

*Universidad de las Ciencias Informáticas
Facultad 1*



*Título: Componente para la extracción y verificación off-line
de las características de la firma manuscrita.*



*Trabajo de Diploma para optar por el título de
Ingeniero Informático*

*Autor(es): Dreyllis Ertíla Díaz Silveira.
Reinier Hernández Pírez*

*Tutor: Eduardo Rojas Duverger
Co-tutor: Yurdik Cervantes Mendoza*

*Ciudad de La Habana, 2011
"Año 52 de la Revolución"*



"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos..."

Ernesto Che Guevara

❖ DECLARACIÓN DE AUTORÍA

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 _____.

Reinier Hernández Pérez

Autor

Dreylis Ertila Díaz Silveira

Autora

Eduardo Rojas Duverger

Tutor

Yurdik Cervantes Mendoza

Co-Tutor

❖ DATOS DE CONTACTO

Tutor: Eduardo Rojas Duverger

Correo electrónico: duverger@uci.cu

Ciudad de la Habana, Cuba

Co-Tutor: Yurdik Cervantes Mendoza

Correo electrónico: yurdik@uci.cu

Ciudad de la Habana, Cuba

Datos: Dreylis Ertila Díaz Silveira.

Correo electrónico: dediaz@estudiantes.uci.cu

Ciudad de la Habana, Cuba

Datos: Reinier Hernández Pérez

Correo electrónico: rpirez@estudiantes.uci.cu

Ciudad de la Habana, Cuba

❖ AGRADECIMIENTOS*Dreylis*

Agradezco primeramente a Dios por todo lo maravilloso que tengo en esta vida.

Agradezco a mi mamita Bárbara Silveira Roque por ser como es, por ser mi motor impulsor, mi fuerza interna, por estar siempre conmigo en las buenas y en las malas, por guiarme, por quererme, por cuidarme, por enseñarme, por escucharme, aconsejarme, mimarme, apoyarme, por ser una madre ejemplar, por ser la mejor madre del mundo. Gracias mamita. Te quiero mucho

Agradezco a la mejor hermana del mundo Dreymis Díaz Silveira, la que siempre está ahí junto a mi mamá para que a mí no me falte nada, la que siempre me ha defendido cada vez que estoy en problemas, y a pesar de que tenemos nuestras diferencias y casi siempre estamos fajadas, nos queremos muchos y sabemos que nos tenemos la una a la otra. Mi hermana mi mamá y yo somos un equipo.

Agradezco a mi tía Teresita Alfonso Martínez por ser la mejor tía del mundo y por ser una excelente persona. Más que mi tía es como mi segunda mamá. Al igual que mi mamita y mi hermana siempre está ahí junto a mí, malcriándome, repasándome, educándome, aconsejándome, escuchándome y cuidándome, a mí y a mi mamita. Siempre tiene una palabra de consuelo para mí, siempre apoyándome. Gracias tía, te quiero mucho.

Agradezco a mi prima Yoanna Viciado Silveira, a mi tía AnaMelba Silveira Roque, a mi mejor amiga Yosleidis González, a mi novio, compañero y amigo Ariam Figueredo Rojas, a mi vecina Milvia Hernández Lovet y su mamá Miriam, a mi amiguita AnnieMarien y a mi sobrinita Liz Marien que me alegra el día siempre que voy a verla.

Agradezco a todas aquellas personas que han estado en el campo de batalla conmigo: a Lienis le tengo que agradecer mucho, a Diana, Lenia, Lisbet, Mairelis, Ariannis, Dasy, Yordy, Maite, Aliannis, todos los chicos. A mi compañero de tesis Reinier y a mis tutores Yurdik y Duverger por guiarnos en la realización de este trabajo. A todos en general muchas gracias.

Reinier

A mis padres por apoyarme siempre incondicionalmente en todos mis propósitos, a mimi y a míma por aguantar todas mis malacrianzas y cuidarme como solo mi madre podría hacerlo. A mi abuelo Héctor que aunque ya no está entre nosotros estaría muy orgulloso de mí en este momento. A mamá por ser la abuela mas cariñosa del mundo. A mis amigos, Ernesto, Javier, Uffo, Fino, Guillermo, Eduerdo, el manga, el bichí, Luis Orlando, Onnier, el italiano, a mi compañera de tesis y a mis tutores Yurdik y Duverger y a todas las personas que de una forma u otra me han apoyado en el transcurso de mi vida.

Dreylis

Este trabajo se lo dedico a toda mi familia y todas aquellas personas que me quieren y que siempre me ha apoyado, pero principalmente a mi mamá por ser la persona que me dio la vida y que contribuyó para que todo esto fuera posible.

Reinier

A mi tata por quererme, entenderme, aguantarme todas mis pataletas y por siempre estar ahí para mí.

❖ [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. (García-Obledo, 2010)

Hoy en día, 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.

La firma ha sido usada para validar la autenticidad de transacciones bancarias u otros documentos legales desde hace muchos años. 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, a pesar de ello, actualmente, en muchos países, incluyendo Cuba, todavía se necesita verificar las firmas manualmente, ya que no se ha encontrado otra manera más efectiva de hacerlo.

Debido a todo lo expuesto anteriormente el presente trabajo de diploma expone la realización de un componente para la extracción y verificación de las características de la firma manuscrita, desarrollado en el Centro de Identificación y Seguridad Digital, para automatizar el proceso de identificación y autenticación de personas a través de esta modalidad. Con la realización de este componente se trata de lograr que la verificación manual, costosa y propensa a errores, sea sustituida por una verificación automática más rápida, confiable y de bajo costo.

PALABRAS CLAVE

Biometría, Proceso, Firma Manuscrita, Tecnología, Métodos, Sistema, Automatizar, Componente, Falsificaciones Burdas.

❖ [TABLA DE CONTENIDOS](#)

❖ DECLARACIÓN DE AUTORÍA	I
❖ DATOS DE CONTACTO	II
❖ AGRADECIMIENTOS	III
❖ DEDICATORIAS	V
❖ RESUMEN	VI
❖ TABLA DE CONTENIDOS	VII
❖ ÍNDICE DE ILUSTRACIONES.....	XI
❖ ÍNDICE DE TABLAS	XII
❖ INTRODUCCIÓN.....	1
❖ CAPITULO #1: FUNDAMENTACIÓN TEÓRICA	6
1.1. Introducción	6
1.2. Introducción a la Biometría	6
1.2.1. Biometría	6
1.2.2. Autenticación Biométrica	7
1.2.2.1. Identificación	7
1.2.2.2. Verificación.....	7
1.2.2.3. Criterios de evaluación de sistemas de verificación.....	7
1.2.3. Sistema Biométrico.....	8
1.2.4. Arquitectura de un Sistema Biométrico	8
1.2.5. Aplicaciones de los Sistemas Biométricos	9
1.2.6. Ventajas y Desventajas de la Biometría.....	10
1.2.7. Tipos de Biometría.....	11
1.2.8. Breve reseña histórica de la Biometría	12
1.3. Introducción a la Firma.....	13
1.3.1. Firma Manuscrita	13

1.3.2.	Características.....	14
1.3.3.	Aplicaciones	15
1.3.4.	La verificación de la firma	15
1.3.5.	Reconocimiento on-line y off-line	15
1.3.5.1.	Ventajas y Desventajas	17
1.3.6.	Trabajos existentes.....	17
1.4.	Descripción actual del dominio del problema	18
1.5.	Tendencias generales para dar solución a este tipo de problemas.....	18
1.6.	Etapas del pre-proceso del reconocimiento de firmas.	19
1.6.1.	Limpieza y eliminación del ruido	19
1.6.1.1.	Binarización	19
1.6.1.2.	Segmentación.....	20
1.6.1.3.	Esqueletizado.	21
1.6.1.4.	¿Qué es el esqueleto de una imagen?.....	21
1.6.1.5.	¿Qué es el adelgazamiento?	22
1.6.1.6.	Principales algoritmos conocidos para la esqueletización de una imagen.	22
1.6.1.7.	Fundamentación del algoritmo seleccionado.....	23
1.6.2.	Técnicas de etiquetado de trazos.	25
1.6.3.	Extracción de características.	25
1.6.4.	Reconocimiento basado en trazos.....	26
1.6.4.1.	Descripción del algoritmo Matrices de convolución.	26
1.6.5.	Algoritmo de seguimiento y extracción de trazos.	28
1.6.5.1.	Descripción del algoritmo Búsqueda con retroceso.....	28
1.6.6.	Cálculo de la caja de Feret	29
1.6.7.	Descripción del algoritmo Ajuste de distancias.	29
1.7.	Tecnologías a utilizar para el desarrollo del trabajo	30
1.7.1.	Metodologías de Desarrollo	30
1.7.1.1.	Fundamentación de la metodología seleccionada.....	31
1.7.2.	Herramientas Case.....	32
1.7.2.1.	Rational Rose Enterprise Edition	32

1.7.2.2.	Altova UModel 2007	33
1.7.2.3.	Visual Paradigm.	33
1.7.2.4.	Fundamentación de la herramienta Case utilizada	34
1.7.3.	Framework de desarrollo	34
1.7.3.1.	Net framework 4.0	35
1.7.3.2.	Visual Studio Ultimate 2010	35
1.7.4.	Lenguajes	36
1.7.4.1.	Fundamentación del lenguaje seleccionado	36
1.8.	Propuesta de selección de tecnologías y herramientas	36
1.9.	Conclusiones Parciales.....	37
❖	CAPÍTULO #2: CARACTERÍSTICAS DEL COMPONENTE.....	38
2.1.	Introducción	38
2.2.	Objeto de Informatización	38
2.3.	Descripción de la solución propuesta.....	38
2.4.	Requerimientos Funcionales	39
2.5.	Requerimientos no Funcionales	40
2.6.	Fase de Exploración: Definición	41
2.6.1.	Historias de Usuario	41
2.7.	Fase de Planificación: Definición	44
2.7.1.	Estimación del esfuerzo por Historia de Usuario.....	44
2.7.2.	Plan de iteraciones	45
2.7.2.1.	Iteración 1	45
2.7.2.2.	Iteración 2	46
2.7.2.3.	Iteración 3	46
2.7.3.	Plan de duración de las iteraciones	46
2.7.4.	Plan de entregas.....	47
2.8.	Conclusiones Parciales.....	48
❖	CAPÍTULO #3: DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DEL COMPONENTE	49
3.1.	Introducción	49
3.2.	Diseño.....	49

3.2.1.	Modelo de Dominio.....	49
3.2.1.1.	Glosario de Términos del Diagrama de Dominio.....	50
3.2.2.	Diagramas de clase de diseño.....	51
3.2.3.	Patrones de Diseño	54
3.2.4.	Arquitectura	56
3.3.	Implementación del componente	57
3.3.1.	Iteración #1	58
3.3.1.1.	Tareas generadas por cada historia de usuario	59
3.3.2.	Iteración #2.....	62
3.3.2.1.	Tareas generadas por cada historia de usuario	63
3.3.3.	Iteración #3.....	64
3.4.	Pruebas	64
3.4.1.	Pruebas de aceptación	64
3.4.2.	Pruebas unitarias.....	65
3.4.3.	Prueba realizada a la aplicación integrando otras aplicaciones.	66
3.4.4.	Análisis de la EER de otros trabajos relacionados con este tema.....	69
3.5.	Conclusiones Parciales.....	69
❖	APORTES.....	70
❖	CONCLUSIONES	71
❖	RECOMENDACIONES.....	72
❖	REFERENCIAS BIBLIOGRÁFICAS	73
❖	BIBLIOGRAFÍAS CONSULTADAS.....	77
❖	ANEXOS	78
❖	GLOSARIO DE TÉRMINOS	87

❖ ÍNDICE DE ILUSTRACIONES

Ilustración 1: Firma manuscrita	14
Ilustración 2: Sistema de adquisición de escritura <i>off-line</i>	16
Ilustración 3: Sistema de adquisición de escritura <i>on-line</i>	16
Ilustración 4: Notación esquelizado	24
Ilustración 5: Ejemplo de transiciones	24
Ilustración 6: Original	25
Ilustración 7: Resultado esquelizado	25
Ilustración 8: Matrices de convolución.....	26
Ilustración 9: Etiquetado de firma según la inclinación	27
Ilustración 10: Firma Original	29
Ilustración 11: Caja de Feret sobre los trazos.	29
Ilustración 12: Ajuste de distancias (distance matching)	30
Ilustración 13: Diagrama de Dominio	50
Ilustración 14: Diagrama de clase de la librería “ImagesWorking”	51
Ilustración 15: Contenido del paquete HandwrittenSignatureRecognition (Aplicación de prueba)”	52
Ilustración 16: Diagrama de clases del “Paquete Iterators perteneciente a ImagesWorking”	53
Ilustración 17: Arquitectura del componente.	57
Ilustración 18 Prueba unitaria realizada al método <i>Compare</i>	65
Ilustración 19: Resultado de la prueba unitaria realizada a la funcionalidad <i>Compare</i>	65
Ilustración 20 Prueba unitaria realizada al método <i>Pattern</i>	66
Ilustración 21 Resultado de la prueba unitaria realizada a la funcionalidad <i>Pattern</i>	66
Ilustración 22: Gráfica del FAR.	67
Ilustración 23: Gráfica del FRR.	67
Ilustración 24: Gráfica del FAR vs FRR. EER.	68
Ilustración 25 Gráfica del ROC.....	68
Ilustración 26: Manual de usuario	78
Ilustración 27: Firmas utilizadas en la validación de la propuesta	83
Ilustración 28: Prototipos de las firmas.....	84
Ilustración 29: Arquitectura de un Sistema Biométrico.	84
Ilustración 30: Módulo de Identificación y Verificación.....	85
Ilustración 31: Módulo de Inscripción.	85
Ilustración 32: Proceso general de un Sistema Biométrico.....	86

❖ [ÍNDICE DE TABLAS](#)

Tabla 1: Funcionalidades del componente	40
Tabla 2: Historia de Usuario “Pre-procesar la imagen de la Firma”	42
Tabla 3: Historia de Usuario “Extraer características de la firma”	43
Tabla 4: Historia de Usuario “Verificar Firmas”	43
Tabla 5: Historia de Usuario “Mostrar criterio de evaluación”	44
Tabla 6: Estimación del esfuerzo por Historia de Usuario.	45
Tabla 7: Plan de duración de las iteraciones.....	47
Tabla 8: Plan de entregas	48
Tabla 9: Historias de Usuario abordadas en esta iteración.	58
Tabla 10: Tarea#1 HU “Pre procesar Imagen de la Firma”	59
Tabla 11: Tarea#2 HU “Pre procesar Imagen de la Firma”	59
Tabla 12: Tarea#3 HU “Pre procesar Imagen de la Firma”	60
Tabla 13: Tarea#4 HU “Pre procesar Imagen de la Firma”	60
Tabla 14: Tarea#1 HU “Extraer características de la Firma”	61
Tabla 15: Tarea#2 HU “Extraer características de la Firma”	61
Tabla 16: Tarea#3 HU “Extraer características de la Firma”	62
Tabla 17: Historias de Usuario abordadas y corregidas en esta iteración.	62
Tabla 18: Tarea#1 HU “Verificar Firmas”	63
Tabla 19: Tarea#1 HU “Mostrar criterio de evaluación”	63
Tabla 20: Historias de Usuario a corregir en esta iteración.	64

❖ INTRODUCCIÓN

El futuro de la humanidad está sustentado, en gran medida, por las Tecnologías de la Información y las Comunicaciones (TIC). La capacidad de las TIC para reducir muchos obstáculos tradicionales, especialmente el tiempo y la distancia, posibilita el uso de su potencial en beneficio de millones de personas en todo el mundo.

Cuba, a pesar de las dificultades y de los obstáculos no se encuentra ajena a estas tecnologías y está enfrascada en la informatización de la sociedad tomando como meta las palabras del comandante en jefe: “Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación”. (CASTRO, 2003).

Gracias a los avances tecnológicos en Cuba, se han obtenido importantes logros y beneficios sociales, entre ellos: un alto nivel de desarrollo en la educación, la salud, la cultura y el empleo.

La Biometría, como tecnología, ha contribuido de cierta manera, al desarrollo tecnológico de país. Su principal objetivo es la identificación y autenticación de personas, ya sea a través de un rasgo físico o conductual. Es una rama de la tecnología muy conocida en todo el mundo debido a su gama de funcionalidades y usos. Se manifiesta en varias modalidades para el reconocimiento de personas como: Huella Digital, Reconocimiento Facial, Reconocimiento de Voz, Escritura y Firma. Sobre esta última modalidad estará haciendo mayor énfasis el presente trabajo.

Todos, alguna vez, han tenido que estampar su firma como señal de aceptación o les ha tocado constatar la legalidad de algún acuerdo verificando que las firmas estampadas en el documento pertenezcan a quienes realmente deben pertenecer, garantizando, de esta manera, que todo sea legal. Sin embargo, este proceso de verificación puede resultar tedioso incluso para expertos especializados en el reconocimiento e identificación de firmas, es por eso que una vez más, la Ciencia de la Computación, a través de la Biometría, se encarga de proveer métodos que son capaces de automatizar este proceso.

Existen multitudes de mecanismos destinados a la verificación de personas utilizando la modalidad de Firma, como pueden ser las técnicas basadas en firmas digitales y electrónicas, como el uso de otra serie de técnicas destinadas a la identificación y verificación de las firmas manuscritas digitalizadas. En la actualidad está muy extendido el uso de las firmas digitales. El uso conjunto de técnicas para el

reconocimiento de firmas manuscritas digitalizadas puede aportar una serie de mejoras bastante considerables como son la rapidez y fiabilidad de este proceso.

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el Centro de Identificación y Seguridad Digital (CISED), actualmente se está desarrollando un proyecto productivo, nuevo en su tipo, que se propone crear módulos que permitan desarrollar sistemas de seguridad con reconocimiento y autenticación biométricos basados en las modalidades de Reconocimiento Facial, Huella Digital y Firma Manuscrita Digitalizada.

Desafortunadamente, a pesar de que en Cuba hoy en día se emplean técnicas para la identificación y verificación de personas mediante la firma manuscrita, dichas técnicas se realizan de forma manual, debido a que el proceso de automatización de la investigación y organización de la información biométrica es de muy poco conocimiento en el país, y por tanto el Dpto. de Biometría del Centro de Identificación y Seguridad Digital, en estos momentos, no cuenta con un componente basado en la verificación off-line de firma manuscrita que permita:

- Simplificar el proceso de reconocimiento de personas.
- Disminuir las falsificaciones burdas que son las más comunes en esta modalidad.
- Disminuir los gastos en recursos.
- Disminuir la pérdida de tiempo.

Constituyendo todo esto, la **situación problemática** del presente trabajo.

Teniendo en cuenta las problemáticas existentes dentro del CISED se ha determinado como **problema científico**: ¿Cómo simplificar el proceso de extracción y verificación de las características de la Firma Manuscrita?

El **objeto de estudio** de la presente investigación lo constituye: Sistemas biométricos de identificación y verificación de personas y el **campo de acción** es: Componente para la extracción y verificación de las características de la firma manuscrita.

Para dar solución a la problemática planteada se define como **objetivo general** de la investigación: Implementar un componente para la extracción y verificación de las características de la firma manuscrita.

Objetivos específicos:

Para lograr una solución factible se ha desglosado el objetivo general en los siguientes objetivos específicos.

- Analizar el estado del arte de la investigación.
- Diseñar una arquitectura que exponga las funcionalidades en una arquitectura de servicios para que pueda ser integrable a otras aplicaciones desarrolladas en el CISED.
- Implementar un componente de verificación de firma off-line que se ajuste a la arquitectura anteriormente diseñada.
- Validar la solución propuesta.

Idea a defender:

Si se desarrolla el componente para la extracción y verificación de las características de la firma manuscrita se logrará aumentar la rapidez, la eficacia y la seguridad del proceso de verificación de personas a través de la firma manuscrita.

Para dar cumplimiento a los objetivos se definen las siguientes **tareas científicas**:

1. Descripción y análisis del estado del arte del proceso de extracción y verificación de firmas manuscritas.
2. Estudio de patrones de arquitectura para definir las clases del sistema para el reconocimiento y verificación de firma manuscrita.
3. Estudio de los algoritmos utilizados para el reconocimiento y verificación de firmas manuscritas.
4. Documentación y análisis de las tecnologías y herramientas a utilizar para el desarrollo de la aplicación.
5. Especificación de los requisitos funcionales y no funcionales del sistema a desarrollar.
6. Implementación de un componente de extracción y verificación de las características de la firma manuscrita, que se ajuste a una arquitectura de tubería de filtros y que cumpla con los requisitos funcionales y no funcionales.
7. Realización de pruebas sencillas para validar la solución propuesta.

Para llevar a cabo las tareas científicas se emplearon métodos teóricos y empíricos de la investigación científica. Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

Histórico Lógico: este método se utiliza porque permite analizar la trayectoria de los procesos, desde sus inicios hasta la actualidad y permite analizar las conexiones históricas más importantes.

Deductivo: se utiliza este método porque permite inferir conclusiones y predicciones a partir de los conocimientos adquiridos de los procesos de reconocimiento y verificación de firmas.

Analítico-Sintético: posibilita el análisis del reconocimiento de firma para determinar con exactitud como este funciona. Se toman todas las características principales para lograr modelar un sistema que logre una integración eficaz y una armonía dentro de los procesos que rigen su comportamiento.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio son:

Observación: este método se utiliza porque permite investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo que ayudó al planteamiento del problema científico, además de permitir conocer el proceso delimitado como objeto de estudio, lo cual ayuda a tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacer.

La investigación está **estructurada** en tres capítulos:

Capítulo 1. Fundamentación Teórica: Se abordan los aspectos relacionados con la extracción y verificación de características de firmas manuscritas a través de un sistema reconocedor de personas, un estudio del estado del arte y de los conceptos principales asociados al dominio del problema. Se mencionan las tecnologías, técnicas y metodologías a emplear para dar solución al problema, se realiza una profunda investigación sobre los algoritmos utilizados en este proceso, haciéndose una descripción breve de los mismos.

Capítulo 2. Características del Componente: En este capítulo se estará tratando la solución propuesta para darle respuesta a la situación problemática que se planteó en el capítulo anterior, se definen los

requerimientos y los no funcionales. Se describen los principales artefactos generados por la metodología seleccionada.

Capítulo 3. Diseño, Implementación y Prueba: Se realiza una valoración crítica del diseño propuesto con el propósito de refinar y estructurar los requisitos obtenidos con anterioridad para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. Se aborda la implementación. Se hace referencia a los diagramas de clases y diagrama de dominio. Se realiza la valoración de las pruebas a realizar para probar el correcto funcionamiento del componente.

❖ CAPITULO #1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se realiza un análisis del estado del arte del problema a tratar. Se abordan algunos de los conceptos fundamentales relacionados con la Biometría y la Firma Manuscrita, y se analizan las posibles tecnologías, metodologías y herramientas a utilizar para la elaboración de la solución.

1.2. Introducción a la Biometría

Desde la antigüedad, se han puesto en práctica técnicas que permitan el reconocimiento de personas, ya sea a través de características físicas o de comportamiento. Esto se debe a que siempre ha sido necesario conocer la identidad de algún individuo respecto a la validación de algún documento, una transacción o cualquier otro evento o propósito; por lo que se puede decir que el uso de la Biometría no es nuevo.

En este epígrafe se estarán tratando algunos conceptos relacionados con la Biometría, su historia, los diferentes módulos en los que se desarrolla, entre otros conceptos.

1.2.1. Biometría

Existen algunas definiciones de la biometría:

- **Una definición tradicional**

La Biometría es la aplicación de técnicas matemáticas y esquemas al estudio de datos de las ciencias biológicas. [1]

- **Una definición en el contexto tecnológico**

La biometría es el conjunto de métodos, técnicas y tecnología, que permiten la identificación y autenticación de las personas de forma automatizada, en base a características corporales o de comportamiento. [1]

Analizando las definiciones anteriores se llegó a la conclusión de que la definición más aceptada es la tecnológica ya que guarda más relación con el presente trabajo.

1.2.2. Autenticación Biométrica

La autenticación biométrica se realiza a través de dos tipos de procesos: la identificación y la verificación.

1.2.2.1. Identificación

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. [2]

1.2.2.2. Verificación

En el proceso de verificación, los usuarios declaran su identidad al sistema ya sea con su nombre y contraseña o una tarjeta de identificación. La tecnología biométrica se usa para verificar esa 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 se emparejan o no. [2]

1.2.2.3. Criterios de evaluación de sistemas de verificación

Las dos posibles salidas en un sistema de verificación dan lugar a la aparición de dos errores distintos:

- **Falso Rechazo (FRR):** se produce si el sistema de verificación devuelve como salida que el usuario en la entrada no se corresponde con la plantilla almacenada, y realmente sí es la misma persona.
- **Falsa Aceptación (FAR):** es complementario al falso rechazo y se produce cuando el sistema indica que la información adquirida del usuario en la entrada sí se corresponde con la plantilla almacenada, cuando realmente se trata de otra persona. [1]

1.2.3. Sistema Biométrico

Es un sistema de reconocimiento de patrones que funciona obteniendo datos biométricos de un individuo. El sistema extrae un conjunto de características de los datos adquiridos, y compara estas características con una o un conjunto de plantillas (características biométricas), almacenadas en la base de datos. Dependiendo del uso del sistema, si está en su fase operacional, puede funcionar en modo de identificación o modo de verificación. [1]

Su modo de operación se puede dividir en cuatro pasos: [2]

1. Captura del rasgo biométrico mediante un sensor apropiado.
2. Extracción de un conjunto de características discriminativas del rasgo.
3. Comparación entre patrones almacenados en una base de datos y el patrón anteriormente extraído.
4. Decisión de si los datos de entrada pertenecen a un individuo determinado o a un impostor.

1.2.4. Arquitectura de un Sistema Biométrico

Un sistema biométrico tiene tres elementos básicos: ([Ilustración 29](#): Arquitectura de un Sistema Biométrico)

1. Elemento de adquisición digital o analógica de datos de factor de autenticación biométrico de una persona.
2. Elemento de compresión, procesamiento, almacenamiento y comparación de datos obtenidos con los datos almacenados.
3. Elemento de interfaz con aplicaciones ya sea del mismo sistema u otro.

Un sistema biométrico puede comprenderse como dos módulos:

1. Módulo de inscripción: El cual se encarga de adquirir y almacenar la información procedente del dispositivo biométrico con el objeto de entrenar al sistema. ([Ilustración 31](#): Módulo de Inscripción)
2. Módulo de identificación y verificación: Este módulo es el responsable del reconocimiento de individuos, mediante la comparación con las plantillas almacenadas en la inscripción. ([Ilustración 30](#): Módulo de Identificación y Verificación)

En los módulos de identificación y verificación existen dos modos de funcionamiento: modo identificación y modo de verificación. La diferencia entre estos dos es que la verificación realiza el proceso de comparación a una sola plantilla almacenada en la base de datos, y la identificación realiza la comparación con N plantillas almacenadas. Por lo que el proceso de identificación es más complejo que el de verificación. [1]

1.2.5. Aplicaciones de los Sistemas Biométricos

Luego de la breve descripción de los sistemas biométricos, se verán algunas de las aplicaciones de los mismos. Así como varían los sistemas biométricos, dependiendo de la modalidad biométrica utilizada, también varían sus aplicaciones. Las siguientes aplicaciones se han clasificado en tres tipos, los cuales juegan los papeles más importantes dentro de la sociedad. [2]

Se pueden establecer los siguientes tipos de **aplicaciones** del reconocimiento biométrico:

Aplicaciones de **cara al ciudadano**, dentro las cuales se engloban:

- Identificación criminal de un sospechoso o detenido.
- Verificación de la identidad en la interacción del ciudadano con servicios públicos como salud, voto, seguridad social, etc.
- Vigilancia de individuos presentes en un lugar en un momento determinado, por ejemplo en eventos públicos.

Aplicaciones de **cara al empleado**, dentro las cuales se engloban:

- Acceso a equipos o redes, sustituyendo o complementando los mecanismos tradicionales mediante clave.
- Acceso físico a instalaciones, típicamente a un edificio, en complemento o sustitución de llaves, tarjetas magnéticas, etc.

Aplicaciones de **cara al cliente**, dentro las cuales se engloban:

- Comercio electrónico y transacciones remotas telefónicas.

- Terminales de punto de venta, complementando o sustituyendo las tradicionales tarjetas con número PIN.

También se tienen aplicaciones orientadas a varios campos de la sociedad, en los cuales se engloban la mayoría de las aplicaciones biométricas.

Estos campos son:

- **Legal (forense)**, que hace uso del reconocimiento biométrico para identificar a individuos sospechosos, detenidos, bajo situación de arresto o con restricciones de libertad (arresto domiciliario, etc.).
- **Gubernamental**, donde el reconocimiento biométrico se utiliza para controlar la interacción del ciudadano con entidades públicas y para la propia administración del sistema público.
- **Financiero**, al igual que en el sector gubernamental, el reconocimiento biométrico controla la interacción del ciudadano con el sistema financiero (acceso a cuentas o transacciones comerciales) y la propia administración del sector, por ejemplo el acceso de empleados a redes protegidas.
- **Salud**, donde al igual que los dos casos anteriores, por un lado se controla la interacción del usuario (utilización de servicios sanitarios) y por otro lado se asegura el correcto funcionamiento del sistema (manejo de información médica por parte de empleados).
- **Inmigración**, donde el reconocimiento biométrico se usa para el control de movimientos a través de fronteras y para el control interno de los propios empleados dentro de las áreas restringidas de acceso.

1.2.6. Ventajas y Desventajas de la Biometría

Ventajas

Con la tecnología de la Biometría se obtienen varias ventajas frente a los sistemas tradicionales de seguridad informática:

- Solucionar el problema del mal uso de las contraseñas. La biometría autentica a los individuos a través de una característica biológica intransferible que no se puede copiar, olvidar ni ser remplazada por otras personas.
- Disminuir las molestias para el usuario y los costos en la administración de contraseñas. Las personas no tienen que recordar ni cambiar continuamente contraseñas, no tienen que estar sujetas a normas de seguridad del uso de contraseñas, claves, tarjetas de identificación que normalmente no se cumplen.
- Incrementa la fiabilidad de saber en todo momento quien es el responsable real de las transacciones que se realiza en el sistema.
- La biometría permite una identificación y verificación automática. El usuario no pierde tiempo y evita la molestia de tener que estar interactuando con el sistema. [1]

Desventajas

El desarrollo y la implementación de nuevas tecnologías, siempre genera problemas o inconvenientes:

- No todas las personas pueden usar la tecnología biométrica. Algunos les puede faltar el requerimiento biométrico, ya sea que sufren de alguna lesión o tienen discapacidades. Por lo que existe un riesgo de exclusión social para los que no puedan usar esta tecnología.
- Los identificadores biométricos son intransferibles e irrevocables, estos producen información que es muy valiosa, por lo que es necesario que se generen leyes para regular su uso y protección.
- Debido al aumento de seguridad en los sistemas, pueden surgir nuevos tipos de crímenes como secuestros y amenazas para obtener el identificador biométrico. [1]

1.2.7. Tipos de Biometría

Biometría Estática¹: Es la que mide la anatomía del usuario. Comprende, entre otras:

- Huellas Digitales.

¹ Característica biométrica basada principalmente en una característica anatómica o fisiológica, y no en un comportamiento aprendido. De algún modo, todas las características biométricas dependen tanto de características de comportamiento como de características biológicas. Ejemplos de modalidades biométricas en las que pueden dominar las características biológicas son la geometría de las huellas dactilares y la geometría de las manos.

- Geometría de la mano.
- Termografía.
- Análisis del iris.
- Análisis de retina.
- Venas del dorso de la mano.
- Reconocimiento Facial.

Biometría Dinámica²: Es la que mide el comportamiento del usuario. Comprende entre otras:

- Patrón de Voz.
- Firma manuscrita.
- Dinámica de tecleo.
- Cadencia del paso.
- Análisis gestual.

1.2.8. Breve reseña histórica de la Biometría

El origen de la biometría data desde 31.000 a.C. con las pinturas en cuevas rodeadas de huellas de manos usadas aparentemente como "firma". Luego 500 a.C. se ven las huellas digitales usadas en transacciones de negocio en Babilonia, marcadas en arcilla. [3]

La biometría, como ciencia, no se puso en práctica en las culturas occidentales hasta finales del siglo XIX, pero era utilizada en China desde al menos el siglo XIV. Un explorador y escritor que respondía al nombre de Joao de Barros escribió que los comerciantes chinos estampaban las impresiones y las huellas de la palma de las manos de los niños en papel con tinta. Los comerciantes hacían esto como método para distinguir entre los niños jóvenes.

En Occidente, la identificación confiaba simplemente en la "memoria fotográfica" hasta que Alphonse Bertillon, jefe del departamento fotográfico de la Policía de París, desarrolló el sistema antropométrico

² Característica biométrica aprendida y adquirida con el tiempo, y no basada en la biología. De algún modo, todas las características biométricas dependen tanto de características de comportamiento como de características biológicas. Ejemplos de modalidades biométricas en las que pueden dominar las características de comportamiento son el reconocimiento de firmas y el dinamismo de pulsación de teclas.

(también conocido más tarde como Bertillonage) en 1883. Éste era el primer sistema preciso, ampliamente utilizado científicamente para identificar a criminales y convirtió a la biométrica en un campo de estudio. Funcionaba midiendo de forma precisa ciertas longitudes y anchuras de la cabeza y del cuerpo, así como registrando marcas individuales como tatuajes y cicatrices. El sistema de Bertillon fue adoptado extensamente en occidente hasta que aparecieron defectos en el sistema - principalmente problemas con métodos distintos de medidas y cambios de medida. Después de esto, las fuerzas policiales occidentales comenzaron a usar la huella dactilar - esencialmente el mismo sistema visto en China cientos de años antes. [3]

1.3. Introducción a la Firma

El ser humano desde siempre ha sentido la necesidad de comprobar la identificación de las personas, es decir, saber si realmente una persona es quien dice ser. La motivación ha sido muy variada y originada por diferentes razones, ya sea por la validación de un documento o por transacciones, entre otras.

En este epígrafe se estarán tratando temas relacionados con la modalidad de firma, así como los diferentes conceptos relacionados con la misma.

1.3.1. Firma Manuscrita

- Mustapich³ 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)⁴ 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]

Luego de analizar detalladamente cada una de las definiciones de firma, se llegó a la conclusión de que

³ Jurista que tiene una gran colección de escrituras relacionadas con el derecho.

⁴ Juristas que se han dado a la tarea de estudiar y crear diversas doctrinas sobre diversos temas.

ninguna brinda información relevante para la investigación, ya que la firma es mucho más de lo que plantean las definiciones anteriores.

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]

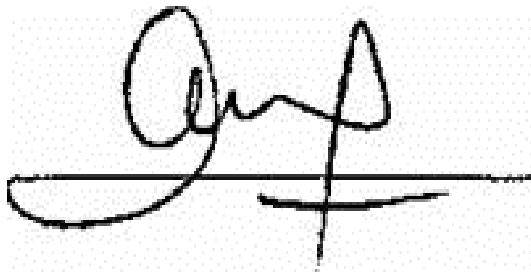


Ilustración 1: Firma manuscrita

Las firmas manuscritas [5], constituyen uno de los principales métodos de justificación de autoría que manejan las personas. Una de sus ventajas radica en la facilidad de realización, pues apenas se necesita un papel y un bolígrafo. Otra ventaja está en la dificultad de realizar una falsificación que pueda engañar a una persona experta. Pero quizás su mayor ventaja esté en que es un mecanismo de autenticación personal ampliamente aceptado por la sociedad, en general, y por organismos públicos y privados, en particular.

1.3.2. Características

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) o un dibujo, pero será algo que le sirva para diferenciarse del resto de las personas. [4]

Por tanto se pueden distinguir las siguientes características que definen una firma manuscrita:

- Identificativa: con ella se identifica al autor de un documento u 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.

1.3.3. Aplicaciones

La firma es aplicada en muchos aspectos de la vida, como por ejemplo: en transacciones bancarias o en otros documentos legales, como puede ser un contrato laboral, un certificado médico, una hipoteca. Cuando se paga con tarjeta de crédito se pide firmar la factura o cuando se va al banco a pedir un crédito se exige poner la firma de la persona que solicita el crédito para dar conformidad a lo que se expone. La firma también es usada en el mundo del arte. En las pinturas se puede observar firmas de sus autores que ponen su nombre real o artístico para que el público pueda reconocer su obra. [4]

1.3.4. La verificación de la firma

El problema de la verificación consiste en que el sistema determina el grado de similitud entre dos firmas, para establecer si son auténticas o no. Como enuncia E. Justino [6] “en el problema de verificación de firmas se trata de maximizar las diferencias interpersonales”. Los falsificadores, que pueden tratar de engañar a un sistema de este tipo, pueden catalogarse en dos grupos: los falsificadores entrenados y los no entrenados [5]. Los primeros, que conocen la firma de la persona que quieren suplantar, se entrenan en reproducir la firma y consiguen unas falsificaciones de gran calidad. Los segundos no sólo no están entrenados, sino que no han visto nunca la firma, por lo que su reproducción no guarda ningún parecido con la auténtica. Curiosamente, y en contra de lo que pudiese pensarse, a este segundo grupo de falsificadores corresponde el 95% del fraude existente en esta modalidad [7].

1.3.5. Reconocimiento on-line y off-line

Dependiendo del modo de adquisición se pueden distinguir dos modalidades de firma, estática (comúnmente a esta modalidad se la denomina también mediante su acepción inglesa off-line) y dinámica. (El término inglés para esta modalidad de firma es on-line).

El reconocimiento *on-line* (Ilustración 3) significa que el proceso se realiza mientras que el usuario está escribiendo por lo que es necesario emplear una tableta digitalizadora y un lápiz especial. Sin embargo

permite contar además de con la firma original con algunos datos extras, como la presión, la velocidad, el punto de inicio y fin de los trazos, etc. [8]

Por el contrario, el reconocimiento *off-line* (Ilustración 2) se realiza posteriormente, es decir, una vez que el usuario ha terminado de firmar. Por ejemplo, se puede escanear un documento con la firma y luego reconocerla. Sin embargo, no hay una separación clara en lo que se refiere a los algoritmos a emplear en cada uno de los dos casos. Un sistema *on-line* puede fácilmente usar algoritmos destinados a un sistema *off-line* simplemente convirtiendo la información a un bitmap⁵. Por el contrario, también hay formas de extraer información de un mapa de bits e intentar adivinar cómo fue dibujado ese trazo (con mayor o menor grado de acierto). [9]

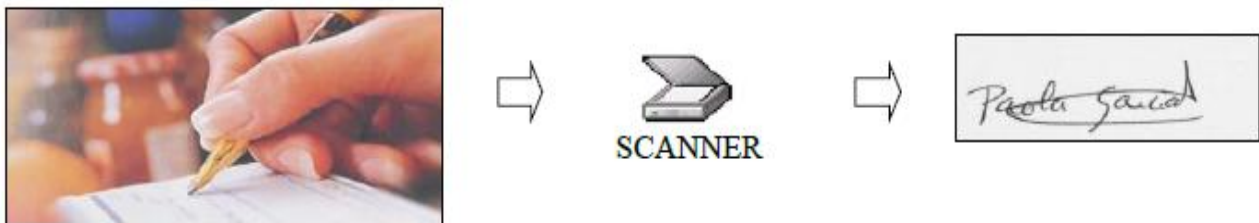


Ilustración 2: Sistema de adquisición de escritura *off-line*

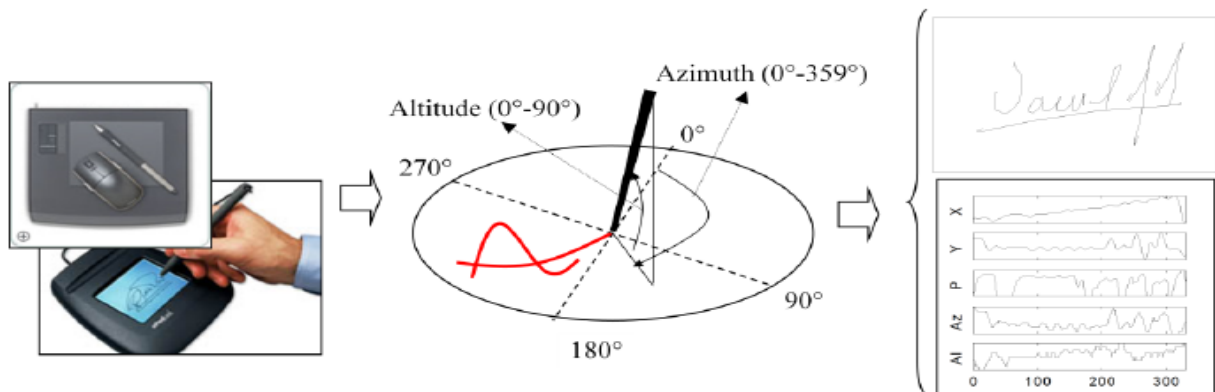


Ilustración 3: Sistema de adquisición de escritura *on-line*

⁵ Mapa de bits

1.3.5.1. Ventajas y Desventajas

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

Para cada punto se puede saber el momento exacto en el que se dibujó lográndose calcular características como la velocidad, la aceleración del lapicero (midiendo la distancia entre dos puntos escaneados en un periodo de tiempo fijo), incluso algunas tabletas permiten saber la presión y ver cómo varía a lo largo de la firma. Teniendo en cuenta que se recibe información inmediata del lápiz, cabe la posibilidad de aprender y ajustar el sistema de reconocimiento en tiempo real mejorando los índices de acierto, incluso el usuario puede enseñar al sistema cómo le gusta escribir ciertas letras o trazos.

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

Por otra parte, al analizar una firma como imagen según un proceso *off-line* se tienen una serie de problemas típicos asociados al Reconocimiento Óptico de Caracteres (OCR), tales como la traslación (vertical y horizontal), la rotación, el problema de dimensionamiento, así como la menor cantidad de información, por mencionar sólo algunos. Sin embargo, presenta alguna ventaja como es la posibilidad de aplicarse sin la presencia del sujeto y resulta menos costoso a nivel de equipamiento. [10]

1.3.6. 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 [11], los que tratan el problema

de la imposibilidad de usar más de una firma por individuo para el entrenamiento [12], los que tratan el problema de la escalabilidad no asistida [13], o que tratan el problema de falsificadores entrenados [14]. 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. Entre estas últimas las más utilizadas son: centros de gravedad (parciales o totales, línea base global (global base line), los límites superiores e inferiores de la firma, número de agujeros de la firma, esqueleto de la firma, tamaño de la envolvente convexa, caja que contiene la firma (bounding box), contorno o perímetro, ejes de mayor y menor inercia, relación entre área y perímetro, densidad de puntos en las regiones de la imagen, ángulo de inclinación (slant), puntos extremos superiores e inferiores, número de trazos, estructura de los trazos, puntos de cruce y de relleno de la firma a partir del eje de mínima inercia.

1.4. Descripción actual del dominio del problema

Actualmente en nuestro país, la Biometría es un tema de mucho auge, no solo como sistema de reconocimiento y verificación de personas sino también en el campo de la salud ha tenido gran impacto obteniendo grandes logros. Cuba cuenta con varios centros dedicados a desarrollar sistemas biométricos para la identificación y autenticación de personas utilizando las modalidades de huella digital, reconocimiento facial, entre otros, pero no cuenta con ninguno que utilice la modalidad de firma manuscrita.

El proyecto CISED de la Universidad de las Ciencias Informáticas (UCI) se ha propuesto realizar un componente para la extracción y verificación de las características de la firma manuscrita, único en su tipo en nuestro país, con el objetivo de optimizar el proceso realizado manualmente y de minimizar el porcentaje de los fraudes que se dan en esta modalidad.

1.5. Tendencias generales para dar solución a este tipo de problemas.

De forma resumida y simplificada las tendencias generales encontradas para la solución del problema de reconocimiento y verificación de las firmas manuscritas obedece al siguiente esquema:

- **Pre-proceso:**

- Limpieza y eliminación del ruido⁶
- Esqueletizado de la imagen utilizando algoritmos de adelgazamiento.
- Etiquetado de los trazos según su inclinación.

- **Extracción de características:**

- Seguimiento y extracción de los trazos.
- Cálculo de la caja de Feret del conjunto de los trazos.
- Normalizado de los trazos respecto a la dimensión de la caja de Feret.

- **Proceso de comparación:**

- Comparación de los trazos de las imágenes y retorno de un resultado basado en porcentos de semejanza.

1.6. Etapas del pre-proceso del reconocimiento de firmas.

El pre-procesamiento de imagen de la firma es muy útil porque ayuda a suprimir información para los objetivos particulares de análisis en un caso dado. [15]

A continuación se hace una breve descripción de cada una de las etapas por las que pasa la imagen en el pre-proceso.

1.6.1. Limpieza y eliminación del ruido

Muchas veces al escanear un documento se introducen errores debido a manchas en el cristal, suciedad en el documento, etc. Estas manchas pueden introducir errores a la hora de interpretar la información. [15]

1.6.1.1. Binarización

La mayor parte de los algoritmos para reconocer escritura están escritos a partir de imágenes binarias, por lo que se hace conveniente el paso de una imagen en niveles de gris (o color) a una binaria, además esto permite reducir el volumen de los datos a tratar. La binarización de una imagen digital consiste en

⁶ Componentes no deseados en una señal que degradan la calidad de los datos o interfieren con las señales deseadas que procesa el sistema.

convertir la imagen digital en una imagen 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 obtiene una gráfica donde se muestran el número de píxeles por cada nivel de gris que aparecen en la imagen. Para binarizar la imagen, se debe elegir un valor adecuado dentro de los niveles de grises (umbral), de tal forma que el histograma forme un valle en ese nivel. Todos los niveles de grises menores al umbral calculado se convierten en negro y todos los mayores en blanco. [16]

1.6.1.2. Segmentación

Una vez obtenida la imagen binaria se deberá fragmentar o segmentar en las diferentes componentes conexas (parte de la imagen donde todos los píxeles son adyacentes entre sí) que la componen.

La fragmentación o segmentación de la imagen constituye una de las mayores dificultades del reconocimiento, y se hace necesaria para poder reconocer cada uno de los caracteres de la imagen binaria. La fragmentación o segmentación es la operación que permite la descomposición de un texto en diferentes entidades lógicas. [15]

1.6.1.2.1. Algoritmos de Segmentación

Los algoritmos de segmentación tienen 3 formas comunes: métodos basados en bordes, técnicas basadas en regiones y técnicas de umbral.

- Segmentación por bordes

Se centran en la detección de contornos. Delimitan los bordes de un objeto y segmentan los píxeles dentro del contorno como pertenecientes a ese objeto. Su desventaja consiste en conectar contornos separados o incompletos lo que los hace susceptibles a fallas. [15]

- Segmentación por regiones

En esta aproximación todos los píxeles que pertenezcan a un objeto se agrupan juntos y son marcados para indicar que pertenecen a una región. Los píxeles son asignados a regiones según algún criterio que los distingue del resto de la imagen. Un criterio muy estricto puede provocar fragmentación mientras uno poco estricto provoca uniones indeseables. [15]

- Segmentación por umbral

Esta técnica segmenta la imagen pixel por pixel. Si el valor de un pixel está dentro del rango de valores especificados para un objeto el pixel es segmentado. Son efectivas cuando todos los objetos y el fondo de la imagen tienen rango de valores diferentes y existe un contraste marcado entre ellos. Como la información de los pixeles vecinos es ignorada, las fronteras de regiones borrosas pueden ocasionar problemas. [15]

1.6.1.2.2. Fundamentación del algoritmo seleccionado

El algoritmo seleccionado para la segmentación de las imágenes de firma manuscrita es el algoritmo de segmentación por umbral, debido a que con este algoritmo las características quedan intactas en cuestión de los rasgos.

1.6.1.3. Esqueletizado.

La esqueletización o adelgazamiento es una técnica muy usada en el reconocimiento de patrones. Consiste en eliminar píxeles de la imagen hasta que queda un esqueleto de un píxel⁷ de grosor. Idealmente, este esqueleto contiene toda la información relevante 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. De hecho, aunque ha habido muchos intentos de resolver el problema, no hay todavía una solución completamente satisfactoria. [17]

1.6.1.4. ¿Qué es el esqueleto de una imagen?

Un esqueleto intenta representar la forma de un objeto con un número relativamente pequeño de píxeles. De esta forma, todos los píxeles del esqueleto son estructuralmente necesarios. La tarea de sacar características de una imagen queda simplificada al obtener su esqueleto. El esqueleto de una imagen se puede extraer fácilmente utilizando alguna de las distintas técnicas de adelgazamiento. [18]

⁷ Elemento de las fotografías. Es el menor elemento de un gráfico que puede recibir un valor de color.

1.6.1.5. ¿Qué es el adelgazamiento?

Se define adelgazamiento como el acto de identificar aquellos píxeles pertenecientes a un objeto que son necesarios para mantener la forma del mismo. Estos píxeles son el esqueleto. [18]

1.6.1.6. Principales algoritmos conocidos para la esqueletización de una imagen.

La mayoría de los algoritmos de adelgazamiento se basan en la eliminación de capas de píxeles de forma iterativa hasta que ya no se puedan eliminar más. Existen un conjunto de reglas que especifican cómo se han de eliminar estos píxeles sobrantes. Cuando estas reglas no se puedan aplicar más a la imagen el algoritmo llegará a su fin.

- Algoritmo de Stentiford.

En este algoritmo, las reglas que especifican como eliminar los píxeles sobrantes se basan en plantillas. Se utilizan cuatro plantillas distintas, cada una utilizada en cada una de las cuatro direcciones en las que se puede recorrer una imagen. Una vez que se encuentra una coincidencia con alguna de las plantillas, el píxel se elimina. Cuando ya no es posible encontrar más coincidencias, se considera que se ha encontrado esqueleto de la imagen. [19]

Problemas del algoritmo:

Los problemas que presenta este algoritmo son clásicos, ya que también se presentan en muchos otros. Son los siguientes:

- Necking: Una intersección ancha de dos líneas se adelgaza en un segmento.
- Tailing: Si dos líneas se unen en un ángulo demasiado cerrado, aparece un segmento extra que no corresponde a la imagen.
- Proyección falsa (line fuzz): Un píxel en el borde de la imagen puede crear un segmento extra que se une al esqueleto.

- Algoritmo de Zhang-Suen

Este método ha sido utilizado durante años como comparación del resto de algoritmos. Es rápido y sencillo de implementar. Es un método paralelo, es decir, cada píxel se puede calcular utilizando el valor

de la iteración anterior. Si se tuviera una CPU por cada píxel, se podría determinar la siguiente iteración instantáneamente. [19]

El método consta de dos sub-iteraciones. En cada una de ellas se eliminarán aquellos píxeles que cumplan todas las reglas definidas para la sub-iteración.

Problemas del algoritmo:

Los resultados son bastante buenos, pero también aparecen los problemas anteriormente descritos, por lo que se hace necesario pre-procesar la imagen antes de adelgazarla.

- Algoritmo de Holt

Este método mejora al anteriormente descrito, es más rápido y además no necesita dos sub-iteraciones. Sólo sobrevivirán aquellos píxeles que cumplan una determinada expresión. [19]

Problemas del algoritmo:

Algunas veces, una vez se ha adelgazado la imagen, es posible eliminar más píxeles aún sin afectar la forma o la conectividad del objeto. Así, se pueden eliminar aquellos píxeles que cuadren en una determinada máscara y cumplan una expresión. Este método se conoce como eliminación en escalera (staircase removal). De todas formas, los problemas anteriormente descritos siguen estando presentes en la imagen adelgazada.

1.6.1.7. Fundamentación del algoritmo seleccionado.

Después del análisis realizado se selecciona el algoritmo de Zhang-Suen [20] para el adelgazamiento de la imagen por las siguientes características:

- Resistente ante ruidos moderados
- Permite un resultado aceptablemente óptimo de la imagen esquelizada.

El algoritmo de Zhang y Suen [20] supone que los píxeles del objeto están marcados como valor uno mientras que el resto están a cero. Los píxeles del contorno son aquellos etiquetados como uno, que tienen al menos uno de los ocho vecinos a cero, el procedimiento aplica los siguientes dos test a los píxeles del contorno:

TEST1: un píxel del contorno se marca para borrado si cumple las siguientes cuatro condiciones:

- (a) $2 \leq N(p1) \leq 6$
- (b) $S(p1) = 1$
- (c) $p2 * p4 * p6 = 0$
- (d) $p4 * p6 * p8 = 0$

Donde $N(p1)$ es el número de vecinos nulos de $p1$, en una 8-vecindad, es decir:

$$N(p1) = p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9$$

Y $S(p1)$ es el número de transiciones 0-1 en la secuencia ordenada $p2, p3, p4, p5, p6, p7, p8, p9$

p9	p2	p3
p8	p1	p4
p7	p6	p5

Ilustración 4: Notación esqueletizado

0	0	1
1	p1	0
1	0	1

Ilustración 5: Ejemplo de transiciones

TEST2: se marcan para eliminar también los píxeles del contorno que verifican las siguientes cuatro propiedades:

- (a) $2 \leq N(p1) \leq 6$
- (b) $S(p1) = 1$
- (c) $p2 * p4 * p8 = 0$
- (d) $p2 * p6 * p8 = 0$

Por tanto, cada iteración de este algoritmo queda esquematizada en el siguiente pseudocódigo:

- a. Para todos los píxeles del contorno, marcarlo como candidato a ser borrado si cumple las cuatro condiciones del Test1.
- b. Una vez recorridos todos los puntos, borrar (poner a 0) todos los píxeles marcados en el paso anterior.
- c. Para todos los píxeles del contorno, marcarlo como candidato a ser borrado si cumple las cuatro condiciones del Test2.

- d. Una vez recorridos todos los puntos, borrar (poner a 0) todos los píxeles marcados en el test 2.
- e. Volver al paso 1 mientras se sigan marcando píxeles.

En la Ilustración 6 se puede observar el efecto del ruido y el resultado de la operación de esqueletizado de Zhang y Suen [20] en la Ilustración 7. Se comprobó que la información de los trazos se mantiene y que la distorsión producida por el ruido se reduce, o por lo menos no introduce mucho error en los trazos o no más que las variaciones que pudiera introducir el autor en posteriores firmas. Por este motivo el sistema del proyecto es resistente al ruido moderado.

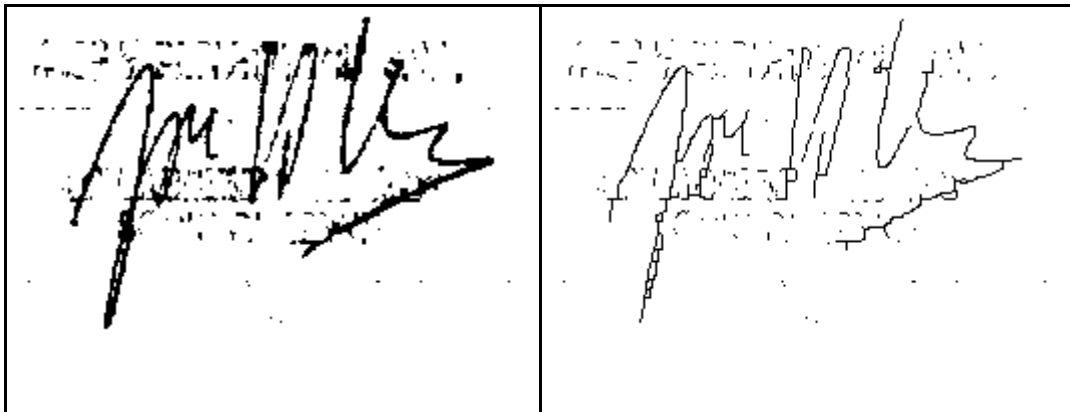


Ilustración 6: Original

Ilustración 7: Resultado esqueletizado

1.6.2. Técnicas de etiquetado de trazos.

El siguiente paso consistirá en detectar los píxeles que forman los trazos, y etiquetarlos según la inclinación del trazo, de esta manera se conseguirá extraer los trazos verticales, horizontales, etc. [21]

1.6.3. Extracción de características.

Las características que se buscan son los trazos que hay en la firma con una orientación de 0° , 45° , 90° y 135° . Para ello y una vez que ya se tienen los píxeles etiquetados con su orientación, se debe hacer un proceso de seguimiento y extracción de los segmentos de color de los ejemplos anteriores.

1.6.4. Reconocimiento basado en trazos.

Como resultado de las etapas anteriores, las firmas se describen ahora mediante una lista de sus trazos que son independientes al tamaño original de la firma y al grosor del bolígrafo utilizado. Un clasificador es un sistema que plantea el problema del reconocimiento como la asignación de los objetos, denominados patrones, a algunos de los grupos representativos, o clases, considerados en el problema. Cada clase representa, por tanto, un tipo diferente de objeto.

1.6.4.1. Descripción del algoritmo Matrices de convolución.

Las matrices de convolución permiten realizar el dominio del espacio filtrado, en el que el resultado de un píxel después de aplicarle una función que se conoce como impulsional depende únicamente de su valor y del de sus vecinos. [22]

Si no se hubiera esqueletizado como etapa de pre-proceso en el análisis de la firma, habría que ir añadiendo matrices de convolución de por ejemplo 9X9 para ir detectando los trazos de distinto grosor, e inclinación. Gracias a que ya se ha esqueletizado se puede reducir el problema a pasar por la imagen de origen cuatro matrices de convolución de 2X2 con las cuatro direcciones como se muestra en la Ilustración 8.

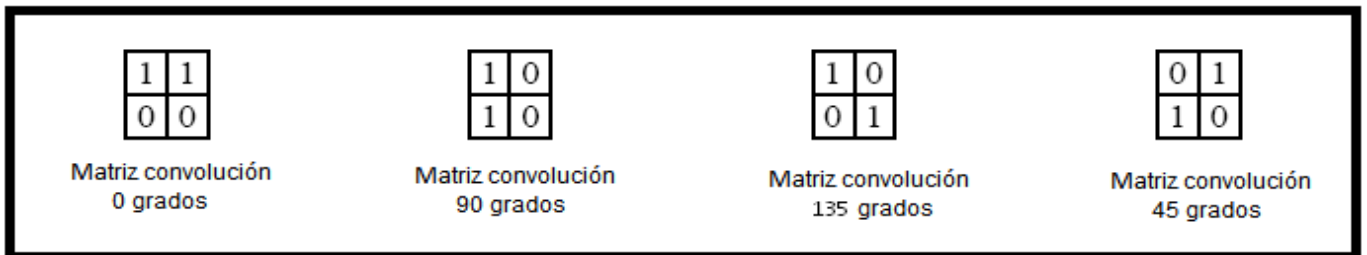


Ilustración 8: Matrices de convolución

Así se obtienen 4 firmas de salida con los trazos de los diferentes ángulos como se muestra en la Ilustración 9.

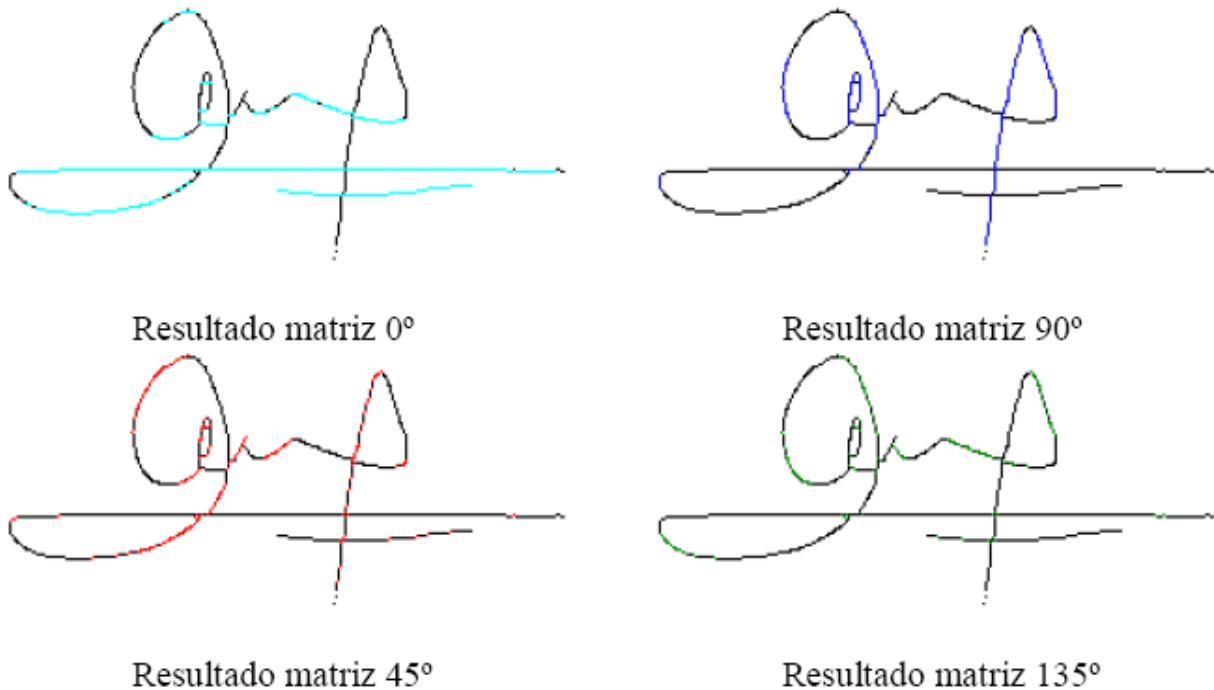


Ilustración 9: Etiquetado de firma según la inclinación

Como se puede apreciar, un mismo píxel puede haber sido etiquetado con cuatro inclinaciones distintas, es decir, un mismo píxel puede formar parte de varios trazos. Esto es interesante en el caso de los cruces de trazos, pues evita pérdida de información.

Otro hecho importante en este proceso es que 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 podría suponer aparentemente un inconveniente, sin embargo, será precisamente este hecho el que permita decir que dos trazos de distintas firmas de la misma persona sean iguales a pesar de que sean ligeramente diferentes.

Lo peculiar de la aproximación, es que aunque en otras firmas del mismo autor éste subrayado sea más horizontal o ligeramente más curvo (hasta los β° siendo $\beta^\circ = 22,5^\circ - \alpha^\circ$ donde α es la inclinación del trazo original) seguirá siendo etiquetado como trazo horizontal, lo que proporciona una gran flexibilidad ante

pequeñas variaciones de los trazos. De esta forma se consigue que si se tienen como entradas dos firmas iguales de la misma persona y una de ellas tenga una ligera inclinación respecto a la otra, el sistema seguirá clasificándolas de igual manera, sin necesidad de tener que rotarlas respecto a su eje de inercia. Lo que otorga una cierta robustez frente a pequeñas inclinaciones, típicas del proceso de adquisición de imágenes y respecto a las variaciones de los trazos habituales en la escritura manuscrita.

1.6.5. Algoritmo de seguimiento y extracción de trazos.

Este algoritmo facilita el trabajo con la firma pues confecciona una lista de trazos que será con la que se trabaje de ahora en adelante, obviando la representación gráfica.

1.6.5.1. Descripción del algoritmo Búsqueda con retroceso.

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

En cada iteración sólo se considera un sucesor, e igualmente se eliminan los nodos que sean una "vía muerta". Cuanto mejor sea el criterio utilizado para limitar el número de estados considerados, más eficiente será el proceso de búsqueda. [23]

Ventajas

- Necesita muy poco espacio de almacenamiento y no se generan las ramas del árbol de búsqueda que se encuentren después del camino de la solución.

Desventajas

- Saber cuál es l_p y no establecer ningún orden previo recorrido de los sucesores de un nodo (la eficiencia temporal del método queda limitada por su carácter fortuito).

1.6.6. Cálculo de la caja de Feret

A partir de este momento se comienza a trabajar directamente con la lista de trazos y no con la representación gráfica de la firma. De esta manera, ya no es necesario recorrer la imagen píxel a píxel, mejorando el rendimiento y reduciendo la complejidad que ya no será factor de n siendo n el número de píxeles de la firma como hasta ahora, sino que la complejidad de las próximas operaciones será función de n' siendo n' el número de trazos localizados, en el caso de este punto, referente al cálculo de la dimensión total del conjunto de los trazos caja de Feret [24], la complejidad es $O(n')$.

El siguiente paso en el algoritmo 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 almacenar 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 consigue que las pequeñas manchas o el ruido no afecten a los siguientes procesos de escalado y normalización.

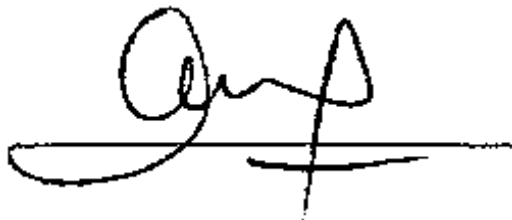


Ilustración 10: Firma Original

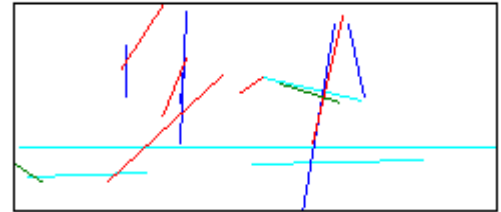


Ilustración 11: Caja de Feret sobre los trazos.

1.6.7. Descripción del algoritmo Ajuste de distancias.

Este método consiste, en normalizar el tamaño de los trazos de las dos imágenes a comparar, seleccionar ciertos puntos de control a una distancia normalizada para obtener un trazo con un determinado número de puntos significativos y calcular la suma de las distancias entre los puntos correspondientes de las dos imágenes.

El principal inconveniente de este método es que no es muy robusto a las transformaciones del patrón y la solución suele estar fuertemente asociada a las características de escritura de un solo usuario. Aunque precisamente, este inconveniente en el reconocimiento de caracteres puede ser de gran ayuda en el

reconocimiento de firmas, ya que es un problema con una fuerte dependencia con la caligrafía de cada sujeto. [24]

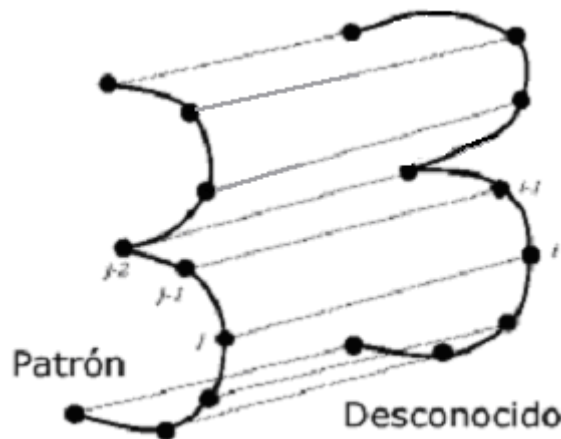


Ilustración 12: Ajuste de distancias (distance matching)

1.7. Tecnologías a utilizar para el desarrollo del trabajo

El desarrollo de un software no es tarea fácil, aunque se piense que si lo es, es por eso que desde hace mucho ha surgido una alternativa que de cierta manera ayude a la realización de los software sin tener mucha dificultad, esta alternativa es la metodología, la cual va acompañada de las herramientas que son aquellas que automatizan la aplicación.

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

1.7.1. Metodologías de Desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software [25]. Si se quiere desarrollar un software de calidad que cumpla con las exigencias de los usuarios, es necesario utilizar una metodología que guíe su desarrollo, para de esta forma poder alcanzar los objetivos esperados. Existen varias metodologías que pueden utilizarse en el proceso de desarrollo del software, entre las más utilizadas están el Proceso

Unificado de Desarrollo (Rational Unified Process, RUP por sus siglas en inglés), Programación Extrema (eXtreme Programming, XP por sus siglas en inglés), MSF (Microsoft Solution Framework), FDD (Feature Driven Development), ASD (Adaptive Software Development), estas cuatro últimas constituyen metodologías ágiles de desarrollo del software.

Las metodologías ágiles de desarrollo de software surgen para dar solución a proyectos donde el entorno de producción es muy cambiante, los plazos de entrega del producto son reducidos, el equipo de desarrollo es escaso (menos de 10 personas), es de interés del usuario que se hagan entregas frecuentes del producto, donde se generan documentos cortos centrados en la funcionalidad, en los que la interacción con el cliente es frecuente y no existe un contrato tradicional o por lo menos es bastante flexible. En cambio las metodologías tradicionales o metodologías pesadas se utilizan cuando el tiempo de vida del proyecto es relativamente largo, existe un equipo de desarrollo en el que los roles están bien definidos, se necesita una documentación abundante que guíe todo el proceso y existe un contrato con el cliente bien definido.

1.7.1.1. Fundamentación de la metodología seleccionada.

Luego de un breve análisis sobre las posibles metodologías a utilizar en el desarrollo del presente trabajo, se decidió utilizar la metodología XP por la descripción que se dará a continuación:

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP [26] sin cubrir los detalles técnicos y de implantación de las prácticas. A diferencia de los procesos tradicionales para desarrollar software, XP asume el cambio como algo natural, y que, indudablemente, en alguna etapa de un proyecto sucede. En XP se realiza el software que el cliente solicita y necesita, en el momento que lo precisa, alentando a los programadores a responder a los requerimientos cambiantes que plantea el cliente en cualquier momento. Esto es posible

porque está diseñado para adaptarse en forma inmediata a los cambios, con bajos costos asociados, en cualquier etapa del ciclo de vida. En pocas palabras, XP “abraza” el cambio. [26]

Características

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto, de esta manera el código es revisado y discutido mientras se escribe.
- Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.

1.7.2. Herramientas Case.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y dinero. Estas herramientas sirven de apoyo en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de 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, entre otras.

1.7.2.1. Rational Rose Enterprise Edition

Rational Rose es una herramienta software para el modelado visual de sistemas software mediante UML. Permite especificar, analizar, diseñar el sistema antes de codificarlo. Esta herramienta mantiene la consistencia de los modelos del sistema software, posee chequeo de la sintaxis UML y generación automática de documentación. [27]

Ofrece:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software"
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables
- Soporte Enterprise Java Beans™ 2.0
- Capacidad de análisis de calidad de código
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.

1.7.2.2. Altova UModel 2007

Altova UModel 2007 es el punto de partida para el desarrollo de un software con éxito. Se utiliza UModel para crear e interpretar los diseños de software a través del poder del UML 2.1 estándar. Elabora el diseño de la aplicación y genera el código Java o C # de sus planes o programas de ingeniería inversa existente en diagramas claros, precisos UML para comprender rápidamente su arquitectura de software. Se puede incluso modificar el código generado o modelos prestados y completar la ida y vuelta de forma automática, la producción de nuevos diagramas o la regeneración de código. De cualquier manera UModel permite mantener el proyecto sincronizado y actualizado. [28]

1.7.2.3. Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a

un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [29]

Visual Paradigm ofrece:

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversión (nueva característica).
- Interoperabilidad con modelos UML2 (meta-modelos UML 2.x para plataforma Eclipse) a través de XML.
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.

1.7.2.4. Fundamentación de la herramienta Case utilizada

Luego de estudiar detalladamente las características y ventajas de las herramientas Case más utilizadas en el CISED, se llegó a la conclusión de que las más factibles para el modelado de este trabajo debido a sus características y potencialidades son Visual Paradigm y Altova UModel 2007.

1.7.3. Framework de desarrollo

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Se puede encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito. En general, el término framework, es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadirle las últimas piezas para construir una aplicación concreta.

1.7.3.1. Net framework 4.0

Se trata de una moderna plataforma que facilita la creación de todo tipo de aplicaciones: de escritorio, de servidor, para XBOX o móviles. .NET Framework 4 se integra a la perfección en los sistemas operativos Windows. Está compuesto, por una parte, por el motor en tiempo de ejecución Common Language Runtime (CLR), el cual administra la memoria, la ejecución de subprocesos y de código, realiza la comprobación de la seguridad del código, la compilación. Por otra parte, incluye la biblioteca de clases base (BCL) orientada a objetos, cuyo uso reduce los tiempos de aprendizaje de las nuevas características de .NET Framework. .NET Framework 4, ha aumentado la compatibilidad con Surface 2.0 SDK y con las nuevas características de Windows 7. Además, ha incluido diversas mejoras relacionadas con: [30]

- Common Language Runtime y la biblioteca de clases base
- Depuración de código
- ASP.NET
- Windows Presentation Foundation (WPF)

1.7.3.2. Visual Studio Ultimate 2010

Microsoft Visual Studio 2010 Ultimate proporciona un entorno integrado de herramientas y la infraestructura de servidor que simplifica el proceso de solicitud de desarrollo. Entrega los resultados de negocio utilizando productivos, los procesos predecibles, personalizable y aumenta la transparencia y la trazabilidad en todo el ciclo de vida con el análisis detallado. Permite la creación de nuevas soluciones o mejorar las aplicaciones existentes con la creación de un prototipo de gran alcance, arquitectura y herramientas de desarrollo que le permiten llevar su visión a la vida dirigidas a un número creciente de plataformas y tecnologías, incluyendo las nubes y la computación paralela. Comprender el incremento de la productividad del equipo mediante la utilización de características avanzadas de colaboración y el uso integrado de las pruebas y herramientas de depuración para encontrar y corregir errores de forma rápida y sencilla la creación de soluciones de alta calidad al tiempo que reduce el coste de desarrollo de soluciones. [31]

1.7.4. Lenguajes

Visual Studio 2010 permite programar en varios lenguajes, dentro de los que se incluyen Visual C++ 7.0, Visual C#, Visual Basic .NET, ASP.NET y JSCRIPT. Incluye además ADO.NET y un gran número de herramientas.

1.7.4.1. Fundamentación del lenguaje seleccionado

C# 4.0 es la última versión del potente lenguaje de programación incluido en el .NET Framework 4.0, abarca una notable variedad de características: tipado dinámico, excepciones, genéricos, colecciones, LINQ, eventos, delegados, métodos anónimos, Threading, patrones, etc.

Tiene como principal característica que se inserta como una opción para programación dinámica, C# 4.0 no lo obliga a usar dynamic, de la misma manera que C# 3.0 no lo obligaba a usar variables, lambdas ni inicializadores de objeto. C# 4.0 brinda la nueva palabra clave dynamic específicamente para que sea más fácil tratar con algunos escenarios ampliamente conocidos. Este lenguaje sigue siendo esencialmente de tipo estático, aunque se ha agregado la capacidad de interactuar de manera más eficaz con objetos dinámicos.

C# es un lenguaje de programación de uso general sencillo, con seguridad de tipos y orientado a objetos. Visual C# ofrece a los desarrolladores herramientas eficaces centradas en código, compatibilidad de lenguajes para crear aplicaciones Web completas y conectadas en .NET Framework. El lenguaje C# también incluye soporte para la programación orientada a componentes. [32]

1.8. Propuesta de selección de tecnologías y herramientas

Anteriormente se realizó el estudio y análisis de algunas de las tecnologías actuales que se aprovechan durante el desarrollo del software, haciendo mención de sus principales características y ventajas. Todo esto les permitió a los autores de este trabajo seleccionar un conjunto de estas tecnologías para el desarrollo del componente.

Como metodología de desarrollo se propone utilizar XP, como herramienta de modelado: Visual Paradigm y Altova UModel 2007, como lenguaje de programación C# 4, para apoyar al proceso de implementación se propone utilizar Visual Studio 2010 y como plataforma de desarrollo .Net Framework 4.0.

Todas estas constituyen la selección de tecnologías para el ideal desarrollo del presente trabajo.

1.9. Conclusiones Parciales

En el estudio realizado en este capítulo se demostró la necesidad de desarrollar un componente para la verificación segura de personas a través de la firma manuscrita. Para esto, se analizaron varios conceptos teóricos, relacionados con el tema, para una mejor comprensión del mismo. Se seleccionaron los algoritmos biométricos Zhang-Suen y Distance Matchig para la solución del problema debido a que la investigación demostró que son los más indicados para lograr una solución factible, además de las tecnologías mencionadas en la propuesta de selección de tecnologías y herramientas, las cuales son capaces de cumplir con las expectativas deseadas.

❖ CAPÍTULO #2: CARACTERÍSTICAS DEL COMPONENTE

2.1. Introducción

En este capítulo se estará tratando la propuesta de solución para responder a la situación problemática, se definen los requerimientos funcionales y los no funcionales, donde los primeros se describen mediante las historias de usuario. Además, se realiza el plan de entrega, en el cual se indican las historias de usuario que se crearán para cada versión de la aplicación y las fechas en las que se publican estas versiones. También se realiza el plan de iteraciones donde se muestran las HU que se realizarán en cada iteración según su prioridad en el negocio.

2.2. Objeto de Informatización

El objeto de informatización es el proceso de verificación de personas a través de su firma manuscrita, para lograr esto, se pretende realizar un componente que permita extraer las características de una firma escaneada y compararlas con las de un prototipo ya almacenado. La automatización de este proceso permitirá minimizar las falsificaciones burdas, así como garantizar el ahorro de costos, tiempo y seguridad.

2.3. Descripción de la solución propuesta

El reconocimiento de firmas, se basa en la búsqueda en una base de datos de aquella firma que más se le parece a una dada, es decir, la que tenga mayor porcentaje de similitud. Sin embargo, la verificación decide si dos firmas son lo suficientemente parecidas como para suponer con un cierto grado de confianza que proceden del mismo autor. La solución que se propone para resolver el problema en cuestión es el desarrollo de un componente que tiene como objetivo principal realizar las funciones anteriormente mencionadas, es decir, reconocer dos firmas manuscritas como pertenecientes al mismo autor y buscar en una base de datos aquella que más se asemeja a una dada.

De forma resumida y simplificada la solución propuesta para el problema de extracción y verificación de las características de las firmas manuscritas obedece al siguiente esquema:

- Pre-procesamiento de la imagen de la firma.

En esta primera funcionalidad se prepara la imagen para pasar la segunda funcionalidad. La imagen es suprimida de aquellas suciedades que impiden que se puedan revelar datos la imagen.

Para que la funcionalidad cumpla con su objetivo debe apoyarse en varias sub-funcionalidades como son: **Binarización, Segmentación, Esqueletización y Etiquetado de trazos.**

- Extracción de las características de la imagen de la firma.

En esta funcionalidad con la utilización de un buen método se obtendrán un conjunto de datos que va a variar a medida que varían las imágenes de firma. Para el cumplimiento de su objetivo se apoyara en las sub-funcionalidades de **Extracción de trazos, FeretBox y Normalización de los trazos.**

- Verificación de las firmas.

En esta funcionalidad se realiza un estado comparativo entre la firma que se desea verificar junto con las que están almacenadas en la base de datos. En esta funcionalidad la sub-funcionalidad utilizada es **DistanceMaching.**

- Decisión de la verificación.

Esta funcionalidad es la que emite el criterio de la verificación, el cual se manifiesta a través del porcentaje de similitud que se establece entre las firmas, siendo este el resultado final del componente.

2.4. Requerimientos Funcionales

En la siguiente tabla se hace referencia a todas las funcionalidades que va a tener el componente. Esta puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El objetivo es asegurar que el producto definido sea el más correcto, útil y competitivo posible, y para esto la tabla debe acompañar los cambios en el entorno y el producto.

A continuación se listan las principales funcionalidades que debe tener el componente de extracción y verificación de las características de la firma manuscrita:

Funcionalidad
1. Pre-procesar la Imagen de la Firma
2. Extracción de características de la Firma
3. Verificar Firmas
4. Mostrar por ciento de similitud

Tabla 1: Funcionalidades del componente

2.5. Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Para resaltar las propiedades o cualidades que hacen este producto más confiable, rápido y usable a continuación se definen los requerimientos no funcionales.

- **Apariencia o Interfaz Externa**

La interfaz de este componente debe contar con un diseño sencillo y de fácil uso, que contenga sólo los gráficos necesarios, con vista a acelerar la velocidad de respuesta hacia el usuario.

- **Rendimiento**

Los tiempos de respuestas de la aplicación deben ser menores de 5 segundos.

- **Portabilidad**

El componente debe funcionar en Windows XP y Windows 7.

- **Usabilidad**

- El uso del diseño debe ser fácil de entender, atendiendo a la experiencia, conocimiento y habilidades de cualquier usuario.
- La aplicación se debe documentar bien, y proporcionar materiales de ayuda para hacer mejor cada uno de los servicios que este ofrece.

- **Restricción en el diseño y la implementación**

Para garantizar el desarrollo de la aplicación se utilizará:

- Lenguaje de programación C# 4.0.
 - Metodología Xp.
 - Visual Studio 2010 como herramienta para la implementación.
 - Visual Paradigm y Altova UModel 2007 como herramientas de modelado.
- **Software**
 - El componente funciona correctamente bajo el sistema operativo Windows XP.
 - El Visual Studio 2010, herramienta con la cual se desarrolla la implementación del componente.
 - **Hardware**
 - Memoria RAM de 512 Mb o superior.
 - Tarjeta de video de 64Mb o superior.
 - Microprocesador de 1 GHz o superior.
 - Disco Duro 120 GHz

2.6. Fase de Exploración: Definición

La metodología de desarrollo XP comienza con la fase de exploración. Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente precisa lo que necesita mediante la redacción de sencillas “historias de usuarios”. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la familiaridad que tengan los programadores con la tecnología. [33]

2.6.1. Historias de Usuario

Las “Historias de usuarios” (“User stories”) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una

estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia [34]. Durante el análisis en la fase de exploración se identificaron 4 HU, cada una de ellas respondiendo a las diferentes funcionalidades solicitadas por el cliente y dando una idea al resto del equipo de desarrollo de cómo debe ser su posterior implementación. Estas se describen a continuación:

<i>Historia de Usuario</i>	
Numero: 1	Nombre: Pre-procesar Imagen de la Firma
Usuario: Sistema	
Modificación de Historia de Usuario: Ninguna	Iteración Asignada: 1
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Puntos Reales: 2
<p>Descripción: Una vez que el usuario tiene las imágenes de la firma y del prototipo, el componente debe pre-procesarlas para eliminar todas las impurezas y mejorar la calidad de las mismas. El componente para el pre-proceso de las imágenes debe seguir los siguientes pasos:</p> <ul style="list-style-type: none"> • Binarización de la Imagen. • Segmentación de la Imagen. • Esqueletización de la Imagen. • Etiquetado de trazos. 	
Observaciones:	

Tabla 2: Historia de Usuario "Pre-procesar la imagen de la Firma"

<i>Historia de Usuario</i>	
Numero: 2	Nombre: Extraer características de la Firma
Usuario: Sistema	
Modificación de Historia de Usuario: Ninguna	Iteración Asignada: 1
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Puntos Reales: 2
Descripción: El componente, una vez que haya realizado el pre-proceso de las imágenes, pasa a la extracción de las características de las mismas. El resultado de esta funcionalidad es un conjunto de trazos de ambas imágenes.	
Observaciones:	

Tabla 3: Historia de Usuario "Extraer características de la firma"

<i>Historia de Usuario</i>	
Numero: 3	Nombre: Verificar Firmas
Usuario: Sistema	
Modificación de Historia de Usuario: Ninguna	Iteración Asignada: 2
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Puntos Reales: 1
Descripción: El componente luego de tener el conjunto de características de las imágenes pasa a realizar la verificación de las mismas comparando los trazos de una con los de la otra.	
Observaciones:	

Tabla 4: Historia de Usuario "Verificar Firmas"

<i>Historia de Usuario</i>	
Numero: 4	Nombre: Mostrar por ciento de similitud
Usuario: Sistema	
Modificación de Historia de Usuario: Ninguna	Iteración Asignada: 2
Prioridad: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 1	Puntos Reales: 1
Descripción: El componente, una vez realizada la verificación de las firmas, devuelve un porcentaje de similitud. Este porcentaje indica si las firmas pertenecen al mismo autor o no. Esto le permite al usuario emitir un criterio sobre la autenticidad de la firma que se desea verificar.	
Observaciones:	

Tabla 5: Historia de Usuario “Mostrar criterio de evaluación”

2.7. Fase de Planificación: Definición

La planificación es una fase corta. En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan 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. Una entrega debería obtenerse en no más de tres meses. Esta fase dura pocos días. [35]

2.7.1. Estimación del esfuerzo por Historia de Usuario.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. [35]

Los resultados obtenidos en esta estimación se exponen en la siguiente tabla:

Historias de Usuario	Puntos estimados
1. Pre-procesar Imagen de la Firma	2
2. Extraer características de la Firma	2
3. Verificar Firmas	1
4. Mostrar porcentaje de similitud	1
Total	6

Tabla 6: Estimación del esfuerzo por Historia de Usuario.

2.7.2. Plan de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para proveer que no vuelvan a ocurrir. [35]

2.7.2.1. Iteración 1

En la primera iteración se realizará la implementación de las dos primeras HU de prioridad alta relacionadas con el trabajo sobre la imagen, y la extracción de las características de la misma, obteniendo al final una primera versión de prueba y dando al componente las primeras funcionalidades.

2.7.2.2. Iteración 2

En la segunda iteración, ya implementadas las funcionalidades principales, se realizará el desarrollo de las últimas historias de usuario con prioridad alta y media dependientes de los resultados obtenidos en las iteraciones anteriores. Se corregirán errores o inconformidades del usuario con las HU implementadas en la iteración anterior. El objetivo de esta iteración es la versión 1.0 del producto final. Esta versión se pondrá en funcionamiento utilizándola en el Centro de Identificación y Seguridad Digital para la evaluación de su comportamiento y rendimiento.

2.7.2.3. Iteración 3

En esta tercera iteración se corregirán los errores que puedan generar las HU implementadas en la iteración anterior, así como inconformidades que puedan existir por parte del usuario.

2.7.3. Plan de duración de las iteraciones

Para una mayor organización del trabajo como lo plantea el ciclo de vida de XP se crea un plan de duración de las iteraciones, en este caso se realizaría un solo plan ya que existe un único equipo de desarrolladores.

Este plan se realiza con el objetivo de reflejar cuáles serán las historias de usuario que serán implementadas en cada una de las iteraciones, así como el tiempo destinado a cada una de ellas y el orden en que se implementarán, lo que ayuda a obtener una idea general del tiempo que durará la confección total del componente.

Iteración	Orden de las HU a implementar	HU a corregir	Duración Total de iteración
Iteracion#1	1. Pre-procesar Imagen de la Firma. 2. Extraer características de la Firma.		4 semanas
Iteracion#2	3. Verificar Firmas 4. Mostrar porciento de similitud	1. Pre-procesar Imagen de la Firma. 2. Extraer características de la Firma.	3 semanas
Iteracion#3		3. Verificar Firmas 4. Mostrar porciento de similitud.	1 semana

Tabla 7: Plan de duración de las iteraciones

2.7.4. Plan de entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

En el plan de entrega que se plantea a continuación se hace una propuesta de la fecha aproximada en que se harán versiones al componente al finalizar cada iteración en la fase de implementación.

Entregable	Final 1era Iteración 2da semana de Abril	Final 2da Iteración 2da semana de Mayo	Final 3ra Iteración 3ra semana de mayo
Calidad	0.1	1.0	1.0

Tabla 8: Plan de entregas

2.8. Conclusiones Parciales

En este capítulo se abordaron las principales fases de la metodología xp, contribuyendo, de esta manera, a definir el alcance del trabajo. Se generaron las HU diseñadas por el cliente posibilitando que el programador tenga una estimación del esfuerzo que debe realizar en cada una, además se definió el plan de entrega de las iteraciones, con el objetivo de tener al cliente informado de todos los avances en el desarrollo del proyecto.

❖ CAPÍTULO #3: DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DEL COMPONENTE

3.1. Introducción

El desarrollo es la parte más importante de la programación extrema. Todos los trabajos tienen como objetivo que se programen lo más rápidamente posible, sin interrupciones y en dirección correcta. También es muy importante el diseño, y se establecen los mecanismos, para que éste sea revisado y mejorado de manera continuada a lo largo del proyecto, según se van añadiendo funcionalidades al mismo. (BECK, 2000).

En este capítulo se estará tratando todo lo referente al diseño, la implementación y las pruebas del componente.

3.2. Diseño

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. El diseño también debe ser incremental y debe estar incrustado en el software, lo cual quiere decir que la estructura de éste debe ser clara. Hay que diseñar lo que las necesidades del problema requieren, no lo que uno cree que debería ser el diseño. Además, siempre hay que tener en cuenta que diseñar cosas para el futuro es una pérdida de tiempo, porque no se van a necesitar.

La metodología XP plantea diseñar lo más sencillo que sea posible, para hacer sólo lo imprescindible en un momento dado, la simplicidad del código y los test continuos hace que los cambios sean posibles tan a menudo como sea necesario. [33]

Para lograr una mejor comprensión sobre el diseño del componente se describirá el modelo de dominio, lo que permitirá, también, que la implementación del componente sea más efectiva.

3.2.1. Modelo de Dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software ni

objetos de software con responsabilidades, sino más bien representa las clases conceptuales u objetos del mundo real en un dominio de interés. [36]

El modelo de dominio se debe concebir como un diccionario visual de abstracciones que será utilizado en fases posteriores y cuya función principal es ayudar a comprender el problema a tratar.

En el modelo de dominio que se muestra en la Ilustración 13 se representan los principales objetos y procesos que serán tratados para darle solución al problema planteado

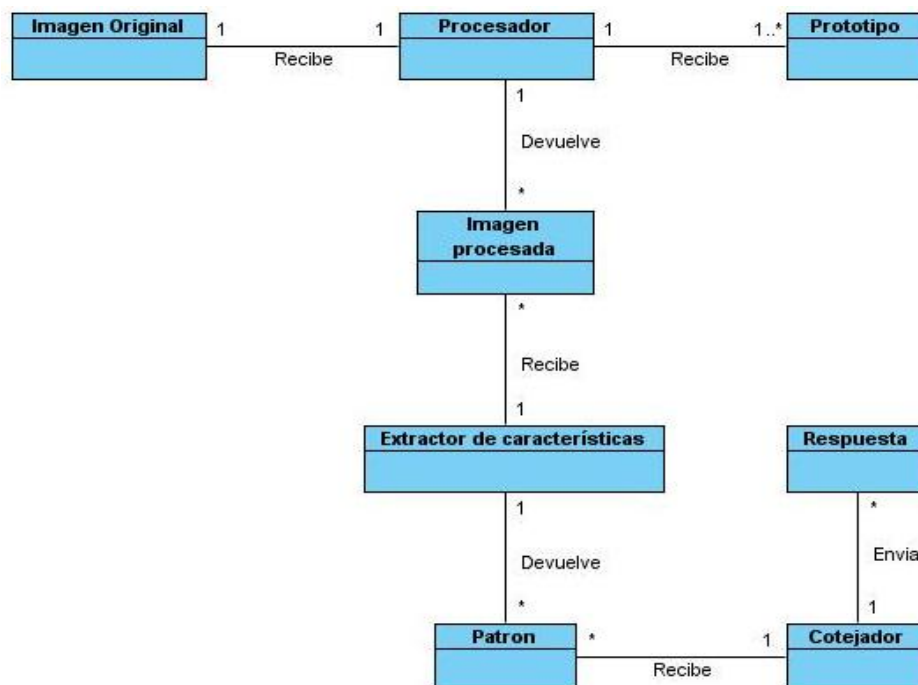


Ilustración 13: Diagrama de Dominio

3.2.1.1. Glosario de Términos del Diagrama de Dominio

- Imagen: firma ya escaneada.
- Prototipo: imagen con la que se quiere comparar.
- Procesador: módulo donde se realiza todo el trabajo con la imagen antes de ser extraídas las características.
- Imagen procesada: es el resultado del procesamiento de la imagen.



Generated by UModel www.altova.com
 Ilustración 15: Contenido del paquete HandwrittenSignatureRecognition (Aplicación de prueba)”

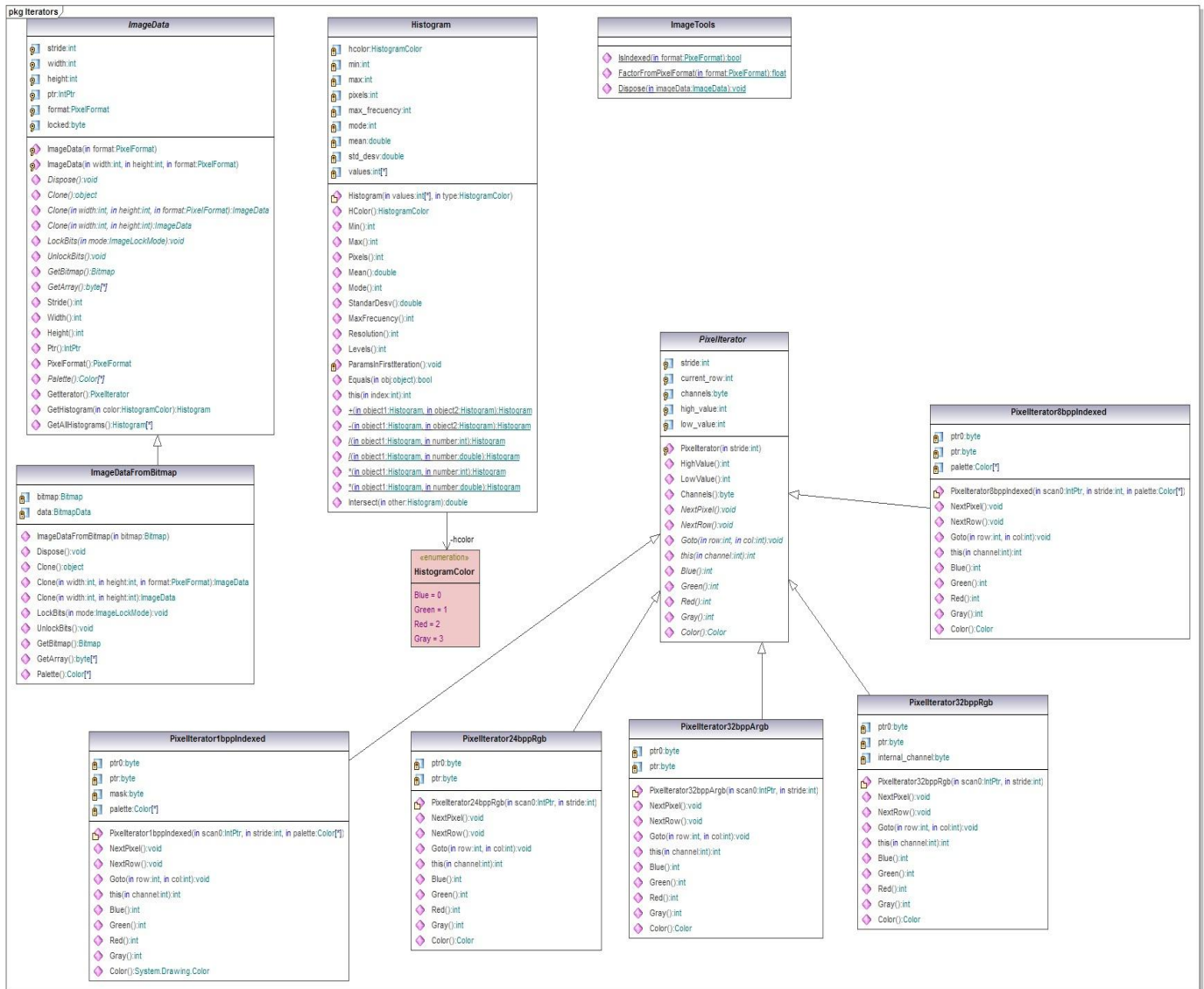


Ilustración 16: Diagrama de clases del "Paquete Iterators perteneciente a ImagesWorking"

3.2.3. Patrones de Diseño

El diseño es un modelo del sistema, que se rige por principios y técnicas que permiten describir detalladamente el sistema para su posterior implementación. Pero todo esto no es suficiente para obtener un buen diseño, los ingenieros además de técnicas utilizan esquemas y estructuras de solución que utilizan repetidas veces en función del contexto del problema. Estos esquemas y estructuras son conceptos reusables y se puede emplear un mismo esquema de solución ante problemas similares. El patrón es un esquema de solución que se aplica a un problema determinado. [37]

"Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles".
(Grady Booch)

Para darle solución a los posibles problemas que se pueden presentar en el diseño del componente se decidió utilizar los patrones GRASP que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Para el diseño del componente se tuvieron en cuenta 4 patrones GRASP: Experto, Creador, Alta Cohesión y Bajo Acoplamiento.

- **Experto en información**

El GRASP de experto en información es el principio básico de asignación de responsabilidades. Indica, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Con este modo de utilizar los patrones GRASP se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento)

Problema: ¿Cuál es el principio general para asignar responsabilidades a los objetos?

Solución: Asignar una responsabilidad al experto en información.

Beneficios: Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

- **Creador**

El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto, o
- Usa directamente las instancias creadas del objeto, o
- Almacena o maneja varias instancias de la clase
- Contiene o agrega la clase.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

- **Alta cohesión**

Cada elemento del diseño debe realizar una labor única dentro del componente, no desempeñado por el resto de los elementos y auto-identificable. Ejemplos de una baja cohesión son clases que hacen demasiadas cosas. En todas las metodologías se considera la refactorización. Uno de los elementos a refactorizar son las clases saturadas de métodos.

- **Bajo acoplamiento**

Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia. Este patrón es un

principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.).

3.2.4. Arquitectura

La arquitectura es de gran importancia en el diseño del software debido a que ofrece una integridad conceptual del sistema. Alude a la estructura general del sistema, es decir, a la estructura jerárquica de los módulos, la manera en que interactúan y la estructura de datos utilizadas por estos módulos.

¿Porque es importante su definición?

- Facilita la comunicación entre los participantes en el desarrollo.
- Resalta las decisiones de diseño que pueden tener un gran impacto en todo el proceso de desarrollo posterior.
- Aporta una visión de cómo se estructura el sistema y como sus componentes trabajan juntos.

Estilo Arquitectónico

Un problema puede satisfacerse utilizando diferentes estructuras a las que se llega utilizando distintas técnicas. [38]

Para satisfacer el problema que se plantea en este trabajo se decidió utilizar el estilo arquitectónico “Arquitectura centrada en los flujo de datos”.

Esta arquitectura se basa en el patrón “pipe and filter” (tuberías y filtro). Consta de un número de componentes denominados filtros conectados entre sí por tuberías que transmiten datos desde un componente al siguiente.

Cada filtro trabaja independiente de los componentes que se encuentran situados antes o después de él. Se diseña de tal manera que se espera un conjunto de datos en un determinado formato y se obtiene como resultado otros datos de salida con un formato específico.

Si el flujo degenera en una única línea de transformación, se denomina secuencial batch.



Ilustración 17: Arquitectura del componente.

3.3. Implementación del componente

Trabajar significa que, al final del día, tienes algo que funcione y que proporcione beneficios al cliente. Por tanto, todo el software se produce mediante la puesta a punto de pequeñas versiones incrementales de producción corta. [39]

Para un mejor desarrollo de la producción la metodología XP plantea una serie de prácticas esenciales las cuales se deben seguir a cabalidad para lograr un proyecto exitoso.

1. **La planeación**, en la cual la opinión del cliente y del equipo de desarrollo deben fusionarse como un todo coherente. Esta práctica se puede ilustrar como un juego conocido como “Juego de Planificación”, donde existen dos tipos de jugadores: Cliente y Programador.
2. **Entregas pequeñas**, que permitan al cliente utilizar el componente con las funcionalidades mínimas lo antes posible, e irlo complementando gradual y continuamente.
3. **Metáforas**, donde la metáfora ayuda a que todo el equipo de desarrollo y el cliente entienda los elementos básicos del componente y las relaciones entre ellos sin la necesidad de una arquitectura muy elaborada y detallada.
4. **Diseño simple**, donde siempre se intenta tener el código más simple, menos redundante y con las funcionalidades estrictamente necesarias en el presente.
5. **Pruebas continuas**, donde es imperativo que los programadores escriban pruebas por cada unidad de código y que el cliente participe en el diseño de pruebas funcionales.
6. **Re-fabricación**, donde este concepto se refiere a mantener una depuración y simplificación constante del componente. Una vez que se ha añadido alguna funcionalidad, es necesario revisar y ser críticos para encontrar puntos de simplificación del código.
7. **Programación en pareja**, donde toda la codificación debe hacerse en parejas de programadores, cada pareja compartiendo un mismo monitor y teclado.

8. **Propiedad colectiva del código:** Aquí se dice que el código puede ser modificado por cualquier elemento del equipo de programación, esto es, que no hay propiedad individual de algún programador sobre alguna sección de código.
9. **Integración continua** (mantener una sola revisión para todo el equipo de programación), tan pronto como pequeños “saltos” en la versión actual sean concluidos. Se recomiendan lapsos entre integraciones de pocas horas a no más de un día.
10. **Semana de 40 horas:** XP afirma que las condiciones de trabajo óptimas para programar “a máxima velocidad” es solo en un turno normal (8 horas).
11. **Ciente in-situ**, donde un representante del cliente debe permanecer por turnos completos en el sitio de programación para contestar cualquier pregunta y ayudar en el desarrollo de pruebas funcionales.
12. **Estándares de codificación.** Esto es esencial para la programación en pares y para la propiedad colectiva del código.

3.3.1. Iteración #1

En esta primera iteración se desarrollan las historias de usuario de mayor prioridad, de las cuales se derivan las otras funcionalidades con el objetivo de obtener la primera versión del componente con las características requeridas para ser mostrado al cliente.

No	Historia de Usuario	Puntos Estimados	Puntos Reales
1	Pre-Procesar Imagen de la Firma	3	2
2	Extraer características de la Firma	3	2
Total		6	4

Tabla 9: Historias de Usuario abordadas en esta iteración.

3.3.1.1. Tareas generadas por cada historia de usuario

<i>Tarea</i>	
Número tarea: 1	Número historia: 1
Nombre tarea: Binarizar la imagen	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 07-03-2011	Fecha fin: 09-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se realizará la binarización de la imagen para obtener la misma pero en blanco y negro.	

Tabla 10: Tarea#1 HU "Pre procesar Imagen de la Firma"

<i>Tarea</i>	
Número tarea: 2	Número historia: 1
Nombre tarea: Segmentar la Imagen	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 10-03-2011	Fecha fin: 12-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se segmentará la imagen para etiquetar cada pixel de acuerdo a la estructura que pertenece (fondo y firma), esto permite separar la imagen en varios objetos (agrupación de etiquetas).	

Tabla 11: Tarea#2 HU "Pre procesar Imagen de la Firma"

<i>Tarea</i>	
Número tarea: 3	Número historia: 1
Nombre tarea: Esqueletizar Imagen	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 14-03-2011	Fecha fin: 16-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se esqueletizará la imagen con el objetivo de eliminar píxeles de la imagen hasta que queda un esqueleto de un píxel de grosor.	

Tabla 12: Tarea#3 HU "Pre procesar Imagen de la Firma"

<i>Tarea</i>	
Número tarea: 4	Número historia: 1
Nombre tarea: Etiquetar trazos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 17-03-2011	Fecha fin: 19-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se detectarán los píxeles que forman los trazos, y se etiquetarán según la inclinación del trazo, de esta manera se conseguirá extraer los trazos aproximándolos a diferentes ángulos.	

Tabla 13: Tarea#4 HU "Pre procesar Imagen de la Firma"

<i>Tarea</i>	
Número tarea: 1	Número historia: 2
Nombre tarea: Extraer trazos.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 21-03-2011	Fecha fin: 24-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Una vez que se tienen los píxeles etiquetados con su orientación se debe hacer un proceso de seguimiento y extracción de los segmentos de color. Esto facilita el trabajo con la firma pues confecciona una lista de trazos que será con la que se trabaje de ahora en adelante.	

Tabla 14: Tarea#1 HU "Extraer características de la Firma"

<i>Tarea</i>	
Número tarea: 2	Número historia: 2
Nombre tarea: Feret Box	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 25-03-2011	Fecha fin: 28-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Al realizar este proceso sobre los trazos de la firma se consigue que las pequeñas manchas o el ruido no afecten al proceso de normalización.	

Tabla 15: Tarea#2 HU "Extraer características de la Firma"

<i>Tarea</i>	
Número tarea: 3	Número historia: 2
Nombre tarea: Normalizar los trazos	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 29-03-2011	Fecha fin: 31-03-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se normaliza el tamaño de los trazos de las dos imágenes a comparar, seleccionando ciertos puntos de control a una distancia normalizada para obtener un trazo con un determinado número de puntos significativos. De esta forma se consigue que los trazos sean independientes a las dimensiones iniciales de la firma, incluso cuando se producen deformaciones de estiramiento.	

Tabla 16: Tarea#3 HU “Extraer características de la Firma”

3.3.2. Iteración #2

En esta segunda iteración se realizarán las funcionalidades de prioridad alta y media que dependen de los resultados de la iteración 1 para su implementación. También se corregirán los errores que se puedan presentar en las HU tratadas en la iteración anterior y por último se mostrará al cliente la versión final del componente.

No	Historia de Usuario	Puntos Estimados	Puntos Reales
1 ⁸	Pre-Procesar Imagen de la Firma	0.3	0.3
2 ⁹	Extraer características de la Firma	0.3	0.3
3	Verificar Firmas	1	1
4	Mostrar porcentaje de similitud	1	1
Total		3	3

Tabla 17: Historias de Usuario abordadas y corregidas en esta iteración.

⁸ HU a corregir en caso de error

⁹ HU a corregir en caso de error

3.3.2.1. Tareas generadas por cada historia de usuario

<i>Tarea</i>	
Número tarea: 1	Número historia: 3
Nombre tarea: Calcular distancias	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 05-04-2011	Fecha fin: 07-04-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Se realiza para tratar directamente con los trazos extraídos y para comparar las imágenes. Se utiliza el algoritmo de cálculo de distancias, con este algoritmo se calcula con muy poco esfuerzo de cómputo, el porcentaje de los trazos de la plantilla que tienen su pareja correspondiente en el prototipo.	

Tabla 18: Tarea#1 HU "Verificar Firmas"

<i>Tarea</i>	
Número tarea: 1	Número historia: 4
Nombre tarea: Compare	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 08-04-2011	Fecha fin: 11-04-2011
Programador responsable: Reinier Hernández Pirez	
Descripción: Devuelve el porciento de similitud entre las firmas.	

Tabla 19: Tarea#1 HU "Mostrar criterio de evaluación"

3.3.3. Iteración #3

El objetivo de esta iteración es corregir los errores que se puedan presentar de la iteración anterior, así como entregar la versión final del producto en caso de error en la iteración #2.

No	Historia de Usuario	Puntos Estimados	Puntos Reales
3	Verificar Firmas	0.3	0.3
4	Mostrar porciento de similitud	0.3	0.3
Total		1	1

Tabla 20: Historias de Usuario a corregir en esta iteración.

3.4. 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.

La metodología xp plantea dos tipos de pruebas a realizar: Pruebas de aceptación y Pruebas unitarias.

3.4.1. Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. [39]

3.4.2. Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados [39]. Son realizados desde el punto de vista del programador y sirven, además de testear el código, para poder realizar el refactoring del mismo. Cada programador, antes de comenzar a programar, debe preparar los test unitarios. Esto hace que dichos test estén preparados para ser corridos durante la codificación y además, hace que al programador le surjan dudas y pueda evacuarlas con el cliente antes de empezar con la codificación.

Para la validación interna del componente, se realizaron pruebas unitarias a varias funcionalidades, como ejemplo se muestran las Ilustraciones 18 y 20. Las pruebas fueron realizadas a las funcionalidades *Compare* y *Pattern* respectivamente, y las Ilustraciones 19 y 20 muestran los resultados arrojados por estas pruebas.

```

/// <summary>
///A test for Compare
///</summary>
[TestMethod()]
public void CompareTest()
{
    Processor processor = new Processor();
    processor.Add(new Segmentation(Thresholds.WhitePercent, Thresholds.BlackPercent));
    processor.Add(new Esqueletization());
    processor.Add(new InformationExtractor());
    string filename1= "C://Users/rej/Desktop/Handwritten Signature Recognition/prototipo/101-1.bmp";
    string filename= "C://Users/rej/Desktop/Handwritten Signature Recognition/firma/101_2.bmp";
    Image i= new ImagesWorking.Image(filename1);
    Image i1= new ImagesWorking.Image(filename);
    processor.ExecuteOperations(i);
    processor.ExecuteOperations(i1);
    Pattern target = i.Pattern; // TODO: Initialize to an appropriate value
    Pattern pattern = i1.Pattern; // TODO: Initialize to an appropriate value
    int threshold = 60; // TODO: Initialize to an appropriate value
    //int expected = 0; // TODO: Initialize to an appropriate value
    int actual;
    actual = target.Compare(pattern, threshold);
    Assert.AreEqual(actual> threshold,true);
    //Assert.Inconclusive("Verify the correctness of this test method.");
}
    
```

Ilustración 18 Prueba unitaria realizada al método *Compare*

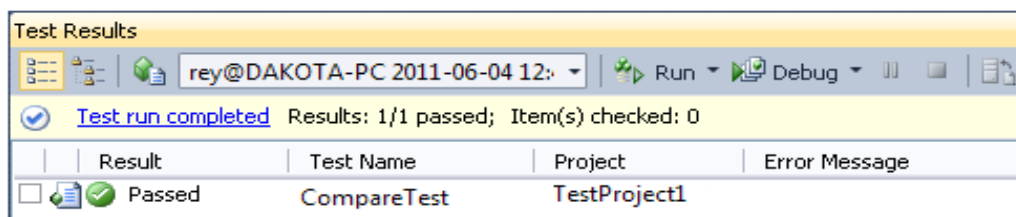


Ilustración 19: Resultado de la prueba unitaria realizada a la funcionalidad *Compare*

```

/// <summary>
///A test for Pattern
///</summary>
[TestMethod()]
public void PatternTest()
{
    Processor processor = new Processor();
    processor.Add(new Segmentation(Thresholds.WhitePercent, Thresholds.BlackPercent));
    processor.Add(new Esqueletization());
    processor.Add(new InformationExtractor());
    string filename = "C://Users/rey/Desktop/Handwritten Signature Recognition/firma/101_2.bmp";
    Image target = new ImagesWorking.Image(filename); // TODO: Initialize to an appropriate value
    Pattern pattern; // TODO: Initialize to an appropriate value
    pattern = target.Pattern;
    bool actual = pattern.BoolPrueba();
    bool expected= false;
    Assert.AreEqual(expected, actual);
    //Assert.Inconclusive("Verify the correctness of this test method.");
}

```

Ilustración 20 Prueba unitaria realizada al método *Pattern*

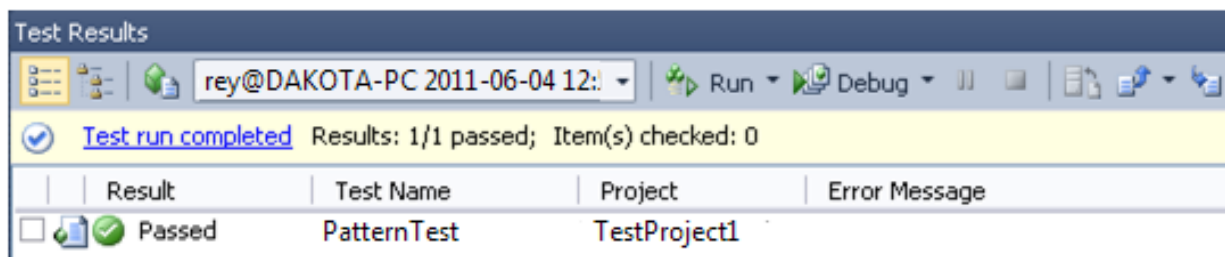


Ilustración 21 Resultado de la prueba unitaria realizada a la funcionalidad *Pattern*

3.4.3. Prueba realizada a la aplicación integrando otras aplicaciones.

Para la realización de esta prueba se integró la aplicación del “Componente para la extracción y verificación de la firma manuscrita” con la “Plataforma de prueba de algoritmos biométricos” utilizando como base de datos la “Base de Datos de referencia para la Prueba de Sistemas de Identificación de personas a través de Firmas Manuscritas”, ambos desarrollados en el CISED.

Para la recogida de firmas en la base de datos participaron 16 personas a las cuales se les tomó 3 muestras por cada 1. (Ilustración 27: Firmas utilizadas en la validación de la propuesta)

El procesado del componente en la plataforma de prueba ofreció los siguientes resultados:

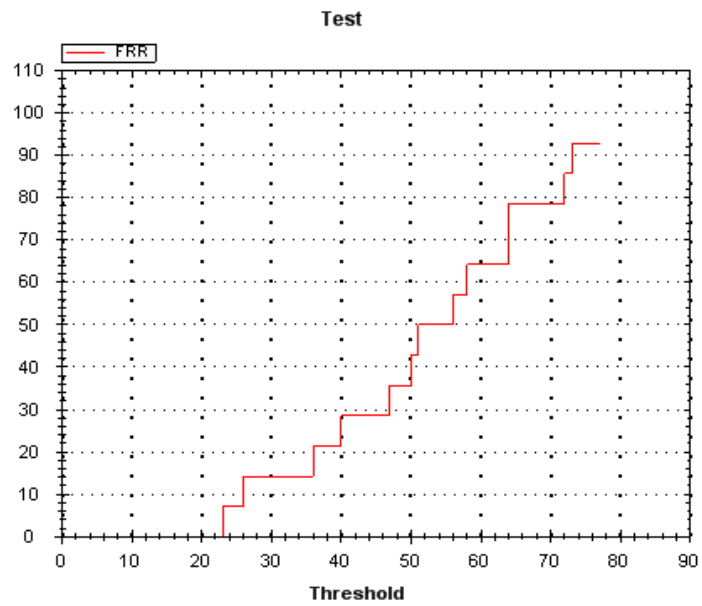


Ilustración 22: Gráfica del FAR.

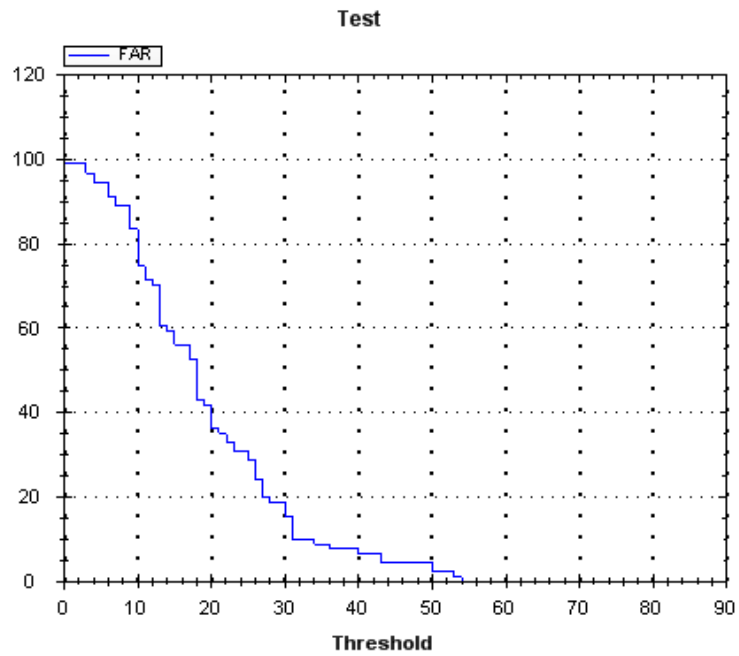


Ilustración 23: Gráfica del FRR.

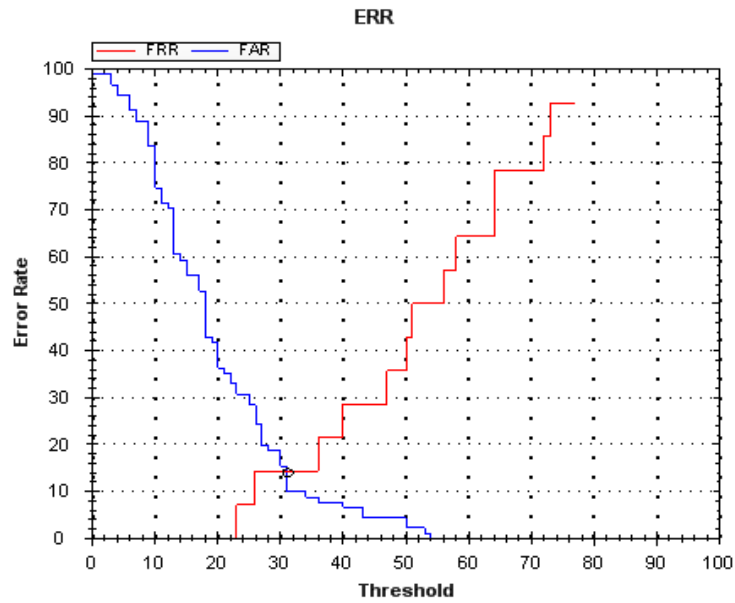


Ilustración 24: Gráfica del FAR vs FRR. EER.

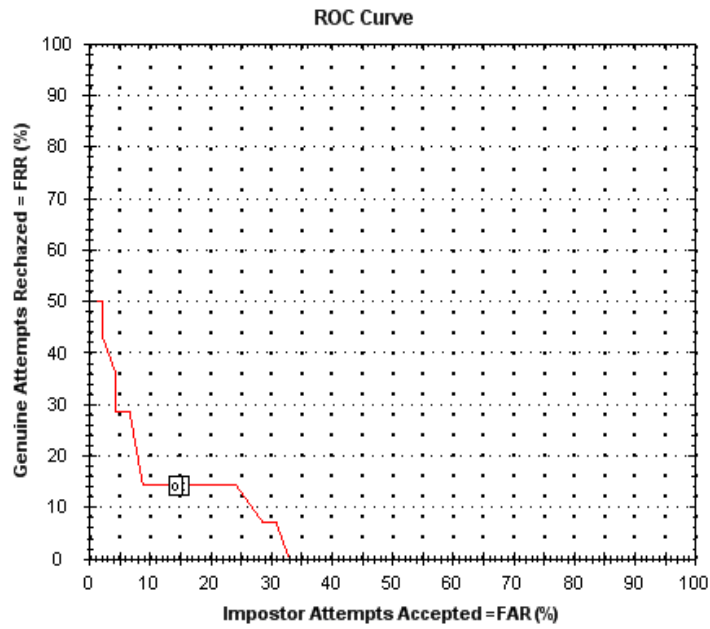


Ilustración 25 Gráfica del ROC.

El análisis del componente arrojó que para un valor fijo del FAR en 10% se obtiene un FRR de 64,2857 %. Se obtuvo un valor de EER = 14,8351%

3.4.4. Análisis de la EER de otros trabajos relacionados con este tema

Es difícil establecer una comparación entre diferentes técnicas de verificación de la firma sobre la base de diferentes bases de datos [40]. Por lo tanto, aquí, sólo se comparó el rendimiento alcanzado por algunos de ellos.

1. Una verificación automática de la firma manuscrita, sistema basado en una arquitectura multi-capa de expertos obtenidos, tasa de falsa aceptación (FAR) y tasa de falso rechazo (FRR) para falsificaciones calificadas 19,8% y 2,04% respectivamente, y EER de 10,92%. [41]
2. Sistema en línea de verificación de la firma manuscrita con el Modelo Oculto de Markov (HMM) con una tasa de falsa aceptación (FAR) de 4% y una tasa de falso rechazo (FRR), de 12% para ambos al azar y falsificaciones calificadas y EER de 11.5%. [42].
3. Distancia de la alineación dinámica basada en línea, método de verificación de firmas presentadas por Amac et al. [40] ha logrado 7,83% de EER con conjunto de datos de falsificaciones calificadas.

Además de las técnicas de verificación antes mencionados, la Primera Competencia Internacional de Verificación de la Firma (SVC2004) ha probado más de 15 sistemas de la industria y la academia, y se encontró que la mejor tasa de error igual es 2,84%. [43]

3.5. Conclusiones Parciales

En este último capítulo se generaron los artefactos de las últimas fases de la metodología xp. Se modelaron el diagrama de dominio y los diagramas de clases del diseño con el objetivo de lograr una mayor comprensión y organización del mismo. Las pruebas realizadas al componente obtuvieron resultados satisfactorios, demostrando el correcto funcionamiento y la buena calidad del mismo. Se quiere destacar que el porcentaje de error devuelto por la prueba realizada a la aplicación es menor que el porcentaje de error establecido mundialmente para los sistemas biométricos, lo que demuestra una vez más el buen trabajo realizado.

El Centro de Identificación y Seguridad Digital está enfrascado en la realización de varias aplicaciones con las cuales se propone alcanzar un alto desarrollo tecnológico tanto en la Universidad de las Ciencias Informáticas como en todo el país.

La realización de este trabajo es de gran importancia por los beneficios que ofrece:

- Aumenta los beneficios económicos.
- Posibilita que la tecnología para el reconocimiento de personas a través de su firma, en nuestro país, este prácticamente al mismo nivel que en los países del primer mundo.
- Aumenta la rapidez y seguridad en el proceso de verificación de firmas manuscritas.
- Posibilita su uso en muchas áreas de la sociedad, facilitando este trabajo y reduciendo obstáculos.

❖ CONCLUSIONES

- El estudio realizado a los procesos de identificación y autenticación biométrica permitió conocer un poco más sobre la situación actual nacional con respecto a estos procesos.
- Se demostró la necesidad de implementar un componente para la automatización del proceso de reconocimiento de personas a través de su firma manuscrita.
- El levantamiento de requisitos funcionales y no funcionales permitió diseñar de manera sencilla y eficaz el Componente para la extracción y verificación de las características de la firma manuscrita.
- Se obtuvo una versión funcional de la solución propuesta y fue puesta a prueba.

En aras de la automatización de los procesos de reconocimiento biométrico en el CISED, la investigación concluye con la satisfacción de haber cumplido exitosamente con el objetivo general propuesto y con las expectativas del cliente. Nada de esto hubiera sido posible sin:

- El estudio intensivo.
- El trabajo en equipo.
- El apoyo de los tutores.
- Los señalamientos del tribunal.
- La dedicación de los autores.

Gracias a todo lo expuesto anteriormente a partir de este momento la Universidad de las Ciencias Informáticas cuenta con un componente para la extracción y verificación de las características de la firma manuscrita a través de un sistema reconocedor de personas.

❖ RECOMENDACIONES

Con el fin de realizar mejoras a la solución propuesta se recomienda:

- Continuar investigando acerca del tema con el objetivo de agregar nuevas funcionalidades.
- Añadir una nueva etapa de pre-proceso que realice el cálculo del eje de inercia de la firma y de esta forma rotarla para hacer que este eje forme un ángulo de 0° con la horizontal.
- El sistema podría ser escalable si se organizan las firmas por el número total de trazos, o jerárquicamente por la cantidad de trazos horizontales, verticales, etc., así se mejoraría el tiempo de respuesta del componente.
- En lugar de almacenar directamente los bitmap o mapas de bits, almacenar en la base de datos las listas de los trazos que la componen, esto reduciría la necesidad del espacio en disco, además de ahorrarse tiempo de cómputo a la hora de comenzar el proceso porque se elimina la etapa de pre-procesamiento de imagen.

❖ REFERENCIAS BIBLIOGRÁFICAS

1. **D, Juan Carlos Landi.** dspace. [Online] [Cited: 1 15, 2011.] Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/814/3/Capitulo1.pdf>.
2. **Hera, Almueda Gilperez de la.** *Reconocimiento off-line de escritura basado en la fusión de características locales y globales.* Universidad Autónoma de Madrid. Madrid : s.n., 2010.
3. **Leal Sistemas.** [Online] [Cited: 1 15, 2011.] Disponible en: <http://www.lealsistemas.com.ar/biometria/historia.php>
4. **José F. Vélez, Ángel Sánchez, Ana B. Moreno, José L. Esteban.** *Verificación Off-Line de Firmas 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,. pp. 643-660. 3.
6. **Justino, E.J.** *The Interpersonal and Intrapersonal Variability Influences of Off-Line Signature Verification using HMM.* Brazil : s.n., 2002.
7. *Investigación y Programas. informe interno.*
8. **Parker, J. R.** *Algorithms for image processing and computer vision.* s.l. : Wiley Computer Publishing. .
9. **Scattolin, P.** *Recognition of handwritten numerals using elastic matching.* Concordia University. 1995.
10. **Powalka, R.K.** *An algorithm toolbox for on-line cursive script recognition.* s.l. : The Nottingham Trent University, 1995.
11. **otros, M. Ammar y.** *Off-line preprocessing and verification of signatures. Int. Journal of Pattern Recog. and Arti. Intel.* 1987.
12. **Ammar, M.** *Off-line preprocessing and verification of signatures. Int. Journal of Pattern Recog. and Arti. Intel.* 1987.

13. **Z. Hou, C. Han.** *Force field analysis snake: an improved parametric active contour model.* *Pattern Recog.* 2005. pp. 513-526.
14. **otros, J. L. Camino y.** *Signature Classification by HMM.* *IEEE 33 International Carnahan Conference.* 1999.
15. **otros, B. Fang y.** *Off-line signature verification with generated training samples.* *IEE Proc.-Vis. Image and Signal Processing.* 2002. Vol. 149.
16. **Néstor O. Balsero García, Diego A. Botero Galeano, Juan P. Zuluaga Morales.** *Reconocimiento e Interpretación de gestos manuales por medio de video.* Pontificia Universidad Javeriana. Bogotá : s.n., 2005.
17. **Hernandez, Alejandro Perez.** *Reconocimiento y verificación de firmas manuscritas off-line basado en el seguimiento de sus trazos componentes.* 2010.
18. **Arias, Segundo Rosendo Cháves.** *Análisis de las Técnicas del Reconocimiento de firmas manuscrita digitalizadas para seguridad y diseño de un prototipo de software.* 2010.
19. **Pedro Real Jurado, José Andrés Armario Sampalo.** *“Procesamiento de Imágenes Digitales.* Universidad de Sevilla.
20. **Suen, T. Y. Zhang and C. Y.** *A fast parallel algorithm for thinning digital patterns.* 1984. pp. 236-239.
21. **Rubine, D.** *Specifying gestures by example.* *Technical report, Information Technology Center.* Carnegie Mellon University. 1993.
22. **José Francisco Vélez, Ángel Sánchez, José Luis Esteban.** *Apuntes Visión Computacional.* Informática de Sistemas , Universidad de Rey Juan Carlos de Madrid. Madrid : s.n.
23. **Parker, J. R.** *Practical Computer Vision .* 1993.
24. **Jr, L. da Fontoura Costa y R. Marcondes Cesar.** *Shape Analisis and Classification.* 2001.
25. **Pressman, Roger S.** *Un enfoque práctico. Ingeniería de Software.*

26. **Beck, K.** *Una explicación de la programación extrema. Aceptar el cambio.* 2002.
27. GSInnova. *Herramientas y Soluciones IBM.* [Online] [Cited: 5 22, 2011.] Disponible en: <http://www.rational.com.ar/herramientas/roseenterprise.html> .
28. Altova UModel 2007. [Online] [Cited: 5 25, 2011.] Disponible en: http://www.download3000.com/download_25675.html.
29. Freedownloadmanager. *Visual Paradigm International Ltd.* [Online] [Cited: 5 22, 2011.] Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p.
30. Malavia .NET Framework 4. [Online] [Cited: 5 25, 2011.] Disponible en: <http://net-framework-4.malavida.com/> .
31. microsoft.com. *Microsoft Visual Studio 2010 Trial Ultimate – ISO Microsoft Download Center* . [Online] [Cited: 5 25, 2011.] Disponible en: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=06a32b1c->.
32. **Burrows, Chris.** MSDN Magazine . *Nuevas características de C# en .NET Framework 4.* [Online] [Cited: 5 22, 2011.] Disponible en: <http://msdn.microsoft.com/es-es/magazine/ff796223.aspx>.
33. **Píriz, Luis Calabria y Pablo.** *Metodología XP.* Universidad ORT Uruguay. 2003.
34. **Penadés, Patricio Letelier y M^a Carmen.** *Métodologías ágiles para el desarrollo de software.* Universidad Politécnica de Valencia.
35. **Escribano, Gerardo Fernández.** *Introducción a Extreme Programming.* Ingeniería del Software . 2002.
36. *Modelo de dominio.* [Online] [Cited: 4 10, 2011.] Disponible en: <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
37. Autentia. [Online] [Cited: 5 16, 2011.] Disponible en: <http://www.autentia.com>.

38. DISEÑO DE LA ARQUITECTURA DEL SOFTWARE. [Online] [Cited: 5 16, 2011.] Disponible en: http://www.slideshare.net/jose_rob/diseo-de-la-arquitectura-del-software.
39. **Joskowicz, José.** *Nuevas Técnicas de Desarrollo de Software en Ingeniería Telemática*. Ingeniería Telemática , Universidad de Vigo. España : s.n., 2008. p. 15, Doctorado.
40. **B. Fang, C.H. Leung, Y.Y. Tang, P.C.K. Kwork, K.W. Tse and Y.K. Wong.** Offline signature verification with generated training samples. s.l. : IEE Proceedings on Vision, Image and Signal, 2002. Vol. 2, 149, pp. 85-90.
41. *Signature verification: increasing performance by a multi-stage system.* **M.Vento., C. Sansone and.** s.l. : International Journal on Document Analysis and Applications, 2000. Vol. 3, pp. 169-181.
42. *A new on-line signature verification algorithm using variable length segmentation and hidden markov models.* **Hamid., M. S. Mohammad and R. R.** s.l. : Proceeding of the Seventh International Conference on Document Analysis and Recognition, 2003.
43. **Ethem, H. Amac and A.** Dynamic alignment distance based online signature verification. s.l. : The 13th Turkish Symposium on Artificial Intelligence and Artificial Neural Networks, Izmir,, 2004.

❖ BIBLIOGRAFÍAS CONSULTADAS

García, Nicolás Luis Fernández. *Contribución al reconocimiento de objetos 2D mediante detección de bordes en imágenes de color.* Córdoba : Universidad Politécnica de Madrid, 2002.

Martínez, M.C. Saúl González Campos y M.C. Luis Felipe Fernández. *Programación Extrema: Prácticas, Aceptación y Controversia.* s.l. : Universidad Autónoma de Cd. Juárez.

Solís., Manuel Calero. *Una explicación de la programación extrema (XP).* Madrid : V Encuentro usuarios xBase, 2003.

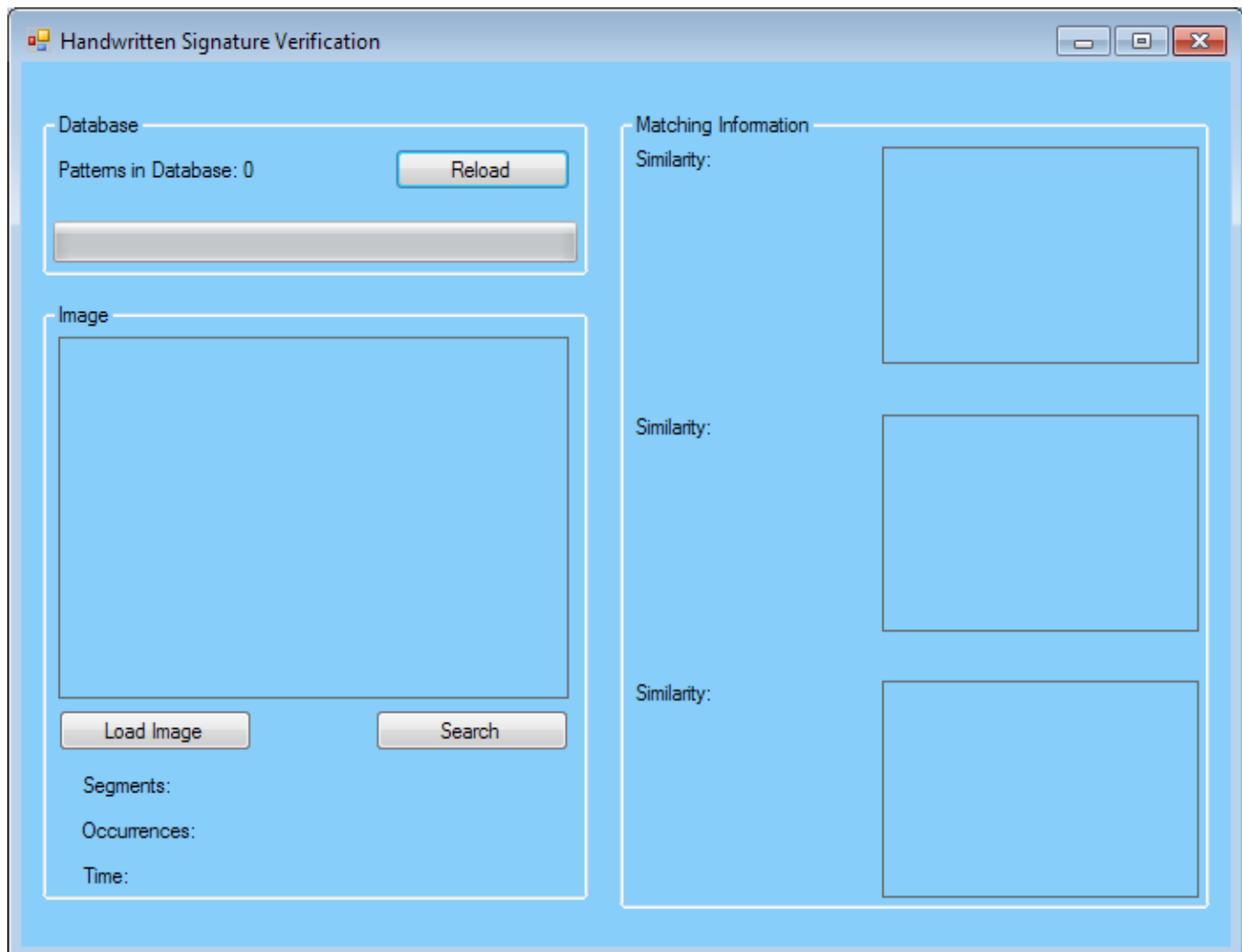
Penadés, Patricio Letelier y M^a Carmen. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* s.l. : Universidad Politécnica de Valencia.

Cristiá, Maximiliano. *Introducción a la Arquitectura de Software.* s.l. : Universidad Nacional de Rosario, 2007.

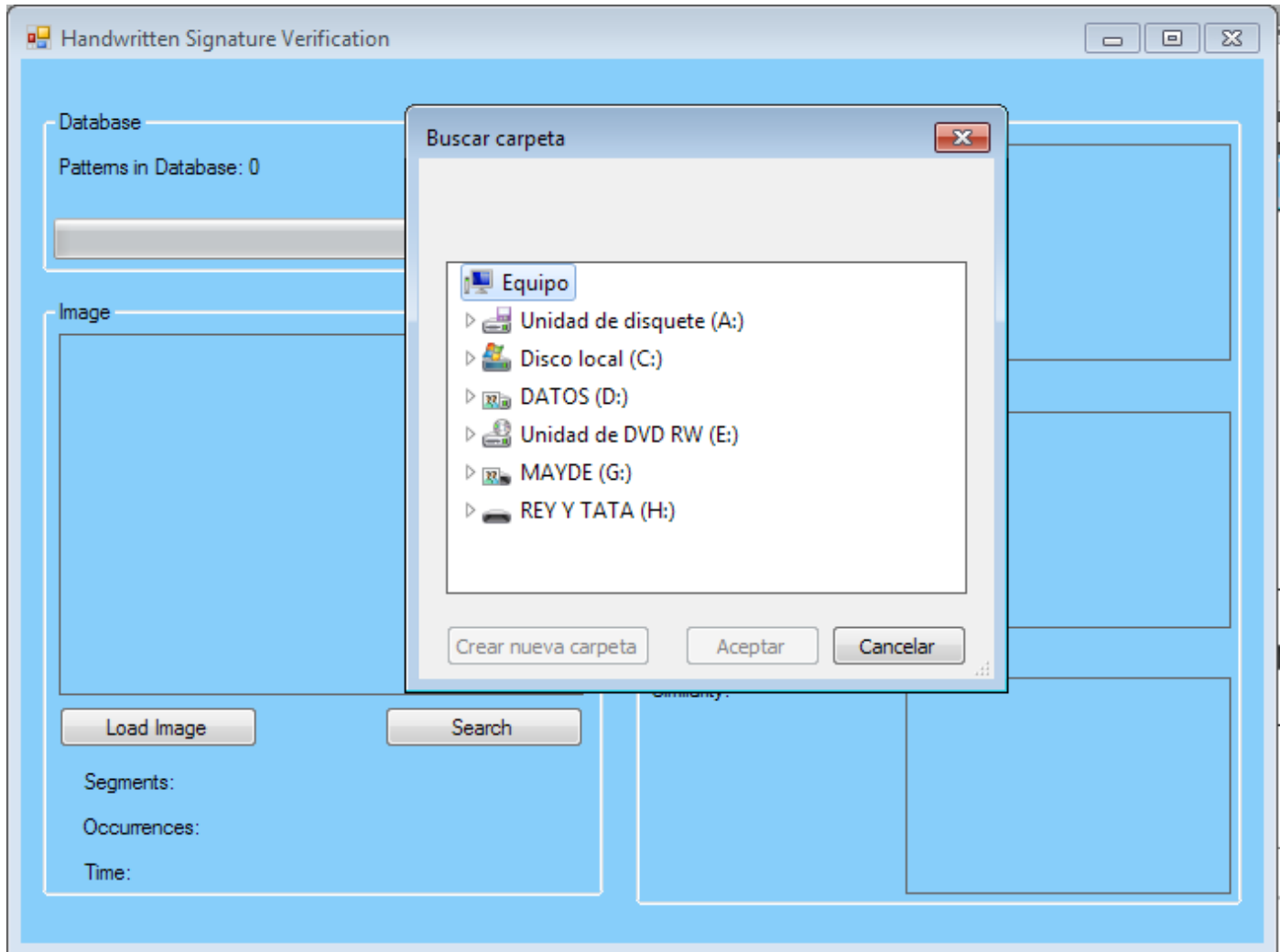
Hernández, Alejandro Pérez. *Reconocimiento y verificación de firmas manuscritas off-line basado en el seguimiento de sus trazos componentes.* 2002/2003.

“Herramientas Case” Fecha de Acceso: 07-03-2011. Disponible en:
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>

González Seco, José Antonio. “Qué es C# “. Fecha de Acceso: 12-03-2011. Disponible en:
<http://www.desarrolloweb.com/articulos/561.php>

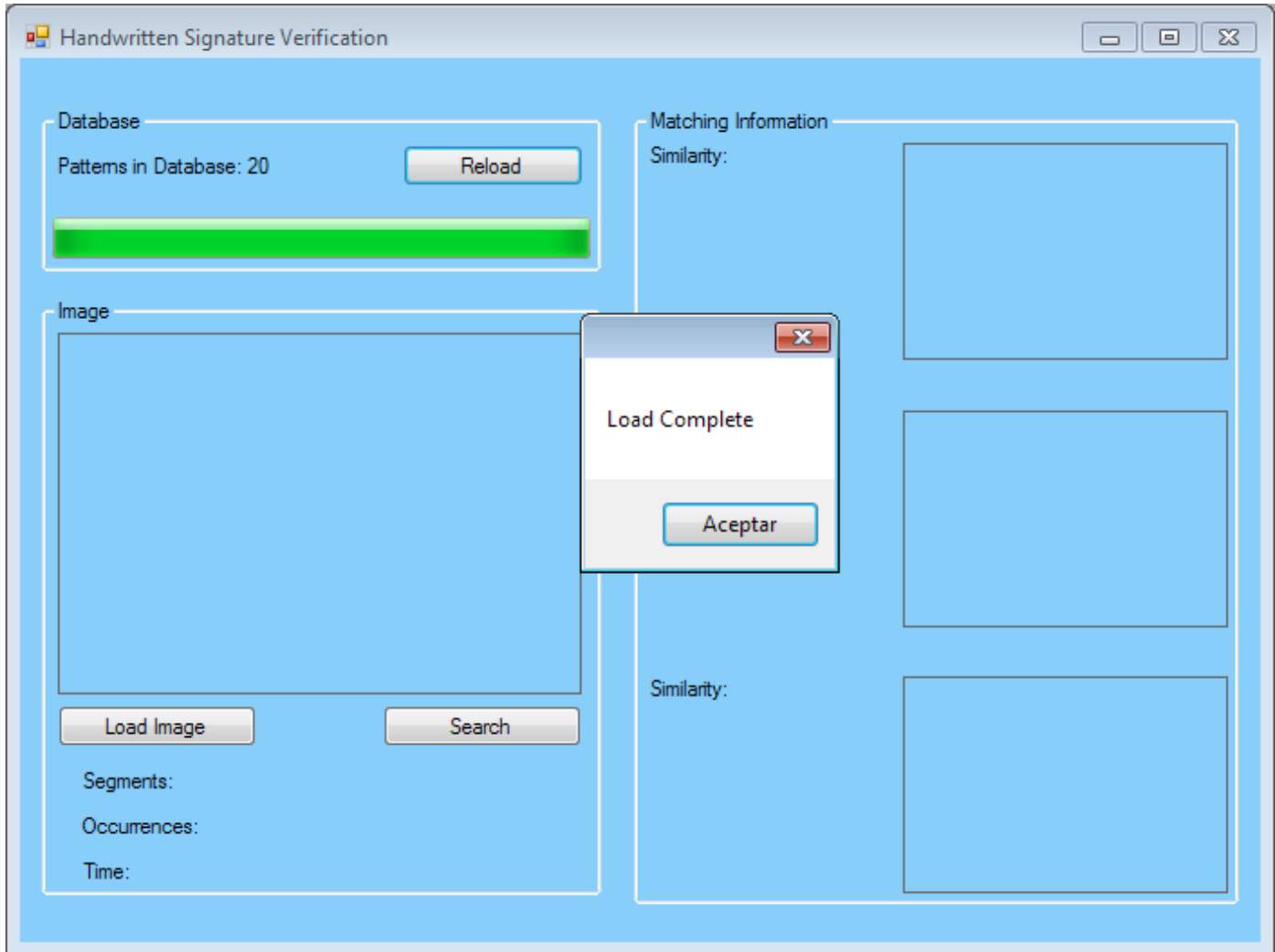
Ilustración 26: Manual de usuario**1. Pantalla principal de la Aplicación.**

2. Selección de la Base de Datos



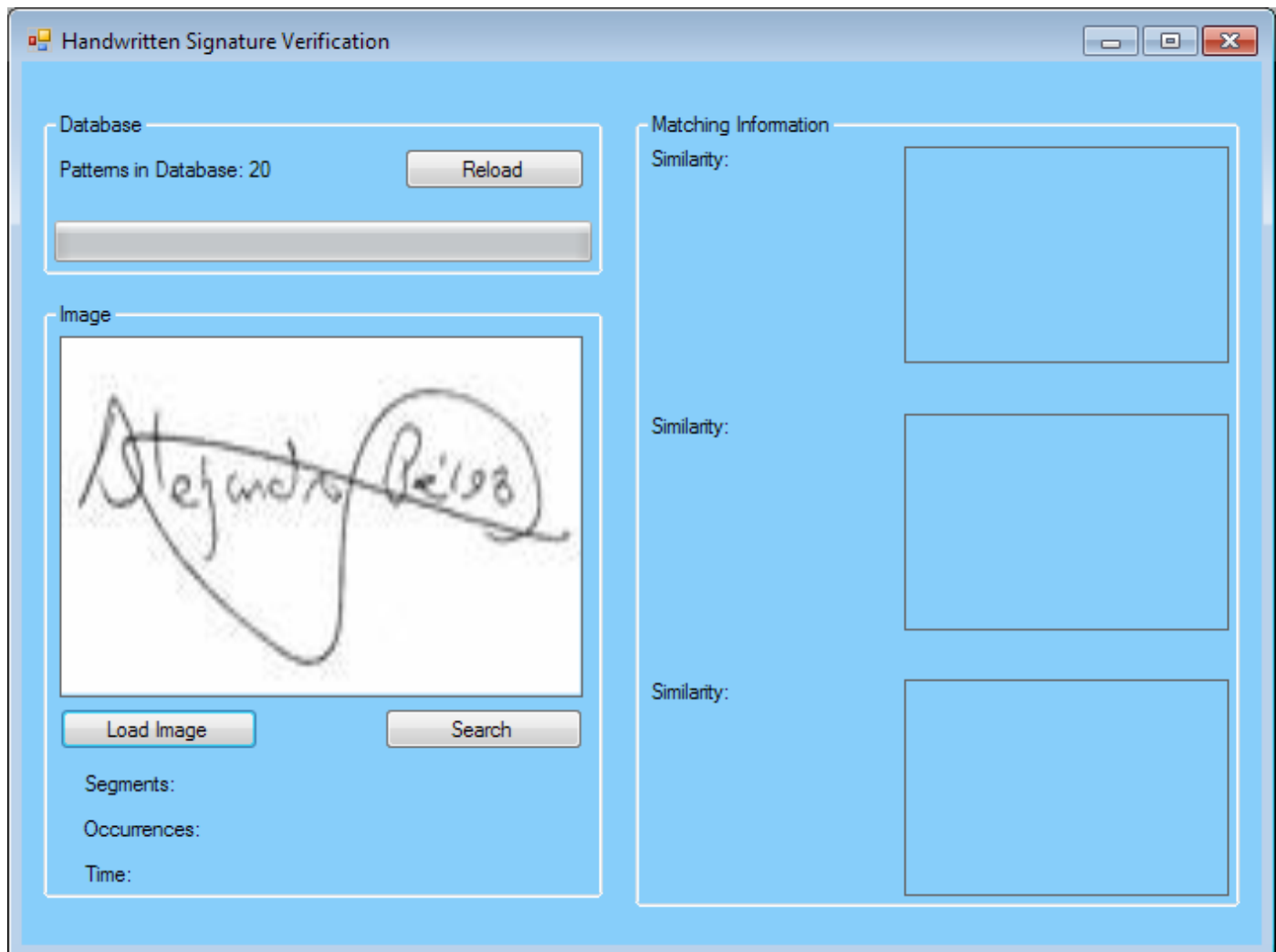
Se da clic en "Reload" y aparece una ventana con las posibles ubicaciones de las firmas que se desean almacenar, se selecciona la carpeta y comienza el proceso de actualización.

3. Carga de la Base de Datos



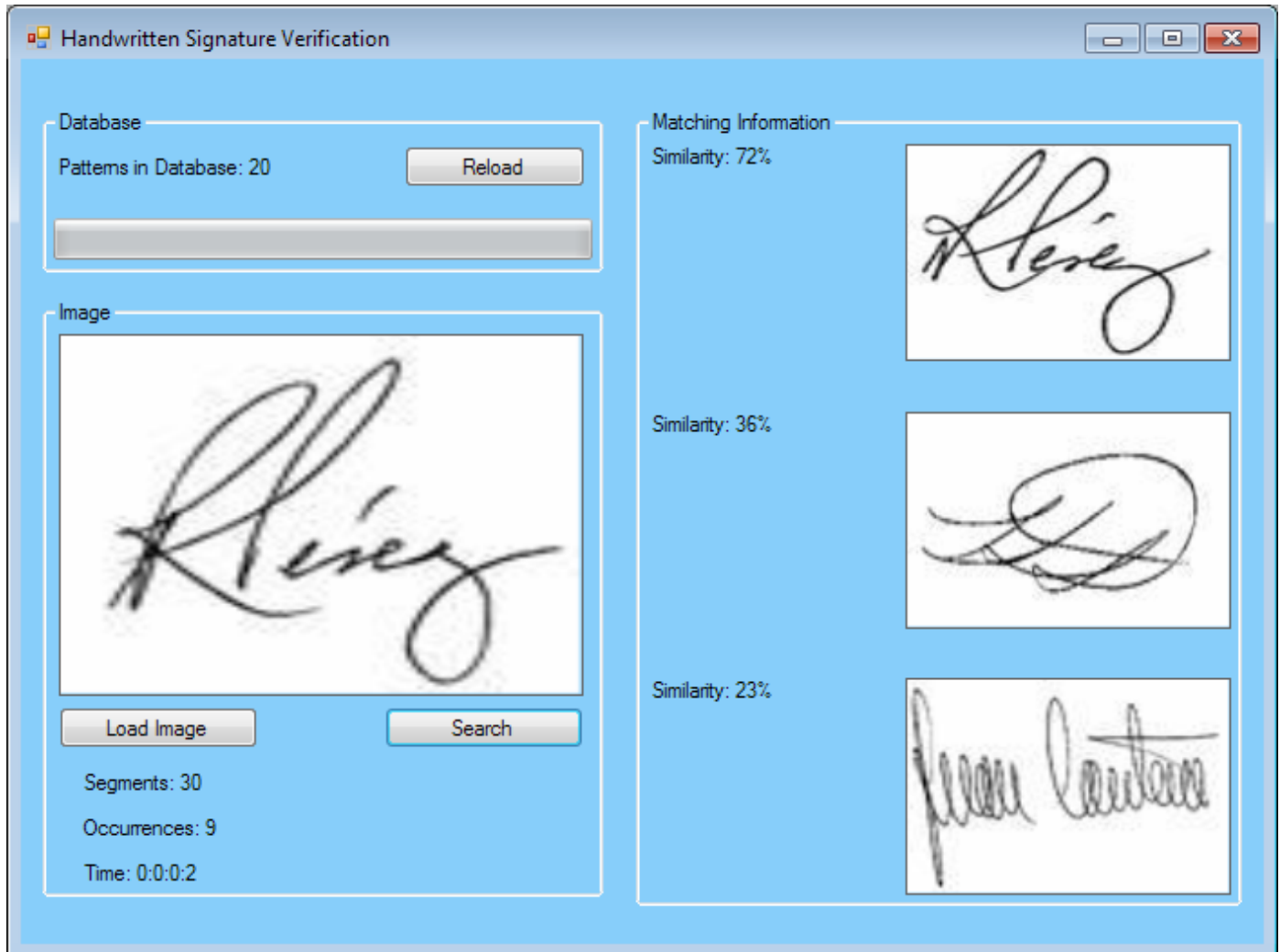
Luego de efectuada la operación de carga se muestra el cuadro de diálogo "Load Complete", que indica que la carga ya está completa y en "Patterns in Database" se muestra el total de firmas almacenadas en la base de datos.

4. Selección de la firma a verificar



Se da click en "Load Image" y se selecciona la imagen de la firma que se desea comparar.

5. Similitud entre firmas

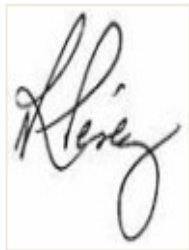


Para hallar la firma que estamos buscando se da click en "Search" y a la derecha aparecen las 3 firmas que más se asemejan a la que estamos buscando, dando el porcentaje de similitud que existe entre las firmas.

Ilustración 27: Firmas utilizadas en la validación de la propuesta



101_2



102_2



103_2



104_2



105_2



105_3



106_2



106_3



107_2



108_2



109_2



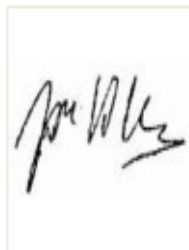
110_2



111_2



112_2



113_2



114_2



115_2



115_3



116_2



116_3



Ilustración 28: Prototipos de las firmas



Ilustración 29: Arquitectura de un Sistema Biométrico



Ilustración 30: Módulo de Identificación y Verificación

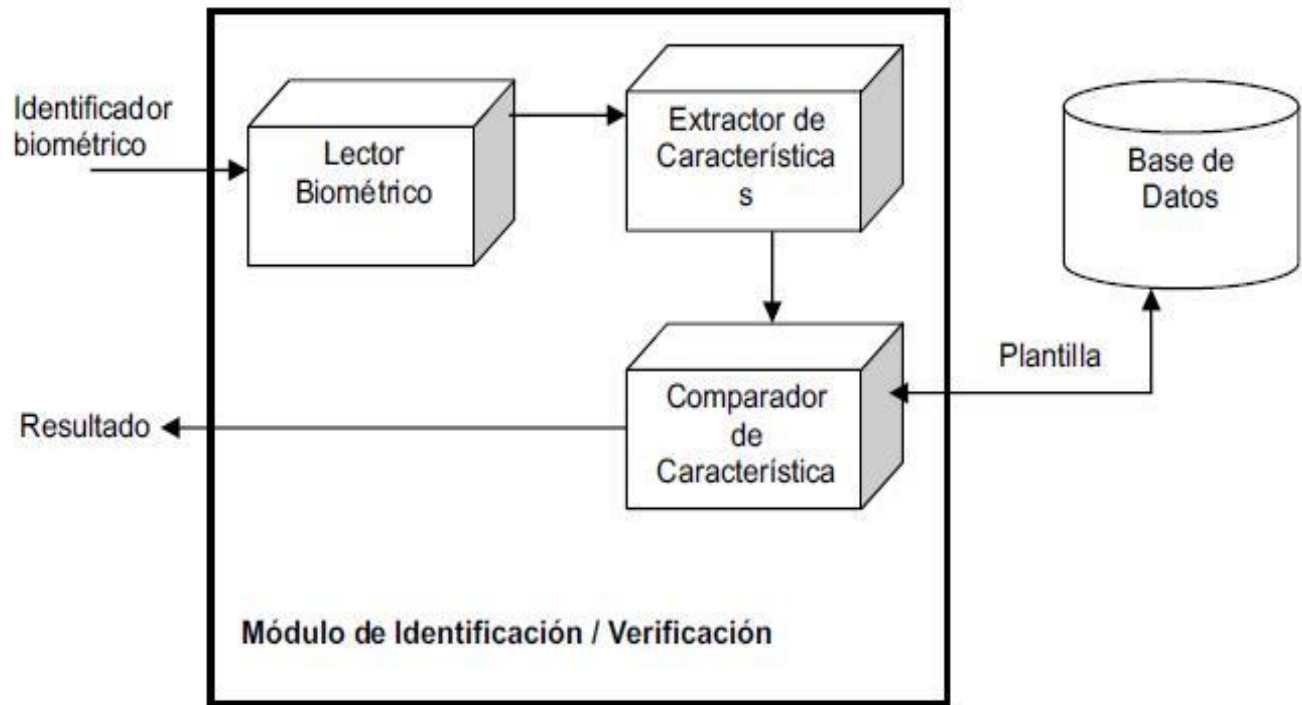


Ilustración 31: Módulo de Inscripción

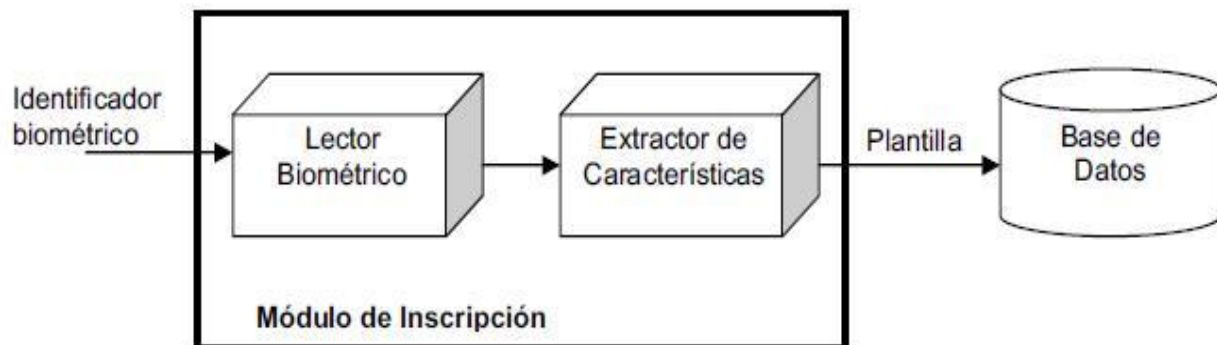
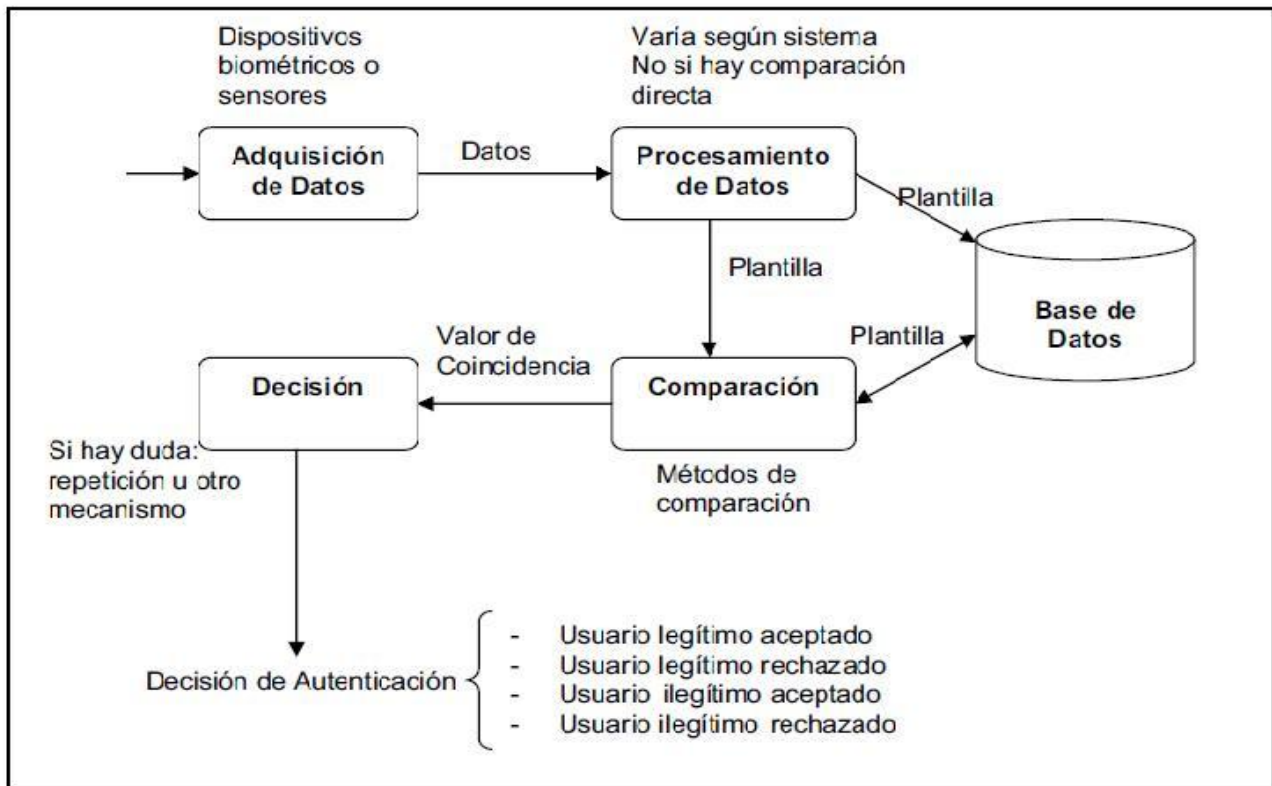


Ilustración 32: Proceso general de un Sistema Biométrico



❖ [GLOSARIO DE TÉRMINOS](#)

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

CISED: Centro de Identificación y Seguridad Digital

EER: Tasa de Igual Error

FAR: Tasa de Falsa Aceptación

FRR: Tasa de Falso Rechazo

HU: Historia de Usuario

OCR: Reconocimiento Óptico de Caracteres

PIN: Número de identificación personal

TIC: Tecnología de la Información y las Comunicaciones

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