



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1**

Herramienta dactiloscópica para la gestión pericial en un AFIS

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

***Autores: Alexis García Rojas
Adalberto Hernández Valdés***

***Tutores: MsC. Adrian A. Machado Cento
Ing. Ramón Santana Fernández***

La Habana. Junio de 2015

DECLARACIÓN JURADA DE AUTORÍA

Nosotros Adalberto Hernández Valdés, con carné de identidad 91052338426, y Alexis García Rojas, con carné de identidad 91100737780, declaramos que somos autores del resultado que exponemos en la presente memoria titulada **Herramienta dactiloscópica para la gestión pericial en un AFIS**, para optar por el título de Ingeniero en Ciencias Informáticas.

El presente trabajo fue desarrollado en el transcurso de los años 2014-2015.

Declaramos que todo lo anteriormente expuesto se ajusta a la verdad, y asumimos la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmamos la presente declaración jurada de autoría en La Habana a los 12 días del mes de junio del año 2015.

Autor: Adalberto Hernández Valdés

Autor: Alexis García Rojas

Tutor: MsC Adrian A Machado Cento

Tutor: Ing Ramón Santana Fernández

Resumen

Los sistemas biométricos son sistemas automatizados para el reconocimiento de personas que fundamentan sus decisiones en rasgos conductuales o físicos intrínsecos de éstas. De manera particular la identificación mediante huellas dactilares, requiere la comparación de una huella dactilar con las almacenadas en una base de datos. Como resultado de este proceso el sistema arroja tres tipos de respuestas: exitosa, no exitosa y dudosa. En el caso de respuesta dudosa se hace necesario realizar un proceso de cotejo visual de huellas dactilares por parte de especialistas periciales. Este proceso comienza con la clasificación de las huellas dactilares y culmina con la comparación de las características de ambas impresiones dactilares. Mediante esta técnica el especialista pericial es capaz de determinar si una huella dactilar pertenece a una persona.

Para facilitar el trabajo de los especialistas periciales se propone realizar una herramienta que permita ejecutar las acciones requeridas en el proceso de cotejo visual. Para el desarrollo de la herramienta de cotejo visual de huellas dactilares, producto de esta investigación, se realiza un estudio de las principales características de los sistemas de cotejo visual existentes y los procesos de comparación asociados a estos. Se determinan en la investigación funcionalidades adicionales que facilitan el proceso de cotejo y que no están en otras herramientas estudiadas. Se explica además la arquitectura y el diseño de la herramienta propuesta, así como la definición de las herramientas y tecnologías utilizadas.

Palabras claves: cotejo visual, especialista pericial, huella dactilar, sistemas biométricos.

Índice general

Introducción	1
Justificación de la investigación	4
Estructura del documento	4
Capítulo 1: Fundamentación teórica	5
Introducción	5
1.1 Reconocimiento biométrico	5
1.2 Definición de huella dactilar	5
1.2.1 Propiedades de las huellas dactilares	5
1.2.2 Características de las huellas dactilares	6
1.3 Clasificación de las huellas dactilares	8
1.3.1 Clasificación exclusiva o de patrón de huellas dactilares	8
1.4 Sistemas automáticos de identificación mediante huellas dactilares	9
1.5 Definición de cotejo	9
1.5.1 Definición de cotejo visual interactivo	9
1.6 Estudio del estado del arte	10
1.6.1 Soluciones existentes en el mundo	11
1.6.2 Soluciones existentes en Cuba	12
1.6.3 Análisis de los sistemas estudiados	13
1.7 Metodología de desarrollo de software	13
1.7.1 Metodologías pesadas	13
1.7.2 Metodologías ágiles	15
1.7.3 Selección de la metodología de desarrollo de software a utilizar	18
1.8 Herramientas y tecnologías a utilizar	19
Conclusiones parciales	25
Capítulo 2: Características del sistema	26
Introducción	26
2.1 Modelo de dominio	26
2.1.1 Glosario de conceptos del modelo de dominio	27
2.2 Metáfora	28
2.3 Captura de requisitos	28

2.3.1 Requisitos funcionales (RF).....	28
2.3.2 Requisitos no funcionales (RNF)	30
2.4 Historias de usuarios (HU)	30
2.5 Planificación	31
2.5.1 Plan de entregas.....	32
2.5.2 Plan de iteraciones	32
2.6 Diseño	33
2.6.1 Definición de la arquitectura	33
2.6.2 Patrones de diseño.....	35
2.6.3 Tarjetas CRC	37
2.6.4 Modelo de datos	38
Conclusiones parciales	39
Capítulo 3: Implementación y pruebas	40
Introducción.....	40
3.1 Implementación	40
3.1.1 Tareas de ingeniería.....	40
3.1.2 Estándares de codificación.....	41
3.1.3 Diagrama de componentes.....	42
3.1.4 Diagrama de despliegue.....	44
3.1.5 Interfaces de usuario	44
3.2 Pruebas	45
3.2.1 Pruebas unitarias.....	45
3.2.2 Pruebas de aceptación	46
3.2.3 Resultados de las pruebas	48
Conclusiones parciales	49
Conclusiones generales.....	50
Recomendaciones	51
Bibliografía referenciada	52
Bibliografía consultada.....	55
Glosario de términos.....	57
Anexos	59

Índice de figuras

Figura 1. 1 Valles y Crestas de una huella dactilar.....	6
Figura 1. 2 Tipos de minucias.....	7
Figura 1. 3 Núcleo y Delta.	7
Figura 1. 4 Clasificación de las huellas dactilares.	9
Figura 1. 5 Relación entre las fases y flujos de trabajo de RUP.....	14
Figura 2. 1 Modelo de dominio.	27
Figura 2. 2 Arquitectura de la herramienta.	34
Figura 2. 3 Patrón de creación Singleton.....	36
Figura 2. 4 Diagrama entidad-relación de la herramienta.....	39
Figura 3. 1 Diagrama de componentes de la herramienta.....	43
Figura 3. 2 Diagrama de despliegue de la herramienta.....	44
Figura 3. 3 Interfaz de cotejo visual de huellas dactilares.	45
Figura 3. 4 PU Algoritmo de extracción de puntos característicos.	46
Figura 3. 5 No conformidades detectadas durante las iteraciones.....	48
Figura A6. 1 Interfaz para autenticar usuario.....	74
Figura A6. 2 Interfaz para la gestión de usuarios.	74
Figura A6. 3 Interfaz de configuración.	75
Figura A7. 1 PU Algoritmo de extracción de puntos singulares.....	76
Figura A7. 2 PU Algoritmo para el conteo de crestas.	76
Figura A7. 3 PU Convertir a escala de gris.....	77

Índice de tablas

Tabla 2. 1 HU Seleccionar puntos característicos y puntos singulares de forma manual.	31
Tabla 2. 2 Plan de entregas.	32
Tabla 2. 3 Plan de iteraciones.	33
Tabla 2. 4 Tarjeta CRC de la clase CotejoLista.	38
Tabla 3. 1 Tareas de ingeniería de la segunda iteración.	41
Tabla 3. 2 CP Seleccionar puntos característicos y puntos singulares de forma manual.	47
Tabla 3. 3 CP Detectar puntos característicos y puntos singulares de forma automática.	48
Tabla A2. 1 HU Autenticar Usuario.	62
Tabla A2. 2 HU Gestionar Usuario.	62
Tabla A2. 3 HU Cargar imágenes de las huellas dactilares.	63
Tabla A2. 4 HU Cargar algoritmos biométricos.	63
Tabla A2. 5 HU Detectar puntos característicos y puntos singulares de forma automática.	64
Tabla A2. 6 HU Medir distancia entre puntos.	64
Tabla A2. 7 HU Contar puntos característicos.	65
Tabla A2. 8 HU Contar crestas papilares.	65
Tabla A2. 9 HU Cotejo automático de huellas.	66
Tabla A2. 10 HU Superponer imágenes de huellas dactilares.	66
Tabla A2. 11 HU Aplicar transformaciones a la imagen de la huella dactilar.	67
Tabla A2. 12 HU Gestionar comparaciones.	67
Tabla A3. 1 Tarjeta CRC de la clase Conf.	68
Tabla A3. 2 Tarjeta CRC de la clase MefAux.	68
Tabla A3. 3 Tarjeta CRC de la clase NegocioImpl.	68
Tabla A3. 4 Tarjeta CRC de la clase Minucia.	68
Tabla A3. 5 Tarjeta CRC de la clase MinuciaTipo.	68
Tabla A3. 6 Tarjeta CRC de la clase Controlador.	69
Tabla A3. 7 Tarjeta CRC de la clase PuntoLineas.	69
Tabla A3. 8 Tarjeta CRC de la clase dComparacion.	69
Tabla A3. 9 Tarjeta CRC de la clase Util.	69

Tabla A4. 1 Descripción de la entidad usuario.	70
Tabla A4. 2 Descripción de la entidad rol.	70
Tabla A4. 3 Descripción de la entidad permiso.	70
Tabla A4. 4 Descripción de la entidad rol_permiso.	70
Tabla A4. 5 Descripción de la entidad comparacion.....	71
Tabla A5. 1 Tareas de ingeniería de la primera iteración.	72
Tabla A5. 2 Tareas de ingeniería de la tercera iteración.	73
Tabla A8. 1 CP Autenticar usuario.....	78
Tabla A8. 2 CP Adicionar usuario.....	78
Tabla A8. 3 CP Listar usuario.	79
Tabla A8. 4 CP Eliminar usuario.....	79
Tabla A8. 5 CP Modificar usuario.	80
Tabla A8. 6 CP Cargar imagen de la huella dactilar a comparar.....	80
Tabla A8. 7 CP Cargar imágenes de las huellas dactilares candidatas.	81
Tabla A8. 8 CP Cargar algoritmos biométricos.....	81
Tabla A8. 9 CP Medir distancia entre puntos.	82
Tabla A8. 10 CP Contar puntos característicos.....	82
Tabla A8. 11 CP Contar crestas papilares.....	83
Tabla A8. 12 CP Cotejo automático de huellas.	83
Tabla A8. 13 CP Superponer imágenes de huellas dactilares.	84
Tabla A8. 14 CP Mover y rotar imagen.....	84
Tabla A8. 15 CP Modificar tamaño de la imagen.	85
Tabla A8. 16 CP Modificar brillo de la imagen.....	85
Tabla A8. 17 CP Modificar contraste de la imagen.....	86
Tabla A8. 18 CP Llevar la imagen a escala de grises.	86
Tabla A8. 19 CP Invertir imagen.....	87
Tabla A8. 20 CP Guardar comparación.....	87
Tabla A8. 21 CP Listar comparaciones.	88
Tabla A8. 22 CP Modificar comparación.	88
Tabla A8. 23 CP Eliminar comparación.....	89

Introducción

El acelerado desarrollo científico tecnológico alcanzado en la actualidad ha propiciado evolución y cambio en diferentes esferas de la ciencia, lo que impacta directamente en el desarrollo de la sociedad. Una de las esferas beneficiadas con este desarrollo es la biometría. Este término proviene del griego *bios* vida y *métron* medida, es la ciencia que se encarga del estudio de las características físicas o conductuales inherentes a un individuo, que son permanentes hasta su muerte y de gran singularidad e invariabilidad (1).

Entre los rasgos biométricos más utilizados para la identificación de personas se encuentran el iris, la geometría de la mano, la huella dactilar, la voz, la firma manuscrita y la manera de caminar. De ellos, las huellas dactilares (patrones gráficos de crestas presentes en los dedos humanos) se encuentran entre las características biométricas más fiables para la identificación de personas. De manera particular, el uso de la huella dactilar es la más antigua de las técnicas biométricas y el rasgo individual más utilizado en procesos de este tipo, por su facilidad de adquisición, de uso, fiabilidad y gran aceptación por parte de los usuarios (1). Sus orígenes se remontan a mediados del siglo XIX, cuando investigadores en criminología intentaron relacionar las características físicas de los individuos con tendencias criminales. No es hasta finales de los años 60 del siglo XX que con el desarrollo de las tecnologías de la computación esta técnica comienza su transición a la automatización, marcando el surgimiento de los sistemas automáticos de identificación mediante huellas dactilares (AFIS del inglés *Automatic Fingerprint Identification Systems*).

Un AFIS es un sistema informático utilizado para analizar imágenes digitalizadas de huellas dactilares individuales y compararlas con imágenes almacenadas en una base de datos en busca de una coincidencia exacta o parcial (2). El proceso automatizado de identificación de una persona comienza con la captura de la huella dactilar y culmina con la devolución de una respuesta exitosa, no exitosa o dudosa. La respuesta exitosa se obtiene en caso de tener certeza de haber encontrado una coincidencia, la no exitosa en caso de no encontrar ninguna coincidencia y la dudosa en caso que el sistema no sea capaz de tomar una decisión en cuanto a si coinciden o no dos huellas, en este último el sistema devolverá una lista de posibles candidatos.

En caso que la respuesta del AFIS sea dudosa se hará necesario realizar un proceso de cotejo visual de huellas dactilares por parte de especialistas periciales, que permita comparar la huella recibida como entrada con cada una de las arrojadas por el sistema en la lista de posibles candidatos. Para realizar este

proceso es necesario ir de lo general a lo particular comenzando con la clasificación de la huella dactilar y terminando con la comparación de las características de ambas huellas. La clasificación corresponde a un análisis de los patrones globales de la huella dactilar que permite asignarla a un conjunto predeterminado o clase. Por otro lado, la comparación se lleva a cabo a partir de los vectores de características que resultan de la representación geométrica de los puntos característicos y singulares presentes en la huella dactilar.

El proceso de cotejo visual de huellas dactilares, necesario para la identificación de una persona puede tornarse engorroso y complicado. Esto se debe a la necesidad de marcar en las imágenes de las huellas dactilares a comparar los puntos característicos y singulares que estén presentes en estas así como llevar a cabo mediciones que sin las herramientas pertinentes no se lograría con exactitud y precisión. Para llevar a cabo esta tarea en la actualidad se utilizan herramientas que en su mayoría son privativas y costosas. En nuestro país se han desarrollado algunas de ellas, las cuales no cuentan con funcionalidades como la detección automática de puntos singulares, la medición de distancias entre puntos, ni el conteo de crestas papilares, que son funciones necesarias en el proceso de cotejo visual de huellas dactilares. Esto hace que el tiempo y el esfuerzo requerido para la realización de la comparación sean mayores.

Los factores anteriormente planteados dificultan la identificación de un individuo constituyendo esto una **problemática** a partir de la cual se plantea como **problema de investigación**: ¿Cómo facilitar el proceso de cotejo visual de huellas dactilares?

Ante el problema definido anteriormente, se determina como **objeto de estudio** el proceso de comparación dactiloscópica de huellas dactilares.

Objetivo general: desarrollar una herramienta informática para el análisis visual de imágenes de huellas dactilares que facilite el proceso de comparación dactiloscópica en los sistemas automatizados de reconocimiento de personas mediante huellas dactilares.

Se definen como **objetivos específicos**:

- ✓ Analizar los referentes teórico-metodológicos que preceden la realización de la presente investigación para un mayor entendimiento del área de conocimientos a abordar.
- ✓ Seleccionar las herramientas que serán utilizadas en el desarrollo de la aplicación que se propone.
- ✓ Realizar análisis y diseño de la solución para la guía del proceso de implementación.

- ✓ Implementar la herramienta informática para el cotejo visual de huellas dactilares.
- ✓ Realizar pruebas de software a la herramienta desarrollada para la validación de su correcto funcionamiento.

Para dar cumplimiento a dichos objetivos se plantean como **tareas de la investigación**:

- 1) Fundamentación de los referentes teórico-metodológicos del proceso de identificación de personas mediante huellas dactilares por especialistas periciales.
- 2) Realización de entrevistas a especialistas periciales para identificar posibles funcionalidades.
- 3) Definición de las estrategias necesarias para el proceso de cotejo visual de huellas dactilares.
- 4) Definición de las tecnologías, metodologías y herramientas a utilizar para el desarrollo de la aplicación.
- 5) Caracterización de la tecnología *Windows Presentation Foundation* para el desarrollo de la herramienta.
- 6) Caracterización de la tecnología *Managed Extensibility Framework* para la gestión de bibliotecas de clases dinámicas.
- 7) Confección del modelo de dominio de la herramienta para identificar los principales conceptos involucrados en la propuesta de solución del problema planteado.
- 8) Especificación de los requisitos del software.
- 9) Definición de la arquitectura de la herramienta.
- 10) Realización de pruebas de software a la herramienta.

Los **Métodos Científicos** empleados son:

Métodos Teóricos:

- ✓ Histórico-Lógico para el estudio de trabajos anteriores y afines con el problema planteado.
- ✓ Analítico-Sintético: para desglosar el problema principal en pequeños problemas de investigación, facilitando profundizar en el conocimiento existente sobre el tema sintetizándolo en la solución final.
- ✓ Modelación: para el desarrollo de los diagramas correspondientes al diseño del software.

Métodos Empíricos

- ✓ Entrevista: para obtener la información necesaria relacionada con el proceso de cotejo visual de huellas dactilares (ver preguntas de la entrevista en el Anexo 1).

Justificación de la investigación:

Un sistema automatizado de identificación mediante huellas dactilares tiene 3 tipos de respuestas, exitosa, no exitosa y dudosa. En el caso de que la respuesta sea dudosa se hace necesario el cotejo visual, por parte de los especialistas periciales, de las imágenes candidatas y de prueba. Las principales herramientas existentes para realizar este proceso son privativas y de compleja adquisición debido a sus costos. Las desarrolladas en el país no satisfacen a cabalidad las necesidades de los especialistas periciales pues carecen de funcionalidades vitales como la detección automática de puntos singulares, la medición de distancias entre puntos, el conteo de crestas papilares, entre otras. A pesar de ser desarrolladas en Cuba no se tiene acceso al código fuente para ser modificado. La presente investigación ofrece como aporte práctico una herramienta informática para realizar el proceso de cotejo visual de huellas dactilares, con un conjunto de funcionalidades que faciliten el trabajo de los especialistas periciales.

Estructura del documento:

El presente trabajo de diploma se divide en tres capítulos estructurados de la siguiente forma:

Capítulo I: Fundamentación teórica. En este capítulo se analizan los principales conceptos relacionados con el problema de la investigación, se realiza el estudio del estado del arte a nivel internacional y nacional, sobre los diferentes sistemas que utilizan la comparación visual de huellas dactilares, además se definen las tecnologías, metodología y herramientas a utilizar durante el desarrollo de la aplicación.

Capítulo II: Características de la solución propuesta. En este capítulo se realiza una descripción general de la solución propuesta y su funcionamiento. Se detallan los principales aspectos del diseño, se definen los requisitos funcionales y no funcionales, así como la arquitectura de la herramienta y se especifican los patrones del diseño que se aplican.

Capítulo III: Implementación y Pruebas. En este capítulo se abordan aspectos relacionados con la implementación de la herramienta. Se realizan las pruebas unitarias dirigidas al código de la misma y se construyen los casos de prueba de aceptación en base a los requisitos definidos para el desarrollo de la herramienta.

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se exponen los conceptos fundamentales que constituyen la base teórica de los sistemas de cotejo visual de huellas dactilares. Se realiza un estudio del estado del arte a nivel internacional y nacional sobre los diferentes sistemas que utilizan la comparación visual de huellas dactilares. Se analizan y escogen las metodologías, tecnologías y herramientas utilizadas durante el ciclo de desarrollo de la solución que se propone.

1.1 Reconocimiento biométrico

La biometría es la ciencia que se encarga del reconocimiento de personas basado en sus características físicas o de comportamiento. El reconocimiento biométrico a partir de las características físicas del individuo se efectúa al realizar el análisis de algún rasgo físico del cuerpo humano en el momento de identificarlo, ya sea la huella dactilar, el rostro, el iris u otras características. Mientras que las técnicas de comportamiento se definen analizando en el proceso de identificación los rasgos derivados de una acción realizada por la persona, tales como la voz, la firma manuscrita, la escritura en el teclado, entre otros (3).

1.2 Definición de huella dactilar

Una huella dactilar es la impresión visible o moldeada que produce el contacto de las crestas papilares de un dedo de la mano sobre una superficie, como característica individual distingue a todos los seres humanos (4).

1.2.1 Propiedades de las huellas dactilares

El reconocimiento biométrico haciendo uso de las huellas dactilares se basa en tres principios fundamentales (5):

- ✓ **Perennidad:** desde que se forman en el sexto mes de la vida intrauterina, permanecen indefectiblemente invariables en número, situación, forma y dirección hasta que la descomposición

del cadáver destruye la piel.

- ✓ **Inmutabilidad:** las crestas papilares no pueden modificarse fisiológicamente, de existir un traumatismo poco profundo, se regeneran, y si es profundo, no reaparecen con forma distinta a la que tenían, sino que la parte afectada por el traumatismo resulta en una cicatriz.
- ✓ **Diversidad Infinita:** son únicas e irrepetibles, por lo que identifican unívocamente a cada individuo.

1.2.2 Características de las huellas dactilares

- ✓ **Valles y crestas:** las huellas dactilares están constituidas por rugosidades que forman salientes y depresiones. Las salientes se denominan crestas papilares y las depresiones surcos inter-papilares o valles. Una cresta está definida como un segmento de curva y un valle es la región entre dos crestas adyacentes (6). Las crestas, en una imagen de una huella dactilar usualmente aparecen como una serie de líneas oscuras que representan los relieves, mientras que los valles entre estas crestas aparecen como espacio en blanco y están en bajo relieve (ver figura 1.1).

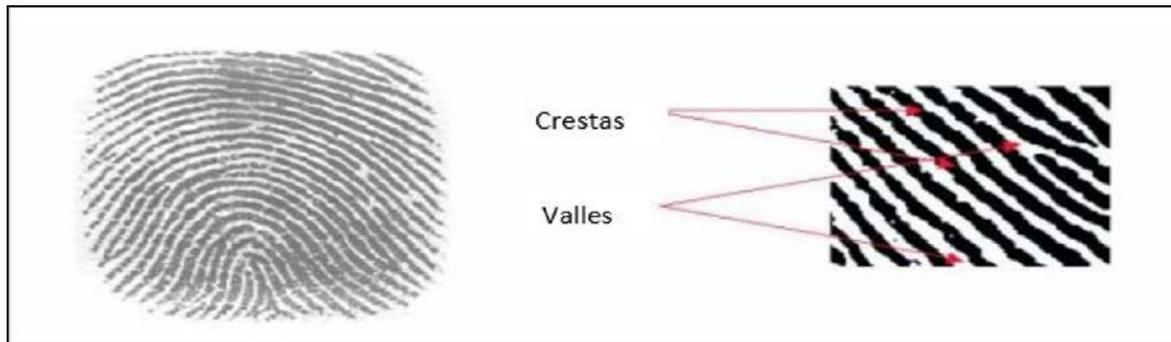


Figura 1. 1 Valles y Crestas de una huella dactilar (6).

- ✓ **Minucia:** detalle pequeño que se refiere a las distintas maneras en las que las crestas de la huella dactilar pueden ser discontinuadas. Una minucia se identifica básicamente por su tipo, posición $[x, y]$ y el ángulo de la tangente de la cresta donde se encuentra ubicada con respecto al eje horizontal, el cual se define como $\theta \in [0, \pi]$ (5). Las minucias se pueden clasificar en diferentes tipos (ver figura 1.2), las dos clases que aportan mayor información de una huella son las terminaciones y las bifurcaciones puesto que el resto de las clasificaciones puede verse como combinaciones de éstas.

Durante la investigación se utiliza el término puntos característicos para hacer referencia a las minucias.

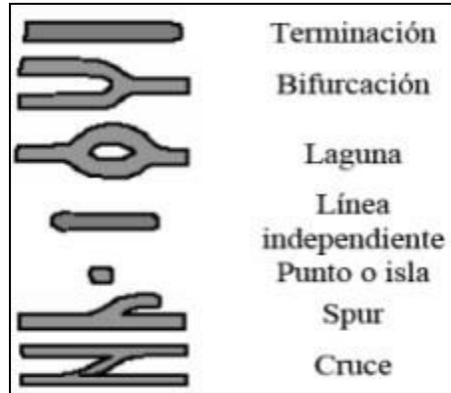


Figura 1. 2 Tipos de minucias (5).

- ✓ **Núcleo:** punto de máxima curvatura de la cresta más interna. Centro de la huella dactilar, donde se genera el inicio del lazo en la huella o el inicio del círculo (ver figura 1.3) (7).
- ✓ **Delta:** centro de un conjunto de crestas inferiores al núcleo, que dibujan varios triángulos concéntricos (ver figura 1.3) (7). Durante la investigación se utiliza el término puntos singulares para hacer referencia al núcleo y al delta.

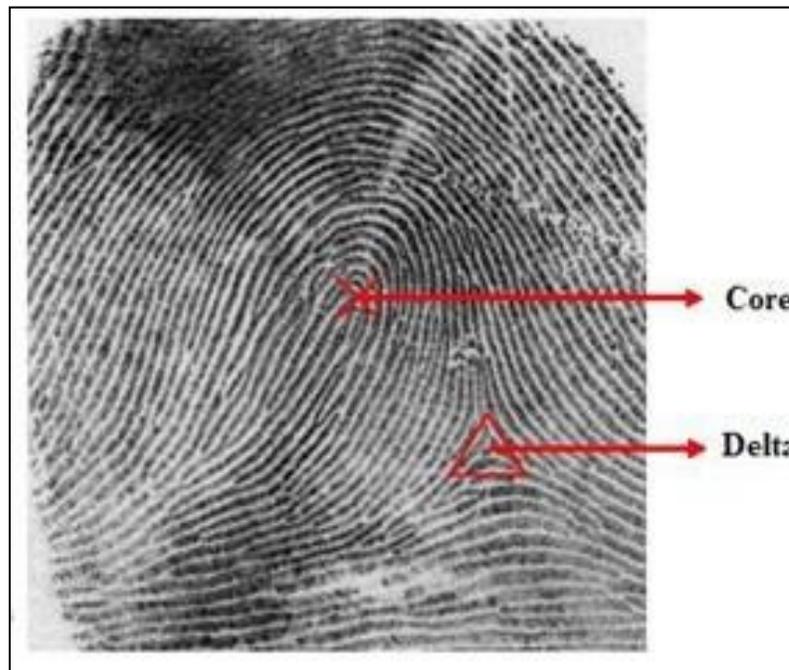


Figura 1. 3 Núcleo y Delta (5).

1.3 Clasificación de las huellas dactilares

La clasificación de huellas dactilares consiste en asignar de modo consistente y fiable cada huella a una clase, de manera tal que en la identificación de una huella desconocida, solo se necesite que ésta sea comparada con el subconjunto de huellas dactilares en la base de datos que pertenece a su misma clase (5).

1.3.1 Clasificación exclusiva o de patrón de huellas dactilares

De acuerdo a la forma que describen los dibujos papilares de una huella dactilar, se pueden distinguir cinco grupos o clases distintas de configuraciones dérmicas según la denominada Clasificación de Henry (5):

- a) **Arco:** este dactilograma carece de puntos delta y de núcleo. Las crestas fluyen de un lado al otro de la imagen, curvándose ligeramente en el centro de la misma (ver figura 1.4).
- b) **Arco Tendido:** presenta un punto delta y un núcleo. Semeja a un arco, pero con la diferencia de que es una línea que rápidamente nace y se pierde en un paso de ángulo (ver figura 1.4).
- c) **Lazo Izquierdo:** las crestas entran por el lado izquierdo de la imagen, fluyen hacia arriba rodeando el núcleo y luego dan vuelta sobre sí mismas para regresar al mismo punto de partida. Este dactilograma cuenta con un núcleo y un punto delta ubicado del lado derecho del observador (ver figura 1.4).
- d) **Lazo Derecho:** las crestas entran por el lado derecho de la imagen, fluyen hacia arriba rodeando el núcleo y luego dan vuelta sobre sí mismas y regresan al mismo punto de partida. Existe un núcleo y un punto delta que aparece del lado izquierdo del observador (ver figura 1.4).
- e) **Espiral:** la característica más importante de este tipo de huella dactilar es que cuenta con dos puntos delta entre los cuales fluyen las crestas curvándose alrededor de un núcleo que puede adoptar formas circulares, elípticas y espirales (ver figura 1.4).



Figura 1. 4 Clasificación de las huellas dactilares (5).

1.4 Sistemas automáticos de identificación mediante huellas dactilares

Los AFIS son los sistemas biométricos especializados en el reconocimiento de personas mediante las huellas dactilares y sus inicios datan de la década del sesenta del pasado siglo, aunque la utilización de las huellas dactilares en ámbitos forenses se realizaba ya desde finales del siglo XIX por especialistas que examinaban dos pares de huellas de forma visual (8). El proceso de identificación llevado a cabo por un AFIS está compuesto por un conjunto de etapas. Primero se adquiere la huella del sujeto que será identificado mediante un sensor especializado para ello. El paso siguiente es pre-procesar la imagen con el objetivo de aumentar su calidad. Llegado a este punto se aplican algoritmos para la extracción de las características que serán comparadas (entre las características más utilizadas se encuentran los puntos característicos y los puntos singulares). El resultado de la comparación entre las características obtenidas con las características de otra huella determinará, bajo un umbral, si las dos muestras pertenecen al mismo individuo (9).

1.5 Definición de cotejo

Acción y efecto de confrontar algo con otra u otras cosas; compararlas teniéndolas a la vista. Esta acción consiste en analizar las características de dos o más objetos o situaciones para establecer similitudes, diferencias y desarrollar algún tipo de valoración (10).

1.5.1 Definición de cotejo visual interactivo

Cotejo visual interactivo: Se ejecuta el cotejo dactilar para efectuar el peritaje dactiloscópico, utilizando el monitor de una computadora. La pantalla se divide en dos paneles o partes, a la izquierda se sitúa la huella o impresión que se investiga y en el panel de la derecha los sucesivos dactilogramas extraídos desde

una base de datos de imágenes. El vidrio de aumento (lentes) es completamente simulado por numerosos niveles de ampliaciones que se les aplican a los dactilogramas. Se añaden herramientas auxiliares de superposición gráfica, rotaciones a voluntad, marcado de puntos, variaciones de contrastes visuales mediante el empleo de colores y otros recursos. La productividad crece y los medios utilizados en el cotejo son menos agresivos para el perito (11).

1.6 Estudio del estado del arte

El uso de las huellas dactilares tuvo sus orígenes en el siglo XIX. Sir William Herschel en 1856 fue el primero en utilizar la huella dactilar como método de identificación en documentos para personas analfabetas. En octubre de 1880 Henry Faulds, médico escocés, publicó un artículo en la revista *Nature* sobre cómo identificar criminales a partir de sus huellas dactilares llamado “*On the Skin-Furrows of the Hand*” (1). Otro hito importante para el campo lo constituyen los estudios realizados por el antropólogo británico Sir Francis Galton en 1892, cuando publicó su trabajo definitivo, *Fingerprint*; en dicha investigación se comprobó científicamente la unicidad de las huellas dactilares (12). En los finales del siglo XIX destaca también el estudio del militar británico Sir Edward Henry, quien desarrolló un método de identificación basado en la huella dactilar del pulgar y cuyos fundamentos fueron publicados en 1900 en su trabajo *Classification and Use of Fingerprints*. Dicho método se convirtió en un estándar para la mayoría de los sistemas de identificación de todo el mundo, siendo asumido por el FBI (*Federal Bureau of Investigation*) que se encargó de sustentarlo. En 1969, hubo un impulso por parte del FBI para desarrollar un sistema que automatizara sus procesos manuales de identificación por huellas dactilares, los cuales requerían muchas horas hombre (13).

Las tecnologías utilizadas para el reconocimiento biométrico continuaron mejorando y para el año 1981, cinco sistemas automatizados de identificación por huella dactilar fueron desplegados (14). El funcionamiento de un AFIS posibilita guardar las imágenes de las huellas dactilares en una base de datos informática, acceso rápido y automático a éstas, y determinación automática de sus puntos característicos, proceso que en forma manual demanda mucho tiempo y recursos. Con el surgimiento de los AFIS y el continuo desarrollo que estos han tenido, se hace necesario lograr un mayor nivel de fiabilidad en sus respuestas, con el fin de lograr esto se realiza un proceso de cotejo visual de huellas dactilares por parte de especialistas periciales.

1.6.1 Soluciones existentes en el mundo

En el mundo, el número de participantes en la industria biométrica crece gradualmente, llevando a que el mercado sea cada vez más complejo con el desarrollo de tecnologías de alta calidad, a la medida de los intereses de los usuarios finales. Entre los sistemas de mayor prestigio y con mejores resultados en el mercado internacional que hacen uso de un módulo de cotejo visual para lograr una mayor fiabilidad en los procesos de identificación se encuentran ImaQuest y CSIpix.

ImaQuest

ImaQuest es un sistema para la mejora y comparación visual de huellas dactilares desarrollado por la compañía SPEX Forensics (15). Es una *Suite* de software que cuenta con poderosas herramientas para la captura, el seguimiento de mejoras, almacenamiento, extracción de minucias y comparación de huellas dactilares.

Este sistema permite mejorar las imágenes de las huellas dactilares utilizando técnicas como: la transformada rápida de Fourier, sustracción del fondo de la imagen, ecualización, nitidez, brillo, contraste, histograma, inversión, entre otras funcionalidades. Además de permitir la extracción automática y manual de minucias y la superposición de impresiones, todas estas funciones hacen que el trabajo de los especialistas sea más fácil.

CSIpix

CSIpix es un sistema para el tratamiento y comparación visual de huellas dactilares desarrollado por iSYS (*Intelligent System Solutions Corporation*) que es una empresa privada de desarrollo de software para el análisis automatizado de imágenes. iSYS se inició en el campo del análisis forense con el CSIpix Comparator proporcionando una forma innovadora y asequible para los profesionales forenses de poder aumentar la eficiencia en la transformación y la comparación de las imágenes de huellas dactilares (16).

Este sistema cuenta con varias funcionalidades para el mejoramiento de la imagen entre las que se encuentran: conversión a escala de grises de la imagen, ecualización, nitidez, brillo, contraste, histograma, inversión entre otras. Permite la rotación automática de las huellas latentes para alinearla con la impresión de muestra. Además de permitir la extracción manual y automática de minucias y la edición de las mismas.

1.6.2 Soluciones existentes en Cuba

Cuba ha incursionado también en el área de la identificación y en específico en el desarrollo de sistemas que hagan uso de un módulo de cotejo visual de imágenes de huellas dactilares para lograr mayor fiabilidad en el proceso de identificación de personas. Entre las instituciones que han obtenido resultados en estas actividades está DATYS Soluciones Tecnológicas que cuenta con una división destinada al desarrollo de un sistema biométrico llamado BIOMESYS SUITE y por otro lado especialistas del Laboratorio Provincial de Criminalística e investigadores del Centro de Bioplasmas de la provincia de Ciego de Ávila que en 2008 desarrollaron el Sistema Automatizado de Identificación de Huellas e Impresiones Palmares (SAIDIP).

BIOMESYS AFIS

BIOMESYS AFIS es el sistema cubano de identificación dactiloscópica desarrollado a partir de 2006 como parte de BIOMESYS SUITE, que ha ahorrado al país más de 10 millones de dólares y ha sido un paso de avance en su soberanía tecnológica (17).

Este sistema cuenta con el AFIS.CriminalSearchAdmin que es el módulo administrador de búsquedas y de cotejo visual de impresiones dactilares y huellas del sistema AFIS de la suite BIOMESYS. Dicho módulo cuenta con varias funcionalidades como son: cargar imagen, seleccionar rectángulo, seleccionar polígono, rotar, negativo, normalizar niveles, normalizar imagen, invertir imágenes, editar minucias, extraer minucias, eliminar minucias, aumentar tamaño de minucias, disminuir tamaño de minucias, cortar minucias dentro, cortar minucias fuera, aumentar tamaño de imagen, disminuir tamaño de imagen, brillo, contraste, superponer imágenes.

SAIDIP

El SAIDIP, desarrollado en 2008 en la provincia de Ciego de Ávila por un grupo de especialistas en dermatoscopia y reconocimiento de patrones. Es el primer sistema cubano de identificación de huellas e impresiones palmares (18).

Este sistema cuenta con un módulo de cotejo visual que logra que dicho sistema tenga una mayor fiabilidad en sus respuestas. Dicho módulo cuenta con varias funcionalidades como son: rotar, normalizar niveles, normalizar imagen, invertir imágenes, editar minucias, extraer minucias, eliminar minucias, aumentar tamaño de imagen, disminuir tamaño de imagen, brillo, contraste, superponer imágenes.

1.6.3 Análisis de los sistemas estudiados

Después de analizar algunas de las herramientas de cotejo visual existentes, comparar sus características y funcionalidades, se considera que:

- ✓ Los sistemas internacionales tienen como principal inconveniente que son software privativo, por lo que además del costo de adquisición que implican, tienen ciertas limitaciones para su uso, como es el caso de tener inaccesible el código fuente para hacerle modificaciones o redistribuirlo, por lo que en el caso particular de Cuba sería más factible utilizar un producto propio que tenga la calidad y precisión adecuada, pero sobre todo que consuma la menor cantidad posible de recursos financieros.
- ✓ Los sistemas nacionales a pesar de contar con un amplio grupo de funcionalidades necesarias para realizar el proceso de cotejo visual, carecen de otras que se tornan necesarias para los especialistas periciales a la hora de realizar una comparación visual, como son: la detección automática o manual de puntos singulares, así como la medición de distancia entre puntos, tampoco permiten la detección de las crestas papilares de las huellas ni el conteo de estas.

1.7 Metodología de desarrollo de software

Una metodología de desarrollo de software es una filosofía o marco de trabajo para estructurar, planificar y controlar el proceso de desarrollo de software. Especifica un conjunto de pasos o procedimientos que definen quién debe hacer qué, cuándo y cómo debe hacerlo. Su objetivo principal es guiar a los desarrolladores de software en la obtención de un producto de calidad que cumpla con los requerimientos establecidos por el cliente ajustándose a los recursos apropiados y a un costo razonable (19).

1.7.1 Metodologías pesadas

Las metodologías pesadas están orientadas al control de procesos. Se caracterizan por establecer estrictamente las tareas, la definición de roles, los artefactos que se deben obtener, así como las herramientas y notaciones a utilizar. Además, implican una detallada planificación y exhaustiva documentación. Son muy efectivas para proyectos de desarrollo de software complejos, aunque su aplicación en entornos volátiles no es recomendada (19).

Rational Unified Process (RUP)

RUP es una metodología para el desarrollo de software que integra los elementos esenciales del ciclo de vida del software, permitiendo ser adaptado a las características específicas de cada organización (20), aunque está especialmente diseñado para llevar a cabo grandes y complejos proyectos.

En el ciclo de vida de RUP las distintas actividades se han organizado en nueve flujos de trabajo bien definidos. De ellos seis son llamados flujos de ingeniería y los restantes tres, flujos de apoyo. Los flujos de trabajo son: Modelación de negocios, Requerimientos, Análisis y Diseño, Implementación, Prueba, Desarrollo, Administración de configuración y cambios, Administración del proyecto y Ambiente (ver figura 1.5).

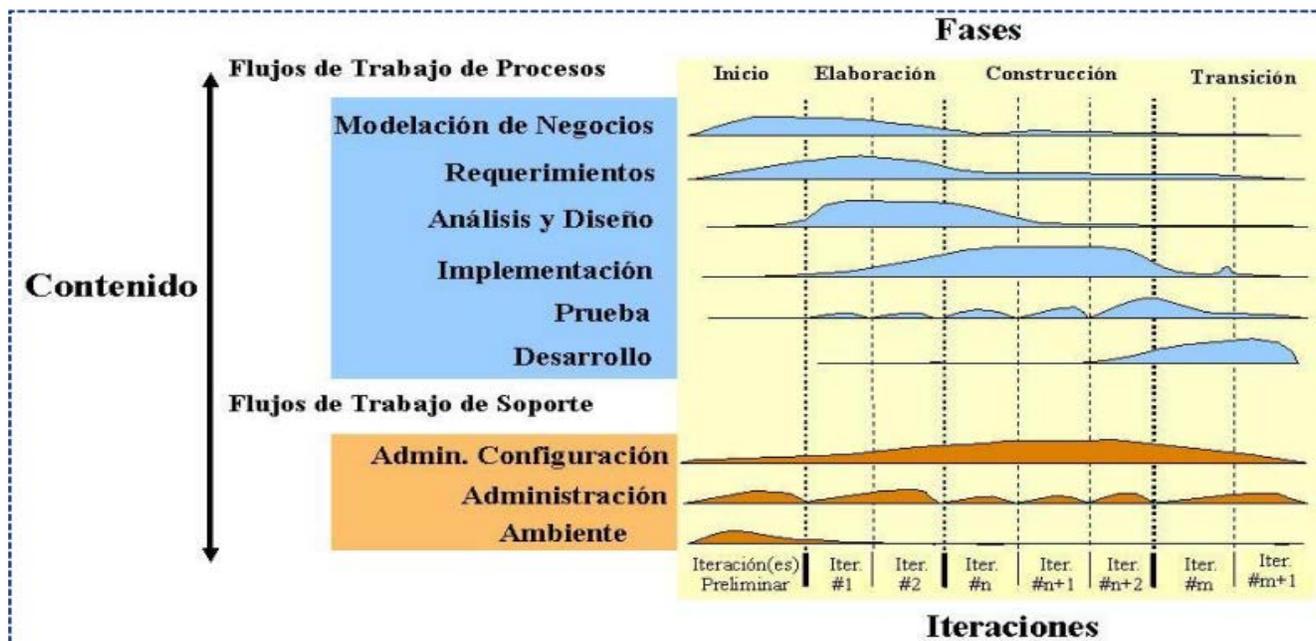


Figura 1. 5 Relación entre las fases y flujos de trabajo de RUP (20).

RUP está dividido en cuatro fases: Inicio, Elaboración, Construcción y Transición. En la primera de ellas el objetivo esencial es delimitar el alcance del proyecto. También se explora el problema, lo cual permite describir el negocio y determinar los casos de uso críticos en él.

La fase de Elaboración está centrada a la definición de la arquitectura (21) mediante la comprensión de los requisitos funcionales y no funcionales identificados en la fase previa.

Obtener una versión del producto lista para ser usada y documentada es la finalidad principal de la fase de Construcción (todos los requisitos son implementados, integrados y probados), siendo esta fase la más prolongada de todas y en la que se pone a consideración de los usuarios el producto final.

La fase de Transición, como indica su nombre, está basada en la transición de la versión final del sistema, del entorno de desarrollo a las manos de los usuarios. Para ello es necesaria la corrección de los últimos errores, la completitud de la documentación y el entrenamiento del usuario en su interacción con el software (22).

El ciclo de vida de RUP tiene tres características fundamentales:

- ✓ **Dirigido por casos de uso:** Los casos de uso representan las funcionalidades del software que el usuario final necesita, los cuales son recogidos en el modelado del negocio y posteriormente transformados en requisitos funcionales. Constituyen la línea principal de todas las actividades que se realizarán a partir de ese momento, pues todos los artefactos resultantes de los flujos de trabajos representan su realización (23).
- ✓ **Centrado en la arquitectura:** Todos los implicados en el proceso de desarrollo de software deben tener una visión clara de la arquitectura. La arquitectura es el conjunto de sistemas, reglas, estándares, convenciones y procesos que entrelazados representan la base para la comprensión y construcción del sistema.
- ✓ **Iterativo e incremental:** Cada fase del ciclo de vida de RUP se realiza mediante iteraciones. Las iteraciones comprenden actividades de todos los flujos de trabajo que tienen como resultado un incremento o crecimiento en el producto software y un refinamiento de la arquitectura (23).

1.7.2 Metodologías ágiles

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que harán la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega. Según el manifiesto ágil esta metodología valora (24):

- ✓ Al individuo y las interacciones del equipo de desarrollo, sobre el proceso y las herramientas.
- ✓ Desarrollar software que funcione, más que conseguir una buena documentación.
- ✓ La colaboración con el cliente, más que la negociación de un contrato.

- ✓ Responder a los cambios, más que seguir estrictamente un plan.

Extreme Programming (XP)

Programación Extrema es una metodología de desarrollo de software cuyos objetivos fundamentales son alcanzar la satisfacción del cliente, debiendo responder muy rápido a sus necesidades, incluso cuando los cambios sean al final del ciclo de programación, y potenciar al máximo el trabajo en grupo teniendo como premisa que tanto los jefes de proyecto, los clientes y los desarrolladores son parte del equipo y están involucrados en el desarrollo del software. El ciclo de vida propuesto por la metodología se divide en las siguientes fases:

- ✓ **Fase de exploración:** los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la familiaridad que tengan los programadores con la tecnología a utilizar.
- ✓ **Fase de planificación:** el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, además de las entregas relacionadas con éstas. El resultado de esta fase es un Plan de Entregas.
- ✓ **Fase de iteraciones:** esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entregas está compuesto por las diferentes iteraciones. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto, escogiendo las historias de usuario que fuercen la creación de esta arquitectura, sin embargo, no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración. Al final de la última iteración el producto estará listo para entrar en producción.
- ✓ **Fase de puesta en producción:** en esta fase se entregan módulos funcionales y sin errores, y según los intereses del cliente el sistema puede ponerse en producción o no. No se realizan desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.
- ✓ **Fase de mantenimiento:** mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para esto se requiere de tareas de soporte para el cliente. La velocidad de desarrollo puede disminuir después

de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

- ✓ **Fase de muerte del proyecto:** el cliente no tiene más historias para ser incluidas en el sistema, haciendo necesario que se satisfagan sus necesidades en otros aspectos tales como rendimiento y confiabilidad del mismo. Se genera la documentación final y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Durante el desarrollo de estas fases XP propone el uso de las siguientes prácticas:

- ✓ El juego de la planificación: el alcance de la siguiente versión está definido por las consideraciones del negocio (prioridad de los módulos, fechas de entrega) y estimaciones técnicas (estimaciones de funciones, consecuencias). El objetivo del juego es maximizar el valor del software producido.
- ✓ Versiones pequeñas: un sistema simple se pone rápidamente en producción. Periódicamente se producen nuevas versiones agregando en cada iteración aquellas funciones consideradas valiosas para el cliente.
- ✓ Metáfora del sistema: cada proyecto es guiado por una historia simple que explica el funcionamiento del sistema en general, reemplaza a la arquitectura y debe estar en lenguaje común, entendible para todos, cliente y desarrolladores.
- ✓ Diseño simple: el sistema se diseña con la máxima simplicidad posible. Se plasma el diseño en las tarjetas CRC (Clase - Responsabilidad - Colaboración), con lo que las clases descubiertas durante el análisis pueden ser filtradas para determinar las que son realmente necesarias para el sistema.
- ✓ Pruebas continuas: los casos de prueba se escriben antes que el código. Los desarrolladores escriben pruebas unitarias y los clientes especifican pruebas funcionales.
- ✓ Refactorización: es posible reestructurar el sistema sin cambiar su comportamiento, por ejemplo eliminando código duplicado, simplificando funciones.
- ✓ Programación por parejas: el código es escrito por dos personas trabajando en la misma computadora.
- ✓ Posesión colectiva del código: nadie es dueño de un módulo, cualquier programador puede cambiar cualquier parte del sistema en cualquier momento, por ello siempre se deben utilizar estándares de codificación.

- ✓ Integración continua: los cambios se integran en el código base varias veces por día. Todos los casos de prueba se deben pasar antes y después de la integración, se dispone de una máquina para la integración y se realizan pruebas funcionales donde participa el cliente.
- ✓ Cliente en el sitio: el equipo de desarrollo tiene acceso todo el tiempo al cliente, el cual está disponible para responder preguntas y fijar prioridades.
- ✓ Estándares de codificación: todo el código debe estar escrito de acuerdo a un estándar de codificación (24).

Scrum

Scrum es una metodología de desarrollo de software iterativa e incremental. Define un marco para la gestión de proyectos orientado a qué construir y en qué orden hacerlo; debido a ello se utiliza en otras prácticas de ingeniería de software tales como RUP o XP. Es muy sencilla y responde a los principios de inspección constante y continua, adaptación del producto a las necesidades del cliente en tiempo real e innovación (25).

Tiene como meta fomentar la efectividad y productividad de los equipos. Entre sus principales características destacan los ciclos de desarrollo llamados *sprints* con una duración de 30 días, cuyo resultado es un incremento funcional del producto. Establece reuniones durante todo el proyecto, de ellas las más importantes son las diarias, en las cuales los miembros del equipo disponen de 15 minutos para chequear el trabajo realizado hasta la fecha y las previsiones para el día siguiente (26).

1.7.3 Selección de la metodología de desarrollo de software a utilizar

Partiendo del análisis de las características particulares de cada una de las metodologías de desarrollo de software y en concordancia con la solución que se propone en esta investigación se elige como metodología XP debido a:

- ✓ RUP es una de las metodologías más robustas a tener en cuenta para el desarrollo de software, propone un exhaustivo levantamiento de requerimientos con la premisa de detectar defectos en las fases iniciales, además requiere la generación de gran cantidad de artefactos factores que indican que la metodología está concebida para grandes grupos de desarrollo, factores que no se ajustan a las particularidades de la solución que se propone ya que el equipo de desarrollo es pequeño y no se

cuenta con una definición detallada y exacta de todos los requisitos que debe cumplir, lo que implica una alta probabilidad de ocurrencia de cambios de forma gradual, a medida que se vayan obteniendo las pequeñas versiones. Por lo que es más conveniente usar una de las metodologías ágiles, diseñadas para pequeños equipos de desarrollo y preparadas para enfrentar cambios durante la creación del software.

- ✓ Si bien Scrum es una metodología ágil especialmente indicada para proyectos con un rápido cambio de requisitos, su poca documentación puede provocar efectos negativos en el modo de confeccionar el producto final.
- ✓ XP está definida para equipos pequeños de desarrollo, tiene prioridad de satisfacer al cliente mediante un desarrollo iterativo e incremental, abierto a los cambios y caracterizado por un código simple, al mismo tiempo que aplica un conjunto de prácticas que harán la entrega de la herramienta menos complicada y más satisfactoria tanto para los clientes como para el equipo de entrega.

1.8 Herramientas y tecnologías a utilizar

En todo proceso investigativo es necesario la utilización de sistemas de soporte que permitan organizar, facilitar, agilizar y automatizar las tareas generadas durante el transcurso de la investigación. Las herramientas y tecnologías empleadas con este fin son vitales para una correcta realización de las pruebas, y por consiguiente, justificar la calidad de los resultados obtenidos. Durante el presente estudio se utilizaron los recursos expuestos a continuación.

✓ **Lenguaje de modelado**

En el proceso de desarrollo de software, la etapa de modelado es de vital importancia y tiene una notable repercusión en su etapa de diseño e implementación, por proveer al ingeniero de un conjunto de notaciones, herramientas y prácticas, que le permiten "visualizar" el sistema a construir, logrando un nivel de abstracción que organice la lógica del mismo.

El **Lenguaje Unificado de Modelado 2.0** (UML por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de los sistemas de software (27). Ofrece un estándar para describir un "plano" del sistema, incluyendo aspectos conceptuales

tales como procesos del negocio, funciones y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

✓ **Herramienta para el diseño de la aplicación**

Una herramienta CASE (de sus siglas en inglés *Computer Aided Software Engineering*, o Ingeniería de Software Asistida por Computadora), es una herramienta individual para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del ciclo de desarrollo del software (o mantenimiento) (28).

Sus objetivos van dirigidos a mejorar la productividad en el desarrollo y mantenimiento de éste, aumentar su calidad, reducir su tiempo y costo de desarrollo y mantenimiento, mejorar la planificación de un proyecto y automatizar la documentación, la generación de código, las pruebas de errores y su gestión, contribuyendo a la reutilización del software, portabilidad y estandarización de la documentación y facilitando el uso de las distintas metodologías propias de la ingeniería de software.

Visual Paradigm 8.0 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 rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (29).

Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

✓ **.NET Framework 4.5**

El Framework .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones, mediante la cual se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables (30).

Los principales componentes de este entorno son:

- Lenguajes de compilación.
- Biblioteca de clases .Net.
- CLR (*Common Language Runtime*).

A continuación se resumen las ventajas más importantes que proporciona .Net *Framework* 4.5:

- **Código administrado:** El CLR realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- **Interoperabilidad multilinguaje:** El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL).
- **Compilación *just-in-time*:** El compilador JIT incluido en el *Framework* compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- **Garbage collector:** El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (*garbage collector*). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por qué liberar la memoria de forma explícita aunque también sea posible hacerlo manualmente (mediante el método *dispose()* se libera el objeto para que el recolector de basura lo elimine de la memoria).
- **Seguridad de acceso al código:** Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente de la web sin tener que preocuparse si esto va a estropear el sistema.
- **Despliegue:** Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El *Framework* realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.

✓ Lenguaje de programación

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft específicamente para ser utilizado en su plataforma .NET, por lo que desarrollar empleando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes con los que se puede codificar en dicha plataforma, actualmente se ha implementado un compilador con el cual generar programas codificados con C# para distintas plataformas como Unix, Mac OS y GNU/Linux. C# está basado en las experiencias adquiridas de los lenguajes anteriores a él. Su sintaxis básica proviene de C y C++, y combina las mejores cualidades de un lenguaje de programación de alto nivel, en la misma medida en que brinda un gran rendimiento. Es un lenguaje que posee muchas semejanzas con Java como son el estándar de código, el tipado seguro y unificado, así como su simplicidad y eficiencia (31).

Para la implementación de la herramienta, se utilizará el lenguaje de programación que se describió anteriormente. Con el empleo del mismo se pretende aprovechar las ventajas que este lenguaje ofrece, principalmente las relacionadas con el dominio en su utilización por parte de los autores así como su rapidez constatada, además de la existencia de gran cantidad de bibliotecas de código creadas en torno al mismo como es el caso de las bibliotecas de clases a utilizar para ejecutar los algoritmos biométricos en la herramienta, todas estas sustentadas por los años de explotación con los que cuenta el lenguaje y su constante actualización y corrección. La amplia comunidad creada en torno este posibilita obtener la retroalimentación adecuada y con la inmediatez requerida.

✓ Entorno de Desarrollo Integrado

El Entorno de Desarrollo Integrado (IDE) utilizado en el desarrollo de la herramienta fue **Microsoft Visual Studio 2013**. El mismo fue creado para sistemas operativos *Windows* y soporta varios lenguajes de programación entre los cuales se encuentre el escogido para el desarrollo del producto de software derivado de este estudio y el cual fue creado con intereses explícitos para dicho IDE. *Visual Studio*, además, permite el desarrollo web con ASP.NET y la programación en otros lenguajes como Visual C++ y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. *Visual Studio* permite a los desarrolladores crear aplicaciones, sitios web y servicios (a partir de la versión .NET 2002) en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se

intercomunicuen entre estaciones de trabajo, páginas web y dispositivos móviles. *Visual Studio* es un lanzamiento acumulativo que incluye todos los beneficios obtenidos en versiones anteriores (32).

✓ **Windows Presentation Foundation (WPF)**

WPF es utilizado para crear aplicaciones con interfaces visuales que sean atractivas para el usuario. Proporciona a los programadores un modelo de programación unificado que recopila experiencias de distintos clientes (33). El núcleo de WPF es un motor de representación basado en vectores independientes, creado para sacar partido de las prestaciones gráficas del hardware. WPF extiende el núcleo con un conjunto completo de características de desarrollo de aplicaciones que incluye el lenguaje XAML¹, controles, enlace de datos, diseño, gráficos 2D y 3D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en Microsoft *.NET Framework*, de modo que es posible compilar aplicaciones que incorporen otros elementos de la biblioteca de clases de *.NET Framework* (33).

WPF proporciona mejoras de programación adicionales para el desarrollo de aplicaciones cliente de *Windows*. Una mejora evidente es la capacidad para programar una aplicación mediante código de lenguaje marcado y subyacente. En general, se utiliza el lenguaje marcado XAML para implementar la apariencia de una aplicación, y los lenguajes de programación administrados (subyacentes) para implementar su comportamiento. Esta separación entre la apariencia y el comportamiento aporta las ventajas siguientes (33):

- Se reducen los costos de programación y mantenimiento, al no estar el marcado específico de la apariencia estrechamente relacionado con el código específico del comportamiento.
- La programación es más eficaz porque los diseñadores pueden implementar la apariencia de una aplicación al mismo tiempo que los programadores implementan su comportamiento.

¹ El lenguaje de marcado de aplicaciones extensibles, o XAML, es un lenguaje de marcado basado en XML desarrollado por Microsoft. XAML es el lenguaje que subyace a la presentación visual de una aplicación desarrollada en *Microsoft Expression Blend*, al igual que HTML es el lenguaje que subyace la presentación visual de una página web. La creación de una aplicación en *Expression Blend* supone tener que escribir código XAML, ya sea de forma manual o visual, mediante la vista Diseño de *Expression Blend*.

- Se pueden usar varias herramientas de diseño para implementar y compartir el marcado XAML, a fin de responder a los requisitos de quienes colaboran en la programación de aplicaciones; *Microsoft Expression Blend* proporciona una experiencia apropiada para los diseñadores, mientras que *Visual Studio* está dirigido a los programadores.

✓ **Managed Extensibility Framework (MEF)**

Es una biblioteca para crear aplicaciones ligeras y extensibles. Permite detectar y utilizar extensiones sin requisitos de configuración, encapsular código con facilidad y evitar dependencias lógicas frágiles, además permite reutilizar extensiones tanto dentro de una aplicación como entre aplicaciones. MEF es una parte integral de *.NET Framework* desde su versión 4 por lo que está disponible siempre que se use *.NET Framework* (34).

✓ **Gestor de base de datos**

Un Sistema Gestor de Base de Datos (SGBD) permite procesar, describir, administrar y recuperar los datos almacenados en un banco de datos. Estos sistemas brindan un conjunto de programas, procedimientos y lenguajes para consultar los datos y realizar acciones sobre ellos, garantizando además, la seguridad de los mismos (35).

PostgreSQL 9.4 es un SGBD objeto-relacional disponible en un amplio rango de plataformas y que se distribuye bajo la licencia BSD (*Berkeley Software Distribution*) de software libre. Es mantenido por la organización *PostgreSQL Global Development Team* y por la colaboración de una amplia comunidad de usuarios y programadores denominada PGDG (*PostgreSQL Global Development Group*). Es en la actualidad el SGBD de código abierto más potente del mercado (36). Disímiles son las empresas y compañías que construyen sus productos de software utilizando este gestor, entre ellas Debian, Apple, Red Hat y Sun Microsystems (37).

Cumple con los estándares SQL de interoperabilidad y compatibilidad, destacándose por su rendimiento, seguridad y alta disponibilidad (38). Utiliza la tecnología MVCC (*Multi-Version Concurrency Control*) para conseguir una mejor respuesta en ambientes de grandes volúmenes de datos e implementa el modelo cliente-servidor.

Entre las principales características de PostgreSQL se encuentran el uso de procedimientos almacenados, el soporte de integridad referencial y manejo de distintos tipos de datos.

✓ ***NHibernate 4.0***

En el desarrollo de gran número de aplicaciones empresariales se hace necesario persistir la información manejada por el sistema en bases de datos. En este proceso se genera una gran cantidad de código repetitivo que es obligatorio para manejar las conexiones a la base de datos, generar consultas entre otros. Esto provoca que el código de las aplicaciones se vuelva engorroso y que los programadores se desconcentren de los problemas que realmente tienen que resolver de lógica del negocio por atender detalles de una tecnología en específico.

NHibernate es un marco que permite hablar de una base de datos relacional de una manera orientada a objetos. Se puede almacenar (o como también se suele decir, "persistir") los objetos de una base de datos y cargar los objetos de la base de datos más adelante. *NHibernate* genera las sentencias SQL necesarias para insertar, actualizar, eliminar y cargar datos (39).

Conclusiones parciales

El análisis de las etapas del proceso de reconocimiento de personas mediante huellas dactilares, permitió identificar la necesidad de desarrollar una herramienta para realizar el proceso de cotejo visual de impresiones dactilares. La caracterización de las principales herramientas de cotejo visual de huellas dactilares, encontradas en la investigación, permitió la detección de funcionalidades requeridas para realizar este proceso. Los conceptos y otras temáticas tratadas en el capítulo posibilitaron una mejor comprensión del negocio. La selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo de esta investigación posibilitará la creación de la herramienta dactiloscópica.

Capítulo 2: Características del sistema

Introducción

En este capítulo se abordan las principales características de la solución propuesta. Se define un modelo de dominio donde se agrupan los principales conceptos a tratar y la metáfora que describe el comportamiento de la herramienta. Se realiza la captura de los requisitos funcionales y no funcionales lo que permite que sean conformadas las historias de usuario relacionadas a estos. En correspondencia con la etapa de planificación del software se puntualizan el plan de entregas y el plan de iteraciones. Se realizan las tarjetas CRC correspondientes a cada una de las clases y una descripción de la arquitectura del sistema, así como de los patrones de diseño utilizados, logrando de esta forma un entendimiento mayor de la herramienta para su implementación.

2.1 Modelo de dominio

El modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, mediante el cual se podrá comprender el entorno de la herramienta a desarrollar y permitirá a los usuarios, desarrolladores e interesados utilizar un vocabulario común para poder entender el contexto en que se desarrolla la herramienta (40). El modelo de dominio mostrará los principales conceptos con los que trabaja la herramienta, describiendo las clases fundamentales dentro del contexto de la aplicación mostradas a continuación (ver figura 2.1)

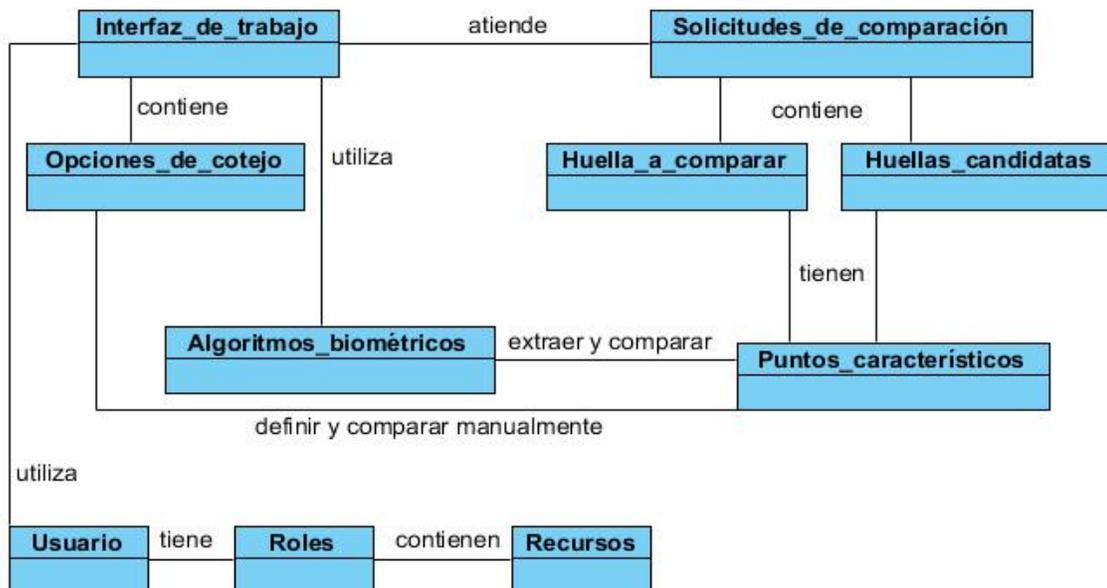


Figura 2. 1 Modelo de dominio.

2.1.1 Glosario de conceptos del modelo de dominio

A continuación se definen brevemente los conceptos ilustrados en el modelo de dominio.

- ✓ **Interfaz de trabajo:** brinda al especialista una interfaz de trabajo acorde a sus necesidades.
- ✓ **Solicitudes de comparación:** permite atender cada una de las solicitudes de comparación.
- ✓ **Huella a comparar:** imagen que contiene la huella dactilar que se desea comparar.
- ✓ **Huellas candidatas:** imágenes que contienen las huellas dactilares con las que se va a comparar.
- ✓ **Puntos característicos:** elementos que caracterizan una huella dactilar.
- ✓ **Opciones de cotejo:** brindan un grupo de funcionalidades que le servirán de ayuda al especialista pericial en el momento de realizar una comparación.
- ✓ **Algoritmos biométricos:** se encargan de realizar el cotejo de huellas dactilares, el conteo de crestas papilares y la extracción automática de puntos característicos y singulares de las huellas dactilares.
- ✓ **Usuario:** encargado de comparar las huellas dactilares tras haberlas cargado en formato de imagen a la herramienta.
- ✓ **Roles:** a cada usuario le corresponde un rol y a partir de este se accede a los recursos de la aplicación.
- ✓ **Recursos:** representan los recursos de la aplicación a los que puede acceder cada usuario dependiendo del rol que le corresponde.

2.2 Metáfora

Una vez descrito el entorno del sistema a desarrollar en el modelo de dominio, es preciso crear una metáfora que describa cómo este debe funcionar. A continuación se describe la metáfora global que guía el desarrollo del presente proyecto.

La herramienta de cotejo visual de huellas dactilares poseerá las funcionalidades necesarias para realizar el cotejo por parte de especialistas periciales. Para acceder a la herramienta el especialista debe autenticarse. Una vez autenticado podrá realizar una comparación, gestionar la configuración de la herramienta así como analizar o editar una comparación guardada con anterioridad. Para iniciar una comparación el especialista debe cargar la huella a comparar y la lista de posibles candidatos desde un directorio local. Una vez cargadas las huellas dactilares, la herramienta brindará un conjunto de funcionalidades que permitirán al especialista decidir si la huella a comparar coincide con alguna de las posibles candidatas. Las comparaciones realizadas en la herramienta se podrán almacenar permitiendo ser modificadas o analizadas posteriormente, registrando en cada momento quien la realizó o modificó. En la base de datos de la herramienta se almacenarán las direcciones de las huellas comparadas así como las características de estas. En la interfaz de configuración el especialista podrá seleccionar los algoritmos de extracción de características y cotejo automático de huellas dactilares que desee emplear en la comparación.

2.3 Captura de requisitos

La captura de los requisitos funcionales y no funcionales es uno de los procedimientos más importantes para el desarrollo de cualquier software. En esta etapa el principal objetivo es identificar lo que el cliente realmente necesita, de manera que el equipo de desarrollo pueda trabajar en base a estos requerimientos y logre obtener un producto que funcione con la calidad deseada.

2.3.1 Requisitos funcionales (RF)

Los requisitos funcionales son características de la herramienta que expresan las funcionalidades que el sistema debe cumplir. A continuación se enumeran los requisitos funcionales de la herramienta a desarrollar, los cuales serán descritos con mayor detalle en las historias de usuario.

- RF1** Autenticar usuario.
- RF2** Gestionar usuario.
 - a) Adicionar usuario.
 - b) Listar usuario.
 - c) Modificar usuario.
 - d) Eliminar usuario.
- RF3** Cargar imágenes de las huellas dactilares.
 - a) Cargar imagen de la huella dactilar a comparar.
 - b) Cargar imágenes de las huellas dactilares candidatas.
- RF4** Cargar algoritmos biométricos.
- RF5** Seleccionar puntos característicos y puntos singulares de forma manual.
- RF6** Detectar puntos característicos y puntos singulares de forma automática.
- RF7** Medir distancia entre puntos.
- RF8** Contar puntos característicos.
- RF9** Contar crestas papilares.
- RF10** Cotejo automático de huellas dactilares.
- RF11** Superponer imágenes de huellas dactilares.
- RF12** Aplicar transformaciones a la imagen de la huella dactilar.
 - a) Mover y rotar imagen.
 - b) Modificar tamaño de la imagen.
 - c) Modificar brillo de la imagen.
 - d) Modificar contraste de la imagen.
 - e) Llevar la imagen a escala de grises.
 - f) Invertir imagen.
- RF13** Gestionar comparaciones.
 - a) Guardar comparación.
 - b) Listar comparaciones.
 - c) Modificar comparación.
 - d) Eliminar comparación.

2.3.2 Requisitos no funcionales (RNF)

Los requisitos no funcionales son propiedades o cualidades que hacen al producto atractivo, usable, rápido y confiable. Se definieron como requisitos no funcionales:

1. Software

- ✓ El sistema operativo debe ser Windows Vista *Service Pack 2* o superior.
- ✓ *Framework* de desarrollo .NET 4.5.

2. Hardware

- ✓ Periféricos: Mouse y teclado.
- ✓ CPU 1.0 GHZ o superior.
- ✓ 512 MB o más de memoria RAM.

3. Restricciones en el diseño y la implementación

- ✓ Lenguaje de programación: C#.
- ✓ IDE: Visual Studio 2013.
- ✓ Para el modelado de UML se utiliza: Visual Paradigm 8.0.
- ✓ Gestor de base de datos: PostgreSQL 9.4.

4. Requisitos de seguridad

- ✓ Proteger la información manejada por la herramienta de accesos no autorizados.
- ✓ Garantizar que las funcionalidades de la herramienta se muestren de acuerdo al nivel de usuario que esté activo.

5. Apariencia o Interfaz

- ✓ Se utilizarán colores claros que ofrezcan suficiente contraste entre texto y fondo para no dificultar la lectura.
- ✓ Empleo de animaciones sencillas, permitiendo la rapidez de la herramienta.

2.4 Historias de usuarios (HU)

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, representando una breve descripción del comportamiento del mismo, con el empleo de una terminología del cliente sin lenguaje técnico. Se emplean para hacer estimaciones de tiempo y para el plan de entregas, además de ser fundamentales en la creación de las pruebas de aceptación. Deben tener como

características fundamentales: ser independientes unas de otras, negociables, valoradas por los clientes o usuarios, estimables, pequeñas y verificables (24). Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario la establecen los programadores utilizando como medida el punto, que equivale a una semana ideal de programación.

A continuación se define como ejemplo una de las historias de usuario de la solución propuesta, las demás se encuentran descritas en el Anexo 2:

Historia de Usuario	
Número: HU_5	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Seleccionar puntos característicos y puntos singulares de forma manual.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá seleccionar los puntos característicos y los puntos singulares que identifique en la imagen de la huella dactilar.	
Observaciones: Los puntos característicos y singulares que podrá seleccionar son bifurcación, terminación, núcleo y delta.	

Tabla 2. 1 HU Seleccionar puntos característicos y puntos singulares de forma manual.

2.5 Planificación

La planificación en XP está basada en un conjunto de decisiones tomadas por el cliente de conjunto con los programadores. Los clientes representan las necesidades propias del negocio como son el alcance del proyecto, la prioridad de las historias de usuario y las fechas de entrega de cada versión del producto final. Mientras que los programadores definen los requerimientos técnicos que complementan las necesidades del negocio, como la duración estimada de la implementación de las historias de usuario y la organización del proceso de desarrollo en general.

2.5.1 Plan de entregas

Después de establecida las prioridades de las historias de usuario por el cliente se obtiene un cronograma sobre las versiones resultantes al final de cada iteración. Para la presente solución se han definido 3 iteraciones en las que se codificarán las 13 historias de usuario obtenidas, para un tiempo total de desarrollo de 16 semanas, según el plan de entregas realizado.

Entregable	Iteración	Fin de iteración
Gestión de usuarios, carga de imágenes de huellas dactilares y carga de algoritmos biométricos.	1	Febrero 2015
Trabajo con las características de las huellas dactilares.	2	Abril 2015
Aplicar transformaciones a la imagen de la huella dactilar y gestión de las comparaciones.	3	Mayo 2015

Tabla 2. 2 Plan de entregas.

2.5.2 Plan de iteraciones

Para obtener un prototipo funcional al final de cada iteración, se ha definido un plan de iteraciones con el orden en el que se implementarán las historias de usuario.

Iteración	Historia de usuario	Semanas estimadas
1	Autenticar usuario.	4
	Gestionar usuario.	
	Cargar imágenes de las huellas dactilares.	
	Cargar algoritmos biométricos.	
2	Seleccionar puntos característicos y puntos singulares de forma manual.	8
	Detectar puntos característicos y puntos singulares de forma automática.	
	Medir distancia entre puntos.	
	Contar puntos característicos.	
	Contar crestas papilares.	
	Cotejo automático de huellas dactilares.	
	Superponer imágenes de huellas dactilares.	

3	Aplicar transformaciones a la imagen de la huella dactilar.	4
	Gestionar comparaciones.	

Tabla 2. 3 Plan de iteraciones.

En la primera iteración se realizarán las historias de usuarios de las que depende la autenticación en el sistema, la gestión de los usuarios que tendrán acceso al mismo, la lectura de las imágenes de huellas dactilares que se van a comparar, así como la carga de los algoritmos biométricos. En las dos últimas iteraciones se realizarán las historias de usuarios más complejas y que inciden críticamente en la lógica de la herramienta. Una vez concluida estas iteraciones la herramienta se encontrará completamente concluida y lista para su explotación.

2.6 Diseño

En el diseño se indica cómo se construirá la solución mediante la definición de una estructura lo más sencilla posible que responda a la satisfacción de los requisitos funcionales y no funcionales. Se realiza la identificación de los objetos lógicos que componen al software para su posterior implementación, de ellos se definen sus atributos y operaciones.

2.6.1 Definición de la arquitectura

Según el documento del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) estándar 1471-2000, la arquitectura de software es la organización fundamental de un sistema donde se definen los componentes que lo integran, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (41). Para el desarrollo y organización estructural de la herramienta se propone como arquitectura base, una arquitectura en capas (ver figura 2.2), constituyendo una distribución jerárquica de responsabilidades para proporcionar una división de los problemas a resolver. La arquitectura propuesta para el desarrollo de la herramienta identifica como capas:

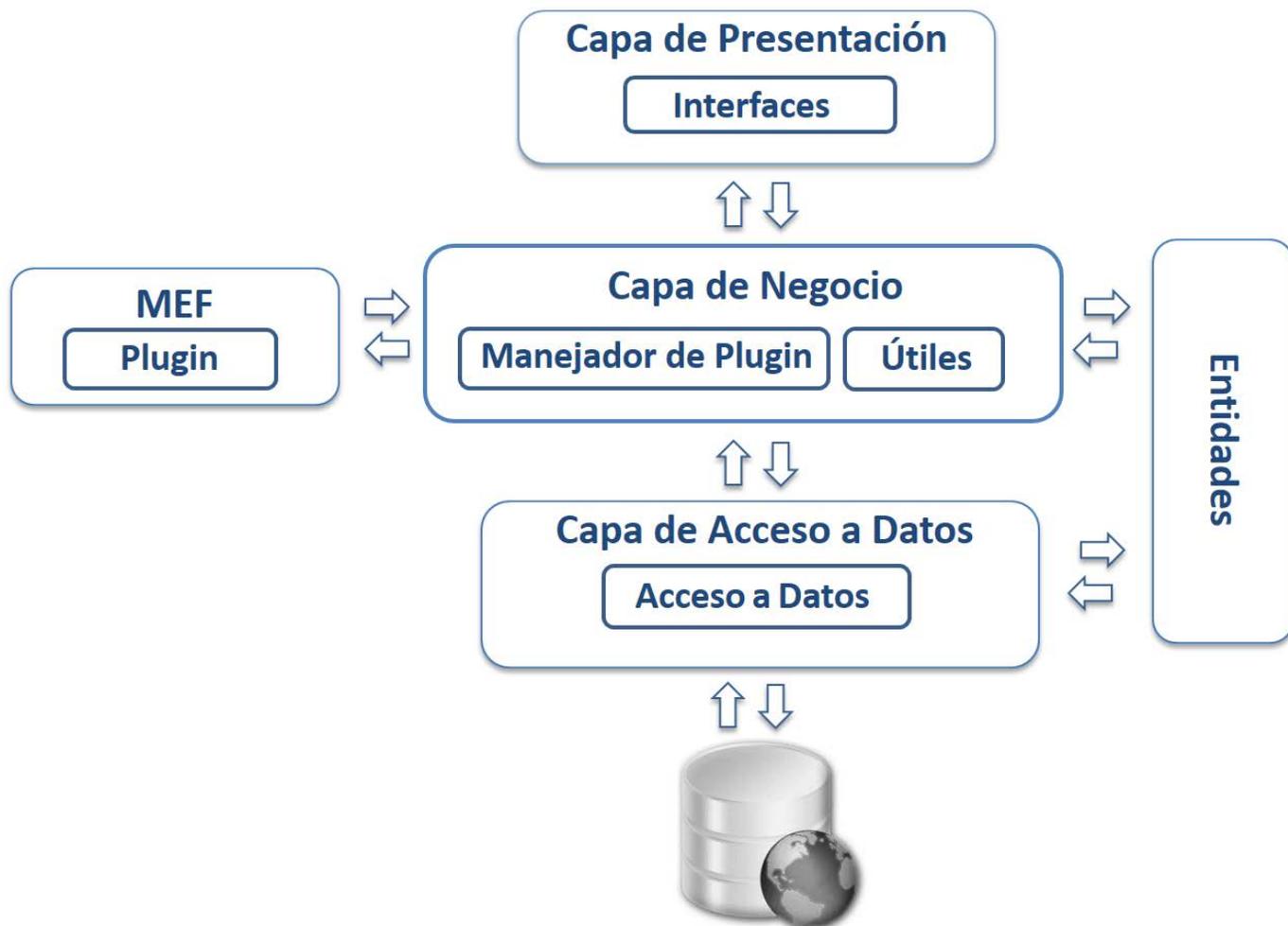


Figura 2. 2 Arquitectura de la herramienta.

Capa de Presentación: Contiene las interfaces del sistema.

Capa de Negocio: Se encarga de manejar la lógica del negocio de la herramienta desde la ejecución de los algoritmos de extracción de características hasta la gestión de la configuración.

Capa de Acceso a Datos: Se encarga de la comunicación de la herramienta con la base de datos, a través de ella se persisten los datos y se realizan las consultas de estos.

MEF: Se encuentran las bibliotecas de clases que contienen los algoritmos biométricos.

Entidades: Contiene las clases entidades que se gestionan en la herramienta, persisten en la base de datos y se muestran en la presentación.

2.6.2 Patrones de diseño

Los patrones de diseño expresan esquemas para solucionar problemas del mismo tipo que comúnmente se presentan en el diseño del software (40).

Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones de este tipo usados son:

Experto: Este patrón 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 realizarla (40). El mismo se ve reflejado en la clase MefAux que implementa los métodos relacionados con la creación de los catálogos de algoritmos biométricos.

Creador: Este patrón es empleado para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia del mismo solo pueda ser creada por la clase que contiene la información necesaria para ello (40). En la arquitectura de la herramienta se ve reflejada la utilización de este patrón en la clase Huella quien contiene un objeto de la clase Minucia, en la cual para crear una instancia de esta es necesario suministrarle la información requerida, dígame posición en el eje de las abscisas y ordenadas, así como el tipo y el ángulo.

Alta cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con funciones estrechamente relacionadas para que estas no realicen un trabajo que no esté asignado directamente a su responsabilidad. Está basado en la asignación de responsabilidades teniendo en cuenta que la cohesión permanezca alta (40). En el diseño de la herramienta, como ejemplo de la aplicación de este patrón se encuentra la clase NegocioImpl, que es responsable de manejar la interacción con la base de datos.

Controlador: Es un intermediario entre la capa de presentación y el núcleo de las clases donde reside la lógica del sistema, que coordina la actividad de otros objetos (40). Por ejemplo, en la herramienta se hace

uso del patrón controlador definiendo la clase Controlador que se encarga de manejar todos los eventos de la herramienta y maneja varias instancias de objetos.

Los patrones GOF (Banda de los cuatros) describen las formas en las que pueden ser organizados los objetos para trabajar unos con otros, formando estructuras de mayor complejidad (40). Los patrones de este tipo usados son:

Patrones de Creación: El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

- ✓ *Singleton* (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia, el patrón se ve reflejado en la herramienta en la clase Util.

```
private static Util instance = null;
public static Util Instance
{
    get
    {
        if (instance == null)
            instance = new Util();
        return instance;
    }
}
```

Figura 2. 3 Patrón de creación Singleton.

Patrones Estructurales: Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.

- ✓ Fachada: Proporciona una interfaz unificada de alto nivel que facilite su uso, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ella. La interfaz genérica IExtraccionMinucia garantiza un punto de comunicación único para todas las clases que necesiten la extracción de puntos característicos de la huella dactilar, con el objetivo de desarrollar los procedimientos que en ella se implementan.

Patrones de comportamiento: Se definen como patrones de diseño de software que ofrecen soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

- ✓ Estrategia: Define una familia de algoritmos, encapsulando por separado cada uno de ellos y haciéndolos intercambiables. Esto permite a los algoritmos variar con independencia de los clientes que lo utilizan. En la solución propuesta se evidencia el uso de este patrón ya que las interfaces IExtraccionMinucia, IExtraccionPtosSingulares, ICotejo representan a las familias de algoritmos encargados de extraer características y comparar huellas dactilares.

2.6.3 Tarjetas CRC

Las tarjetas CRC (Clase - Responsabilidad - Colaboración), constituyen uno de los artefactos de la metodología XP que guía el proceso de desarrollo de la solución propuesta. Se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

Una clase es la definición de las características concretas de un determinado tipo de objetos. Las responsabilidades de una clase son las funcionalidades que realiza y sus atributos. Los colaboradores son las demás clases con las que interactúa con el fin de cumplir sus responsabilidades.

A continuación se define como ejemplo una de las tarjetas CRC de la solución propuesta, las demás se encuentran descritas en el Anexo 3:

CotejoLista	
Responsabilidades	Colaboradores
Cargar imágenes de huellas dactilares.	✓ Controlador.
Obtener puntos característicos y singulares de forma automática.	✓ MefAux. ✓ Minucia.
Cargar una comparación.	✓ Negociolmpl. ✓ Minucia. ✓ PuntoLinea. ✓ dComparacion.
Dibujar puntos característicos y singulares.	✓ Minucia.

Tabla 2. 4 Tarjeta CRC de la clase CotejoLista.

2.6.4 Modelo de datos

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases entidad son las que contienen toda la información persistente manejada por la herramienta, cuya estructura física y lógica describe el modelo de datos. A continuación se muestra el modelo de datos de la herramienta a desarrollar (ver figura 2.4). La descripción detallada de las tablas se encuentra en el Anexo 4.

usuario: Guarda los datos de los usuarios del sistema.

rol: Guarda los roles a los que puede pertenecer un usuario.

permiso: Almacena el nombre de los recursos a los que pueden acceder los usuarios.

rol_permiso: Representa la asociación de las tablas rol y permiso.

comparacion: Guarda los datos de las comparaciones que se realizan en el sistema.

huella: Almacena la dirección de las imágenes de las huellas dactilares que se han comparado y la lista de minucias de cada una de ellas.

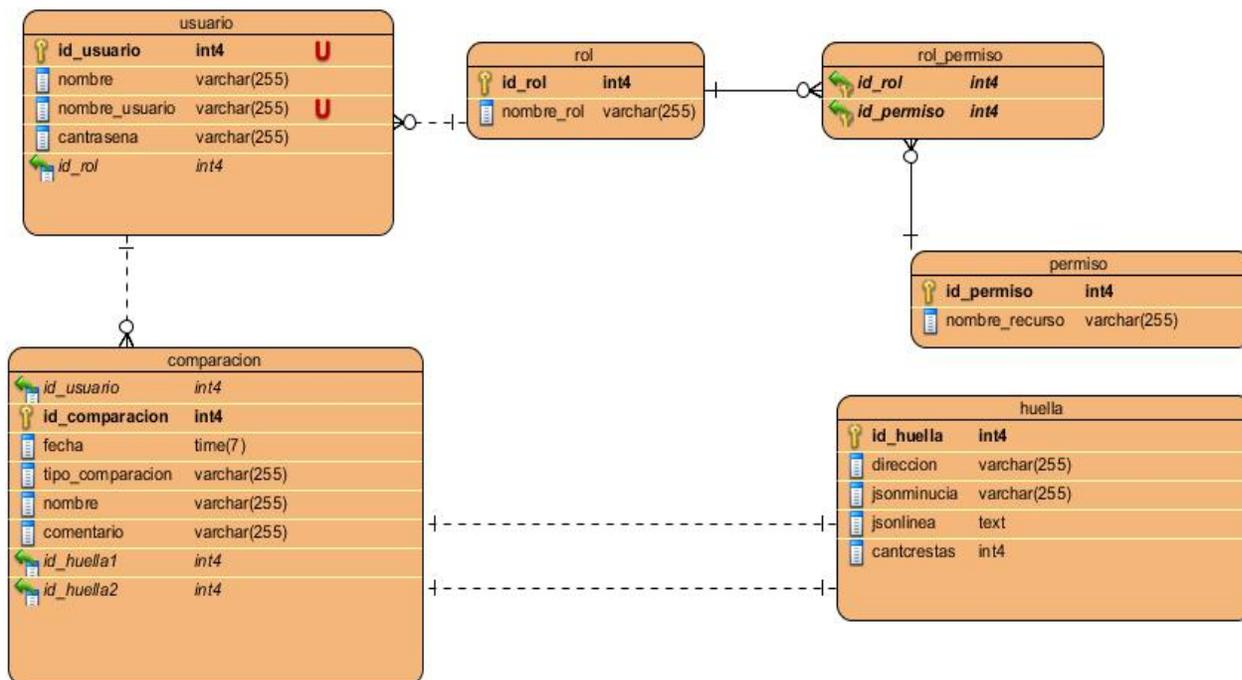


Figura 2. 4 Diagrama entidad-relación de la herramienta.

Conclusiones parciales

La definición del modelo de dominio y la explicación detallada del proceso que debe ejecutar la herramienta, permitieron identificar con mayor facilidad los requisitos funcionales que esta debe satisfacer, propiciando así la comprensión de su funcionamiento. Las historias de usuario especificadas para cada uno de estos requisitos sin emplear lenguaje técnico y sí términos del cliente, detallando entre otros elementos el tiempo que requieren para su codificación, favorecieron la estimación del plan de entregas del producto delimitando el propósito y duración de cada iteración.

Con el uso de la arquitectura global en capas especificada, así como algunos de los patrones GRASP y GOF, se logró organizar la vista lógica de la solución, de manera que las clases identificadas y sus relaciones sean las bases para su implementación.

Capítulo 3: Implementación y pruebas

Introducción

En este capítulo se exponen las particularidades de la implementación de la herramienta, en aras de obtener un producto final que esté en correspondencia con los requisitos definidos por el cliente, lo cual se validará aplicando diferentes tipos de prueba que propone la metodología de desarrollo de software utilizada. Para ello se definen los estándares de codificación que debe cumplir el equipo de desarrollo, se crea el diagrama de componentes y el de despliegue. A partir del código resultante y su funcionamiento se ejecutan las pruebas unitarias y de aceptación.

3.1 Implementación

Una vez definidas las historias de usuario y concluido el diseño se pasa a la etapa de codificación de la solución propuesta cuyos objetivos van destinados a desarrollar de forma iterativa e incremental un producto completo que esté preparado para la transición a su comunidad de usuarios, consiguiendo versiones útiles de forma rápida y práctica, que paulatinamente completen la planeación, diseño, desarrollo y prueba de todas las funcionalidades necesarias. Durante la codificación la programación debe ser por parejas, con el objeto de crear código para cada historia de usuario en dependencia de lo concebido en el plan de iteraciones y las tareas ingenieriles. Esto da un mecanismo para la solución de problemas en tiempo real y también para el aseguramiento de la calidad en tiempo real. A medida que concluya cada iteración, el código desarrollado se integrará con el resto, puesto que esta estrategia de integración continua ayuda a evitar los problemas de compatibilidad e interfaces, y a descubrir a tiempo los errores.

3.1.1 Tareas de ingeniería

La metodología XP plantea que la implementación de un software se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de desarrollo en la etapa de planificación. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. En la tabla 3.1 se describen las tareas correspondientes a las historias de

usuario de la segunda iteración del proceso de desarrollo de la herramienta. Las tareas del resto de las iteraciones se encuentran en el Anexo 5.

Iteración 2	
Historia de Usuario	Tareas
Seleccionar puntos característicos y puntos singulares de forma manual.	<ol style="list-style-type: none"> 1. Obtener las coordenadas del mouse donde se quiere colocar el punto. 2. Dibujar el punto.
Detectar puntos característicos y puntos singulares de forma automática.	<ol style="list-style-type: none"> 1. Ejecutar algoritmo de detección de puntos característicos. 2. Dibujar los puntos detectados sobre la imagen de la huella.
Medir distancia entre puntos.	<ol style="list-style-type: none"> 1. Obtener punto donde comienza la medición. 2. Obtener el punto donde culmina la medición. 3. Dibujar la línea entre los puntos. 4. Mostrar la distancia que existe entre los puntos.
Contar puntos característicos.	<ol style="list-style-type: none"> 1. Obtener la cantidad de puntos que se hayan detectado. 2. Mostrar la cantidad de puntos.
Contar crestas papilares.	<ol style="list-style-type: none"> 1. Ejecutar algoritmo de detección de crestas. 2. Obtener la cantidad de crestas detectadas. 3. Mostrar cantidad de crestas.
Cotejo automático de huellas.	<ol style="list-style-type: none"> 1. Ejecutar algoritmo de cotejo. 2. Mostrar respuesta del algoritmo.
Superponer imágenes de huellas dactilares.	<ol style="list-style-type: none"> 1. Superponer las imágenes. 2. Hacer transformaciones a las imágenes.

Tabla 3. 1 Tareas de ingeniería de la segunda iteración.

3.1.2 Estándares de codificación

Entre las prácticas a aplicar durante el proceso de desarrollo de software que propone la metodología XP se encuentran la refactorización del código y la propiedad compartida de éste, de forma que todo el personal pueda corregir y extender cualquier parte del producto. Para complementar estas prácticas, la metodología enfatiza en que la comunicación de los programadores es a través del código, por lo cual es indispensable que se sigan ciertos estándares de programación que le provean legibilidad. En la propuesta de solución

para declarar el nombre de las variables, métodos y clases se tendrán en cuenta las siguientes convenciones:

- ✓ En todos los casos la definición de los mismos será en español.
- ✓ Se utiliza el estilo de escritura CamelCase² y en específico el tipo lowerCamelCase.
- ✓ No se utilizará una misma línea para definir más de una variable y siempre que sea posible éstas se inicializarán en su misma línea de declaración.

3.1.2.1 Comprensión y legibilidad del código

Con el objetivo de incrementar la legibilidad del código se comentaron todas las declaraciones de clases y funciones más complejas. Los atributos, propiedades y constructores aparecen en la parte superior de las clases, mientras que los métodos privados y públicos están a continuación. Además, se organizó el código de forma estructurada en bloques de código, para una mejor comprensión del mismo.

3.1.3 Diagrama de componentes

El diagrama de componentes representa, como indica su nombre, la división del software en partes más pequeñas denominadas componentes. Muestra la organización y las dependencias entre los distintos elementos de software, sean estos código fuente, binario o ejecutable. Las dependencias en este diagrama indican que un componente utiliza los servicios ofrecidos por otro componente. A continuación se describe el diagrama de componentes de la herramienta (ver figura 3.1).

² Estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos de CamelCase: UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula y lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula.

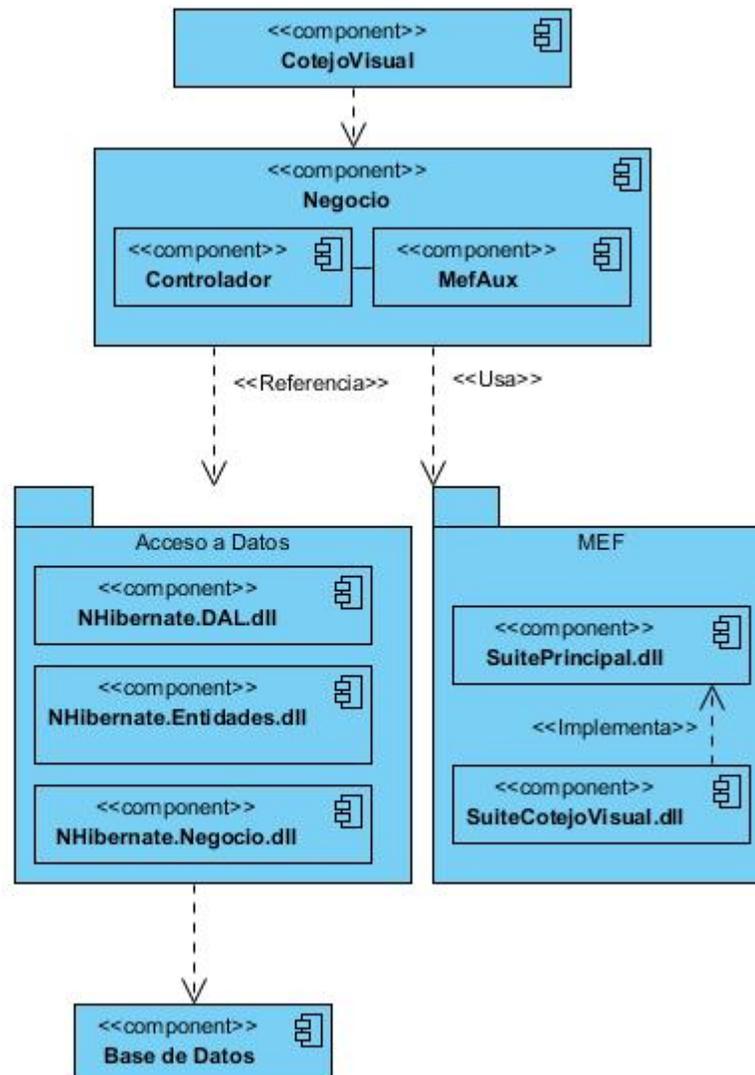


Figura 3. 1 Diagrama de componentes de la herramienta.

En el diagrama se muestra la conformación de la solución por componentes. El proyecto CotejoVisual contiene todos los elementos de la interfaz de la herramienta.

El componente Negocio se encarga de manejar la lógica de toda la herramienta, este contiene el elemento Controlador que se encarga de manejar los datos que van a ser mostrados en la interfaz, además contiene el elemento MefAux que es utilizado para comunicarse con el componente MEF en el que se encuentran los algoritmos biométricos.

Los componentes de acceso a datos (Acceso a Datos, Entidades y Negocio) contienen los elementos que modelan los datos y el acceso a estos en la bases de datos.

3.1.4 Diagrama de despliegue

El modelo de despliegue es uno de los artefactos obtenidos durante el proceso de desarrollo del software, con el objetivo de capturar los elementos de configuración del procesamiento, las conexiones entre esos elementos y visualizar su distribución en los nodos físicos. De manera general este tipo de modelo está compuesto por:

- ✓ **nodos** o elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- ✓ **dispositivos** caracterizados por ser nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- ✓ **conectores** que expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

El modelo de despliegue de la herramienta dactiloscópica para la gestión pericial en un AFIS está compuesto por una computadora que tendrá instalada la herramienta y mediante el protocolo TCP se realizará el intercambio de información con el servidor de base de datos (ver figura 3.2).

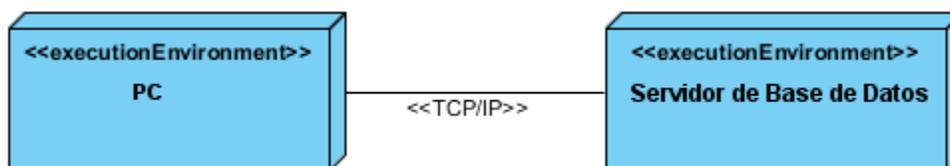


Figura 3. 2 Diagrama de despliegue de la herramienta.

3.1.5 Interfaces de usuario

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, por lo cual su diseño debe ser amigable y consistente, suelen ser fáciles de entender y fáciles de accionar. Una aplicación con una interfaz bien diseñada, además de un buen diseño gráfico, debe tener una buena navegabilidad, usabilidad y distribución de los contenidos.

A continuación se expone la interfaz que permite realizar el cotejo visual de huellas dactilares. Las interfaces restantes se muestran en el Anexo 6.

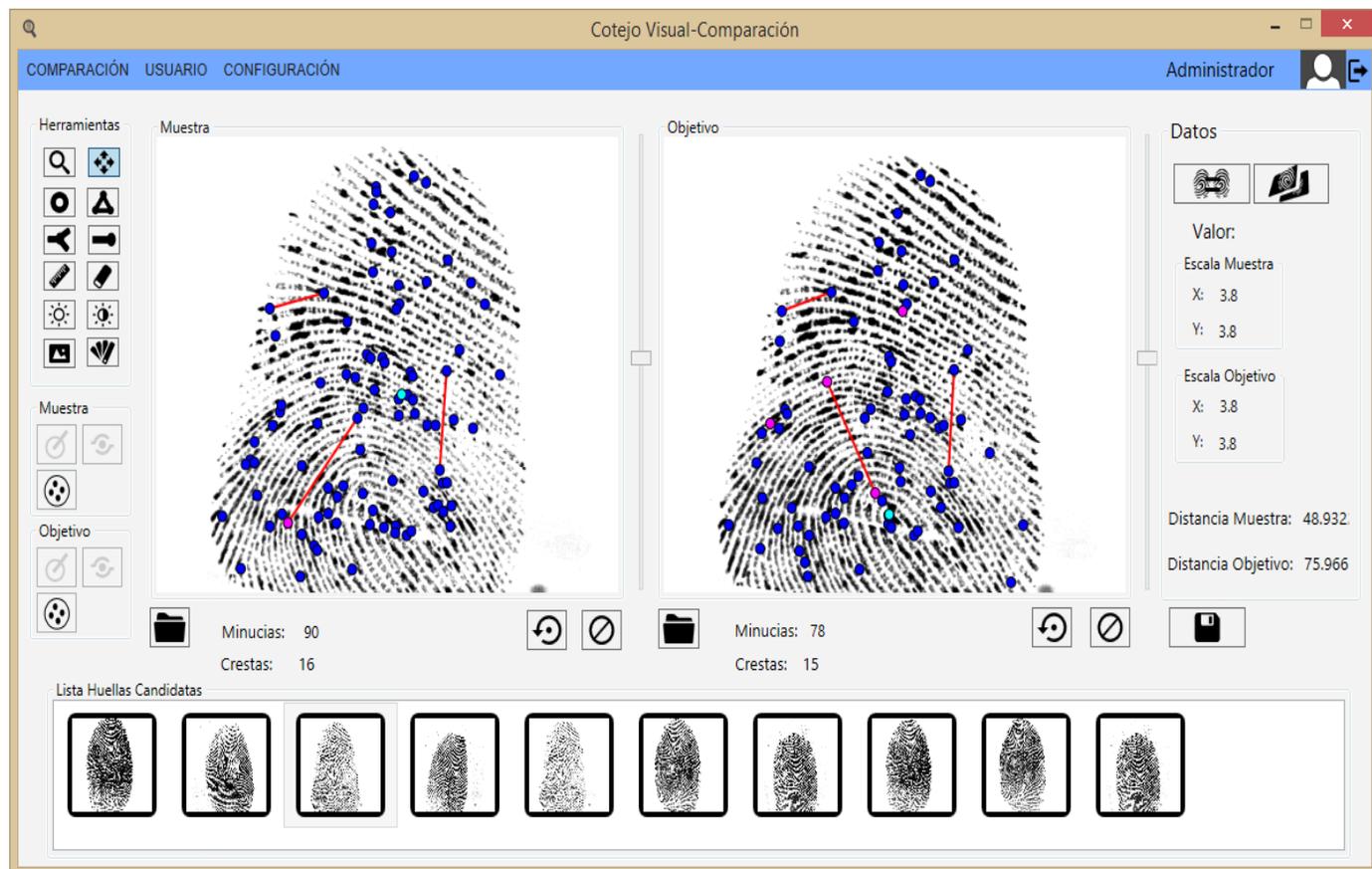


Figura 3. 3 Interfaz de cotejo visual de huellas dactilares.

3.2 Pruebas

Las pruebas de software son fundamentales para verificar la calidad del producto software teniendo como base el control de la satisfacción de los requisitos. Su principal objetivo es detectar errores. La metodología XP establece dos tipos de pruebas: pruebas unitarias y pruebas de aceptación o funcionales.

3.2.1 Pruebas unitarias

Las pruebas unitarias son escritas por los programadores antes de comenzar la codificación. El objetivo de estas pruebas es aislar pequeñas e individuales porciones del código para verificar que no tengan errores.

Para la realización de estas pruebas se utilizó un *Add-in* de *Visual Studio* llamado *ReSharper* (R#) que permite detectar errores y brinda soluciones instantáneas para corregirlos. Los resultados de estas pruebas se muestran en el Anexo 7.

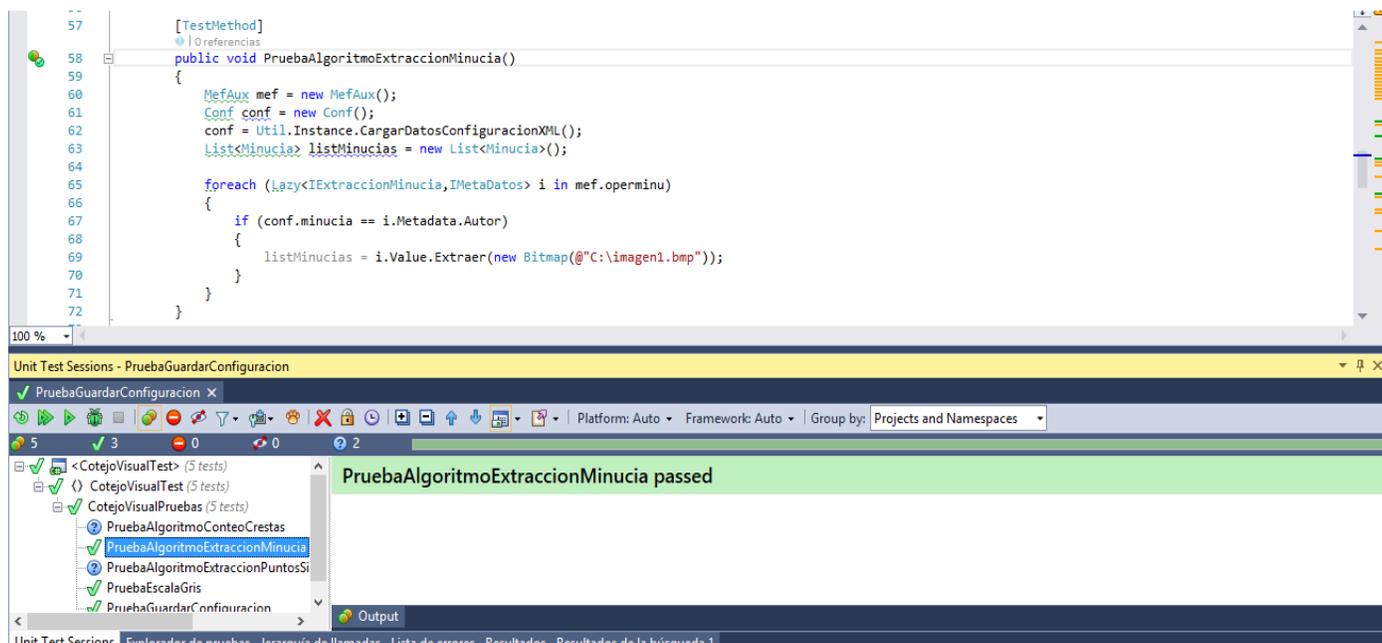


Figura 3. 4 PU Algoritmo de extracción de puntos característicos.

3.2.2 Pruebas de aceptación

La prueba de aceptación del usuario es la prueba final antes del despliegue del producto obtenido. Su objetivo es verificar que el software esté listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido.

En el caso de XP las pruebas de aceptación son creadas en base a las historias de usuario, en cada iteración del ciclo de desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que sus resultados sean correctos, y en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una

historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación (42).

A continuación se muestran los casos de pruebas de aceptación correspondiente a las historias de usuario “Seleccionar puntos característicos y puntos singulares de forma manual y Detectar puntos característicos y puntos singulares de forma automática”. Los casos de prueba de aceptación definidos por el cliente para el resto de las historias de usuario se especifican en el Anexo 8.

Caso de Prueba de Aceptación	
Código: CP1_HU5	HU_5: Seleccionar puntos característicos y puntos singulares de forma manual.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se puedan seleccionar los puntos característicos y puntos singulares de forma manual tanto en la imagen de la huella dactilar a comparar como en la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas el tipo de punto que se desea ubicar. 	
Resultado esperado: Se muestran los puntos característicos y puntos singulares seleccionados.	
Evaluación de la prueba: Satisfactoria	

Tabla 3. 2 CP Seleccionar puntos característicos y puntos singulares de forma manual.

Caso de Prueba de Aceptación	
Código: CP1_HU6	HU_6: Detectar puntos característicos y puntos singulares de forma automática.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se puedan detectar los puntos característicos y los puntos singulares de forma automática tanto en la imagen de la huella dactilar a comparar como en la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares y tener seleccionado algún algoritmo de extracción de puntos.	

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona la opción de extracción automática del tipo de punto que se desea obtener.
<p>Resultado esperado: Se muestran los puntos extraídos de forma automática.</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Tabla 3. 3 CP Detectar puntos característicos y puntos singulares de forma automática.

3.2.3 Resultados de las pruebas

Después de haber realizado las pruebas de aceptación de la herramienta a partir de los casos de pruebas definidos por el cliente durante las tres iteraciones de codificaciones planificadas, se detectaron en cada una de éstas 4, 10 y 5 no conformidades respectivamente, lo que equivale a un total de 19 no conformidades, que revelaban errores en la codificación que en la mayoría de los casos no incidían significativamente en las respuestas esperadas. Estas dificultades fueron solucionadas en un corto plazo, antes de pasar a la iteración posterior.

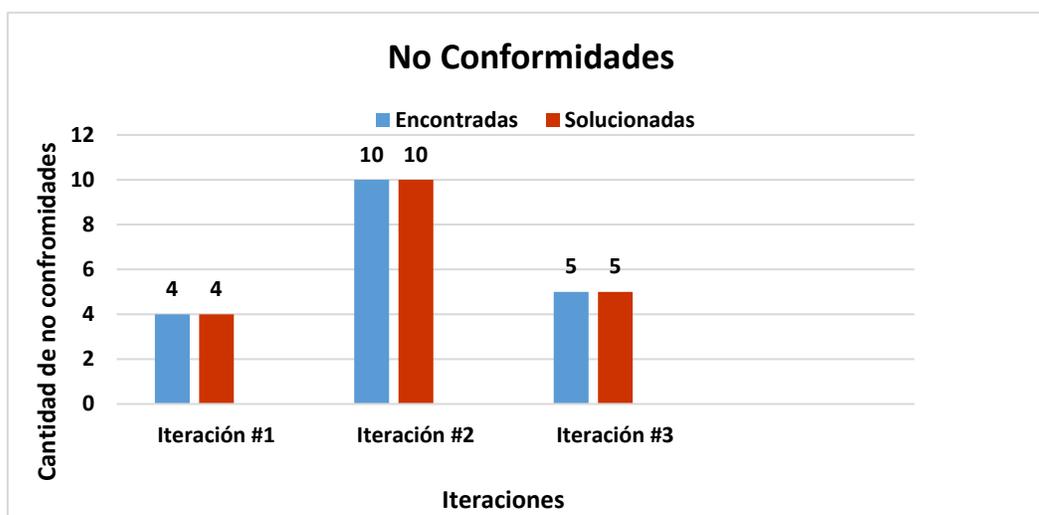


Figura 3. 5 No conformidades detectadas durante las iteraciones.

Conclusiones parciales

El desglose de las historias de usuario en tareas de ingeniería aumentó la comprensión de las funcionalidades a desarrollar. El uso de los estándares de codificación definidos permitió desarrollar un código fácil de comprender por los desarrolladores. La representación del diagrama de componentes, el diagrama de despliegue y las interfaces de usuario permitió una mejor visualización de la estructura y funcionamiento de la herramienta. La ejecución de varias iteraciones de pruebas permitió detectar y corregir varias no conformidades durante en el proceso de desarrollo. La ejecución de las pruebas permitió validar el correcto funcionamiento de la herramienta.

Conclusiones generales

- ✓ El análisis de los referentes teórico-metodológicos facilitó la comprensión de los procesos de reconocimiento de personas mediante huellas dactilares.
- ✓ La caracterización de las herramientas de cotejo visual analizadas, posibilitaron la detección de un conjunto de funcionalidades necesarias para facilitar el proceso de cotejo visual de impresiones dactilares a especialistas periciales.
- ✓ La selección de las herramientas y tecnologías que dan vida al ciclo de desarrollo de software guiado por un enfoque ágil, garantizan la creación de la herramienta dactiloscópica de acuerdo a necesidades específicas del cliente.
- ✓ La realización de las pruebas unitarias y de aceptación permitió comprobar el buen funcionamiento de la herramienta, demostrando la validez de la misma con resultados satisfactorios.
- ✓ El resultado de las entrevistas realizadas a especialistas periciales permitió, inicialmente detectar un conjunto de funcionalidades imprescindibles para facilitar el proceso de cotejo visual de impresiones dactilares y posteriormente, validó el correcto funcionamiento de cada una de ellas en la herramienta desarrollada.

Recomendaciones

- ✓ Desarrollar e incorporar a la herramienta nuevos algoritmos de extracción de puntos característicos y puntos singulares.
- ✓ Desarrollar e incorporar a la herramienta nuevos algoritmos de cotejo automático de huellas dactilares.
- ✓ Implementar métodos de conexión con sistemas externos.
- ✓ Realizar otras investigaciones para continuar el desarrollo de funcionalidades que faciliten el trabajo pericial.

Bibliografía referenciada

1. **Tapiador Mateos, Marino.** *Tecnologías Biométricas aplicadas a la seguridad.* Madrid : Alfaomega, 2005.
2. **Colectivo de Autores.** Dictionary.com. *afis.* [En línea] Random House, 23 de noviembre de 2012. [Citado el: 13 de febrero de 2015.] <http://dictionary.reference.com/browse/afis>.
3. **Observatorio de la Seguridad de la Información.** *Guía sobre las tecnologías biométricas.* España : Instituto Nacional de Tecnologías de la Comunicación (INTECO), Octubre 2011.
4. **Gordon, Jeffrey I.** Forensic identification using skin bacterial communities. [En línea] febrero de 2010. [Citado el: 13 de febrero de 2015.] <http://www.pnas.org/content/107/14/6477.full>.
5. **Maltoni, Davide y Maio, Dario y Jain, Anil K y Prabhakar, Salil.** *Handbook of fingerprint recognition. Second Edition.* Londres : Springer, 2009. ISBN: 978-1-84882-253-5.
6. **Rosales Cruz, Aradí.** Clasificación de huellas digitales mediante minucias. [En línea] 28 de Abril de 2009. [Citado el: 20 de enero de 2015.] http://ccc.inaoep.mx/~esucar/Clasesmgrp/Proyectos/reporte_modelos_huellas.pdf.
7. **Martín Rodríguez, Fernando y Suárez López, Fransico Javier.** *Identificación Dactilar basada en filtros de Gabor.*
8. **Muñoz Almudena, Lindoso.** *Tesis Doctoral: Contribución al reconocimiento de huellas dactilares mediante técnicas de correlación y arquitecturas hardware para el aumento de prestaciones.* Madrid : Universidad Carlos III de Madrid. Departamento de Tecnología Electrónica, 2009.
9. **Aguilar, Gualberto.** Reconocimiento de Huellas Dactilares. [En línea] Rev. Fac.Ing. Univ, Antioquia, Diciembre de 2008. [Citado el: 2 de febrero de 2015.] http://jaibana.udea.edu.co/grupos/revista/revistas/nro046/46_10.pdf.
10. **Colectivo de Autores.** Definición de. [En línea] [Citado el: 20 de febrero de 2015.] <http://definicion.de/cotejo/>.
11. **Tomas Pérez, Carlos y Barada Entenza, Lazara y Portela Falgueras, Armando.** *Glosario de Términos Dermatoscópicos.* Habana : s.n., 2007.
12. **Komarinski, P.** *Automated Fingerprint Identification Systems (AFIS).* New York : Elsevier Academic Press, 2005.
13. **Subcomité de, Biometría.** biometria.gov.ar. [En línea] Agosto de 2006. [Citado el: 3 de Febrero de 2015.] <http://www.biometria.gov.ar/metodos-biometricos/dactilar.aspx>.
14. **Salinas Barrera, Virginia y de la Cruz Martínez, Christian A.** *Estudio de estándares aplicables a sistemas biométricos.* México : Escuela Superior de Ingeniería Mecánica y Eléctrica, junio de 2009.

15. **SPEK Forense.** ImaQuest Fingerprint Enhancement Software. [En línea] [Citado el: 20 de Febrero de 2015.] <http://www.spexforensics.com/applications/imaquest-fingerprint-enhancement-software>.
16. **iSYS.** CSIpix. [En línea] [Citado el: 20 de Febrero de 2015.] <http://www.csipix.com/products/csipix-products/>.
17. **Datys.** Sistema automatizado de Identificación civil. [En línea] Datys, 2011. [Citado el: 14 de Febrero de 2015.] <http://www.datys.cu/wpinfo/producto.aspx?22>.
18. **García Borroto, Milton, y otros.** *Sistema Automatizado de Identificación De Huellas e Impresiones Palmares (SAIDIP)*. Ciego de Ávila : s.n., 2010.
19. **Figueroa, Roberth G. y Solís, Camilo J. y Cabrera, Armando A.** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] [Citado el: 14 de Febrero de 2015.] <http://www.google.com.cu/url?sa=t&rct=j&q=metodologias+de+desarrollo+de+software+pesadas&source=web&cd=6&ved=0CEkQFjAF&url=http%3A%2F%2Fadonisnet.files.wordpress.com%2F2008%2F06%2Farticulo-metodologia-de-sw-formato.doc&ei=bTn0ULbfJ0qH0QHcmIDACg&usg=AFQjCN>.
20. **Kruchten, Philippe.** *The Rational Unified Process An Introduction*. s.l. : Addison Wesley, 2001.
21. **Letelier, Patricio.** Rational Unified Process (RUP). [En línea] [Citado el: 15 de febrero de 2015.] http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/RUP/RUP.pdf.
22. **Sommerville, Ian.** *Procesos del software. Ingeniería del Software 7ma Edición*. 2005. págs. 76-78.
23. **Kroll, Per y Kruchten, Philippe.** *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. s.l. : Addison-Wesley Professional, 2003.
24. **Letelier, Patricio y Penadés, M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España : Universidad Politécnica de Valencia.
25. **Schwaber, Ken y Beedle, Mike.** *The Agile software development with Scrum*. s.l. : Prentice Hall, 2008.
26. **Schwaber, Ken.** *Agile Project Management with Scrum*. s.l. : Microsoft Press, 2008.
27. **IBM.** Unified Modeling Language (UML). [En línea] <http://www-01.ibm.com/software/rational/uml/>.
28. **McClure, Carma.** *The CASE Experience*. s.l. : BYTE, Abril 1989.
29. **Visual Paradigm.** [En línea] Enero de 2013. [Citado el: 16 de Febrero de 2015.] <http://www.visual-paradigm.com/>.
30. **desarrolloweb.com.** [En línea] [Citado el: 1 de Marzo de 2015.] <http://www.desarrolloweb.com/articulos/1328.php>.

31. **Microsoft.** Introducción al lenguaje C# y .NET Framework. *msdn*. [En línea] Diciembre de 2012. [Citado el: 17 de Febrero de 2015.] <http://msdn.microsoft.com/es-es/library/z1zx9t92%28v=VS.80%29.aspx>.
32. —. Visual Studio 2013. *Visual Studio*. [En línea] Diciembre de 2013. [Citado el: 18 de Febrero de 2015.] <http://www.visualstudio.com/es-es>.
33. —. *msdn*. [En línea] [Citado el: 1 de Marzo de 2015.] <https://msdn.microsoft.com/es-es/library/ms754130%28v=vs.110%29.aspx>.
34. —. *msdn*. [En línea] [Citado el: 1 de Marzo de 2015.] <https://msdn.microsoft.com/es-es/library/dd460648%28v=vs.110%29.aspx>.
35. **Sánchez Asenjo, Jorge.** Apuntes Completos sobre Sistemas gestores de Bases de Datos. [En línea] 2009. [Citado el: 17 de Febrero de 2015.] <http://ubuntuone.com/p/sqt/ubuntuone.pdf>.
36. **Martinez, Rafael.** Sobre PostgreSQL. [En línea] [Citado el: 20 de Febrero de 2015.] http://www.postgresql.org/es/sobre_postgresql.
37. **The PostgreSQL Global Development Group.** PostgreSQL. *PostgreSQL Featured Users*. [En línea] 2013. [Citado el: 17 de Febrero de 2015.] <http://www.postgresql.org/about/users/>.
38. **Riggs, Simon y Krosing, Hannu.** *PostgreSQL 9 Administration Cookbook*. Birmingham : Packt Publishing, 2010. pág. 8.
39. **Dr.Gabriel, Nicolas Schenker.** NHibernate 3. [En línea] [Citado el: 1 de Marzo de 2015.]
40. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999.
41. **Hilliard, Rich.** *IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems*. noviembre del 2000.
42. **Joskowicz, Ing. José.** *Reglas y Prácticas en eXtreme Programming*. España : Doctorado de Ingeniería Telemática de la Universidad, 2008.

Bibliografía consultada

- Tapiador Mateos, Marino.** *Tecnologías Biométricas aplicadas a la seguridad.* Madrid : Alfaomega, 2005.
- Observatorio de la Seguridad de la Información.** *Guía sobre las tecnologías biométricas.* España : Instituto Nacional de Tecnologías de la Comunicación (INTECO), Octubre 2011.
- Gordon, Jeffrey I.** Forensic identification using skin bacterial communities. [En línea] febrero de 2010. <http://www.pnas.org/content/107/14/6477.full>.
- Maltoni, Davide y Maio, Dario y Jain, Anil K y Prabhakar, Salil.** *Handbook of fingerprint recognition. Second Edition.* Londres : Springer, 2009. ISBN: 978-1-84882-253-5.
- Rosales Cruz, Aradí.** Clasificación de huellas digitales mediante minucias. [En línea] 28 de Abril de 2009. http://ccc.inaoep.mx/~esucar/Clasesmgp/Proyectos/reporte_modelos_huellas.pdf.
- Martín Rodríguez, Fernando y Suárez López, Fransico Javier.** *Identificación Dactilar basada en filtros de Gabor.*
- Muñoz Almudena, Lindoso.** *Tesis Doctoral: Contribución al reconocimiento de huellas dactilares mediante técnicas de correlación y arquitecturas hardware para el aumento de prestaciones.* Madrid : Universidad Carlos III de Madrid. Departamento de Tecnología Electrónica, 2009.
- Aguilar, Gualberto.** Reconocimiento de Huellas Dactilares. [En línea] Rev. Fac.Ing. Univ, Antioquia, Diciembre de 2008. http://jaibana.udea.edu.co/grupos/revista/revistas/nro046/46_10.pdf.
- Tomas Pérez, Carlos y Barada Entenza, Lazara y Portela Falgueras, Armando.** *Glosario de Términos Dermatoscópicos.* Habana : s.n., 2007.
- Komarinski, P.** *Automated Fingerprint Identification Systems (AFIS).* New York : Elsevier Academic Press, 2005.
- Salinas Barrera, Virginia y de la Cruz Martínez, Christian A.** *Estudio de estándares aplicables a sistemas biométricos.* México : Escuela Superior de Ingeniería Mecánica y Eléctrica, junio de 2009.
- García Borroto, Milton, y otros.** *Sistema Automatizado de Identificación De Huellas e Impresiones Palmares (SAIDIP).* Ciego de Ávila : s.n., 2010.
- Gunnerson, Eric.** *A Programmer's Introduction to C#.* United States on America, 2000.
- Figuroa, Roberth G. y Solís, Camilo J. y Cabrera, Armando A.** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] <http://www.google.com/cu/url?sa=t&rct=j&q=metodologias+de+desarrollo+de+software+pesadas&source=web&cd=6&ved=0CEkQFjAF&url=http%3A%2F%2Fadonisnet.files.wordpress.com%2F2008%2F06%2Farticulo-metodologia-de-sw-formato.doc&ei=bTn0ULbfJ0qH0QHcmIDACg&usg=AFQjCN>.
- Kruchten, Philippe.** *The Rational UnifiedProcess An Introduction.* s.l. : Addison Wesley, 2001.

Letelier, Patricio. Rational Unified Process (RUP). [En línea] http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/RUP/RUP.pdf.

Cooper, James. *Introduction to Design Patterns in C#*. United States on America, 2002.

Sommerville, Ian. *Procesos del software. Ingeniería del Software 7ma Edición*. 2005. págs. 76-78.

Kroll, Per y Kruchten, Philippe. *The Rational Unified Process Made Easy: A Practitioner's Guide to the Rup*. s.l. : Addison-Wesley Professional, 2003.

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

Schwaber, Ken y Beedle, Mike. *The Agile software development with Scrum*. s.l. : Prentice Hall, 2008.

Schwaber, Ken. *Agile Project Management with Scrum*. s.l. : Microsoft Press, 2008.

McClure, Carma. *The CASE Experience*. s.l. : BYTE, Abril 1989.

MacDonald, Matthew. *Windows Presentation Foundation in .NET 4.0*. United States on America, 2010.

Raghuram, Ranga y Reid, Jon. *Beginning C# 2008 Databases From Novice to Professional*. 2008.

Sánchez Asenjo, Jorge. Apuntes Completos sobre Sistemas gestores de Bases de Datos. [En línea] 2009. <http://ubuntuone.com/p/sqt/ubuntuone.pdf>.

Riggs, Simon y Krosing, Hannu. *PostgreSQL 9 Administration Cookbook*. Birmingham : Packt Publishing, 2010.

Dr. Gabriel, Nicolas Schenker. NHibernate 3.

Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999.

Nathan, Adams. *WPF 4 unleashed*. United States on America, Junio 2010.

Hilliard, Rich. *IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems*. noviembre del 2000.

Joskowicz, Ing. José. *Reglas y Prácticas en eXtreme Programming*. España : Doctorado de Ingeniería Telemática de la Universidad, 2008.

Clark, Dan. *Beginning C# Object-Oriented Programming*. United States on America, Abril 2013.

Glosario de términos

AFIS Sistema Automático de Identificación de Huellas Dactilares (Traducido de las siglas en inglés AFIS: Automatic Fingerprint Identification System).

ANSI Instituto Nacional Estadounidense de Estándares (De sus siglas en inglés ANSI: American National Standards Institute).

CASE Ingeniería de Software Asistida por Computadora (Traducido de las siglas en inglés CASE: Computer Aided Software Engineering).

CISED Centro de Identificación y Seguridad Digital.

CPU Unidad Central de Procesamiento.

CRC Clase-Responsabilidad-Colaboración.

DATYS Empresa de Desarrollo de Aplicaciones, Tecnologías y Sistemas.

FBI Buró Federal de Investigaciones (Traducido de sus siglas en inglés FBI: Federal Bureau of Investigation).

GRASP Patrones Generales de Software para Asignar Responsabilidades.

GOF Banda de los Cuatro (Traducido de las siglas en inglés GUI: Gang of Four).

IDE Entorno Integrado de Desarrollo (Traducido de las siglas en inglés IDE: Integrated Development Environment).

IEEE Instituto de Ingenieros Eléctricos y Electrónicos (Traducido de las siglas en inglés IEEE: (Institute of Electrical and Electronics Engineers).

ISO/IEC Sub-comité 17 (SC17) de la Comisión Técnica Mixta sobre Tecnologías de la Información (De su traducción en inglés: Joint Technical Committee on Information Technology (ISO/IEC JTC1), de la Organización Internacional para la Estandarización, de su traducción inglesa: International Organization for Standardization (ISO) y la Comisión Electrotécnica Internacional (International Electrotechnical Commission, IEC). A nivel mundial, es el principal organismo que coordina las actividades de

estandarización biométrica). Estándar ISO 19794-2: creado en 2005 por la ISO/IEC con propósitos similares a la norma ANSI 378, respecto a la que guarda mucha similitud Subcommittee on Biometrics (2006).

NIST Instituto Nacional de Normas y Tecnología. Es una agencia de la Administración de Tecnología del Departamento de Comercio de los Estados Unidos. Traducido de las siglas en inglés NIST: National Institute of Standards and Technology.

RAM Memoria de Acceso Aleatorio.

SGBD Sistema de Gestor de Base de Datos.

TIC Tecnología de la Información y las Comunicaciones.

UCI Universidad de las Ciencias Informáticas.

UML Lenguaje de Modelado Unificado (Traducido de las siglas en inglés UML: Unified Modeling Language).

XP Programación Extrema (Traducido de las siglas en inglés XP: Extreme Programming).

Anexos

Anexo 1: Preguntas de la entrevista.

Entrevistado: 1erTte MsC Yóskel Arcia Matos que ejerce como perito criminalístico y especialista en dermatoscopia del laboratorio provincial de criminalística de la provincia de Ciego de Ávila cuenta con 12 años de experiencia en el tema.

Entrevista realizada al comienzo de la investigación

1. ¿Cuáles son los pasos a seguir para realizar el proceso de cotejo visual de huellas dactilares?

Respuesta: Para realizar el proceso de cotejo visual de huellas dactilares lo primero es comprobar que las huellas dactilares que se están cotejando presenten el mismo tipo de dibujo papilar, este paso resulta muy sencillo de verificar ya que a simple vista se puede determinar el tipo de dibujo que presenta cada huella. En el caso de que el tipo de dibujo papilar de las dos huellas dactilares que se están cotejando sea el mismo se aplican una serie de técnicas para comparar las características específicas que presentan cada una de estas huellas.

2. ¿Cuáles son y en qué consisten estas técnicas a utilizar en la comparación de las características específicas de las huellas dactilares?

Respuesta: Las técnicas que se utilizan en este paso son:

- ✓ La detección de los puntos característicos (minucias) y de los puntos singulares (núcleo y delta) esta técnica consiste en detectar y señalar la presencia de estos puntos en la huella dactilar.
- ✓ El conteo de puntos característicos esta técnica consiste en realizar un conteo de la cantidad de minucias que sean detectadas en la huella dactilar.
- ✓ La medición de distancia entre puntos esta técnica consiste en realizar una medición para obtener la distancia que existe entre dos puntos que hayan sido seleccionados por el especialista
- ✓ El conteo de crestas esta técnica consiste en realizar un conteo de las crestas que estén presentes en la huella dactilar.

- ✓ La superposición de impresiones esta técnica consiste en superponer las dos huellas dactilares que se están cotejando.

3. ¿Tienen conocimiento de alguna herramienta que permita realizar el cotejo visual de huellas dactilares?

Respuesta: Tengo conocimiento de dos herramientas desarrolladas en el país que permiten realizar el proceso de cotejo visual de huellas dactilares, una de ellas es la usada por los especialistas periciales del Ministerio del Interior que esta herramienta viene integrada en un módulo de BIOMESYS AFIS y la otra herramienta que conozco es la desarrollada en el Centro de Bioplasmas de la provincia de Ciego de Ávila.

4. ¿En qué lenguaje de programación están implementadas la mismas?

Respuestas: Las dos están implementadas en C#.

5. ¿Estas herramientas cuentan con todas las funcionalidades necesarias para realizar el proceso de cotejo visual de huellas dactilares?

Respuestas: Estas herramientas no cuentan con todas las funcionalidades necesarias para realizar el proceso de cotejo visual de huellas dactilares, esto propicia que se nos dificulte realizar este proceso.

6. ¿Qué funcionalidades consideras se deban incluir en una herramienta de este tipo para que sea más fácil realizar el proceso de cotejo visual de huellas dactilares?

Respuestas: Se deberían incluir un grupo de funcionalidades que permitan realizar este proceso de forma más fácil y amena para el especialista como son: que tanto el conteo de los puntos característicos como el de las crestas papilares se realice de forma automática, que se permita medir distancia entre puntos de la huella y se muestre el valor de esta medición, que se puedan superponer las imágenes de las huellas dactilares, que se permita realizar extracción tanto manual como automática de los puntos característicos y de los puntos singulares que estén presente en la huella dactilar.

Entrevista realizada luego de mostrarle la herramienta desarrollada

1. ¿Qué opina de la herramienta dactiloscópica presentada?

Respuesta: Considero que la herramienta dactiloscópica que me presentan está muy acorde al objetivo con que fue desarrollada, presentando un gran cúmulo de funcionalidades que satisfacen a cabalidad las necesidades de los especialistas periciales, con el uso de esta herramienta se nos facilitará el trabajo a la hora de realizar el proceso de cotejo visual de huellas dactilares.

2. Un grupo de las funcionalidades que presenta la herramienta fueron implementadas teniendo en cuenta las sugerencias que nos hizo al comenzar esta investigación. ¿Qué consideración tiene de estas funcionalidades?

Respuesta: Considero que las sugerencias que les hice están reflejadas en las funcionalidades implementadas, la interacción con las funcionalidades de la herramienta me resulta amena ya que las mismas están bien distribuidas en la interfaz visual.

Anexo 2: Descripción de las historias de usuarios.

Historia de Usuario	
Número: HU_1	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Autenticar usuario	
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: Todos los usuarios habilitados tendrán la posibilidad de identificarse en el sistema.	
Observaciones: Si el sistema no identifica el usuario este mostrará un mensaje comunicando que sus identificadores no son válidos.	

Tabla A2. 1 HU Autenticar Usuario.

Historia de Usuario	
Número: HU_2	Usuario: Administrador
Nombre historia: Gestionar Usuario.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El administrador tendrá la posibilidad de gestionar los usuarios que operarán en el sistema, lo cual incluyen el rol y los datos personales del mismo.	
Observaciones: Gestionar usuario incluye adicionar, listar, modificar y eliminar usuarios.	

Tabla A2. 2 HU Gestionar Usuario.

Historia de Usuario	
Número: HU_3	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Cargar imágenes de las huellas dactilares.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá cargar la imagen de la huella dactilar que desea comparar y la lista de imágenes de huellas dactilares con las que va a realizar la comparación.	
Observaciones: Las imágenes de las huellas dactilares podrán ser cargadas desde un directorio local.	

Tabla A2. 3 HU Cargar imágenes de las huellas dactilares.

Historia de Usuario	
Número: HU_4	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Cargar algoritmos biométricos.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: La herramienta cargará los algoritmos de extracción de características y cotejo automático de huellas dactilares desde un directorio donde se encuentran las bibliotecas de clases en las que están implementados.	
Observaciones: El usuario podrá seleccionar de los algoritmos de extracción de características y cotejo automático de huellas dactilares que hayan sido cargados por la herramienta los que desee utilizar en la comparación.	

Tabla A2. 4 HU Cargar algoritmos biométricos.

Historia de Usuario	
Número: HU_6	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Detectar puntos característicos y puntos singulares de forma automática.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: Se mostrarán en la imagen de la huella los puntos característicos y puntos singulares que esta presenta, dichos puntos se obtendrán mediante la aplicación de un algoritmo de extracción de características previamente seleccionado por el usuario.	
Observaciones: Los puntos característicos y puntos singulares que podrá detectar de forma automática son: terminación, bifurcación, núcleo y delta.	

Tabla A2. 5 HU Detectar puntos característicos y puntos singulares de forma automática.

Historia de Usuario	
Número: HU_7	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Medir distancia entre puntos.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá medir la distancia entre los puntos presentes en la imagen de la huella dactilar.	
Observaciones:	

Tabla A2. 6 HU Medir distancia entre puntos.

Historia de Usuario	
Número: HU_8	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Contar puntos característicos.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: La herramienta se encargará de realizar un conteo automático de los puntos característicos que sean identificados.	
Observaciones:	

Tabla A2. 7 HU Contar puntos característicos.

Historia de Usuario	
Número: HU_9	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Contar crestas papilares.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: La herramienta se encargará de realizar el conteo automático de las crestas papilares de la huella dactilar, haciendo uso de un algoritmo de extracción y conteo de crestas papilares.	
Observaciones:	

Tabla A2. 8 HU Contar crestas papilares.

Historia de Usuario	
Número: HU_10	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Cotejo automático de huellas.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá realizar un cotejo automático de las huellas que se quieran comparar.	
Observaciones:	

Tabla A2. 9 HU Cotejo automático de huellas.

Historia de Usuario	
Número: HU_11	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Superponer imágenes de huellas dactilares.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá superponer las imágenes de las huellas dactilares que se quieran comparar.	
Observaciones:	

Tabla A2. 10 HU Superponer imágenes de huellas dactilares.

Historia de Usuario	
Número: HU_12	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Aplicar transformaciones a la imagen de la huella dactilar.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario podrá realizar las transformaciones que estime necesarias en las imágenes de huellas dactilares que se vayan a comparar.	
Observaciones: Transformaciones a la imagen de la huella dactilar incluye mover y rotar la imagen, modificar tamaño de la imagen, modificar brillo de la imagen, modificar contraste de la imagen, invertir la imagen y llevar la imagen a escala de grises.	

Tabla A2. 11 HU Aplicar transformaciones a la imagen de la huella dactilar.

Historia de Usuario	
Número: HU_13	Usuario: Todos los usuarios habilitados en el sistema.
Nombre historia: Gestionar comparaciones.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Alexis García Rojas y Adalberto Hernández Valdés	
Descripción: El usuario tendrá la posibilidad de gestionar las comparaciones que se realicen en la herramienta.	
Observaciones: Gestionar comparaciones incluye guardar, listar, modificar y eliminar comparaciones.	

Tabla A2. 12 HU Gestionar comparaciones.

Anexo 3: Descripción de las tarjetas CRC.

Conf	
Responsabilidades	Colaboradores
Contiene los datos de una configuración.	-

Tabla A3. 1 Tarjeta CRC de la clase Conf.

MefAux	
Responsabilidades	Colaboradores
Crear un catálogo de algoritmos biométricos.	✓ Configuracion

Tabla A3. 2 Tarjeta CRC de la clase MefAux.

NegocioImpl	
Responsabilidades	Colaboradores
Ejecuta las consultas a la Base de Datos.	✓ RepositoryFactory

Tabla A3. 3 Tarjeta CRC de la clase NegocioImpl.

Minucia	
Responsabilidades	Colaboradores
Contiene los datos de una minucia.	✓ MinuciaTipo

Tabla A3. 4 Tarjeta CRC de la clase Minucia.

MinuciaTipo	
Responsabilidades	Colaboradores
Enumerativo que establece los tipos de minucias.	-

Tabla A3. 5 Tarjeta CRC de la clase MinuciaTipo.

Controlador	
Responsabilidades	Colaboradores
Invertir imagen de la huella dactilar.	✓ CotejoLista
Realizar escala de gris a la imagen de la huella dactilar.	✓ CotejoLista
Modificar brillo y contrastes de la imagen de la huella dactilar.	✓ CotejoLista

Tabla A3. 6 Tarjeta CRC de la clase Controlador.

PuntoLineas	
Responsabilidades	Colaboradores
Contiene los datos de una línea.	-

Tabla A3. 7 Tarjeta CRC de la clase PuntoLineas.

dComparacion	
Responsabilidades	Colaboradores
Contiene los datos de una comparación.	-

Tabla A3. 8 Tarjeta CRC de la clase dComparacion.

Util	
Responsabilidades	Colaboradores
Guarda una configuración.	-
Carga una configuración.	-

Tabla A3. 9 Tarjeta CRC de la clase Util.

Anexo 4: Descripción de las entidades.

Tabla usuario: Guarda los datos de los usuarios de la herramienta.

Atributos	Tipo de Datos	Nulo	Definición
Id_usuario	<i>serial</i>	no	Identificador del usuario
nombre	<i>varchar</i>	no	Nombre completo del usuario
nombre_usuario	<i>varchar</i>	no	Nombre de usuario
contrasena	<i>varchar</i>	no	Contraseña del usuario
Id_rol (FK)	<i>serial</i>	no	Identificador del rol al que pertenece el usuario

Tabla A4. 1 Descripción de la entidad usuario.

Tabla rol: Guarda los roles a los que puede pertenecer un usuario.

Atributos	Tipo de Datos	Nulo	Definición
Id_rol	<i>serial</i>	no	Identificador del rol
nombre_rol	<i>serial</i>	no	Nombre del rol

Tabla A4. 2 Descripción de la entidad rol.

Tabla permiso: Almacena el nombre de los recursos a los que pueden acceder los usuarios.

Atributos	Tipo de Datos	Nulo	Definición
Id_permiso	<i>serial</i>	no	Identificador del permiso
nombre_recurso	<i>serial</i>	no	Nombre del recurso

Tabla A4. 3 Descripción de la entidad permiso.

Tabla rol_permiso: Se encarga de asociar los roles y los permisos.

Atributos	Tipo de Datos	Nulo	Definición
Id_permiso (fk)	<i>serial</i>	no	Identificador del permiso
Id_rol (fk)	<i>serial</i>	no	Identificador del rol

Tabla A4. 4 Descripción de la entidad rol_permiso.

Tabla comparacion: Guarda los datos de las comparaciones que se realizan en la herramienta.

Atributos	Tipo de Datos	Nulo	Definición
Id_comparacion	<i>serial</i>	no	Identificador de comparación
Id_huella1	<i>serial</i>	no	Identificador de la huella uno de la comparación
Id_huella2	<i>serial</i>	no	Identificador de la huella dos de la comparación
resultado	<i>varchar</i>	no	Almacena el resultado de la comparación
comentario	<i>varchar</i>	no	Almacena la observación del especialista que realiza la comparación
nombre	<i>varchar</i>	no	Nombre de la persona a la que pertenece la imagen de la huella conocida
fecha	<i>timestamp</i>	no	Fecha en que se realizó la comparación
Id_usuario (fk)	<i>serial</i>	no	Identificador del usuario que realizó la comparación

Tabla A4. 5 Descripción de la entidad comparacion.

Tabla huella: Almacena los datos de las huellas que forman parte de las comparaciones guardadas en la herramienta.

Atributos	Tipo de Datos	Nulo	Definición
Id_huella	<i>serial</i>	no	Identificador de la huella
direccion	<i>varchar</i>	no	Dirección donde se encuentra la imagen de la huella dactilar
jsonminucia	<i>text</i>	no	Almacena la lista de minucias
jsonlinea	<i>text</i>	no	Almacena la lista de líneas
cantcrestas	<i>int</i>	no	Cantidad de crestas que contiene la imagen de la huella dactilar

Tabla A4.6 Descripción de la entidad huella.

Anexo 5: Tareas de ingeniería.

Iteración 1	
Historia de Usuario	Tareas
Autenticar usuario.	<ol style="list-style-type: none"> 1. Validar campos de entrada. 2. Cifrar contraseña. 3. Verificar autenticación del usuario. 4. Mostrar mensaje de autenticación correcta o incorrecta en el sistema.
Gestionar usuario.	<p style="text-align: center;">Adicionar usuario</p> <ol style="list-style-type: none"> 1. Validar campos de entrada. 2. Mostrar mensaje de enrolamiento. <p style="text-align: center;">Listar usuario</p> <ol style="list-style-type: none"> 1. Mostrar usuarios existentes. <p style="text-align: center;">Eliminar usuario</p> <ol style="list-style-type: none"> 1. Seleccionar el usuario que desea eliminar. 2. Eliminar el usuario de la base de datos. 3. Mostrar mensaje de usuario eliminado correctamente. <p style="text-align: center;">Modificar usuario</p> <ol style="list-style-type: none"> 1. Seleccionar el usuario a modificar. 2. Validar campos de entrada. 3. Modificar los datos del usuario en la base de datos. 4. Mostrar mensaje modificación efectuada con éxito.
Cargar imágenes de las huellas dactilares.	<p style="text-align: center;">Cargar imagen de la huella dactilar a comparar</p> <ol style="list-style-type: none"> 1. Buscar directorio donde se encuentra la imagen. 2. Mostrar la imagen seleccionada en pantalla. <p style="text-align: center;">Cargar imágenes de las huellas dactilares candidatas</p> <ol style="list-style-type: none"> 1. Buscar una o varias imágenes de huellas desde un directorio. 2. Mostrar las huellas seleccionadas.
Cargar algoritmos biométricos.	<ol style="list-style-type: none"> 1. Crear un catálogo con las bibliotecas de clases. 2. Agregar el catálogo a un contenedor de catálogos. 3. Mostrar en la interfaz los algoritmos existentes en el catálogo.

Tabla A5. 1 Tareas de ingeniería de la primera iteración.

Iteración 3	
Historia de Usuario	Tareas
Aplicar transformaciones a la imagen de la huella dactilar.	<p>Mover y rotar imagen</p> <ol style="list-style-type: none"> 1. Trabajar con la traslación y la rotación del componente que muestra la imagen. <p>Modificar tamaño de la imagen</p> <ol style="list-style-type: none"> 1. Trabajar con el escalado del componente que contiene la imagen. <p>Modificar brillo de la imagen</p> <ol style="list-style-type: none"> 1. Utilizar algoritmo para modificar el brillo de la imagen. <p>Modificar contraste de la imagen</p> <ol style="list-style-type: none"> 1. Utilizar algoritmo para modificar el contraste de la imagen. <p>Llevar la imagen a escala de grises</p> <ol style="list-style-type: none"> 1. Utilizar algoritmo para llevar la imagen a escala de grises. <p>Invertir imagen</p> <ol style="list-style-type: none"> 1. Utilizar algoritmo para invertir la imagen.
Gestionar comparaciones.	<p>Guardar comparación</p> <ol style="list-style-type: none"> 1. Crear json con las minucias pertenecientes a las huellas. 2. Almacenar en la bases de datos. <p>Listar comparaciones</p> <ol style="list-style-type: none"> 1. Realizar consulta a la base de datos y mostrar las comparaciones existentes. <p>Modificar comparación</p> <ol style="list-style-type: none"> 1. Cargar una comparación existente y permitir realizarle modificaciones. <p>Eliminar comparación</p> <ol style="list-style-type: none"> 1. Eliminar de la base de datos una comparación existente.

Tabla A5. 2 Tareas de ingeniería de la tercera iteración.

Anexo 6: Interfaces de usuario.

HDGP
Herramienta Dactiloscópica para la Gestión Pericial

Usuario: admin

Contraseña: ●●

Aceptar Cancelar

Figura A6. 1 Interfaz para autenticar usuario.

Cotejo Visual

ARCHIVO USUARIO CONFIGURACIÓN Administrador

Usuarios:

Nombre	Usuario	Rol
Administrador	admin	Administrador
Alejandro	amvalle	Especialista
Alexis Garcia Rojas	agrojass	Especialista
Adalberto Hernández Valdes	ahvaldes	Administrador

Agregar

Modificar Usuario

Nombre Completo: Alexis Garcia Rojas

Usuario: agrojass

Especialista Administrador

Aceptar Cancelar

Modificar

Eliminar

Figura A6. 2 Interfaz para la gestión de usuarios.

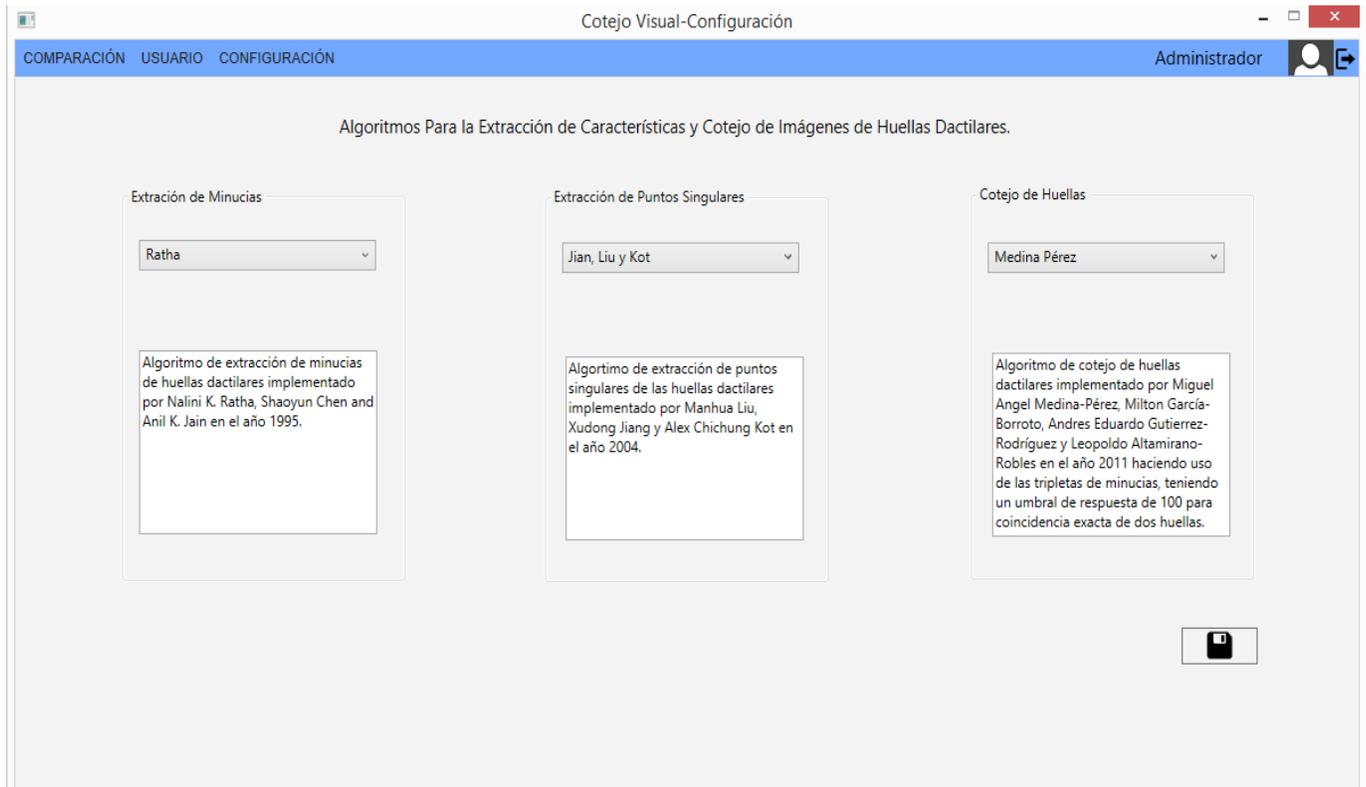


Figura A6. 3 Interfaz de configuración.

Anexo 7: Código y resultados de pruebas unitarias.

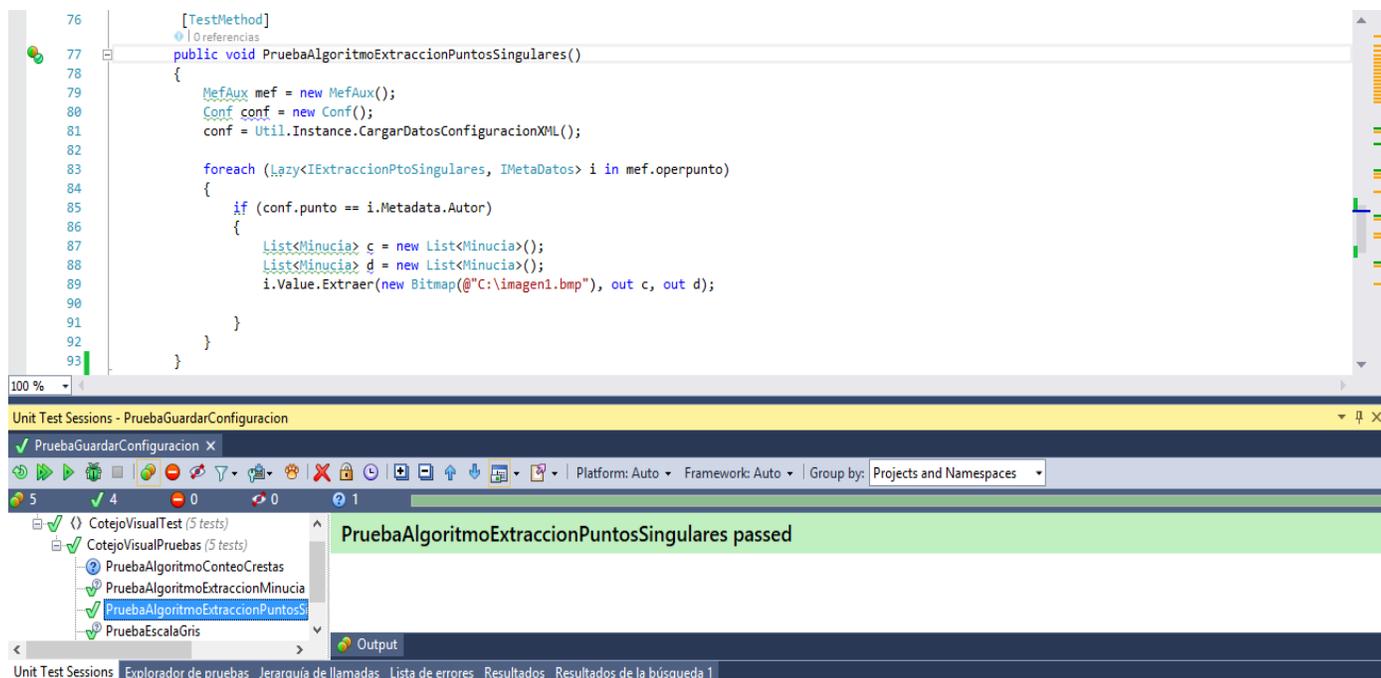


Figura A7. 1 PU Algoritmo de extracción de puntos singulares.

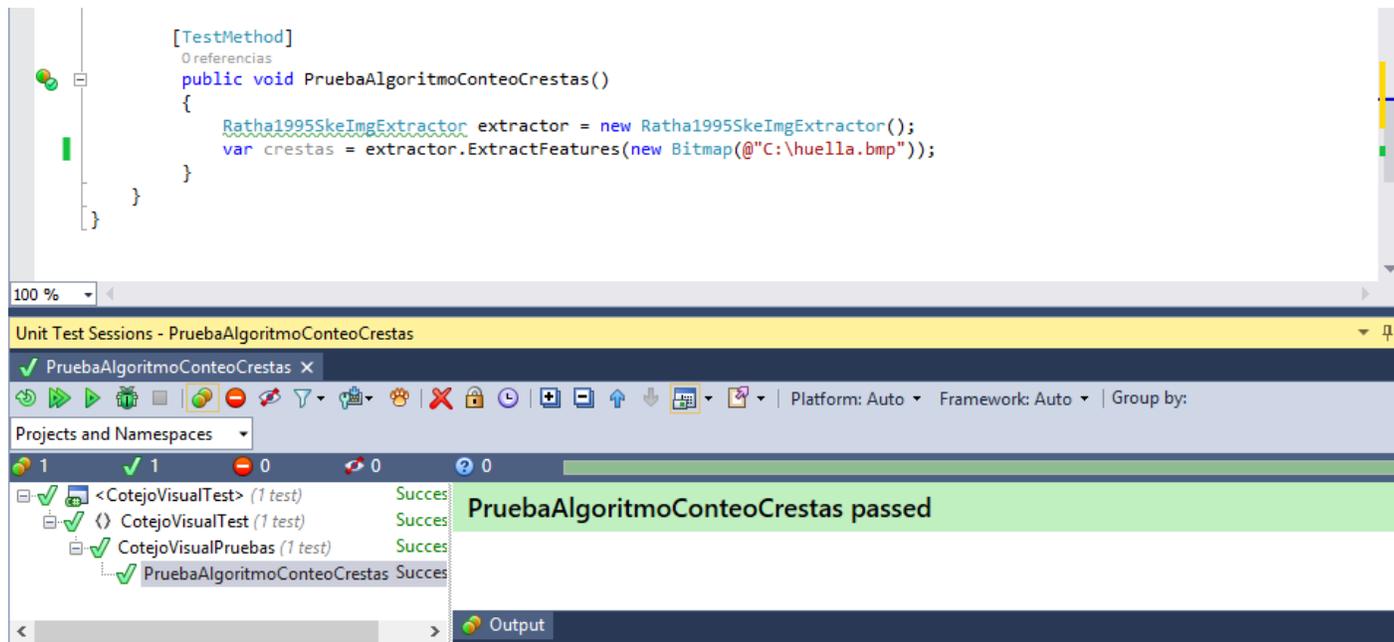


Figura A7. 2 PU Algoritmo para el conteo de crestas.

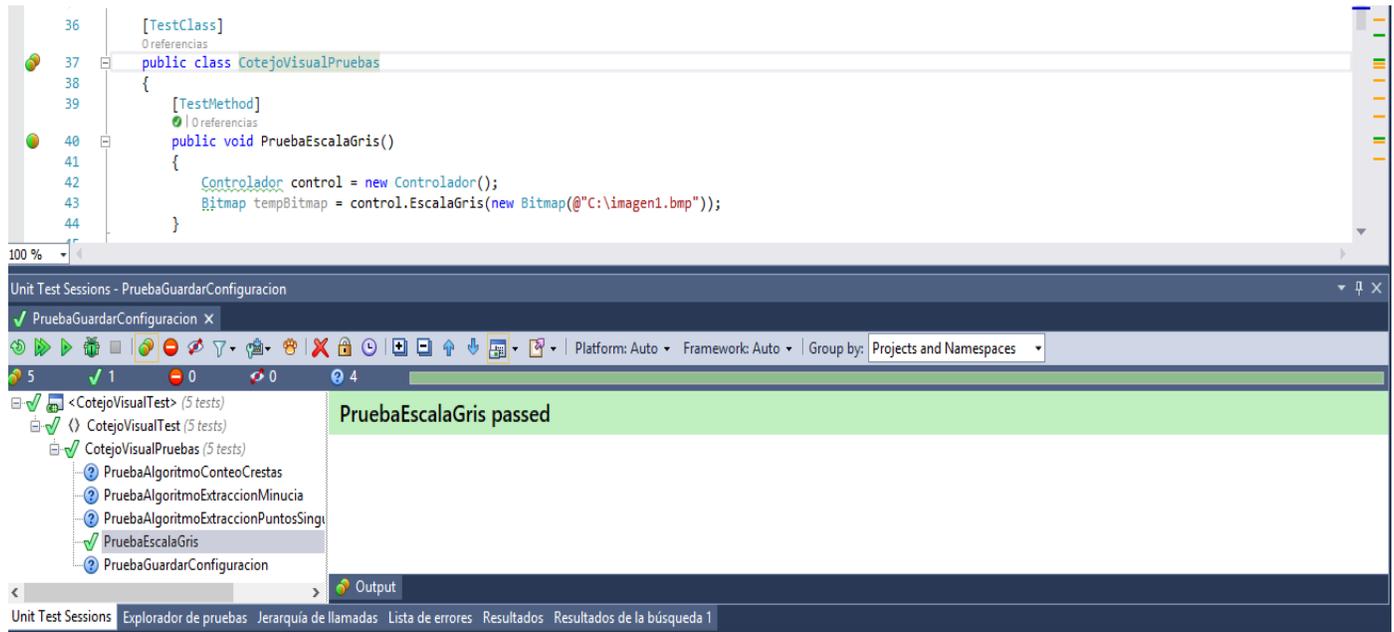


Figura A7. 3 PU Convertir a escala de gris.

Anexo 8: Casos de prueba de aceptación.

Caso de Prueba de Aceptación	
Código: CP1_HU1	HU_1: Autenticar usuario
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que un usuario se autentique en la herramienta.	
Condiciones de ejecución: El usuario debe existir.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Entrada del nombre de usuario y contraseña. 2. Validar si el usuario existe en el sistema. 3. Validar que la contraseña proporcionada coincida con la contraseña almacenada en la herramienta correspondiente al usuario. 	
Resultado esperado: Usuario autenticado en el sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 1 CP Autenticar usuario.

Caso de Prueba de Aceptación	
Código: CP1_HU2	HU_2: Gestionar usuario.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se adicione un nuevo usuario.	
Condiciones de ejecución: Haber iniciado sesión en la herramienta con un usuario administrador.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de usuarios. 2. Se selecciona la opción de adicionar usuario. 3. Se introducen los datos requeridos. 	
Resultado esperado: Se muestra en la lista de usuarios el nuevo usuario creado.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 2 CP Adicionar usuario.

Caso de Prueba de Aceptación	
Código: CP2_HU2	HU_2: Gestionar usuario.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se listen los usuario.	
Condiciones de ejecución: Haber iniciado sesión en la herramienta con un usuario administrador.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de usuarios. 2. Se selecciona la opción de listar usuario. 	
Resultado esperado: Se muestra la lista de usuarios.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 3 CP Listar usuario.

Caso de Prueba de Aceptación	
Código: CP3_HU2	HU_2: Gestionar usuario.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se elimine un usuario.	
Condiciones de ejecución: Haber iniciado sesión en la herramienta con un usuario administrador.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de usuarios. 2. Se selecciona el usuario que desee eliminar. 3. Se selecciona la opción de eliminar usuario. 	
Resultado esperado: Se elimina de la lista de usuarios el usuario seleccionado.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 4 CP Eliminar usuario.

Caso de Prueba de Aceptación	
Código: CP4_HU2	HU_2: Gestionar usuario.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se modifique un usuario.	
Condiciones de ejecución: Haber iniciado sesión en la herramienta.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de usuarios. 2. Se selecciona el usuario que desee modificar. 3. Se selecciona la opción de modificar usuario. 4. Se introducen los datos requeridos que desee modificar. 	
Resultado esperado: Se muestra en la lista de usuarios los datos modificados al usuario seleccionado.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 5 CP Modificar usuario.

Caso de Prueba de Aceptación	
Código: CP1_HU3	HU_3: Cargar imágenes de las huellas dactilares.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que la herramienta carga correctamente desde un archivo especificado por el usuario la imagen de la huella dactilar a comparar.	
Condiciones de ejecución: El usuario debe especificar la dirección de la imagen de la huella dactilar a comparar.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se selecciona la opción abrir en el área de la huella. 3. Se busca la dirección de la imagen de la huella dactilar. 	
Resultado esperado: Se muestra en la interfaz de la herramienta la imagen de la huella dactilar a comparar.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 6 CP Cargar imagen de la huella dactilar a comparar.

Caso de Prueba de Aceptación	
Código: CP2_HU3	HU_3: Cargar imágenes de las huellas dactilares.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que la herramienta carga correctamente desde un archivo especificado por el usuario las imágenes de las huellas dactilares candidatas.	
Condiciones de ejecución: El usuario debe especificar la dirección de las imágenes de las huellas dactilares candidatas.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se selecciona la opción abrir en el área de la impresión. 3. Se busca la dirección de las imágenes de las huellas dactilares candidatas. 	
Resultado esperado: Se muestra en la interfaz de la herramienta las imágenes de las huellas dactilares candidatas.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 7 CP Cargar imágenes de las huellas dactilares candidatas.

Caso de Prueba de Aceptación	
Código: CP1_HU4	HU_4: Cargar algoritmos biométricos.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que la herramienta cargue correctamente los algoritmos biométricos.	
Condiciones de ejecución: Deben existir bibliotecas de clases que contengan los algoritmos biométricos.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de configuración. 2. Se selecciona el tipo de algoritmo que desee utilizar. 	
Resultado esperado: Se muestra en la interfaz de la herramienta el algoritmo que se seleccionó y la descripción del mismo.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 8 CP Cargar algoritmos biométricos.

Caso de Prueba de Aceptación	
Código: CP1_HU7	HU_7: Medir distancia entre puntos.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se pueda medir distancia entre puntos de forma manual tanto en la imagen de la huella dactilar a comparar como en la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción de medir distancia entre puntos. 4. Se seleccionan los puntos entre los que quiere realizar la medición. 	
Resultado esperado: Se muestra una línea entre los dos puntos en que se realizó la medición y la distancia que hay entre ellos.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 9 CP Medir distancia entre puntos.

Caso de Prueba de Aceptación	
Código: CP1_HU8	HU_8: Contar puntos característicos.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se cuenten los puntos característicos encontrados manualmente o de forma automática tanto en la imagen de la huella dactilar a comparar como en la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se detectan los puntos característicos. 	
Resultado esperado: Se muestra la cantidad de puntos característicos detectados.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 10 CP Contar puntos característicos.

Caso de Prueba de Aceptación	
Código: CP1_HU9	HU_9: Contar crestas papilares.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se cuenten las crestas papilares encontradas de forma automática tanto en la imagen de la huella dactilar a comparar como en la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona la opción contar crestas. 	
Resultado esperado: Se muestra la cantidad de crestas encontradas de forma automática.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 11 CP Contar crestas papilares.

Caso de Prueba de Aceptación	
Código: CP1_HU10	HU_10: Cotejo automático de huellas.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se realice el cotejo automático entre la imagen de la huella dactilar a comparar y la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares y tener seleccionado algún algoritmo de cotejo de huellas.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona la opción cotejo automático. 	
Resultado esperado: Se muestra la respuesta que arroja el algoritmo.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 12 CP Cotejo automático de huellas.

Caso de Prueba de Aceptación	
Código: CP1_HU11	HU_11: Superponer imágenes de huellas dactilares.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se puede realizar la superposición entre la imagen de la huella dactilar a comparar y la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona la opción superponer imágenes. 	
Resultado esperado: Se muestra la superposición entre las huellas.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 13 CP Superponer imágenes de huellas dactilares.

Caso de Prueba de Aceptación	
Código: CP1_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se puedan rotar y mover tanto la imagen de la huella dactilar a comparar como la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción de mover imagen. 4. Se selecciona la imagen que se desee rotar y se realiza esta operación a través del <i>slider</i> ubicado a la derecha de la imagen seleccionada. 	
Resultado esperado: La imagen seleccionada se mueve y se rota a voluntad del usuario.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 14 CP Mover y rotar imagen.

Caso de Prueba de Aceptación	
Código: CP2_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se pueda modificar el tamaño tanto de la imagen de la huella dactilar a comparar como el de la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción para modificar el tamaño de la imagen. 	
Resultado esperado: La imagen seleccionada aumenta o disminuye su tamaño a voluntad del usuario.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 15 CP Modificar tamaño de la imagen.

Caso de Prueba de Aceptación	
Código: CP3_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se pueda modificar el brillo tanto de la imagen de la huella dactilar a comparar como el de la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción para modificar el brillo de la imagen. 	
Resultado esperado: La imagen seleccionada aumenta o disminuye el brillo a voluntad del usuario.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 16 CP Modificar brillo de la imagen.

Caso de Prueba de Aceptación	
Código: CP4_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se pueda modificar el contraste tanto de la imagen de la huella dactilar a comparar como el de la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción para modificar el contraste de la imagen. 	
Resultado esperado: La imagen seleccionada aumenta o disminuye el contraste a voluntad del usuario.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 17 CP Modificar contraste de la imagen.

Caso de Prueba de Aceptación	
Código: CP5_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se pueda llevar a escala de grises tanto la imagen de la huella dactilar a comparar como la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción de llevar la imagen a escala de grises. 	
Resultado esperado: Se muestra en escala de grises la imagen que haya sido seleccionada.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 18 CP Llevar la imagen a escala de grises.

Caso de Prueba de Aceptación	
Código: CP6_HU12	HU_12: Aplicar transformaciones a la imagen de la huella dactilar.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se pueda invertir tanto la imagen de la huella dactilar a comparar como la imagen de la huella dactilar candidata.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se selecciona en la barra de herramientas la opción de invertir imagen. 	
Resultado esperado: Se muestra la inversión de la imagen que haya sido seleccionada.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 19 CP Invertir imagen.

Caso de Prueba de Aceptación	
Código: CP1_HU13	HU_13: Gestionar comparaciones.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se guarde correctamente una comparación.	
Condiciones de ejecución: El usuario debe tener cargadas las imágenes de las huellas dactilares.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se inicia una nueva comparación. 2. Se cargan las imágenes de las huellas dactilares. 3. Se realiza la comparación. 4. Se selecciona la opción guardar comparación. 5. Se introducen los datos requeridos para guardar una comparación. 	
Resultado esperado: Se muestra un mensaje que indica que la comparación ha sido guardada correctamente.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 20 CP Guardar comparación.

Caso de Prueba de Aceptación	
Código: CP2_HU13	HU_13: Gestionar comparaciones.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se listen las comparaciones que se hayan realizado.	
Condiciones de ejecución: Debe existir alguna comparación.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de comparaciones. 	
Resultado esperado: Se muestra la lista de comparaciones.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 21 CP Listar comparaciones.

Caso de Prueba de Aceptación	
Código: CP3_HU13	HU_13: Gestionar comparaciones.
Responsable de la prueba: Alexis García Rojas.	
Descripción: Prueba para verificar que se modifique una comparación.	
Condiciones de ejecución: Debe existir alguna comparación.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de comparaciones. 2. Se selecciona la comparación que se desea modificar. 3. Se modifica la comparación. 4. Se selecciona la opción de guardar comparación. 5. Se introducen los datos requeridos para guardar la comparación. 	
Resultado esperado: Se muestra en la lista de comparaciones la comparación modificada.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 22 CP Modificar comparación.

Caso de Prueba de Aceptación	
Código: CP4_HU13	HU_13: Gestionar comparaciones.
Responsable de la prueba: Adalberto Hernández Valdés.	
Descripción: Prueba para verificar que se elimine una comparación.	
Condiciones de ejecución: Debe existir alguna comparación.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a la interfaz de gestión de comparaciones. 2. Se selecciona la comparación que se desea eliminar. 3. Se elimina la comparación. 	
Resultado esperado: Se elimina de la lista de comparaciones la comparación seleccionada.	
Evaluación de la prueba: Satisfactoria	

Tabla A8. 23 CP Eliminar comparación.