



Facultad 1

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

Título: Sistema de identificación de personas mediante firma manuscrita.

Autores:

Omar Barzaga Angulo

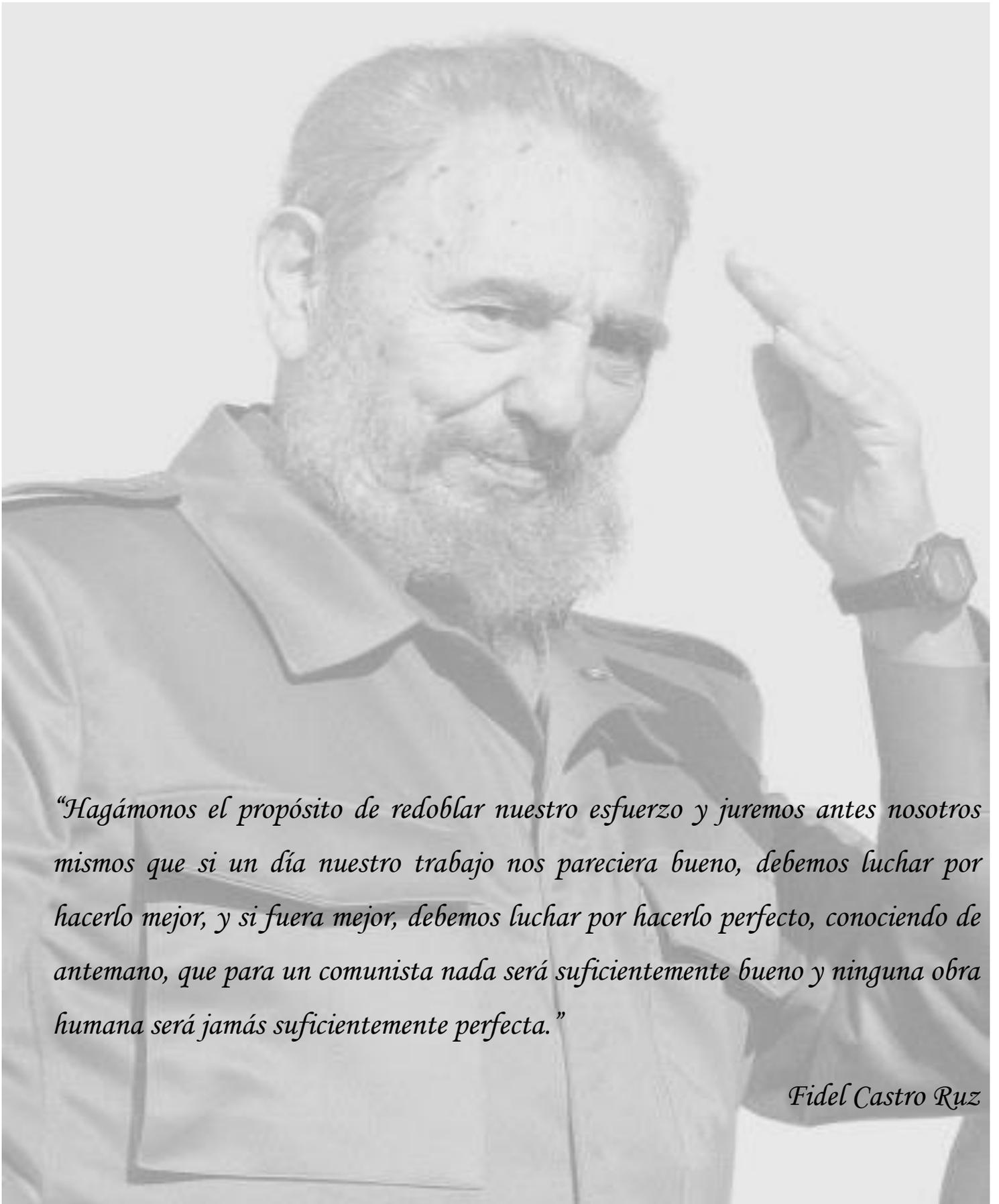
Glenys Rojas Morales

Tutores:

MsC. Adrian A. Machado Cento

Ing. Yainier Labrada Nueva

“La Habana, 2013”



“Hagámonos el propósito de redoblar nuestro esfuerzo y juremos antes nosotros mismos que si un día nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor, y si fuera mejor, debemos luchar por hacerlo perfecto, conociendo de antemano, que para un comunista nada será suficientemente bueno y ninguna obra humana será jamás suficientemente perfecta.”

Fidel Castro Ruz

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Omar Barzaga Angulo

MsC. Adrian A. Machado Cento

Glenys Rojas Morales

Ing. Yainier Labrada Nueva

Datos de Contacto

Tutor: Adrian Alberto Machado Cento

Correo electrónico: amachado@uci.cu

La Habana, Cuba

Tutor: Yainier Labrada Nueva

Correo electrónico: ylabrada@uci.cu

La Habana, Cuba

Datos: Glenys Rojas Morales

Correo electrónico: gmorales@estudiantes.uci.cu

La Habana, Cuba

Datos: Omar Barzaga Angulo

Correo electrónico: obarzaga@estudiantes.uci.cu

La Habana, Cuba

Agradecimientos

A la Revolución por ofrecernos la posibilidad de estudiar y realizarnos como profesionales. A todos los profesores que de una forma u otra influyeron en nuestra preparación. Al equipo del laboratorio, por su aporte en este trabajo. A los tutores, por la paciencia y dedicación que tuvieron con nosotros. A los miembros del tribunal y el oponente que influyeron en gran medida en el perfeccionamiento de esta investigación.

Glenys

A mis padres por todo el esfuerzo, la dedicación y el amor incondicional con el que me han guiado haciendo de mí una mujer de bien. Gracias por enseñarme a levantarme después de cada tropiezo, por enseñarme a ser perseverante, paciente y a decir en todo momento “yo sí puedo”. Les agradezco por todo lo que soy y puedo llegar a ser.

A mis abuelos por ese amor incondicional que siempre me han dado, por enseñarme lo fuerte que debo ser para afrontar la vida y aunque hoy uno no de ellos no puede estar a mi lado compartiendo este maravilloso momento, yo sé que se siente orgullosa de todo lo que he logrado y si hoy he tenido la fuerza de estar aquí con serenidad es gracias a ti abuela, porque tú eres mi ángel.

A mis tías y mis primos, por ese cariño inmenso que siempre me ofrecen, por hacerme reír cuando estoy a su lado, por poder contar con ellos en cualquier momento.

A mi novio por esa paciencia que ha tenido, gracias por esperarme, gracias por ese amor, alegría y satisfacción que me has dado.

A mis amigos por su apoyo, dedicación y consejos brindados los cuales tendré en mis pensamientos y en mi corazón.

Omar

A mis padres por ser los mejores del mundo, por enseñarme a vivir, soñar y caminar por el camino correcto, por darme los mejores consejos, por quererme, por confiar siempre en mí, por apoyarme en todo momento, por todo el esfuerzo, la dedicación y el amor incondicional con el que me han guiado, orientado y hecho un hombre de bien, les agradezco por todo lo que soy y puedo llegar a ser. A ellos porque con solo su presencia me llena de vida, alegrías y ganas de luchar cada día más. Los amo con todas las fuerzas de mi corazón.

A mi hermana por su amor incondicional, la adoro.

A toda mi familia por el amor, dedicación, empeño y confianza puestos en mí.

A mi esposa por sus consejos, su compañía, su dedicación, su apoyo, su amor, por estar cuando más la he necesitado y por quererme como me quiere.

A mis amigos y amigas en general, con todos con los que he compartido alegrías y tristezas en estos cinco años, les agradezco su amistad y acuérdense que no importa donde estemos siempre me acordaré de ustedes.

A nuestros tutores Machado y Labrada por todo el apoyo que nos brindaron, por la confianza que depositaron en nosotros, por dedicarnos su tiempo y paciencia.

A los profesores Adrián Rivera y Arnel Ledo por toda la ayuda que han brindado, muchas gracias.

A todos mis profesores, por contribuir con mi formación personal y profesional.

Quisiera agradecer a todas las personas que de una forma u otra han contribuido a que este sueño se haga realidad.

Dedicatoria

Glenys

A mis padres, quienes estarán orgullosos de verme al fin, graduada como Ing.

A mi familia: mis abuelos, tía y primos, que siempre me han apoyado en aras de cumplir mis sueños y metas.

A Leo, quien con su amor ha logrado ocupar un lugar importante en mi vida y en mi corazón.

A los amigos, que me han apoyado y han confiado siempre en mí.

Omar

A las dos personas que más admiro en la vida, mis padres del alma, Omar y Nallibe, por ser únicos y especiales.

A mi hermanita, por llenar mi vida de alegría y me ha dado tanto apoyo.

A mi esposa, por haber estado conmigo en los mejores y más difíciles momentos de mi carrera, dándome su eterno apoyo.

A toda mi familia que es maravillosa y de la cual me siento orgulloso.

A todos mis compañeros que me han apoyado y aguantado en estos cinco años.

Resumen:

Los conocimientos acerca de la Biometría y el desarrollo de los sistemas biométricos están muy avanzados en la actualidad. Es muy importante la evolución de estos sistemas ya que uno de sus principales objetivos es el de identificar a un individuo a partir de un rasgo biológico propio o de un rasgo conductual como puede ser la firma.

La firma ha sido usada para constatar la legalidad de un documento, verificando que las personas que firman el documento concuerdan con su identidad, garantizando, de esta manera, que todo sea legal. El reconocimiento de personas a través de la firma manuscrita es una realidad que está muy introducida en la vida de millones de personas alrededor de todo el mundo. Esto se debe a que el desarrollo de esta modalidad posibilita que la vida de las personas sea más cómoda y sobre todo más segura. En esta época, donde han evolucionado considerablemente las Tecnologías de la Información y las Comunicaciones (TIC), ya se proveen métodos que permiten automatizar este proceso de reconocimiento de persona mediante la firma manuscrita.

Debido a todo lo expuesto anteriormente el presente trabajo expone la realización de un sistema de identificación de personas mediante firma manuscrita, desarrollado en el Centro de Identificación y Seguridad Digital. Con la realización de este sistema se trata de lograr que la verificación manual, propensa a errores, sea sustituida por una verificación automática más rápida y confiable.

Palabras claves: Biometría, Firma Manuscrita, Persona, Sistemas biométricos.

Índice

Introducción	1
Capítulo I: Fundamentación teórica.	4
1.1 Introducción a la Biometría.	4
1.1.1 Biometría.	4
1.1.2 Identificación Biométrica.	5
1.1.3 Verificación Biométrica.....	5
1.1.4 Sistema Biométrico.	5
1.2 Introducción a la Firma.	5
1.2.1 Firma Manuscrita	6
1.2.2 Las Firmas Manuscritas en el Mundo.....	6
1.2.3 Características de la firma.....	7
1.2.4 Reconocimiento on-line y off-line.	7
1.2.5 Trabajos existentes.	8
1.3 Softwares actuales de comparación de firmas.	9
1.3.1 Sigcheck (SQN Banking System).....	9
1.3.2 Esolva.com.	9
1.3.3 Parascript.....	10
1.4 Tecnologías a utilizar para el desarrollo del trabajo.	10
1.4.1 Metodologías Ágiles.....	10
1.4.2 Metodologías Pesadas.....	11
1.4.3 Fundamentación de la metodología seleccionada.....	13
1.5 Librerías utilizadas.	13
1.5.1 ImagesWorking.	13
1.5.2 Itextcsharp.	17
1.6 Lenguaje unificado de modelado (UML por sus siglas en inglés).	18
1.7 Herramientas CASE (Computer Aided Software Engineering)	18
1.7.1 Rational Rose Enterprise Edition.....	18
1.7.2 Visual Paradigm.....	19

1.7.3 Fundamentación de la herramienta Case utilizada.....	20
1.8 Sistemas Gestores de Bases de Datos (SGBD)	20
1.8.1 MySQL.....	21
1.8.2 PostgreSQL.	22
1.8.3 Justificación del gestor de bases de datos seleccionado.	23
1.9 Framework de desarrollo	23
1.9.1 .Net framework 4.5.....	23
1.10 Visual Studio 2012.....	24
1.11 Lenguaje de programación seleccionado.....	24
1.12 Propuesta de selección de tecnologías y herramientas.....	25
1.13 Conclusiones Parciales.	25
Capítulo II: Características, Análisis y Diseño del sistema.....	26
2.1 Objeto de Informatización.....	26
2.2 Propuesta del sistema.....	26
2.3 Modelo de Dominio.	27
2.4 Requisitos Funcionales.	28
2.5 Requisitos no Funcionales.	29
2.6 Metáfora.	30
2.7 Fase de Exploración.....	30
2.7.1 Historias de Usuario (HU).	30
2.8 Fase de Planificación.	32
2.8.1 Estimación del esfuerzo por Historia de Usuario.	32
2.8.2 Plan de duración de las iteraciones.....	32
2.8.3 Plan de entrega.....	34
2.9 Diseño.	34
2.9.1 Modelo de Datos.....	34
2.9.2 Arquitectura.....	35

2.9.3 Patrones de diseño utilizados.	36
2.10 Conclusiones Parciales.	40
Capítulo III: Implementación y Prueba del sistema.	41
3.1 Tarjetas CRC (Clase-Responsabilidad-Colaborador).....	41
3.2 Tareas de Ingeniería.....	42
3.3 Estándar de Codificación.....	42
3.3.1 Reglas de Codificación.	43
3.4 Diagrama de Componente.	43
3.5 Diagrama de Despliegue.....	44
3.6 Interfaz de Usuario.	45
3.7 Pruebas.	48
3.7.1 Pruebas de Caja Blanca o Estructural.....	49
3.7.2 Pruebas de Caja Negra o Funcional.	52
3.7.3 Resultado de las pruebas.....	54
3.8 Conclusiones parciales.....	54
Conclusiones generales.....	56
Recomendaciones.....	57
Bibliografía referenciada.....	58
Bibliografía consultada.....	61
Glosario de Términos.....	63
Anexos.....	64
Anexo 1: Historias de Usuario.	64
Anexo 2: Descripción de las tablas de la base de datos.	68
Anexo 3: Tarjetas CRC.....	72
Anexo 4: Tareas de ingeniería asociadas a las Historias de Usuario.....	75

Anexo 5: Interfaces de Usuario.77

Anexo 6: Casos de Prueba de aceptación para cada Historia de Usuario.82

Índice de Figuras.

Figura 1. Firma manuscrita.	6
Figura 2. Reconocimiento on-line.....	8
Figura 3. Reconocimiento off-line.....	8
Figura 4. Metodología de desarrollo de software RUP.	12
Figura 5. Matrices de convolución.....	15
Figura 6. Etiquetado de la firma según su inclinación.	16
Figura 7. Cálculo de la Caja de Feret.....	17
Figura 8. Etapas del sistema de identificación de personas mediante firmas manuscritas.	27
Figura 9. Modelo de dominio.....	27
Figura 10. Modelo de datos.	35
Figura 11. Arquitectura N-capas.	36
Figura 12 Patrón Singleton	37
Figura 13. Patrón de asociación.....	39
Figura 14. Patrón de llaves subrogadas.....	40
Figura 15. Diagrama de componentes.	44
Figura 16. Diagrama de despliegue.	45
Figura 17. IU_Sistema.	45
Figura 18. IU_Insertar firma.	46
Figura 19. IU_Mostrar firmas.....	47
Figura 20. IU_Comparar firmas.....	48

Figura 21. Grafo de Flujo: Funcionalidad ObtenerPatrones.	51
Figura 22. Resultados de la pruebas de aceptación.....	54
Figura 23. IU_Autenticar usuario.....	77
Figura 24. IU_Gestionar usuarios.	78
Figura 25. IU_Gestionar persona.	79
Figura 26. IU_Modificar perfil.	80
Figura 27. IU_Obtener imagen de la firma del documento.	81
Figura 28. IU_Mostrar reportes.	82

Índice de Tablas.

Tabla 1. Funcionalidades del sistema.	28
Tabla 2. HU_Gestionar firma de persona.	31
Tabla 3. HU_Comparar firma	31
Tabla 4. Estimación de esfuerzo.	32
Tabla 5. Plan de iteraciones.	33
Tabla 6. Plan de entrega.	34
Tabla 7. Tarjeta CRC - Controladora.cs	41
Tabla 8. Estándares de codificación.	42
Tabla 9. Reglas de codificación.	43
Tabla 10. Caso de prueba Insertar firmas a personas.	52
Tabla 11. Caso de prueba Mostrar firmas de personas.	52
Tabla 12. Caso de prueba Comparar firmas.	53
Tabla 13. HU_Autenticar usuario.	64
Tabla 14. HU_Gestionar usuario.	64
Tabla 15. HU_Modificar perfil.	65
Tabla 16. Gestionar persona.	65
Tabla 17. Obtener imagen de la firma del documento.	66
Tabla 18. Identificar persona.	66
Tabla 19. Mostrar reportes de comparación de firmas.	67
Tabla 20. Exportar reportes de comparación a pdf.	67

Tabla 21. Tabla usuario.	68
Tabla 22. Tabla rol.	69
Tabla 23. Tabla patron.	69
Tabla 24. Tabla persona.	69
Tabla 25. Tabla firma.	70
Tabla 26. Tabla segmento.	70
Tabla 27. Tabla reporte.	71
Tabla 28. Tarjeta CRC correspondiente a Utiles.cs.	72
Tabla 29. Tarjeta CRC correspondiente a UsuarioDAO.cs.	72
Tabla 30. Tarjeta CRC correspondiente a RolDAO.cs.	73
Tabla 31. Tarjeta CRC correspondiente a SegmentoDAO.cs.	73
Tabla 32. Tarjeta CRC correspondiente ReporteDAO.cs.	73
Tabla 33. Tarjeta CRC correspondiente a PersonaDAO.cs.	73
Tabla 34. Tarjeta CRC correspondiente a PatronesDAO.cs.	74
Tabla 35. Tarjeta CRC correspondiente a FirmaDAO.cs.	74
Tabla 36. Tarjeta CRC correspondiente a Conexion.cs.	75
Tabla 37. Tareas de ingeniería.	75
Tabla 38. Caso de prueba Autenticar usuario.	82
Tabla 39. Caso de prueba Insertar usuarios.	83
Tabla 40. Caso de prueba Eliminar usuarios.	83
Tabla 41. Caso de prueba Listar usuarios.	84

Tabla 42. Caso de prueba Modificar usuario.....	85
Tabla 43. Caso de prueba Modificar contraseña.....	85
Tabla 44. Caso de prueba Modificar tipo de usuario.....	86
Tabla 45. Caso de prueba Insertar persona.....	87
Tabla 46. Caso de prueba Listar y modificar datos de persona.....	87
Tabla 47. Caso de prueba Obtener imagen de la firma del documento.....	88
Tabla 48. Caso de prueba Identificar persona.....	89
Tabla 49. Caso de prueba Mostar reportes de comparación de firmas.....	89
Tabla 50. Caso de prueba Exportar reportes de comparación de firmas a pdf.....	90

Introducción

En la actualidad, la humanidad depende en gran medida de las Tecnologías de la Informática y las Comunicaciones (TIC). Con el uso de las TIC se han automatizado numerosos procesos que antiguamente se realizaban de forma manual, beneficiando en gran medida a la sociedad. Ejemplo de esto es la informatización de la biometría.

La Biometría consiste en la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos y conductuales, para poder verificar identidades e identificar individuos [1]. Entre las diferentes modalidades para el reconocimiento de personas en la que esta se manifiesta se encuentra, el reconocimiento de la huella, del rostro y del iris, como ejemplo de características físicas (estáticas) y de la voz, los gestos y la firma, como ejemplo de características conductuales (dinámicas). Sobre esta última se estará haciendo mayor énfasis a lo largo del desarrollo de este trabajo.

La mayoría de las personas, han tenido que plasmar su firma como señal de conocimiento o aceptación o les ha tocado constatar la legalidad de algún acuerdo, verificando que las personas que firman el documento concuerdan con su identidad. Muchas veces este proceso de verificación puede resultar molesto, sin embargo, la Biometría brinda diferentes técnicas destinadas a la identificación y verificación de las firmas manuscritas, aportando una serie de mejoras considerables como la rapidez y la fiabilidad.

En Cuba la mayoría de los procesos de identificación y verificación de personas, se realiza mediante un proceso de comparación manual: entre el documento oficial del sujeto y el firmado por él, siendo un proceso inexacto.

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el Centro de identificación y Seguridad Digital (CISED), se están llevando a cabo diferentes proyectos para desarrollar sistemas de seguridad, con reconocimiento y autenticidad biométrica, basados en el reconocimiento de rasgos faciales, huellas dactilares y firmas manuscritas. En estos momentos en el centro no existe un sistema de identificación de personas a través de la firma manuscrita.

Atendiendo a la situación problemática descrita con anterioridad, se plantea el siguiente **problema a resolver**: ¿Cómo identificar a una persona mediante su firma manuscrita?

El presente trabajo tiene como **objeto de estudio**: Proceso de identificación de personas mediante firmas manuscritas.

El **objetivo general** que se define en la presente investigación es: Desarrollar un sistema que permita identificar a una persona mediante la comparación de su firma con otras de las que ya se conoce su autor.

Para dar cumplimiento al objetivo general, se definieron los siguientes **objetivos específicos**:

- ✓ Determinar las tendencias mundiales en la comparación de firmas e identificación de personas.
- ✓ Definir la metodología y herramientas a utilizar en el desarrollo del sistema.
- ✓ Efectuar el análisis y diseño de la aplicación.
- ✓ Implementar la aplicación para la identificación de personas.
- ✓ Realizar pruebas al sistema.

Para dar cumplimiento a los objetivos específicos, se plantean las siguientes **tareas investigativas**:

- ✓ Análisis de los fundamentos teóricos, referente a la comparación de firmas e identificación de personas.
- ✓ Determinación de las tecnologías, metodologías y herramientas a utilizar para el desarrollo del sistema.
- ✓ Levantamiento de los requisitos funcionales y no funcionales, así como la definición de la arquitectura del sistema y el diseño de la base de datos del sistema.
- ✓ Integración con el componente de comparación de firmas.
- ✓ Realización de pruebas unitarias y de aceptación del sistema.

Para la realización de esta investigación se utilizaron los siguientes **métodos científicos**:

Analítico - Sintético: Permite realizar un estudio bibliográfico, así como constatar y organizar la información, de varias fuentes que caracterizan los sistemas de identificación de persona utilizando la firma manuscrita.

Inducción – Deducción: Permite comprender los principales fundamentos teóricos de la bibliografía consultada, la cual enriquece el marco teórico.

Histórico - Lógico: Permite analizar los antecedentes y tendencias actuales de los sistemas de identificación de personas a través de firmas manuscritas.

Modelación analógica: Posibilita diseñar una base de datos que satisfaga el objetivo de la investigación.

Observación: Permite obtener una visión del problema existente y comprender las ventajas y desventajas de estos sistemas.

La presente investigación está dividida en 3 capítulos, organizados de la siguiente forma:

Capítulo 1. Fundamentación teórica.

Se analizan diferentes conceptos relacionados con el desarrollo del tema tratado. Contiene el estado de arte en el mundo, en Cuba y en la UCI. Se describen las tecnologías, metodologías y herramientas utilizadas para solucionar el problema planteado.

Capítulo 2. Características, Análisis y Diseño del sistema.

Se aborda la situación problemática y su propuesta de solución, se especifican los requerimientos funcionales y no funcionales, así como se contextualizan los principales artefactos generados por la metodología seleccionada, por ejemplo: las historias de usuarios. También se realiza la descripción de la arquitectura.

Capítulo 3. Implementación y Prueba del sistema.

Se lleva a cabo el desarrollo del sistema, y una vez finalizada la implementación, se realiza la validación del diseño propuesto, con el propósito de sistematizar los requisitos obtenidos con anterioridad, para facilitar la comprensión, modificación y mantenimiento de los mismos.

Capítulo I: Fundamentación teórica.

En este capítulo se abordan conceptos fundamentales relacionados con la Biometría y la firma manuscrita que ayudarán al entendimiento del sistema que se quiere desarrollar, además se realiza un estudio del estado del arte, del problema a tratar y se realiza un análisis de las tecnologías, metodologías y herramientas a utilizar en el desarrollo del sistema.

1.1 Introducción a la Biometría.

Los conocimientos acerca de la Biometría y el desarrollo de los sistemas biométricos están muy avanzados en la actualidad. Es muy importante la evolución de estos sistemas ya que uno de sus principales objetivos es el de identificar a un individuo a partir de un rasgo biológico propio o de un rasgo conductual.

1.1.1 Biometría.

Al estudiar el concepto de Biometría se constataron las siguientes clasificaciones:

- **Biometría.**

Es la aplicación de técnicas matemáticas y esquemas al estudio de datos de las ciencias biológicas. [2]

- **Biometría en el Contexto Tecnológico.**

Es el conjunto de métodos, técnicas y tecnologías, que permiten la identificación y verificación de las personas de forma automatizada, sobre la base de características corporales o de comportamiento. [2]

Después de haber analizado las clasificaciones expuestas anteriormente, se decidió que la más aceptada es la desarrollada en el contexto tecnológico, ya que tiene mayor aceptación con el objeto de estudio.

Tipos de Biometría.

- **Biometría Estática:** Es la encargada de medir las características físicas de un individuo. Entre ellas se encuentran:

- ✓ Huellas Digitales.

- ✓ Geometría de la mano.

- ✓ Termografía.

- ✓ Análisis del Iris.

- ✓ Análisis de Retina.
- ✓ Venas del dorso de la mano.
- ✓ Reconocimiento Facial.
- **Biometría Dinámica:** Es la encargada de medir los rasgos de comportamiento de un individuo. Entre estas se encuentran:
 - ✓ Patrón de voz.
 - ✓ Firma manuscrita.
 - ✓ Dinámica de tecleo.
 - ✓ Cadencia del paso.
 - ✓ Análisis gestual.

1.1.2 Identificación Biométrica.

Cuando se utiliza la Biometría para la identificación; los usuarios no declaran que son ellos, el proceso es definir cuál es la identidad de la persona. Este proceso es de uno a varios, ya que la lectura de datos realizada, debe compararse con muchos perfiles potenciales que pueden coincidir. [3]

1.1.3 Verificación Biométrica.

En el proceso de verificación, los usuarios exponen su identidad al sistema. La tecnología biométrica se usa para comprobar la identidad de forma más confiable. Este proceso es considerado de uno a uno, ya que los datos ingresados al sistema biométrico, se comparan con un perfil almacenado y se verifica si coinciden o no. [3]

1.1.4 Sistema Biométrico.

Es un sistema de reconocimiento de patrones, que funciona extrayendo un conjunto de características biométricas de un individuo, comparándolas con otras almacenadas en una base de datos. Dependiendo del uso del sistema, puede funcionar en modo de identificación o verificación. [2]

1.2 Introducción a la Firma.

La firma manuscrita, constituye uno de los principales métodos de justificación de autoría que manejan las personas, es un mecanismo de autenticación personal ampliamente aceptado por la sociedad en

general, así como por los organismos públicos y privados; una de sus ventajas radica en la facilidad de realización, pues apenas se necesita un papel y un bolígrafo.

A continuación se tratarán los diferentes conceptos relacionados con la firma, así como temas asociados con la modalidad de la misma.

1.2.1 Firma Manuscrita

Mustapich define la firma como:

“... el nombre escrito por propia mano en caracteres alfabéticos y de una manera particular, al pie del documento, al efecto de autenticar su contenido”. [4]

Planiol y Ripert la definen de esta manera:

“... la firma es una inscripción manuscrita que indica el nombre de una persona que entiende hacer suyas las declaraciones del acto”. [4]

Se puede decir que la definición de firma se ha asociado, al trazado gráfico conteniendo habitualmente: el nombre, apellido y eventualmente la rúbrica de una persona, con el cual se suscriben los documentos para darles autoría y obligarse con lo que en ellos se dice. [4] (Ver Figura .1)



Figura 1. Firma manuscrita.

1.2.2 Las Firmas Manuscritas en el Mundo.

Desde hace mucho tiempo la firma manuscrita ha servido para identificar y validar documentos, pero desde su surgimiento ha arrastrado consigo grandes problemas, como por ejemplo la falsificación. A pesar de esto la firma ha sido el método más eficiente y aceptado para verificar e identificar la autenticidad de un documento.

La firma en si está compuesta por dos acciones, la acción de firmar y la de verificación de la firma. Para la primera, solo hace falta que la persona plasme sobre un documento: su nombre o alguna secuencia de caracteres que lo identifique personalmente. La acción de verificación de la firma es más compleja; este proceso en la mayoría de las ocasiones se realiza de forma visual, es decir, la persona encargada de verificar la firma, solo compara la del documento con otra que se encuentra en alguna identificación o que se realice en ese momento, y de esta forma es aceptada o rechazada. Este proceso de verificación también puede ser ordenado por un juez a persona natural o jurídica, por lo que la verificación visual no es suficiente para que esta sea aceptada o rechazada, ya que este proceso es llevado a cabo, por un experto en el campo de la firma y se realiza utilizando una serie de pruebas, que son las encargadas de chequear las características propias de las líneas asociadas a cada persona.

1.2.3 Características de la firma.

Cuando un autor ha culminado su obra, para poner de manifiesto que le pertenece, pone una “firma” o un “signo” que puede ser su nombre (real o artístico), que le servirá para diferenciarse del resto de las personas. Por otra parte, cuando una persona desea dejar un testamento o un poder legal sobre algo plasma su firma sobre estos documentos. [4]

Características que distinguen una firma:

- **Identificativa:** con ella se identifica el autor de un documento o una obra.
- **Declarativa:** el autor asume el contenido del mensaje; en la conclusión de un contrato, la firma implica la voluntad de obligarse a cumplirlo.
- **Probatoria:** sirve para identificar si efectivamente ha sido esa persona la que ha realizado la firma.

Aplicaciones de la firma.

Es frecuente la utilización de la firma en diferentes documentos, como por ejemplo: en transacciones bancarias o en otros documentos legales, como puede ser un contrato laboral, un certificado médico, una hipoteca, verificando a través de la misma que todo sea legal. La firma también es usada en el mundo del arte, como por ejemplo: en las pinturas, donde los autores firman sus obras poniendo su nombre real o artístico, para que el público pueda reconocer su obra. [4]

1.2.4 Reconocimiento on-line y off-line.

Los sistemas de verificación de firmas se dividen generalmente en dos categorías: sistemas de reconocimiento *on-line* y sistemas de reconocimiento *off-line*.

En el reconocimiento *on-line* se requiere la presencia física del firmante, por lo que es necesario emplear una tableta digitalizadora (*Tablet*) y un bolígrafo electrónico, lo que puede traer consigo, alteración en algunas características de la firma, ya que no es lo mismo escribir con un bolígrafo en un papel, que escribir sobre una tabla de plástico.

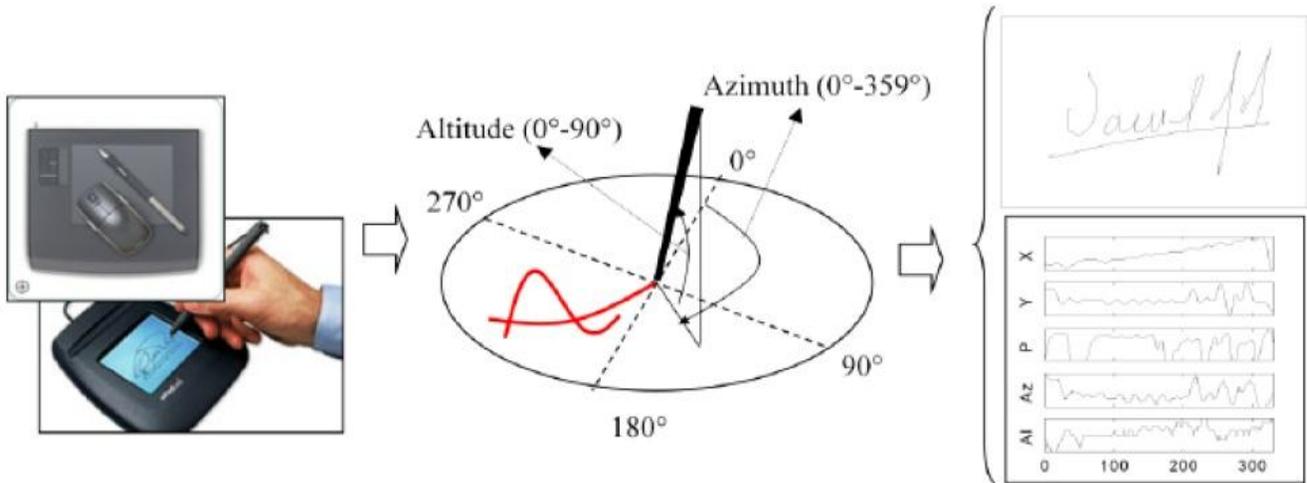


Figura 2. Reconocimiento on-line.

Por otra parte, el reconocimiento *off-line* parte de la firma plasmada en un documento, y tras escanear la imagen, se extraen las características de la firma, para verificar la identidad del firmante y resulta menos costoso a nivel de equipamiento.

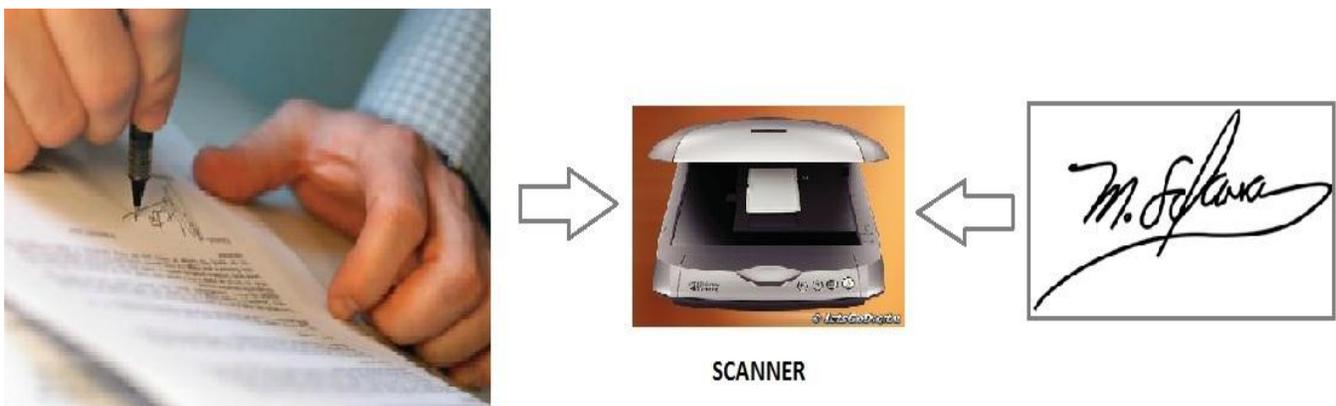


Figura 3. Reconocimiento off-line.

1.2.5 Trabajos existentes.

El tratamiento de firmas es un área de investigación muy activa desde mediados de 1970 [5]. Existen multitud de trabajos que abordan cada uno de los aspectos que se han presentado, por ejemplo: los hay

que tratan el problema de la localización de la firma en un entorno ruidoso, los que tratan el problema de la imposibilidad de usar más de una firma por individuo para el entrenamiento [6], los que tratan el problema de la escalabilidad no asistida [7], o los que tratan el problema de falsificadores entrenados [8].

En general, los enfoques existentes comparten el esquema clásico de extracción de características discriminantes y el posterior uso de un clasificador basado en tales características.

1.3 Softwares actuales de comparación de firmas.

1.3.1 Sigcheck (SQN Banking System).

Este sistema incluye un *software* para la detección de fraudes en imagen, la comparación automatizada de firmas y verificación de la misma. También realiza la extracción de la firma y las compara con las almacenadas en una base de datos, determinando que tan bien coinciden, con el objetivo de detectar un posible fraude.

Tiene un único proceso dentro del programa de comparación de firmas que analiza y genera un factor de confianza de 0 a 100. [18] [31]

Principales características de este software:

- Ahorra tiempo.
- Es muy flexible.
- Es auditable.
- Es una aplicación distribuida.

1.3.2 Esolva.com.

Esolva.com incluye un *software* desarrollado por Alejandro Pérez Hernández, que permite comparar dos firmas por medio del esqueletizado, la normalización, matrices de convolución y redes neuronales. Esto permite dar un nivel de confianza a la comparación de firmas. [18]

Funcionalidades del software:

- ✓ Es capaz de decidir si dos firmas se parecen lo suficiente para decir que son del mismo autor.
- ✓ Se centra en detectar falsificaciones burdas, ya que estas son las que una persona sin entrenamiento especial consigue diferenciar.

- ✓ Busca en una base de datos de firmas, aquella que más se parece a la firma que se quiera reconocer.

1.3.3 Parascript.

Es un sistema integral de verificación de la firma, que analiza dos áreas diferentes de un individuo: las características específicas de una imagen estática de su firma y las características específicas del proceso de firma.

El primer tipo incluye, aplicaciones que tratan con la imagen estática de dos dimensiones de la firma resultante de una acción, es decir, firma que ya ha tenido lugar. Estas aplicaciones que solo analizan los datos estáticos de una imagen de la firma se llaman, fuera de línea. [18]

El segundo tipo abarca, las aplicaciones que permiten el seguimiento del movimiento en el proceso de firma en el momento de la presentación.

1.4 Tecnologías a utilizar para el desarrollo del trabajo.

Desarrollar un *software* puede resultar en muchos casos muy complicado y por esta razón es que se utilizan las metodologías, las cuales van acompañadas de herramientas que son las encargadas de facilitar el desarrollo de la aplicación.

En el presente epígrafe, se hace referencia a las diferentes metodologías y herramientas que se pueden utilizar en el desarrollo del sistema.

1.4.1 Metodologías Ágiles.

EXtreme Programming (XP): es una metodología ágil cuyo propósito es potenciar las relaciones interpersonales, con el objetivo de lograr el éxito en el desarrollo del *software*. Esta metodología promueve el trabajo en equipo, proporcionando de esta forma un ambiente de trabajo adecuado. La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Esta metodología trata de dar al cliente el *software* que él requiere y cuando lo necesita. [22]

Las técnicas y principios fundamentales de XP son:

- ✓ **Un desarrollo incremental e iterativo:** pequeñas y continuas mejoras.
- ✓ **Pruebas unitarias continuas y de integración:** se realizan pruebas por cada una de las clases creadas, añadidas o modificadas, pruebas funcionales, pruebas de regresión y pruebas realizadas por parte del cliente para asegurar el cumplimiento de los requisitos. [22]

- ✓ **Programación en dúos o cuartetos:** dependiendo del tamaño del proyecto y el número de desarrolladores con que se disponga se puede usar dos o cuatro personas. Las tareas serán llevadas a cabo en conjunto en un mismo puesto de trabajo. El código es escrito, revisado y discutido al mismo tiempo, dando menos posibilidad a errores. [22]
- ✓ **El cliente se integra al equipo de trabajo,** siendo él mismo un integrante más, así se tiene la certeza de desarrollar lo que el cliente desea y como lo desea.
- ✓ **El código** es desarrollado de la manera más sencilla posible.

Ventajas de XP:

- ✓ Programación organizada.
- ✓ Menor tasa de errores.
- ✓ Satisfacción del programador.

Desventajas de XP:

- ✓ Es recomendable emplearla solo en proyectos a corto plazo.
- ✓ Altas comisiones en caso de fallar.

1.4.2 Metodologías Pesadas.

Las metodologías pesadas, se basan en tener una fuerte planificación durante todo el desarrollo del proyecto. *Rational Unified Process* (RUP: por sus siglas en inglés) o Proceso Unificado de Desarrollo es un proceso de desarrollo de *software*, el cual constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. [13]

RUP posee tres características principales:

- ✓ **Está dirigido por los casos de uso:** un caso de uso es aquel que describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante.
- ✓ **Centrado en la arquitectura:** la arquitectura es una vista del diseño completo del *software*, con las características más importantes resaltadas, dejando a un lado los detalles, lo que le permite a los desarrolladores una mayor visibilidad del sistema.
- ✓ **El desarrollo es iterativo e incremental:** por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo.

RUP divide el proceso de desarrollo del *software* en cuatro fases, dentro de las cuales se realizan varias iteraciones, en un número variable según el proyecto. La terminación de cada fase ocurre con el cumplimiento del hito u objetivo correspondiente a cada una.

Fases de RUP:

Inicio: En esta fase las iteraciones se centran con mayor énfasis en las actividades de moldeamiento de la empresa y en sus requerimientos. [12]

Elaboración: Durante esta fase se establece la estructura base para la arquitectura del sistema, proporcionando el diseño del mismo y el desarrollo de la siguiente fase. [12]

Construcción: Se obtiene un producto documentado, con un manual de usuario listo para su utilización. Se obtiene uno o varios entregables del producto que han pasado las pruebas. Se ponen estos entregables a consideración de un subconjunto de usuarios. [12]

Transición: El entregable ya está listo para su instalación en las condiciones reales, es decir, se busca garantizar un producto preparado, para su entrega al usuario. [12]

RUP cuenta con 9 flujos de trabajo o disciplinas, los 6 primeros conocidos como flujos de ingeniería y los 3 últimos de apoyo. (Ver Figura. 4)

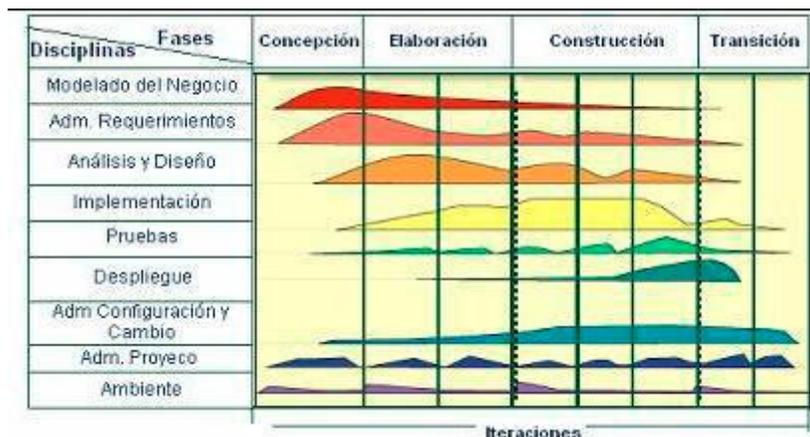


Figura 4. Metodología de desarrollo de software RUP.

1.4.3 Fundamentación de la metodología seleccionada.

Después del análisis hecho anteriormente sobre las posibles metodologías a utilizar en el desarrollo de este trabajo, se decidió utilizar la metodología XP por las razones que se darán a continuación:

El objetivo de XP se basa en la satisfacción del cliente. Esta metodología trata de dar al cliente el *software* que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.

XP está especialmente orientada para proyectos pequeños, lo que constituye una solución a medida para ese entorno, aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo con lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del *software*.

1.5 Librerías utilizadas.

1.5.1 ImagesWorking.

Es una librería de clases que fue desarrollada en el Centro de Identificación y Seguridad Digital (CISED) en el trabajo de diploma Componente para la extracción y verificación *off-line* de características de la firma manuscrita, de los autores Dreyllis Ertilla Díaz Silveira y Reinier Hernández Pérez.

A continuación se explican las principales funcionalidades con las que cuenta esta librería.

Pre-procesamiento de la imagen de la firma.

En muchas ocasiones, al escanear un documento, se introducen errores debido a manchas en el cristal y suciedad en el documento. Estas manchas, pueden resultar muy molestas a la hora de interpretar la información. [9]

En esta funcionalidad, la imagen de la firma es suprimida de aquellas suciedades que impiden, que se puedan revelar datos la imagen.

Para que la funcionalidad cumpla con su objetivo debe apoyarse otras sub-funcionalidades como son:

- **Binarización.**

La mayoría de los algoritmos para reconocer escritura están desarrollados a partir de imágenes binarias, por lo que se recomienda pasar una imagen que está en escala de grises o en colores a binaria, con el objetivo de reducir el volumen de datos a tratar. La binarización de una imagen digital radica en convertir

la imagen en una en blanco y negro, de tal manera que se preserven las propiedades esenciales de la imagen. Uno de los métodos para poder binarizar una imagen digital es mediante el histograma de dicha imagen. A través del histograma se adquiere una gráfica donde se muestra el número de *píxeles* por cada nivel de gris que aparece en la imagen. Para binarizar la imagen se elige un valor que se encuentre dentro de los niveles de grises (umbral), de manera que el histograma forme un valle en ese nivel. Luego todos los niveles de grises menores que ese umbral se convierten a negro y los mayores que el umbral a blanco. [10]

- **Segmentación.**

Después de obtenerse la imagen binaria esta se segmenta en las diferentes componentes conexas (parte de la imagen donde los *píxeles* son adyacentes entre sí) que la componen. La segmentación de la imagen es uno de los mayores problemas del reconocimiento, y es muy necesaria para reconocer cada uno de los caracteres de la imagen binaria. Mediante esta segmentación se descompone el texto en diferentes entidades lógicas. [9]

El algoritmo utilizado para realizar la segmentación de la imagen de la firma es el de Segmentación por umbral ya que este segmenta la imagen *pixel a pixel*. Si el valor de un *pixel* está dentro del rango de valores especificados para un objeto el *pixel* es segmentado. [9]

- **Esqueletizado.**

La esqueletización o adelgazamiento es una técnica muy usada en el reconocimiento de patrones. Consiste en ir eliminando *píxeles* de la imagen hasta que esta se quede en un esqueleto de un *pixel* de grosor. Este esqueleto contiene toda la información de la región, conservando el mismo número de regiones de la imagen, y debe ser similar a la región original. Esto resulta complicado de conseguir y puede introducir ciertos “artefactos” de ruido no presentes en la firma original. [11]

El algoritmo utilizado para realizar el adelgazamiento de la imagen es el de Zhang-Suen ya que es rápido y sencillo de implementar, además mediante cada *pixel* se puede calcular utilizando el valor de la iteración anterior.

- **Técnicas de etiquetado de trazos.**

El siguiente paso consiste en detectar los *píxeles* que forman los trazos, y etiquetarlos según la inclinación del trazo, de esta forma, se logrará extraer los trazos verticales y horizontales. [23]

Extracción de características.

Las principales características que se buscan en las firmas son los trazos con una orientación de 0°, 45°, 90° y 135°. Una vez que se tienen los *píxeles* etiquetados con su orientación, se debe hacer un seguimiento y extracción de los segmentos.

Reconocimiento basado en trazos.

Como resultado de las etapas anteriores, las firmas se describen ahora mediante una lista de trazos que son independientes al tamaño original de la firma y al grosor del bolígrafo utilizado. [24]

Algoritmos de Matrices de convolución

Las matrices de convolución permiten realizar el dominio del espacio filtrado, en el que el nuevo valor de un *píxel* depende únicamente de su valor actual y del de sus vecinos. [25]

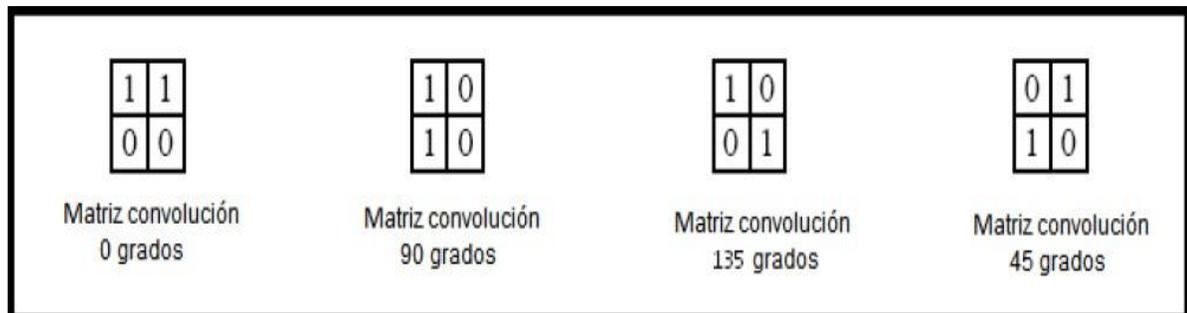


Figura 5. Matrices de convolución.

De esta manera, se obtienen 4 firmas de salida como se muestra en la Figura 6.

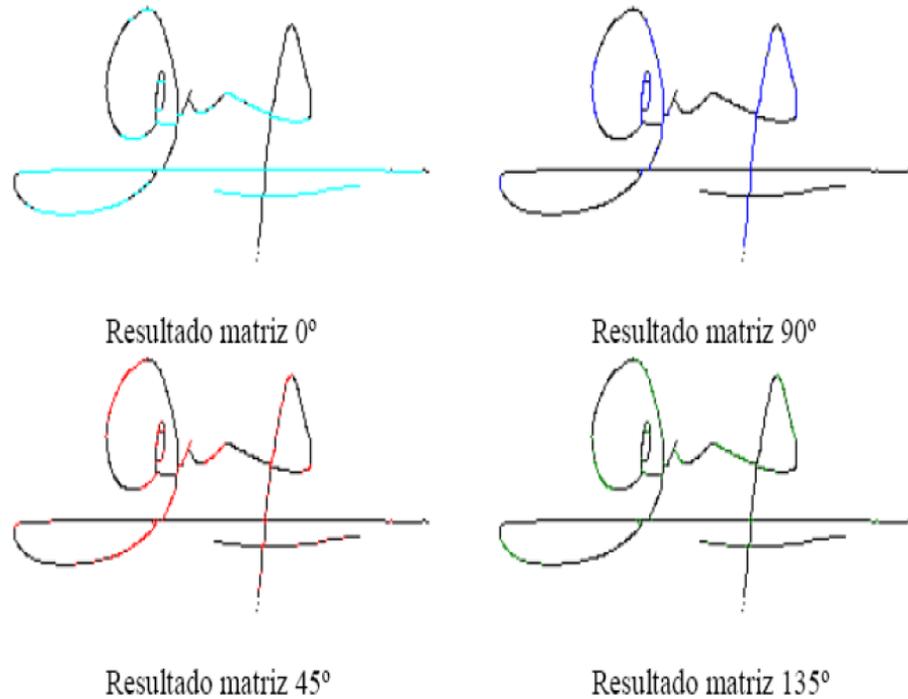


Figura 6. Etiquetado de la firma según su inclinación.

Un mismo *pixel* puede haber sido etiquetado con cuatro inclinaciones distintas, un *pixel* puede formar parte de varios trazos. Esto evita la pérdida de información.

En este proceso se produce una simplificación y por tanto una pérdida de información respecto al original, ya que los trazos se han aproximado a rectas de 0° , 45° , 90° y 135° . Esto será precisamente lo que permitirá decir que dos trazos de distintas firmas de la misma persona sean iguales a pesar de que sean ligeramente diferentes.

Algoritmo de seguimiento y extracción de trazos.

Mediante la utilización de este algoritmo se confecciona una lista de trazos con la cual se trabajará de ahora en adelante.

Descripción del algoritmo Búsqueda con retroceso.

El algoritmo comenzará por un *pixel* rojo y lo seguirá hasta encontrarse con uno negro, seguirá el negro hasta una distancia de 6 *pixeles* que se llamará horizonte, en ese caso, si encuentra un *pixel* rojo, etiquetará los anteriores *pixeles* negros a rojos y continuará. En caso de no encontrar ningún *pixel* rojo, parará y extraerá el trazo hasta el último *pixel* rojo encontrado, marcando como visitados los anteriores puntos para no volver a visitarlos en posteriores iteraciones. [26]

Cálculo de la caja de Feret.

A partir de este momento se comienza a trabajar directamente con la lista de trazos descartando de esta manera la imagen de la firma. [19]

El siguiente paso es el cálculo de la caja de *Feret* (*Feret Box*), que se define como el rectángulo que circunscribe al conjunto de los trazos alineado respecto a los ejes cartesianos. Para calcularlo, es necesario recorrer los n' trazos de la firma y guardar las coordenadas de aquellos que tengan como origen o destino la mayor y menor X , y la mayor y menor Y . Al realizar este proceso sobre los trazos de la firma se evita que las manchas o el ruido afecten a los siguientes procesos de escalado y normalización.



Figura 7. Cálculo de la Caja de Feret.

Descripción del algoritmo Ajuste de distancias.

El objetivo de este algoritmo es normalizar los trazos de las dos imágenes que se desean comparar, y luego seleccionar ciertos puntos de control a una distancia normalizada para obtener un trazo con un determinado número de puntos y calcular la suma de las distancias entre los puntos de las dos imágenes. [27]

Comparación.

Se comparan los trazos de la imagen de la firma con los trazos de las firmas almacenadas en la base de datos. Esta comparación tiene en cuenta el color del trazo y la distancia entre el punto origen del trazo y el punto destino de este. Este proceso retorna un resultado expresado en porcentaje de similitud, el cual es comparado con un umbral para decidir si las firmas coinciden.

1.5.2 Itextsharp.

Es una biblioteca original de Java portada a .Net. Permite que se puedan generar archivos con formato Adobe PDF, desde la plataforma .Net. Se trata de un proyecto de código abierto totalmente libre.

1.6 Lenguaje unificado de modelado (UML por sus siglas en inglés).

El UML se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. UML describe un plano del sistema, en el cual se incluyen aspectos conceptuales y concretos. El mismo puede ser utilizado en cualquier metodología, ya que es independiente del ciclo de desarrollo a seguir. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual, que tengan generadores de código, así como generadores de informes. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. [13]

1.7 Herramientas CASE (Computer Aided Software Engineering).

Las herramientas CASE, nacen para auxiliar a los desarrolladores de *software*, lo que permite el apoyo computarizado en todo o en parte del ciclo de vida del desarrollo de un sistema de *software*. Han surgido para solucionar el problema de la mejora de la calidad del desarrollo de sistemas de mediano y gran tamaño, y en segundo término, por el aumento de la productividad. Son muy utilizadas en varias tareas como realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

1.7.1 Rational Rose Enterprise Edition.

Es una herramienta de diseño orientada a objetos, que se encarga de dar soporte al modelado visual, permitiendo de esta forma, representar gráficamente el sistema, haciendo énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un *software* de mayor calidad, empleando un lenguaje estándar común, que facilita la comunicación. Brinda la posibilidad de que varias personas trabajen a la vez, permitiendo que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y permite que tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. [14]

Incluye las siguientes funciones:

- ✓ Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular.
- ✓ Generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código.
- ✓ Funciones de análisis de calidad de código.

- ✓ Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web.
- ✓ Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones a través de diseños lógicos y analíticos.
- ✓ Creación de definiciones de tipo de documentos DTD en XML.
- ✓ Posibilidad de publicar en las Web modelos e informes para mejorar la comunicación entre los miembros del equipo. [14]

1.7.2 Visual Paradigm

Visual Paradigm es una herramienta muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo de *software* que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante; creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de *software* de gran escala con el uso del acercamiento orientado a objeto. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. [20]

Características.

- ✓ Producto de calidad.
- ✓ Soporta aplicaciones Web.
- ✓ Varios idiomas.
- ✓ Generación de código en varios lenguajes.
- ✓ Fácil de instalar y actualizar.
- ✓ Compatibilidad entre ediciones.

Ventajas que ofrece.

- ✓ Apoya todo lo básico en cuanto a: artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- ✓ Tiene apoyo adicional en cuanto a: generación de artefactos automáticamente.
- ✓ Genera modelos VP-UML instantáneamente a partir de código binario .Net.

- ✓ Generación de documentación en formatos HTML y PDF.
- ✓ Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.
- ✓ Brinda la posibilidad de intercambiar información utilizando la exportación e importación de ficheros con aplicaciones, por ejemplo: Visio y Rational Rose.
- ✓ Generación de códigos e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para las plataformas .Net, Java y PHP, así como obtener los diagramas a partir del código.
- ✓ Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

1.7.3 Fundamentación de la herramienta Case utilizada.

Después de haber analizado las características y ventajas que ofrecen las herramientas CASE, se decidió utilizar para la realización del modelado de este trabajo, Visual Paradigm, ya que es una herramienta multiplataforma que 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. Con Visual Paradigm se puede generar código para diferentes lenguajes de programación. [21]

1.8 Sistemas Gestores de Bases de Datos (SGBD).

Un SGBD, es una colección de programas cuyo objetivo es: servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Las características de un Sistema Gestor de Base de Datos SGBD son:

- ✓ **Abstracción de la información:** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- ✓ **Independencia:** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones.
- ✓ **Redundancia mínima:** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

- ✓ **Consistencia:** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- ✓ **Seguridad:** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- ✓ **Integridad:** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de *hardware*, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- ✓ **Respaldo y recuperación:** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar, a partir de estas copias, los datos que se hayan podido perder.[15]

1.8.1 MySQL.

MySQL es un sistema de gestión de bases de datos relacional, bajo licencia GPL (*General Public License*) de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el *copyright* del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es *software* libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un *software* propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del *software* libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. [16]

Características de MySQL:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos. [16]

Desventajas.

MySQL surgió como una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Por esta razón, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

- **Subconsultas:** tal vez esta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
- **Triggers y Procedures:** Se tiene pensado incluir el uso de procedimientos almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- **Transacciones:** a partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
- **Integridad referencial:** aunque admite la declaración de claves ajenas en la creación de tablas, internamente no las trata de forma diferente al resto de los campos.[16]

1.8.2 PostgreSQL.

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [16]

Características principales.

- Herencia de tablas.
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.
- Soporte para consultas con UNION, UNION ALL y EXCEPT.
- Es una base de datos ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).
- Copias de seguridad en caliente (*Online/hot backups*).
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Soporte para distintos tipos de datos, como son: datos de tipo fecha, datos sobre redes (MAC, IP).
- Posee Interfaz de Programación de Aplicaciones (API's) para programar en C/C++, Java, .Net, Perl, Python, PHP.

1.8.3 Justificación del gestor de bases de datos seleccionado.

Como SGBD se utilizará PostgreSQL, ya que es recomendado para sistemas que manejen gran cantidad de información. Sus características más importantes son: la estabilidad, potencia, robustez y facilidad de administración. Utiliza multiprocesos en vez de multihilos para garantizar la estabilidad del sistema, por lo que un fallo en uno de los procesos no afectará el resto y el sistema sigue funcionando. PostgreSQL está diseñado para entornos con altos volúmenes de tráfico/transacciones, presenta herramientas gráficas y de línea de comandos para diseñar bases de datos y administrarlas, requerimientos de administración y mantenimiento relativamente bajos con respecto al resto de bases de datos comerciales. Este gestor tiene un rendimiento excelente, es extensible y multiplataforma.

1.9 Framework de desarrollo

El término *framework* es muy utilizado en el desarrollo de sistemas de *software*. En general, un *framework*, es una estructura del *software* compuesta de componentes personalizables e intercambiables, para el desarrollo de una aplicación. Un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación.

1.9.1 .Net framework 4.5

Se trata de una plataforma que permite la creación de diversos tipos de aplicaciones de escritorio, de servidor y móviles. .NET Framework 4.5 se integra a la perfección en los sistemas operativos Windows. Por otra parte, incluye la biblioteca de clases base (BCL) orientada a objetos y trae consigo varias

características y nuevas mejoras por ejemplo: incluye rendimiento mejorado, mayor control, una nueva biblioteca de flujo de datos, y mejor compatibilidad para la depuración y el análisis de rendimiento del *software*. [32]

A continuación se muestran otras características importantes de .NET Framework 4.5:

- ✓ Capacidad de búsqueda de flujo de trabajo mejoradas en Visual Studio, incluida búsqueda rápida y buscar archivo.
- ✓ Mejora en el tiempo de inicio de la aplicación y la utilización de memoria (tamaño del espacio de trabajo).
- ✓ Compatibilidad con los tipos de datos genéricos.

1.10 Visual Studio 2012.

Visual Studio 2012 está diseñado para mejorar el entorno de trabajo. Posee una interfaz que ha sido rediseñada para simplificar el flujo de trabajo y brindar un fácil acceso a las herramientas. Entre sus principales características se encuentran: [30]

- Barras de herramientas están simplificadas.
- Reducción del desorden de pestañas.
- Fácil navegación por la aplicación.
- Es totalmente compatible con las nuevas versiones de Windows. [30]
- Ofrece nuevas plantillas, pantallas de diseño y herramientas de evaluación y depuración. [30]

Compatibilidad con la plataforma de desarrollo.

Visual Studio 2012 permite trabajar en una gran variedad de plataformas, entre las que se incluyen Windows, Windows Server, Web, Cloud, Office y SharePoint, todo en un único entorno de desarrollo integrado.

1.11 Lenguaje de programación seleccionado.

C# 5.0 es un lenguaje de programación incluido en el *.NET Framework 4.5*. Una de sus principales características es que brinda la opción para programación dinámica. Es un lenguaje de programación

sencillo, con seguridad de tipos y orientado a objetos. [29] C# sigue siendo esencialmente de tipo estático, aunque se ha incorporado la capacidad de interactuar con objetos dinámicos.

Ofrece a los desarrolladores herramientas centradas en código, compatibilidad de lenguajes para crear aplicaciones Web, completas y conectadas en .NET *Framework*. También incluye soporte para la programación orientada a componentes. [17]

1.12 Propuesta de selección de tecnologías y herramientas.

Luego de analizar varias tecnologías, metodologías y herramientas que son utilizadas hoy en día para el desarrollo de *software* y de haber visto sus principales características, ventajas y desventajas, se seleccionaron un conjunto de estas para desarrollar este sistema.

Como lenguaje de programación se propone utilizar C# 5.0, para apoyar la implementación, se utilizará Visual Studio 2012, como plataforma de desarrollo .Net *Framework* 4.5, como sistema gestor de base de datos PostgreSQL, como herramienta de modelado: Visual Paradigm y por último como metodología XP.

1.13 Conclusiones Parciales.

Con el estudio y análisis realizado a los sistemas de identificación de personas mediante la firma manuscrita se fundamentaron las bases teóricas de la investigación, obteniendo los conocimientos sobre las características, así como, el correcto funcionamiento de estos sistemas. A partir de la investigación realizada se evidencia la necesidad de desarrollar un sistema que permita identificar a una persona mediante su firma manuscrita, debido que los sistemas existentes a nivel internacional no brindan información sobre las tecnologías y herramientas que utilizan. El estudio realizado a diferentes metodologías, tecnologías y herramientas permitió definir cuáles utilizar y de esta manera lograr un mejor desarrollo del *software*.

Capítulo II: Características, Análisis y Diseño del sistema.

En el desarrollo de este capítulo se estará tratando la propuesta de solución para dar respuesta al problema a resolver y todo lo referente al diseño del sistema, además se definen los requisitos funcionales y no funcionales, donde los primeros se describen en las Historias de Usuario (HU). También se realiza el plan de entrega en el cual se indican las historias de usuario que se crearán para cada iteración de la aplicación. Es confeccionado el plan de iteraciones donde se muestran las HU que se realizarán en cada iteración según su prioridad en el negocio.

2.1 Objeto de Informatización

El objeto de informatización es el proceso de identificación de personas mediante la firma manuscrita, para lograr esto, se pretende desarrollar un sistema que permita obtener la firma de la persona comparándola con otras y de esta forma poder identificar su autor. La automatización de este proceso permitirá garantizar, rapidez y fiabilidad a la hora de realizar dicho proceso.

2.2 Propuesta del sistema.

El sistema estará basado en una aplicación de escritorio, que permitirá identificar a una persona mediante su firma manuscrita, esto será posible una vez que se obtenga la imagen de la firma de un documento o cargar la imagen desde un directorio. El sistema contará con una base de datos en la que estarán almacenadas imágenes de firmas de diferentes personas, junto con la lista de trazos correspondientes a cada imagen almacenada.

El primer paso en el proceso de identificación de personas es recortar la imagen de la firma del documento o cargar esta imagen de un directorio, una vez obtenida la imagen de la firma, se realiza el pre-procesamiento a esta imagen con el objetivo de hacer una limpieza y eliminación del ruido.

Después de haber terminado el primer paso se procede a extraer las características de la imagen de la firma, dando como respuesta una lista de trazos para dicha imagen. En ambos pasos se utiliza la librería de clases ImagesWorking.

Una vez obtenida la lista de trazos de la firma de la persona que se desea identificar, son comparados con otros trazos pertenecientes a otras firmas almacenadas en la base de datos, dando como resultado, un porcentaje de similitud, el cual es comparado con un umbral ya definido. En caso de ser identificada la persona, se mostrarán los datos de la misma como son: nombre, apellidos, carné de identidad y sexo.

El sistema posibilitará al usuario consultar los reportes de comparación de firmas y exportar estos reportes para su posterior análisis.

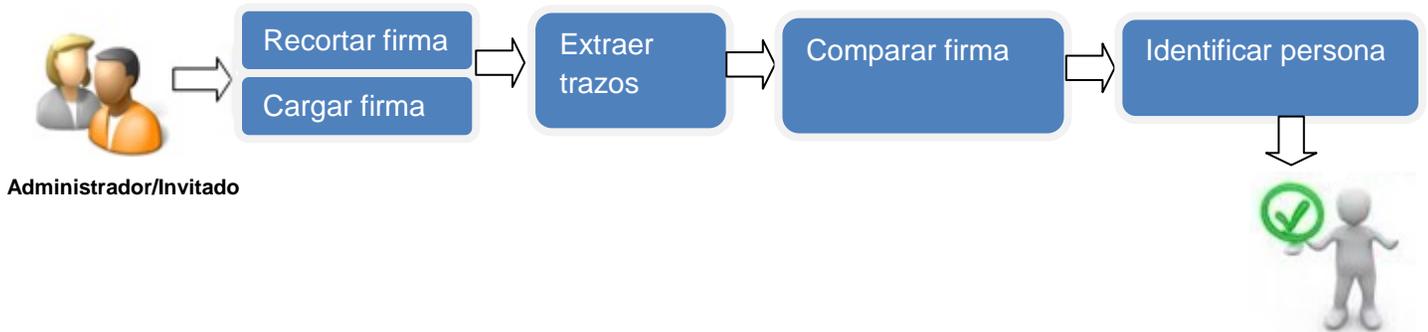


Figura 8. Etapas del sistema de identificación de personas mediante firmas manuscritas.

2.3 Modelo de Dominio.

El modelo de dominio expresa los principales conceptos con los que trabaja la aplicación, lo que ayuda a entender de una manera más fácil el funcionamiento del sistema. También contribuye a la comprensión de los requisitos funcionales y no funcionales que se derivan de él.

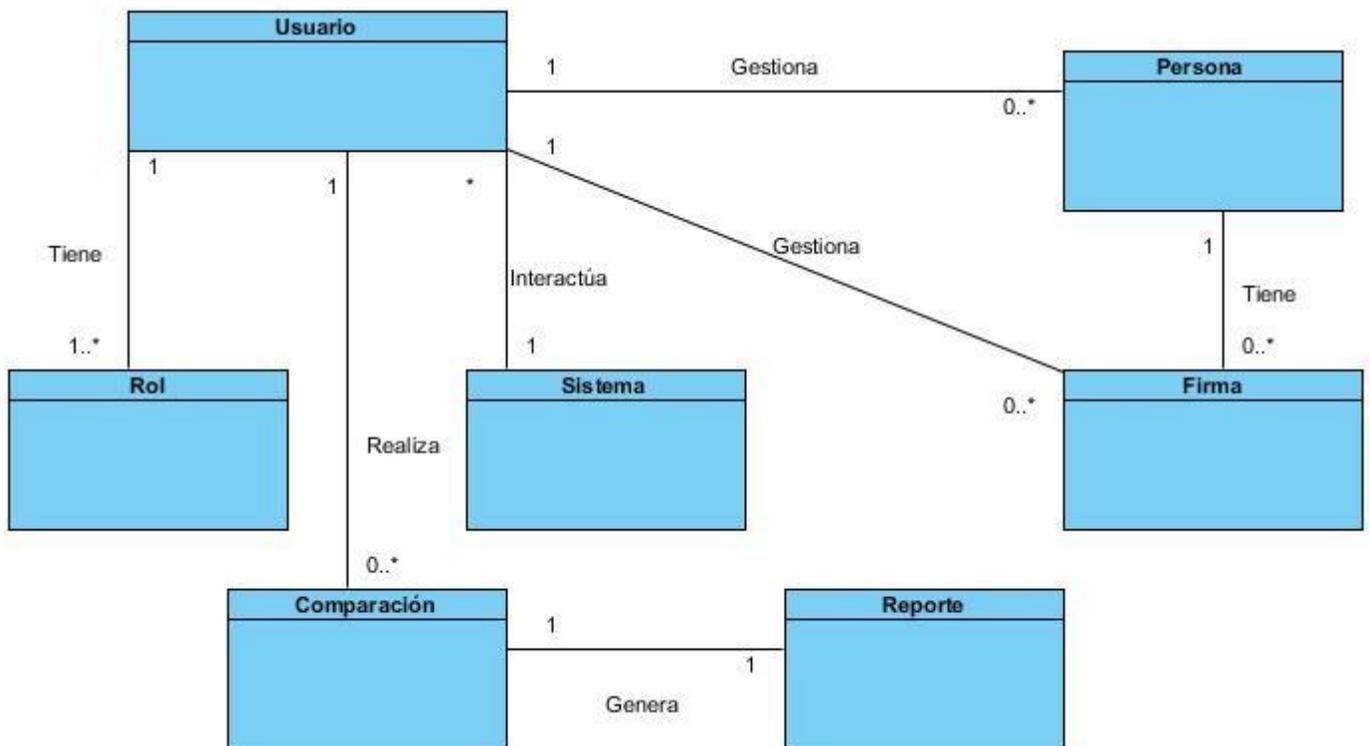


Figura 9. Modelo de dominio.

Elementos del modelo de dominio:

Sistema: Aplicación que sirve como interfaz a la base de datos, para realizar las operaciones sobre la misma.

Firma: Rasgo característico de una persona, utilizado para la identificación del firmante.

Persona: Individuo que provee la firma.

Usuario: Son los actores que interactúan con el sistema, y en dependencia del rol que posean podrán tener acceso a unas u otras funcionalidades.

Rol: Función o papel que cumple un usuario.

Reporte: Informe sobre los resultados de las comparaciones.

Comparación: Acción que se realiza para descubrir diferencias o semejanzas entre dos o más firmas.

2.4 Requisitos Funcionales.

En la siguiente tabla se hace referencia, a las funcionalidades que va a tener el sistema.

Tabla 1. Funcionalidades del sistema.

Funcionalidades
Autenticar usuario
Gestionar usuario
Modificar perfil
Gestionar persona
Gestionar firma de persona
Obtener firma del documento
Comparar firma

Identificar persona
Mostrar reportes de comparación de firma
Exportar a PDF reportes de comparación de firma

2.5 Requisitos no Funcionales.

Los requisitos no funcionales son propiedades que el producto debe tener. Estas propiedades hacen que el producto sea más confiable y rápido. A continuación se muestran estos requisitos:

Software

- ✓ El sistema será usado bajo los Sistemas Operativos Windows XP o una versión superior.
- ✓ *Framework* .NET 4.5.

Requisitos de hardware

Requisitos de la Pc cliente:

- ✓ Procesador: 1GHz
- ✓ RAM: 512 MB

Requisitos para el servidor PostgreSQL:

- ✓ Procesador: 1GHz
- ✓ RAM: 1GB

Restricciones en el diseño y la implementación

- ✓ Lenguaje de programación: C# 5.0.
- ✓ *Framework* de desarrollo que se utilizará es: *.NET Framework* 4.5.
- ✓ IDE: Visual Studio 2012.
- ✓ Para el Modelado de UML se utilizará: Visual Paradigm 8.0.
- ✓ Gestor de base de datos: PostgreSQL 9.1.

Apariencia o interfaz externa

- ✓ El sistema debe tener una interfaz sencilla, agradable, con colores suaves y sin aglomeración de imágenes que distraigan al cliente.

Usabilidad

- ✓ El uso del diseño debe ser fácil de entender, atendiendo a la experiencia, conocimiento y habilidades de cualquier usuario.
- ✓ La aplicación estará escrita en español.

2.6 Metáfora.

Después de haberse definido las funcionalidades que el sistema debe cumplir y los requerimientos no funcionales, se crea la metáfora, esta es una breve descripción de cómo debe funcionar el sistema y se encarga de regir todo el desarrollo del *software* como una gran historia de usuario, ayudando de esta manera a entender los elementos fundamentales del sistema.

El sistema de identificación de personas mediante firma manuscrita, funcionará en aquellas entidades que deseen hacer uso del mismo. Desde el instante que se recorte una firma de un documento o se obtenga esta de un directorio en la computadora, se podrá realizar la comparación de la misma con otras almacenadas en una base de datos de las cuales ya se conoce su autor. Mediante esta comparación la aplicación siempre devolverá una respuesta que puede ser satisfactoria, por ejemplo: identificando el autor de la firma y mostrando sus datos, o mostrará un mensaje de error en caso de que la firma no coincida con ninguna de las almacenadas en la base de datos. Además, el sistema proporcionará al usuario la opción de consultar los reportes de comparación de firmas, dichos reportes podrán ser exportados en formato PDF para su posterior análisis.

2.7 Fase de Exploración.

La metodología XP comienza con la fase de exploración, en la cual se define el alcance general del proyecto. En esta fase, el cliente precisa lo que necesita por medio de las Historias de Usuario (HU), al mismo tiempo, el equipo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

2.7.1 Historias de Usuario (HU).

Las HU son utilizadas en XP para especificar los requisitos del *software* desde el punto de vista del cliente, es decir, estas historias son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Las HU deben tener el detalle mínimo para que el programador pueda realizar una estimación poco riesgosa del tiempo que llevará el desarrollo del *software*.

A continuación se describen las HU, donde cada una de ellas responde a las principales funcionalidades

dando una idea de cómo debe ser su posterior implementación, las restantes HU se pueden observar en el *anexo 1*.

Tabla 2. HU_Gestionar firma de persona.

Historia de Usuario	
Número: 6	Usuario: Administrador , Invitado
Nombre historia: Gestionar firma de persona.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Omar Barzaga Angulo	
Descripción: El administrador o el invitado tendrán la posibilidad de insertar una o varias firmas a una persona y mostrarlas.	
Observaciones: Gestionar firma incluye: insertar y listar firmas.	

Tabla 3. HU_Comparar firma

Historia de Usuario	
Número: 8	Usuario: Administrador , Invitado
Nombre historia: Comparar firma.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite al administrador o al invitado comparar una firma con otras que se	

encuentran almacenadas en la base de datos.

2.8 Fase de Planificación.

La Planificación es una fase corta en la cual el cliente establece la prioridad de cada historia de usuario y correspondientemente, el programador realiza una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

2.8.1 Estimación del esfuerzo por Historia de Usuario.

Los programadores estiman el tiempo necesario para desarrollar cada historia de usuario, este valor se expresa en semanas. Con el transcurso de las iteraciones, se irá acercando a la realidad.

Tabla 4. Estimación de esfuerzo.

Historia de Usuario	Estimación
Autenticar usuario	1
Gestionar usuario	2
Modificar perfil	1
Gestionar persona	2
Gestionar firma de persona	2
Obtener firma del documento	3
Comparar firma	3
Identificar persona	2
Mostrar reportes de comparación de firma	1
Exportar a PDF reportes de comparación de firma	1
Total	18

2.8.2 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto usando la metodología XP se crea el plan de duración de cada una de las iteraciones que se han definido, que tiene como objetivo mostrar la duración y el orden en

que serán implementadas las historias de usuario dentro de cada iteración. Para la solución se han definido 10 historias de usuarios divididas en 2 iteraciones, de acuerdo con los intereses del cliente, para una duración total del proyecto de 18 semanas.

- **Iteración 1:** En esta iteración se pretende diseñar, implementar y probar algunas de las historias de usuarios como por ejemplo: Gestionar usuarios, Gestionar personas, Modificar perfil, Autenticar usuarios y Gestionar firma de persona. Una vez que se termine de implementar y probar todas las funcionalidades de esta iteración se liberará un ejecutable el cual será evaluado con el fin de eliminar posibles errores.
- **Iteración 2:** Esta iteración tiene como objetivo principal diseñar, implementar, y probar las historias de usuarios de mayor prioridad, las cuales son: Obtener la firma de la persona, Comparar firma, Identificar persona, Mostrar reportes de comparación de firma y Exportar a PDF los reportes de comparación de firma. Después de haber terminado el diseño, la implementación y la prueba de estas historias de usuarios se tendrá la versión 1.0 de la aplicación.

Tabla 5. Plan de iteraciones.

Iteración	No. HU	Historia de Usuario	Duración estimada (semana)	Duración total de Iteración (semana)
Iteración 1	HU_1	Autenticar usuario	1	8
	HU_2	Gestionar usuario	2	
	HU_3	Modificar perfil	2	
	HU_4	Gestionar personas	1	
	HU_5	Gestionar firma de persona	2	
Iteración 2	HU_6	Obtener firma del documento	3	10
	HU_7	Comparar firma	3	
	HU_8	Identificar persona	2	
	HU_9	Mostrar reportes de comparación de firma	1	
	HU_10	Exportar a PDF reportes de comparación de firma	1	

2.8.3 Plan de entrega.

Una vez que se concluye la tarea de elaborar las distintas historias de usuarios, se comienza con la creación del Plan de entrega, para estimar el tiempo de desarrollo de las mismas. Este artefacto se elabora con la intención de fijar, que período de tiempo puede tardar la implementación de cada una de las historias, definiéndose las fechas en que serán liberadas las versiones funcionales del producto.

Tabla 6. Plan de entrega.

Entregable	Fin Iteración 1	Fin Iteración 2
Sistema de identificación de persona mediante firma manuscrita.	Marzo 2013	Mayo 2013

2.9 Diseño.

El diseño crea una estructura que organiza la lógica del sistema, este debe ser incremental y debe estar incrustado en el *software*. Hay que diseñar lo que las necesidades del problema requieren y no como se piensa que debe ser el diseño. La metodología XP plantea diseñar lo más sencillo posible, para hacer solo lo imprescindible en un momento dado.

2.9.1 Modelo de Datos.

Es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de integridad, las cuales definen un conjunto de estados consistentes. El modelo de datos puede ser usado como una herramienta para especificar los tipos de datos y la organización de los mismos, además para la manipulación de consultas y datos, así mismo es el elemento clave en el diseño de la arquitectura de un manejador de la base de datos (BD).

A continuación se muestra el modelo de datos de la propuesta de solución, la descripción de las tablas que componen este modelo se pueden ser observadas en el *anexo 2*:

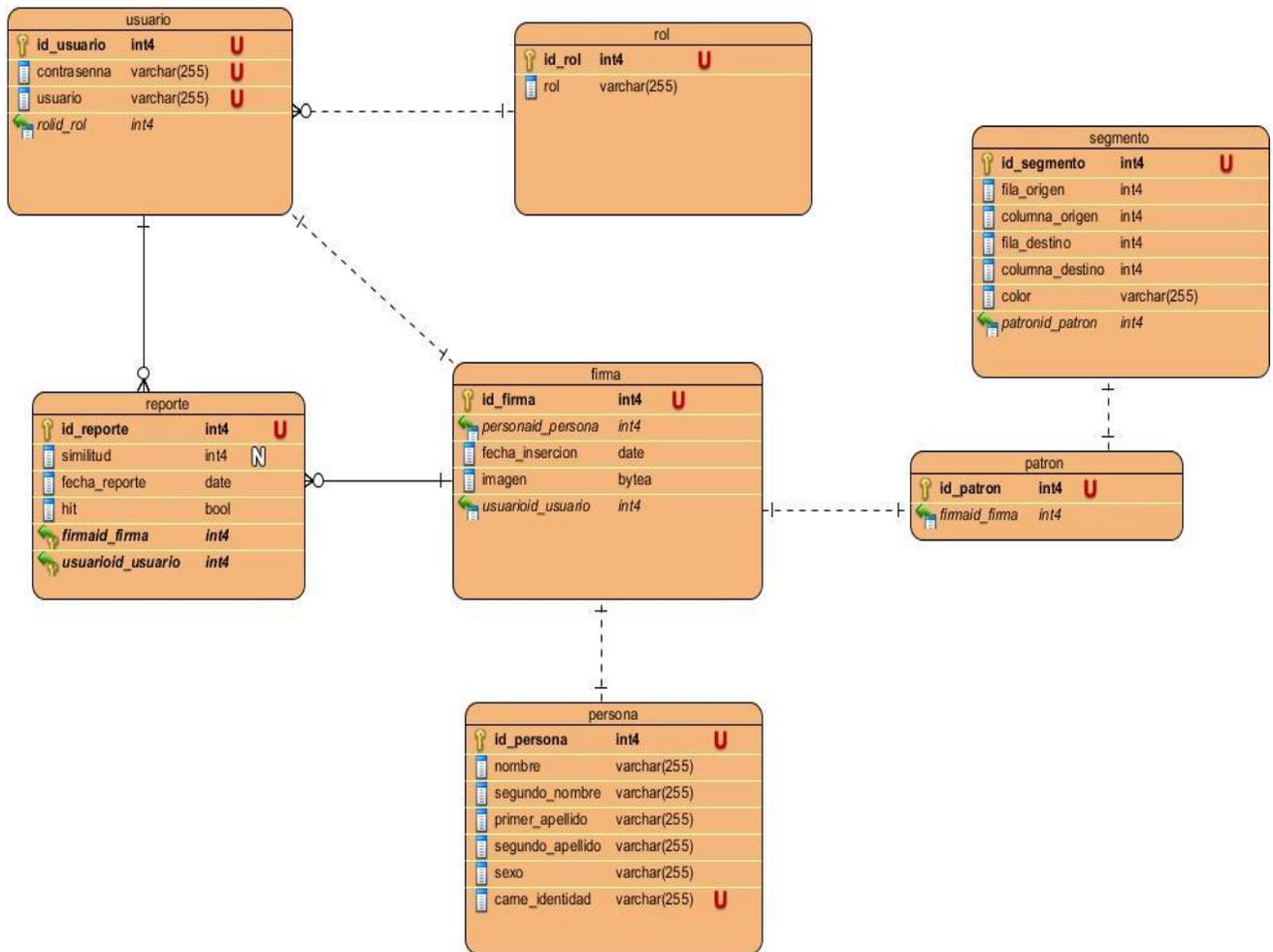


Figura 10. Modelo de datos.

2.9.2 Arquitectura.

La arquitectura de *software*, es una vista del sistema que incluye los principales componentes del mismo, alcanzando a través de los mismos, la visión del sistema. Establece además, los fundamentos para que analistas, diseñadores, programadores y otros roles, trabajen en una línea común que permita alcanzar los objetivos del sistema, cubriendo todas las necesidades. [34]

Como línea base, para la organización estructural del sistema de identificación de persona mediante firma manuscrita, se utilizó la arquitectura N-capas. Esta arquitectura, se basa en una distribución jerárquica de roles y responsabilidades, para proporcionar una división efectiva de los problemas a resolver. Los roles indican: el tipo y la forma de interacción con otras capas y las responsabilidades: la funcionalidad que implementan. Dividiéndose el sistema en las siguientes capas: [33]

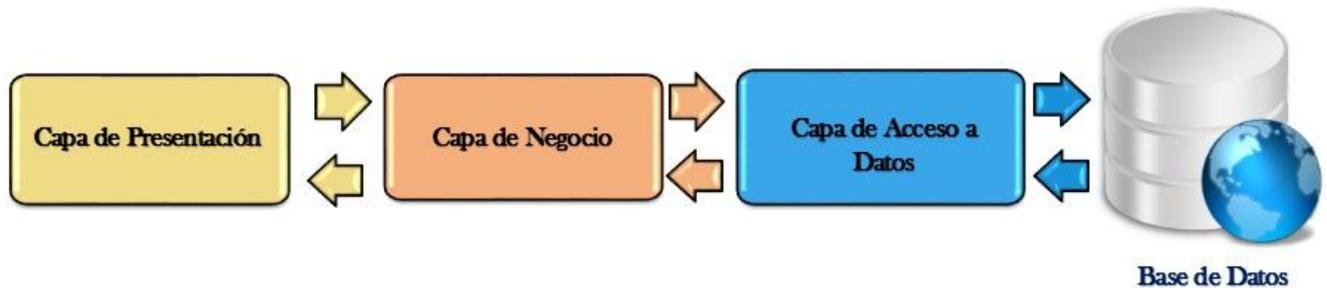


Figura 11. Arquitectura N-capas.

- **Capa Presentación:** Es la que presenta el sistema al usuario, le comunica la información y captura los datos que el usuario introduce en el sistema. Esta capa se comunica únicamente con la del negocio enviando peticiones a la misma. Tiene la característica de ser entendible y fácil de usar. [33]
- **Capa del Negocio:** Es donde residen los programas que se ejecutan durante la ejecución de la aplicación. Esta capa se comunica con la de presentación, para recibir las peticiones y presentar los resultados y con la de acceso a datos, para interactuar con la base de datos, consultando y modificando su información. En la misma se encuentran las clases del negocio que no son más que aquellas que permiten la gestión de usuarios y de personas, se encuentra además, el componente de extracción y verificación de características *off-line* de la firma manuscrita. [33]
- **Capa de Acceso a Datos:** Es donde residen los datos y la encargada de acceder y modificar a los mismos. Está formada por uno o más sistemas de gestión de bases de datos que realizan todo el almacenamiento de informaciones y que reciben solicitudes de almacenamiento o recuperación de datos desde la capa de negocio. [33]

2.9.3 Patrones de diseño utilizados.

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.

- El patrón **Experto** define como asignar de forma adecuada las responsabilidades en un modelo de clases. Este indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer en la clase que conoce toda la información necesaria para crearlo. Entre sus beneficios se encuentra que permite obtener un diseño con mayor cohesión y conservar el encapsulamiento. [21]

- El patrón **Creador** guía la asignación de responsabilidades de la creación de objetos, el propósito fundamental de este patrón es encontrar un creador que deba conectar con el objeto producido en cualquier evento. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creadora. [21]
- El patrón **Alta cohesión** plantea que la información que almacena una clase debe de ser coherente y debe estar en la medida de lo posible relacionada con la clase. [21]
- El patrón **Controlador** sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación lo que posibilita aumentar la reutilización de código y a la vez tener un mayor control. [21]
- El patrón **Bajo Acoplamiento** 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. [21]

Patrones GOF.

- Singleton

Este patrón 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. A continuación se muestra como se evidencia este patrón en la aplicación: [21]

```
public static Controladora ObtenerInstancia()
{
    if (control == null)
        control = new Controladora();
    return control;
}
```

Figura 12 Patrón Singleton

Patrón de Estructural.

- Objeto de Acceso a Datos (DAO)

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

Patrones de diseño de Base de Datos.

- Patrón de asociación.

Este patrón representa las relaciones entre entidades del modelo, utilizado para representar asociaciones de tipo: muchos a muchos. Las bases de datos relacionales no soportan relaciones directas de este tipo. Esta situación puede ser resuelta creando una tabla en el modelo de la BD que contenga las claves primarias de dos tablas involucradas en este tipo de relación, mostrando así su asociación, sin afectar la funcionalidad de lo que se quiere representar. La tabla resultante como se puede observar en la Figura. 13, estará compuesta por llaves primarias y foráneas al mismo tiempo.

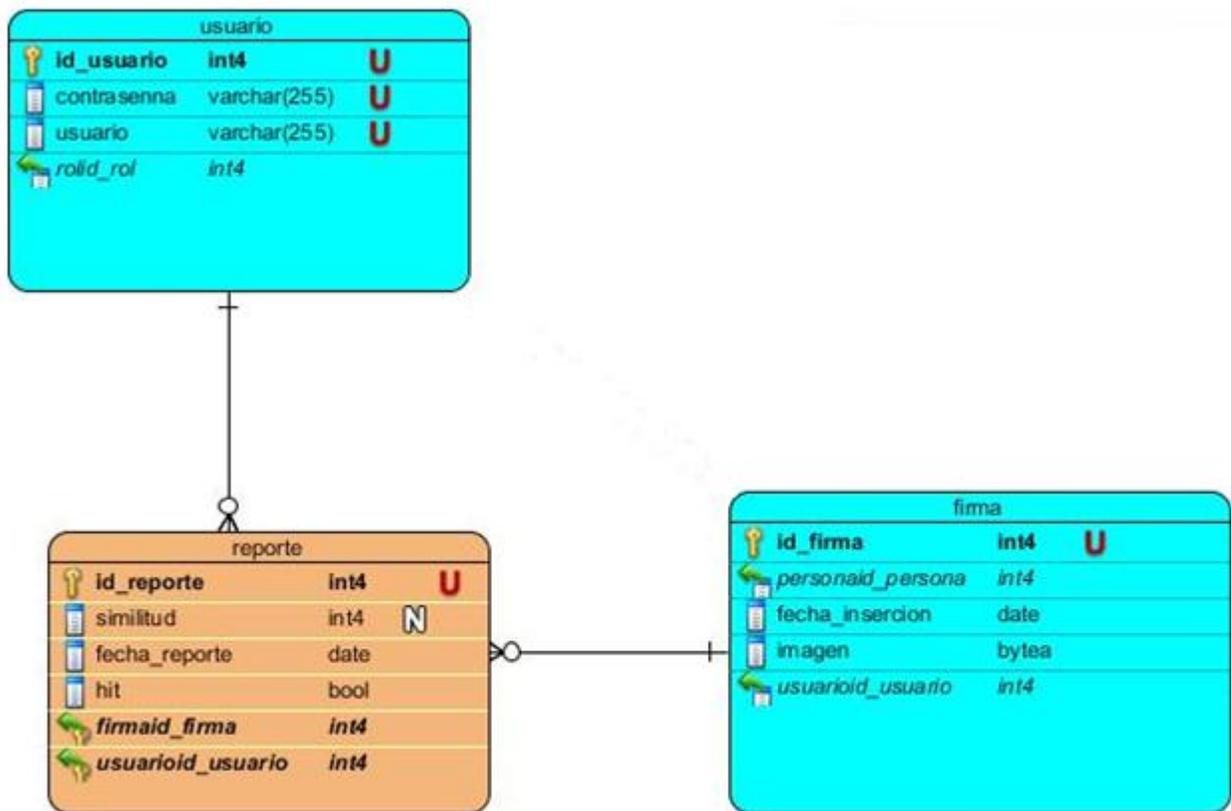


Figura 13. Patrón de asociación.

- Patrón de llaves subrogadas.

Este patrón fue empleado en la BD, para generar llaves primarias únicas con el objetivo de conducir el diseño a implementaciones más eficientes. Garantizando una absoluta certeza de que su utilización, no será susceptible de cambiar con el tiempo, facilitando así el manejo de las relaciones en lugar de dificultarlo. Este patrón es muy utilizado por las entidades que integran el sistema que se desarrollará. Un ejemplo de este patrón se puede observar en la siguiente figura:



Figura 14. Patrón de llaves subrogadas.

2.10 Conclusiones Parciales.

Con el desarrollo de este capítulo, se logró especificar, las características fundamentales de la propuesta de solución. Se definieron las historias de usuarios que caracterizan la solución, además se especificaron los requisitos funcionales y no funcionales, los cuales proponen las condiciones que se deben tener en cuenta, para desarrollar el sistema. También se realizó un análisis de los patrones de diseño, que fueron utilizados en la aplicación y se pudo obtener una visión para definir la arquitectura, además se confeccionaron otros artefactos por ejemplo el modelo de datos y modelo de dominio.

Capítulo III: Implementación y Prueba del sistema.

En este capítulo, se lleva a cabo el desarrollo de uno de los artefactos generados por la metodología XP, las tarjetas CRC, las cuales expresan el diseño del sistema. A través de los diagramas de componente y de despliegue se ofrece una panorámica del funcionamiento del sistema y como ocurren los principales flujos de procesos. Se visualizan las interfaces de usuario con el objetivo de lograr un mayor entendimiento del sistema, realizando además, las pruebas donde se valida el funcionamiento de los requisitos funcionales.

3.1 Tarjetas CRC (Clase-Responsabilidad-Colaborador).

Uno de los artefactos que se generan según la metodología XP son las tarjetas CRC. El objetivo de la misma, es hacer por medio de las tarjetas, un inventario de las clases necesarias para implementar el sistema y la forma en que van a interactuar, de esta forma, se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto, con el objetivo de que el diseño sea lo más simple posible, verificando las especificaciones del sistema.

Un esquema típico de tarjeta CRC, puede ser aquel en el que se indiquen los siguientes datos:

- Nombre de la clase.
- Nombre de las superclases y subclases (si procede).
- Las responsabilidades de la clase.
- Las clases con las que va a colaborar para poder realizar las responsabilidades indicadas.
- Autor, fecha.

A continuación se muestra la Tarjeta CRC correspondiente a la clase Controladora, las restantes tarjetas CRC se pueden observar en el anexo 3.

Tabla 7. Tarjeta CRC - Controladora.cs

Clase: Controladora.cs	
Responsabilidad	Colaboran

<p>Es la encargada de controlar todas las funcionalidades del sistema así como:</p> <ul style="list-style-type: none"> • Autenticar usuario. • Obtener firma de la persona • Insertar reporte. 	<p>ImagesWorking.dll FirmaDAO.cs PatronesDAO.cs ReporteDAO.cs PersonaDAO.cs RolDAO.cs SegmentoDAO.cs Firma.cs Patron.cs Persona.cs Reportes.cs Rol.cs Usuario.cs</p>
---	--

3.2 Tareas de Ingeniería.

Las historias de usuarios, describen a un alto nivel lo que se necesita implementar, pero no brindan los detalles suficientes para poder hacer una codificación correcta de lo que se necesita. Por lo que se dividen todas las historias de usuarios en tareas de programación, las mismas, especifican con detalle lo que deben realizar los programadores para cada una de las historias de usuarios. (Ver Anexo 4)

3.3 Estándar de Codificación.

Un estándar de codificación completo, comprende todos los aspectos de la generación del código. Al comenzar un proyecto de *software*, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto, trabajen de forma coordinada. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad, es establecer un estándar de codificación, sobre el que se efectuarán luego, revisiones del código.

Tabla 8. Estándares de codificación.

Estilos	Descripción
Pascal	La primera letra en el identificador y la primera letra de cada subsiguiente palabra conectada se capitaliza. Se utiliza en caso de identificadores con tres o más caracteres. Por ejemplo: <i>ArregloImagen</i> .

Camello	La primera letra en el identificador está en minúscula y la primera letra de cada subsiguiente palabra conectada en mayúscula. Por ejemplo: <i>personaAux</i> .
Mayúscula	Todas las letras en el identificador se capitalizan. Esta convención se utilizara solo para identificadores que constan de dos o menos letras. Por ejemplo: <i>System.Text</i> .

3.3.1 Reglas de Codificación.

- El código fuente debe ser escrito en español.
- Se debe evitar las líneas de más de 80 caracteres, ya que no son bien manejadas por muchas herramientas.
- Cada funcionalidad debe tener comentario de su funcionamiento.

Tabla 9. Reglas de codificación.

Tipos de identificadores	Reglas de nombre	Ejemplos
Clases o Interfaces.	Los nombres de las clases o interfaces debe tener la primera letra de cada palabra en mayúscula.	Firma
Métodos.	Los nombres de los métodos deben reflejar la acción a realizar y siempre comenzando con letra mayúscula, en caso de ser compuesto ambas palabras en mayúscula.	InsertarFirma
Variables.	Todas las variables empezarán con minúscula y la primera letra de las siguientes palabras en mayúsculas.	firmaAux

3.4 Diagrama de Componente.

Un diagrama de componentes representa, cómo un sistema de *software* es dividido en componentes. Debido a que estos diagramas son más parecidos a los diagramas de casos de usos, son utilizados para modelar vista estática y dinámica de un sistema, además, estos diagramas muestran la organización y las dependencias entre un conjunto de componentes, los cuales incluyen: archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes. Los diagramas de componentes prevalecen en el campo

de la arquitectura de *software*, pero pueden ser usados para modelar y documentar cualquier arquitectura de un sistema.

A continuación se muestra el diagrama de componente de la solución para el sistema que se llevará a cabo, el cual refleja todos aquellos componentes que conforman la solución propuesta, correspondiéndose los mismos con la arquitectura definida para su implementación.

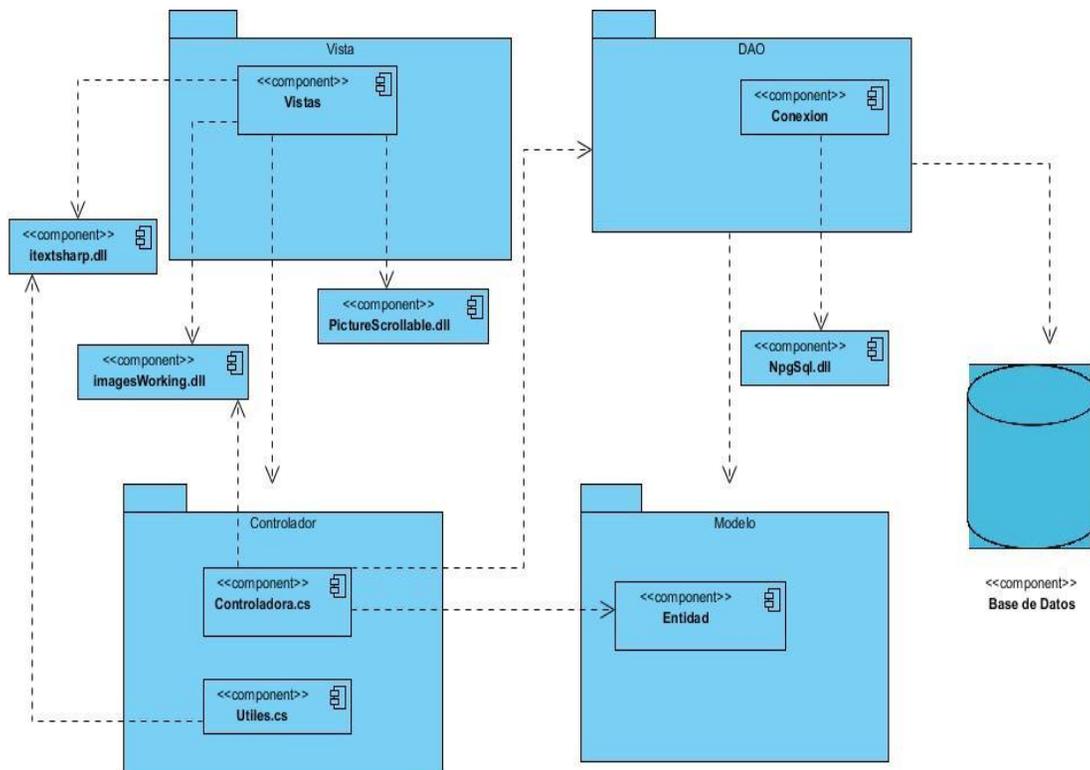


Figura 15. Diagrama de componentes.

3.5 Diagrama de Despliegue.

Los diagramas de despliegue, son los complementos de los diagramas de componentes, que unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de *hardware* y el *software* que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP.

A continuación se muestra el diagrama de despliegue del sistema:

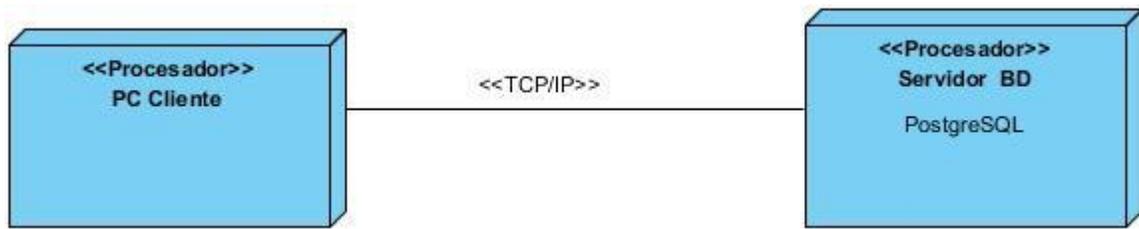


Figura 16. Diagrama de despliegue.

3.6 Interfaz de Usuario.

El diseño de interfaces de usuario, es una tarea que ha adquirido relevancia en el desarrollo de un sistema, ya que la calidad de la interfaz de usuario, puede ser uno de los motivos que conduzcan a su aceptación. La interfaz de usuario de un programa es un conjunto de elementos que presentan información al usuario y le permiten interactuar con la información y con la computadora. Es de vital importancia que las interfaces sean amigables y sencillas de comprender, pues de ello depende la aceptación del sistema por parte de los usuarios. [27]

A continuación se muestran algunas de las interfaces de usuario del sistema, las restantes se pueden observar en el *anexo 5*.

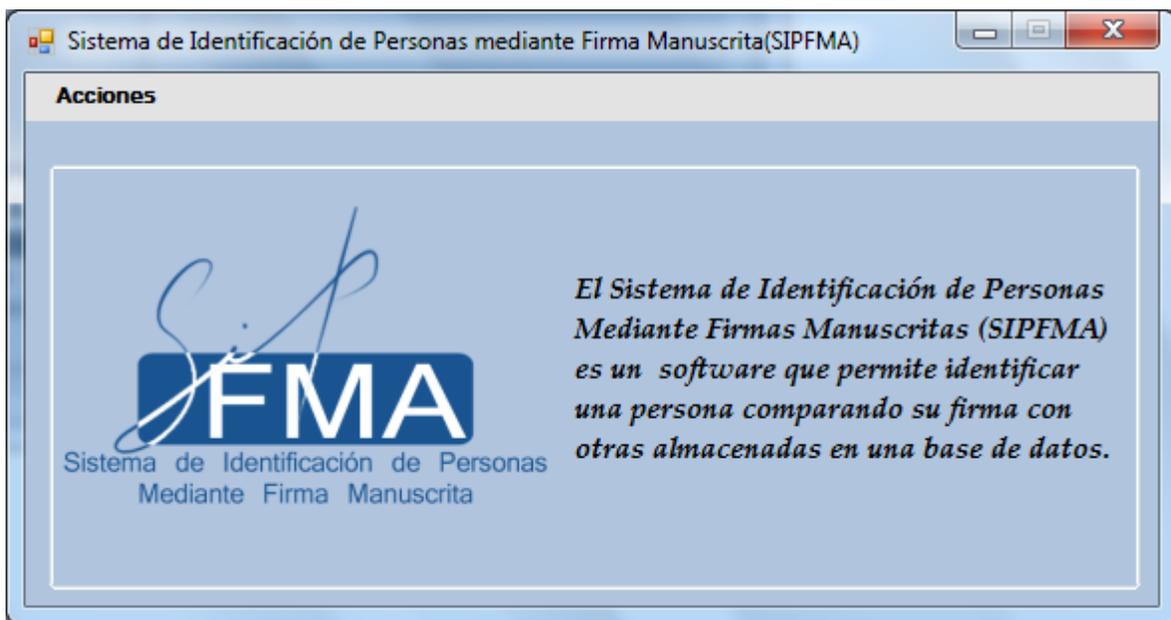


Figura 17. IU_Sistema.

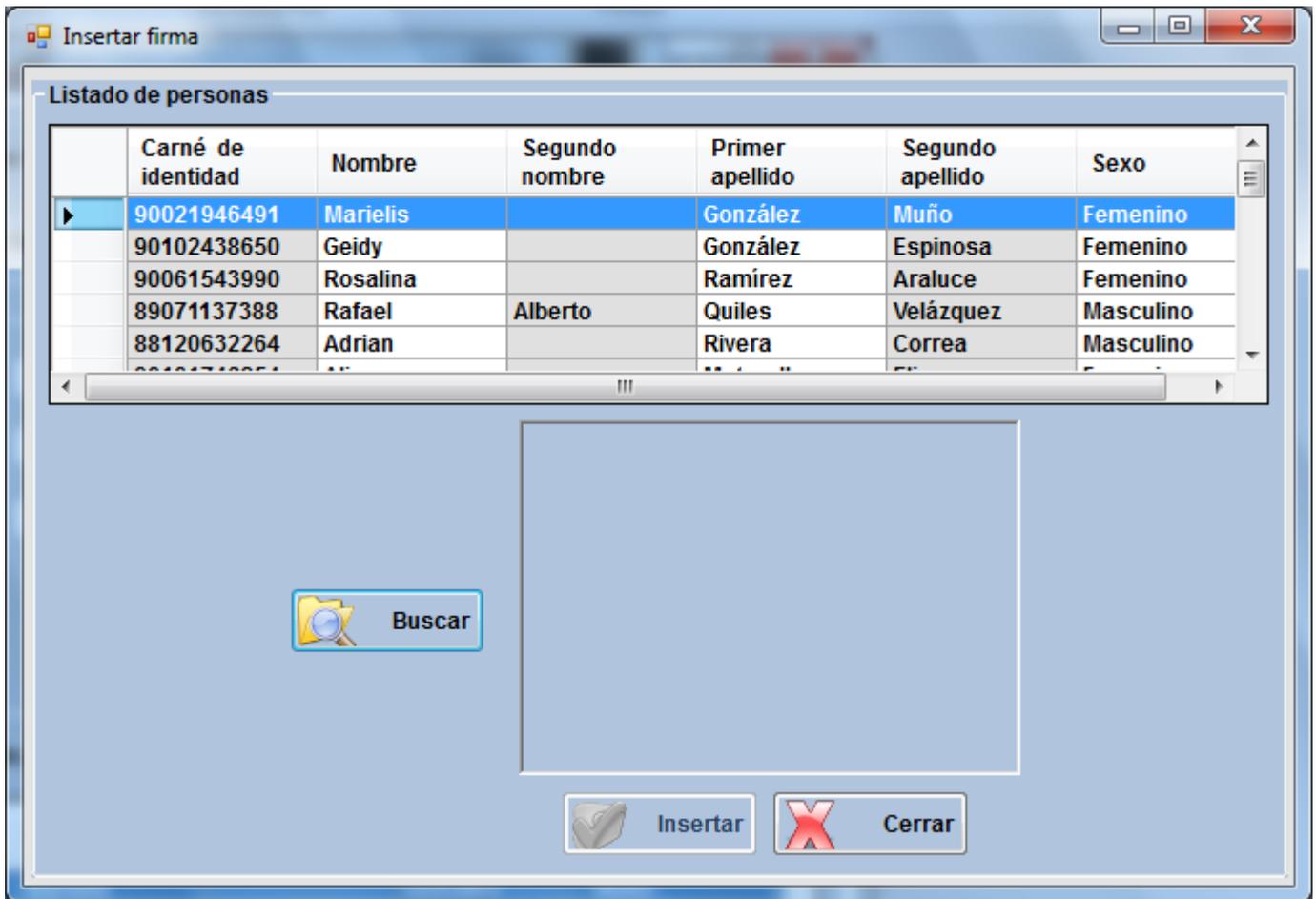


Figura 18. IU_Insertar firma.

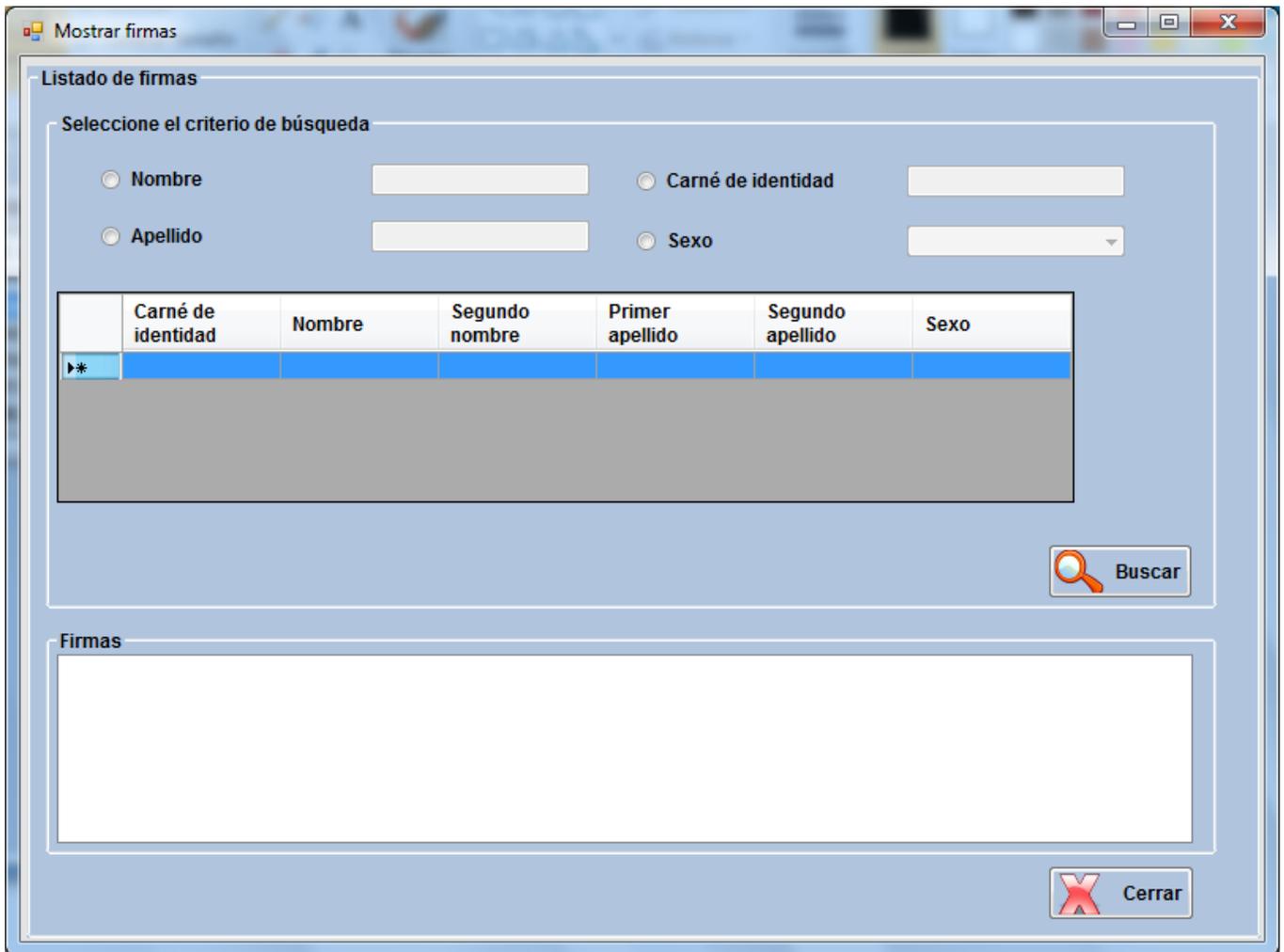


Figura 19. IU_Mostrar firmas.

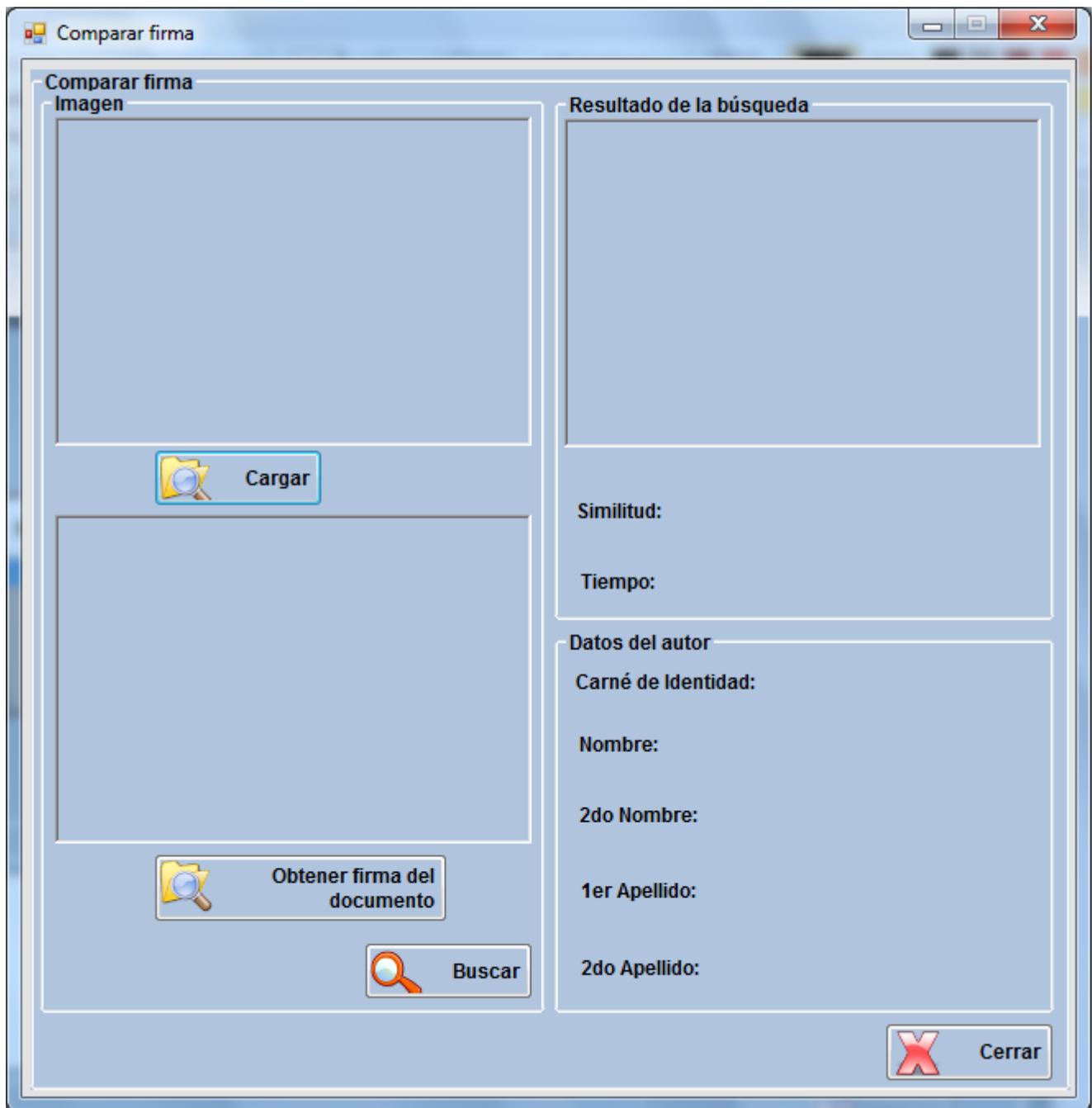


Figura 20. IU_Comparar firmas.

3.7 Pruebas.

La metodología XP pone la comprobación como el fundamento del desarrollo. La realización de pruebas permite aumentar la calidad del producto y disminuir la cantidad de errores que puedan ir apareciendo en

el transcurso del desarrollo del *software*. Le da una idea al cliente de lo que está hecho y favorece la comunicación entre el cliente y el equipo de desarrollo, también permite aumentar la seguridad de evitar efectos no deseados a la hora de realizarle modificaciones al código.

XP divide las pruebas del sistema en dos grupos: pruebas caja blanca o estructural y pruebas de caja negra o funcional.

3.7.1 Pruebas de Caja Blanca o Estructural.

Las pruebas de caja blanca, intentan garantizar que se ejecuten al menos una vez todos los caminos independientes de cada módulo, así como que sean utilizadas las decisiones en su parte verdadera y en su parte falsa. [28]

Prueba del camino básico.

La prueba del camino básico, es una técnica de prueba de Caja Blanca, permite obtener una medida de la complejidad lógica de un diseño y usa esta medida, como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes, por los cuales, puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Aplicación de la prueba al método *ObtenerPatrones*.

```
public static List<Pattern> ObtenerPatrones()
{
    List<Pattern> patrones = new List<Pattern>();//1
    Conexion conexion = Conexion.obtenerConexion();//1
    string consulta = "Select * from public.patron";//1

    IList<object[]> listaPatrones = conexion.EjecutarConsultaArreglo(consulta);//1
    foreach (object[] obj in listaPatrones)//2
    {
        var patron = new Pattern((int)obj[0], (int)obj[1]);//3

        var cyan = SegmentoDAO.ObtenerSegmentoColor((int)obj[0], "Cyan");//3
        var red = SegmentoDAO.ObtenerSegmentoColor((int)obj[0], "Red");//3
        var blue = SegmentoDAO.ObtenerSegmentoColor((int)obj[0], "Blue");//3
        var green = SegmentoDAO.ObtenerSegmentoColor((int)obj[0], "Green");//3

        patron.Cyan = new ArrayList(cyan);//4
        patron.Red = new ArrayList(red);//4
        patron.Blue = new ArrayList(blue);//4
        patron.Green = new ArrayList(green);//4

        patrones.Add(patron);//4
    }

    return patrones;//5
}
```

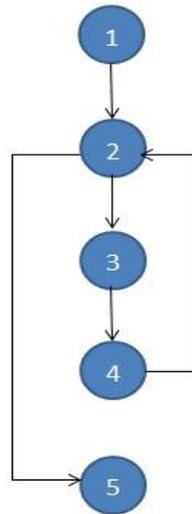


Figura 21. Grafo de Flujo: Funcionalidad ObtenerPatrones.

Complejidad ciclomática para la funcionalidad.

Fórmula 1

$$V(G) = (A \text{ (Aristas)} - N \text{ (Nodos)}) + 2$$

$$V(G) = (5 - 5) + 2 = 2$$

Fórmula 2

$$V(G) = P \text{ (Nodos Predicados)} + 1$$

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

Fórmula 3

$$V(G) = R \text{ (Regiones)} = 2$$

El valor de $V(G)$ brinda el número de **camino linealmente independientes** de la estructura de control del programa, por lo que se definen los siguientes caminos:

Caminos independientes:

- 1-2-5

- 1-2-3-4-2-5

3.7.2 Pruebas de Caja Negra o Funcional.

Las pruebas de caja negra se llevan a cabo, sobre la interfaz del *software*, obviando el comportamiento interno y la estructura del programa. Este tipo de pruebas pretenden demostrar que:

- Las funciones del *software* son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene. [28]

Se realizaron pruebas a diferentes funcionalidades del sistema una de ellas se muestra a continuación y las restantes se pueden observar en el *anexo 6*.

Tabla 10. Caso de prueba Insertar firmas a personas.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU5	Nombre de la HU: Gestionar firmas de personas.
Nombre: Insertar firmas a personas.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para insertar una firma a una persona.	
Condiciones de ejecución: Para insertar una firma a una persona se deberá seleccionar la persona del listado y luego buscar la imagen de la firma que se desea insertar.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar firmas de personas. • Seleccionar la opción Insertar firma a persona. • Pulsar la opción Insertar. 	
Resultado esperado: La firma ha sido insertada correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 11. Caso de prueba Mostrar firmas de personas.

CASO DE PRUEBA DE ACEPTACIÓN.

Código de caso de prueba: T2_HU4	Nombre de la HU: Gestionar firmas de personas.
Nombre: Mostrar firmas de personas.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para mostrar las firmas de una persona.	
Condiciones de ejecución: Para mostrar las firmas de las personas se deberá seleccionar el criterio de búsqueda.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar firmas de personas. • Seleccionar la opción Mostrar firmas de personas • Seleccionar criterio de búsqueda. • Pulsar la opción Buscar. 	
Resultado esperado: Lista de firmas de persona.	
Evaluación de la prueba: Satisfactoria.	

Tabla 12. Caso de prueba Comparar firmas.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU7	Nombre de la HU: Comparar firmas.
Nombre: Comparar firmas.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para comparar firmas.	
Condiciones de ejecución: Para comparar firmas se busca la imagen de la firma o del documento para obtener la firma.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Comparar firmas. • Buscar firma a comparar. • Pulsar la opción Buscar. 	
Resultado esperado: Se devolverá la firma con la cual tenga mayor similitud.	

Evaluación de la prueba: Satisfactoria.

3.7.3 Resultado de las pruebas.

Una vez realizadas las pruebas, se analizan los elementos más relevantes de estas, representando así, el mayor peso en el análisis de los resultados de las pruebas ejecutadas. Luego de vencer las 3 iteraciones de pruebas al sistema, se evidencia que las No Conformidades (NC) surgidas fueron resueltas en su totalidad satisfactoriamente, demostrando de este modo, que el sistema se encuentra listo para ser desplegado.

En la siguiente figura se muestra la relación entre el número de NC generadas y corregidas en cada iteración:



Figura 22. Resultados de la pruebas de aceptación.

3.8 Conclusiones parciales.

En este capítulo, se generaron algunos de los principales artefactos de la metodología XP, como por ejemplo: las Tarjetas CRC y Tareas de Ingeniería. Se modeló el diagrama de componente y despliegue del sistema, con el objetivo de lograr una mayor comprensión y organización de la misma. Se valida el

sistema desarrollado, realizando casos de prueba a todas las funcionalidades del mismo en tres iteraciones. Este proceso permite detectar la mayor cantidad de no conformidades presentes en el *software*, para así darle solución a los problemas detectados. De esta manera, es posible obtener finalmente un sistema que satisface los requisitos definidos y que cuenta con la calidad requerida.

Conclusiones generales

- El estudio de las soluciones similares y de los conceptos fundamentales para el desarrollo de un sistema de identificación de personas mediante firmas manuscritas, permitió obtener los conocimientos necesarios para una mayor comprensión del objeto de estudio, para sentar las bases de la investigación.
- El análisis de las metodologías, herramientas y tecnologías relacionadas con la propuesta de solución, permitió una correcta elección de las mismas.
- El estudio y análisis realizado al sistema propuesto, permitió generar los artefactos definidos por la metodología seleccionada, así como, definir los elementos fundamentales de la arquitectura y el diseño de la propuesta de solución.
- Se realizó la implementación del sistema de identificación de personas mediante firmas manuscritas, teniendo en cuenta las funcionalidades definidas.
- Los resultados arrojados por las pruebas realizadas, mostraron el cumplimiento de los requerimientos y la estabilidad necesaria del sistema para un correcto funcionamiento.

Por todo lo expuesto anteriormente, se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

Recomendaciones

- Integrar la solución al sistema de identificación multibiométrico que se desarrolla en el CISED.
- Integrar a la aplicación los dispositivos de entrada scanner y *Tablet* de firma para poder realizar un reconocimiento *on-line*.

Bibliografía referenciada

1. **Firma-electronica.** Firma-electronica. *Firma-electronica*. [En línea] 28 de octubre de 2010. [Citado el: 14 de noviembre de 2012.] <http://firma-electronica.eu/7.html>.
2. **Landi, Juan Carlos.** dspace. *dspace*. [En línea] [Citado el: 18 de noviembre de 2012.] <http://dspace.ups.edu.ec/bitstream/123456789/814/3/Capitulo1.pdf>.
3. **Gilperez de la Hera, Almudena.** *Reconocimiento off-line de escritura basado en características locales y globales*. Madrid : s.n., 2010.
4. **Vélez, José F., y otros, y otros.** *Verificación Off-Line de Formas Manuscritas:Una Propuesta basada en Snakes Paramétricos*. Madrid : s.n.
5. **Plamondon, F.Leclerc y R.** *Automatic Signature Verification:The State of the Art.Intl.J.Pattern Recog and Artificial Intelligence*. 1994.
6. **Otros y Ammar, M.** *Off-line preprocessing and verification of signatures. Int. Journal of Pattern Recog. and Arti. Intel. .* 1987.
7. **Hou, Z. y Han, C.** *Force field analysis snake: an imported parametric active contour model. Pattern Recog.* 2005.
8. **Camino, J.L. y Otros.** *Signature Classification by HMM. IEEE 33 International Carnahan Conference. .* 1999.
9. **Otros y B. Fang .** *Off-line signature verification with generated training samples. IEE Proc.-Vis. Image and Signal Processing. .* [En línea] 2002.
10. **Balsero García, Néstor O, Botero Galeano, Diego A. y Zuluaga Morales. , Juan P.** *Reconocimiento e Interpretación de gestos manuales por medio de video. .* Bogotá : s.n., 2005.
11. **Hernández Pérez, Alejandro.** *Reconocimiento y verificación de firmas manuscritas off-line basado en el seguimiento de sus trazos componentes*. 2013.
12. **fabianbermeop.** *fabianbermeop. fabianbermeop*. [En línea] 2010. <http://www.fabianbermeop.blogspot.com/2010/12/metodologia-rup-desarrollo-de-software.html> .

13. **EVA.** eva. eva. [En línea]
http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Complementarios/El_Lenguaje_Unificado_de_Modelado/05_Parte_1_Antecedentes.pdf.
14. **rational.** rational. *rational*. [En línea] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
15. **cavsi.** cavsi. *cavsi*. [En línea] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
16. **Ian y Grifillan, Ian.** La Biblia de MySQL. [aut. libro] Ian Grifillan. La Biblia de MySQL. s.l. : Editorial Anaya,, 2003. [En línea] 2009. http://www.postgresql.org.es/sobre_postgresql.
17. **Burrows, Chris.** MSDN Magazine. Nuevas características de C# en .NET Framework 4.5. . *MSDN Magazine. Nuevas características de C# en .NET Framework 4.5.* . [En línea] [Citado el: 5 de febrero de 2013.] <http://msdn.microsoft.com/es-es/magazine/ff796223.aspx>.
18. **Morales Ruiz., Luis A.** *PROPUESTA DE COMPARACION DE FIRMAS BASADO EN EL SEGUIMIENTO DE SUS TRAZOS COMPONENTES.*
19. **Computacion.** [En línea] 2010. [Citado el: 11 de febrero de 2013.]
http://computacion.cs.cinvestav.mx/~jjangel/Pagina_Todos_es.html.
20. **Visual Paradigm.** Visual Paradigm. *Visual Paradigm*. [En línea] 2010. <http://www.visual-paradigm.com>.
21. **Larman,Craig.** *UML y Patrones*. México : s.n., 1999.
22. **Píriz Calabria, Luis y Pablo.** *Metodología XP*. Uruguay : s.n., 2003.
23. **Rubine, D.** *Specifying gestures by example. Technical report, Information Technology Center.* 1993.
24. **Vélez, José Francisco, Sánchez, Ángel y Esteban, José Luis.** *Apuntes Visión Computacional. Informática de Sistemas.* Madrid : s.n.
25. **Parker, J. R.** *Practical Computer Vision.* 1993.
26. **Jr, L. da Fontoura Costa y R. Marcondes Cesar.** *Shape Analisis and Classification.* 2001.
27. **Gómez, Leopoldo Sebastián M.** *Web Adentro,Diseño de Interfaces de Usuario Principios , Prototipos y Heurísticas para Evaluación . Web Adentro,Diseño de Interfaces de Usuario Principios , Prototipos y*

- Heurísticas para Evaluación* . [En línea] 2008. [Citado el: 2013 de abril de 2.]
<http://webadentro.wordpress.com/2008/05/05/disenio-de-interfaces-de-usuario-principios-prototipos-y-heuristicas-para-evaluacion/>.
28. **PruebasSoftware**. Gestión de Calidad y Pruebas de Software. [En línea] 2005. [Citado el: 2013 de marzo de 20.] <http://www.pruebasdesoftware.com/laspruebasdesoftware.htm>.
29. **Rotelli, Maria Belen**. LAS VENTAJAS DE CONTAR CON VISUAL STUDIO 2012 OFICIAL. [En línea] 30 de abril de 2013. <http://www.puertadeenlace.net/post/LAS-VENTAJAS-DE-CONTAR-CON-VISUAL-STUDIO-2012-OFICIAL.aspx>.
30. **Gómez, Sandra**. [En línea] 10 de octubre de 2012. <http://www.globetesting.com/2012/10/nuevas-caracteristicas-de-visual-studio-2012/>.
31. **sqnbankingsystems**. sqnbankingsystems. [En línea]
<http://www.sqnbankingsystems.com/contet/privacy-policy>.
32. **msdn.microsoft.com**. **msdn.microsoft.com**. [En línea] 2009. [Citado el: 15 de mayo de 2013.]
<http://msdn.microsoft.com/es-es/library/ms171868.aspx>.
33. **Sommerville, Ian**. Ingeniería del Software. Madrid : s.n., 2005.
34. **Software, Departamento Central de Ingeniería de**. Flujo de trabajo Captura de requisitos. Modelo de Negocio" s.l: Ciudad de la Habana, Universidad de Ciencias Informáticas , 2004.

Bibliografía consultada

Balsero García, Néstor O, Botero Galeano, Diego A. y Zuluaga Morales. , Juan P. 2005.
Reconocimiento e Interpretación de gestos manuales por medio de video. . Bogotá : s.n., 2005.

Parker, J. R. 1993. *Practical Computer Vision.* 1993.

Colectivo de Bases de Datos. eva. eva. [En línea] [Citado el: 18 de abril de 2013.]
http://eva.uci.cu/file.php/256/1._Tema_0/3._Bibliografia/Herramientas/5-Manual_DBDesigner.pdf.

MATO GARCIA, ROSA MARIA. *DISEÑO de Bases de Daros.* 1999.

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

Díaz Silveira, Dreylis Ertila and Hernández Pérez, Reinier. biblioteca. *biblioteca.* [Online] [Cited: mayo 26, 2013.] <http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=11247>.

Canales Mora, Roberto. 2003. Patrones de GRASP. [En línea] 2003.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp>.

El lenguaje Unificado Modelado.Manual de referencia. [En línea]
http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Complementarios/El_Lenguaje_Unificado_de_Modelado/01_Presentacion_y_contenido.pdf.

Herramientas Case. [En línea]
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.html>.

Jeffries, Ron, y otros. *Extreme Programming Installed.*

Fernández Escribano, Gerardo . *Introducción a Extreme Programming.* 2002.

Turtschi, Adrian y Werry, Jason . *C# .Net Developer's Guide.*

Biosecure. *A biometric reference system for signature.* 2007.

Reconocimiento online. [En línea]
http://tesis.ula.ve/pregrado/tde_busca/arquivo.php?codArquivo=18

Ittextcsharp. [En línea]

<http://www.linux-magazine.es/issue/45/058-061PythonLM45.pdf>

de la Torre Llorente, Cesar, y otros, y otros. *Guía de Arquitectura N-Capas orientada al Dominio .Net 4.0.*

Farina, Alessandro, . Kovacs-Vajna, Zsolt M y Leone, Alberto. *Fingerprint minutiae extraction from skeletonized.* Bologna, Italy : s.n., 1999.

Ferguson, Jeff, et, al. La biblia de C#.

Nagel, Christian, et al. Professional C# 4.0 and .Net 4.

Date, Cj. Introducción a los Sistemas de bases de datos. Mexico: s.n., 2001.

Cervantes Maceda, Humberto. humbertocervantes.net. [Online] 2008. [Citado: mayo 24, 2013.]
<http://www.humbertocervantes.net/cursos/ingsoft/PresentacionCurso.pdf>.

Suen, T. Y. Zhang y C. Y. A fast parallel algorithm for thinning digital patterns. 1984.

Bustos, Guillermo. *Guía de Uso de la Herramienta CASE.* 2012.

Santos González, , Manuel, Patiño Cortés, Ismael and Carrasco Vallinot, Raúl. *Fundamentos De Programación.*

HUESO IBAÑEZ GALINDO, LUIS . *GESTIÓN DE BASES DE DATOS. 2ª EDICIÓN. CFGS. (GUIA DEL PROFESOR).*

HUESO IBAÑEZ, LUIS. ADMINISTRACIÓN DE SISTEMAS GESTORES DE BASES DE DATOS. CFGS.

Real Jurado, Pedro y Armario Sampalo., José Andrés. Procesamiento de Imágenes Digitales. . Sevilla: s.n.

Glosario de Términos

CASE: (Computer Aided Software Engineering) Ingeniería de *Software* Asistida por Computadora.

CISED: Centro de Identificación y Seguridad Digital.

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

GRASP: Acrónimo de *General Responsibility Assignment Software Patterns*. Patrón de asignación de responsabilidades.

Histograma: Representación gráfica de cómo están distribuidos los píxeles de la imagen en cada nivel de intensidad del color.

HU: Historia de Usuario.

TIC: Tecnología de la Informática y las comunicaciones.

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.

Umbral: Cantidad mínima de señal que ha de estar presente para ser registrada por un sistema.

Anexos

Anexo 1: Historias de Usuario.

Tabla 13. HU_Autenticar usuario.

Historia de Usuario	
Número: 1	Usuario: Administrador , Invitado
Nombre historia: Autenticar usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite que el administrador o el invitado puedan autenticarse en el sistema.	
Observaciones: En caso que el usuario o la contraseña no sean correctos el sistema muestra un mensaje indicándolo.	

Tabla 14. HU_Gestionar usuario.

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Gestionar usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Omar Barzaga Angulo	

Descripción: El administrador tendrá la posibilidad de gestionar los usuarios que interactúan con el sistema.
Observaciones: Gestionar usuario incluye, insertar, eliminar, modificar y listar los usuarios.

Tabla 15. HU_Modificar perfil.

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Modificar perfil.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Omar Barzaga Angulo	
Descripción: El administrador tendrá la posibilidad de modificar el perfil de los usuarios del sistema.	
Observaciones: Modificar perfil incluye: modificar la contraseña del usuario así como también modificar el tipo de usuario.	

Tabla 16. Gestionar persona.

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Gestionar Persona	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Omar Barzaga Angulo	
Descripción: El administrador tendrá la posibilidad de gestionar las personas que estarán insertadas en la base de datos.	
Observaciones: Gestionar persona incluye: Insertar persona, listar y modificar los datos de una persona.	

Tabla 17. Obtener imagen de la firma del documento.

Historia de Usuario	
Número: 6	Usuario: Administrador , Invitado
Nombre historia: Obtener imagen de la firma del documento.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite al administrador o al invitado marcar y recortar la firma de la persona en la imagen del documento para luego comparar esta con otras firmas almacenadas en la base de datos.	
Observaciones: Esta funcionalidad incluye: Cargar la imagen del documento firmado y obtener la imagen de la firma del documento.	

Tabla 18. Identificar persona.

Historia de Usuario	
Número: 8	Usuario: Administrador , Invitado

Nombre historia: Identificar persona.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite al administrador o al invitado identificar la persona a la cual pertenece dicha firma.	
Observaciones: Esta funcionalidad incluye: Mostrar datos de la persona identificada.	

Tabla 19. Mostrar reportes de comparación de firmas.

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre historia: Mostrar reportes de comparación de firmas.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite mostrar los reportes de comparación de firmas.	
Observaciones:	

Tabla 20. Exportar reportes de comparación a pdf.

Historia de Usuario

Número: 10	Usuario: Administrador
Nombre historia: Exportar reportes de comparación de comparación de firmas a pdf.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Omar Barzaga Angulo	
Descripción: Permite una vez listado los reportes de comparación de firmas exportarlos a pdf.	
Observaciones:	

Anexo 2: Descripción de las tablas de la base de datos.

Tabla 21. Tabla usuario.

Nombre	usuario	
Descripción	En esta tabla se almacenarán los datos de los usuarios del sistema.	
Atributos	Tipo	Descripción
Id_usuario	Int4	Llave primaria de la tabla.
contrasenna	Varchar	Contraseña del usuario.
usuario	Varchar	Nombre del usuario.
rolid_rol	Int4	Llave foránea que representa a la llave primaria de la tabla rol.

Tabla 22. Tabla rol.

Nombre	rol	
Descripción	En esta tabla se almacenarán los roles de los usuarios del sistema.	
Atributos	Tipo	Descripción
id_rol	Int4	Llave primaria de la tabla.
rol	Varchar	Nombre del rol.

Tabla 23. Tabla patron.

Nombre	patron	
Descripción	En esta tabla se almacenarán los patrones de las firmas guardadas en la base de datos.	
Atributos	Tipo	Descripción
id_patron	Int4	Llave primaria de la tabla.
firmaid_firma	Int4	Llave foránea que representa a la llave primaria de la tabla firma.

Tabla 24. Tabla persona.

Nombre	persona	
Descripción	En esta tabla se almacenarán los datos de las personas registradas en el sistema.	
Atributos	Tipo	Descripción
id_persona	Int4	Llave primaria de la tabla.
nombre	Varchar	Nombre de la persona.

segundo_nombre	Varchar	Segundo nombre de la persona.
primer_apellido	Varchar	Primer apellido de la persona.
segundo_apellido	Varchar	Segundo apellido de la persona.
sexo	Varchar	Sexo de la persona.
carne_identidad	Varchar	Carné de identidad de la persona.

Tabla 25. Tabla firma.

Nombre	firma	
Descripción	En esta tabla se almacenarán las firmas de las personas registradas en la base de datos.	
Atributos	Tipo	Descripción
id_firma	Int4	Llave primaria de la tabla.
personaid_persona	Int4	Llave foránea que representa a la llave primaria de la tabla persona.
fecha_insercion	Date	Fecha en que fue insertada una firma.
imagen	Bytea	Imagen de la firma.
usuarioid_usuario	Int4	Llave foránea que representa a la llave primaria de la tabla usuario.

Tabla 26. Tabla segmento.

Nombre	segmento	
Descripción	En esta tabla se almacenarán todos los segmentos de las firmas almacenadas en la base de datos.	
Atributos	Tipo	Descripción

id_segmento	Int4	Llave primaria de la tabla.
fila_origen	Int4	Fila origen del segmento.
columna_origen	Int4	Columna origen del segmento.
fila_destino	Int4	Fila destino del segmento.
columna_destino	Int4	Columna destino del segmento.
color	Varchar	Color del segmento.
patronid_patron	Int4	Llave foránea que representa a la llave primaria de la tabla patron.

Tabla 27. Tabla reporte.

Nombre	reporte	
Descripción	En esta tabla se almacenarán todos los reportes de comparación de firmas.	
Atributos	Tipo	Descripción
id_reporte	Int4	Llave primaria de la tabla.
similitud	Int4	Similitud que devuelve la comparación de firmas.
fecha_reporte	Date	Fecha en que se realizó un reporte.
hit	Bool	Devuelve verdadero o falso si la firma coincide con algunas de las almacenadas en la base de datos o no.
columna_destino	Int4	Columna destino del segmento.
firmaid_firma	Int4	Llave foránea que representa a llave primaria de la tabla firma.
Usuarioid_usuario	Int4	Llave foránea que representa a llave primaria de la tabla usuario.

Anexo 3: Tarjetas CRC.

Tabla 28. Tarjeta CRC correspondiente a Utiles.cs.

Clase: Utiles.cs	
Responsabilidad	Colaboran
Convertir arreglo de byte a imagen y viceversa, además de encriptar contraseña. También se encarga de crear un reporte en formato pdf.	itextsharp.dll

Tabla 29. Tarjeta CRC correspondiente a UsuarioDAO.cs.

Clase: UsuarioDAO.cs	
Responsabilidad	Colaboran
Esta clase es la responsable de realizar las siguientes acciones insertar un usuario, eliminar este usuario, modificar el nombre de este usuario y obtener todos los usuarios registrados en la base de datos. También cuenta con otras funcionalidades importantes ejemplos de estas son: obtener el nombre de un usuario dado el identificador del usuario, obtener el identificador de usuario, obtener el identificador del rol de usuario y por último modificar la contraseña del usuario seleccionado.	Usuario.cs

Tabla 30. Tarjeta CRC correspondiente a RolDAO.cs.

Clase: RolDAO.cs	
Responsabilidad	Colaboran
Esta clase tiene la responsabilidad de obtener todos los roles de los usuarios registrados en la base de datos.	Rol.cs

Tabla 31. Tarjeta CRC correspondiente a SegmentoDAO.cs.

Clase: SegmentoDAO.cs	
Responsabilidad	Colaboran
Esta clase tiene la responsabilidad insertar y obtener los segmentos de una firma.	Segmento.cs

Tabla 32. Tarjeta CRC correspondiente ReporteDAO.cs.

Clase: ReporteDAO.cs	
Responsabilidad	Colaboran
Esta clase tiene la responsabilidad insertar y obtener todos los reportes de comparación de firmas.	Reportes.cs

Tabla 33. Tarjeta CRC correspondiente a PersonaDAO.cs.

Clase: PersonaDAO.cs	
Responsabilidad	Colaboran

<p>Esta clase es la responsable de insertar y obtener todos los datos de una persona para poder modificarlos. También cuenta con otras funcionalidades importantes ejemplos de estas son: obtener todas las personas que poseen al menos una firma almacenada en la base de datos y permitir buscar personas por nombre, apellidos, carné de identidad y sexo.</p>	<p>Persona.cs</p>
--	-------------------

Tabla 34. Tarjeta CRC correspondiente a PatronesDAO.cs.

Clase: PatronesDAO.cs	
Responsabilidad	Colaboran
<p>Esta clase es la responsable de insertar y obtener de la base de datos todos los patrones de una firma. También permite obtener el identificador de un patrón en específico.</p>	<p>Patron.cs</p>

Tabla 35. Tarjeta CRC correspondiente a FirmaDAO.cs.

Clase: FirmaDAO.cs	
Responsabilidad	Colaboran
<p>Esta clase es la responsable de insertar y obtener de la base de datos</p>	<p>Firma.cs</p>

<p>todas las firmas de una persona. También permite obtener la firma de una persona dado el identificador de persona y permite obtener todos los identificadores de todas las firmas almacenadas en la base de datos. Además, esta clase posee la funcionalidad de buscar el identificador de la firma de una persona.</p>	
--	--

Tabla 36. Tarjeta CRC correspondiente a Conexion.cs.

Clase: Conexion.cs	
Responsabilidad	Colaboran
<p>Esta clase es la encargada de establecer la conexión del sistema con la base de datos. Posee varias funcionalidades ejemplo: obtener la conexión con la base de datos, devolver un arreglo de la base de datos, devolver una consulta y actualizar el campo bytes de la base de datos.</p>	<p>Firma.cs</p>

Anexo 4: Tareas de ingeniería asociadas a las Historias de Usuario.

Tabla 37. Tareas de ingeniería.

Tareas de Ingeniería				
Id_Tarea	No_HU	Nombre_HU	Nombre_Tarea	Responsable
1	1	Autenticar usuarios	Verificar que el usuario y	Glenys Rojas

			la contraseña coincidan con los de la base de datos.	
1	2	Gestionar usuario	Insertar usuarios.	Glenys Rojas y Omar Barzaga
2	2	Gestionar usuario	Eliminar usuario.	Glenys Rojas y Omar Barzaga
3	2	Gestionar usuario	Listar usuarios.	Glenys Rojas y Omar Barzaga
4	2	Gestionar usuario	Modificar usuario.	Glenys Rojas y Omar Barzaga
1	3	Modificar perfil	Modificar contraseña.	Glenys Rojas
2	3	Modificar perfil	Modificar tipo usuario.	Glenys Rojas
1	4	Gestionar personas	Insertar persona.	Glenys Rojas y Omar Barzaga
2	4	Gestionar personas	Listar y modificar datos de personas.	Glenys Rojas y Omar Barzaga
1	5	Gestionar firma de personas	Insertar firma a personas	Glenys Rojas y Omar Barzaga
2	5	Gestionar firma de personas	Mostrar firmas de personas.	Glenys Rojas y Omar Barzaga
1	6	Obtener la firma de la persona	Cargar la imagen del documento firmado.	Omar Barzaga
2	6	Obtener la firma de la persona	Obtener firma del documento.	Omar Barzaga
1	7	Comparar firma	Obtener lista de trazos de la firma.	Omar Barzaga

2	7	Comparar firma	Comparar lista de trazos con los trazos almacenados en la base de datos	Omar Barzaga
1	8	Identificar persona	Mostrar datos de la persona identificada.	Omar Barzaga
1	9	Mostrar reportes de comparación de firmas.	Mostrar reportes de las firmas que hayan sido comparadas.	Omar Barzaga
1	10	Exportar reportes de comparación de firmas a pdf	Una vez que se haya hecho el reporte de comparación de firmas poder exportarlo a pdf.	Omar Barzaga

Anexo 5: Interfaces de Usuario.

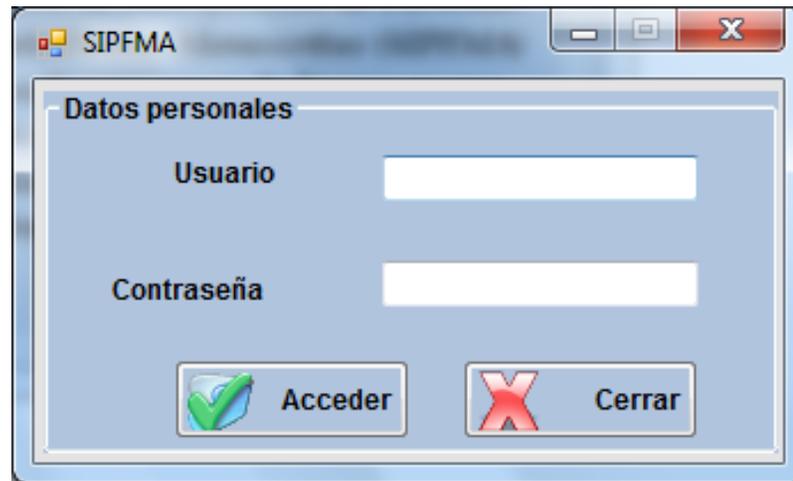


Figura 23. IU_Autenticar usuario.

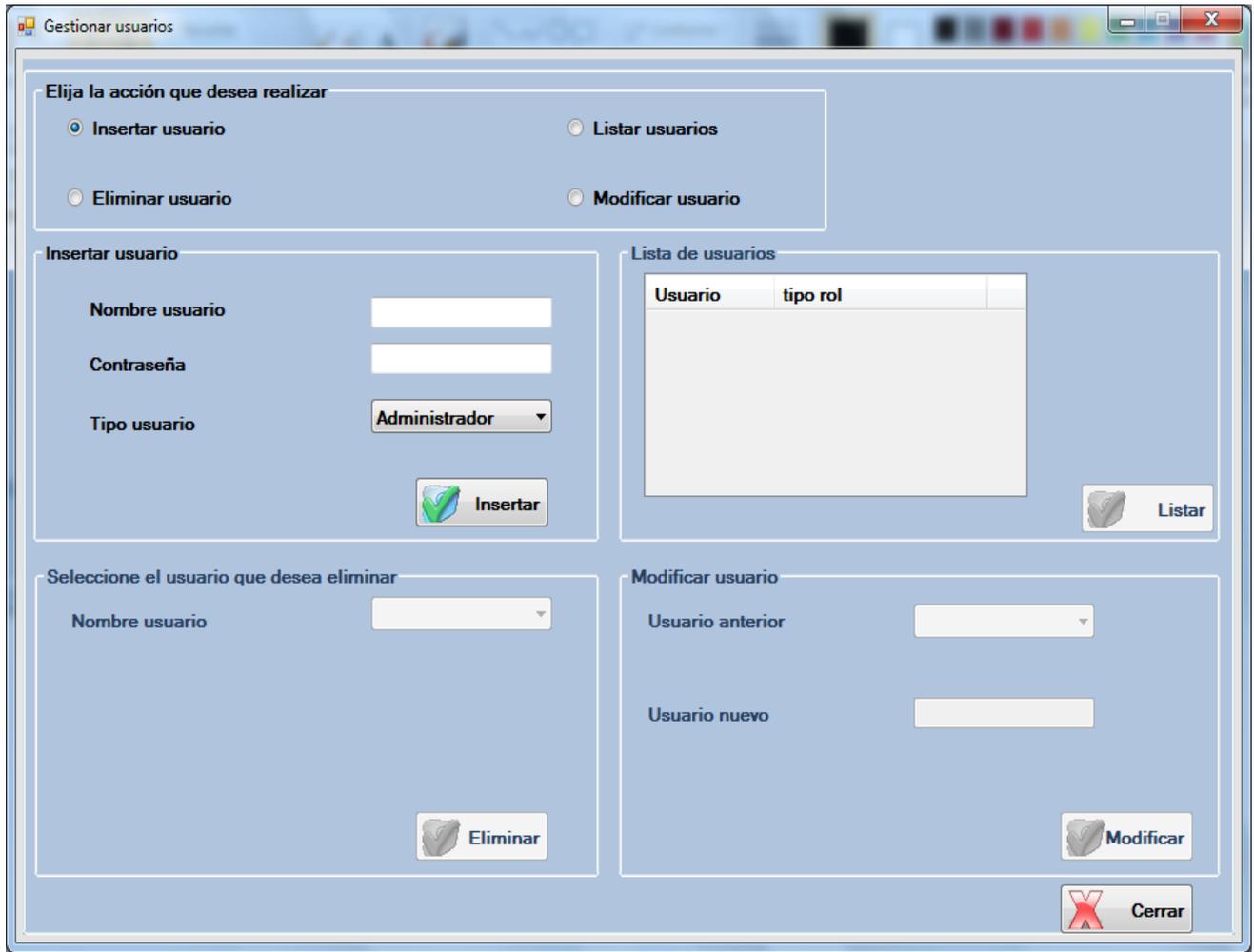


Figura 24. IU_Gestionar usuarios.

Gestión de personas

Elija la acción que desea realizar

Insertar persona

Listar y modificar datos de persona

Insertar persona

Carnet de identidad

Primer apellido

Nombre

Segundo apellido

Segundo nombre

Sexo

 Insertar

Selección del listado de persona la persona a modificar

	Carné de identidad	Nombre	Segundo nombre	Primer apellido	Segundo apellido
▶	90021946491	Marielis		González	Muño
	90102438650	Geidy		González	Espinosa
	90061543990	Rosalina		Ramírez	Araluce
	89071137388	Rafael	Alberto	Quiles	Velázquez
	88120632264	Adrian		Rivera	Correa

 Cerrar

Figura 25. IU_Gestionar persona.

Modificar perfil

Seleccione la acción que desea realizar

- Modificar contraseña
- Modificar tipo usuario

Modificar contraseña

Usuario: omar

Contraseña nueva: []

Confirmar contraseña: []

Cambiar

Modificar rol

Usuario: []

Tipo rol: []

Cambiar

Cerrar

Figura 26. IU_Modificar perfil.

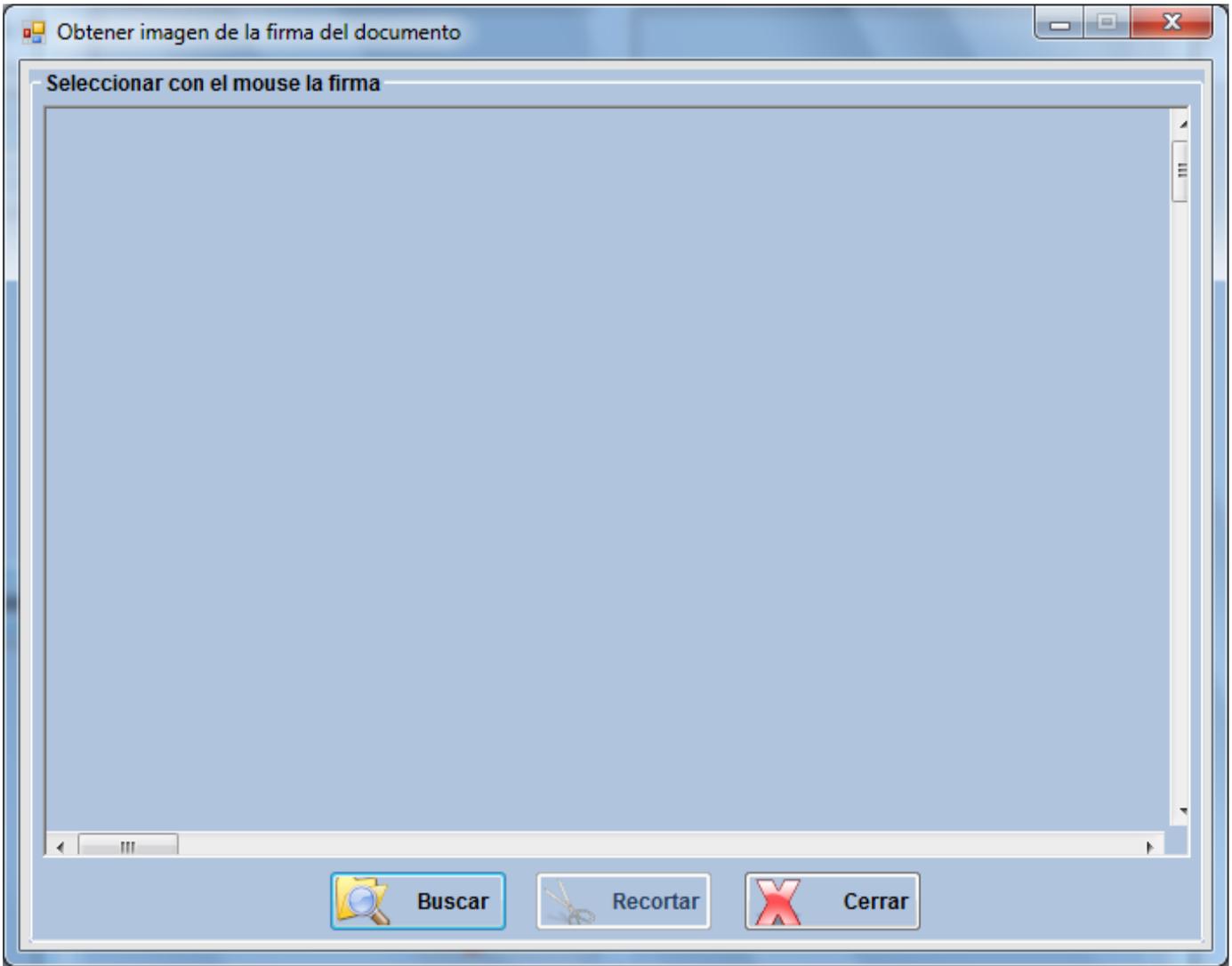


Figura 27. IU_Obtener imagen de la firma del documento.

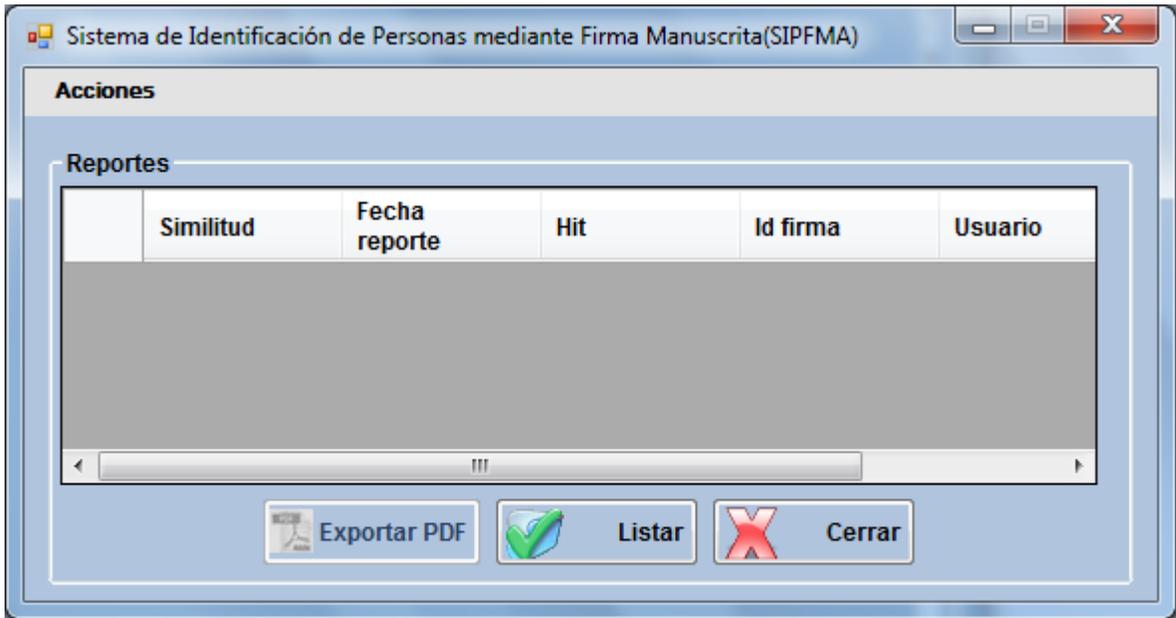


Figura 28. IU_Mostrar reportes.

Anexo 6: Casos de Prueba de aceptación para cada Historia de Usuario.

Tabla 38. Caso de prueba Autenticar usuario.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU1	Nombre de la HU: Autenticar usuarios.
Nombre: Autenticar usuarios.	
Responsable: Glenys Rojas Morales	
Descripción de la prueba: Esta prueba verifica si la contraseña y el usuario son correctas.	
Condiciones de ejecución: Para autenticarse el usuario debe estar registrado en el sistema.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar usuario. • Entrar contraseña. • Pulsar la opción acceder. 	

Resultado esperado: El usuario se autentica correctamente.
Evaluación de la prueba: Satisfactoria.

Tabla 39. Caso de prueba Insertar usuarios.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU2	Nombre de la HU: Gestionar usuarios.
Nombre: Insertar usuarios.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para verificar si se ha creado la cuenta de un usuario en el sistema.	
Condiciones de ejecución: Para insertar un usuario el nombre de este no puede tener números ni coincidir con ninguno de los almacenados en la base de datos.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar usuarios. • Seleccionar la acción Insertar usuario. • Llenar los campos necesarios. • Pulsar la opción Insertar. 	
Resultado esperado: El usuario se insertó correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 40. Caso de prueba Eliminar usuarios.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T2_HU2	Nombre de la HU: Gestionar usuarios.
Nombre: Eliminar usuarios.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para verificar si se ha eliminado la cuenta de un usuario del sistema.	

Condiciones de ejecución: Para eliminar un usuario este debe estar registrado en el sistema.
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar usuarios. • Seleccionar la acción Eliminar usuario. • Selecciona el usuario que desea eliminar. • Pulsar la opción Eliminar.
Resultado esperado: El usuario se eliminó correctamente.
Evaluación de la prueba: Satisfactoria.

Tabla 41. Caso de prueba Listar usuarios.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T3_HU2	Nombre de la HU: Gestionar usuarios.
Nombre: Listar usuarios.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para listar todos los usuarios del sistema.	
Condiciones de ejecución: Para listar los usuarios debe existir algún usuario registrado en el sistema.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar usuarios. • Seleccionar la acción Listar usuarios. • Pulsar la opción Listar. 	
Resultado esperado: Listado de usuarios.	
Evaluación de la prueba: Satisfactoria.	

Tabla 42. Caso de prueba Modificar usuario.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T4_HU2	Nombre de la HU: Gestionar usuarios.
Nombre: Modificar usuario.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para modificar el nombre de un usuario.	
Condiciones de ejecución: Para modificar el nombre de un usuario este nuevo nombre no puede coincidir con el de otros usuarios que estén registrados en el sistema.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar usuarios. • Seleccionar la acción Modificar usuarios. • Pulsar la opción Modificar. 	
Resultado esperado: El usuario de ha modificado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 43. Caso de prueba Modificar contraseña.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU3	Nombre de la HU: Modificar perfil.
Nombre: Modificar contraseña.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para modificar la contraseña de un usuario.	
Condiciones de ejecución: Para modificar la contraseña de un usuario, esta contraseña y su confirmación deben coincidir.	

<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Modificar perfil. • Seleccionar la acción Modificar contraseña. • Pulsar la opción Cambiar.
<p>Resultado esperado: La contraseña se ha modificado correctamente.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 44. Caso de prueba Modificar tipo de usuario.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T2_HU3	Nombre de la HU: Modificar perfil.
Nombre: Modificar tipo usuario.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para modificar el tipo de usuario.	
Condiciones de ejecución: Para modificar el tipo de un usuario este nuevo tipo no puede ser igual al que el usuario tenía antes.	
<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Modificar perfil. • Seleccionar la acción Modificar tipo usuario. • Pulsar la opción Cambiar. 	
Resultado esperado: El rol se ha modificado correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 45. Caso de prueba Insertar persona.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU4	Nombre de la HU: Gestionar personas.
Nombre: Insertar persona.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para insertar una persona al sistema.	
Condiciones de ejecución: Para insertar una persona en el sistema se verifica que esta no registrada en el sistema.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar personas. • Seleccionar la acción Insertar persona. • Pulsar la opción Insertar. 	
Resultado esperado: La persona se insertó correctamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 46. Caso de prueba Listar y modificar datos de persona.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T2_HU4	Nombre de la HU: Gestionar personas.
Nombre: Listar y modificar datos de persona.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para listar y modificar los datos de una persona.	
Condiciones de ejecución: Para modificar los datos una persona se deberá seleccionar del listado de personas aquella a la cual se le desea modificar sus datos.	

<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Gestionar personas. • Seleccionar la acción Listar y modificar datos de persona. • Pulsar la opción Modificar.
<p>Resultado esperado: Los datos de la persona han sido modificados correctamente.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 47. Caso de prueba Obtener imagen de la firma del documento.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU6	Nombre de la HU: Obtener imagen de la firma del documento.
Nombre: Obtener imagen de la firma del documento.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para obtener la firma del documento.	
Condiciones de ejecución: Para obtener la firma del documento se debe cargar la imagen de este y luego marcar la parte que se quiere recortar y recórtala.	
<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Comparar firmas. • Selecciona la opción Obtener firma del documento • Buscar la imagen del documento. • Marcar la firma en la imagen del documento. • Pulsar la opción Recortar. 	
Resultado esperado: Imagen de la firma a comparar.	

Evaluación de la prueba: Satisfactoria.
--

Tabla 48. Caso de prueba Identificar persona.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU8	Nombre de la HU: Identificar persona.
Nombre: Identificar persona.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para identificar el autor de la firma.	
Condiciones de ejecución: Para identificar el autor de la firma se deberá buscar la imagen de la firma a comparar.	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Comparar firmas. • Buscar firma a comparar. • Pulsar la opción Buscar. 	
Resultado esperado: Los datos del autor y la similitud de la firma con la encontrada en la base de datos.	
Evaluación de la prueba: Satisfactoria.	

Tabla 49. Caso de prueba Mostrar reportes de comparación de firmas.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU9	Nombre de la HU: Mostrar reportes de comparación de firmas.
Nombre: Mostrar reportes de comparación de firmas.	
Responsable: Glenys Rojas Morales.	
Descripción de la prueba: Prueba para mostrar los reportes de comparación de firmas.	
Condiciones de ejecución: Para mostrar los reportes de comparación de firmas se deberá de haber comparado alguna firma.	

<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Visualizar reportes. • Pulsar la opción Listar.
<p>Resultado esperado: Los reportes de comparación de firmas.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 50. Caso de prueba Exportar reportes de comparación de firmas a pdf.

CASO DE PRUEBA DE ACEPTACIÓN.	
Código de caso de prueba: T1_HU10	Nombre de la HU: Exportar reportes de comparación de firmas a pdf.
Nombre: Exportar reportes de comparación a pdf.	
Responsable: Glenys Rojas Morales	
Descripción de la prueba: Prueba para exportar los reportes a pdf.	
Condiciones de ejecución: Para exportar los reportes de comparación de firmas se deberá listar estos reportes.	
<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar en la sección Sistema. • Seleccionar la opción Visualizar reportes. • Pulsar la opción Exportar. 	
Resultado esperado: Lista de reportes en formato pdf.	
Evaluación de la prueba: Satisfactoria.	