

“Facultad 15”



Título: Análisis y Diseño del Módulo Firmas Manuscritas.

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

Autores: Irislaidy Clemente García.
Misael Águila Oquendo.

Tutores: Ing. Javier Martínez Muñoz.

Fecha de presentación

Junio 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de tesis y se le reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Irislaidy Clemente García

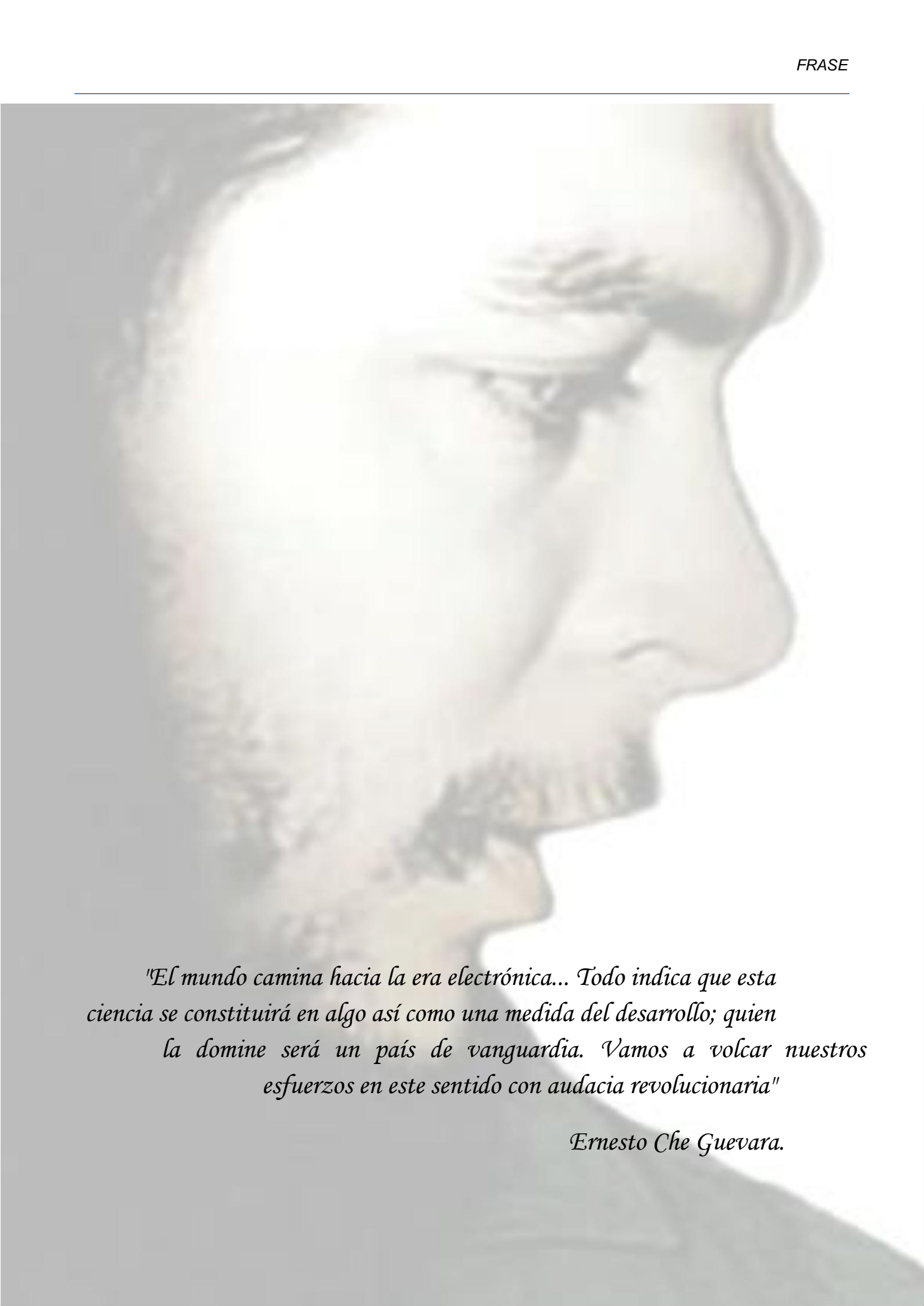
Misael Águila Oquendo

Firma del Autor

Firma del Autor

Javier Martínez Muñoz

Firma del Tutor



"El mundo camina hacia la era electrónica... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria"

Ernesto Che Guevara.

DEDICATORIA

A nuestros padres que siempre confiaron en nosotros y nos brindaron todo su apoyo para que vieran realizados nuestros sueños, queremos regalarles este momento y honrarlos por tanto amor y dedicación. Los queremos mucho.

A la Revolución, a Fidel y a todos los profesores que contribuyeron en la formación de la carrera.

AGRADECIMIENTOS

Muchas han sido las personas involucradas en el logro de este sueño. Quiero agradecer en primer lugar a mis padres por quererme con tanta intensidad y darme todo lo que tuvieron a su alcance, a quienes no solo agradezco sus consejos y el apoyo constante, sino además, el privilegio de ser su hijo.

A mi abuelita querida Benita y papi Grueiro, por demostrarme que si podía cuando yo pensaba que no, gracias por su apoyo, en verdad gracias.

A mi hermana Leo y su esposo Alain por brindarme todo su apoyo en todo estos años de estudio.

A mi compañera de tesis por confiar en mí y ayudarme en todo, gracias Irislaidy.

A mi primo principalmente, porque sin él nunca sería lo que soy hoy, esto te lo debo a ti Rubén Calaña te quiero hermano.

A mis más apreciados amigos: Manuel, Yasmany, Darien y Adolfo, gracias por todo su apoyo en toda mi carrera universitaria, se los agradezco hermanos.

A mis compañeros y compañeras de estudio, a mis amistades: Andriel (Koki), Yohairo, Armin, Elide, José Antonio (Pepe), Roniel, Osnielkys (Rocker), Livan y Javier (Aldo).

A mis profesores que tuve todos estos años (Karina, Canalejo, Carlos) a todos muchas gracias.

Por último quiero agradecer especialmente a la Revolución por darme la oportunidad de estudiar en la UCI que ha resultado tan positiva para mi formación personal y profesional. A todos los que de una forma u otra han contribuido a la realización de este trabajo de diploma, de corazón, muchas gracias.

Sin más Misael Águila.

Le agradezco a mis padres Estrella y Frank, a mis abuelos Dao y Papo, por demostrarme que si podía cuando yo pensaba que no, gracias por su apoyo.

A mi novio Yohairo por brindarme su confianza, por darme fuerzas y ayudarme cuando lo necesite; al igual que su familia por apoyarme en todo.

A mis primos Yosvani, Zulemi, Yasla y Yili que me ayudaron en todo, a Isis, Greidi, Iraelito.

A mi tíos y tías como (tío titi, tía Xiomara, Maritza, tito, Alfredo, blanco, prieto) a todos muchas gracias.

A mi compañero de tesis por confiar en mí y ayudarme en todo, más conocido, como Misael el Mago.

Al núcleo familiar que fueron como mi familia todos estos años, (Pepex el pelú, a Yusbel la rata prieta, a Néstor el sensible).

A mis compañeros y compañeras de estudio, a mis amistades, Luisa, Alena, Gretell, Damaris, Yuri, Estopa el lindo, Alberto conocido por Makano, Yai, Pilo, Yaimara, Ángel, Armin, Elide, Dany, José Manuel, Horshel, Tony, Tatiana, Diosnel, Livan.

A mis profesores que tuve todos estos años (Karina, Canalejo, Geiser) a todos muchas gracias.

Sin más Irislaidy Clemente.

RESUMEN

En el Sistema Automatizado de Gestión Bancaria (SAGEB) se gestionan gran parte de los procesos que son de vital importancia para el buen funcionamiento de un banco, actualmente existen dificultades a la hora de realizar el proceso de verificación de firmas manuscritas debido a que no existe un sistema automatizado en la Banca Nacional de Cuba que gestione dicho control de firmas.

Con el propósito de eliminar estas deficiencias y para contribuir con la modernización del Sistema Bancario Cubano la Universidad de las Ciencias Informáticas (UCI) tiene la misión de desarrollar un nuevo sistema automatizado que sea extensible a todas las entidades bancarias del país y que soporte cada una de sus funcionalidades.

El objetivo que se persigue con el presente trabajo es desarrollar el Análisis y Diseño del Módulo de Firmas Manuscritas del proyecto SAGEB, el mismo se ajusta a los lineamientos arquitectónicos establecidos dentro del proyecto. Se describe un estudio del estado del arte, así como características, elementos y procesos de las firmas manuscritas, como mecanismo de seguridad en los sistemas bancarios; además las técnicas y herramientas utilizadas, como la aplicación de patrones dentro del flujo de desarrollo de software siguiendo la metodología y procesos dictados por el proceso unificado de desarrollo.

Palabras claves

SAGEB, Análisis, Diseño, Firmas Manuscritas, Sistemas Bancarios.

TABLA DE CONTENIDO

DECLARACIÓN DE AUTORÍA	II
DEDICATORIA.....	IV
AGRADECIMIENTOS	V
RESUMEN	VII
INTRODUCCIÓN	1
ESTRUCTURA DEL DOCUMENTO	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	2
1.1. Introducción.....	2
1.2. Sistema Bancario Nacional	2
1.3. Las Firmas Manuscritas en el Mundo.	4
1.4. Firma	5
1.4.1. Características de las Firmas.....	5
1.4.2. Elementos de las Firmas.....	5
1.4.3. Tipos de Firmas.....	6
1.5. Pasos a seguir para el Reconocimiento y Verificación de Firmas Manuscritas.	7
1.5.1. Técnicas de Procesamiento de Imágenes.....	7
1.5.2. Métodos para la Esqueletización de Imágenes.	8
1.5.3. Comparación de Imágenes.....	9
1.6. Procesos de Control de Firmas Manuscritas.....	10
1.6.1. Ventajas e Inconvenientes.....	11
1.7. Ingeniería de Software.....	12
1.7.1. Metodología.....	12
1.7.1.1. Proceso Unificado de Desarrollo (RUP, Rational Unified Process).....	13
1.7.1.2. Lenguaje Unificado de Modelado (UML, Unified Modeling Language).....	14
1.7.1.3. Visual Paradigm	14
1.7.2. Lenguaje de programación Java.....	15
1.7.3. Plataforma J2EE (J2EE, Java Platform Enterprise Edition).....	15
1.7.4. Tecnologías y Herramientas de Desarrollo.....	15

1.8. Conclusiones Parciales.....	17
CAPÍTULO 2. ANÁLISIS DEL SISTEMA.....	18
2.1 Introducción.....	18
2.2 Requisitos Funcionales.....	18
2.3 Casos de Uso del Sistema.....	19
2.4 Análisis.....	19
2.5 Modelo de Análisis.....	20
2.6 Conclusiones Parciales.....	23
CAPÍTULO 3. DISEÑO DEL SISTEMA.....	24
3.1 Introducción.....	24
3.2 Fundamentación de la arquitectura.....	24
3.3 Diseño.....	27
3.3.1 Diagramas de clases del diseño.....	27
3.3.2 Diagrama de interacción del diseño.....	31
3.4 Aplicación de los Patrones.....	35
3.4.1 Patrón Arquitectónico Modelo Vista Controlador (MVC).....	35
3.4.2 Patrones GOF.....	36
3.4.3 Patrones de Diseño.....	37
3.4.4 Patrón Estructural.....	37
3.4.5 Patrón de Acceso a Acceso a Datos.....	37
3.4.6 Patrones de Asignación de Responsabilidades (GRASP).....	38
3.5 Conclusiones parciales.....	39
CONCLUSIONES.....	40
RECOMENDACIONES.....	41
BIBLIOGRAFÍA.....	42
ANEXOS.....	44
GLOSARIO DE TÉRMINOS.....	67

INTRODUCCIÓN

El desarrollo acelerado del mundo actual hace imposible que cualquier entidad o institución pueda controlar de forma manual su actividad fundamental. Es por ello que la mayoría utilizan sistemas automatizados que permitan agilizar su trabajo de manera óptima, eficiente y con el personal necesario. Los sistemas bancarios controlan la vida económica del mundo y su actuación estratégica se ha potenciado en las últimas décadas con el uso de los sistemas informáticos, los cuales han contribuido a lograr una mayor eficiencia en la toma de decisiones y en el resto de sus operaciones.

En particular, el Banco Central de Cuba, constituye una entidad en la cual la contabilidad es su actividad fundamental y para ejecutar sus funciones requiere del procesamiento de un gran volumen de datos donde se permita los accesos simultáneos de muchos usuarios (cajeros, supervisores, etc) interconectados entre sí.

Cuba a pesar de ser un país con grandes problemas económicos, se ha vinculado a diferentes tareas, unas de las cuales es automatizar el Sistema Bancario Nacional.

Respecto a esto el Comandante en Jefe "Fidel Castro Ruz" expresó:

" ..Ser economista excelente depende de la idea de lo que quiera hacer quien dirige un frente de la economía del país, y quien dirige el frente del Banco Nacional de Cuba; así, en su doble carácter de comunista y economista... " (CMHW, 2010)

Situación problemática

En Cuba a partir de la desintegración del campo socialista y el arreciamiento del bloqueo por parte del gobierno estadounidense, la economía se vio imposibilitada de mantenerse al tanto de los adelantos tecnológicos que en el mundo iban surgiendo. El país en aras de revertir esta situación ha destinado grandes cantidades de recursos de acuerdo a las posibilidades económicas para el desarrollo informático. De este modo, durante los primeros años de este siglo, se ha propuesto informatizar la sociedad, lo que condujo a la creación de nuevos sistemas a la altura de las nuevas tecnologías emergentes.

Al calor de la batalla de ideas, como una idea del Comandante en Jefe Fidel Castro, surgió la Universidad de las Ciencias Informáticas (UCI), para poner en marcha la total explotación de las tecnologías con el objetivo de formar profesionales altamente calificados y contribuir a la informatización de la sociedad.

En la facultad 15, perteneciente a la mencionada universidad, se desarrolla el proyecto SAGEB, el cual tiene como una de sus misiones, la automatización de la gestión de Firmas Manuscritas en las entidades financieras bancarias del país.

En nuestro país, el problema del reconocimiento de firmas en documentos bancarios ha sido obviado o relegado a la responsabilidad de las personas, en muchos casos a funcionales no expertos en la detección de Firmas Manuscritas, ya que no existe un producto terminado que cumpla con los requisitos sino solo investigaciones acerca de las firmas. Esto podría atribuirse a la necesidad de altas capacidades de cómputo y a la complejidad funcional de los sistemas que existen.

En la actualidad para el Banco Nacional de Cuba representa un riesgo la verificación de firmas, debido a que no se tiene un sistema capaz de reconocer o rechazar dos firmas manuscritas del mismo autor, por lo que puede traer como consecuencias los siguientes factores negativos:

- Riesgos económicos en caso de operaciones fraudulentas.
- Empobrecimiento de la imagen ante los clientes.
- Retrasos en las operaciones.
- Elevados tiempos de espera para los clientes.
- Costes de personal innecesarios.

Apoyándose en el estudio desarrollado se puede definir el **problema a resolver**: ¿Cómo automatizar la gestión de Firmas Manuscritas para el proyecto SAGEB? .Se define como **objetivo general**; desarrollar un módulo que permita la gestión de Firmas Manuscritas para dicho proyecto. A partir de ahí se puede identificar como **objeto de estudio** las Firmas Manuscritas en entidades financieras bancarias de Cuba, teniendo como **campo de acción** la gestión de Firmas Manuscritas en el Banco Nacional de Cuba. Como **resultado**, el análisis y diseño de una solución general para la gestión de las Firmas Manuscritas a todas las entidades financieras bancarias de Cuba para su posterior implementación.

Para dar cumplimiento al objetivo planteado se definen las siguientes **Tareas a realizar**:

- Caracterizar los requisitos funcionales del módulo Firmas Manuscritas del proyecto SAGEB.
- Caracterizar los sistemas automatizados nacionales e internacionales que soportan la realización de dichos procesos.
- Caracterizar algoritmos de procesamiento de imágenes.
- Caracterizar las herramientas de desarrollo a utilizar durante el diseño del módulo Firmas Manuscritas del proyecto SAGEB.
- Realizar el modelo de diseño del módulo Firmas Manuscritas del proyecto SAGEB.

ESTRUCTURA DEL DOCUMENTO

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Este capítulo constituye la fundamentación teórica del presente trabajo. Su lectura le ofrecerá información referente a los principales conceptos tratados. Lo pondrá al tanto del estado del arte del tema y por último podrá conocer la herramienta, la metodología y los lenguajes de modelado a utilizar para dar solución al problema en cuestión.

CAPÍTULO 2. ANÁLISIS DEL SISTEMA

Se establecerán los requisitos funcionales para crear un sistema automatizado que cumpla con las expectativas de los usuarios finales y así comenzar con el modelado del mismo. Se hace, además, una descripción de los casos de uso y se muestra el diagrama de casos de uso del sistema.

CAPÍTULO 3. DISEÑO DEL SISTEMA

En este capítulo se describe la arquitectura del Módulo Firmas Manuscritas, se explica el modelo del diseño, así como los patrones utilizados para dar respuesta a la solución planteada. Se esbozan todas las clases para cada uno de los objetos determinados y se propone un modelo de datos que sustente las clases desarrolladas.

Finalmente se muestran las conclusiones del trabajo, las recomendaciones propuestas, la bibliografía, los anexos y el glosario de términos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se pretende mostrar un estado del arte del tema. Se abordarán conceptos, características, relacionados con las firmas, tipos de firmas, procesos de control de firmas manuscritas, algoritmos que pueden utilizarse para la verificación de dichas firmas entre otros aspectos; los cuales permitirán un mejor entendimiento de los procesos a analizar, así como una síntesis de la propuesta del software para la gestión de procesos bancarios del proyecto SAGEB. Además se hará un breve resumen sobre la Ingeniería de Software haciéndose énfasis en la metodología, herramienta y lenguaje que se utilizará para modelar el subsistema.

1.2. Sistema Bancario Nacional.

Se ha hecho evidente que las transformaciones organizativas y las normativas efectuadas en la economía cubana, demandan una ampliación y diversificación del sistema bancario y financiero. Tales cambios deben ser capaces de permitir enfrentarse al establecimiento de una relación diferente con la comunidad internacional en materia comercial y financiera. Consecuentemente, a partir de 1995, se diseñó e implementó, gradualmente, un sistema encaminado a garantizar el funcionamiento de la economía cubana en las nuevas circunstancias y en el marco de la estrategia antes mencionada. (Bello, 2008)

- El Banco Central de Cuba (BCC) se constituyó como autoridad rectora, reguladora y supervisora de las instituciones financieras y de las oficinas de representación que radican en el país. Sus objetivos fundamentales consisten en emitir la moneda nacional y velar por su estabilidad, proponer e implementar la política monetaria del país y actuar como órgano rector del sistema bancario y financiero. (Bello, 2008)
- Banco de Crédito y Comercio (BANDEC), está autorizado a ejercer funciones de banca comercial o de múltiples servicios, constituyéndose el sector empresarial en su principal centro de atención y tiene como misión proporcionar a sus clientes la máxima seguridad y rentabilidad en la gestión de sus recursos financieros con una adecuada política de colocación de pasivos, a la satisfacción de las necesidades económicas y de desarrollo del país. (Bello, 2008)
- Banco Popular de Ahorro (BPA), asumió funciones de banca universal, extendiendo sus servicios al sector empresarial, ampliando así el número de sucursales que prestan servicios a ese sector. Continúa siendo el banco líder de los clientes naturales en moneda nacional y en moneda libremente convertible. (Bello, 2008)

- Banco Financiero Internacional, S.A. (BFI). Desde su constitución hasta la fecha ha operado como un banco comercial, dirigido en lo fundamental a prestar servicios a las entidades jurídicas cubanas y extranjeras, asociaciones económicas, empresas mixtas y a personas naturales cubanas .Presta servicios de créditos a las empresas y mantiene relaciones de corresponsalía con una amplia red de bancos en el extranjero. (Bello, 2008)
- Banco Internacional de Comercio, S. A. Brinda una amplia gama de servicios bancarios a entidades cubanas, extranjeras y mixtas. Realiza operaciones en moneda libremente convertible, fundamentalmente relacionadas con el comercio exterior, financiamientos y operaciones de compraventa de moneda. (Bello, 2008)
- Banco Metropolitano, S.A .Tiene como objetivo fundamental prestar servicios especiales de banca privada y colateralmente operaciones lucrativas relacionadas con el negocio de banca en moneda libremente convertible y moneda nacional, así como operaciones comerciales a través del Banco Internacional de Comercio, S. A. Sus principales clientes lo forman el Cuerpo Diplomático acreditado en Cuba y firmas extranjeras; extranjeros residentes, permanentes o temporales, en el país y particulares cubanos. (Bello, 2008)
- Banco de Inversiones, S. A .Orienta sus actividades a la prestación de servicios financieros especializados en materia de inversión, canalizando hacia ella el ahorro externo. Extiende sus servicios hacia el asesoramiento y preparación técnica de las empresas cubanas; promueve sus proyectos de inversión mediante la asesoría financiera corporativa y comercial, la cobertura de riesgos cambiarios y otras afines con sus atribuciones. (Bello, 2008)
- Banco Exterior de Cuba (BEC). De propiedad estatal, está dotado de autonomía orgánica, personalidad jurídica independiente y patrimonio propio. Está facultado para desarrollar funciones inherentes a la banca universal o de múltiples servicios, en el territorio nacional, centro bancario extraterritorial, zonas francas, parques industriales y en el extranjero, estando autorizado a realizar transacciones en divisas y en moneda nacional. (Bello, 2008)
- Banco Industrial Venezuela Cuba S.A. Recibió del Banco Central de Cuba, por Resolución 47 del 2005, la autorización para operar como institución bancaria en el territorio nacional, otorgándosele una Licencia Especial que le faculta para llevar a cabo operaciones de intermediación financiera en divisas. La licencia recibida le permite operar el negocio bancario en Cuba por 50 años y precisa que debe destinar anualmente un porcentaje de sus utilidades netas para crear e incrementar una reserva legal que cubra riesgos y posibles pérdidas futuras, que alcance, como mínimo un monto igual al de su capital. (Bello, 2008)

- Banco Nacional de Cuba (BNC), liberado de las funciones de banca central y de rector del sistema bancario, continuará existiendo con el carácter de banco comercial, autorizado a ejercer funciones inherentes a la banca universal, teniendo además la función de registro, control, servicio y atención de la deuda externa que el estado y el propio banco han contraído con acreedores extranjeros con la garantía del estado, como hasta el presente. (Bello, 2008)

1.3. Las Firmas Manuscritas en el Mundo.

Por décadas la firma manuscrita ha servido para identificar la autoría de documentos, sin embargo esta firma desde su invención ha acarreado imperfecciones. Unas de estas son la falsificación y el procedimiento de verificación de la firma. A pesar de esto dicha firma ha servido como el método más aceptado para verificar la identidad de una persona.

La firma en sí, tiene dos acciones, la acción de firmar y la acción de verificación de la firma. Para la primera solo basta que un individuo pueda escribir su nombre, o algún conjunto de caracteres y líneas particulares. La acción de la verificación de la firma, es más complicada sin embargo, en la práctica podemos ver todos los días este procedimiento al cambiar un cheque o al comprar con una tarjeta de crédito. El proceso de verificación de la firma en general se realiza de forma visual, es decir, el cajero de un banco solo compara la firma del cheque con otra que está en alguna identificación o que se realice ahí mismo, y de esta manera acepta o rechaza la firma, es decir, acepta o rechaza la identidad del firmante del cheque.

Para casos más difíciles, una firma se puede someter a un procedimiento de verificación, en un litigio. Esta verificación es ordenada por un juez y quiere decir que el procedimiento visual no basta para poder declarar a la firma como aceptada o rechazada. Este procedimiento lo lleva a cabo un perito (un experto) en verificación de firmas. Aunque no existe una norma mundial para determinar con precisión que la firma es auténtica, el procedimiento se basa en una serie de pruebas que chequean las características propias de las líneas y que están asociadas a cada individuo. Tales características son por ejemplo: el grosor de la línea, la inclinación de los caracteres usados, la terminación de los caracteres, la curvatura de las líneas, etc. Se considera a la firma manuscrita una biometría, es decir, una característica física propia de cada individuo y que deja marcada estas características en la firma. Según lo documentado, con más del 75% de las pruebas sean positivas, entonces un perito puede declarar a una firma como aceptada. De la misma forma el número de pruebas a que se somete una firma es determinado por el perito y depende de su experiencia en el ramo. Estos métodos se han ido refinado al pasar el tiempo, sin embargo, muchos de ellos no corresponden a procedimientos determinísticos y dependen mucho de la experiencia del perito. Los mismos grupos de expertos reconocen que estos procedimientos pueden contar con una probabilidad de error que puede alcanzar el 20%. (Computación, 2010)

1.4. Firma

La **Firma** es el nombre y apellido, o título, que una persona escribe de su propia mano en un documento, para darle autenticidad o para expresar que aprueba su contenido.

1.4.1. Características de las Firmas.

➤ Identificativa

Sirve para identificar quién es el autor del documento.

➤ Declarativa

Significa la asunción del contenido del documento por el autor de la firma. Sobre todo cuando se trata de la conclusión de un contrato, la firma es el signo principal que representa la voluntad de obligarse.

➤ Probatoria

Permite identificar si el autor de la firma es efectivamente aquél que ha sido identificado como tal en el acto de la propia firma.

1.4.2. Elementos de las Firmas.

➤ Elementos formales

Son aquellos elementos materiales de la firma que están en relación con los procedimientos utilizados para firmar y el grafismo mismo de la misma.

➤ Elementos funcionales

Tomando la noción de firma como el signo o conjunto de signos, podemos distinguir una doble función:

➤ Identificadora

La firma asegura la relación jurídica entre el acto firmado y la persona que lo ha firmado. La identidad de la persona determina su personalidad a efectos de atribución de los derechos y obligaciones. La firma manuscrita expresa la identidad, aceptación y autoría del firmante. No es un método de autenticación totalmente fiable. En el caso de que se reconozca la firma, el documento podría haber sido modificado en cuanto a su contenido -falsificado- y en el caso de que no exista la firma autógrafa puede ser que ya no exista otro modo de autenticación. En caso de duda o negación puede establecerse la correspondiente pericial caligráfica para su esclarecimiento.

➤ Autenticación

El autor del acto expresa su consentimiento y hace propio el mensaje. (Informática, 2010)

1.4.3. Tipos de Firmas.

**Firma Digital****Firma Manuscrita**

La **Firma Digital** es la transmisión de mensajes telemáticos, un método criptográfico que asegura su integridad así como la autenticidad del remitente, es una de las aplicaciones de la criptografía asimétrica de mayor éxito y es usada para establecer la autenticidad e integridad de mensajes y documentos electrónicos. La verificación de la firma digital permite determinar cuando un mensaje o documento ha sido alterado. Además tiene la propiedad de que solo puede ser producida de manera correcta por una entidad, y ser verificada por cualquiera que reciba el mensaje o documento firmado digitalmente.

En nuestro país, los sistemas bancarios para gestionar los procesos de verificación, utilizan las Firmas Manuscritas las cuales constituyen uno de los principales métodos de justificación de autoría que manejamos las personas. Una de sus ventajas radica en la facilidad de realización, pues apenas se necesita un papel y un bolígrafo

Otra ventaja está en la dificultad de realizar una falsificación que pueda engañar a una persona experta. Pero quizás su mayor ventaja esté en que es un mecanismo de autenticación personal ampliamente aceptado por la sociedad, en general, y por organismos públicos y privados, en particular.

Es por estas razones que la construcción de un sistema que realice la tarea de verificación de firmas de manera automática suscita un enorme interés. El problema se ha abordado desde múltiples enfoques, habiéndose logrado resultados esperanzadores bajo condiciones controladas.

1.5. Pasos a seguir para el Reconocimiento y Verificación de Firmas Manuscritas.

1. Procesamientos de Imágenes.

2. Esqueletización de Imágenes.

3. Comparación de Imágenes.

1.5.1. Técnicas de Procesamiento de Imágenes.

La formación de una imagen digital es el primer paso para cualquier procesamiento de imágenes digitales, y consiste básicamente en un sistema óptico y el digitalizador, mediante el cual la imagen óptica se transforma en una señal eléctrica que permitirá el procesamiento. Al digitalizar una imagen, es común introducir ruido o degradación de la misma, por ello es importante considerar técnicas para restaurarla antes de procesarla; éstas consisten generalmente en disminuir el nivel de ruido, mejorar el contraste de la imagen, lograr la no-uniformidad de la imagen, su alineación, vecindad, etc.

Objetivo del Procesamiento de Imágenes

Mejorar la calidad de las imágenes para su posterior utilización o interpretación.

Técnicas de Procesamiento:

Adquisición: Se refiere al proceso de convertir un documento a una representación apta para ser procesada por la computadora. Tal adquisición y conversión puede ser realizada por un escáner, cámara fotográfica o de video, etc. y el resultado puede variar dependiendo del proceso de digitalización usado y el método de codificación.

Binarización: Convierte la imagen recibida en una imagen binaria separando así el fondo de los objetos a analizar.

Análisis de documento: Trata de analizar la estructura de un documento y entender la información contenida en sus componentes.

Segmentación: Se refiere a delimitar las regiones que nos interesan. En el caso del reconocimiento de caracteres se refiere a las líneas y caracteres individuales.

Normalización: Trata de ajustar la forma, tamaño y posición del carácter esto con el objetivo de minimizar la variación entre imágenes de la misma clase.

Extracción de características: Permite conocer las características como lo son el tamaño, perímetro, área, etc. Así como características topográficas como lo es la orientación de segmentos.

Reconocimiento: Convierte la imagen binaria en una representación electrónica. La misma permite realizar operaciones de validación y análisis.

1.5.2. Métodos para la Esqueletización de Imágenes.

Se intenta abordar aquí los distintos métodos conocidos para extraer el esqueleto de una imagen. Un esqueleto intenta representar la forma de un objeto con un número relativamente pequeño de píxeles. De esta forma, todos los píxeles del esqueleto son estructuralmente necesarios. La posición, orientación y longitud de las líneas del esqueleto se corresponden con aquellas equivalentes de la imagen original. La tarea de sacar características de una imagen queda simplificada al obtener su esqueleto.

El esqueleto de una imagen se puede extraer fácilmente utilizando alguna de las distintas técnicas de adelgazamiento.

Se define adelgazamiento como el acto de identificar aquellos píxeles pertenecientes a un objeto que son necesarios para mantener la forma del mismo. Estos píxeles son el esqueleto. (Universidad de Deusto, 2008)

Tipos de Algoritmos

- Algoritmo de Stentiford.
- Algoritmo de Zhang-Suen.
- Algoritmo de Holt.

Algoritmo de Stentiford

En este algoritmo, las reglas que especifican como eliminar los píxeles sobrantes se basan en plantillas. Se utilizan cuatro plantillas distintas, cada una utilizada en cada una de las cuatro direcciones en las que se puede recorrer una imagen. Una vez que se encuentra una coincidencia con alguna de las plantillas, el píxel se elimina. Cuando ya no es posible encontrar más coincidencias, se considera que se ha encontrado el esqueleto de la imagen.

Esta forma de adelgazar es muy costosa, pero es un ejemplo típico de algoritmos basados en plantillas. (Universidad de Deusto, 2008)

Algoritmo de Zhang-Suen

Este método ha sido utilizado durante años como comparación del resto de algoritmos.

Es rápido y sencillo de implementar. Es un método paralelo, es decir, cada píxel se puede calcular utilizando el valor de la iteración anterior. Si tuviésemos una CPU por cada píxel, se podría determinar la siguiente iteración instantáneamente.

El método consta de dos subiteraciones. En cada una de ellas se eliminarán aquellos píxeles que cumplan todas las reglas definidas para la subiteración. (Universidad de Deusto, 2008)

Algoritmo de Holt

Este método mejora al anteriormente descrito, es más rápido y además no necesita dos subiteraciones. Sólo sobrevivirán aquellos píxeles que cumplan una determinada expresión.

El resultado es bueno, pero no se parece al obtenido por Zhang-Suen. (Universidad de Deusto, 2008)

Sugerencias

Si lo que se necesita es velocidad en el adelgazamiento, lo mejor es utilizar el algoritmo de Holt. Si lo que se necesita es calidad en el esqueleto, lo mejor es una mezcla de tres de los elementos anteriores. Así, el proceso sería el siguiente:

- Preprocesado de Stentiford: Primero suavizado y a continuación énfasis en los ángulos cerrados.
- Adelgazamiento: Por el método de Zhang-Suen.
- Post-procesado: Eliminación en escalera de Holt.

1.5.3. Comparación de Imágenes.

Una vez realizado el procesamiento y la esqueletizado de la imagen se realiza la comparación de estas utilizándose diferentes algoritmos tales como:

- Verificación de firmas mediante Snakes.
- Aplicación del Kernel de Fisher.
- Descriptores de Fourier.

Los **Snakes** son un tipo de modelos de contornos activos, que se basan en el estudio del movimiento de un contorno abierto o cerrado sobre una imagen a la que trata de adaptarse.

Asociado al contorno se define una función de energía con una componente interna y otra externa. La componente interna lo dota de elasticidad y flexibilidad. La componente externa permite a una imagen influir sobre él. (Miguel, 2008)

Kernel de Fisher: Este método de verificación de firmas manuscritas offline está basado en modelos ocultos de Markov cuya misión es transformar el vector característico de entrada al espacio de puntuaciones de Fisher. (Ferrer, 2008)

1.6. Procesos de Control de Firmas Manuscritas.

➤ **Online.**

➤ **Offline.**

El proceso de captura de firmas puede realizarse de dos formas diferentes: **on-line** y **offline**. En el modo **on-line** la captura de la firma se realiza utilizando un dispositivo especial (clásicamente una lápiz electrónico y una tableta gráfica especial) que recoge información dinámica del escritor durante el acto de firmar. Esta información incluye, además del grafismo, datos como la presión, la velocidad, los puntos de inicio, las direcciones de los trazos, la inclinación, etc. Existen actualmente sistemas que realizan procesos muy precisos de reconocimiento y verificación utilizando esta información. (Tormo, 2008)

Un ejemplo, que utiliza el proceso Online en nuestro país es el (BICSA), Banco Internacional de Comercio S.A, el cual fue constituido por Escritura de Constitución No. 49, de 29 de octubre de 1993 y es miembro del Grupo Nueva Banca S.A. (BC, 2010)

Su actividad fundamental la constituye la "banca de empresas", la que brinda a través de sus servicios centrales, y de cinco sucursales radicadas en la capital del país, en Santiago de Cuba y en Villa Clara. El registro de todas las transacciones es realizado en tiempo real, ofreciendo a sus clientes servicios de tarjetas y de banca remota, a la vez que se trabaja en el desarrollo de otras modalidades de banca electrónica.

Sus clientes institucionales, tanto nacionales como extranjeros, reciben un completo servicio de cuentas y documentario, mientras que las entidades nacionales, además, disfrutan de importantes volúmenes de facilidades crediticias. Prácticamente todos los sectores de la economía se benefician de esto, entre ellos, el comercio tanto exterior como interno, el azúcar, la informática y las comunicaciones, así como otros de importante incidencia, no sólo económica sino también social, como la salud, el abasto de agua, la educación, la cultura y el deporte.

El método de captura **off-line** se basa en el escaneo de una firma, una vez realizada sobre un soporte ordinario (clásicamente papel).

En este caso, la información de que se dispone es mucho menor y la resolución espacial y radiométrica a la que se escanee la firma son elementos que pueden influir en la verificación. Además, en la formulación **off-line**, aparece el problema de localizar y segmentar la firma dentro del documento. A su vez, la segmentación de la firma dentro de un documento general presenta diferentes problemas: desconocimiento de la posición exacta de la firma, existencia de ruido blanco y de ruido con estructura.

La Figura 1 ilustra algunos de estos problemas: sellos que se han superpuesto a la firma (c) y (d), tramas de logotipos sobre los que se ha firmado (b), texto adyacente a la zona de firmado (a) y líneas o recuadros sobre los que se firma (a).

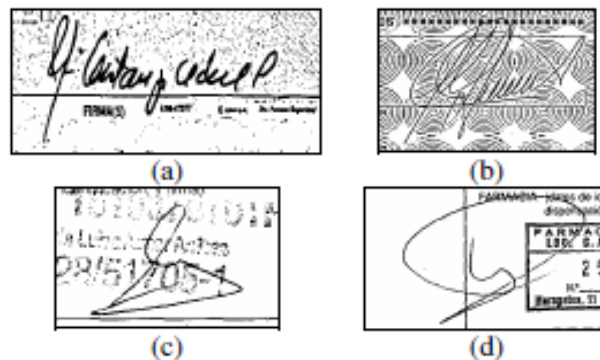


Figura 1. Ejemplos de ruido blanco (a) y estructural (b), (c) y (d) que dificultan la segmentación de la firma.

1.6.1. Ventajas e Inconvenientes.

Los dos grupos de métodos de reconocimiento de firmas presentan ventajas e inconvenientes. En el reconocimiento **on-line** se obtiene mucha más información. Se pueden obtener secuencias ordenadas de puntos agrupados en trazos simplemente fijándose cuándo el usuario apoya y levanta el lápiz, en comparación con el bitmap desordenado que se obtiene al escanear un documento.

Para cada punto se puede saber el momento exacto en el que se dibujó pudiéndose calcular características como la velocidad, la aceleración del lapicero (midiendo la distancia entre dos puntos escaneados en un periodo de tiempo fijo), incluso algunas tabletas permiten saber la presión y ver cómo varía a lo largo de la firma.

Teniendo en cuenta que se recibe información inmediata del lápiz, cabe la posibilidad de aprender y ajustar el sistema de reconocimiento en tiempo real mejorando los índices de acierto, incluso el usuario puede enseñar al sistema como le gusta escribir ciertas letras o trazos.

Uno de los mayores inconvenientes de los sistemas **on-line** es que requieren el uso de material especial (tableta, lápiz, etc.) y la posibilidad de que varíen ciertas características ya que no es lo mismo escribir con un bolígrafo sobre papel que con uno de plástico sobre una pantalla táctil. En este tipo de sistemas es muy importante la velocidad de respuesta puesto que el usuario se encuentra esperando a que finalice el reconocimiento. (Eslova, 2010)

Por otra parte, al analizar una firma como imagen según un proceso **off-line** se tienen una serie de problemas típicos asociados al Reconocimiento Óptico de Caracteres (OCR), tales como la traslación (vertical y horizontal), la rotación, el problema de dimensionamiento, así como la menor cantidad de información, por mencionar solo algunos. Sin embargo, presenta alguna ventaja como es la posibilidad de aplicarse sin la presencia del sujeto y resulta menos costoso a nivel de equipamiento.

1.7. Ingeniería de Software.

La Ingeniería de Software es una disciplina que ofrece métodos y técnicas para desarrollar y mantener software con calidad que cumplan con las expectativas de los usuarios. Se define como la práctica del conocimiento científico en el desafío y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software. (Inc, 2001)

“Los objetivos claves de la ingeniería de software son definir, crear y aplicar una metodología definida, dirigida a un ciclo de vida de planeamiento, desarrollo, y mantenimiento; un conjunto establecido de componentes de software que documenta cada paso en el ciclo de vida y muestra un seguimiento paso a paso, y un conjunto de hitos predecibles que pueden ser revisados a intervalos regulares a través del ciclo de vida del software”. (Pressman, 1997)

1.7.1. Metodología.

Una metodología no es más que el estudio de los métodos más apropiados que se emplean para desarrollar software de manera eficiente; o como precisan otros autores. (Jacobson, 2000)

El proyecto decidió utilizar como metodología de desarrollo RUP (Rational Unified Process), para aplicar durante toda la vida de desarrollo del software. La utilización de esta metodología propone dividir el proceso en cuatro fases, dentro de las cuales se realizan iteraciones que varían de acuerdo a las características de cada software.

1.7.1.1. Proceso Unificado de Desarrollo (RUP, Rational Unified Process).

RUP fue la metodología de desarrollo de software que se definió para utilizar en el SAGEB. Su objetivo es asegurar la construcción de sistemas de software de alta calidad que satisfagan las necesidades de los usuarios finales y clientes cumpliendo con los cronogramas y presupuestos previstos. Es adecuada para sistemas con extensos cronogramas y equipos de desarrollo numerosos. RUP es un proceso que se caracteriza por ser según. (Jacobson, 2000)

- Dirigido por casos de uso. Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo.
- Centrado en la arquitectura. La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto.
- Iterativo e incremental. Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. En cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión de proyecto y Ambiente. Cada iteración añade funcionalidades al producto de software o mejora las existentes.

En la Figura 1.1 se muestran las fases y las disciplinas que define RUP.

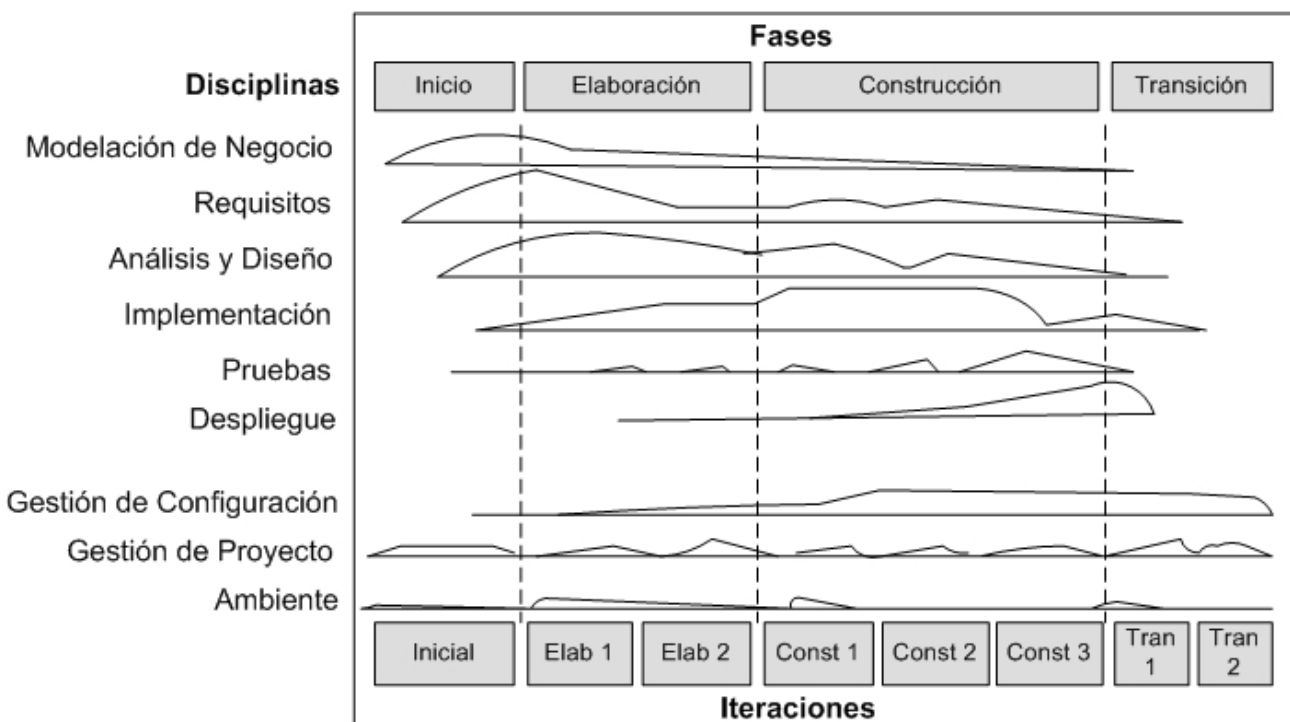


Figura 1.1 Fases y disciplinas en RUP.

Esta metodología propone un gran número de actividades y la elaboración de un amplio conjunto de artefactos, que usualmente los proyectos de desarrollo de software se comprometen a desarrollar pero que no realizan en su totalidad debido a la carencia de tiempo o a que descubren que no eran necesarios. RUP indica que al inicio del proyecto se realice una adecuación de cada flujo de trabajo de manera que se produzcan solo los artefactos y se realicen las actividades que tienen un propósito dentro del proyecto.

1.7.1.2. Lenguaje Unificado de Modelado (UML, Unified Modeling Language).

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Contiene construcciones para representar decisiones de implementación y para elementos de tiempo de ejecución en componentes. (Rumbaugh, 2007)

1.7.1.3. Visual Paradigm

Visual Paradigm es una herramienta CASE (Computer Aided Software Engineering) para visualizar y diseñar elementos de software. Utiliza UML como lenguaje de modelado, y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación, pruebas y despliegue. Tiene entre sus características el soporte de aplicaciones Web, y la integración con herramientas Java como: JBuilder, NetBeans IDE, Eclipse IDE. Con Visual Paradigm se puede generar código para diferentes lenguajes de programación. (Visual Paradigm, 2010)

Características

- Diagramas de Procesos de Negocio (Proceso, Decisión, Actor de negocio, Documento).
- Generación de código (Modelo a código, diagrama a código).
- Generación de bases de datos (Transformación de diagramas de Entidad-Relación en tablas de base de datos).
- Diagramas de flujo de datos.

1.7.2. Lenguaje de programación Java.

Java como lenguaje de programación ofrece diversas características como ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos para el desarrollo de aplicaciones. Es un lenguaje distribuido, proporcionando 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. Java es robusto ya que fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Siendo un lenguaje seguro presenta barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina. Por otra parte, es interpretado, ya que se puede ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real. Independiente de la arquitectura de hardware: 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.

1.7.3. Plataforma J2EE (J2EE, Java Platform Enterprise Edition)

J2EE es una plataforma que define un estándar para el desarrollo de aplicaciones empresariales multicapa. Simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un conjunto completo de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

1.7.4. Tecnologías y Herramientas de Desarrollo

Las herramientas y tecnologías a utilizar en el desarrollo de un software tienen estrecha relación con el tiempo de desarrollo y la calidad final del producto. Todas las herramientas y tecnologías utilizadas en la realización del Módulo Firmas Manuscritas fueron definidas por el proyecto SAGEB con el objetivo de garantizar la calidad del producto. A continuación se hace una breve descripción de las herramientas y tecnologías utilizadas.

- **Eclipse 3.5 SDK:** Ambiente de desarrollo integrado.
- **Dojo toolkit**

Es una biblioteca JavaScript de código abierto, la cual brinda una variedad de clases y widgets para facilitar el desarrollo de aplicaciones web. Dojo puede ser interpretado por diferentes navegadores web y posee un sistema de empaquetado muy parecido al del JDK de Java. La gran variedad de clases componentes y widgets, es un punto muy importante en la elaboración en la capa presentación de la aplicación, posibilita a los programadores de interfaz una serie de funcionalidades y elementos dinámicos que facilita la programación en el cliente.

➤ **JavaScript**

Lenguaje de programación interpretado por un cliente web, presenta una sintáxis parecida Java y C++. No es un lenguaje orientado a objetos, debido a que no dispone de herencia y otros paradigmas de la Programación Orientada a Objetos, es un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base y extendiendo su funcionalidad. Permite interactuar con diferentes tecnologías web como CSS, DOM y está condicionado en la unión de las diferentes tecnologías web.

➤ **Apache Tomcat**

Apache Tomcat es un contenedor de Servlet usado en la implementación de referencia oficial para las tecnologías Java Servlet y Java Server Pages (JSP). Es desarrollado en un ambiente colaborativo y abierto. Apache Tomcat es usado en numerosas aplicaciones web de gran escala y críticas en diversas industrias y organizaciones que se referencian en su sitio oficial.

➤ **Hibernate Tools**

Constituye un conjunto de herramientas para facilitar el uso del framework Hibernate. Las principales funcionalidades que brinda son: un editor de mapeos de los metadatos de la base de datos a las clases y una consola para la ejecución de consultas HQL. Permite realizar ingeniería inversa a partir de la base de datos de las clases y de los mapeos de las entidades.

➤ **Spring IDE**

Sirve como interfaz gráfica para la configuración de los archivos usados por Spring Framework. Permite el completamiento de etiquetas, valores de atributos y elementos en estos archivos de configuración.

1.8. Conclusiones Parciales.

En el presente capítulo se realizó una investigación sobre la situación de las entidades bancarias a nivel nacional e internacional, donde se dejó plasmado los estándares y sistemas automatizados que son utilizados en ambos niveles. Se realizó además un estudio de las firmas, como se desarrolla el control de verificación de dichas, así como las metodologías, herramientas y lenguajes más conocidos a nivel mundial y nacional, para definir cual utilizar y así lograr un mejor desarrollo del software.

CAPÍTULO 2. ANÁLISIS DEL SISTEMA.

2.1 Introducción

En este capítulo se presenta la lista de los requerimientos correspondientes al Módulo Firmas Manuscritas así como los Casos de Uso del sistema y una breve descripción de cada uno de ellos, de esta forma se crean las bases para comenzar a realizar el análisis y el diseño del subsistema. Luego se expone el análisis del sistema correspondiente a dichos requerimientos, mediante el modelo de análisis.

2.2 Requisitos Funcionales.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. (Pressman, 1997)

Durante la misma se capturan y definen las necesidades que debe cubrir el sistema. En cualquier proyecto software, dicha ingeniería es fundamental para garantizar los resultados del sistema. A continuación se muestra el listado de los requerimientos del Módulo de Firmas Manuscritas identificados en el documento de Especificación de Requisitos definidos para el proyecto.

1. Gestionar Firmas Manuscritas
 - 1.1 Adicionar Firma Manuscrita
 - 1.2 Actualizar Firma Manuscrita
 - 1.3 Consultar Firma Manuscrita
 - 1.4 Consultar Histórico de Firmas Manuscritas
 - 1.5 Modificar Firma Manuscrita
 - 1.6 Cancelar Firma Manuscrita
 - 1.7 Buscar Firmas Manuscritas
 - 1.8 Cargar Firmas Manuscritas
 - 1.9 Comparar Firmas Manuscritas
 - 1.10 Control de Firmas Manuscritas
 - 1.11 Pre-procesar Firmas Manuscritas

2.3 Casos de Uso del Sistema.

Los casos de uso describen la manera en que los actores interactúan con el sistema, para lograr un mejor entendimiento de las necesidades del cliente y una mejor organización del desarrollo de la solución se definieron los siguientes casos de uso:

- Gestionar Firmas Manuscritas
- Buscar Firmas Manuscritas
- Comparar Firmas Manuscritas
- Control de Firmas Manuscritas
- Cargar Firmas Manuscritas
- Pre-procesar Firmas Manuscritas

2.4 Análisis.

En el análisis se utiliza una combinación de texto y de diagramas, para representar los requisitos de datos, funciones y comportamientos, que es relativamente fácil de entender y más importante aún, sencillo para revisar su conexión, completitud y consistencia. El mismo es de gran importancia ya que para validar los requisitos del software se necesita examinarlos desde diferentes puntos de vista y el análisis estructura los requisitos de manera que es más fácil su comprensión, su preparación, su modificación y en general su mantenimiento. A través del análisis se representa una vista interna del sistema en la que, usando el lenguaje de los desarrolladores se refina los requisitos y se estructuran en base a clases y paquetes.

Entre los artefactos que se generan en este importante flujo de trabajo se pueden mencionar:

Modelo de Análisis, Clases del Análisis, Realización de Casos de Uso del Análisis, Descripción de la Arquitectura de los cuales el proyecto SAGEB (Sistema Automatizado de Gestión Bancaria) decide que se realice el Modelo de Análisis.

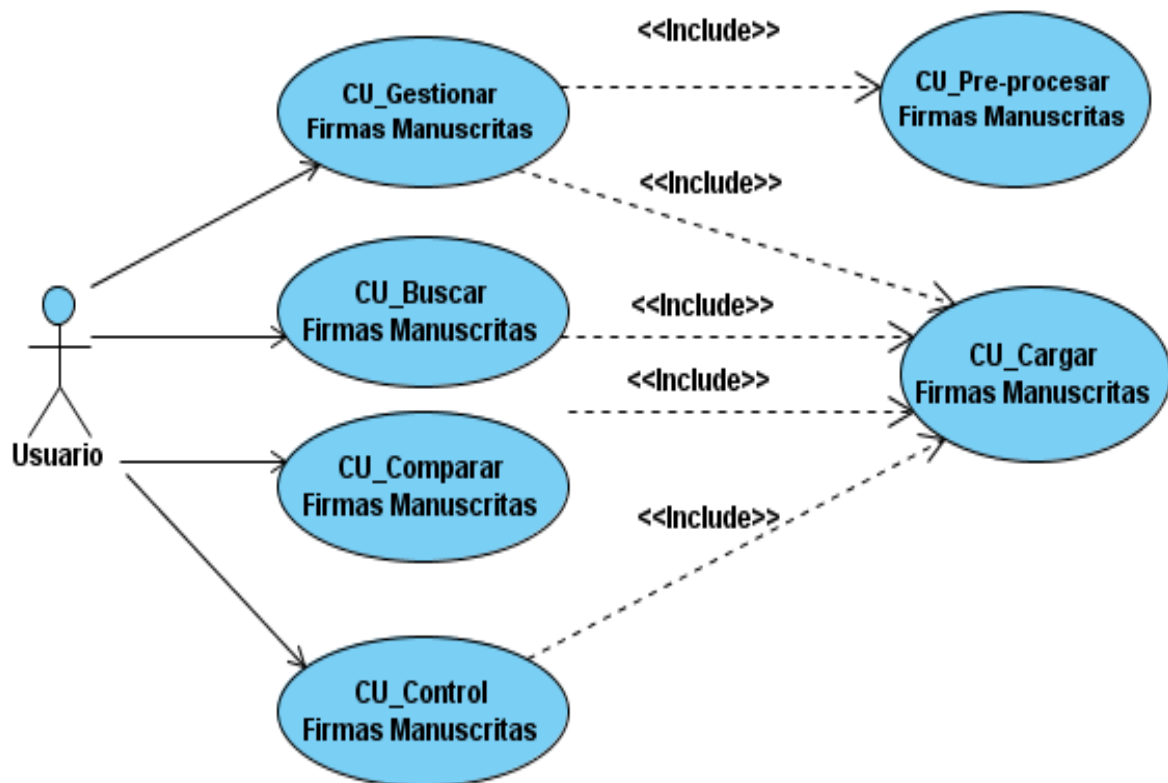


Fig. 1 Diagrama de casos de uso del sistema.

2.5 Modelo de Análisis.

El modelo de análisis debe lograr tres objetivos primarios: describir lo que requiere el cliente, establecer una base para la creación de un diseño de software, y definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

Para realizar la construcción del modelo de análisis primeramente se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis. Las clases que se identifican están asociadas con el contexto del dominio del problema por lo que representan conceptos y relaciones.

Estas clases se clasifican en:

Clases Interfaz: Modelan la interacción entre el sistema y sus actores.

Clases Entidad: Modelan información que posee larga vida y que es a menudo persistente.

Clases de Control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

A continuación se muestra el Modelo de Análisis correspondiente al Módulo Firmas Manuscritas conformado por los diagramas de clases del análisis perteneciente a los casos de uso siguientes:

- Gestionar Firmas Manuscritas
- Buscar Firmas Manuscritas
- Comparar Firmas Manuscritas
- Control de Firmas Manuscritas
- Cargar Firmas Manuscritas
- Pre-procesar Firmas Manuscritas

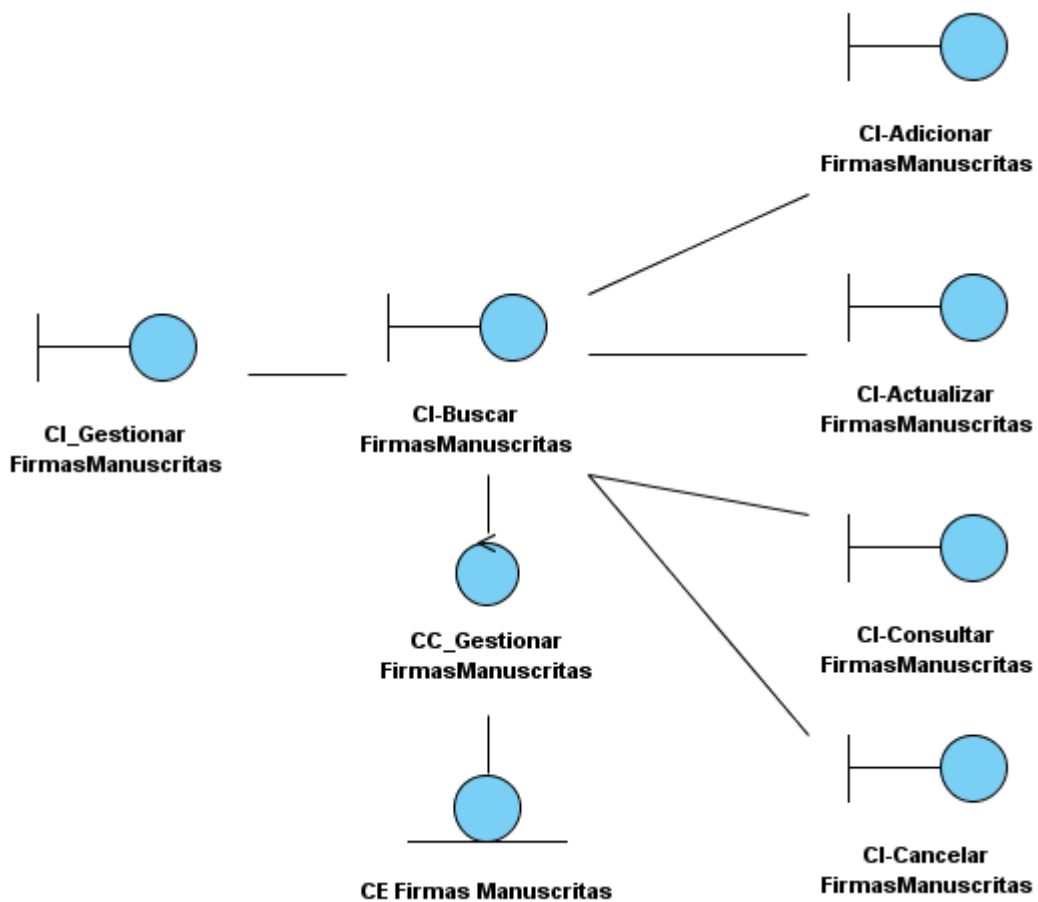


Fig. 2 Modelo de Análisis de los casos de uso Gestionar Firmas Manuscritas.

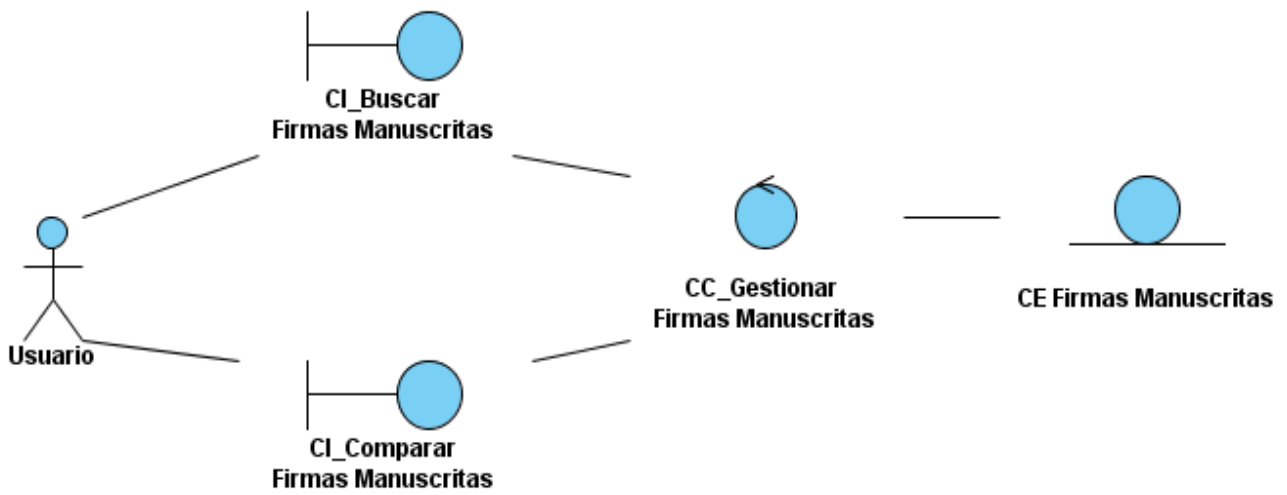


Fig. 3 Diagrama de Clases del Análisis del caso de uso Comparar Firmas Manuscritas.

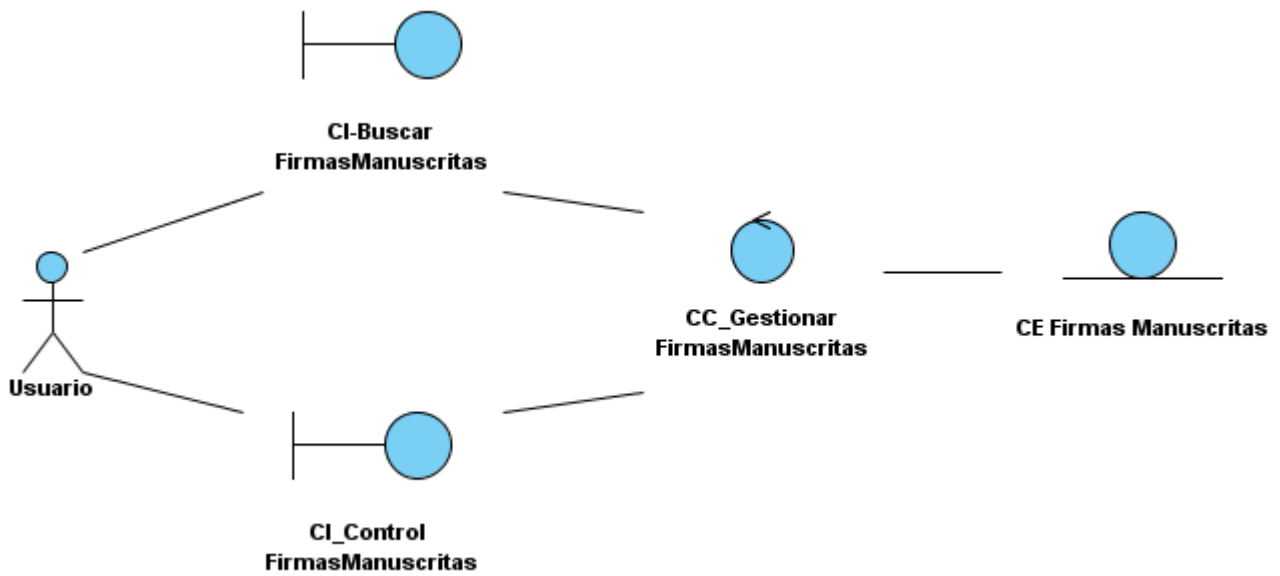


Fig. 4 Diagrama de Clases del Análisis del caso de uso Control Firmas Manuscritas.

Los modelos correspondientes a los demás casos de uso se podrán encontrar en los anexos.

2.6 Conclusiones Parciales.

En este capítulo se mostró una primera visión del Módulo Firmas Manuscritas, mediante la descripción de los casos de uso que se definieron a partir de los requerimientos del software. Además de proporcionar como un primer aporte a la solución el modelo de análisis.

CAPÍTULO 3. DISEÑO DEL SISTEMA.

3.1 Introducción.

En este capítulo se muestran las características que posee la arquitectura que sostiene el desarrollo del Módulo Firmas Manuscritas y una breve panorámica acerca de los patrones de diseño que se utilizan, lo que permite dar paso al diseño donde se realizan los diagramas de clases, diagramas de interacción, así como el modelo de datos.

3.2 Fundamentación de la arquitectura.

La Arquitectura de Software son las técnicas metodológicas desarrolladas con el fin de facilitar la programación, esta se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiar el desarrollo de software dentro de un sistema informático, lo que posibilita que el software sea fácilmente analizable, modificable, corregible, etc.

La arquitectura que se propuso en el proyecto y sobre la que se está realizando el desarrollo del software está compuesta por una arquitectura de tres capas:

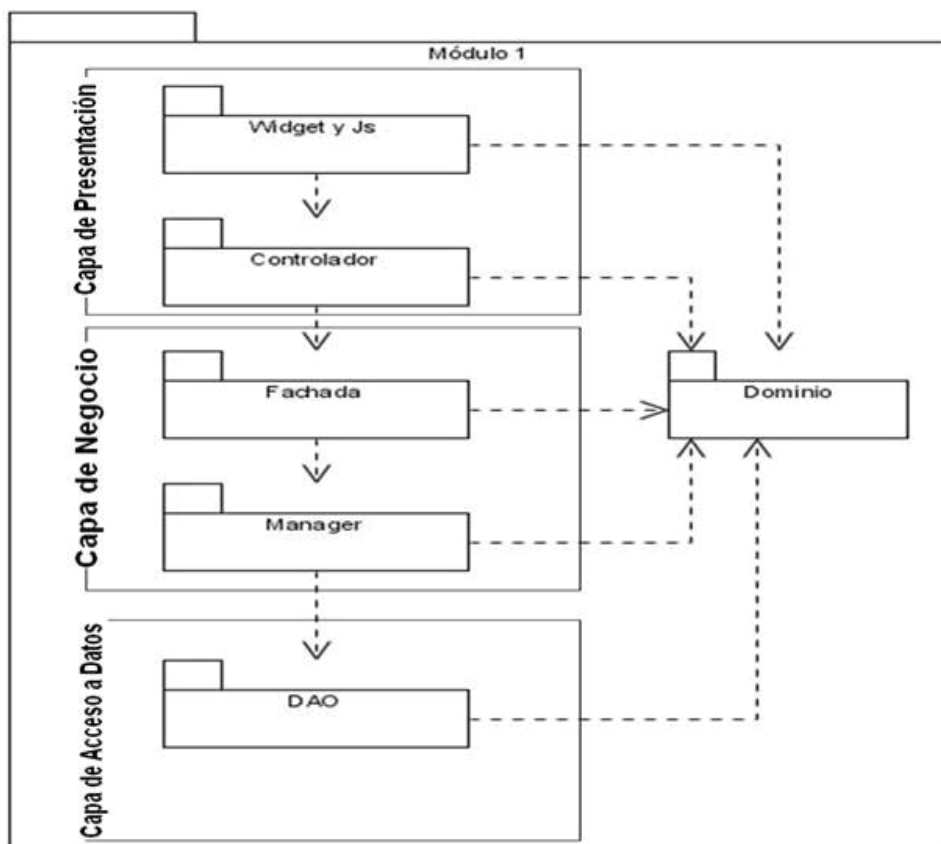


Fig. 1 Arquitectura de un Módulo.

➤ Capa de Presentación.

Esta capa estará dividida en dos partes. Una subcapa del lado del servidor, encargada de recibir todos los pedidos de la interfaz de usuario, controlar el flujo de presentación del sistema y enviar las respuestas correspondientes a la interfaz de usuario. La otra subcapa estará en el cliente, utilizándose los componentes visuales de Java Script para manejar los eventos y validaciones del lado de cliente. La subcapa colocada en el lado del servidor estará relacionada con la capa de Negocios y de Dominio.

➤ Capa de Negocios.

Esta capa está dividida en dos subcapas principales sin dejar de incluir otras que se necesiten y que estén relacionadas con el negocio. En la Fachada se expondrán todas las funcionalidades que la capa de presentación necesitará. Esta capa invocará métodos de la subcapa de desarrollo del negocio. En la capa de desarrollo del negocio se implementará el negocio de los módulos en cuestión, y de aquí se accederá de ser necesario a la Capa de Acceso a Datos, a otras Capas de Negocios y/o a la Capa de Dominio.

➤ Capa de Acceso a Datos.

En esta capa se implementarán los métodos encargados en interactuar con el gestor de Base de Datos. Esta capa tendrá solamente dependencia con la Capa de Dominio.

➤ Capa de Dominio.

En esta capa se declararán todas las clases que representan entidades del negocio. Estas clases de dominio estarán presentes en todas las capas anteriormente descritas.

A continuación se muestra el modelo de paquetes del Módulo Firmas Manuscritas para una mejor comprensión de cómo está estructurado por paquetes dicho módulo.

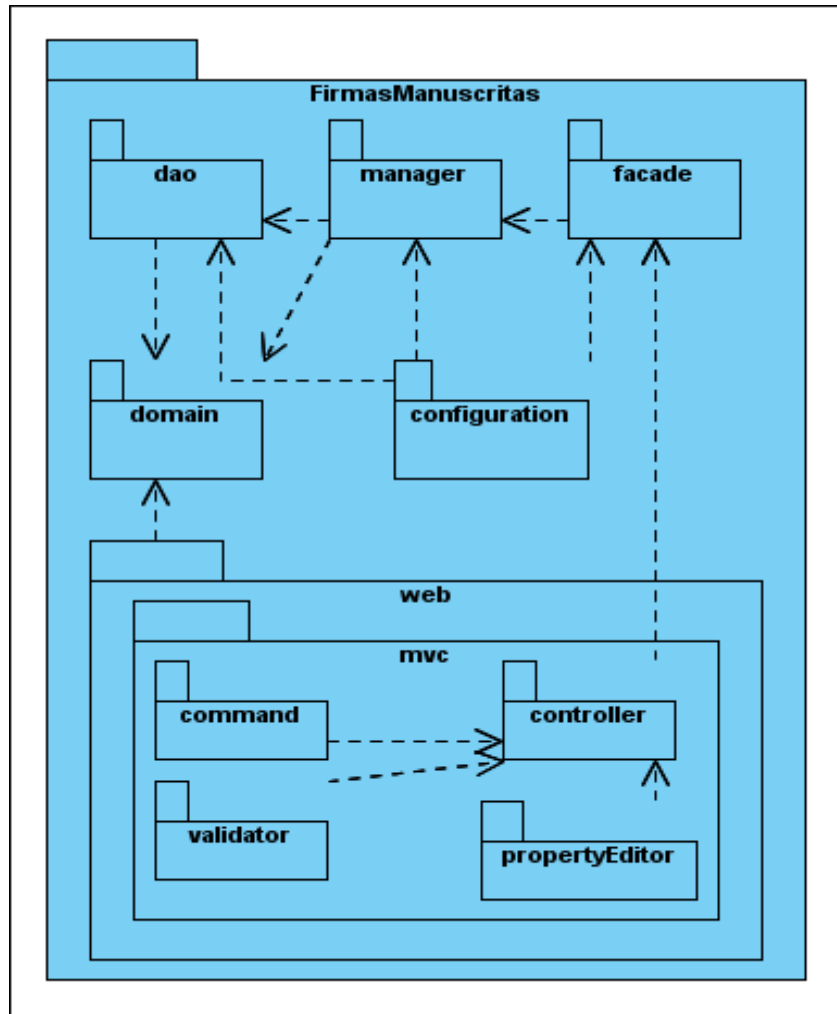


Fig. 2 Modelo de Paquetes del Módulo Firmas Manuscritas.

Organizar por criterios en paquetes las clases y ficheros de un módulo ayuda a la fácil comprensión de la aplicación. En breve se mencionan las clases que componen cada paquete.

Paquete configuration: en este paquete están los ficheros XML de configuración de los diferentes contextos de Spring:

- -dataaccess.xml: contexto de acceso a datos.
- -business.xml: contexto de negocio
- -servlet
- .xml : contexto de Spring

Paquete dao: en este paquete están las interfaces y las clases de implementen el acceso a datos del módulo, además de los ficheros de mapeos de Hibernate.

Paquete manager: en este paquete se encuentran las interfaces e implementaciones del negocio del módulo correspondiente.

Paquete facade: en este paquete se encuentran la interface y la implementación de las funcionalidades que se le brindaran a la presentación.

Paquete domain: en este paquete están las clases del dominio del módulo.

Paquete web: este paquete es el contenedor del paquete mvc.

Paquete mvc: en este están los paquetes que contienen la lógica de presentación en el servidor para SpringMVC.

Paquete commad: contiene clases que representan objetos a manipular en los formularios.

Paquete validator: contiene las clases encargadas de validar los datos.

Paquete propertyEditor: clases para convertir objetos.

Paquete controller: contiene las diferentes clases que heredan de los controladores de Spring

3.3 Diseño.

El diseño es un proceso iterativo mediante el cual los requisitos se traducen en un plano para construir el software, el cual se encuentra en el núcleo técnico de la ingeniería del software. La importancia del diseño se puede traducir en una sola palabra, calidad, ya que sin este se corre el riesgo de construir un sistema inestable, que fallará cuando se lleven a cabo cambios; que pueden resultar difícil de comprobar; y cuya calidad no puede evaluarse hasta muy avanzado el proceso.

El diseño posee los siguientes propósitos:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando se utilizan interfaces como elementos de sincronización entre diferentes equipos de desarrollo. (Pressman, 1997)

3.3.1 Diagramas de clases del 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.

Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación. (Jacobson, 2000.)

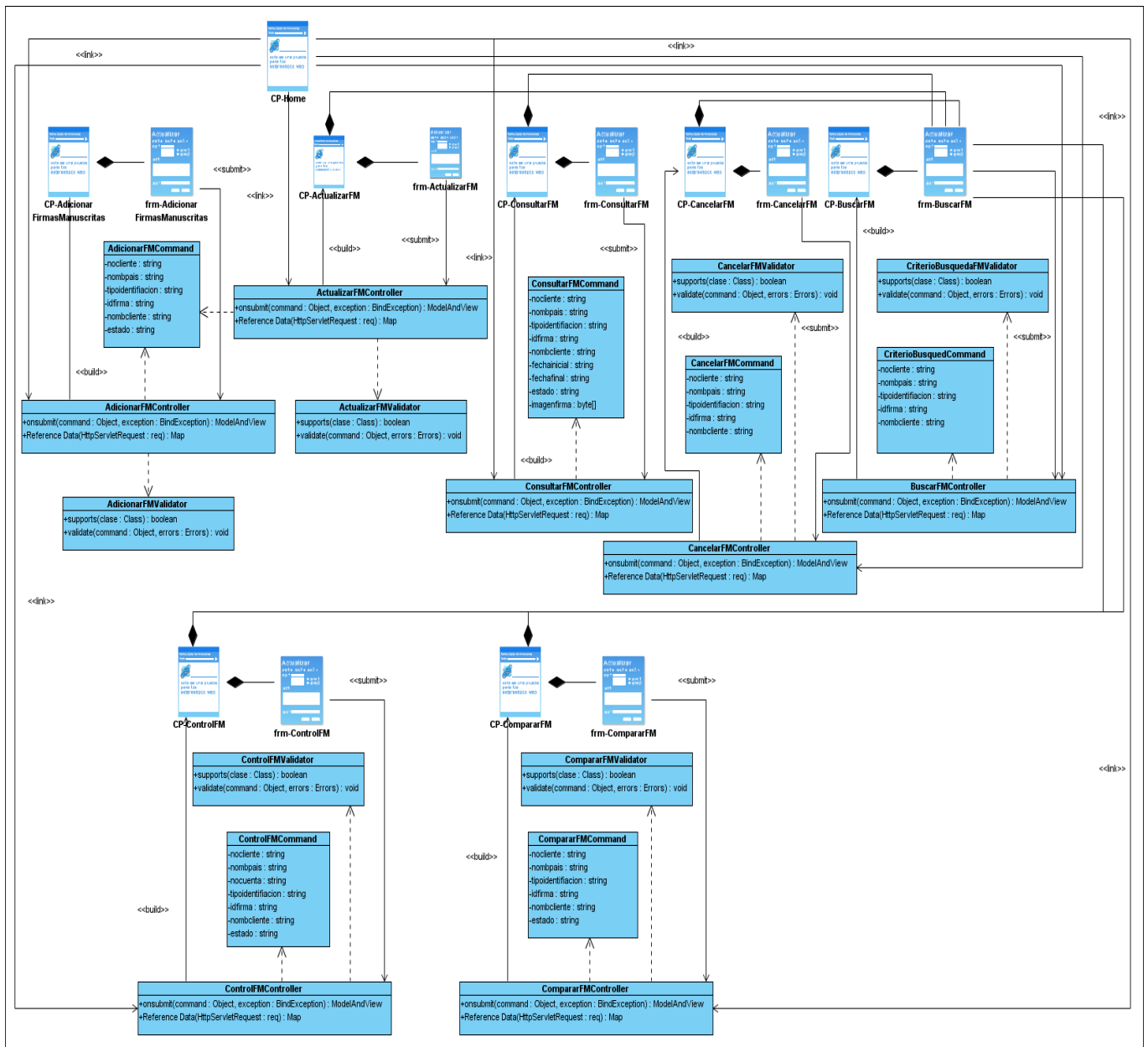


Figura 3. Diagrama de clases de la Capa de Presentación del Módulo Firmas Manuscritas.

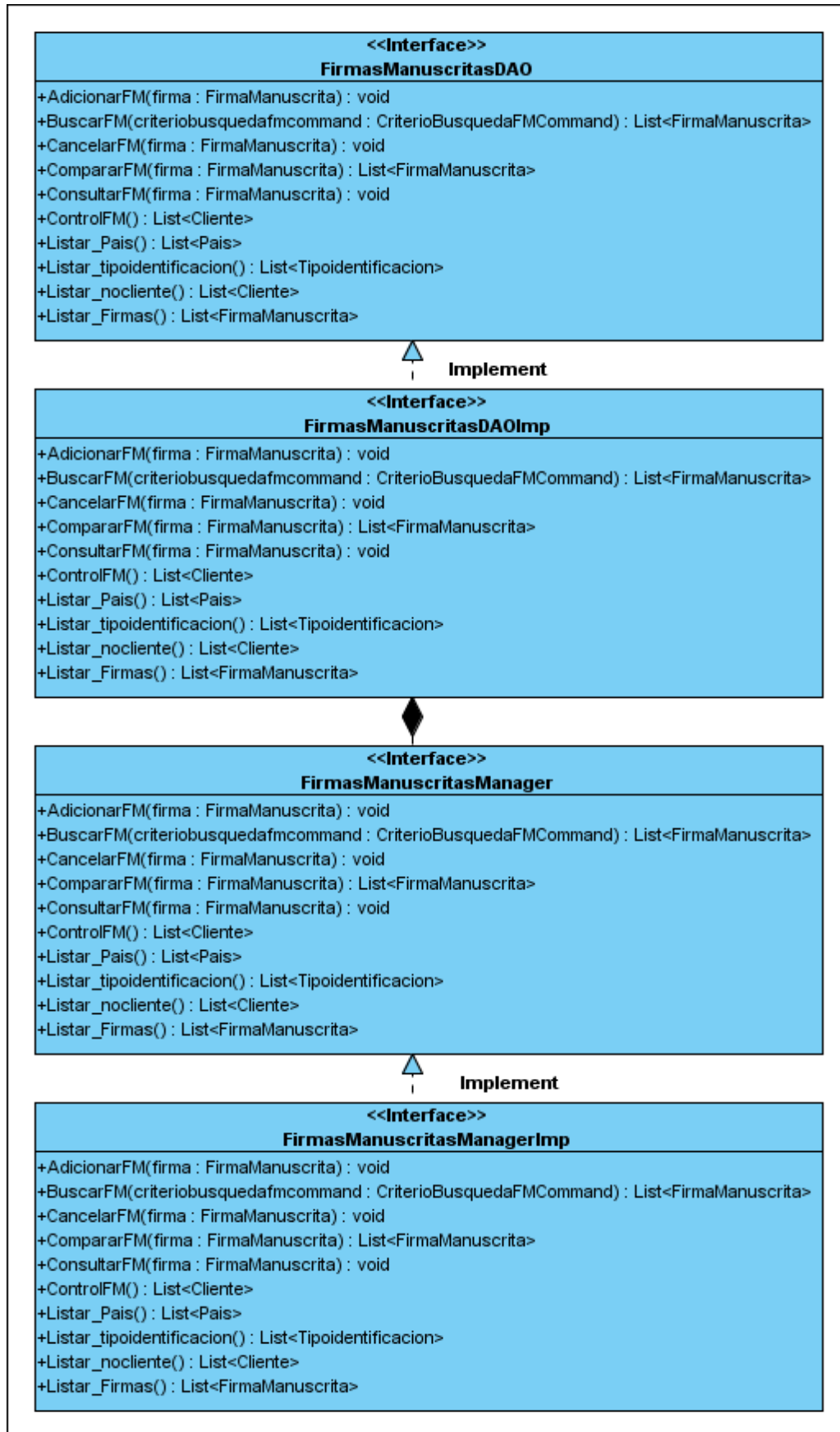


Figura 4. Diagrama de clases de la Capa de Negocio del Módulo Firmas Manuscritas.

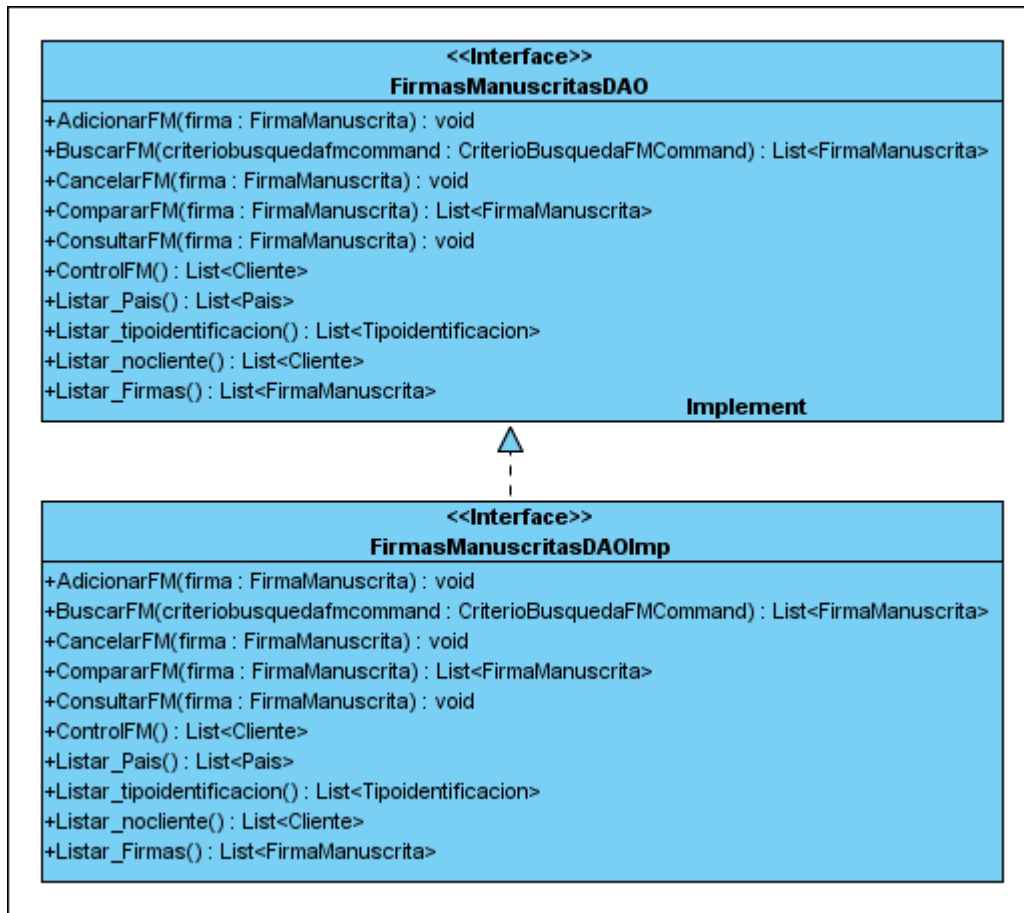


Figura 5. Diagrama de clases de la Capa de Acceso a Datos del Módulo Firmas Manuscritas.

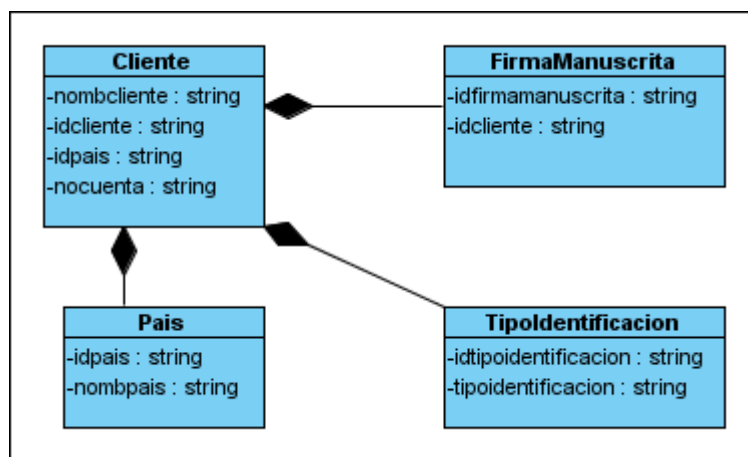


Figura 6. Diagrama de clases de la Capa de Dominio del Módulo Firmas Manuscritas.

3.3.2 Diagrama de interacción del diseño.

En el flujo de diseño se utiliza fundamentalmente el diagrama de secuencia. En él se incluyen las interacciones entre las clases del diseño, a través de mensajes. Un mensaje significa una operación en la clase a la que va el mensaje. El objetivo de realizar diagramas de secuencia hace que se tenga en cuenta el orden y el momento en que se envían los mensajes a los objetos. Los mensajes que se envían entre las clases pueden ser síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada.

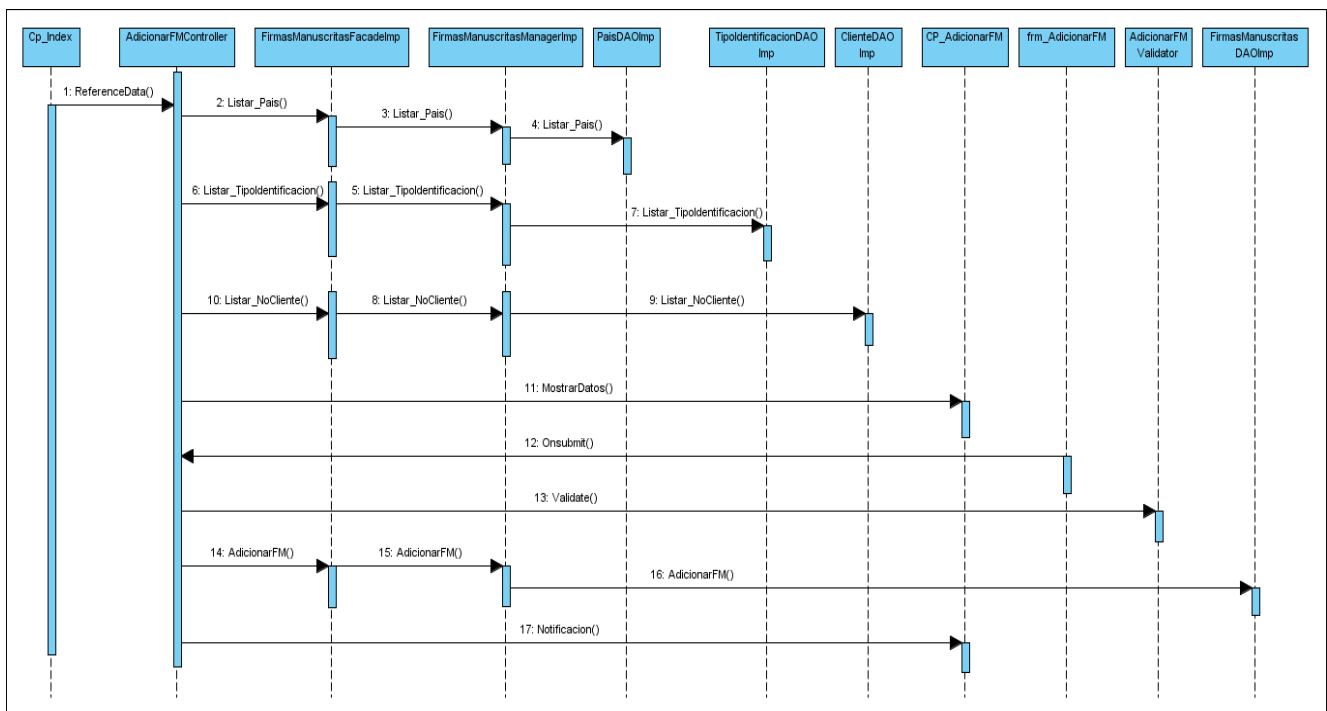


Figura 7. Diagrama de Secuencia del Caso Uso Adicionar Firmas Manuscritas.

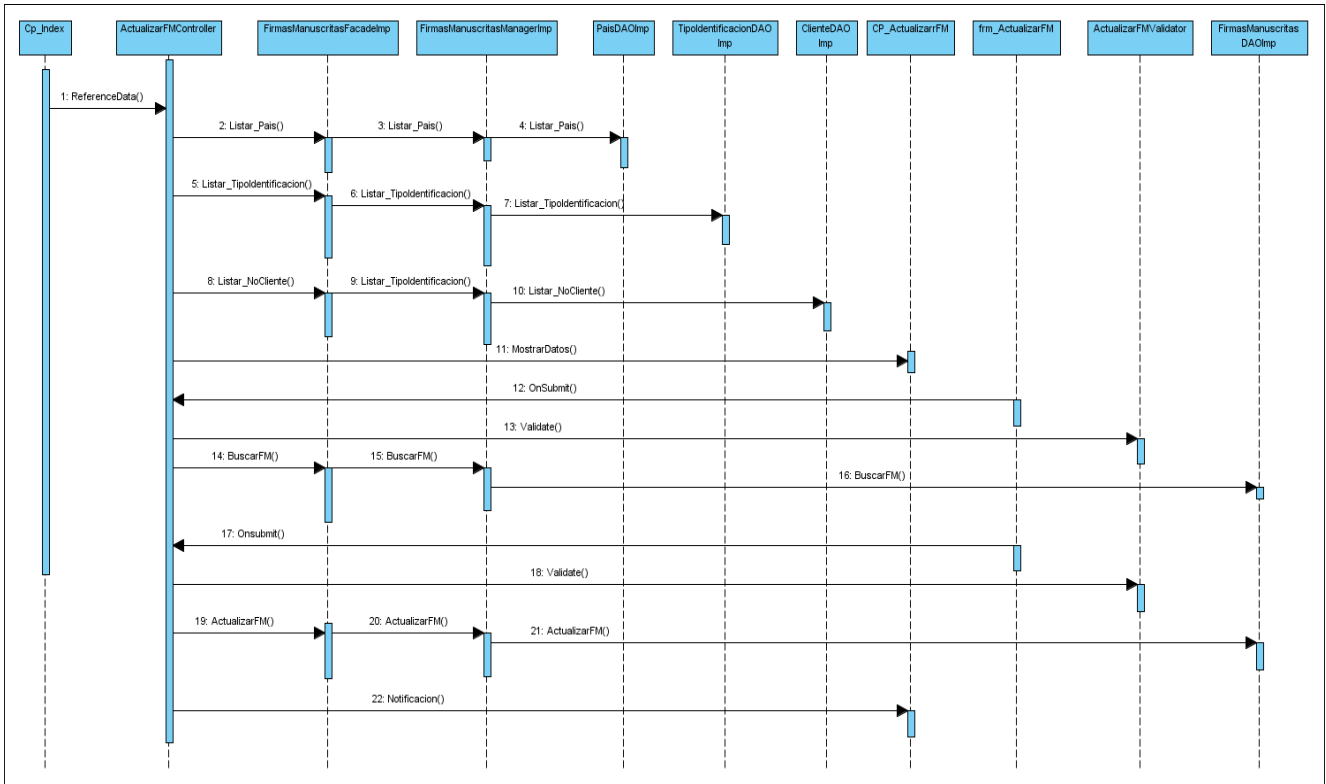


Figura 8. Diagrama de Secuencia del Caso Uso Actualizar Firmas Manuscritas.

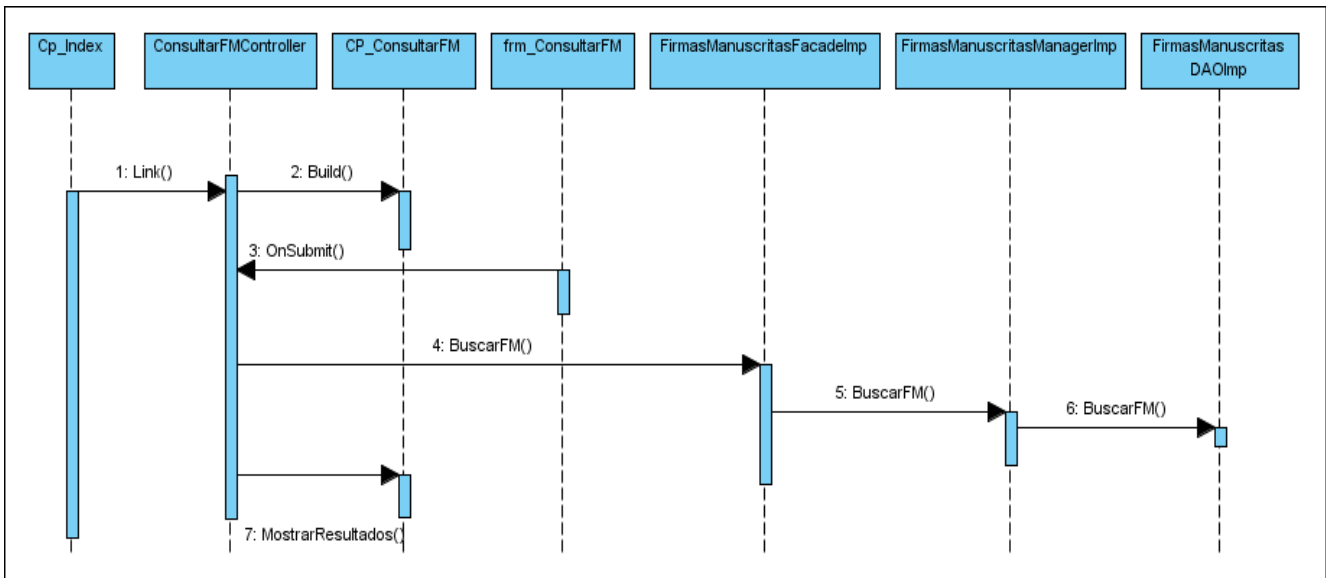


Figura 9. Diagrama de Secuencia del Caso Uso Consultar Firmas Manuscritas.

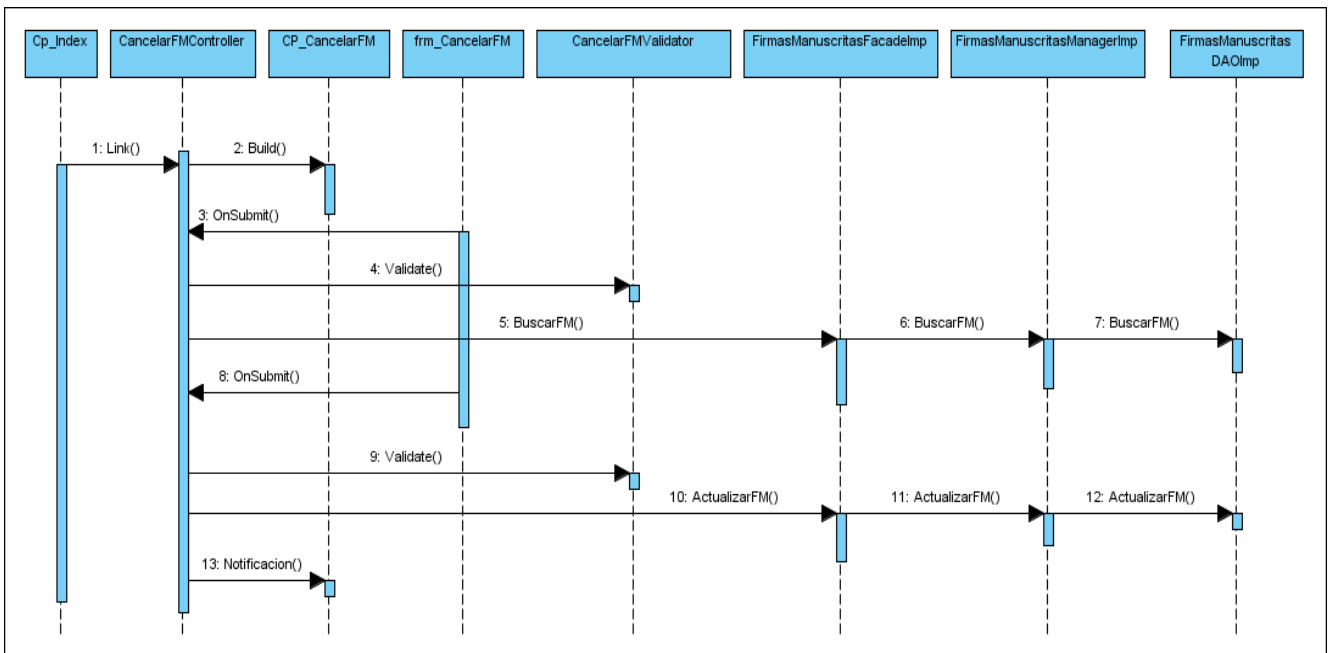


Figura 10. Diagrama de Secuencia del Caso Uso Cancelar Firmas Manuscritas.

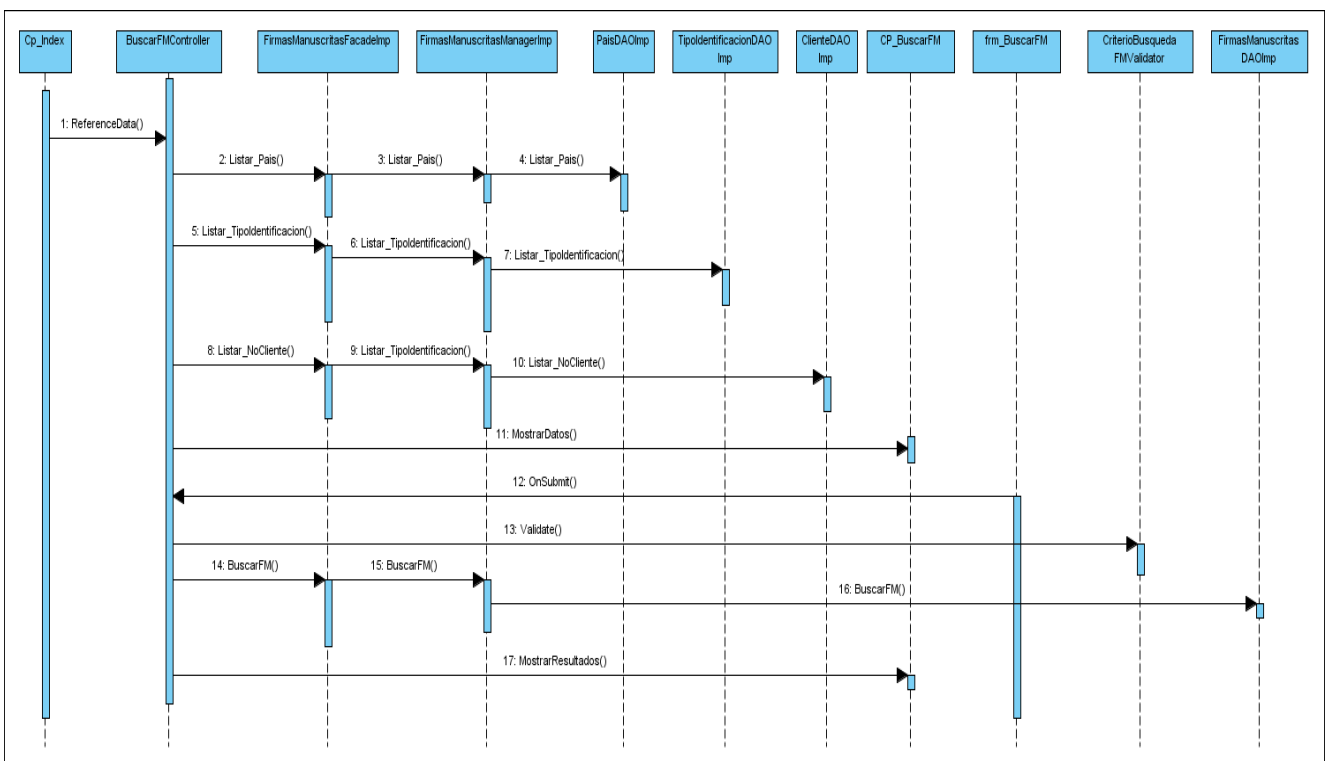


Fig. 11 Diagrama de Secuencia del Caso Uso Buscar Firmas Manuscritas.

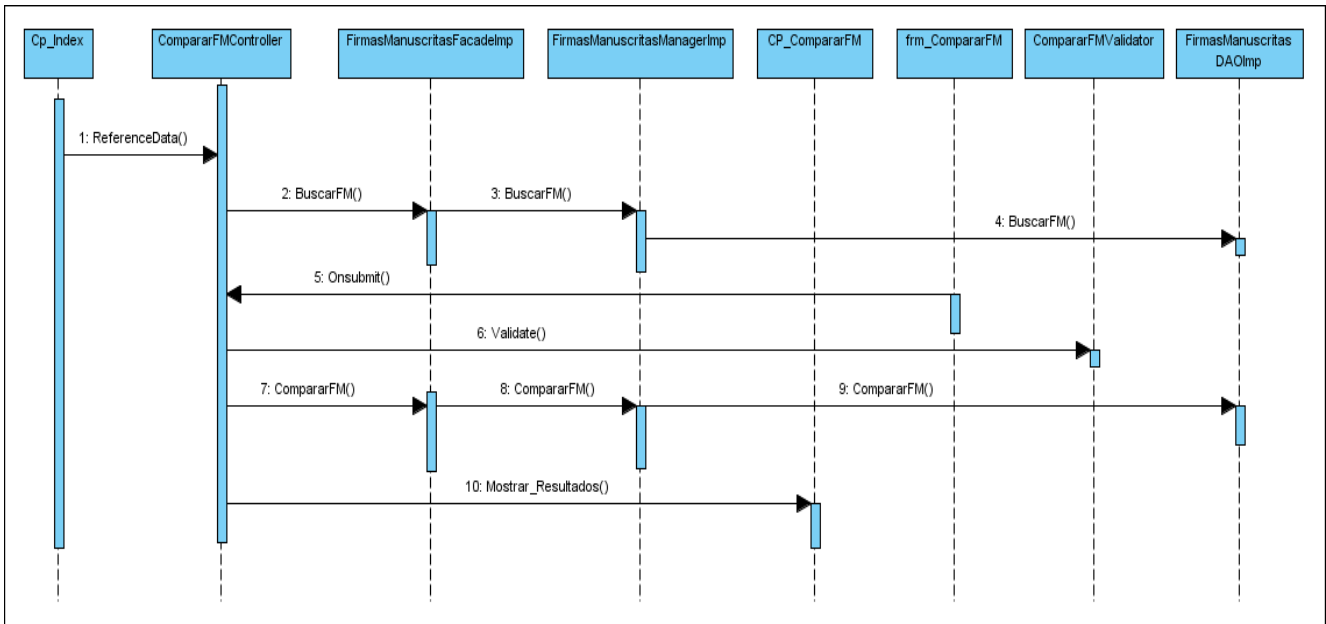


Fig. 12 Diagrama de Secuencia del Caso Uso Comparar Firmas Manuscritas.

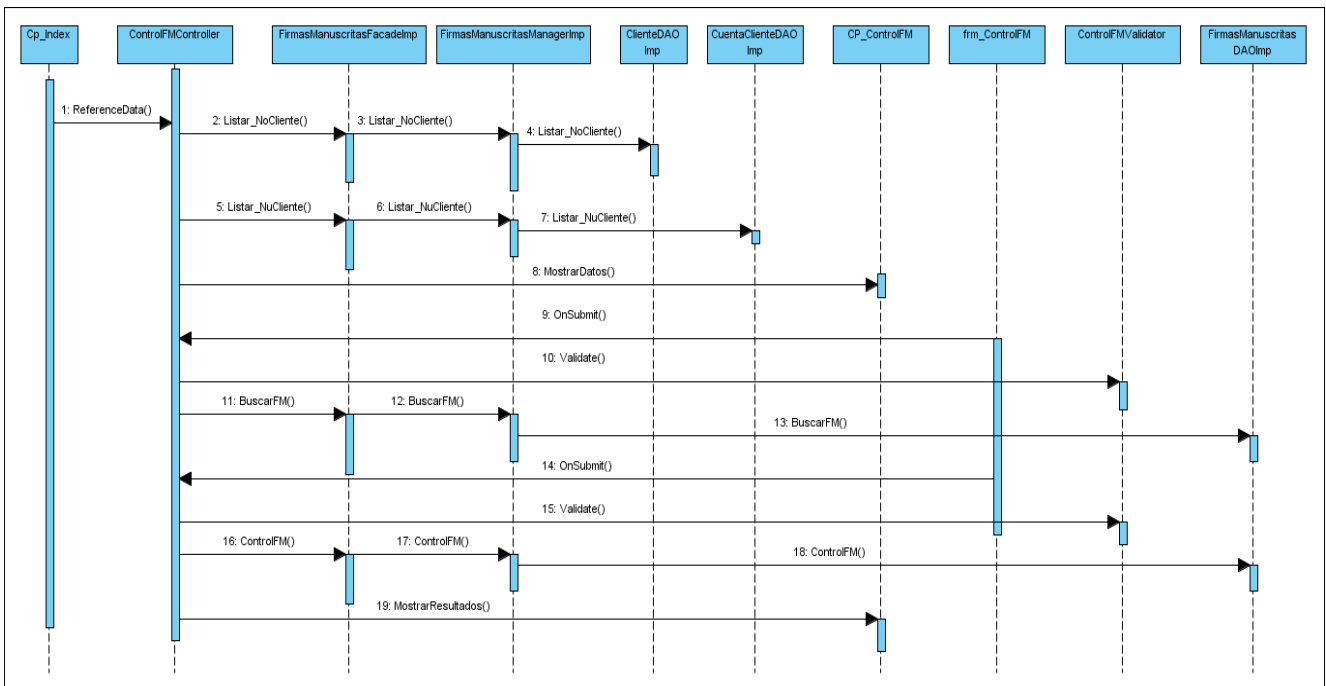


Fig. 13 Diagrama de Secuencia del Caso Uso Control Firmas Manuscritas.

3.4 Aplicación de los Patrones.

Para realizar un diseño más eficiente es conveniente utilizar uno o varios patrones de este conjunto, que al ser experiencias de diseñadores expertos en orientación a objetos permiten dar solución a problemas a través de la codificación del conocimiento y principios existentes, facilitando notablemente el trabajo posterior. Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema.

Los patrones más utilizados en actualidad son: el patrón arquitectónico Modelo – Vista – Controlador (MVC), los patrones de diseño, GRASP para la asignación de responsabilidades y patrones de comportamiento, entre otros. (Larman, 1999)

3.4.1 Patrón Arquitectónico Modelo Vista Controlador (MVC).

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio.

Descripción del patrón:

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.



3.4.2 Patrones GOF.

➤ Adaptador

El patrón Adaptador se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda.

Convierte la interfaz de una clase en otra interfaz que el cliente espera. Adaptador permite a las clases trabajar juntas, lo que de otra manera no podría hacerlo debido a sus interfaces incompatibles.

➤ Factoría

Consiste en utilizar una clase constructora (al estilo del Abstract Factory) abstracta con unos cuantos métodos definidos y otro(s) abstracto(s): el dedicado a la construcción de objetos de un subtipo de un tipo determinado. Es una simplificación del Abstract Factory, en la que la clase abstracta tiene métodos concretos que usan algunos de los abstractos; según usemos una u otra hija de esta clase abstracta, tendremos uno u otro comportamiento.

Abstract Factory: Nos da una interface para crear objetos de alguna familia, sin especificar la clase en concreto.

➤ Singleton

El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una. En este caso, si más de un objeto necesita utilizar una instancia de la clase Singleton, esos objetos comparten la misma instancia de la clase Singleton. En un uso más avanzado, este patrón puede ser utilizado también para administrar *n* instancias de una clase.

➤ Compositor

El patrón compositor es un patrón de diseño muy poderoso en sí mismo, consiste en crear objetos a partir de otros más pequeños, también puede contener objetos que tienen otros objetos, dando el aspecto de un árbol en anillo.

➤ Observador

El patrón Observador también conocido como "spider" define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes.

El objetivo de este patrón es desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje, así como evitar bucles de actualización (espera activa o polling).

➤ Estrategia

Cuando tenemos un algoritmo que puede cambiar, por cualquier motivo, bien en tiempo de ejecución, bien cuando lo implementemos en otro sitio, y no queremos modificar el código que usa ese algoritmo, viene bien usar el patrón estrategia.

Básicamente, consiste en hacer una interface con los métodos del algoritmo. Luego implementamos las distintas versiones del algoritmo en clases que implementen esa interface.

Teniendo en cuenta estos patrones, se recomienda como patrón a utilizar el patrón **Estrategia**.

3.4.3 Patrones de Diseño.

Los patrones de diseño describen un problema que ocurre repetidas veces en algún contexto determinado de desarrollo de software, y entregan una buena solución ya probada. Esto ayuda a diseñar correctamente en menos tiempo, a construir problemas reutilizables y extensibles, facilita la documentación, y facilita la comunicación entre los miembros del equipo de desarrollo. (Larman, 1999)

- Descripciones de clases cuyas instancias colaboran entre sí que deben ser adaptados para resolver problemas de diseño generales en un contexto particular.
- Un patrón de diseño identifica: Clases, Instancias, Roles, Colaboraciones y la Distribución de responsabilidades.

3.4.4 Patrón Estructural.

➤ Fachada.

Fachada provee una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. El uso de este patrón está recomendado para simplificar uso y comprensión de una librería software, encapsular una interfaz de librería poco amigable en una interfaz más coherente o mejor estructurada, además de centralizar las dependencias externas hacia la librería en uno o pocos puntos de entrada.

3.4.5 Patrón de Acceso a Datos.

➤ Objeto de Acceso a Datos (DAO)

Este patrón tiene como principal objetivo abstraer y encapsular todos los accesos a la fuente de datos DAO, este oculta completamente los detalles de implementación de la fuente de datos a sus clientes y su interface no se modifica cuando cambia la implementación de la fuente de datos subyacente.

3.4.6 Patrones de Asignación de Responsabilidades (GRASP).

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

➤ Experto

La aplicación del patrón Experto consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Dicha aplicación permite conservar el encapsulamiento; soportar un bajo acoplamiento, lo que posibilita tener sistemas más robustos y de fácil mantenimiento; además de que las clases son más sencillas y más cohesivas. (Larman, 1999)

➤ Creador

Permite crear instancias de otras clases según la responsabilidad de la misma. Esto posibilita un bajo acoplamiento, mejores oportunidades de reutilización etc. (Larman, 1999)

➤ Bajo Acoplamiento

Este patrón soluciona el problema de lograr una dependencia escasa y un aumento de la reutilización, asignando responsabilidades a las clases de tal forma que la dependencia entre las mismas sea la menor posible. (Larman, 1999)

➤ Alta Cohesión

La utilización de este patrón permite que las clases contengan las responsabilidades estrechamente relacionadas, sin tener que realizar un trabajo excesivo, posibilitando la claridad y facilidad con que se entiende el diseño, simplificando el mantenimiento y las mejoras en funcionalidad, logrando en ocasiones un bajo acoplamiento. (Larman, 1999)

➤ Controlador

La aplicación del patrón Controlador consiste en asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente: el negocio, la organización global, o el sistema global, etc. (Larman, 1999)

3.5 Conclusiones parciales.

En este capítulo además de una breve panorámica acerca de los patrones de diseño y de la arquitectura que sostiene la solución propuesta, se mostró el diseño del Módulo Firmas Manuscritas, el cual se realizó de forma tal que corresponda con la arquitectura y que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. De esta forma quedó conformado el modelo de diseño realizado en este capítulo donde su principal objetivo era realizar los diagramas de clases del diseño, describir las clases y realizar los diagramas de interacción correspondiente, por lo que queda hecha la propuesta del sistema permitiendo dar paso a la realización posterior de la implementación.

CONCLUSIONES

Como resultado de la investigación realizada en el presente trabajo se logró una mejor comprensión sobre la situación de los procesos de verificación de Firmas Manuscritas de las entidades bancarias a nivel nacional e internacional, se dejaron plasmados los estándares y sistemas automatizados que son utilizados en ambos niveles. Además el estudio realizado sobre las metodologías, herramientas y lenguajes más utilizados en el mundo permitió un correcto análisis y diseño para el Módulo Firmas Manuscritas.

A partir de los casos de uso propuestos, las descripciones y los prototipos no funcionales correspondientes a cada requerimiento, se definieron las clases del análisis, propiciándose así una mejor comprensión de las necesidades del cliente para realizar el diseño, lo que permitió la correcta realización del mismo.

Entre los artefactos que se generaron como parte de la solución se pueden mencionar fundamentalmente: el Modelo de Análisis y el Modelo de Diseño los que permiten que se encuentre lista la entrada al flujo de Implementación y de esta forma lograr la realización del software.

RECOMENDACIONES

Como parte de la investigación realizada en el presente trabajo de diploma se recomienda:

- Continuar con la implementación del software a partir de los artefactos generados como parte de la solución del trabajo.
- Continuar el estudio del tema con el objetivo de encontrar nuevas funcionalidades para futuras versiones de la aplicación.
- Obtener una versión del sistema que sea genérica de manera que pueda ser utilizada por cualquier banco o entidad bancaria mundial.

BIBLIOGRAFÍA

- BCC.2010.** Banco Central de Cuba. [En línea] 2010. [Citado el: 10 de Enero de 2010.]
<http://www.bc.gov.cu/Espanol/default.asp>.
- CMHW.2010.** [En línea] [Citado el: 15 de Febrero de 2010.]
<http://www.cmhw.co.cu/noticia.asp?aid=11125>.
- Ferrer, Miguel A., Travieso, Carlos M. y Vargas, Francisco. 2008.** Aplicación del Kernel de Fisher para la verificación de Firmas Manuscritas. [En línea] 2008. [Citado el: 15 de Junio de 2010.]
- Miguel A., Ferrer, y otros. 2008.** *Verificación Off-Line de Firmas Manuscritas: Una Propuesta basada en Snakes Paramétricos.* [En línea] 2008.
- Jacobson, Ivar. 2000.** *El proceso Unificado de desarrollo de software. Vol I.* s.l. : Félix Varela, 2000.
- Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2007.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Pearson Educación : s.n., 2007.
- Pressman, Roger S. 1997.** *Ingeniería del Software: Un enfoque práctico. Cuarta edición.* Madrid : McGraw-Hill, 1997.
- Universidad de Deusto, 2008.** [En línea] 2008. [Citado el: 15 de Marzo de 2010.]
http://www.eside.deusto.es/grupos/eside_pas.
- Larman, Craig.1999** *UML y Patrones. Mexico* . 1999. 970-17-0261-1.
- Alambre.2010.**[En línea] [Citado el: 20 de Febrero de 2010.]
<http://www.alambre.info/2003/12/08/origenes-de-la-firma-autografa/>.
- Pimentel., Luis Alberto.2008** *Herramientas de Desarrollo.* 2008.
- Tormo, Carmen Cantero,2008.** *Software de reconocimiento y verificación de firmas manuscritas digitalizadas.* Barcelona : Universidad Ramon Llull.
- Visual Paradigm.2010.** [En línea] [Citado el: 10 de febrero de 2010.]
<http://www.visual-paradigm.com>.
- León, Fernando y Cano García, Welicki.2007** *Metodologías de la Ingeniería en Software.* s.l. : Electrocast Team, 2007.
- Apuntes.2010** [En línea] [Citado el: 10 de Mayo de 2010.]
<http://www.apuntes.delibertad.com/.../patron-composite/>.
- BC.2010** [En línea] [Citado el: 15 de Mayo de 2010.]
http://www.bc.gov.cu/Espanol/bancos_comerciales.asp#BICSA.
- ARMENTEROS., MERCEDES GARCÍA.** El Sistema Financiero Cubano una víctima del bloqueo. [En línea] .[Citado el: 13 de enero de 2010.]. [En línea] [Citado el: 24 de marzo de 2010.] <http://edicionesanteriores.trabajadores.cu>.
- IBM.2008** [En línea] 2008. [Citado el: 10 de Enero de 2010.] <http://www.ibm.com>
- D. Rojas P., Bernardo y otros.** Una Introducción. [PDF]. s.l. : Banca Centra, 2003. I.

Eslova.2010 [En línea] [Citado el: 10 de Mayo de 2010.]

<http://www.esolva.com/firmas/index.html>.

Muñoz., Javier Martínez.2008 *Definición De Los Requerimientos Funcionales Del Módulo Cuentas de Clientes y Órdenes de Pago Inmediato Del Proyecto Banco Nacional*. Ciudad Habana : s.n., 2008.

Walls, Craig y Breidenbach, Ryan.2005 *Spring in Action*. Manning Publications Co., 2005.

Computación.2010 [En línea] [Citado el: 18 de Febreo de 2010.]

http://computacion.cs.cinvestav.mx/~jjangel/Pagina_Todos_es.html

Informática.2010 [En línea] [Citado el: 20 de Febreo de 2010.]

http://www.informatica-juridica.com/trabajos/firma_digital.asp

Bauer, Christian y King, Gavin.2004 *Hibernate in Action*. 2004.

Kruchten, P. 1998. *The Rational Unified Process*. 1998.

IGP SQLServer.2010 Información General del Producto SQLServer 2005 [En línea]

<http://www.microsoft.com/spain/sql/productinfo/overview/default.mspx>

Bello, Perdomo Yesenia.2008 *Análisis y Diseño del Subsistema Títulos valores del Proyecto Modernización del Sistema Bancario Cubano*. Ciudad Habana : s.n., 2008.

ANEXOS

Descripciones expandidas de los casos de uso.

Caso de Uso	Gestionar Firmas Manuscritas
Actor	Usuario
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción Gestionar Firmas Manuscritas. El sistema registra, actualiza, consulta, cancela y permite realizar búsquedas de Firmas Manuscritas. El caso de uso termina cuando el usuario acepta realizar la operación.
Precondiciones	
Pos condiciones	
Referencia	RF1.1, RF1.2, RF1.3, RF1.4, RF1.5, RF1.6, RF 1.7, RF 1.8 RF 1.9, RF 1.10, RF 1.11
Prioridad	Crítico
Casos de Uso Asociado	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Usuario puede seleccionar una de las opciones siguientes:</p> <ul style="list-style-type: none"> a) Adicionar Firma Manuscrita. Ver Sección “Adicionar Firma Manuscrita”. b) Actualizar Firmas Manuscritas Ver Sección “Actualizar Firma Manuscrita”. c) Consultar Firma Manuscrita Ver Sección “Consultar Firma Manuscrita”. d) Consultar Histórico de Firmas Manuscritas Ver Sección “Consultar Histórico de Firmas Manuscritas”. 	
	2. Finaliza el caso de uso.

Sección “Adicionar Firma Manuscrita”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra la interfaz para registrar una Firma Manuscrita.
4. El Usuario introduce los datos de la Firma Manuscrita.	5. El sistema valida los datos. a. En caso de datos incorrectos. Ver Sección “Datos Incorrectos”. b. Si la Firma Manuscrita existe en el sistema. Ver sección “Firma Manuscrita Existente”.
	6. El sistema registra los datos de una Firma Manuscrita.
	7. El sistema muestra las opciones de registrar, actualizar, consultar y cancelar una Firma Manuscrita.
8. El Usuario pasa a la acción 3 del flujo básico de eventos.	

Sección “Actualizar Firma Manuscrita”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra la interfaz para actualizar una Firma Manuscrita.
4. El Usuario selecciona e introduce los datos para actualizar la Firma Manuscrita.	5. El sistema valida que los datos sean correctos. a. En caso de datos incorrectos. Ver Sección “Datos Incorrectos”.
	6. El sistema actualiza la Firma Manuscrita
	7. El sistema muestra las opciones de adicionar, actualizar, consultar y cancelar una Firma Manuscrita.
8. El Usuario pasa a la acción 1 del flujo básico de eventos.	

Sección “Consultar Firma Manuscrita”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra la interfaz para consultar una Firma Manuscrita.
4. El Usuario selecciona e introduce los datos necesarios para consultar la Firma Manuscrita.	5. El sistema muestra la Firma Manuscrita.
6. El Usuario consulta la Firma Manuscrita.	7. El sistema muestra las opciones de adicionar, actualizar, consultar y cancelar una Firma Manuscrita.
8. El Usuario pasa a la acción 1 del flujo básico de eventos.	

Sección “Consultar Historial Firma Manuscrita”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra la interfaz para consultar el historial de Firmas Manuscritas.
4. El Usuario selecciona e introduce los datos necesarios para consultar el historial de Firmas Manuscritas.	5. El sistema muestra los datos del historial de las Firmas Manuscrita.
6. El Usuario consulta los datos del historial de las Firmas Manuscritas.	7. El sistema muestra las opciones de adicionar, actualizar y consultar una Firma Manuscrita.
8. El Usuario pasa a la acción 1 del flujo básico de eventos.	

Sección “ Modificar Firma Manuscrita ”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra las interfaces para modificar una Firma Manuscrita
4. El Usuario selecciona la opción para modificar una Firma Manuscrita.	5. El sistema modifica la Firma Manuscrita.
	6. El sistema muestra las opciones de adicionar, actualizar, consultar una Firma Manuscrita.
7. El Usuario pasa a la acción 1 del flujo básico de eventos.	

Sección “ Cancelar Firma Manuscrita ”	
Acción del Actor	Respuesta del Sistema
	3. El sistema muestra la interfaz para actualizar una Firma Manuscrita.
4. El Usuario selecciona e introduce los datos para cancelar una Firma Manuscrita.	5. El sistema valida los datos. a. En caso de datos incorrectos. Ver Sección “Datos Incorrectos”.
	6. El sistema cancela la Firma Manuscrita.
	7. El sistema muestra las opciones de adicionar, actualizar y consultar una Firma Manuscrita.
8. El Usuario pasa a la acción 1 del flujo básico de eventos.	

Caso de Uso	Buscar Firmas Manuscritas	
Actor	Usuario	
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción de buscar Firmas Manuscritas. El caso de uso termina cuando se muestran los resultados de la búsqueda.	
Precondiciones		
Poscondiciones		
Referencia	RF 1.7	
Prioridad	Crítico	
Casos de uso asociados	Caso de Uso Gestionar Firmas Manuscritas	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra la interfaz para buscar una Firma Manuscrita.	
2. El Usuario selecciona e introduce los datos para buscar la Firma Manuscrita. Todos los campos son obligatorios.	3. El sistema valida los datos. En caso de datos incorrectos. Ver Sección "Datos Incorrectos".	
	4. El sistema muestra los resultados de la búsqueda de la Firma Manuscrita.	
	5. El sistema muestra las opciones de Gestionar, Buscar, Comparar y Controlar las Firmas Manuscrita.	

Caso de Uso	Cargar Firmas Manuscritas	
Actor	Usuario	
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción de cargar una Firma Manuscrita. El caso de uso termina cuando se muestra la Firma Manuscrita cargada.	
Precondiciones		
Poscondiciones		
Referencia	RF 1.8	
Prioridad	Crítico	
Casos de uso asociados	Caso de Uso Gestionar Firmas Manuscritas	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra la interfaz para cargar una Firma Manuscrita.	
2. El Usuario carga la Firma Manuscrita.	3. El sistema valida los datos. En caso de datos incorrectos. Ver Sección "Datos Incorrectos".	
	4. El sistema carga la Firma Manuscrita.	
	5. El sistema muestra las opciones de Gestionar, Buscar, Comparar y Controlar las Firmas Manuscrita.	

Caso de Uso	Comparar Firmas Manuscritas	
Actor	Usuario	
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción de comparar Firmas Manuscritas. El caso de uso termina cuando se muestran los resultados de la comparación.	
Precondiciones		
Poscondiciones		
Referencia	RF 1.9	
Prioridad	Crítico	
Casos de uso asociados	Caso de Uso Gestionar Firmas Manuscritas	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra la interfaz para comparar una Firma Manuscrita.	
2. El Usuario carga la Firma Manuscrita.	3. El sistema valida los datos. En caso de datos incorrectos. Ver Sección "Datos Incorrectos".	
	4. El sistema muestra los resultados de la comparación de la Firma Manuscrita.	
	5. El sistema muestra las opciones de Gestionar, Buscar, Comparar y Controlar las Firmas Manuscrita.	

Caso de Uso	Control de Firmas Manuscritas	
Actor	Usuario	
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción de control de Firmas Manuscritas. El caso de uso termina cuando se muestran los resultados de controlar la Firma Manuscrita.	
Precondiciones		
Poscondiciones		
Referencia	RF 1.10	
Prioridad	Crítico	
Casos de uso asociados	Caso de Uso Gestionar Firmas Manuscritas	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra la interfaz para controlar una Firma Manuscrita.	
2. El Usuario selecciona e introduce los datos de la Firma Manuscrita.	3. El sistema valida los datos. En caso de datos incorrectos. Ver Sección “Datos Incorrectos”.	
	4. El sistema muestra los resultados del control de la Firma Manuscrita.	
	5. El sistema muestra las opciones de Gestionar, Buscar, Comparar y Controlar las Firmas Manuscrita.	

Caso de Uso	Pre-procesar Firmas Manuscritas	
Actor	Usuario	
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción de pre-procesar Firmas Manuscritas. El caso de uso termina cuando se muestran los resultados de pre-procesar la Firma Manuscrita.	
Precondiciones		
Poscondiciones		
Referencia	RF 1.11	
Prioridad	Crítico	
Casos de uso asociados	Caso de Uso Gestionar Firmas Manuscritas	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema muestra las interfaces para pre-procesar una Firma Manuscrita.	
2. El Usuario selecciona e introduce los datos, selecciona la opción preprocesar una Firma Manuscrita.	3. El sistema valida los datos. En caso de datos incorrectos. Ver Sección "Datos Incorrectos".	
	4. El sistema muestra los resultados del pre-procesar la Firma Manuscrita.	
	5. El sistema muestra las opciones de Gestionar, Buscar, Comparar y Controlar las Firmas Manuscrita.	

Flujos Alternos	
Sección "Datos incorrectos"	
Acción del Actor	Respuesta del Sistema
	1. El sistema señala los datos incorrectos y muestra el mensaje "Datos Incorrectos".
2. El usuario corrige los datos incorrectos.	3. El sistema pasa al flujo normal de eventos.

Flujos Alternos	
Sección "Firma Manuscrita Existente"	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra el mensaje "La Firma Manuscrita ya existe".
2. El Usuario acepta el mensaje y selecciona la opción "Cancelar".	3. El sistema pasa al flujo normal de eventos.

Anexos correspondientes a los prototipos no funcionales del Módulo Firmas Manuscritas.

The screenshot shows a software interface with a menu bar at the top containing the following options: "Gestionar Firmas Manuscritas", "Buscar Firmas Manuscritas", "Comparar Firmas Manuscritas", "Control Firmas Manuscritas", "Adicionar Firmas Manuscritas", "Consultar Firmas Manuscritas", "Consultar Histórico de Firmas Manuscritas", and "Actualizar Fir". The "Adicionar Firmas Manuscritas" option is currently selected.

Below the menu bar, there are three dropdown menus labeled "País:", "Tipo de Identificación", and "Número".

Underneath these menus, there are two labels: "-Facsimil-" and "-Foto de la Firma-".

In the center of the interface, there is a photograph of a hand holding a fountain pen, writing a signature on a piece of paper.

Below the photograph, there is a button labeled "Examinar...".

At the bottom left, there is a label "Estado" followed by a text input field.

At the bottom right, there is a button labeled "Aceptar".

Anexo 1. Prototipo no funcional del caso de uso Adicionar Firmas Manuscritas.


Gestionar Firmas Manuscritas Buscar Firmas Manuscritas Comparar Firmas Manuscritas Control Firmas Manuscritas

Adicionar Firmas Manuscritas **Consultar Firmas Manuscritas** Consultar Histórico de Firmas Manuscritas Actualizar Fir

Pais: **Tipo de Identificación** **Número**

-Facsimil-

-Foto de la Firma-



Aceptar

Anexo 2. Prototipo no funcional del caso de uso Consultar Firmas Manuscritas.

The interface features a top navigation bar with four buttons: "Gestionar Firmas Manuscritas", "Buscar Firmas Manuscritas", "Comparar Firmas Manuscritas", and "Control Firmas Manuscritas". Below this is a secondary bar with five buttons: "Manuscritas", "Consultar Firmas Manuscritas", "Consultar Histórico de Firmas Manuscritas" (highlighted with a dashed border), and "Actualizar Firmas Manuscritas".

The main content area contains three dropdown menus for filtering: "País:", "Tipo de Identificación", and "Número". Below these is a table with the following headers: "Fecha Inicial", "Fecha Final", "Imagen", and "Estado". The table body is currently empty. At the bottom of the table area is a horizontal scrollbar.

Anexo 3. Prototipo no funcional del caso de uso Consultar Histórico de Firmas Manuscritas.

Gestionar Firmas Manuscritas Buscar Firmas Manuscritas Comparar Firmas Manuscritas Control Firmas Manuscritas


Manuscritas Consultar Firmas Manuscritas Consultar Histórico de Firmas Manuscritas Actualizar Firmas Manuscritas

Pais: **Tipo de Identificación:** **Número**

Cancelar Facimil Actual

-Facsimil-

-Foto de la Firma-



Estado del Vector

Anexo 4. Prototipo no funcional del caso de uso Actualizar Firmas Manuscritas.

Gestionar Firmas Manuscritas	Buscar Firmas Manuscritas	Comparar Firmas Manuscritas	Control Firmas Manuscritas
------------------------------	---------------------------	-----------------------------	----------------------------

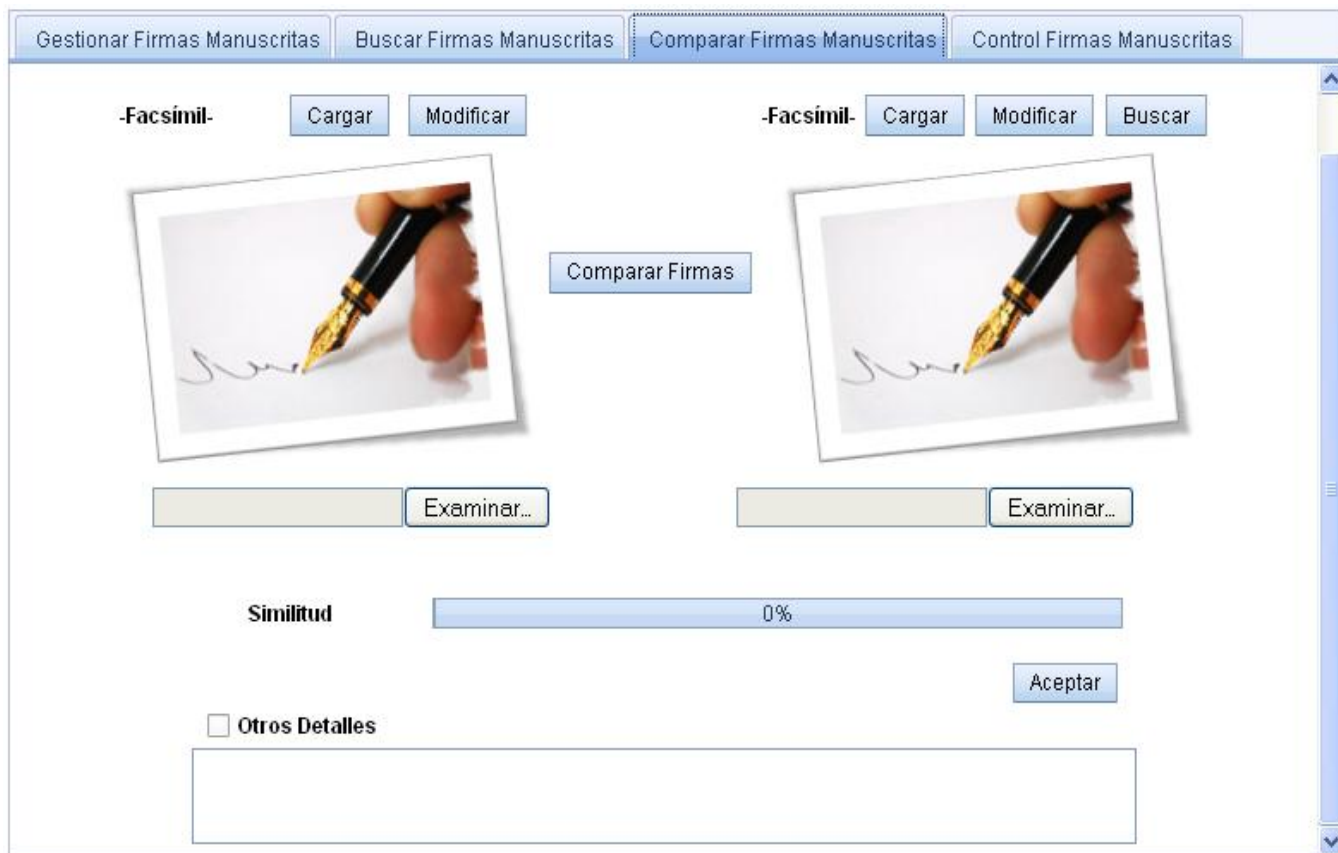
-Facsimil-	Cargar	Modificar		-Foto de la Firma-
------------	--------	-----------	--	--------------------

	Buscar Firmas	
---	---------------	--

Resultados de la Búsqueda

Por Ciento(%)	<input type="text"/>	
País	<input type="text"/>	5 Más Proximos Parecidos
Tipo de Identificación	<input type="text"/>	
Número	<input type="text"/>	Aceptar


Anexo 5. Prototipo no funcional del caso de uso Buscar Firmas Manuscritas.



Anexo 6. Prototipo no funcional del caso de uso Comparar Firmas Manuscritas.

Gestionar Firmas Manuscritas Buscar Firmas Manuscritas Comparar Firmas Manuscritas **Control Firmas Manuscritas**

-Facsimil- Cargar Modificar




-Personas Autorizadas-

Cliente

Cuenta

-Facsimil-



Resultados

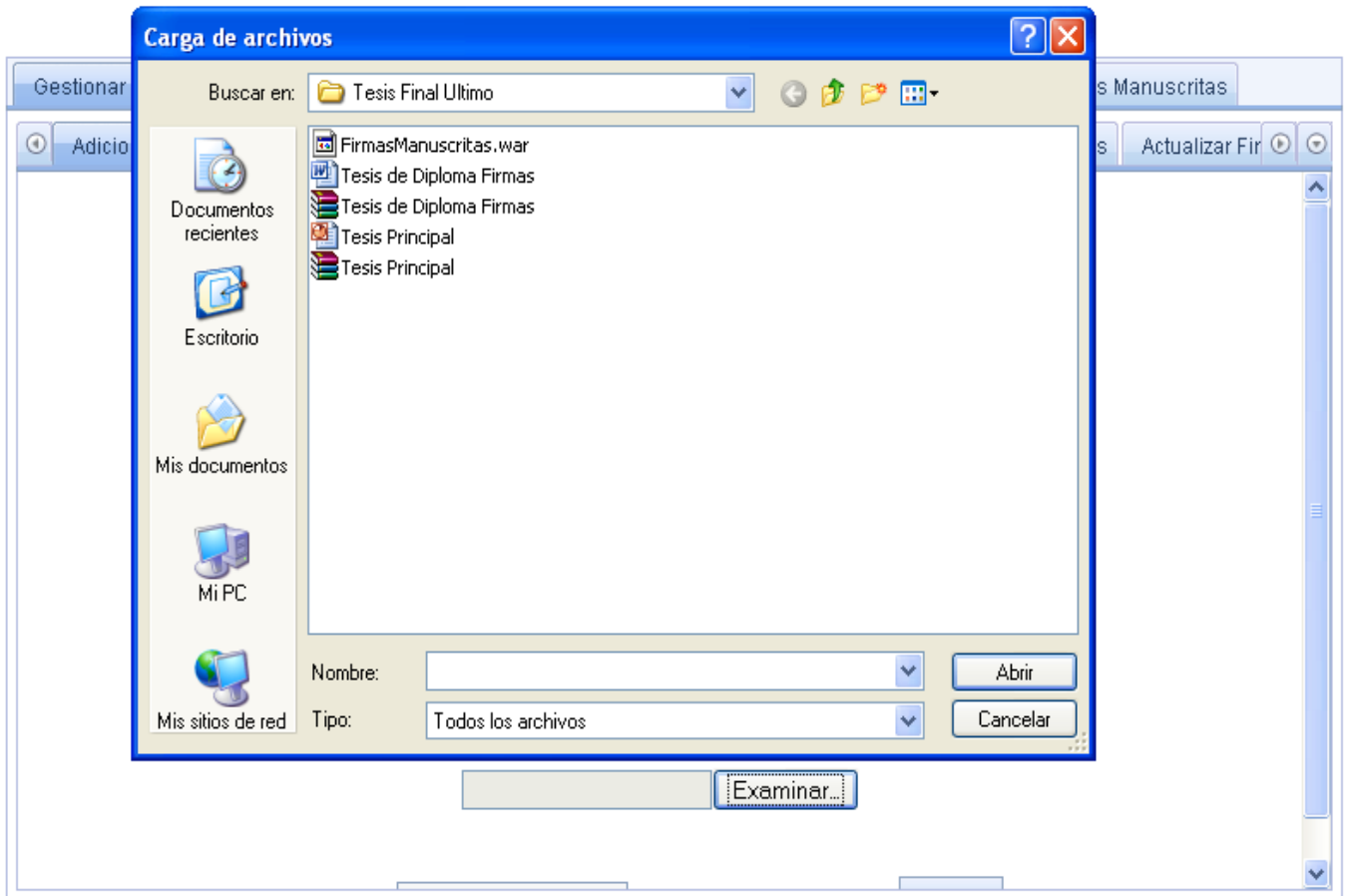
Por Ciento(%)

País

Tipo de Identificación

Número

Anexo 7. Prototipo no funcional del caso de uso Control de Firmas Manuscritas.



Anexo 8. Prototipo no funcional del caso de uso Cargar Firmas Manuscritas.

Anexos correspondientes a las clases del análisis que muestran las relaciones entre funcionalidades a nivel de casos de uso.



Anexo 9. Diagrama de Clases del Análisis del caso de uso Adicionar Firmas Manuscritas.



Anexo 10. Diagrama de Clases del Análisis del caso de uso Actualizar Firmas Manuscritas.



Anexo 11. Diagrama de Clases del Análisis del caso de uso Consultar Firmas Manuscritas.



Anexo 12. Diagrama de Clases del Análisis del caso de uso Cancelar Firmas Manuscritas.

GLOSARIO DE TÉRMINOS

Sistemas Bancarios

Conjunto de instituciones que permiten el desarrollo de todas aquellas transacciones entre personas, empresas y organizaciones que impliquen el uso de dinero.

Prototipo no funcional

El prototipo no funcional corresponde al diseño de las pantallas acorde con el estándar definido.

Framework

Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Módulo

Encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

IDE

Integrated Development Enviroment (Ambiente Integrado de Desarrollo).

SAGEB

Sistema Automatizado de Gestión Bancaria.

JavaServer Page (JSP)

Tecnología orientada a crear páginas web con programación en Java.

Sistema gestor de base de datos

Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

SQL (Structured Query Language)

Conjunto estándar de comandos para gestionar bases de datos relacionales por sus mismas características relacionales.

UML (Unified Modeling Language)

Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.