminación o conservación permanente, todo esto siguiendo las diversas etapas que constituyen el ciclo de vida de los documentos, y por supuesto respetando el principio de orden original y el principio de procedencia. (3)

La Unesco plantea en el Programa de Gestión Documental, dándole a este término una definición muy sencilla que aclara el alcance en la práctica profesional: “Esfera de la gestión administrativa encaminada a conseguir la economía y la eficacia de las operaciones con vista a la creación, mantenimiento, utilización y por último a la eliminación de los documentos o a su conservación definitiva durante su ciclo vital”.

Desde hace algunos años los fondos de los centros de documentación vienen incorporando archivos en soporte digital creados a partir de la digitalización de los documentos. Estas copias digitales cumplen una doble finalidad: por un lado, facilitar el acceso a la información y la divulgación del conocimiento, y por otro, preservar y conservar los documentos originales.

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.” (4)

1.2 Digitalización de documentos

La digitalización de documentos permite su consulta on-line y facilita el acceso a la información a través de diferentes dispositivos electrónicos. Pero, además, asegura la conservación de muchos de los fondos más valiosos y antiguos de las bibliotecas y archivos. El paso del tiempo o una mala conservación pueden deteriorar estos documentos, de modo que muchos centros documentales se ven obligados a restringir el acceso público a determinados ejemplares. (5)

El uso de la digitalización y conversión 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?

Digitalizar es “expresar datos en forma digital.” (4)

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

1.3 Proceso de digitalización de documentos

El proceso de digitalización se realiza a través de distintos procedimientos electrónicos. Para obtener archivos digitales, a partir de imágenes analógicas reproducidas en soportes físicos, una de las herramientas necesarias es el escáner, utilizado, principalmente, para la digitalización de fotografías o en los casos en los que se requiere una digitalización lo más fiel posible al original; por ejemplo, cuando las

páginas de un libro o documento tienen valor en sí mismas por la presencia de grabados, dibujos u otros detalles.

Por el contrario, cuando el valor del archivo reside tan sólo en la información contenida en el texto, se pueden utilizar técnicas de reconocimiento óptico de caracteres (OCR) que transforman el texto en un formato editable, creando una nueva paginación y elaborando una versión moderna del original.

Otra posibilidad es la microfilmación del documento. Esta opción es más costosa pero los resultados suelen ser de mayor calidad y su inclusión en la base de datos de la biblioteca digital es mucho más sencilla.

En cualquiera de los casos, en la captura de imágenes digitales se deben tener en cuenta tanto los procesos técnicos, como los propios atributos de los documentos fuente: dimensiones físicas y presentación, nivel de detalles, rango tonal, y presencia de color.

Se definen como procesos comunes:

Recepción y devolución de los documentos

El archivo entregará la documentación debidamente preparada y ordenada, habiendo verificado el estado de los documentos; es decir si los documentos poseen roturas, dobleces, entre otros, procederá al planchado y parchado de ser necesario. El subproceso Preparación de documentos recibirá la documentación y verificará la cantidad recibida y registrará las observaciones referentes a la cantidad o estado de conservación que pudiera encontrar.

La devolución de los documentos al archivo, luego que ya han pasado por los procesos anteriores, a fin de ser reorganizados por el personal autorizado, de manera tal que se garantice que la unidad de información retorne en perfectas condiciones.

Control de calidad

Los factores claves en la valoración de la calidad de la imagen son la resolución, el color, el tono y la apariencia general. La calidad de la imagen final está determinada por una variedad de factores entre los que intervienen el rendimiento del sistema de captura, la resolución, el rango dinámico y la precisión del color.

Una vez se ha finalizado el proceso de digitalización suele realizarse un control de calidad de la imagen con el objetivo de asegurar que se han cumplido las expectativas en cuanto a calidad de los resultados

finales. Este control abarca procedimientos y técnicas para comprobar la calidad, precisión y consistencia de las imágenes y pueden ser implementadas en diferentes niveles: evaluación inicial (verifica las decisiones técnicas tomadas durante la evaluación de referencia) y evaluación continuada (ampliación del proceso de calidad a la totalidad de la colección para asegurar la calidad de todo el programa de digitalización de imágenes). (5)

Otorgamiento

La Firma digital es como nuestra firma manuscrita, permite acreditar nuestra identidad. Una firma válida implica que se nos puede atribuir de forma indudable ese documento electrónico como autores de esa firma. Implica que no podemos negar en ningún caso que el documento ha sido firmado por nosotros. No podemos repudiarlo. La firma de un documento nos permite confirmar su Integridad. Si sufriese alguna modificación desde el momento de su firma hasta que se consulta de nuevo u otra persona lo recibe, la firma ya no será válida.

Para firmar un documento de forma digital se necesita un certificado digital pues para firmar tendrá que acreditarse. Mediante el certificado establezco mi identidad.

Un certificado digital es un documento firmado electrónicamente por una autoridad de confianza que establece que “yo soy quien digo ser” lo que me permite identificarme en la comunidad electrónica. Incluye, entre otros elementos, una clave pública y una privada que están emparejadas. Gracias a una autoridad de confianza que así lo acredite, podremos confiar en que se corresponden con la identidad de una persona física o jurídica determinada. (6)

El certificado, es fundamental en el proceso de firma. El uso del certificado digital nos permite identificar al emisor de los mensajes a través de la clave pública contenida en su certificado. En este sentido sin el certificado, no podríamos garantizar la autenticación de origen y el no repudio.

La Autoridad de Certificación (CA) es la agencia responsable de emitir los certificados y debe ser una entidad en quien las partes involucradas confían. Para ello, el contenido de un certificado incluye la firma digital de la Autoridad de Certificación. (6)

El contenido de un certificado es variado, pero como norma general contiene:

La clave pública y el nombre del propietario; puede también incluir la clave privada

La fecha de expedición

Período de validez del certificado

Un número de seriado

La identificación de la Autoridad de Certificación.

1.4 Centros y aplicaciones para la digitalización. Estrategias y tendencias mundiales y nacionales.

1.4.1 En el mundo

Hoy en día a nivel mundial existen empresas y productos que desarrollan esta rama, cada uno con sus características y particularidades.

KernelDoc

Está dedicada a la prestación de servicios de asesoramiento de empresa, orientada a proporcionar la colaboración necesaria para desarrollar y potenciar todas las ventajas estratégicas y competitivas que aportan las telecomunicaciones y tecnologías de la información, dedicándose específicamente a la prestación de Servicios para el Tratamiento de la Documentación.

Asesora en temas de Gestión Documental, Tratamiento de la información y Archivística, da respuesta en temas de Productividad, Tecnología y Soporte al Cliente. (7)

Solución Integral

Recogida de documentación: disponemos de servicios de logística para efectuar la recogida de los documentos en los puntos que sean solicitados y con la frecuencia deseada.

Preparación de la documentación: eliminación de grapas, sobres, etc. realizándose también expurgos y clasificación del material.

Digitalización de documentación: obtención de la imagen digital del documento físico. En este proceso se realizan los controles de calidad requeridos según parámetros del cliente.

Indexación: los documentos necesitan su identidad para luego poder ser localizados unívocamente por el usuario; la asignación de esta información se hace a después de la digitalización y puede acometerse de forma automática, mediante la aplicación de un software específico que reconoce los datos y su ubicación en el documento como de forma manual con una actividad de entrada de datos.

Integración de los documentos en internet: integra los Documentos digitalizados en una Web, de forma automática; en una primera etapa y como proyecto piloto, el usuario puede acceder a nuestra Web y consultar los documentos en cuestión; una vez que el usuario interpreta que la solución es buena, instala la solución en su empresa, como una Intranet. (7)

ADDOC

Es una empresa mexicana especializada en servicios de digitalización y administración de documentos físicos y electrónicos. Ofrece soluciones de administración y digitalización de documentos a empresas de todo tipo. Prácticamente en cualquier negocio donde los documentos sean un componente esencial de los procesos diarios y donde la eficacia pueda ser mejorada por un mejor manejo de la información. (8)

Cuenta con los siguientes servicios: Digitalización de documentos de distintos tamaños (Carta, oficio, planos, etc.) a archivos electrónicos; Diseño de procesos de digitalización; Capacitación e instalación de software especializado para digitalización, indexación y administración de documentos, entre otros.

El proceso que realizan para digitalizar documentos es el siguiente:



Fig. 1 Proceso de Digitalización en ADDOC

Brinda Sistemas de Gestión de documentos con las siguientes características:

- Control de creación y verificación de documentos.
- Control y creación de múltiples versiones de los documentos.
- Administración y almacenamiento de los documentos de tal manera que se facilita la consulta de los mismos.
- Acceso seguro e inmediato desde cualquier lugar y en cualquier momento.
- Reporte y control de actividades realizadas por los usuarios que tienen acceso al sistema.
- Protección de los documentos de tal manera que previene el acceso no autorizado a la información y permite la recuperación del daño físico o pérdida de documentos.

- Creación de políticas de retención para archivar documentos antiguos al terminar su ciclo de vida.
- Administración de cualquier tipo de archivo electrónico.
- Soluciones 100% web.
- Capacidad de establecer procesos predefinidos para el flujo de los documentos.

FENA

Es un centro de documentación y digitalización español, cuenta con el software de gestión documental XeroxDocushare, un potente sistema, basado en Web, de gestión documental y de contenidos que permite almacenar, colaborar y compartir información. La labor de este Centro consiste en la digitalización de los fondos históricos de empresas y organizaciones empresariales y sindicales de la región para su posterior inclusión en una biblioteca virtual de acceso gratuito a través de Internet. Cuenta con los siguientes subprocesos:



Fig. 2 Proceso de Digitalización en FENA

En definitiva, supone la creación de un fondo documental virtual que recoja la historia de las empresas asturianas y su devenir a lo largo del tiempo.

QuickScan

Es una aplicación de procesamiento digital de imágenes de alto rendimiento para Microsoft Windows dotada de las siguientes características:

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.

Visualización: es también un visor de imágenes de alto rendimiento que incluye funciones diseñadas para facilitar la visualización y manipulación de las imágenes tales como un visor principal, un panel de miniaturas, funciones de dimensionamiento rápido, rotación, anotaciones, conversión de imágenes binarias o una ventana panorámica.

Detección automática de color: El filtro de detección automática de color (ACD) distingue de forma automática las páginas en color de las páginas binarias durante la digitalización.

MultiStreaming: le permite activar la función MultiStream, que le permite crear las imágenes en color y binarias a partir de una única imagen origen.

Procesamiento de imágenes: el conjunto de filtros de procesamiento de imágenes de QuickScan es capaz de limpiar imágenes sucias, enderezar imágenes torcidas por causa del alineamiento incorrecto del papel durante la digitalización, eliminar los círculos negros originados por perforaciones de las imágenes digitalizadas, etc. Los filtros se pueden usar de forma individual o secuencial y se pueden configurar para obtener un resultado óptimo. Los filtros de procesamiento de imágenes pueden aplicarse a imágenes ya existentes abiertas en QuickScan o bien de forma automática durante la digitalización de un lote.

Separación de trabajos: permite el uso de páginas de separación de trabajos, que permiten dividir los documentos de su lote en varias imágenes de múltiples páginas. Puede separar sus lotes mediante códigos de barras, códigos de parche o páginas en blanco.

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.

Indexación: permite a los usuarios indexar documentos y agregarles información pormenorizada acerca de su organización interna antes de guardarlos o exportarlos.

Exportación: permite exportar información de datos de índice y de archivos de lotes.

Sellado digital: permite añadir un sello digital a las páginas de un lote.

Anotaciones: las herramientas de anotación de QuickScan le permiten añadir texto, líneas, resaltadores, flechas y otros objetos que resalten una determinada zona de la página con el fin de poner de relieve o ensombrecer ciertos pasajes o de añadir explicaciones en forma de texto o ventanas emergentes.

Guardar: es capaz de almacenar las imágenes en diversos formatos de archivo estándar y esquemas de compresión.

Impresión: puede imprimir imágenes mediante cualquier impresora estándar compatible con Windows.

1.4.2 En Cuba

La empresa nacional de software DESOFT ofrece soluciones integrales en Tecnologías de la Información para la informatización de la sociedad cubana. Ofrece productos y servicios a las organizaciones y posee una creciente actividad exportadora. Los paquetes de productos se clasifican en:

Aplicaciones de Gestión Empresarial y Comercial

Aplicaciones de Gestión Económica

Aplicaciones para la Toma de Decisiones

Aplicación de Gestión de Recursos Humanos

Aplicación de la Gestión de la Información que contiene los productos:

AvilaDocVersión 3.07

Es una aplicación para el control de documentos. Permite controlar la entrada y salida de documentos y gestionar el tránsito interno de los mismos, constituyendo un archivo digital que agiliza las tareas de registro, búsqueda, reproducción y distribución de los documentos. Este sistema permite ser enlazado a Microsoft Outlook para generar automáticamente tareas a cumplir, notificaciones e información asociada al documento. Ofrece un historial del documento desde que fue creado hasta su eliminación. Es una aplicación web desarrollada sobre la plataforma de software libre con una base de datos centralizada, destinada a la gestión, tramitación y resguardo de archivos electrónicos y/o digitales, facilitando la búsqueda o recuperación de información de forma rápida. (9)

Papiro

Producto informático de uso libre que emplea herramientas igualmente libres y permite conservar documentación de valor histórico al evitar su manipulación; pues, al digitalizarse el documento, este se consulta en formato electrónico. Esta versión incluye también la gestión de publicaciones periódicas, especialmente revistas. (10)

1.4.3 En la Universidad

DigiPyrus

Es un sistema de Digitalización que permite digitalizar los Tomos de los Registros y Notarías, contribuyendo a agilizar el acceso a la información que a diario se tramita en cada una de sus Oficinas.

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. Cuenta con los siguientes subprocesos:

Recepción de Documentos: donde se realiza la verificación y registro de los tomos provenientes del almacén temporal.

Preparación de Documentos: donde se realiza la ejecución de las operaciones establecidas para garantizar que los tomos cumplan con los parámetros necesarios para el proceso de digitalización.

Escaneo de Documentos: donde se realiza conversión de los tomos físicos en objetos digitales multipáginas.

Control de la Calidad: se realiza para inspeccionar el 100% del objeto digital multipágina proveniente del área de Digitalización de Documentos.

Encuadernación de Documentos: donde se realiza la recuperación y preservación de las condiciones iniciales de los tomos digitalizados.

Devolución de Documentos: donde se realiza el registro de salida de los Tomos que han sido digitalizados en el proceso.

Asociación de Metadatos: donde se realiza identificación de los datos de los objetos digitales multipáginas para la asociación de sus metadatos.

Exportación: donde se realiza la exportación de las imágenes digitales multipágina de los documentos del tomo al Centro de Datos del SAREN.

1.5 Valoración crítica

Después de un análisis realizado sobre los centros y aplicaciones para la digitalización de documentos existentes en el mundo, en Cuba y nuestra universidad se concluye que no resultan soluciones factibles debido a que algunos fueron desarrollados con software propietario lo que implica gastos muy elevados en licencias y mantenimiento, otros desarrollados en software libre pero por sus características no son adaptables a los requisitos del sistema en cuestión.

1.6 Tecnologías utilizadas para el desarrollo.

Metodología para el desarrollo de software

Para guiar el proceso de desarrollo se utiliza Rational Unified Process (RUP) ya que proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades con el propósito de asegurar la producción de software de alta calidad que se ajuste a las necesidades de los usuarios finales con unos costos y calendario predecibles. Esta metodología intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos software. Los verdaderos aspectos definitorios de RUP se resumen, de acuerdo a (13), en:

Guiado por casos de uso: La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema. RUP permite el desarrollo del software guiado por casos de uso lo que facilita la trazabilidad de los requisitos a través de los mismos durante todo el ciclo de vida de la aplicación.

Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Se representa mediante varias vistas 4+1, recibe este nombre porque lo forman las vistas lógica, de implementación, proceso y despliegue, más la de casos de uso que es la que da cohesión a todas.

Iterativo e Incremental: Para hacer manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. Esta característica ha permitido la mejora continua del sistema a partir de versiones anteriores. Además esta metodología se divide en 4 fases para el ciclo de vida de desarrollo de software, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas: la disciplina de desarrollo y la disciplina de soporte.

Herramienta CASE

Teniendo en cuenta las necesidades de la solución se decidió utilizar Visual Paradigm en su versión 3.2 para el desarrollo de la misma, debido a las facilidades de esta herramienta para el trabajo colaborativo, la integración, el trabajo con modelos relacionales de Bases de Datos, modelado de procesos de negocio a través de BPMN así como el modelado de todos los artefactos propuestos por la metodología a través de UML, en una sola herramienta. Es una poderosa herramienta CASE que al igual que Rational Rose, utiliza UML para el modelado. 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. (11) 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. Tiene integrado el MS Visio y es compatible con otras ediciones, posibilita un entorno de creación de diagramas para UML 2.x. (12)

Herramienta para el desarrollo

NetBeans IDE es un producto libre y gratuito, sin restricciones de uso. Soporta el desarrollo de todos los tipos de aplicación Java, tales como: JEE (Java Platform Enterprise Edition), JSE (Java Standard Edition) y Java Mobile Edition. 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.

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.

Debido a que soporta todas las tecnologías definidas para el desarrollo de la solución y es multiplataforma ha sido seleccionado para la implementación del sistema.

Sistema Gestor de Base de Datos

Se utilizó 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. (14)

PostgreSQL ofrece sofisticadas características como: (15)

- 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.

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

- 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.
- Ofrece control de acceso simultáneo a través de la gestión de múltiples versiones de un mismo registro (MVCC). (16)
- Las restricciones y disparadores tienen la función de mantener la integridad y consistencia en las bases de datos.
- Usa una arquitectura cliente/servidor basada en un proceso por usuario; existe un proceso maestro que se ramifica para proporcionar conexiones adicionales por cada cliente que se intenta conectar a PostgreSQL.
- Tiene incorporado el mecanismo WAL (Write-Ahead Logging), que incrementa la confiabilidad de las bases de datos al registrar los cambios antes de ser escritos al disco; lo que asegura que, en caso de ocurrir un fallo crítico en las bases de datos, exista un registro de transacciones del cual se pueda restaurar. (17)

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.

- 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).
- Soporta funciones con privilegios por usuario: que pueden definirse para ejecutarse con los derechos del usuario administrador o con los derechos de un usuario previamente definido, además retornan filas donde las salidas pueden tratarse como un conjunto de valores retornados por una consulta.
- Soporta bloques de código: que se ejecutan en el servidor y que pueden ser escritos en diferentes lenguajes de programación con la potencia que presenta cada uno.
- El máximo tamaño de bases de datos es ilimitado: depende del sistema de almacenamiento. (18)
- El máximo número de filas por tablas es ilimitado. (19)

Lenguaje de Programación

El lenguaje de programación seleccionado por el equipo de desarrollo es Java debido a que 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 de una gran cantidad de características que lo convierte en la herramienta ideal para la creación aplicaciones distribuidas. (20)

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, 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.

Swing

Es un conjunto de clases de java que simplifica la construcción de aplicaciones de escritorio. Permite tener un sistema de ventanas y componentes gráficos, independiente del sistema operativo y la librería de dibujo que se tengan disponibles en las máquinas clientes. Se utiliza para el diseño y construcción de los formularios de la aplicación debido a las ventajas que ofrece para el desarrollo de componentes que permiten la visualización y registro de los datos.

Tecnología 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. (21)

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).

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). (21)

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.

Los **Beans de entidad** (Entity Beans, por su término en inglés) Representan un objeto concreto que tiene existencia en alguna base de datos. Una instancia de un Bean de entidad representa una fila en una tabla de la base de datos, es decir es la representación en el modelo orientado a objetos de un registro en el modelo relacional. (21)

Los **Beans dirigidos por mensajes** (Message Driven Beans) Pueden escuchar mensajes de un servicio de mensajes JMS. Los clientes de estos Beans nunca los llaman directamente, sino que es necesario enviar un mensaje JMS para comunicarse con ellos. Los Beans dirigidos por mensajes no necesitan objetos EJBObject porque los clientes no se comunican nunca con ellos directamente. Permite el envío de notificaciones asíncronas entre los clientes y el servidor. (21)

Por último, un **Bean de sesión** representa un proceso o una acción de negocio. Normalmente, cualquier llamada a un componente del servidor debería comenzar con una llamada a un Bean de sesión. Son utilizados para publicar las funcionalidades de lógica de negocio en el servidor de aplicaciones para que sea invocada desde aplicaciones clientes u otros Bean de sesión. Estos pueden ser Sin Estado

(Stateless), o sea que no guardan el estado de las transacciones que permiten realizar o Con Estado (Statefull), que significa que si lo realizan. (21)

EJB ofrece un conjunto de funcionalidades que serán de gran ayuda para el desarrollo de la solución y permitirán construir y desplegar aplicaciones de manera rápida. A continuación se muestran algunas de las funcionalidades más importantes:

Manejo de transacciones: apertura y cierre de transacciones asociadas a las llamadas de los métodos de un Bean.

Seguridad: comprobación de permisos de acceso a los métodos del Bean.

Concurrencia: llamada simultánea a un mismo Bean desde múltiples clientes.

Servicios de red: comunicación entre el cliente y el Bean en máquinas distintas.

Gestión de recursos: gestión automática de múltiples recursos, como colas de mensajes, bases de datos o fuentes de datos en aplicaciones heredadas que no han sido traducidas a nuevos lenguajes/entornos y siguen usándose en la empresa.

Persistencia: sincronización entre los datos del Bean y tablas de una base de datos.

Gestión de mensajes: manejo de Java Message Service (JMS).

Escalabilidad: Posibilidad de constituir clúster de servidores de aplicaciones con múltiples hosts para poder dar respuesta a aumentos repentinos de carga de la aplicación con sólo añadir hosts adicionales.

Adaptación en tiempo de despliegue: posibilidad de modificación de todas estas características en el momento del despliegue del Bean.

Para el desarrollo de aplicaciones de escritorios con arquitectura Cliente / Servidor basadas en el desarrollo de componentes EJB ha demostrado ser una solución de alta calidad.

JPA

Se utiliza para el acceso a datos Java Persistence API (JPA) que proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos. Es una API de persistencia de POJOs (PlainOld Java Object). Es decir, objetos simples que no heredan ni implementan otras clases. (22)

Servidor de aplicaciones

Un servidor de aplicaciones no es más que un software que se ejecuta en un sistema de cómputo, el cual se encarga de ejecutar ciertas aplicaciones informáticas. Este tipo de software es muy común cuando de

aplicaciones Web se trata, lo que no quiere decir que sea el único tipo de aplicaciones que pueda desplegarse.

Los principales beneficios de la utilización de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo y el mantenimiento de las aplicaciones.

Glassfish es un servidor de aplicaciones que implementa la plataforma JavaEE5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías.

Conclusiones

En este capítulo se analizó el proceso de digitalización haciendo énfasis en las principales áreas a desarrollar en la investigación. Se ejemplificó la existencia de soluciones informáticas que se utilizan para la digitalización de documentos con valor legal en Cuba, el mundo y en nuestra universidad. Culminando con el análisis de las principales metodologías, herramientas y tecnologías útiles para el desarrollo de software definidas por el proyecto con el fin de alcanzar la calidad, confiabilidad y rapidez en la gestión y digitalización de la información.

CAPÍTULO 2: Características y Diseño del sistema

Introducción

En el presente capítulo se propone el diseño de la solución a partir de la especificación de los requisitos de software y los diagramas de casos de uso del sistema. Se exponen los patrones de diseño utilizados y se realiza una breve descripción de la arquitectura para lograr el entendimiento del diseño. Se muestran como resultado del modelo de diseño, los diagramas de clases y la descripción de cada una de ellas por casos de uso, además las normas de diseño, el diagrama de paquetes y el modelo de datos utilizado en el sistema.

2.1 Propuesta de sistema

Con el propósito de darle cumplimiento al problema de investigación se propone el diseño de las áreas de proceso que necesitan ser informatizadas para garantizar la funcionalidad del Centro de Digitalización.

2.1.1 Especificación de los requisitos de software

La especificación de requisitos fue realizada por el equipo de analistas los cuales sirvieron de base para el sistema a construir. En el expediente de proyecto se encuentra el documento donde quedaron recogidos los requisitos funcionales y no funcionales con los que debe cumplir el sistema. (23) A continuación se muestran los requisitos funcionales del módulo.

1. RF_D.001 Recepcionar expedientes.
2. RF_D.002 Actualizar cantidad de folios de unidad documental.
3. RF_D.003 Añadir unidades documentales a un expediente
4. RF_D.004 Eliminar unidades documentales de un expediente.
5. RF_D.005 Dar entrada a expedientes al Centro de Digitalización.
6. RF_D.006 Devolver expedientes.
7. RF_D.017 Mostrar datos generales de unidad documental
8. RF_D.027 Revisar trámites.
9. RF_D.028 Revisar unidad documental de expedientes.
10. RF_D.036 Emitir respuesta del proceso de revisión
11. RF_D.039 Buscar trámites a otorgar.
12. RF_D.040 Otorgar trámites.
13. RF_D.041 Buscar trámites pendientes de firma digital.

- 14. RF_D.049 Actualizar área del trámite del Centro de Digitalización
- 15. RF_D.050 Actualizar estado de trámite de Centro de Digitalización.
- 16. RF_D.051 Iniciar trámite.

2.1.2 Definición de casos de uso

Recepción y devolución de documentos

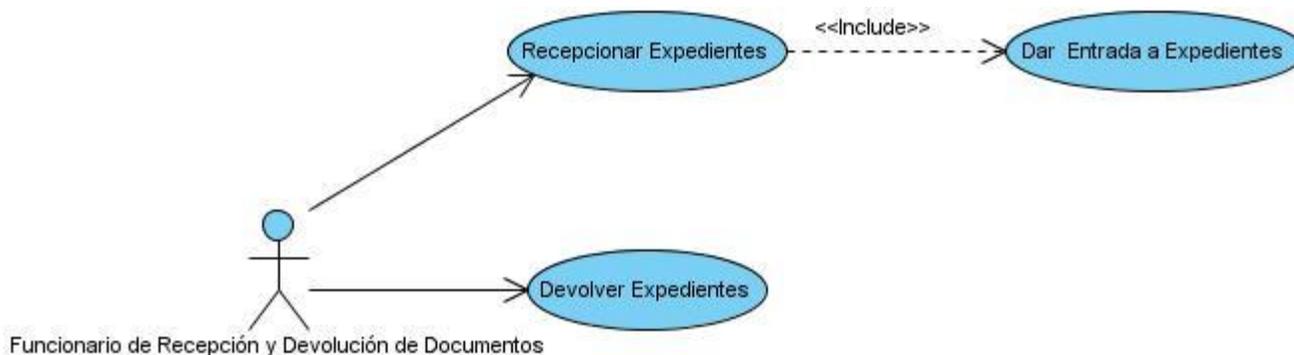


Fig. 3 Diagrama de caso de uso del proceso Recepción y devolución de documentos

Recepcionar expediente: El caso de uso consiste en recepcionar expedientes provenientes del archivo físico para ser digitalizados. El actor puede actualizar la cantidad de folios de las unidades documentales; así como eliminar y añadir unidades documentales al expediente para garantizar la veracidad de los datos del mismo antes de darle entrada al Centro de Digitalización. El caso de uso inicia cuando el Funcionario de Recepción y Devolución de Documentos le indica al sistema la opción de “Recepcionar Expedientes” que van a ser procesados en el Centro de Digitalización. Los expedientes pueden ser revisados antes de confirmar su recepción, en caso de error en sus datos registrados estos deben ser rectificadas. El caso de uso termina cuando el actor presiona el botón aceptar y da entrada a los expedientes recepcionados hasta el momento.

Dar entrada de expediente: El caso de uso inicia cuando es invocado por el caso de uso Recepcionar Expedientes. El sistema da entrada a los expedientes que han sido recepcionados, iniciando un trámite para cada uno de ellos que los identifican durante todo el proceso. Para cada expediente que entra al Centro de Digitalización se inicia un trámite que no cierra hasta que se concrete la devolución del mismo.

Devolver expediente: El caso de uso consiste en devolver los expedientes que han sido digitalizados. El actor registrará el código de los expedientes que quiere devolver y se actualiza el listado de entrega

respectivamente. El caso de uso tiene inicio cuando el Funcionario de Recepción y Devolución de Documentos le indica al sistema la opción “Devolver Expedientes” para devolver los expedientes procesados en el Centro de Digitalización al Archivo Documental.

Control de calidad

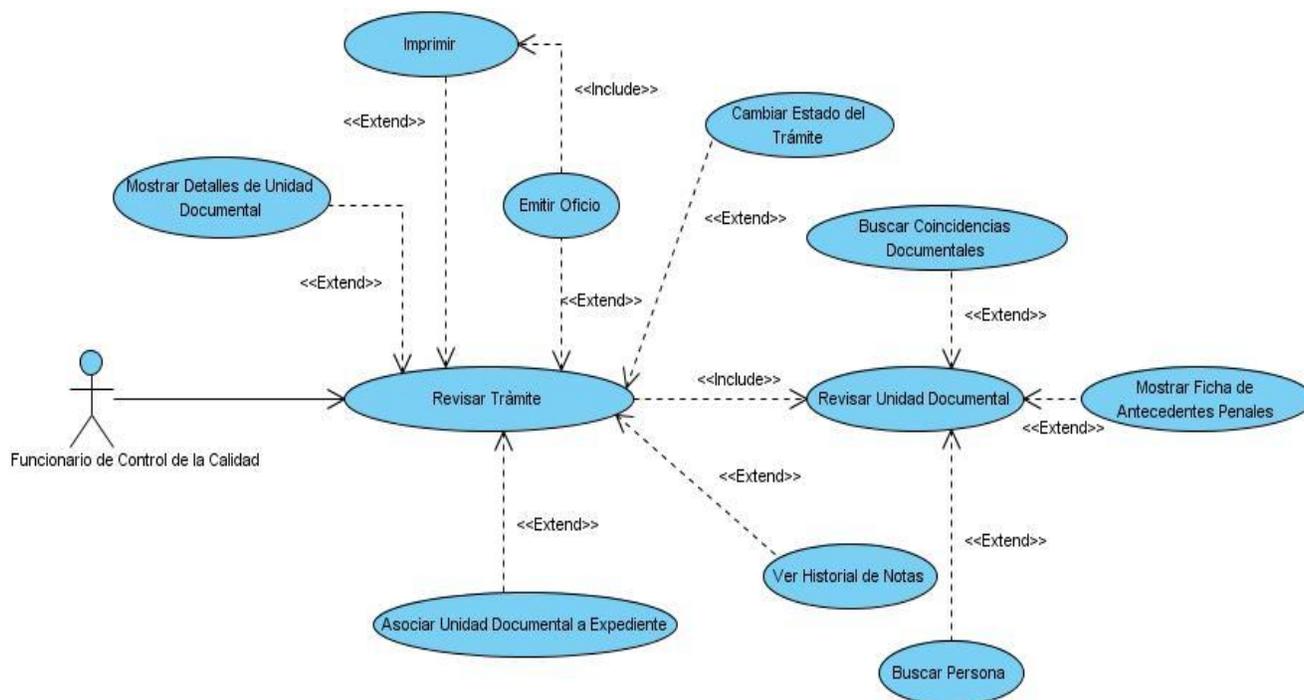


Fig. 4 Diagrama de caso de uso del proceso Control de Calidad

Revisar trámite: El caso de uso inicia cuando el Funcionario de Control de Calidad le indica al sistema la opción “Revisar trámite” que se han realizado en el Centro de Digitalización. Registra el código del expediente físico que tiene y se muestran los datos asociados al mismo. Se pueden reubicar los trámites al área de Digitalización de Documentos o de Asociación de Metadatos si existe algún error en el proceso. Si el trámite ha sido realizado correctamente se actualiza su estado y se imprimen las salidas del mismo para proceder al área de Otorgamiento. Invoca otros casos de uso como se muestra en la figura 4 para completar sus funcionalidades, los mismos pertenecen a otros módulos del proyecto. El caso de uso termina cuando el actor emite una respuesta del proceso de revisión.

Revisar Unidad documental: El caso de uso inicia cuando es invocado por el caso de uso **Revisar Trámite**. El Funcionario de Control de Calidad revisa el proceso de asociación de metadatos de las

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

unidades documental es el cual garantiza la revisión de cada una de las unidades documentales como continuación del proceso de Control de la Calidad. El sistema muestra los datos asociados en dependencia del tipo documental.

Otorgamiento

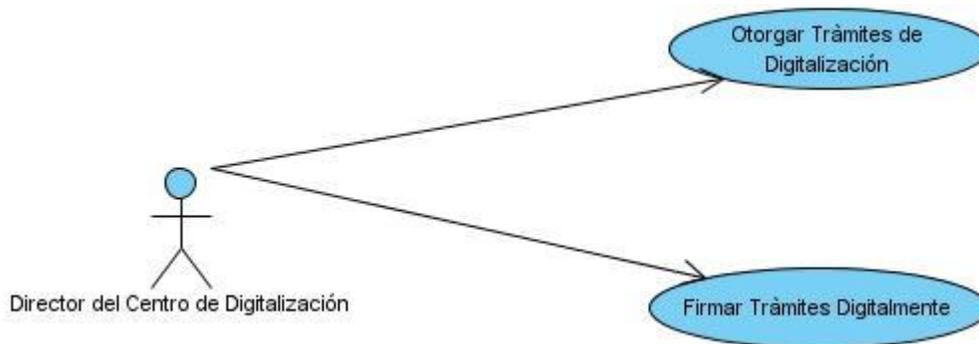


Fig. 5 Diagrama de caso de uso del proceso Otorgamiento

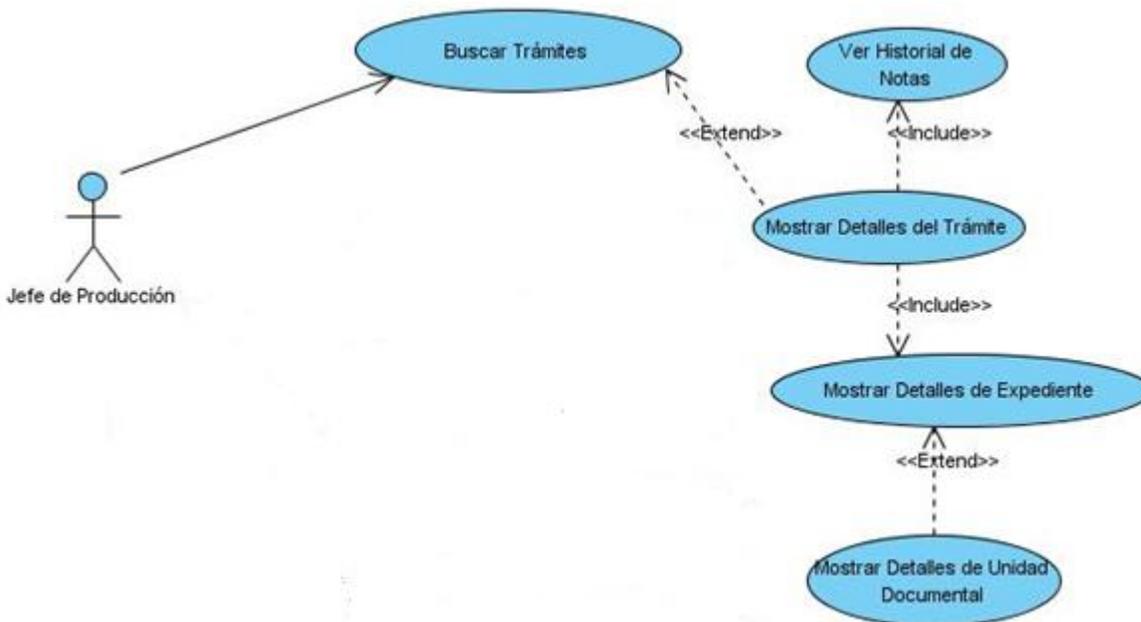


Fig. 6 Diagrama de caso de uso del proceso Otorgamiento

Otorgar Trámite: El caso de uso inicia cuando el actor Director de Centro de Digitalización le indica al sistema la opción “Otorgar trámites”. El sistema muestra los trámites que han sido revisados y están listos para otorgar. El actor puede filtrar la búsqueda de trámites para otorgar solo aquellos que él desea según un criterio de búsqueda determinado. Luego de ser devueltos cambia el estado del trámite y se invoca al caso de uso **Firmar Trámites Digitalmente**, además se pueden seleccionar un trámite y revisarlo, mediante la invocación del caso de uso revisar trámite.

Buscar Trámite: El caso de uso inicia cuando el actor Jefe de Producción le indica al sistema la opción “Buscar trámites” del menú de opciones para localizar un trámite que se encuentra en el Centro de Digitalización. El actor introduce los criterios para realizar la búsqueda de los trámites. El sistema le muestra los trámites que coinciden con el criterio de búsqueda o emite un mensaje si esta no produce resultados. Invoca otros casos de uso como se muestra en la figura 4 para completar sus funcionalidades, los mismos pertenecen a otros módulos del proyecto.

Firmar Trámites Digitalmente: El caso de uso inicia cuando el actor Director de Centro de Digitalización le indica al sistema la opción “Firmar Trámites Digitalmente”. El sistema muestra los trámites que han sido revisados y están listos para firmar. El actor puede filtrar la búsqueda de trámites para firmar solo aquellos que él desea según un criterio de búsqueda determinado. Luego de ser devueltos el trámite tiene un valor legal.

2.2 Arquitectura

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 simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), 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. (24)

La arquitectura del sistema tiene como principales características:

- Cliente/Servidor
- Distribuida en n capas
- Basada en el desarrollo de componentes.

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 afectar 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 (digidap) se instala en las máquinas clientes e interactúa con la otra desplegada en el servidor (digidap-ejb) que 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. (25)

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 Fachada Gestores Remotos que sirve de fachada para la comunicación entre los gestores clientes y servidor, es donde se procesan los datos del cliente.

Existe además la capa de negocio en el servidor que contiene los gestores de negocio 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, 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.

2.3 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. (24)

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:

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 del 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 de datos utilizadas.

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 todos 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.

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.

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.

Alta cohesión para brindar una alta cohesión funcional cuando los elementos de un componente (clase, por ejemplo) "colaboran para producir algún comportamiento bien definido".

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 utilizó en los gestores, cliente, servidor y acceso a datos.

2.4 Paquetes del subsistema DigiDAP

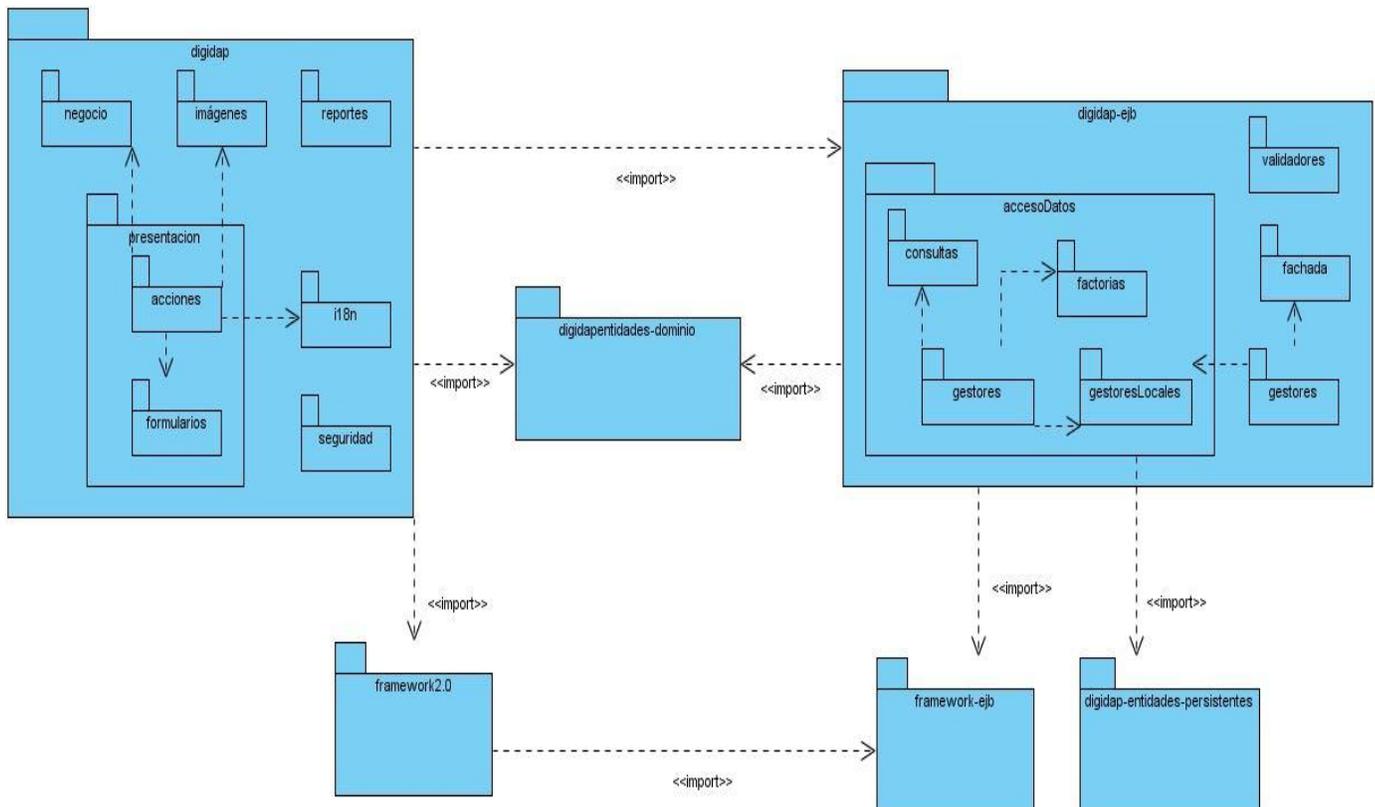


Fig. 7 Paquetes del subsistema DigiDAP

digidap

i18n: Permite diseñar el software de manera que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código.

presentación: Contiene las acciones y formularios.

negocio: Gestores de negocio en el cliente que invocan a los gestores de negocio en el servidor.

imágenes: Contiene los iconos e imágenes utilizados en la aplicación.

digidap-ejb:

acceso a datos: Almacena todos los paquetes que contienen las clases de la capa de acceso a datos.

consultas: Contiene los ficheros de consulta .xml utilizadas para el acceso a 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: Interfaces que implementan los gestores de acceso a datos.

gestores: Gestores de acceso a datos que implementan la gestión y manipulación de los datos.

fachada: Interfaces que implementan los gestores de negocio publicados en el servidor.

gestores: Gestores de negocio que implementan la lógica de negocio en el servidor.

digidap-entidades persistentes: Contiene las clases entidades para el acceso a datos del servidor.

digidap-entidades dominio: Contiene las clases entidades de dominio de la aplicación.

2.5 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. (26)

El modelo de diseño proporciona detalles acerca de la estructura de datos, las arquitecturas, las interfaces y los componentes de software que son necesarios para implementar el sistema. (24)

Como parte del modelo de diseño se realizaron los diagramas de clases y descripciones por caso de uso. Se pueden consultar en el [anexo 1](#), a continuación se muestra el caso de uso Recepcionar expediente que ilustra un ejemplo.

2.5.1 DCD- Recepcionar expediente

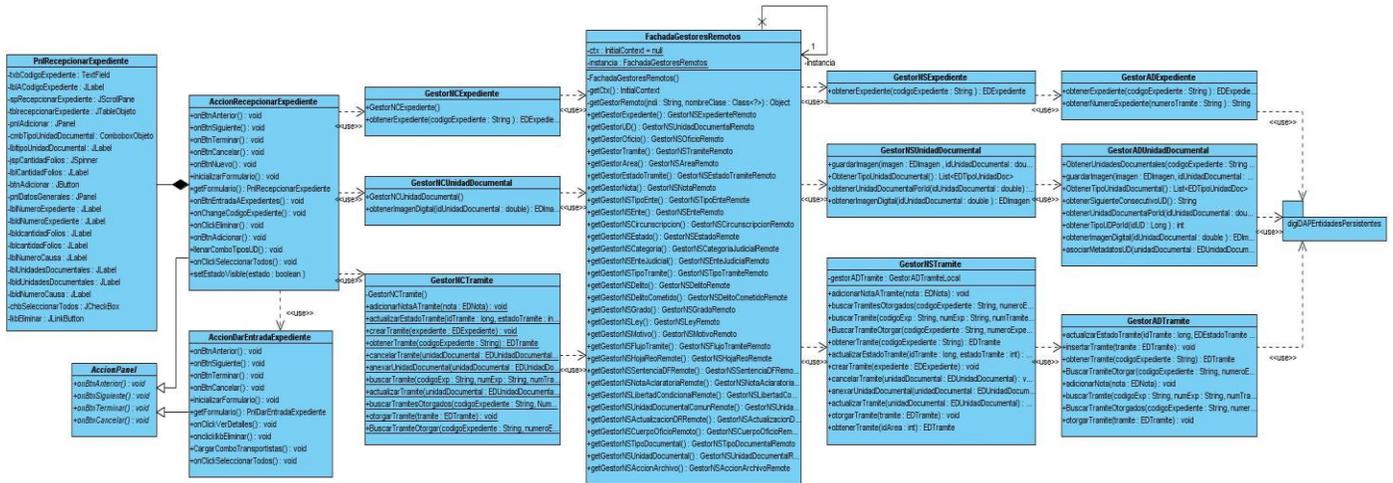


Fig. 8 Diagrama de clases del diseño CU Recepcionar Expediente

2.5.2 Descripción de clases

Nombre de la clase: PnlRecepcionarExpediente	
Tipo de clase: Interfaz	
Descripción: Clase que representa el formulario <i>Recepcionar Expediente</i> en el que se introducen los datos correspondientes a un expediente para su recepción en el centro.	
Atributos	Tipo
btnAdicionar	JButton
btnEntrada	JButton
cmbTipoUnidadDocumental	ComboBoxObjeto
JScrollPane1	JScrollPane
JtableRecepcionarExpediente	JTable
lblCodigoExpediente	JLabel
lblCantidadFolios	JLabel
lblNumeroExpediente	JLabel
lblTipoExpediente	JLabel
lblTipoUnidadDocumental	JLabel

txbCodigoExpediente	TextField
txbCantidadFolios	TextField

Tabla 1 Descripción de la clase PnlRecepcionarExpediente.

Nombre de la clase: AccionRecepcionarExpediente	
Tipo de clase: Controladora	
Descripción: Clase acción que controla las funcionalidades del formulario Recepcionar Expediente para dar entrada a un expediente.	
Atributos	Tipo
expedientesRecepcionados	List<EExpediente>
Responsabilidades de la clase:	
Nombre:	onBtnEntradaAExpediente()
Descripción:	Da entrada a los expediente a la base de datos
Nombre:	onChangeCodigoExpediente()
Descripción:	Muestra todos los datos del expediente asociados a un código de expediente.
Nombre:	onClickEliminar()
Descripción:	Elimina la unidad documental seleccionada del expediente.
Nombre:	onBtnAdicionar()
Descripción:	Adiciona una unidad documental al expediente mostrado.
Nombre:	llenarCombo()
Descripción:	Adiciona al combo las opciones a seleccionar.

Tabla 2 Descripción de la clase AccionRecepcionarExpediente

Nombre de la clase: GestorNCExpediente	
Tipo de clase: Controladora	
Descripción: Clase que gestiona la Lógica de Negocio del Cliente para la entidad Expediente. Realiza las llamadas correspondientes a la Lógica de Negocio del Servidor para completar las funcionalidades.	
Responsabilidades de la clase:	
Nombre:	adicionarUD()

Descripción:	Invoca al método adicionarUD del gestor expediente del servidor
Nombre:	eliminarUD()
Descripción:	Invoca el método eliminarUD del gestor expediente del servidor
Nombre:	obtenerExpediente
Descripción:	Invoca el método obtenerExpediente del gestor expediente del servidor

Tabla 3 Descripción de la clase GestorNCExpediente

Nombre de la clase: GestorNCUnidadDocumental	
Tipo de clase: Controladora	
Descripción: Clase que gestiona la Lógica de Negocio del Cliente para la entidad Unidad Documental. Realiza las llamadas correspondientes a la Lógica de Negocio del Servidor para completar las funcionalidades.	
Responsabilidades de la clase:	
Nombre:	obtenerDatosGeneralesUD
Descripción:	Invoca al método obtenerDatosGeneralesUD del gestor servidor unidad documental

Tabla 4 Descripción de la clase GestorNCUnidadDocumental

Nombre de la clase: GestorNSEpediente	
Tipo de clase: Controladora	
Descripción: Clase que se utiliza para gestionar la lógica de negocio del servidor para la entidad Expediente.	
Responsabilidades de la clase:	
Nombre:	adicionarUD()
Descripción:	Invoca al método adicionarUD del gestor acceso a datos
Nombre:	eliminarUD()
Descripción:	Invoca el método eliminarUD del gestor acceso a datos
Nombre:	obtenerExpediente()
Descripción:	Invoca el método obtenerExpediente del gestor acceso a datos

Tabla 5 Descripción de la clase GestorNSEpediente

Nombre de la clase: GestorNSUnidadDocumental	
Tipo de clase:	
Descripción: Clase que se utiliza para gestionar los métodos asociados al GestorNSUnidadDocumental.	
Responsabilidades de la clase:	
Nombre:	obtenerDatosGeneralesUD
Descripción:	Invoca al método obtenerDatosGeneralesUD del gestor acceso a datos unidad documental.

Tabla 6 Descripción de la clase GestorNSUnidadDocumental

Nombre de la clase: GestorADEXpediente	
Tipo de clase: Controladora	
Descripción: Clase que se utiliza para gestionar la lógica de acceso a datos para la entidad Expediente.	
Responsabilidades de la clase:	
Nombre:	adicionarUD()
Descripción:	Adiciona una unidad documental al expediente
Nombre:	eliminarUD()
Descripción:	Elimina una unidad documental del expediente
Nombre:	obtenerExpediente()
Descripción:	Obtiene todos los datos del expediente asociado a un código de expediente

Tabla 7 Descripción de la clase GestorADEXpediente

Nombre de la clase: GestorADUnidadDocumental	
Tipo de clase:	
Descripción: Clase que se utiliza para implementar los métodos asociados al GestorADUnidadDocumental	

Responsabilidades de la clase:	
Nombre:	obtenerDatosGeneralesUD
Descripción:	Muestra todos los datos asociados a la unidad documental.

Tabla 8 Descripción de la clase GestorADUnidadDocumental

2.5.3 Estándares del diseño

En el expediente de proyecto se encuentra el documento Pautas del diseño donde se explican normas y estándares utilizadas y definidas por el equipo de desarrollo para realizar los diagramas de clases. A continuación se muestra algunos aspectos importantes que se aplicaron para el diseño de clases.

En los casos que los nomencladores sean compuestos por varias palabras se usará una notación de Camello. La notación Camello consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula: EndOfFile. Se llama notación “Camello” porque los identificadores recuerdan las jorobas de un camello. Existen dos variantes:

UpperCamelCase: en esta variante la primera letra también es mayúscula.

lowerCamelCase: la primera letra es minúscula.

Se utilizará la primera variante, aplicando la técnica verbo-sustantivo.

Camello se utilizará en los nombres de las Entidades persistentes, Entidades de Dominio, Propiedades y Funcionalidades del sistema.

Las clases persistentes comenzarán con las siglas definidas en la base de datos para cada tabla.

Los formularios de tipo Dialog comienzan con las siglas Frm, y los de tipo Panel con las siglas Pnl.

Los gestores de negocio del cliente comienzan con las siglas GestorNC.

Las clases de la lógica de negocio del servidor comienzan con las siglas GestorNS.

Las clases de la lógica de negocio de acceso a datos comienzan con las siglas GestorAD.

Las interfaces publicadas en el servidor terminan con el prefijo Remoto.

Las clases interceptoras de validación terminan con el prefijo Interceptor.

Las relaciones entre clases usarán el estereotipo <<use>>.

Nomenclatura para los componentes de formularios:

Los botones (JButton) comienzan con las siglas btn.

Los campos de texto (JTextField) comienzan con las siglas txb.

Los campos fecha (Date) comienzan con las siglas fch.

Para el componente JDateChooser ----- dtc

Las cajas de texto (JComboBox) comienzan con las siglas cmb.

Las áreas de texto (JTextArea) comienzan con las siglas txa.

Los labels (JLabel) comienzan con las siglas lbl.

Los cuadros de chequeo (JCheckBox) comienzan con las siglas chb.

Las casillas de selección (JRadioButton) comienzan con las siglas rbt.

Las tablas (JTable) comienzan con las siglas tbl.

2.6 Diseño de la base de datos

El objetivo principal de realizar el diseño de la base de datos es generar las tablas que modelan los registros en los que se guarda la información del sistema.

2.6.1 Modelo de datos

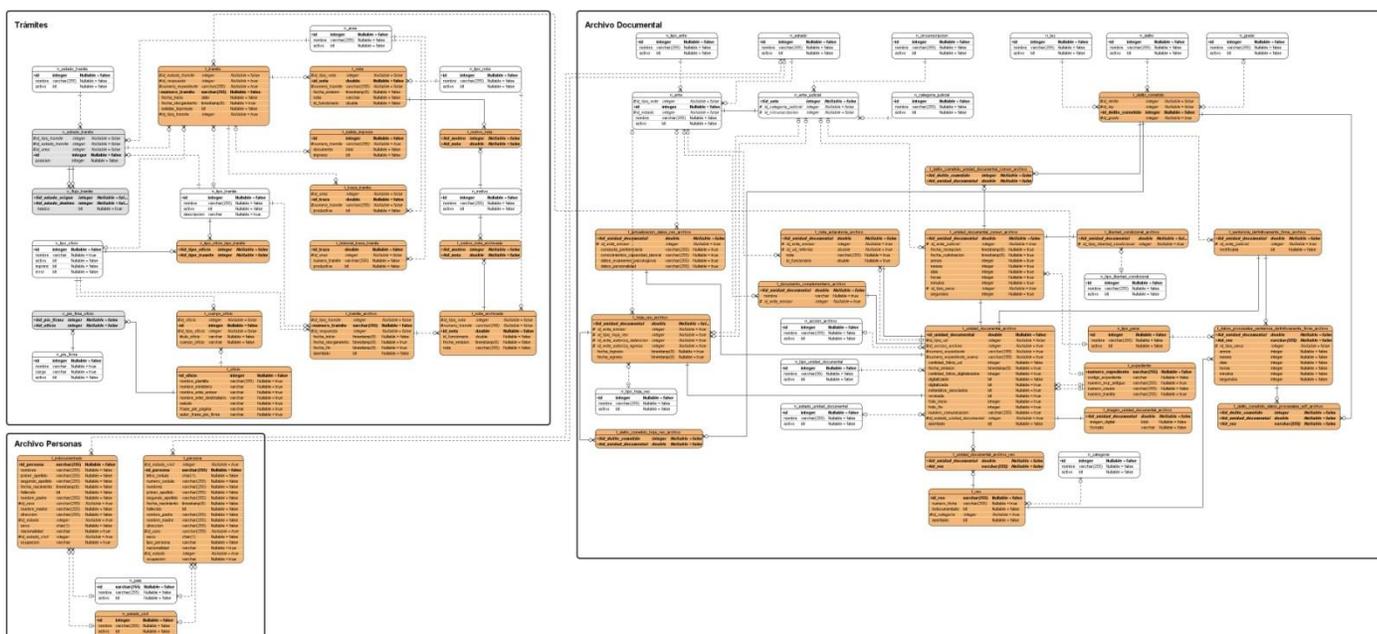


Fig. 9 Modelo de datos

2.6.2 Descripción de las tablas

Tablas de configuración

c_estado_tramite: Tabla que almacena la combinación de Tipo de trámite, estado del trámite y Área del trámite. Es utilizada para establecer el estado en que se encuentra un trámite determinado.

c_flujo_tramite: Tabla que almacena los flujos de estados por los que pasa el trámite.

c_pie_firma_oficio: Tabla que almacena los pie de firma para los oficio.

c_tipo_oficio_tipo_tramite: Tabla que almacena el tipo de oficio por tipo de trámite.

Nomencladores

n_accion_archivo: Nomenclador que almacena los tipos de acciones que pueden realizarse con una unidad documental en el archivo (crear, eliminar, anexar).

n_area: Nomenclador que almacena las diferentes Áreas por la que puede pasar un trámite.

n_categoria_judicial: Nomenclador que almacena la clasificación del ente judicial según la categoría.

n_circunscripcion: Nomenclador que almacena las circunscripciones.

n_ciudad: Nomenclador que almacena la ciudad.

n_delito: Nomenclador que almacena delito.

n_ente: Nomenclador que almacena los entes que pueden emitir unidades documentales y enviarlas a la división.

n_ente_judicial: Nomenclador que almacena los entes judiciales que pueden emitir unidades documentales y enviarlas a la división.

n_estado: Nomenclador que almacena los estados de Venezuela.

n_estado_civil: Nomenclador que almacena el estado civil posible definido en el sistema de cada persona.

n_estado_tramite: Nomenclador que almacena los diferentes estados por los que puedes pasar un trámite.

n_grado: Nomenclador que especifica el grado del delito cometido.

n_ley: Nomenclador que almacena todas las leyes.

n_motivo: Nomenclador que almacena los diferentes motivos por los que puede ser asociada una nota a un trámite.

n_ocupacion: Nomenclador que almacena la ocupación de la persona.

n_pais: Nomenclador que almacena los países de acuerdo a las necesidades del sistema.

n_pie_firma: Nomenclador que almacena los datos de quien firma el oficio.

n_respuesta: Nomenclador que almacena los tipos de respuestas que puede tener el trámite. Para los trámites con errores en datos se emite un oficio, los que concluyen satisfactoriamente se genera una ficha de antecedentes penales, los que cuyas unidades documentales están duplicadas se cancela y las que

están asociadas a un expediente se Anexa. El funcionario de control de calidad selecciona el tipo de respuesta que se le dará al trámite para que el Director del Centro de Digitalización pueda verla en caso que quiera revisar el trámite nuevamente.

n_tipo_ente: Nomenclador que almacena todos los entes de acuerdo a su tipo.

n_tipo_hoja_reo: Nomenclador que almacena los tipos de hoja de reo.

n_tipo_libertad_condicional: Nomenclador que almacena los tipos de Libertad condicional.

n_tipo_nota: Nomenclador que describe los tipos de notas que pueden ser asociados a un trámite. Nota de Preparación, de cambio de Estado y de Respuesta de un trámite.

n_tipo_oficio: Nomenclador que almacena los tipos de oficio.

n_tipo_pena: Nomenclador que almacena los tipos de pena.

n_tipo_tramite: Nomenclador que almacena los diferentes tipos de trámite.

n_tipo_unidad_documental: Nomenclador que almacena los diferentes tipos de unidades documentales.

Tablas

t_actualizacion_datos_reo_archivo: Tabla que almacena la unidad documental de tipo actualización de datos de reo.

t_cuerpo_oficio: Tabla que almacena el contenido de los oficios que se emiten.

t_datos_procesales_sentencia_definitivamente_firme_archivo: Tabla que almacena los metadatos de la unidad documental sentencia definitivamente firme.

t_delito_cometido: Tabla que almacena los delitos cometidos.

t_delito_cometido_datos_procesales_sdf_archivo: Tabla que almacena los delitos cometidos del reo en la unidad documental.

t_delito_cometido_hoja_reo_archivo: Tabla que almacena los delitos cometidos que pertenecen a la unidad documental hoja de reo.

t_delito_cometido_unidad_documental_comun_archivo: Tabla que almacena todos los delitos cometidos de acuerdo a la unidad documental a la que están asociados, especifica que cada unidad documental puede contener varios delito cometidos, a su vez que un delito cometido puede encontrarse en varias unidades documentales.

t_expediente: Tabla que almacena los datos de los expedientes.

t_historial_traza_tramite: Tabla que almacena las acciones realizadas sobre un trámite que pasan de una fecha dada.

t_hoja_reo_archivo: Tabla que almacena la unidad documental de tipo Hoja de Reo.

t_indocumentado: Tabla que almacena los datos de una persona.

t_libertad_condicional_archivo: Tabla que almacena los datos procesales de la libertad condicional asociados a una unidad documental.

t_libertad_condicional: Tabla que almacena los datos procesales de la libertad condicional asociados a una unidad documental.

t_motivo_nota: Tabla que almacena los motivos de la nota asociada al trámite.

t_motivo_nota_archivada: Tabla que almacena las notas emitidas en el archivo.

t_nota: Tabla que almacena notas que pueden ser asociadas al trámite.

t_nota_aclaratoria_archivo: Tabla que almacena la unidad documental de Nota Aclaratoria.

t_nota_archivada: Tabla que almacena las notas asociadas a los trámites archivados

t_oficio: Tabla que almacena los oficios emitidos en archivo.

t_persona: Tabla que almacena los datos asociados a la persona.

t_reo: Tabla que almacena los Identificadores de las personas que son reos.

t_salida_impresa: Tabla que almacena la respuesta que se imprime y se entrega a los usuarios.

t_sentencia_definitivamente_firme_archivo: Tabla que almacena la unidad documental sentencia definitivamente firme y la rectificada.

t_tramite: Tabla que almacena los trámites que se realizan en el sistema.

t_tramite_archivado: Tabla que almacena los trámites que se realizan en el sistema.

t_traza_tramite: Tabla que almacena las acciones realizadas sobre un trámite.

t_unidad_documental_archivo: Tabla que almacena las unidades documentales del sistema.

t_unidad_documental_archivo_reo: Tabla que almacena los reos en las diferentes unidades documentales.

t_unidad_documental_comun_archivo: Tabla que almacena las unidades documentales Suspensión Condicional de la Pena, Suspensión Condicional de la Ejecución de la Pena, Destacamento de Trabajo, (Conmutación de la Pena o Libertad Condicional), Sometimiento a Juicio.

Conclusiones

En este capítulo se obtuvieron los diagramas de clases del diseño y sus descripciones como artefactos de este flujo, en los cuales se aplicaron la arquitectura, los patrones, normas y estándares del diseño

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

definidos por el equipo de desarrollo. Se mostró el diagrama de paquetes de DigiDAP y se obtuvo el diseño de la base de datos con sus clases persistentes.

CAPÍTULO 3: Implementación y validación de la solución propuesta

Introducción

En el presente capítulo se muestran los diagramas de componentes por capas: Presentación, Negocio, Acceso a Datos. Se exponen las normas y estándares de codificación y de diseño de interfaz utilizadas y definidas por el equipo de desarrollo. Se evalúa el grado de calidad y fiabilidad de los datos de los resultados obtenidos para el diseño con métricas orientadas a objetos y la implementación con pruebas de caja negra.

3.1 Diagrama de componentes del sistema.

Un diagrama de componentes representa como esta 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.

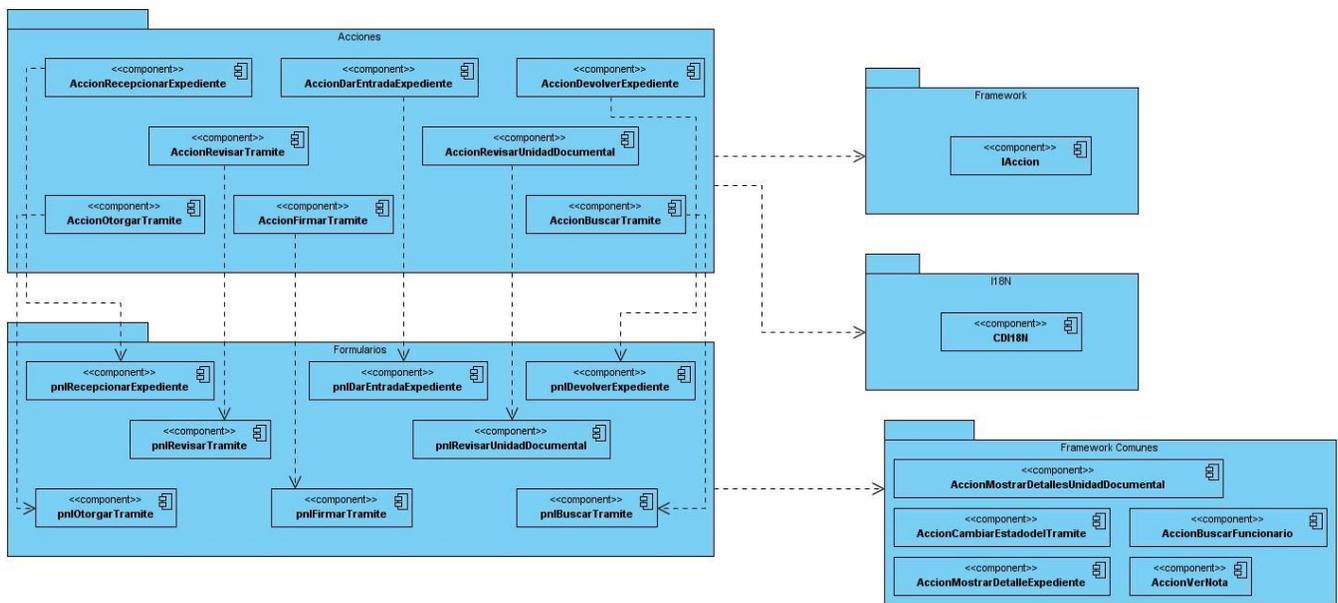


Fig. 10 Diagrama de componentes Capa de Presentación

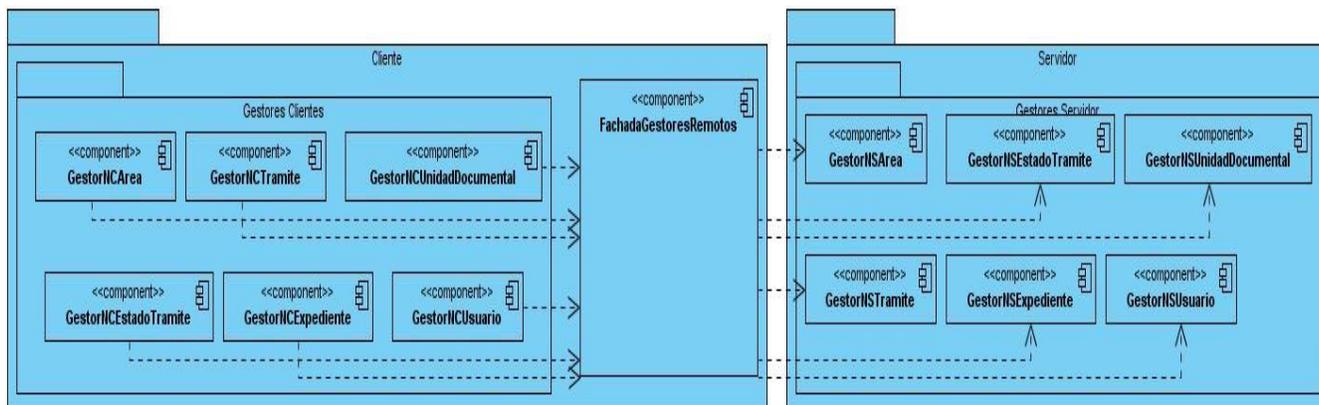


Fig. 11 Diagrama de Componentes Capa de Negocio

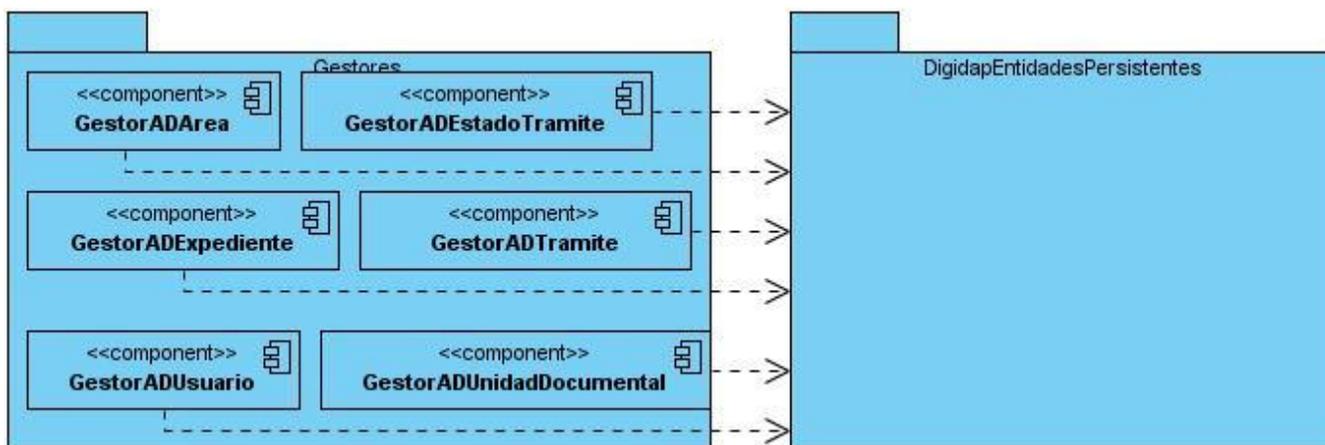


Fig. 12 Diagrama de Componentes Capa de Acceso a Datos

3.2 Arquitectura de información

En el expediente de proyecto se encuentra el documento Pautas del diseño de interfaces donde se explican normas y estándares utilizadas y definidas por el equipo de desarrollo para diseñar los formularios de la aplicación. A continuación se muestra algunos aspectos recogidos en el documento.

En un formulario, los datos asociados a un mismo criterio se agruparán utilizando paneles con título. El título del panel se escribe en formato de oración, en negrita y sin dos puntos al final.

Cuando el formulario tiene varios componentes. El nombre de las etiquetas se escribe encima de los componentes. El texto en las etiquetas se escribe en formato de oración, sin utilizar negrita y terminado en dos puntos. Los componentes irán debajo, a una distancia de 8 píxeles, cada componente de entrada de

datos debe tener una altura de 23 píxeles. La separación horizontal entre componentes debe ser de 20 píxeles y vertical de 15 píxeles. Todo alineado a la izquierda, y en caso de que aparezcan unos debajo de los otros se debe tratar de que todos los componentes tengan un mismo tamaño (el del mayor componente).

Código de expediente:

Datos generales

Número de Expediente: 57	Cantidad de folios: 7
Número de causa: 8c-11635-09	Unidades documentales: 2

Adicionar unidades documentales

Unidad documental: Cantidad de folios:

Seleccionar todos

Tipo de Unidad Documental	Digitalizar	Cantidad de Folios
<input type="checkbox"/> Sentencia Definitivamente Firme	<input checked="" type="checkbox"/>	6
<input type="checkbox"/> Sometimiento a Juicio	<input type="checkbox"/>	1

Fig. 13 Formulario para Recepcionar Expedientes.

3.3 Estándares de codificación

En el expediente de proyecto se encuentra el documento que contiene los estándares de codificación utilizados y definidos por el equipo de desarrollo para la implementación. A continuación se muestra algunos aspectos recogidos en el documento.

El uso de comentarios en el código permitirá lograr un mayor entendimiento del mismo en tiempo de mantenimiento y de consulta por otros desarrolladores. Estos comentarios deberán redactarse en español

y al principio de cada clase y de cada método. Un comentario podría consistir en una breve introducción que explicará por qué existe y qué hace.

Ejemplo 1:

```
/**
 * @see Método que permite eliminar unidades documentales del grid y de la lista del expediente actual.
 */
private void onClickEliminar() {
    if (this.getFormulario().tblrepcionarExpediente.getObjetosChecked().isEmpty()) {
        GestorMensajes.MensajeAviso(CDI18n.getInstancia().getMensaje("AccionRepcionarExpediente.mensajeSeleccionarObjetoAEliminar"));
    } else {
        if (GestorMensajes.MensajeConfirmar(CDI18n.getInstancia().getMensaje("AccionRepcionarExpediente.mensajeConfirmarEliminarObjeto")) == 1) {
            try {
                this.getFormulario().tblrepcionarExpediente.eliminarObjetosChecked();
                this.getFormulario().lblUnidadesDocumentales.setText(String.valueOf(this.expedienteActual.getUnidadesDocumentales().size()));
                this.getFormulario().lblCantidadFolios.setText(String.valueOf(this.expedienteActual.obtenerCantidadFolios()));
            } catch (Exception ex) {
                GestorExcepcion.reportarExcepcion(ex);
            }
        }
    }
}
```

Fig. 14 Segmento de código que ejemplifica el uso de comentarios

Por otro lado no se debe dejar espacios entre los nomencladores de los métodos y los paréntesis que encapsulan los parámetros de dicho método, ya sean globales o instantáneos. De igual forma no se darán espacios entre el paréntesis "(" y el primer parámetro y entre el último parámetro y el ")".

Cuando una expresión no entre en una línea, romperla de acuerdo con estos principios:

- Romper después de una coma.
- Romper antes de un operador.
- Preferir roturas de alto nivel (más a la derecha que el "padre") que de bajo nivel (más a la izquierda que el "padre").
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

Si las reglas anteriores llevan a código confuso o a código que se aglutina en el margen derecho, indentar justo 8 espacios en su lugar.

Ejemplo 2:

```

*
* @param codigoExpediente
*/
public static EDTramite obtenerTramite(String codigoExpediente) throws Exception {
    EDTramite tramite = FachadaGestoresRemotos.getInstancia().getGestorNSTramite().obtenerTramite(codigoExpediente);
    return tramite;
}

/**
 *
 * @see Devuelve true si existe algún trámite asociado al código del
 * expediente y false en caso contrario.
 * @param codigoExpediente Código del expediente para el cual se busca si
 * existe un trámite asociado.
 */
public static boolean existeTramite(String codigoExpediente)
    throws Exception {
    return
        FachadaGestoresRemotos.getInstancia().getGestorNSTramite()
            .existeTramite(codigoExpediente);
}

```

Fig. 15 Segmento de código que ejemplifica el tamaño y espacio

3.4 Validación de la solución propuesta

3.4.1 Métricas para el modelo de diseño

Para validar el diseño se utilizaron métricas orientadas a objetos propuestas por Lorenz y Kidd.

En su libro sobre métricas orientadas a objetos, Lorenz y Kidd [LOR94] dividen las métricas basadas en clases en 4 amplias categorías, cada una con un diseño al nivel de componentes: tamaño, herencia, valores internos y valores externos. Las métricas orientadas al tamaño aplicadas a una clase de diseño orientado a objetos se concentran en el conteo de atributos y operaciones de una clase individual, así como en los valores promedios para el sistema orientado a objetos como un todo, Las métricas basadas en la herencia se concentran en la manera en que las operaciones se reutilizan en la jerarquía de clases.

(24)

Métrica Tamaño de Clase (TC)

El tamaño general de una clase se determina con las siguientes medidas:

El número total de operaciones (de instancia heredada y privada) que están encapsuladas dentro de una clase. El número de atributos (de instancia heredada y privada) que están encapsulados por la clase. Deben calcularse promedios para el número de atributos y operaciones de la clase.

Los valores grandes de TC indican que tal vez una clase tenga demasiada responsabilidad. Cuando menores sean los valores promedio para el TC más probable será que las clases dentro del sistema puedan reutilizarse ampliamente.

Para evaluar las métricas es necesario utilizar los umbrales, se clasifican en 3 grupos según su tamaño. Se muestran en la tabla que aparece a continuación.

Tamaño de clase	Umbral
Pequeña	≤ 20
Media	>20 y ≤ 30
Grande	>30

Tabla 9 Tamaño de clase

En la siguiente tabla se muestran la cantidad total de clases del diseño y el promedio de las operaciones y atributos de las mismas.

Total de clases	Operaciones	Atributos
54	5.3	2.51

Tabla 10 Resultados de la métrica

Cantidad de clases	Umbral	Tamaño
47	≤ 20	Pequeña
6	>20 y ≤ 30	Media
1	>30	Grande

Tabla 11 Cantidad de clases por clasificación

Los resultados obtenidos son positivos según esta métrica pues casi todas las clases que conforman el diseño que será la entrada del flujo de trabajo de implementación están dentro del grupo Pequeña.

Métrica Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

Atributo de Calidad.	Modo en que lo afecta.
Responsabilidad.	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de Implementación.	Un aumento del TOC implica un aumento en la complejidad de implementación de la clase.
Reutilización.	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Atributo	Categoría	Criterio
Responsabilidad (Rp).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
Complejidad de Implementación (CI).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
Reutilización (Rt).	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio

Resultados del instrumento de evaluación de la métrica Tamaño Operacional de clase (TOC):

Total de clases	Promedio de procedimientos
54	5.3

Tabla 12 Cantidad de clases y promedio de procedimientos

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

Criterio	Cantidad de clases	Promedio
Entre 1 y 5 procedimientos	33	62.26415094
Entre 6 y 10 procedimientos	10	18.86792453
Entre 11 y 15 procedimientos	8	15.09433962
Más de 15 procedimientos	2	3.773584906
Total	53	100

Tabla 13 Cantidad de procedimientos por clases

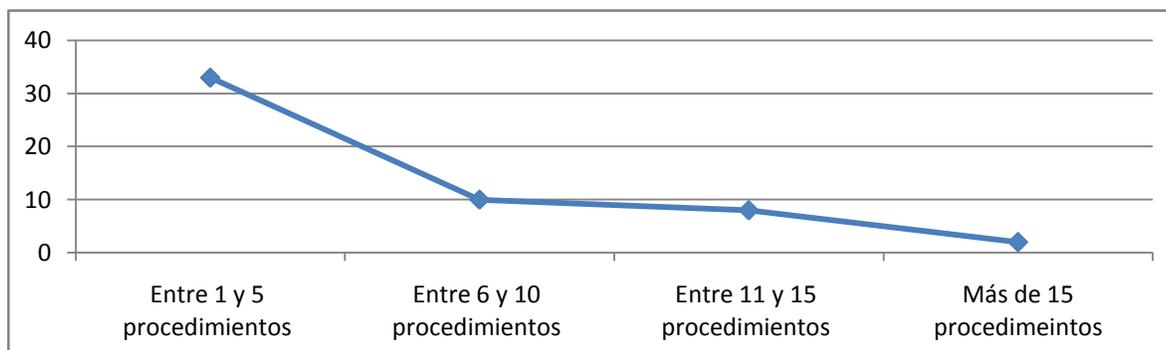


Fig. 16 Gráfica de los resultados de la evaluación de la métrica TOC agrupados por las tendencias de los Valores

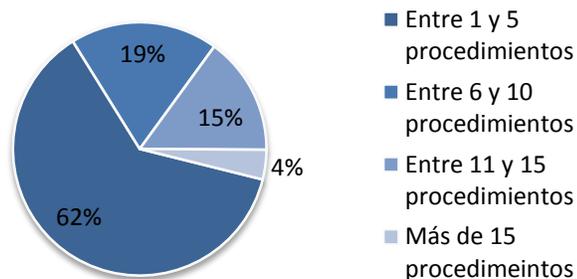


Fig. 17 Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos

Representación de la incidencia de los resultados de la evaluación de la métrica TOC en cada uno de los atributos:

Responsabilidad	Cantidad de clases	Promedio
Baja	33	61.11111111
Media	10	18.51851852
Alta	10	18.51851852

Tabla 14 Atributo Responsabilidad

Responsabilidad

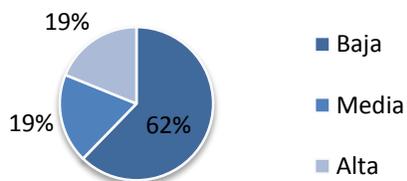


Fig. 18 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad

Complejidad	Cantidad de clases	Promedio
Baja	33	61.11111111
Media	10	18.51851852
Alta	10	18.51851852

Tabla 15 Atributo Complejidad

Complejidad

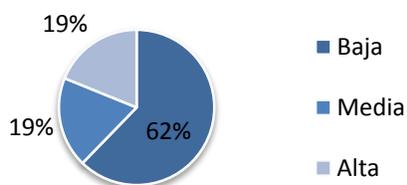


Fig. 19 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad

Reutilización	Cantidad de clases	Promedio
Alta	33	61.11111111
Media	10	18.51851852
Baja	10	18.51851852

Tabla 16 Atributo Reutilización

Reutilización

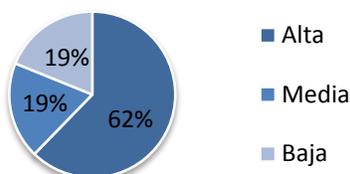


Fig. 20 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización

Se puede concluir que el diseño propuesto está entre los límites aceptables de calidad, teniendo en cuenta que la mayoría de las clases (62%) posee menos cantidad de operaciones que la media registrada en las mediciones. Los atributos de calidad se encuentran en un nivel medio satisfactorio en el 62% de las clases; de manera que se puede observar cómo se fomenta la Reutilización (elemento clave en el proceso de desarrollo de software) y cómo están reducidas en menor grado la Responsabilidad y la Complejidad de implementación.

Métrica relaciones entre clases (RC)

Atributo de Calidad.	Modo en que lo afecta.
Responsabilidad.	Un aumento del RC implica un aumento de la responsabilidad asignada a la clase.
Complejidad del mantenimiento.	Un aumento del RC implica un aumento en la complejidad del mantenimiento de la clase.
Reutilización.	Un aumento del RC implica una disminución del grado de reutilización de la clase.

Cantidad de pruebas.	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
-----------------------------	--

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Atributo.	Categoría.	Criterio.
Acoplamiento.	Ninguno.	0
	Bajo.	1
	Medio.	2
	Alto.	>2
Complejidad de mantenimiento.	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$>2 \times$ Promedio
Reutilización.	Baja.	$>2 \times$ Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	\leq Promedio
Cantidad de pruebas.	Baja.	\leq Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$>2 \times$ Promedio

Resultados del instrumento de evaluación de la métrica Relaciones entre clases (RC):

Total de clases	Promedio de asociaciones de uso
------------------------	--

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

54	1.72
-----------	------

Tabla 17 Cantidad de clases y promedio de asociaciones de uso

Criterio	Categoría	Cantidad de clases	Promedio
0 dependencias	Muy Bueno	23	46
1 dependencias	Bueno	22	44
2 dependencias	Regular	2	4
3 dependencias	Malo	1	2
> 3 dependencias	Muy Malo	5	10
Total		53	106

Tabla 18 Cantidad de dependencias por clasificación

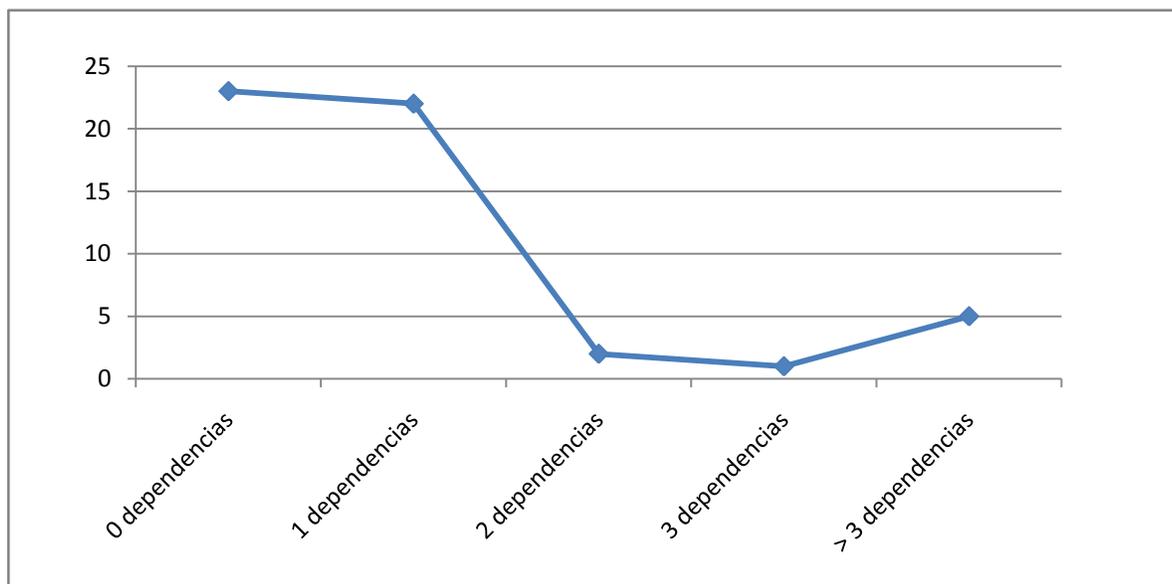


Fig. 21 Gráfica de los resultados de la evaluación de la métrica RC agrupados por las tendencias de los Valores

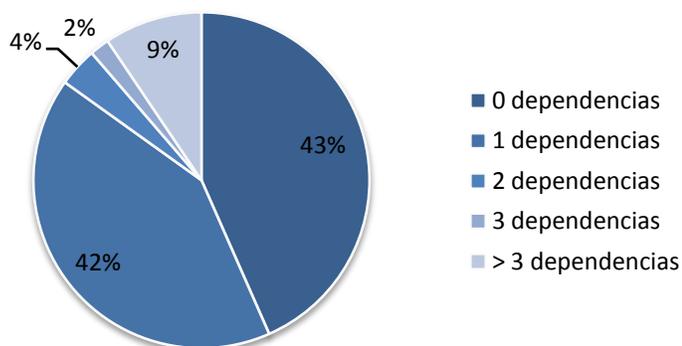


Fig. 22 Representación en % de los resultados obtenidos en el instrumento agrupados en intervalos definidos

Representación de la incidencia de los resultados de la evaluación de la métrica RC en cada uno de los atributos:

Acoplamiento	Cantidad de clases	Promedio
Ninguno	3	6
Bajo	8	16
Medio	2	4
Alto	2	4

Tabla 19 Atributo Acoplamiento

Acoplamiento

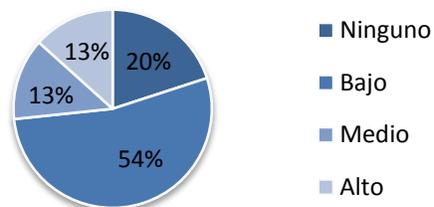


Fig. 23 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento

Complejidad de Mantenimiento	Cantidad de clases	Promedio
Baja	23	46
Media	2	14.28571429
Alta	1	7.142857143

Tabla 20 Atributo Complejidad de Mantenimiento

Complejidad de Mantenimiento

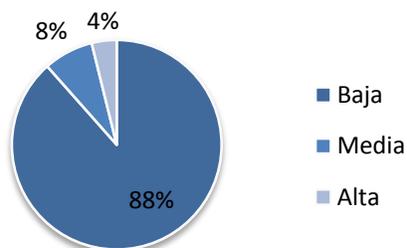


Fig. 24 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Mantenimiento

Cantidad de Pruebas	Cantidad de clases	Promedio
Baja	23	46
Media	2	4
Alta	5	10

Tabla 21 Atributo Cantidad de pruebas

Cantidad de Pruebas

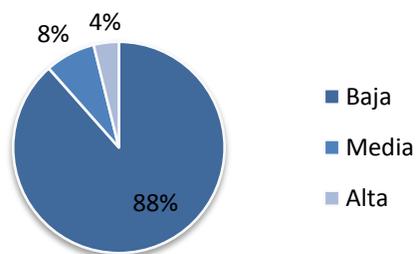


Fig. 25 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Mantenimiento

Reutilización	Cantidad de clases	Promedio
Baja	5	10
Media	2	4
Alta	23	46

Tabla 22 Atributo Reutilización



Fig. 26 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización

Se puede concluir que el diseño propuesto está entre los límites aceptables de calidad, teniendo en cuenta que la mayoría de las clases (91%) poseen menos de 3 dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio, el grado de acoplamiento es mínimo, la Complejidad de Mantenimiento, la Cantidad de Pruebas y la Reutilización para un 88%, 88%, 77% respectivamente, se comportan favorablemente.

3.5.2 Validación de la implementación

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. Tratan de encontrar errores en la siguientes categorías: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores de estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o desempeño y 5) errores de inicialización y término. (24)

Casos de Prueba

Al aplicar técnicas de caja negra se deriva un conjunto de casos de prueba que satisfacen los siguientes criterios: 1) casos de prueba que reducen, mediante una cuenta mayor que uno, el número de casos de prueba adicionales que deben diseñarse para alcanzar una prueba razonable y 2) casos de prueba que

indican algo acerca de la presencia o ausencia de clases de errores, en lugar de un error asociado sólo con la prueba específica a la mano. (24)

En el [anexo 2](#) se encuentran los casos de pruebas realizados a la aplicación donde se identificaron no conformidades, las cuales fueron corregidas para la entrega, garantizando un proceso de pruebas satisfactorio.

A continuación se muestran algunas estadísticas del proceso:

Cantidad de casos de prueba	8
Cantidad de personas involucradas	4
Cantidad de no conformidades	10
Etapas en las que se realizó el proceso	3

Tabla 23 Estadísticas de prueba

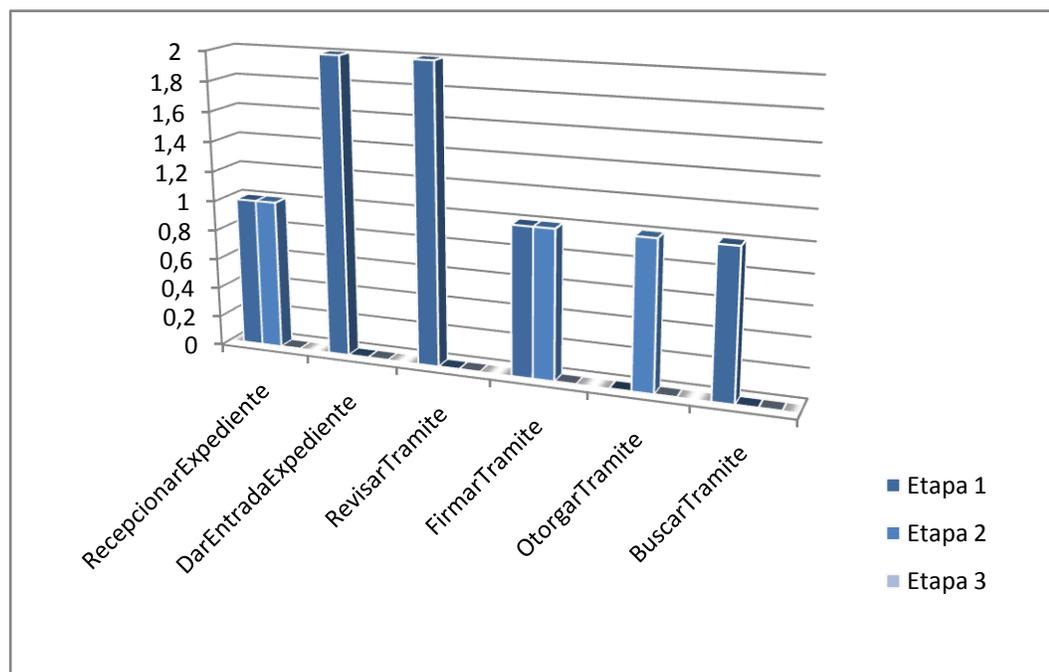


Fig. 27 Representación de no conformidades por etapas

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

Con la validación del diseño y la implementación se evaluó el grado de calidad y fiabilidad de la solución. Los artefactos generados serán entregados al equipo de calidad del Centro para su revisión y aprobación. Para alcanzar como resultado el acta de liberación de los artefactos del trabajo de investigación.

Conclusiones

En este capítulo se realizó el diagrama de componentes por capas, se explicaron algunas de las normas y estándares de codificación y de diseño de interfaz utilizadas y definidas por el equipo de desarrollo, las cuales se encuentran en el expediente de proyecto. Se validó la solución propuesta: para el diseño se utilizaron las métricas propuestas por Lorenz y Kidd obteniendo resultados satisfactorios y para la implementación se utilizó pruebas de caja negra, se diseñaron los casos de prueba comprobando que el flujo de trabajo de los mismos estuvo correcto y fueron rectificadas posteriormente las no conformidades.

CONCLUSIONES

Al concluir el presente trabajo se puede afirmar que:

1. La correcta selección de la metodología y tecnologías utilizadas para la construcción del software se realizó a partir de la elaboración del marco teórico de la investigación.
2. 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.
3. La solución informática como un sistema ejecutable 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 todas las funcionalidades del sistema.
4. El alto nivel de calidad y fiabilidad que presenta el diseño, la implementación y las pruebas realizadas se evaluó con la aplicación de métricas orientadas a objetos, diseño y aplicación de pruebas de caja negra y la corrección de las no conformidades respectivamente.
5. La solución informática logró satisfacer los requisitos acordados con los clientes con la realización del diseño e implementación de los módulos Recepción y devolución de documentos, Control de calidad y Otorgamiento del Centro de Digitalización para la División de Antecedentes Penales.

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

RECOMENDACIONES

A partir del análisis de los resultados alcanzados se recomienda:

1. Aplicar las experiencias obtenidas en el desarrollo de aplicaciones similares.
2. Continuar el proceso de validación del sistema para la obtención de la certificación de calidad del Grupo de Calidad UCI.
3. Trabajar en nuevas versiones del sistema que permitan la mejora continua del mismo a partir de nuevos requerimientos del cliente.

BIBLIOGRAFÍA

1. **Ruz, Fidel Catro.** 2002.
2. **Edmonson, R.** “*Directrices para la salvaguardia del Patrimonio Documental*”. s.l. : División de la Sociedad de la Información. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura., 2002.
3. **Valderrama, Luis David Fernández.** *La gestión documental.* Caracas-Venezuela : IESA Instituto de Estudios Superiores en Administración. .
4. **Colectivo de autores.** *Diccionario de la Real Academia de Española.*23ª Edición. 2001.
5. **Buendía, Lola.** *Digitalización de archivos en bibliotecas y centros documentales.* 2010.
6. *La Firma Digital.* **Colectivo de autores.** s.l. : Centro Informático CIXTEC.
7. **Colectivo de autores.** Kernel Doc. [Online] 2010. www.kerneldoc.es.
8. —. ADDOC. [Online] <http://www.addoc.com.mx>.
9. *Informatizamos para la sociedad cubana.* **Colectivo de autores.** s.l. : Desoft División de Servicios Generales, 2009.
10. **Fernández Bertot, Víctor and Orozco González, Delio G.** *Un sistema de conservación, digitalización, gestión y socialización de información documental para los archivos en Cuba.* Manzanillo, Granma : s.n., 200.
11. **Booch, Grady, Rumbaugh, Jim y Jacobson, Ivar.** *El Lenguaje Unificado de Modelado.* s.l. : Addison Wesley, 1999. ISBN: 84-7829-028-1.
12. **Colectivo de autores.** Sitio Web oficial Visual-Paradigm. [Online] Visual Paradigm Organización, 2009. <http://www.visual-paradigm.com>.
13. **JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James.** . *El Proceso Unificado de Desarrollo de Software.* s.l. : Pearson Addison-Wesley, 2000.
14. **Group, PostgreSQL Global Development.** *PostgreSQL 8.4.1 Documentation.* California : s.n., 2009.
15. **Medrano, Jesus Rafael Sanchez.** *Introducción a Base de Datos con PostgreSQL.* 2009.
16. **Casanova, Jaime y Herrera, Álvaro.** *MVCC. [presentación]* . Ciudad de la Habana : s.n., 2009.
17. **Herrera, Álvaro y Casanova, Jaime.** *WAL. [presentación].* Ciudad de la Habana : s.n., 2009.
18. **Martinez, Rafael.** PostgreSQL-es.org. [Online] 2009. http://www.postgresql-es.org/sobre_postgresql.

Diseño e implementación de los módulos de Recepción y Devolución, Control de Calidad y Otorgamiento de Documentos del Centro de Digitalización para la División de Antecedentes Penales.

19. **Ibarra, Antonio Aliaga y Flores, Marcos Agustin Miani.** 2008. **iessanvicente.com.** [En línea] 21 de enero de 2008. [Citado el: 16 de febrero de 2010.] Disponible en:. [Online] 2008. <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>.
20. **Wilson, Leslie B.** *Comparative Programming Languages, Second Edition.* s.l. : Addison-Wesley, 1993.
21. **Gómez, Omar.** *Especificación e implementación de software basado en componentes con la tecnología EJB 3.0.* [Online] Enero 2008. http://osgg.net/omarsite/resources/papers/cbse_ejb.pdf.
22. **Keith, Mike and Schincariol, Merrick.** *Pro JPA2 Mastering the Java Persistence API.* New York : Springer -Verlag, 2009. ISBN978-1-4302-1956-9.
23. **Colectivo de autores.** *Especificación de requisitos de software. Módulos de Inscripción, Certificación, Archivo, Reportes, Digitalización.* 2010.
24. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico Sexta Edición.*
25. **Garlan, David and Shaw, Mary.** *An introduction to software architecture.* s.l. : CMU Software Engineering Institute Technical Report, 1994.
26. **Díaz Rodríguez, Ivelisse Mercedes and Rodríguez Santos, Nallelys.** *Diseño e implementación del proceso de inscripción de documentos que se realizan en las Notarías Públicas de la República Bolivariana de Venezuela.* Ciudad de la Habana, Cuba : Institución Univesidad de las Ciencias Informáticas, 2010. Vol. 71.

Glosario de términos

API

API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.

DESOFT SA

Empresa surgida a partir de la integración y asociación de capital de entidades que funcionaban de manera aislada, pertenecientes al Grupo de Tecnologías de la Información que pertenecía al Ministerio de la Informática y las Comunicaciones. Su objetivo radica en la comercialización en el mercado nacional e internacional de productos y servicios, para lo cual cuenta con 15 divisiones territoriales, repartidas en todas las provincias, y una casa matriz en Ciudad de La Habana.

Framework

Es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otras aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

J2EE

Java 2 Enterprise Edition. Es una parte de la plataforma de programación Java. Basada en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

UNESCO

La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO). Lo más importante para este organismo de las Naciones Unidas no es construir escuelas en países devastados o publicar hallazgos científicos. El objetivo de la Organización es mucho más amplio y ambicioso: construir la paz en la mente de los hombres mediante la educación, la cultura, las ciencias naturales y sociales y la comunicación.