UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

Título: Componente para la protección de plantillas de minucias de huellas dactilares**.**

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

Autor: Yasmany Cruz Cordero

Tutor(es): Ing. Ramón Santana Fernández

Msc. Adrian Alberto Machado Cento

La Habana, Junio, 2016

Agradecimientos

A mi familia por su apoyo y dedicación incondicional, A todas las personas que me han ayudado de una forma u otra para lograr este sueño, a mis amigos y a todos los profesores que me impartieron clases en estos cinco años. A todos muchas gracias.

Dedicatoria

A mi querida madre con todo mi amor y esfuerzo y a toda mi familia. A mi querido país, a mis amigos, a la informática, la ciencia y el conocimiento.

Declaración de Autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yasmany Cruz Cordero	Ramón Santana Fernández	Adrián Alberto Machado Cento
_____	_____	_____
Firma del Autor	Firma del Tutor	Firma del Co-Tutor

Datos del contacto

Tutor: Ing. Ramón Santana Fernández.

Ingeniero en Ciencias Informáticas. Especialista asociado al Centro de Identificación y Seguridad Digital (CISED), graduado en el curso 2010-2011. Tiene 6 años de experiencia en el campo dedicado al reconocimiento de personas mediante huellas dactilares. Ha participado en varios eventos científicos relacionados con la biometría en la universidad, a nivel nacional e internacional obteniendo buenos resultados. Actualmente es autor de varias publicaciones asociadas a la reconstrucción de huellas dactilares y a la protección de plantillas de minucias de huellas dactilares. Actualmente es aspirante al título: Doctor en Ciencias Técnicas en la especialidad Informática.

Correo electrónico: rsfernandez@uci.cu

Tutor: Ms C. Adrián Alberto Machado Cento

Graduado de Ingeniería en Ciencias Informáticas en la UCI en el año 2006, Máster en Informática Aplicada en la UCI en el año 2008, Jefe del departamento de Desarrollo del Aplicaciones del Centro de Identificación y Seguridad Digital. Ha investigado en el área de sistemas de identificación y biometría.

Resumen

La huella dactilar como identificador biométrico es muy utilizada en el reconocimiento de personas, por lo que se ha convertido en uno de los rasgos biométricos más estudiados y utilizados. Los identificadores biométricos a diferencia de los identificadores tradicionales no pueden ser reemplazados ni cambiados. La realización de ataques informáticos en puntos de vulnerabilidad de un sistema biométrico puede traer como consecuencia la obtención de la plantilla de minucias en texto claro. El usuario puede ser rastreado en otras bases de datos en las cuales utilice el mismo identificador, lo que implica la pérdida de la seguridad de todos los recursos protegidos. Puede reconstruirse la huella dactilar y ser insertada en otros puntos del sistema biométrico para violar la seguridad del sistema y obtener acceso a los recursos protegidos.

La presente investigación tiene como objetivo general desarrollar un componente de protección de plantillas de minucias de huellas dactilares, utilizando métodos criptográficos, para ser integrado a un sistema automático de identificación mediante huellas dactilares.

Como principales resultados se obtiene un componente que realiza el cifrado de las plantillas de minucias y permite su comparación en el dominio protegido. Las pruebas realizadas demostraron un aumento en la seguridad criptográfica de las plantillas de minucias en comparación con los modelos pioneros.

Palabras Claves: AFIS, criptografía, huellas dactilares, minucias, sistemas biométricos.

Índice General

Índice de Tablas

Índice de Figuras

Introducción

El uso de métodos de reconocimiento de personas utilizando rasgos biométricos existe desde tiempos antiguos y las técnicas utilizadas han evolucionado con el paso del tiempo. A finales del siglo XIX Francis Galton publica el libro *Fingerprints*, en el cual se clasifican y describen las huellas dactilares para el reconocimiento de criminales. De esta manera se dan los primeros pasos en el uso de una característica anatómica distintiva con el fin de reconocer a una persona (Maltoni, Maio, Jain, & Salil Prabhakar, 2009).

Entre los rasgos biométricos utilizados para el reconocimiento de personas se encuentran las huellas dactilares, la geometría facial, la huella palmar, el iris, entre otros, los cuales son clasificados como rasgos físicos mientras que la voz, la forma de caminar, la firma manuscrita entre otros se clasifican como rasgos conductuales (Maltoni et al., 2009). La huella dactilar es muy utilizada como identificador biométrico en el reconocimiento de personas, debido a su facilidad de adquisición se ha convertido en uno de los rasgos biométricos más estudiados y utilizados.

La implementación de la biometría en sectores civiles y gubernamentales como instrumento de seguridad a través de redes públicas y/o privadas, ha generado mayor preocupación por la seguridad de los datos biométricos. La seguridad y privacidad de los datos biométricos transmitidos a través de redes públicas constituye un campo de investigación activo dentro de la biometría. Un identificador biométrico, como las huellas dactilares, facilita la autenticación de personas de manera inequívoca con altos niveles de seguridad (Maltoni et al., 2009).

Sin embargo, los identificadores biométricos a diferencia de los identificadores tradicionales basados en lo que se conoce o lo que se posee no puede ser reemplazado ni cambiado. La característica de la revocabilidad del identificador no es posible utilizando identificadores biométricos debido al principio de invariabilidad. La obtención de la plantilla de minucias de huellas dactilares o de cualquier otro identificador biométrico trae como principal consecuencia la pérdida del identificador para toda la vida.

Un sistema biométrico (Dahiya & Kant, 2012) es esencialmente un sistema de reconocimiento de patrones que opera a partir de la adquisición de datos biométricos de un individuo, extrae un conjunto de características de los datos capturados y las compara con las almacenadas. En dependencia del contexto puede ser utilizado para realizar verificación o identificación biométrica. El proceso de verificación valida la identidad de un individuo al comparar la muestra obtenida con la que se encuentra almacenada en la base

de datos. El proceso de identificación compara la muestra dactilar adquirida con todas las almacenadas en la base de datos. Este proceso es un componente crítico en la aplicación del reconocimiento negativo, el cual evita que una persona pueda tener múltiples identidades.

Análisis realizados por diversos autores (Dahiya & Kant, 2012; Gobi & Kannan, 2014; Jain, Nandakumar, & Nagar, 2008, 2013) han detectado ocho puntos de vulnerabilidad en la arquitectura general de un sistema biométrico en los cuales es posible obtener acceso ilegitimo a los recursos protegidos biométricamente u obtener el rasgo biométrico, como se muestra en la figura 1.

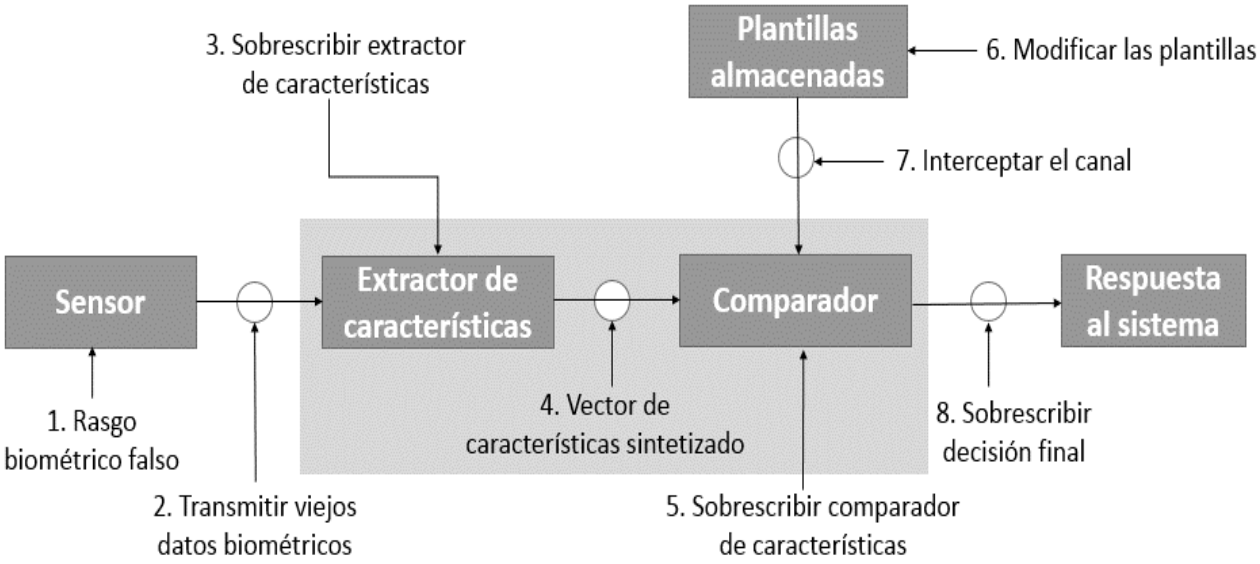

Figura 1: Principales puntos de vulnerabilidad en un sistema biométrico

Tomado de (Dahiya & Kant, 2012)

Entre las vulnerabilidades más significativas se encuentran:

1. La reescritura de los módulos de extracción y comparación: en los puntos 3 y 5 de la Figura 1 se evidencia esta vulnerabilidad que pudiera materializarse si un atacante determinado logra intercambiar dichos módulos por una copia que además de realizar su función primaria le suministrara al atacante datos de las características biométricas, de esta forma el intruso podría reconstruir el identificador biométrico o la plantilla de minucias.
2. La intercepción del canal de comunicación entre los módulos de extracción y comparación: en el punto 4 de la Figura 1 tiene lugar un flujo de datos importante, se trata de los datos biométricos

procesados y listos para comparar, si un intruso logra adquirir estos datos además de la plantilla biométrica también tendría información de los rasgos y propiedades que se utilizan para la comparación a partir de los datos biométricos originales.

3. La intercepción del canal de comunicación entre el módulo de comparación y la base de datos: si las plantillas no se encuentran protegidas pueden ser obtenidas en este canal cuando se realice una comparación determinada, ya que el sistema biométrico intenta buscar en la base de datos el rasgo coincidente consultando cada una de las plantillas existentes o una parte de ellas en el caso de que se utilice un módulo de indexado, cada una de las plantillas consultadas puede ser interceptada si se materializa esta vulnerabilidad.

4. La base de datos donde se almacenan las plantillas de minucias de huellas dactilares: en el peor de los casos si el atacante logra obtener acceso a la base de datos puede disponer del total de plantillas existentes lo cual pudiera ser catastrófico ya que en algunos sistemas incluso se guardan varias muestras para una misma persona.

La realización de ataques informáticos en estos puntos de vulnerabilidad puede traer como consecuencia la obtención de la plantilla de minucias en texto claro. El usuario puede ser rastreado en otras bases de datos en las cuales utilice el mismo identificador, lo que implica la pérdida de la seguridad de todos los recursos protegidos. Puede reconstruirse la huella dactilar como se propone en (Cappelli, Lumini, Maio, & Maltoni, 2007) y ser insertada en otros puntos del sistema biométrico para violar la seguridad del sistema y obtener acceso a los recursos protegidos.

La manera más sencilla de proteger los datos biométricos almacenados sería utilizando la criptografía clásica (Jain et al., 2013; Roja & Sawarkar, 2013) sin embargo, las propiedades de las funciones utilizadas por estos métodos dificultan el proceso de comparación de minucias en un dominio protegido. Esto se debe principalmente a que pequeños cambios en el conjunto de datos a cifrar provocan grandes cambios en el conjunto de datos cifrados. Las muestras de huellas dactilares cambian debido a varios factores como la traslación, la rotación, la superposición parcial y la deformación no lineal que experimenta el dedo al hacer contacto con una superficie. Esto implica que es necesario realizar la comparación en texto claro cuando se utiliza este tipo de funciones criptográficas para proteger plantillas de minucias de huellas dactilares, lo que constituye la **problemática** de la investigación.

Una vez analizada la problemática anterior se formula el siguiente **problema científico**:

Las plantillas de minucias de huellas dactilares son transmitidas en texto claro en los sistemas biométricos, lo que facilita su obtención mediante diferentes ataques informáticos.

Se define como **objeto de estudio** los modelos de protección de plantillas de minucias de huellas dactilares.

Como **idea a defender** se plantea que: la implementación de un componente para la protección de los datos almacenados en las plantillas de minucias disminuye la probabilidad de obtención de los datos ante diferentes tipos de ataques realizados al sistema biométrico.

Se define como **objetivo general** desarrollar un componente de protección de plantillas de minucias de huellas dactilares, utilizando métodos criptográficos, para ser integrado a los sistemas automáticos de identificación mediante huellas dactilares.

Como objetivos específicos se definen:

1. Analizar los referentes teórico-metodológicos asociados al dominio de la investigación para mejorar la comprensión del problema de investigación.
2. Realizar análisis y diseño del componente para modelar la solución propuesta.
3. Implementar el componente propuesto para el cifrado de las plantillas de minucias de huellas dactilares.
4. Validar el correcto funcionamiento del componente desarrollado mediante la realización de ataques de fuerza bruta.

Para dar cumplimiento a los objetivos planteados se definen como **tareas de investigación:**

1. Análisis del estado del arte referente a los modelos de protección de plantillas de minucias de huellas dactilares.
2. Comparación de los modelos analizados atendiendo a las vulnerabilidades encontradas.
3. Análisis de ataques realizados a componentes de protección de plantillas de minucias de huellas dactilares.
4. Análisis comparativo de los modelos de alineación de plantillas de minucias de huellas dactilares.

5. Análisis de metodologías, tecnologías y herramientas.
6. Análisis y diseño del componente propuesto.
7. Implementación del componente para la protección de plantillas de minucias de huellas dactilares.
8. Validación del componente implementado.

Con la realización del trabajo se esperan los siguientes **resultados:**

- Componente para el cifrado de las características identificativas extraídas de las huellas dactilares y que garantice que sea factible emplear en un modelo de comparación con pequeños índices de errores.
- Documentación arrojada durante el proceso de investigación e implementación del componente.

Para el desarrollo de la investigación se utilizaron diferentes métodos científicos, tanto teóricos como empíricos, que se ajustan al objeto de estudio y al cumplimiento de los objetivos trazados. Entre los que se encuentran:

Teóricos:

- **Analítico-Sintético**: el empleo del análisis-síntesis posibilitará la consulta de diversas fuentes bibliográficas, que favorecerán a conformar el estado del arte de la investigación, haciendo énfasis en el proceso de extracción de minucias en imágenes de huellas dactilares, sintetizando con los elementos más importantes; lo cual viabiliza el establecimiento de una estrecha relación con el objeto de estudio.
- **Análisis Histórico-Lógico:** la utilización de este método facilitará la compresión lógica del objeto de estudio haciendo un análisis riguroso del proceso evolutivo por el cual ha transitado la extracción de características basada en minucias, además de contribuir a dar cumplimiento de las tareas de la investigación.

Empíricos:

- **Observación:** utilizado con la finalidad de analizar los resultados que se van adquiriendo y establecer la comparación con sistemas homólogos.
- **Experimental:** método que posibilita la comprobación de las funcionalidades a medidas que estas han sido implementadas, con el objetivo de validar si cumplen o no con los requerimientos establecidos.

El presente trabajo de diploma consta de 3 capítulos estructurados de la siguiente manera:

Capítulo 1: Fundamentación teórica: constituido por el estudio del estado del arte de los métodos de protección de plantillas de minucias de huellas dactilares, se abordan elementos teóricos de la investigación y conceptos de interés con respecto al objeto de estudio; se hace énfasis en la selección de las metodologías, tecnologías y herramientas que serán utilizadas para desarrollar la solución.

Capítulo 2: Características del componente: se definen las características del componente de protección de plantillas de minucias a implementar, se realiza el levantamiento de requisitos, se definen, describen y priorizan los artefactos de ingeniería que serán confeccionados para mejorar el entendimiento del equipo de desarrollo, se determina la arquitectura, el estilo arquitectónico y los patrones utilizados en el proceso de implementación del componente.

Capítulo 3: Implementación y prueba: se describen los artefactos relacionados con la implementación y las pruebas realizadas al componente de protección de plantillas de minucias para validar su correcto funcionamiento utilizando bases de datos internacionales.

Capítulo 1: Fundamentación teórica

En el presente capítulo se definen los elementos asociados al dominio del problema. Se analiza el estado del arte de los modelos de protección de plantillas de minucias, así como las implementaciones realizadas de cada uno de ellos. Finalmente se escogen las tecnologías, metodologías y herramientas a utilizar en el desarrollo del trabajo de diploma.

1.1 Elementos asociados al dominio del problema

La identificación de personas mediante características físicas o conductuales, únicas, persistentes e invariantes en el tiempo, constituye el campo de investigación del reconocimiento biométrico (Kumar, 2014; Maltoni et al., 2009). La huella dactilar como característica identificativa es la más usada en la actualidad, encontrándose presente en el 87 por ciento de los sistemas de reconocimiento biométrico existentes en el mundo.

Sistemas biométricos

Un sistema biométrico es un método automático de identificación y verificación de un individuo utilizando características físicas y de comportamiento precisas. Las características básicas que un sistema biométrico para identificación personal debe cumplir son: desempeño, aceptabilidad y fiabilidad (Dahiya & Kant, 2012) (Maltoni et al., 2009).

En dependencia del contexto los sistemas biométricos pueden ser utilizados para realizar verificación o identificación biométrica (Dahiya & Kant, 2012). El proceso de verificación valida la identidad de un individuo al comparar la muestra obtenida con la que se encuentra almacenada en la base de datos; de este proceso forma parte el reconocimiento positivo, el cual consiste en verificar o conocer si un individuo es quien dice ser. El proceso de identificación compara la muestra dactilar adquirida con todas las almacenadas en la base de datos, lo que constituye un componente crítico en la aplicación del reconocimiento negativo, el cual evita que una persona pueda tener múltiples identidades.

Clasificación de los rasgos biométricos

Los rasgos biométricos pueden ser clasificados en físicos o conductuales. La huella dactilar, el iris, la geometría facial, la huella palmar entre otros son parte de los rasgos físicos. La forma de caminar, la voz, la firma manuscrita son considerados rasgos conductuales. Los rasgos conductuales son menos estables

debido a que dependen de diferentes factores ambientales y emocionales que los pueden afectar (Maltoni et al., 2009).

Huellas dactilares

Son características anatómicas identificativas presentes en todas las personas. Están constituidas por rugosidades que forman salientes y depresiones. Las salientes se denominan crestas papilares y las depresiones surcos inter-papilares. En las crestas se encuentran las glándulas sudoríparas. El sudor que éstas producen contiene un aceite el cual es retenido en los surcos de la huella, de tal manera que cuando el dedo hace contacto con una superficie, queda un residuo, produciendo el facsímil o negativo de la huella (Maltoni et al., 2009). En la figura 2 se puede ver el patrón de crestas y los valles que componen un segmento de huella dactilar.

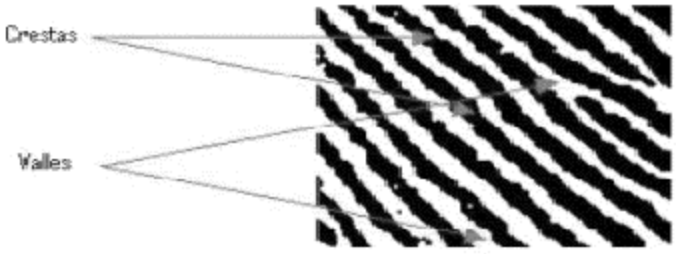

Figura 2: Crestas y Valles

Tomado de (Maltoni et al., 2009)

Minucias

Se representan como discontinuidades locales en el patrón de crestas. Existen varios tipos de minucias, sin embargo, solo las bifurcaciones y terminaciones presentan valor identificativo. La principal razón reside en que las demás pueden verse como una extensión o combinación de estas. Una huella dactilar de buena calidad contiene de 50 a 150 minucias aproximadamente (Maltoni et al., 2009). En la figura 3 se representan los diferentes tipos de minucias.

Figura 3: Tipos de minucias

Tomado de (Maltoni et al., 2009)

Protección de plantillas de minucias

Conceptualmente una plantilla de minucias puede ser protegida al transformar la plantilla en texto claro en otro espacio de transformación utilizando una transformación no invertible (Maltoni et al., 2009). La transformación no invertible más popular es la función hash[1] $H(x) = c$ la cual es utilizada con una función de verificación $V(x, c) = \{verdadero, falso\}$.

1.2 Protección de datos biométricos

Los modelos de protección de plantillas biométricas, han sido clasificados por varios autores (Nagar, Nandakumar, & Jain, 2010; Nandakumar, 2010; Rathgeb & Uhl, 2011) según su funcionamiento en: cripto-sistemas biométricos y transformación de plantillas no invertibles.

1. Cripto-sistemas biométricos: se genera un esquema seguro junto a una llave asociada, a partir de la plantilla biométrica original. Estos datos son almacenados en lugar de la plantilla biométrica

[1] Función hash: es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.

original. Al realizar el proceso de verificación o identificación es posible comparar ambos esquemas seguros y determinar por su grado de similitud si pertenecen a una misma persona. La plantilla original y la llave pueden ser recuperadas utilizando un código de corrección de errores. Este mecanismo no solo cifra la plantilla biométrica, sino que también facilita la administración de las llaves.

2. Transformación de plantillas no invertibles: transforma una plantilla biométrica utilizando una llave específica. Para ello son utilizadas funciones no invertibles, garantizando de esta manera la seguridad criptográfica. El proceso de comparación es realizado en el dominio protegido, de esta manera disminuyen las posibilidades de filtrar información sensible.

El proceso de cifrado de los datos biométricos puede ser realizado de tres formas diferentes (Jain et al., 2013):

1. Cifrando directamente el conjunto desordenado de minucias.
2. Cifrando el conjunto de características biométricas desordenadas derivadas del conjunto de características originales.
3. Cifrando un vector de longitud fija derivado de las características originales.

Para realizar el cifrado de las características biométricas se han formulado tres principios a tener en consideración, tomando en cuenta la naturaleza de los datos (Jain et al., 2013) los cuales se enuncian a continuación:

1. Seguridad criptográfica: consiste en la dificultad computacional de obtener el conjunto de datos biométricos originales a partir del conjunto de datos biométricos protegidos.
2. Revocabilidad: consiste en la posibilidad de obtener múltiples plantillas biométricas seguras a partir de los mismos datos biométricos originales.
3. Rendimiento: Consiste en la capacidad del modelo de mantener el rendimiento del proceso de reconocimiento en términos de tasas de falsos aceptados y tasas de falsos rechazo.

1.3 Análisis de los referentes teóricos sobre la protección de plantillas de minucias de huellas dactilares.

Los modelos de protección de plantillas de minucias de huellas dactilares se clasifican en dos grupos, los dependientes de un método para la alineación de los datos y los libres de alineación. Los modelos que

incluyen métodos de alineación (Jeffers & Arakala, 2007; Nandakumar, Jain, & Pankanti, 2007; Uludag, 2006) almacenan un conjunto de datos denominados datos de ayuda; esto constituye una vulnerabilidad debido a que quedan datos sin cifrar que pueden ser utilizados para realizar ataques de correlación[2]. Los modelos libres de alineación realizan el cifrado de un conjunto de características provenientes de las minucias, invariantes a rotación, traslación y resistentes a deformación no lineal, cifrando toda la información disponible. La seguridad criptográfica en los modelos libres de alineación es mayor debido a que no dejan información sin cifrar. De esta manera resulta más complejo correlacionar diferentes plantillas de minucias protegidas pertenecientes al mismo rasgo biométrico.

1.3.1 Alineación de plantillas de minucias de huellas dactilares

Durante el proceso de comparación de plantillas de minucias de huellas dactilares se verifica si dos plantillas pertenecen al mismo identificador biométrico. Como parte de este proceso es necesario alinear las plantillas de minucias debido a las rotaciones, traslaciones, deformación no lineal y superposición parcial que sufren en cada toma de la muestra. Para alinear las plantillas de minucias en texto claro se utilizan tanto la posición como el ángulo de cada minucia. En el caso de texto cifrado no se dispone de estos datos, por lo que realizar el proceso de alineación es todo un reto.

Para realizar la alineación de plantillas de minucias, antes de realizar el cifrado, se han propuesto varios métodos entre los que se destacan:

- la detección de las singularidades de la huella dactilar (Maltoni et al., 2009).
- la detección de un punto focal (Nandakumar, 2010).
- la selección de una minucia de referencia (C. Lee & Kim, 2010).
- el cálculo del punto iterativo más cercano (X. Zhang, Feng, & He, 2014; Nandakumar, Jain, & Pankanti, 2007; Uludag, 2006).

[2] Ataques de correlación: son ataques que intentan recuperar parte de una secuencia de cifrado ya empleada, dentro de estos ataques hay una clase que podemos denominar divide y vencerás que consiste en encontrar algún fragmento característico de la secuencia de cifrado.

- la formación de estructuras topológicas (Jeffers & Arakala, 2007; J. Li, Yang, Tian, Shi, & Li, 2008; Zhe & Beng Jin, 2011).

Estos métodos realizan el proceso de alineación, sin embargo, presentan un conjunto de limitaciones que afectan el rendimiento biométrico de los modelos de protección de plantilla de minucias que lo utilizan. De realizarse el proceso de alineación en texto claro y almacenarse un conjunto de datos de ayuda no solo se estaría afectando el rendimiento, sino que también se considera una vulnerabilidad debido a que se utilizan estos datos para realizar ataques de multiplicidad de valores. La ausencia de las singularidades (núcleo y delta) en imágenes de huellas dactilares imposibilita la utilización del modelo de alineación utilizando las singularidades de la huella dactilar.

Durante el proceso de selección del punto focal o la minucia de referencia el más mínimo cambio o error conllevaría a una errónea selección y por consiguiente a una pérdida significativa en el rendimiento biométrico. La inclusión o eliminación de minucias durante el proceso de captura y extracción provocan inestabilidad en las estructuras topológicas que se forman para la alineación, además, el almacenamiento de algunas estructuras como datos de ayuda para el proceso de alineación puede ser utilizado para correlacionar dos plantillas de minucias.

Algunos enfoques de modelos de protección utilizan un conjunto de características extraídas de las estructuras topológicas para realizar el proceso de cifrado libre de alineación. En este caso, persiste el problema del cambio de las minucias entre dos extracciones de un mismo rasgo biométrico, lo que degrada el rendimiento biométrico. Entre los esquemas de alineación analizados se encuentran los que a continuación se detallan.

En (Ahn, Kong, Chung, & Moon, 2008) se describe un modelo de cifrado híbrido compuesto por un método de representación de la información contenida en las minucias y una transformación cartesiana. Para ello se extraen un conjunto de características identificativas provenientes de las minucias, invariantes a rotación, traslación y resistentes a deformación no lineal las cuales serán cifradas. El proceso de extracción comienza con la selección de tres minucias de la plantilla de minucias, a continuación, se forma un círculo donde las minucias se encuentran en el borde, se calcula el circuncentro de la tripleta y los ángulos formados entre cada vértice del triángulo y el circuncentro. El proceso de cifrado de las características identificativas consiste en el almacenamiento como se muestra en la figura 4 de:

1. las coordenadas del circuncentro $(x';y')$,
2. el ángulo mayor φ_{12} y su vecino o ángulo adyacente φ_{23}
3. los ángulos ϕ_1, ϕ_2, ϕ_3
4. el tipo.

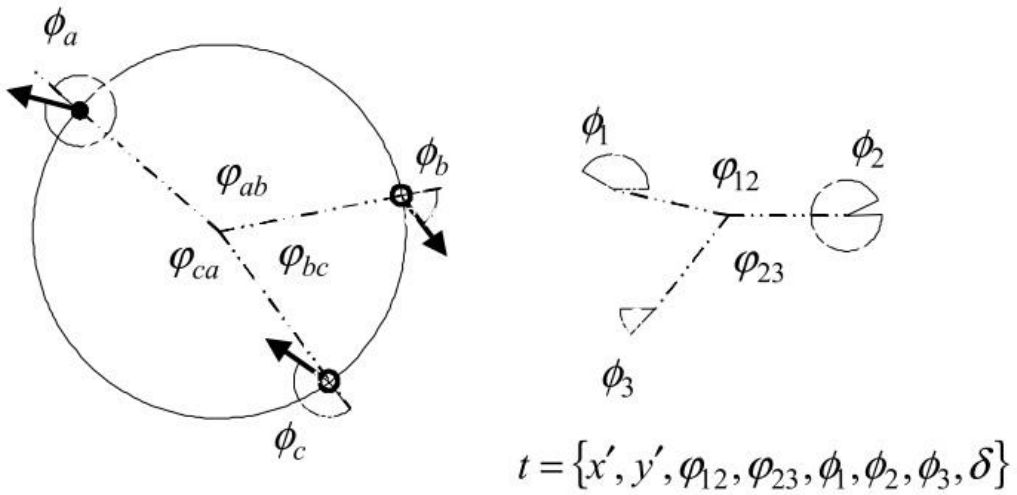

Figura 4: Vector de almacenamiento para el cifrado de las características identificativas

Tomado de (Ahn et al., 2008)

Otro modelo híbrido de protección planteado en (Zhe & Beng Jin, 2011) formula un método de representación y extracción de características identificativas a partir de tripletas de minucias de huellas dactilares. El proceso consta de 4 pasos:

1. Formulación de vecindades de minucias.
2. Descomposición de las vecindades.
3. Extracción de características invariantes.
4. Protección de las plantillas.

La formulación de vecindades de minucias se realiza mediante la selección de las 3 minucias más cercanas (medida por la distancia euclidiana) a la minucia *m* que está siendo analizada. El proceso de descomposición de vecindades consiste en la formación de 4 tripletas de minucias, 3 formadas mediante

la unión de la minucia *m* y dos vecindades y una formada por sus vecindades como se muestra en la figura 5:

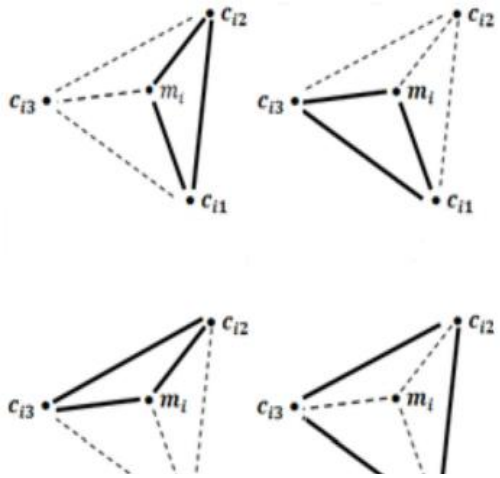

Figura 5: Estructuras triangulares de minucias.

Tomado de (Zhe & Beng Jin, 2011)

En (W. Yang, Hu, & Wang, 2014) se describe un modelo híbrido de protección de plantillas de minucias. Este modelo propone un método de representación y extracción de características identificativas basado en la triangulación de Delaunay denominado cuadrángulo de Delaunay. En este enfoque se genera el diagrama de Voronoi asociado a una plantilla de minucias y se unen los centros de cada par de minucias vecinas para crear la red de Delaunay. Para formar los cuadrángulos de Delaunay se seleccionan dos triángulos que compartan un mismo lado.

Este enfoque registra la información local de las estructuras topológicas formadas y contiene un lado y un ángulo más que lo registrado en la triangulación de Delaunay. De esta manera se evita el registro de información global de la plantilla de minucias. Como principal ventaja sobre la triangulación de Delaunay este enfoque alcanza mayor robustez en cuanto a variación por distorsión no lineal. Las características invariantes que son seleccionadas en este método son:

1. La longitud de los lados.
2. Los ángulos formados entre la dirección de cada minucia y el lado correspondiente a su vecina en dirección a las manecillas del reloj.

3. Los ángulos entre dos lados.
4. El tipo de cada minucia.

Cada vector extraído de un cuadrángulo de Delaunay es cuantizado en una cadena binaria corta y luego son concatenadas todas las cadenas en un único vector característico de longitud fija. Para aumentar la discriminatividad de esta estructura se obtiene una característica adicional de cada cuadrángulo de Delaunay. La necesidad de registrar un punto central o punto de referencia es otra limitación que afecta el rendimiento del sistema en este modelo.

1.3.2 Bóveda difusa

Este modelo es una construcción criptográfica basada en el compromiso difuso propuesto en (Juels & Wattenberg, 1999), es un cripto-sistema biométrico diseñado para realizar el cifrado de conjuntos desordenados. El modelo de bóveda difusa propuesto por primera vez en (Juels & Sudan, 2002) fue concebido como una forma de cifrado tolerante a errores. El método de codificación y decodificación propuesto se basa en el siguiente problema:

Alice busca cifrar su secreto k en un conjunto A que pertenece a un universo público U, para ello se selecciona un polinomio p de manera tal que pueda ser evaluado cada elemento de k como los coeficientes de p obteniendo el conjunto cifrado c. De manera aleatoria Alice crea un conjunto de puntos basuras b, considerados ruido aleatorio, cuya única restricción es que no pueden solaparse con los elementos contenidos en c. Para crear el texto cifrado se mezclan aleatoriamente los elementos del conjunto b y el conjunto c.

Si Bob desea obtener el secreto k utilizando un conjunto de elementos B debe comparar ambos conjuntos y si A y B se superponen de manera sustancial entonces se pueden identificar el conjunto de puntos b. De esta manera Bob puede recobrar un conjunto significativo de puntos reales, sin embargo, el resultado de Bob puede no ser exacto debido a las propiedades especiales de los datos biométricos. Para ello se utiliza un código de corrección de errores. A pesar de que puede ser utilizado cualquier código de corrección de errores se observa en la bibliografía que el más utilizado es el propuesto por Reed-Solomon. Como resultado Bob puede reconstruir p y por lo tanto k de manera exacta.

El modelo propuesto en (Juels & Sudan, 2002) para el cifrado de plantillas de minucias de huellas dactilares está compuesto por dos métodos, un método de codificación y un método de decodificación de la bóveda.

El método de codificación de los datos biométricos propuesto por (Juels & Sudan, 2002) consiste en crear una palabra de código Reed-Solomon generalizada que representa el secreto κ (junto al polinomio correspondiente p donde k representa los coeficientes del polinomio). Se evalúan las coordenadas x correspondientes al conjunto de datos originales A en el polinomio. Para ocultar el resultado de esta operación se genera un conjunto de puntos basura o puntos de burla de la forma (x, y) y se mezclan aleatoriamente. Como premisa en la generación de los puntos basura se tiene que, deben ser seleccionados de manera tal que no se intercepten en el conjunto A ni en el polinomio p.

El procedimiento para decodificar los datos contenidos en la bóveda difusa tiene como entrada el conjunto de muestra B junto a la bóveda V_a y consiste en determinar la palabra de código que codifica el secreto k. Se realiza $k' \leftarrow p$ (procedimiento inverso al cifrado) para denotar la conversión de un polinomio a lo sumo de grado k en el secreto f^k. Se denota $(x_i, y_i) \xleftarrow{(b_i, o)} R$ como la proyección del conjunto cifrado R en la coordenada x (b_i). Si se encuentra un par (b_i, y) que pertenece al conjunto R para cualquier valor de y, entonces $(b_i, y) = (x, y)$, sino es asignado *null* al punto (x, y). De ser exitoso este proceso se obtiene como resultado el secreto k' el cual debe ser igual al original si el conjunto de prueba B es parecido al conjunto original A.

Otros enfoques del modelo de bóveda difusa han sido propuestos para diferentes aplicaciones en específico (Clancy, Kiyavash, & Lin, 2003) o para añadir el proceso de alineación de los datos biométricos (Jeffers & Arakala, 2007; S. Lee, Moon, Choi, & Chung, 2008; J. Li et al., 2008; Uludag, 2006). Los principales cambios realizados al modelo original están en la utilización de un polinomio de grado mayor para la generación de los puntos basura, la inclusión de estructuras topológicas, la selección de un punto focal, del punto de máxima curvatura y el cálculo de un hash geométrico para realizar el proceso de alineación.

1.3.3 Plantillas cancelables

Este modelo consiste en una intencionada y repetida distorsión de la señal biométrica basada en una transformación no invertible (Bolle, Ratha, Connell, & Bolle, 2001). La transformación tiene como propiedad fundamental la no invertibilidad de los datos. Este tipo de transformación puede ser aplicada tanto en el dominio de la señal como a las características extraídas del rasgo biométrico.

En (N. Ratha, Connell, & Bolle, 2006; N. K. Ratha, Chikkerur, Connell, Bolle, & Member, 2007) se describe otro enfoque del modelo plantillas cancelables para la protección de plantillas de minucias de huellas

dactilares basado en el propuesto en 2001. En este enfoque se realiza el análisis en el dominio de las características extraídas y no en el dominio de la señal. Para el cifrado de las características se utiliza una función de un solo sentido o función no invertible con la propiedad de uno a muchos. El proceso de alineación se realiza detectando la simetría parabólica y triangular asociada a los puntos singulares de la huella dactilar. Un hecho importante a destacar en este enfoque es que se describen tres tipos de transformaciones para realizar el cifrado de los datos, las transformaciones cartesianas, polares y funcionales.

La transformación cartesiana mapea las minucias en coordenadas rectangulares utilizando como referencia uno de los puntos singulares, orienta el eje x en la misma dirección que la singularidad y divide el área rectangular en celdas o subáreas de tamaño fijo. Esta transformación consiste en el cambio de celda de las minucias y pueden realizarse rotaciones en múltiplos de 90 grados después de transformada. El mapeo de las celdas se realiza a partir de una matriz de mapeo M por lo que el proceso puede denotarse como $C' = CM$, donde C' es el conjunto transformado y C es el conjunto original.

La transformación radial o polar consiste en el mapeo de las minucias originales en el espacio de coordenadas polares con referencia a la singularidad núcleo. Para realizar el mapeo de las minucias se divide el espacio en sectores polares y se cambian las minucias de sector para alterar la posición y el valor del ángulo. El mapeo es realizado teniendo en cuenta la llave de traslación $1 \times LS$ donde L es la cantidad de niveles y S representa el ángulo. La función de transformación puede describirse como $C' = C + M$.

La transformación funcional consiste en la evaluación de las minucias en una función paramétrica, suavizada localmente pero no globalmente, que se rige por una clave. La función tiene tres restricciones que a continuación se detallan:

1. La transformación debe ser suavizada localmente. Esto asegura que pequeños cambios en la posición de las minucias antes de realizar la transformación conduzcan a pequeños cambios en la posición de los datos transformados.
2. La transformación no debe ser suavizada globalmente. Esto asegura que los datos originales y transformados no estén altamente correlacionados, para asegurar la seguridad criptográfica del modelo.

3. La transformación de los datos debe garantizar que la distancia entre los datos originales y los transformados sea mayor que la aceptada por el algoritmo de comparación.

El proceso de codificación utilizando plantillas cancelables es realizado en cada autenticación y en cada enrolamiento en el sistema biométrico. Si una plantilla protegida es comprometida es posible cambiar la función de transformación para generar una nueva plantilla a partir de los datos biométricos del usuario. De esta manera, aunque se conozca la plantilla protegida y la función de transformación, los datos biométricos originales no pueden ser recuperados.

Otros enfoques de este modelo han sido propuestos para mejorar el proceso de alineación de las plantillas protegidas. En (H. Yang, Jiang, & Kot, 2009) se describe un enfoque libre de alineación en el cual se extraen un conjunto de características identificativas que provienen de las minucias, invariantes a rotación y traslación. Para proteger los datos se utiliza una transformación no invertible de tipo funcional. En (C. Lee & Kim, 2010) se describe el proceso de cifrado utilizando una transformación no invertible del tipo cartesiana. El proceso de alineación en este enfoque se realiza extendiendo el método de alineación mediante la selección de una minucia de referencia. En este caso particular se seleccionan una a una las minucias como minucia de referencia y se conforma un vector binario con las relaciones entre la minucia de referencia y las demás.

En (Jin, Goi, Teoh, & Tay, 2014) se describe otro enfoque del método plantillas cancelables libre de alineación. La transformación seleccionada para el cifrado de los datos es la transformación cartesiana. En este caso se realiza el cifrado de un conjunto de características identificativas extraídas de las minucias, invariantes a rotación y traslación propuestas en (Zhe & Beng Jin, 2011).

1.3.4 Hash biométrico.

Este modelo consiste en la representación y transformación de un conjunto de datos extraídos de las minucias a partir de un punto de referencia, utilizando la técnica de extracción propuesta en (Jain, Prabhakar, Hong, & Pankanti, 1999). Este modelo de protección es aplicado exclusivamente a características de textura de la huella dactilar y consta de un método de representación y un método de filtrado. El método de representación de la información utilizado es denominado *FingerCode* el cual consta de tres pasos fundamentales:

1. Determinar el marco de referencia en la imagen de la huella dactilar.

2. Filtrar la imagen en 8 direcciones diferentes utilizando el banco de filtros de Gabor.
3. Calcular la desviación estándar de los valores de grises en sectores alrededor del punto de referencia.

El filtrado de las características genera un conjunto de discos que contienen la información a ser filtrada para formar un vector de longitud fija que representa el hash biométrico de la huella dactilar bajo análisis. El cálculo de la desviación estándar en estos filtros define los componentes del vector de características. En (Teoh, Ngo, Ling, & Goh, 2004) se describe un nuevo enfoque al descomponer el método en dos componentes, en una transformación integral, invariable y discriminativa de los datos de huellas dactilares con un moderado grado de tolerancia y en la discretización de los datos a través de un producto interno de números aleatorios y datos del usuario. En este enfoque se utiliza el *framework* de transformación de Fourier-Mellin que contiene un conjunto de mejoras en el procesamiento de la imagen. Entre ellas se pueden mencionar la preservación de bordes locales y la reducción de ruido. En vez de utilizar el banco de filtros de Gabor se utiliza la transformada de Fourier para calcular los discos que representan los niveles de grises.

En (Belguechi, Rosenberger, & Aoudia, 2010) se describe otro enfoque del modelo de protección denominado hash biométrico. El principal aporte, en relación con el modelo descrito anteriormente, de este enfoque es la eliminación de la dependencia del núcleo de la huella dactilar como punto de referencia. En este caso se representa cada minucia por su *FingerCode* y para realizar la protección de cada *FingerCode* se realiza el proceso de formación del hash biométrico el cual se describe mediante los siguientes pasos:

1. Cálculo de las características.
    a. Se extrae la plantilla de minucias a partir de la imagen.
    b. Se calcula para cada minucia su *FingerCode*, el resultado es denominado *MinuCode*.
    c. Se obtiene el hash biométrico de cada *MinuCode*.
2. Comparación de las características
    a. Se corrigen las deformaciones causadas por la rotación.
    b. Se procesa el hash biométrico del conjunto de nuevos *MinuCode*.
    c. Se realiza el proceso de comparación local entre las dos plantillas.

En este enfoque se utilizan los filtros de Gabor para filtrar la región de interés y a diferencia del método original esta no se normaliza.

En (Belguechi, Cherrier, Rosenberger, & Ait-aoudia, 2013) se describe otro enfoque para la obtención del hash biométrico. En este trabajo se proponen dos descriptores, un descriptor basado en textura para capturar el patrón de flujo de crestas y otro descriptor basado en minucias para la relación de cada minucia con su vecindad. La extracción de características se realiza similar a lo propuesto por (Belguechi et al., 2010), la variación reside en la utilización de la estructura k-vecindades (*K-Plet*) con centro en la minucia en análisis, para el descriptor basado en minucias.

Este descriptor permite la representación local de la información entre minucias y es seleccionado para verificar si la comparación del descriptor basado en textura es consistente a nivel global. Para minimizar el impacto de los cambios en las minucias que conforman la estructura *K-Plet* se realiza la comparación de las estructuras utilizando una técnica de alineación.

Para realizar el proceso de selección de minucias válidas, el autor de este modelo propone tener en cuenta que la minucia esté circundante al área de interés. Para validar cada minucia se define que, la minucia se encuentre en el límite de la imagen y cada sector se encuentre representado por valles y crestas alternativamente. El proceso de protección comienza con la representación del *MinuCode* de cada minucia seleccionada como válida, se normaliza cada *MinuCode* en el rango [-1, 1], se genera una matriz aleatoria M y se aplica el método de Graham-Smith para convertir la matriz en conjuntos ortonormales. Luego se realiza la proyección de cada *MinuCode* en la matriz, se binariza el resultado, se eliminan los *MinuCode* y se almacenan los *BioCode*.

1.3.5 Extractor difuso.

El modelo denominado extractor difuso es una construcción criptográfica que habilita dos entradas y realiza la comparación en el dominio protegido. En (Dodis, Ostrovsky, Reyzin, & Smith, 2008; Dodis, Reyzin, & Smith, 2004) se explica el proceso y el objetivo principal de este modelo. La idea principal del modelo de cifrado es generar una llave criptográfica partiendo de un conjunto de datos de ayuda para garantizar una autenticación segura y confiable.

El modelo extractor difuso se separa en dos procesos, la generación del esquema seguro y el extractor difuso en sí. La generación del esquema seguro está compuesta por dos procedimientos denominados

generación del esquema y recuperación. La generación de estos procedimientos se realiza a partir de un espacio M con una función de distancia $dist$ bajo dos propiedades esenciales:

- La generación del esquema toma como entrada $w \in M$ y como salida brinda una cadena de bits s.
- El procedimiento de recuperación toma como entrada un elemento $i \in M$ y la cadena de caracteres, además si $dis(w, i) \leq t$ entonces $recuperación\big(i, esquema\ seguro(w)\big) = w$.

El segundo proceso, extractor difuso, es definido como un par de procedimientos aleatorios denominados generación (g) y reproducción (r). En este proceso se cumplen dos propiedades esenciales:

- El procedimiento de generación recibe como entrada una contraseña (w) y extrae una cadena de bits b junto a una cadena de ayuda d.
- El procedimiento de reproducción recibe como entrada un elemento de la contraseña o palabra clave $(i \in M)$ y una cadena de bits d. Este procedimiento verifica si $dis(w, i) \leq t$ y si b, d fueron generados por el procedimiento de generación utilizando la palabra clave w.

La eficiencia de este modelo puede ser medida por el tiempo de ejecución de los procedimientos de generación y recuperación. Si ambos son realizados en tiempo polinómico puede decirse que es un esquema eficiente. El esquema analizado no es invariante a rotación y traslación, por lo que depende de un proceso de alineación para realizar el proceso de comparación. En (Arakala, Jeffers, & Horadam, 2007) es discutido un enfoque dirigido a eliminar esta limitación mediante la adición de un descriptor a cada minucia basada en la estructura de 5 vecindades más cercanas propuestas en (Arakala & Jeffers, 2006). El propósito fundamental de este modelo es realizar el cifrado de datos biométricos de huellas dactilares, sin embargo, puede ser utilizado en otros tipos de datos biométricos como el iris o el rostro.

Inicialmente el modelo extractor difuso fue diseñado para la generación de llaves de cifrado y no abordaba directamente la privacidad de las plantillas de minucias. Este modelo almacena un conjunto de datos de ayuda para realizar la reconstrucción de la plantilla de minucias original. En (Q. Li, Guo, & Chang, 2008) se expone además que al conocerse el esquema seguro es posible obtener parcialmente la plantilla de minucias. Este tipo de esquema solo es aplicable al proceso de verificación de identidad y no al proceso de identificación debido a que cada usuario tiene un esquema de seguridad en específico.

Este modelo constituye las bases para la proposición de los ataques de multiplicidad de valores propuestos por (Scheirer & Boult, 2007).

1.3.6 Modelo de protección de plantillas de minucias de huellas dactilares

Como parte de una tesis doctoral titulada "Modelo para la Protección de Plantillas de Minucias de Huellas dactilares" en la Universidad de las Ciencias Informáticas se está trabajando en la elaboración de un modelo para la protección de plantillas de minucias de huellas dactilares. El modelo que se propone por el autor de esa investigación está compuesto por tres componentes:

1. Componente para la representación y extracción de información identificativa proveniente de las minucias.
2. Componente para el cifrado de características identificativas.
3. Componente para la comparación de plantillas de minucias protegidas.

El componente de representación y extracción de la información identificativa proveniente de las minucias se describe en (Fernández, Cento, & Sentí, 2015). Este componente consta de:

1. Algoritmo para la formación de estructuras complejas de minucias.
    a. Formación de la estructura 5 vecindades más cercanas a una minucia.
    b. Extracción de las tripletas que pueden formarse utilizando el centro y las minucias vecinas de la estructura.
2. Algoritmo para la extracción de características invariantes a rotación y traslación provenientes de las tripletas.
3. Algoritmo para la clasificación de las características extraídas.

Para el cifrado de las plantillas de minucias de huellas dactilares se:

1. Genera una llave de cifrado.
2. Evalúan las minucias en la función polinómica.
3. Se almacenan los datos cifrados.

Para la comparación de plantillas de minucias de huellas dactilares cifradas:

1. Se crean las estructuras protegidas.
2. Se comparan las estructuras protegidas.

   a. Cálculo de la probabilidad de ocurrencia.
   b. Comparación de estructuras primarias.
   c. Comparación de estructuras secundarias.
3. Algoritmo para la consolidación de los resultados.

Hasta el momento en este modelo, no ha sido detectada ninguna vulnerabilidad y no ha sido posible obtener los datos originales a partir de los datos cifrados. Las principales ventajas sobre los modelos analizados anteriormente consisten en:

1. Se almacenan solo los datos transformados y las llaves se gestionan de manera independiente.
2. El modelo desde su concepción es invariante a rotación y traslación y resistente a deformación no lineal y a superposición parcial.
3. La seguridad criptográfica recae sobre la imposibilidad de la reconstrucción polinómica, la cual es considerada un problema NP[3].
4. La revocabilidad del modelo se facilita mediante la utilización de llaves para la generación de funciones de cifrado de manera automática.

En el epígrafe 1.3.7 se analizarán las vulnerabilidades de los modelos de protección de plantillas de minucias de huellas dactilares y los ataques que pueden ser realizados para obtener los datos en texto claro. Debido a que casi ninguno de estos ataques aplica al modelo de protección de plantillas expuesto en el epígrafe 1.3.6 se decide implementar un componente de protección de plantillas de minucias de huellas dactilares utilizando este modelo.

Los modelos analizados realizan la protección de plantillas de minucias de huellas dactilares a partir de diferentes transformaciones y almacenan un conjunto de datos para realizar la comparación en el dominio protegido. La bóveda difusa almacena el dominio y la imagen de la función utilizada para el cifrado de los datos. Esta vulnerabilidad puede ser explotada por ataques de multiplicidad de valores descritos en (Scheirer & Boult, 2007) para obtener las características originales. En el caso de las plantillas cancelables la solución de un conjunto de ecuaciones propuestas en (Quan, Fei, Ann, & Feifei, 2008) dan como

[3] Problema NP: estos problemas son conocidos en la informática como problemas con bajas probabilidades de ser resueltos, en concreto solo pueden ser resueltos con un autómata finito no determinista, lo que implica que se necesite adivinar un conjunto de pasos importantes para su resolución.

resultado las características originales. El hash biométrico es una representación de textura lo que difiere de los sistemas automáticos de identificación de personas los cuales en su mayoría utilizan las plantillas de minucias para la autenticación de personas, además es posible obtener un conjunto cercano al original mediante los ataques (Lacharme, Cherrier, & Rosenberger, 2014) de imagen previa.

Por otra parte, el modelo seleccionado transforma los datos biométricos de forma no invertible y hacia un espacio mucho mayor que el original, además resulta bastante complejo correlacionar datos entre plantillas cifradas ya que pequeños cambios originan grandes transformaciones. Estos aspectos influyen en el dominio de búsqueda de los ataques planteados, por lo cual existen menores posibilidades de encontrar los identificadores biométricos o incluso conocido el identificador biométrico, llevarlo al mismo espacio de trasformación en el cual lo almacena el sistema biométrico.

La utilización de diferentes estrategias para la alineación de los datos ha sido efectiva en el proceso de comparación, sin embargo, las bases criptográficas de cada uno de estos modelos siguen siendo las mismas. Los modelos de bóveda difusa, plantillas cancelables, hash biométrico y extractor difuso por sí solos no presentan un esquema de alineación que permita mejorar los resultados. El rendimiento biométrico se ve afectado por la falta de un esquema de alineación que prediga o detecte las minucias que han sido cambiadas en el conjunto de minucias de muestra en comparación con el conjunto de minucias original. Este problema no afecta al modelo seleccionado a desarrollar en la presente investigación.

1.3.7 Vulnerabilidades de los modelos analizados

Los modelos de protección de plantillas de minucias realizan el cifrado de los datos para garantizar su seguridad criptográfica sin embargo, varios ataques han sido propuestos (Lacharme et al., 2014; Merkle & Tams, 2013; Quan et al., 2008; Rozsa, 2014; Scheirer & Boult, 2007; Security, 2012; Tams, 2012) para obtener, de manera total o parcial, las minucias en texto claro.

Se debe destacar que existen más ataques que afectan el buen funcionamiento de un sistema de autenticación biométrica, sin embargo, la presente investigación se centra en los ataques realizados a las plantillas protegidas para obtener las minucias en texto claro. Los principales ataques son:

1. Ataque vía multiplicidad de valores.
    a. Ataques de correlación.
    b. Ataques de comparación cruzada.

2. Ataques por inversión de llave encubierta.
3. Ataque de sustitución mezclada.
4. Ataque de fuerza bruta.
    a. Ataque de falso aceptado.
5. Ataque de enmascaramiento.
6. Ataque de pre-imagen al hash biométrico.

Los ataques vía multiplicidad de valores definidos en (Scheirer & Boult, 2007) y consisten en la recopilación de varias plantillas protegidas por parte del atacante con el objetivo de compararlas y obtener las características que se mantienen constantes (Minucias reales). En el peor de los casos, utilizando esta técnica, es posible obtener la plantilla real X y la clave K. Al obtener dos o más plantillas de minucias protegidas por el modelo de bóveda difusa y generadas a partir del mismo rasgo biométrico el atacante puede correlacionar los datos biométricos y obtener parcial o totalmente las minucias originales (Tams, 2012).

Existe la probabilidad de que el atacante obtenga algunos puntos basura como reales cuando se utiliza el modelo de bóveda difusa propuesto en 2002. Esto se debe a que algunos puntos basura pueden coincidir con los reales a pesar de ser generados de manera aleatoria, ya que los valores de la plantilla del atacante pueden resultar cercanos y distintos a los originales. En (Kholmatov & Yanikoglu, 2008) se describe el proceso de correlación de plantillas de minucias y los autores del artículo concluyen que, es posible obtener los datos originales en un 59% de los datos analizados.

En el caso del modelo de plantillas cancelables en (Quan et al., 2008) se detalla un proceso para obtener las características originales a partir de dos o más plantillas protegidas mediante este tipo de ataque. Debido a la propiedad uno a muchos que presentan las funciones de cifrado, es posible encontrar varias soluciones inversas por cada minucia a partir de una plantilla cifrada. Todas las soluciones posibles pueden ser tratadas como valor difuso, tomando algunas características como puntos basuras y otras como puntos originales. De esta manera si el atacante obtiene otra plantilla cifrada a partir de la misma plantilla original puede obtener las minucias reales realizando el ataque vía multiplicidad de valores al comparar ambas plantillas. Cuando se obtiene una sola solución pueden revelarse el 90.2 por ciento de los datos originales (Quan et al., 2008).

Los ataques de comparación cruzada y de correlación son tipos específicos de los ataques vía multiplicidad de valores realizados al modelo de bóveda difusa y de plantillas cancelables. Los ataques de comparación cruzada pueden realizarse teniendo como mínimo dos plantillas protegidas. Este tipo de ataque se utiliza en primer lugar para conocer si dos plantillas protegidas pertenecen al mismo rasgo biométrico y en segundo lugar para obtener tanto las características originales como la llave o polinomio de cifrado, o alguna de estas dos. Esta última es la más eficiente debido a que puede desbloquear una bóveda con otra, obteniendo las características originales y la llave de cifrado. En el caso de saber si dos plantillas pertenecen al mismo rasgo puede realizarse solo con los datos de ayuda almacenados para el proceso de alineación.

Los ataques de correlación consisten en obtener el conjunto de pares de puntos presentes en ambas plantillas que comparan entre sí, para ello se auxilian de una función de medición de distancia entre dos puntos que son considerados puntos que comparan entre sí, posteriormente se buscan puntos que sean vecinos a estos y compartan esta medida. El cálculo de una medida de distancia razonable o apropiada es una premisa necesaria para el éxito de esta técnica y se describe en detalle en (Tams, 2012). Una vez lograda la decodificación de los puntos genuinos, que interpolados dan como resultado el polinomio y las características originales, se termina el ataque de manera satisfactoria. Los ataques de correlación y comparación cruzadas pueden realizarse en conjunto. Un ejemplo de ello es el ataque a dos bases de datos que contengan usuarios en común para obtener las plantillas protegidas de ambos y a partir de ellas los datos en texto claro.

Los ataques por repetición de llave encubierta tienen como objetivo obtener una llave de encriptación válida. El modelo de bóveda difusa propone la liberación de la clave de cifrado en cada intento de comparación que sea positivo, lo que constituye una vulnerabilidad explotada por este tipo de ataque. En este caso el atacante puede decodificar la plantilla protegida y obtener los datos biométricos originales o codificar un conjunto de datos biométricos.

Los ataques de sustitución mezclada se realizan mediante la alteración del registro biométrico con o sin conocimiento de los datos biométricos almacenados en la plantilla protegida. Este tipo de ataque produce la denegación del servicio de autenticación al usuario genuino mientras asegura la autenticación del impostor. Otro enfoque de este ataque propone la combinación de los datos biométricos del atacante con

los del usuario genuino. Este enfoque es conocido como mezcla interna y es mucho más difícil de detectar debido a que no provoca la denegación del servicio de autenticación.

Los ataques de fuerza bruta se realizan mediante la generación de todas las posibles combinaciones de muestras para revelar la auténtica. En (Tams, 2012) se implementaron varios enfoques del método de protección bóveda difusa para realizar ataques de fuerza bruta. El ataque de fuerza bruta clásico realiza enviando al sistema una bóveda *V* que contiene un conjunto de puntos y un polinomio *p* de grado *k* generados aleatoriamente. El ataque termina cuando se satisface el criterio establecido para encontrar el polinomio correcto. El criterio utilizado consiste en comprobar cuántos puntos pertenecientes a la bóveda *V* son interpolados de manera correcta en cada intento de autenticación. Este ataque tiene bajas probabilidades de ser efectivo en el modelo de protección de minucias de huellas dactilares seleccionado, ya que el dominio de los datos crece considerablemente mediante las transformaciones realizadas por los polinomios, por lo cual se requieren un valor más elevado de intentos.

El ataque de falso aceptado es una mejora del ataque de fuerza bruta. Este ataque se realiza utilizando una base de datos de plantillas de minucias y una bóveda difusa que haya sido interceptada. Se realiza la interpolación de cada muestra en la base de datos y se comparan los resultados hasta satisfacer el criterio para encontrar el polinomio correcto descrito anteriormente. Este tipo de ataque requiere mucho más esfuerzo y se estima que la probabilidad de encontrar el polinomio correcto es de $1-(1-\in)^n$. De conocerse el valor de la tasa de falso aceptado $\in$ es posible realizar este ataque de una manera más sencilla.

El ataque de imagen previa descrito en (Lacharme et al., 2014) se realiza al modelo de hash biométrico. El objetivo principal es utilizar un algoritmo genético para obtener un aproximado de los estados intermedios *FingerCode* y *BioCode*, en la generación del hash biométrico. Para ello se describe como genotipo al candidato a *FingerCode* descrito como un vector de dimensión *m*.

Como población se escoge un conjunto de candidatos (10000 en (Tams, 2012)) caracterizados por su genotipo. Se describe además una función de aptitud que permite cuantificar la aptitud de un individuo con su medio ambiente, teniendo en cuenta su genotipo. Por último, se describe un operador de genotipos que permite definir alteraciones en el genotipo para permitir que evolucione la población en el tiempo utilizando tres operadores:

1. Mutación.
2. Selección.
3. Cruzamiento.

Utilizando este tipo de ataque es posible obtener una plantilla similar a la plantilla original bajo condiciones reales.

Los ataques descritos en la bibliografía para los modelos de protección de plantillas de minucias de huellas dactilares demuestran que el requisito de seguridad criptográfica no se cumple. La obtención de las plantillas de minucias en texto claro constituye un riesgo para la seguridad de la información protegida biométricamente. En la tabla 1 se realiza un resumen de los modelos y los ataques por los cuales son afectados.

Tabla 1: Resumen de los modelos y los ataques por los cuales son afectados

Tomado de (Tams, 2012)

Ataques	Bóveda difusa	Plantillas cancelables	Hash biométrico	Modelo de protección de plantillas de minucias de huellas dactilares
Multiplicidad de valores	X	X		
Correlación	X	X		
Fuerza bruta	X	X		X
Sustitución mezclada	X			
Imagen previa			X	
Enmascaramiento	X	X	X	
Repetición de llave encubierta	X	X		

1.4 Análisis de Tecnologías, metodologías y herramientas.

Para el desarrollo del presente trabajo de diploma se analizan un conjunto de metodologías, tecnologías y herramientas que guiarán el proceso de desarrollo de software.

1.4.1 Metodologías de Desarrollo de Software

En todo proceso de desarrollo de software se corren riesgos que en ocasiones son difíciles de controlar, más aún cuando no se cuenta con una planificación que encierre en todos los aspectos las acciones que se deben de llevar a cabo para minimizar o eliminar el impacto que pueden traer consigo. Detallar una planificación consiste en documentar los pasos a seguir ante las dificultades, y no solo eso, las planificaciones rigen todo el proceso de forma organizada.

En la actualidad estas funciones las realizan las metodologías de desarrollo de software, las cuales surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dada la variedad de metodologías existentes y la clasificación de las mismas, son muchos los factores a tener en cuenta a la hora de seleccionar una, que se ajuste al ambiente de desarrollo y ayude, en vez de dificultar, a obtener los resultados esperados.

Existen dos grandes enfoques de metodológicos: los tradicionales o pesados y los ágiles.

Las **metodologías pesadas o tradicionales** se centran en la definición detallada de los procesos y tareas a ejecutar, en las herramientas a utilizar y además requieren una extensa documentación, pues pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias en proyectos grandes, con respecto al tiempo y a los recursos que se necesitan emplear, donde una gran organización es requerida.

Las **metodologías ágiles** emergen como una posible respuesta para simplificar todos los procesos que traen aparejadas las metodologías tradicionales y obtener un enfoque directo hacia el desarrollo del producto. Por estar especialmente orientadas para proyectos pequeños, constituyen una solución a medida para ese entorno (el entorno de proyectos pequeños), aportando una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. El punto de partida para estas metodologías es el Manifiesto Ágil (Beedle et al., 2001), un documento que resume la filosofía ágil. Según el Manifiesto se valora:

1. Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
2. Desarrollar software que funciona más que conseguir una buena documentación.
3. La colaboración con el cliente más que la negociación de un contrato.
4. Responder a los cambios más que seguir estrictamente un plan

Para la selección del enfoque metodológico ágil de la presente investigación se tienen en cuenta los siguientes aspectos (Figueroa, 2004):

- Tamaño del equipo de desarrollo: 1 persona.
- Flexibilidad de la investigación ante los cambios.
- Necesidad de desarrollo antes de documentación exhaustiva.
- Tiempo de entrega del proyecto

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil, cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación, se resumen dichas metodologías ágiles:

SCRUM (Schwaber K., Beedle M., 2001) Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Tiene dos características fundamentales: el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente, la segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Crystal Methodologies (Cockbun, 2001). Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo: Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

Dynamic Systems Development Method (**DSDM**) (Stapleton J., 1997). Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD[4] unificada. Sus principales características son: se considera un proceso iterativo e incremental donde el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación.

Adaptive Software Development (**ASD**) (Highsmith J., 2000). Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje; en la primera de ellas se inicia el proyecto y se planifican las características del software, en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

Feature-Driven Development (**FDD**) (Coad P., Lefebvre E., 1999). Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

Lean Development (**LD**) (Poppendieck M., 2003). Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

Extreme Programming (XP) (Beck, 1999).Desarrollada por Kent Beck, la cual se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo. Este tipo de método se basa en una retroalimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca simplificar

[4] El **Desarrollo rápido de aplicaciones** (o *RAD*) definido por James Martin a principios de la década de 1980, consiste en un ciclo de desarrollo corto basado en tres fases (Requisitos, Diseño y Construcción) con un plazo de entrega ideal de 90 a 120 días como máximo.

las soluciones implementadas y ayuda a flexibilizar el impacto de múltiples cambios. Este tipo de metodología es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo.

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Kent Beck plantea que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores. El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres. XP se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito, estos valores esencialmente son comunicación, simplicidad, retroalimentación y coraje.

El ciclo de vida de un proyecto XP es muy dinámico y se puede separar en fases las cuales tienen un conjunto de reglas y prácticas que se deben de llevar a cabo.

1. Planificación**:** XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. Los conceptos básicos de esta planificación son los siguientes: Historias de usuarios, Plan de entregas, Plan de iteraciones y las Reuniones diarias de seguimiento.
2. Diseño**:** XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son: Simplicidad, Soluciones “spike” o claves, Recodificación y Metáforas.
3. Implementación: se considera la puesta en marcha del proyecto a todos los niveles. Los conceptos básicos a tener en cuenta son: Disponibilidad del cliente, Uso de estándares, Programación dirigida por las pruebas, Programación en pares, Integraciones permanentes, Propiedad colectiva del código y llevar un ritmo sostenido.
4. Pruebas**:** las pruebas son las piedras angulares de todo proceso en XP. Dentro de esta fase se encuentran: Pruebas unitarias, Detección y corrección de errores y Pruebas de aceptación.

Resumiendo, los aspectos más generales de XP, se puede decir que esta es una metodología que se basa en la simplicidad, comunicación e interacción permanente con el cliente, pues forma parte del equipo de desarrollo, lo que posibilita comprobar los requisitos a lo largo de todo el proceso de desarrollo.

La Tabla 2 obtenida de (Highsmith, 2002), compara las distintas aproximaciones ágiles en base a cuatro parámetros: vista del sistema como algo cambiante, la colaboración entre los miembros del equipo, experiencia particular del equipo de desarrollo utilizando la metodología y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados y adaptabilidad. Para medir los parámetros se utiliza una puntuación en el intervalo [1,5] donde el valor 5 representa la mejor estimación para una metodología en cuestión.

Tabla 2: Resultados de las mediciones ágiles para seleccionar la metodología

Tomado de (Highsmith, 2002)

Parámetros a medir	ASD	Crystal	DSDM	FDD	LD	SCRUM	XP
Sistema como algo cambiante	5	4	3	3	4	5	5
Colaboración	5	5	4	4	4	5	5
Experiencia del equipo de desarrollo utilizando la metodología	1	1	1	1	1	2	4
Características de las metodologías.							
-Simplicidad	4	4	3	5	3	5	5
-Excelencia técnica	3	3	4	4	4	3	4
-Resultados	5	5	4	4	4	5	5
-Adaptabilidad	5	5	3	3	4	4	3
Total	28	27	22	24	24	29	31
Promedio	4.0	3.85	3.14	3.42	3.42	4.14	4.42

Los valores de la tabla justifican la selección de la metodología XP para desarrollar el componente y cumplir con los objetivos propuestos.

1.4.2 Lenguajes de Programación.

Entre los lenguajes de programación más utilizados para el desarrollo de componentes biométricos se pueden encontrar C#, Java, C y C++. Estos lenguajes brindan comodidades para facilitar la implementación de los requerimientos funcionales, algunas de estas comodidades son: el uso de estructuras de datos predefinidas, brindan un considerable número de librerías que pueden ser utilizadas

para simplificar y agilizar los procesos de implementación, en última instancia soportan varios paradigmas de programación, lo que permite adaptar las soluciones y mejorar la robustez de las aplicaciones (Marcelo Gabriel, Ana Lía Ramona, Carlos Eduardo, 2015). Análisis realizados por diferentes autores en (Stroustrup, 1997) señalan a C y C++ como los lenguajes con mejores condiciones para el desarrollo de librerías, módulos o componentes de software, estos análisis se sustentan fundamentalmente por la independencia de estos lenguajes para la gestión de recursos en los ordenadores, también tienen en cuenta el tiempo de respuesta de las llamadas a los procedimientos, las cuales son significativamente menores en comparación con C# y Java.

Por otra parte, el equipo de desarrollo ha tenido experiencia en el desarrollo de aplicaciones utilizando a C++ como lenguaje de programación. Por las razones antes expuestas se selecciona el lenguaje C++ en su versión 11.0 para la implementación del componente. A continuación, se ofrece una descripción más detallada del lenguaje

C++ (Stroustrup, 1997): Es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multi-paradigma. Algunas ventajas del uso de este lenguaje son: incorpora soporte para el paradigma orientado a objetos, es muy potente en lo que se refiere a creación de sistemas complejos, se caracteriza por su robustez, incorpora el lenguaje C lo que permite el uso de sus instrucciones, librerías y características, gracias a sus herramientas otorga un control mucho más amplio sobre las operaciones que se desean realizar, permite administrar de forma manual y automática la memoria del sistema por lo que no es obligatorio el uso de recolectores de basura. Su más reciente versión (C++11) brinda un conjunto de herramientas mucho más potentes y cómodas para los programadores lo cual permite agilizar el proceso de desarrollo de los proyectos. En esta última década se han desarrollado un sin número de aplicaciones de escritorio, webs y para todo tipo de dispositivos electrónicos con el uso de este lenguaje.

1.4.3 Entorno de desarrollo

Qt Creator: Es un Entorno Integrado de Desarrollo (IDE) creado por Trolltech, es multiplataforma y fue diseñado para desarrollar utilizando lenguajes como C/C++ y Python con la Integración del Framework Qt para lograr una agilización y simplificación en el desarrollo de aplicaciones que pueden ser para escritorio, web o para móviles. En el año 2008 Nokia compra Qt a Trolltech y luego en el 2011 tras su alianza con Microsoft decide vender la licencia comercial de Qt a Digia.

Qt Creator no pretende ser un reemplazo de Eclipse ni Visual Studio, sino un IDE ligero pensado especialmente para el desarrollo en múltiples plataformas: Windows, Linux (desde la versión 2.6) y Mac OSX (desde 10.4 en adelante).

Qt Creator se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a aprender y comenzar a desarrollar rápidamente, también aumenta la productividad de los desarrolladores con experiencia en Qt.

- Editor de código con soporte para C+, *QML y ECMAscript*
- Herramientas para la rápida navegación del código
- Resaltado de sintaxis y auto-completado de código
- Control estático de código y estilo a medida que se escribe
- Soporte para re-factorizar código
- Ayuda sensitiva al contexto
- Plegado de código (code folding)
- Paréntesis coincidentes y modos de selección

El depurador visual (visual debugger) para C++ es consciente de la estructura de muchas clases de Qt, lo que aumenta la capacidad de mostrar los datos de Qt con claridad. Además, Qt Creator muestra la información en bruto procedente de GDB[5] de una manera clara y concisa.

[5] **GDB** fue escrito por Richard Stallman en 1986. **GDB** es software libre distribuido bajo la licencia GPL. **GDB** ofrece la posibilidad de trazar y modificar la ejecución de un programa. El usuario puede controlar y alterar los valores de las variables internas del programa.

- Interrupción de la ejecución del programa.
- Ejecución línea por línea o instrucción a instrucción.
- Puntos de interrupción (breakpoints).
- Examinar el contenido de llamadas a la pila (stack), los observadores y de la variables locales y globales.

Conclusiones parciales

- El análisis de los referentes teóricos metodológicos facilitó la comprensión de diferentes temas asociados al problema de la investigación.
- Los conceptos asociados al dominio de la investigación aportaron claridad en la división del problema de la investigación en pequeños problemas menos complejos.
- El análisis de los modelos para la protección de plantillas de minucias de huellas dactilares en conjunto con sus vulnerabilidades posibilitó la selección de un método de cifrado de plantillas de minucias de huellas dactilares.
- La caracterización de los modelos de protección en libres y dependientes de alineación facilitó la selección de un esquema de transformación para obtener datos libres de alineación.
- La selección de las metodologías, tecnologías y herramientas aportó una mejor organización de los procesos necesarios para la implementación del componente.

Capítulo 2: Características de la solución propuesta

En este capítulo se describen las principales características del componente propuesto. Se analizan detalladamente los algoritmos que utiliza el modelo. Se confecciona el modelo de dominio donde se relacionan los principales conceptos que fundamentan al componente. Se describen las historias de usuario, se planifica el proyecto y se estima un plan de entrega. Además, se define la arquitectura y se especifican los patrones arquitectónicos utilizados, y mediante ellos se logra un mejor entendimiento para el posterior desarrollo del sistema.

2.1 Principales conceptos asociados a la propuesta de solución:

La presente investigación sentará sus bases en un modelo de dominio mediante el cual se podrá comprender el entorno del componente a desarrollar y permitirá a los usuarios, desarrolladores e interesados utilizar un vocabulario común para poder entender el contexto en que se desarrolla el sistema. Los principales conceptos con los que trabaja el componente se describen a continuación:

Minucias: constituyen las unidades atómicas extraídas de las huellas dactilares.

Tripletas: constituyen subestructuras triangulares que pertenecen a una estructura compleja.

Estructuras complejas: constituyen estructuras formadas por la unión de las n vecindades más cercanas y tripletas de minucias.

Plantillas de minucias: datos que representan la medida del identificador biométrico luego de ser enrolado en el sistema y que será utilizado en el proceso de verificación de identidad para comparar con diferentes plantillas biométricas de prueba.

Algoritmo de comparación: se encarga de realizar una comparación entre dos plantillas de minucias de huellas dactilares cifradas que tiene como resultado el grado de similitud.

Umbrales de comparación: representan índices de ajustes que utiliza el algoritmo de comparación para flexibilizar la comparación de dos plantillas de minucias de huellas dactilares.

Llaves: representan polinomios de un grado determinado que se utilizan para cifrar las plantillas de minucias de huellas dactilares.

Algoritmo de cifrado: se encarga de transformar los datos de las plantillas de minucias de huellas dactilares utilizando una llave (polinomio).

Algoritmo de extracción: se encarga de extraer los datos biométricos de las minucias y conformar las estructuras complejas (Fernández et al., 2015).

En la figura 4 se representa el diagrama de dominio mediante el cual se pueden ver los conceptos relacionados. Se evidencian las plantillas de minucias de huellas dactilares como el centro del diagrama, las cuáles serán transformadas por un algoritmo cifrado utilizando una llave polinómica, de esta forma los rasgos biométricos se normalizarán a otro espacio de transformación con la intensión de ser protegidos.

El algoritmo de extracción realizará el pre-procesamiento de los datos biométricos protegidos para preparar a las plantillas cifradas para la comparación, agrupando a las minucias por su vecindad o cercanía espacial y conformando las estructuras complejas. Cuando sea solicitada una verificación de identidad, el algoritmo de comparación determinará el índice de similitud entre dos plantillas de minucias de huellas dactilares protegidas utilizando la misma llave con la cual fueron cifradas y flexibilizando el proceso de comparación mediante la utilización de umbrales de ajuste, con el fin de emparejar dos muestras por el desplazamiento o rotación presente en el momento que se realizan las tomas en el escáner.

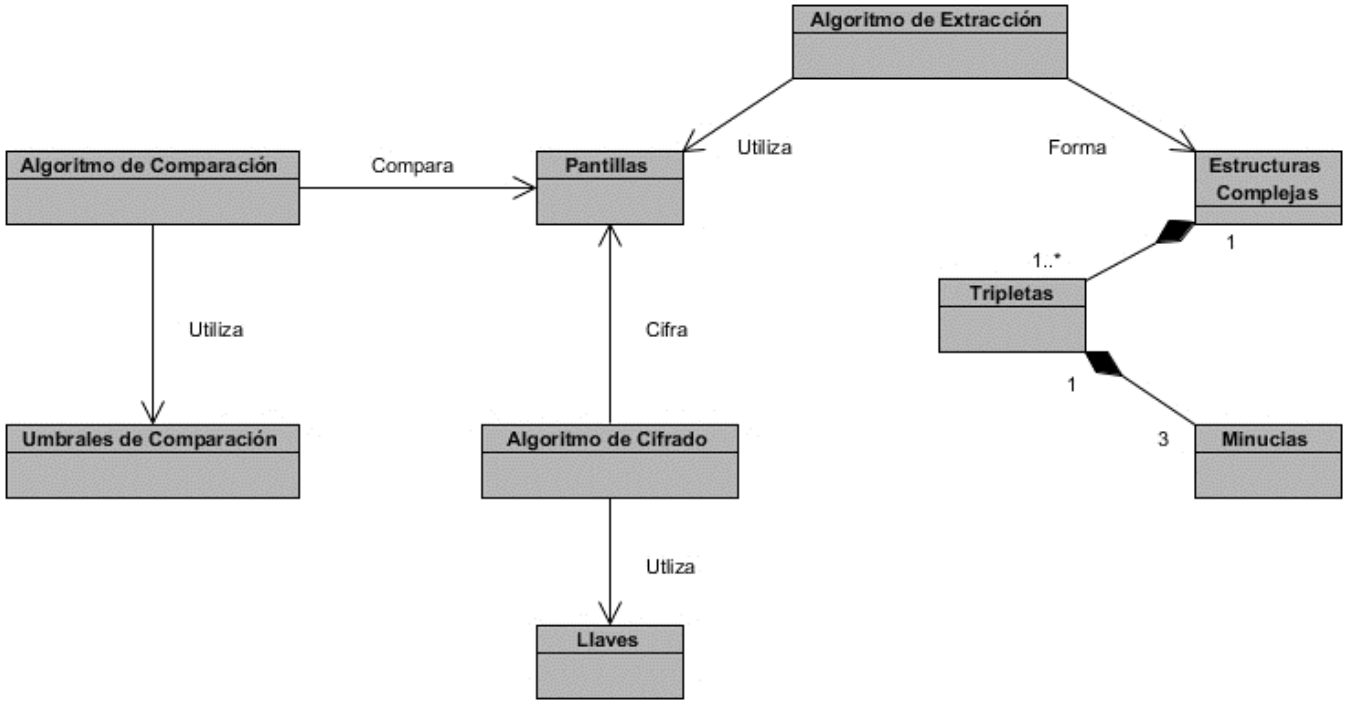

Figura 6: Modelo de dominio

Elaboración propia a partir del estudio de los modelos de protección.

2.2 Algoritmos utilizados en la propuesta de solución.

Con el objetivo de proteger el identificador biométrico y reducir los errores de comparación en cuanto a índices de falsos rechazos y falsos aceptados originados por la calidad de las extracciones de las diferentes muestras de plantillas de minucias se propone utilizar los siguientes algoritmos:

2.2.1 Descripción del algoritmo de cifrado.

El algoritmo de cifrado propuesto se basa en la transformación de los datos de las minucias a partir de un polinomio de grado *n*, denotado por $p(x)$ con el objetivo de obtener otro espacio de transformación proporcional al inicial. Para la generación de los coeficientes del polinomio se utiliza el **Método de Generación de Valores Aleatorios: Congruencial Mixto** (Moraleda, 2010) dado por la fórmula siguiente:

$$Xi + 1 = (a * Xi + c\,) mod\, m$$

Donde los Xi son los valores aleatorios obtenidos a partir de un valor semilla X_0 para la primera iteración y los valores de a , b y m son valores enteros positivos que permiten acotar el valor máximo de la sucesión obtenida y obtener el siguiente elemento.

Una vez obtenidos los *n* valores aleatorios que definen los coeficientes ya está conformado el polinomio $p(x)$ y se procede realizar la evaluación de cada valor correspondiente a cada minucia dentro de la plantilla de minucias. De esta forma se realiza el cifrado de los datos de las plantillas de minucias.

La figura 5 muestra la plantilla original y la plantilla cifrada usando la llave de cifrado generada por el método congruencial mixto:

$$p(x) = 3X^2 + 8X + 4$$

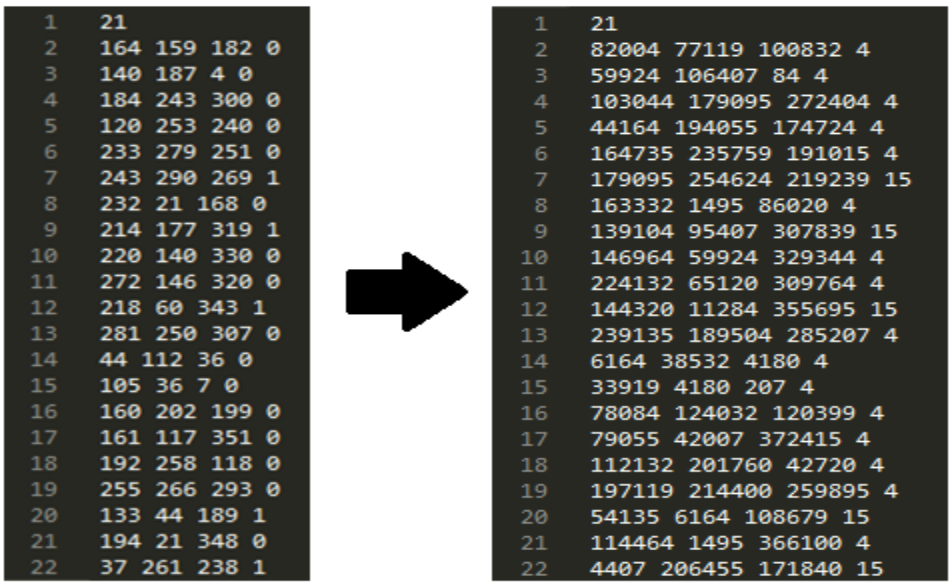

Figura 7: Vista del Cifrado de la Plantilla

Elaboración propia a partir del resultado del cifrado.

2.2.2 Descripción del algoritmo de extracción.

Para la extracción de las características identificativas de las plantillas de minucias primeramente se realiza una lectura de la plantilla cifrada, obteniendo la colección de minucias cifradas. El segundo paso consiste en localizar el **centro de gravedad de la plantilla** $Ptc(x, y)$. Este consiste en determinar el centro de la colección de minucias utilizando las expresiones:

$$Ptc(x) = \frac{1}{n} * \sum_{i=1}^{n} (Mi.X)$$

$$Ptc(y) = \frac{1}{n} * \sum_{i=1}^{n} (Mi.Y)$$

Donde: $Mi.X$ y $Mi.Y$ son las abscisas y las ordenadas de las minucias.

Posteriormente se calcula la distancia euclidiana[6] desde $Ptc(x, y)$ hacia cada minucia. Se crea el conjunto MOi con las minucias ordenadas de menor a mayor según la distancia calculada y se comienza a seleccionar, del conjunto ordenado, cada minucia MOi. De esta manera se conforma una estructura compleja teniendo en cuenta las 5 minucias más cercanas del conjunto global a la minucia central MOi seleccionada en cada iteración, el proceso se explica mejor en (Belguechi et al., 2010). El resultado de este algoritmo se muestra en la figura 6:

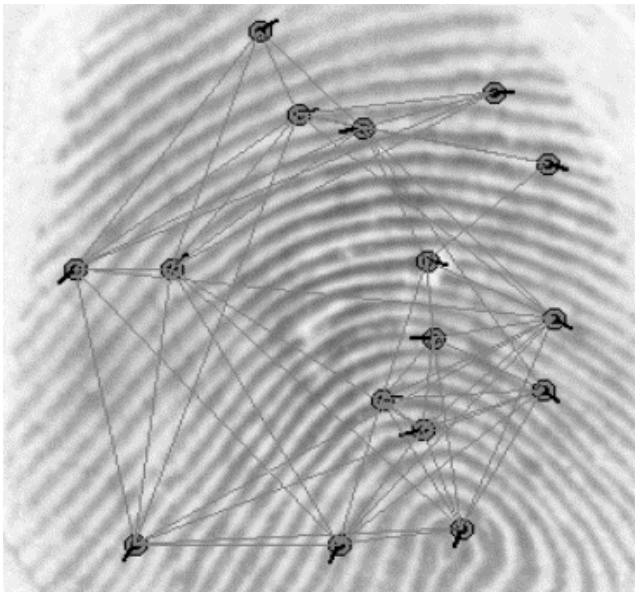

Figura 8: Resultado de la Extracción

Elaboración propia a partir del resultado de la extracción.

El último paso es calcular los lados y los ángulos interiores de las tripletas, a continuación, se muestran las tripletas primarias[7] y secundarias[8] seleccionadas para cada estructura compleja, la notación usada se define como [*a, b, c*], donde *a*, *b* y *c* son los vértices representados por las minucias pertenecientes a una estructura compleja dónde, el vértice central corresponde al valor 0 y los restantes (1-5) en orden anti-horario. El método se describe en detalle en (Fernández et al., 2015).

[6] Distancia Euclidiana: dado dos puntos p1(x1,y1) y p2(x2,y2) se calcula como: $d(p1,p2) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$
[7] Tripletas Primarias: estructuras formadas (de tres lados) a partir de la minucia central.
[8] Tripletas Secundarias: estructuras formadas (de tres lados) que no tienen ninguna minucia central como vértice.

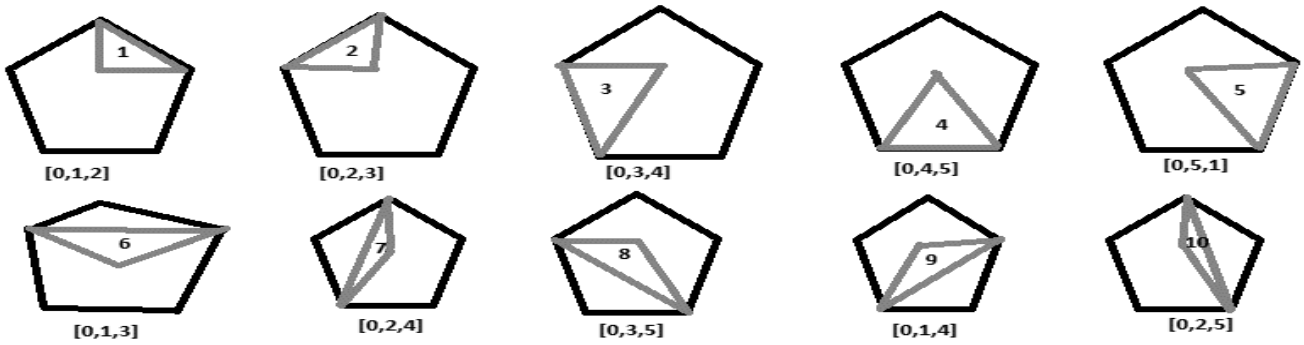

Figura 9: Tripletas Primarias

Elaboración propia a partir del análisis de las estructuras complejas

.

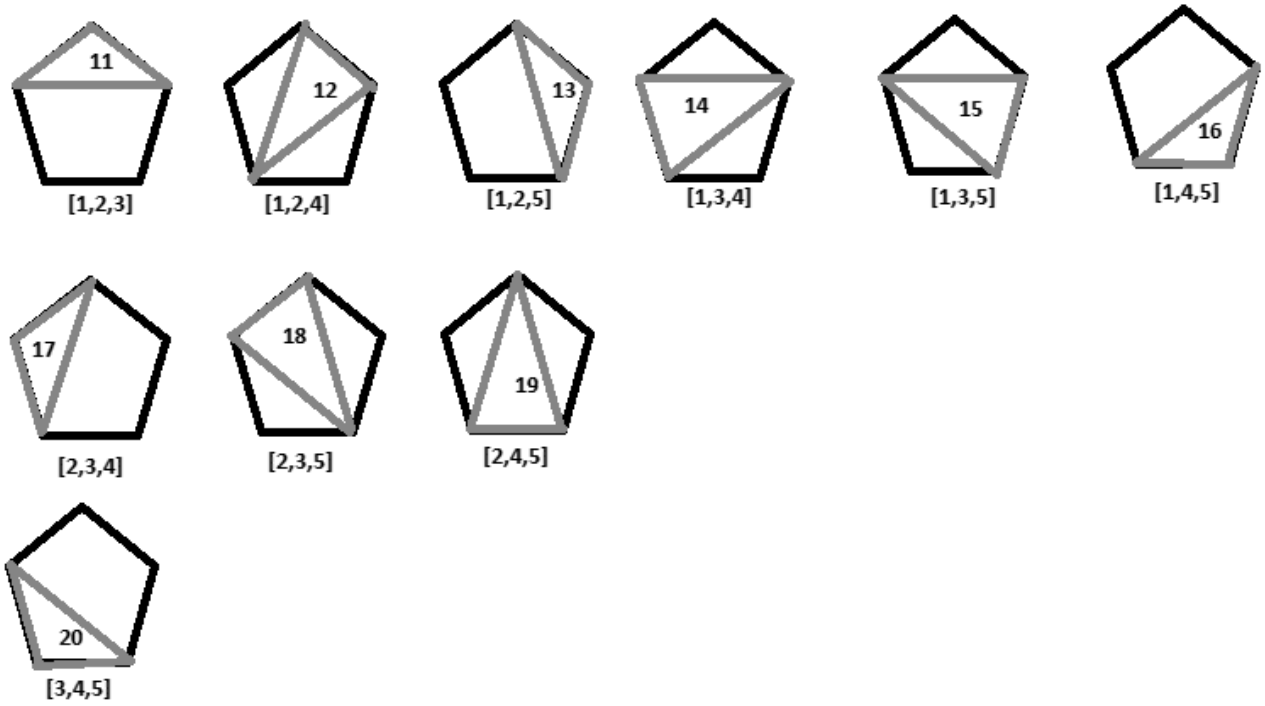

Figura 10: Tripletas Secundarias

Elaboración propia a partir del análisis de las estructuras complejas

2.2.3 Descripción del algoritmo de comparación

Para comparar dos plantillas de minucias de huellas dactilares se definen varios descriptores que a continuación se detallan:

Grados de libertad de los lados de las tripletas (Gll): se utiliza para considerar dos lados iguales, si el cálculo de su diferencia es menor que este valor.

Grados de libertad de los ángulos de las minucias (Gla): se utiliza para considerar dos orientaciones de minucias iguales, si la diferencia absoluta es menor o igual que este valor.

Grados de libertad de los ángulos interiores de las tripletas ($Glat$): se utiliza para considerar dos ángulos interiores de dos tripletas iguales, si el cálculo de su diferencia es menor que este valor.

Grados de validez de las tripletas (Gv): se utiliza para descartar tripletas que tengan un ángulo interior mayor que este valor.

Umbral de similitud de cada estructura compleja (Us): se utiliza para considerar la similitud entre dos estructuras complejas de acuerdo a su grado de semejanza, el cual se establece en porciento.

Dadas dos plantillas de minucias de huellas dactilares denominadas *P1* y *P2* respectivamente, se obtienen los conjuntos de estructuras complejas de cada plantilla de minucias denotados por *P1S* y *P2S*.

El segundo paso consiste en: para cada elemento Si de *P1S,* obtener cada elemento Sj de *P2S* para comparar Si con Sj atendiendo a la cantidad de tripletas que se consideran iguales, para cada par de tripletas $T1$ y $T2$ de Si y Sj respectivamente. En caso de que alguna tripleta $T1$ o $T2$ contenga un ángulo interior del triángulo que forman mayor a Gv no se considera válida y no se tiene en cuenta en la comparación.

Para realizar la comparación entre tripletas válidas se utilizan las siguientes expresiones:

$$absLi = \frac{|T1.Ladoi - T2.Ladoi|}{p(500)}$$

$$absAMi = \frac{|T1.Minuciai - T2.Minuciai|}{p(360)}$$

$$absATi = | T1.ATi - T2.ATi |$$

$$i \in N, 0 \leq i \leq 2$$

Donde cada $Ladoi$, $Minuciai$ y ATi representan la longitud de los lados de las tripletas, la orientación de las minucias y los ángulos interiores de las tripletas respectivamente. Los valores fijos $p(500)$ y $p(360)$ representan una cuota superior para los valores cifrados de las tripletas; $p(x)$ es el polinomio utilizado para cifrar *P1* y *P2.* Se pueden seleccionar diferentes valores para la cuota superior, pero en la presente investigación se proponen estos debido a que el mayor lado de una tripleta se estima que sea 500 por la resolución de la imagen y 360º es un ángulo completo, por lo que la orientación de las minucias siempre será menor.

Posteriormente, se comparan los **umbrales de ajuste** con cada valor absoluto obtenido. Esto es importante para garantizar que las plantillas de minucias pertenecen al mismo dedo de la misma persona, a pesar de que sufran traslación, rotación, superposición parcial o deformación no lineal.

Se consideran iguales $T1$ y $T2$ si:

$$absLi \leq Gll, absAMi \leq Gla, absATi \leq Glat$$

En caso de coincidencia se elimina la tripleta Ti de $T2$ para evitar coincidencias múltiples.

Se cuentan la cantidad de tripletas que coinciden (Cij) entre Si y Sj y se calcula el índice de similitud ($ISij$) con la siguiente fórmula:

$$ISij = \frac{Cij}{TT} * 100$$

Donde TT es la cantidad de tripletas válidas de Si

Si el $ISij$ de las estructuras complejas Si y Sj es mayor o igual que Us entonces se consideran similares dichas estructuras complejas.

El último paso consiste en calcular el índice de similitud de las Plantillas (ISP) y se calcula con la siguiente expresión:

$$ISP_{1,2} = \frac{CSS_{1,2}}{TS_1} * 100$$

Donde $CSS_{1,2}$ constituyen la cantidad de estructuras complejas similares para *P1* y *P2.*

2.4 Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando "historias de usuarios", las que sustituyen a los tradicionales "casos de uso". Una vez obtenidas las "historias de usuarios", los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene "riesgos" que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas "historias de usuarios" (Beck, 1999).

2.4.1 Historias de Usuarios

Las historias de usuario son el artefacto utilizado por la metodología seleccionada para especificar los requisitos del software. Las historias de usuario deben ser independientes unas de otras, negociables, valoradas por los clientes o usuarios, estimables, pequeñas y verificables. A continuación, se describen las historias de usuario de la presente investigación.

Para lograr el cifrado de los datos de las plantillas el componente debe permitir generar, cargar y guardar llaves, a continuación, se muestra la descripción de esta historia de usuario:

Tabla 3: Historia de usuario generar llaves

Elaboración propia.

Historia de Usuario	
Número: 1	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Generar llaves	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto

Puntos Estimados: 1	Iteración Asignada: 1
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El componente debe generar polinomios de un grado especificado por el usuario obteniendo los coeficientes a partir del método congruencial mixto configurado por el usuario, además el componente debe permitir cargar o guardar los polinomios en ficheros llaves.	
Observaciones: Antes de generar un polinomio se debe inicializar el método congruencial mixto.	

Cuando existe una llave cargada en el componente, este debe permitir realizar el cifrado propuesto utilizando dicha llave, el cifrado debe poder realizarse sobre una plantilla en específica y además sobre las plantillas que contenga un directorio seleccionado:

Tabla 4: Historia de usuario cifrar plantillas

Elaboración propia.

Historia de Usuario	
Número: 2	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Cifrar plantillas	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 1	Iteración Asignada: 1
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El componente debe ser capaz de cifrar una plantilla de huellas dactilares, así como todas las plantillas en un directorio determinado, guardando los elementos cifrados en una ruta especificada.	
Observaciones: Antes de cifrar una plantilla se debe generar o cargar un polinomio.	

Para preparar a las plantillas para la comparación, el componente debe conformar las estructuras complejas de su conjunto de minucias protegidas, posteriormente debe proceder a realizar los cálculos de orientación:

Tabla 5: Historia de Usuario Extracción de las Plantillas Cifradas

Elaboración propia.

Historia de Usuario	
Número: 3	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Extracción de las plantillas cifradas	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 3	Iteración Asignada: 2
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El componente debe ser capaz de cargar una plantilla cifrada y formar las estructuras complejas a partir de las minucias que la componen, finalmente debe realizar los cálculos de orientación.	
Observaciones: Antes de cargar una Plantilla se debe cargar una llave.	

La siguiente historia de usuario hace referencia al proceso de comparación entre dos plantillas cifradas por el modelo propuesto:

Tabla 6: Historia de usuario comparar plantillas de minucias cifradas

Elaboración propia.

Historia de Usuario	
Número: 4	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Comparar plantillas de minucias cifradas	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 3	Iteración Asignada: 3
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El componente debe ser capaz de comparar plantillas de minucias cifradas e indicar el grado de similitud, también brindará la posibilidad de comparar directorios con varias plantillas de muestra para la investigación, calculando las tasas de error y mostrando información gráfica del resultado de la comparación según los umbrales de ajustes configurados.	
Observaciones:	

La siguiente historia de usuario recoge los requisitos que debe realizar el componente relacionados con el umbral de similitud, lo que permite encontrar un valor adecuado de este parámetro para el conjunto de plantillas de un directorio determinado.

Tabla 7: Historia de Usuario: Cálculo del umbral de similitud

Elaboración propia.

Historia de Usuario	
Número: 5	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Cálculo del umbral de similitud.	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 4
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El componente debe mostrar en una gráfica el valor óptimo de umbral de similitud para diferentes valores de umbrales de ajuste.	
Observaciones: El umbral coincide con la intercepción de las dos funciones (falsos aceptados y genuinos aceptados) en la gráfica.	

Finalmente, el componente debe ser capaz de dejar plasmado el resultado de las operaciones realizadas, por esta razón se redacta la siguiente historia de usuario:

Tabla 8: Historia de Usuario generar reportes

Elaboración propia.

Historia de Usuario	
Número: 6	Usuarios: Desarrolladores, investigadores
Nombre de la Historia: Generar reportes.	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 1	Iteración Asignada: 4
Programador Responsable: Yasmany Cruz Cordero	
Descripción: El Componente debe ser capaz de guardar los reportes de las operaciones realizadas.	

Observaciones: Los reportes se almacenarán en archivos de texto claro dentro de los directorios de Plantillas que se procesen, bajo el nombre informe.txt.

2.4.2 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. Un punto de estimación equivale a una semana de programación.

Tabla 9: Estimaciones de esfuerzo por Historias de Usuario

Elaboración propia.

Iteración	Historias de Usuario	Puntos Estimados (semanas)
1	1- Generar llaves	2
	2- Cifrar plantillas	
2	3- Extracción de las plantillas cifradas	3
3	4- Comparar plantillas de minucias cifradas	3
4	5- Cálculo del umbral de similitud.	3
	6- Generar reportes.	

2.4.3 Plan de entrega

El plan de entrega es el resultado de tomar acuerdos con el cliente sobre el contenido de la primera entrega y determinar un cronograma para las demás entregas del producto:

Tabla 10: Plan de Entrega

Elaboración propia.

Iteración	Fecha de entrega
Primera Iteración	1 de enero del 2016
Segunda Iteración	23 de enero del 2016

Tercera Iteración	18 de febrero del 2016
Cuarta Iteración	12 de marzo del 2016

2.5 Diseño

El diseño es el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o componente que resulte del mismo. La metodología XP sugiere que se debe diseñar de la forma más simple y sencilla posible para lograr que la aplicación sea entendible para su implementación. Un correcto diseño se enfoca en un modelo de sistema que cubra las necesidades inmediatas del cliente.

2.5.1 Definición de la arquitectura

La arquitectura de software es la organización fundamental de un sistema. Define los componentes que lo integran, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (Pressman, 2002).

La principal función del componente debe recibir como entrada una plantilla en texto claro, los datos de las plantillas deben ser sometidos al proceso de cifrado, realizando una transformación en el espacio de sus valores, este proceso origina la salida de los datos protegidos, las bases del componente operan con determinados flujos de datos, los cuales deben realizarse en un orden determinado mediante el procesamiento secuencial, por esta razón se selecciona la arquitectura de flujo de datos basada en el estilo arquitectónico de tuberías y filtros para el desarrollo del componente.

Características de la arquitectura flujo de datos basada en tuberías y filtros:

La arquitectura de flujo de datos se aplica cuando los datos de entrada son transformados a través de una serie de componentes computacionales o manipulativos en los datos de salida. Un patrón de tuberías y filtros tiene un grupo de componentes llamados filtros, conectados por tuberías que trasmiten datos de un componente al siguiente, cada filtro trabaja independientemente de aquellos componentes que se encuentran en el flujo de entrada o de salida; está diseñado para recibir la entra de datos de una cierta forma y producir una salida de datos (hacia el siguiente filtro) de una forma específica. Sin embargo, el filtro no necesita conocer el trabajo de los filtros vecinos (Pressman, 2002).

A continuación, se expone el diseño de la arquitectura:

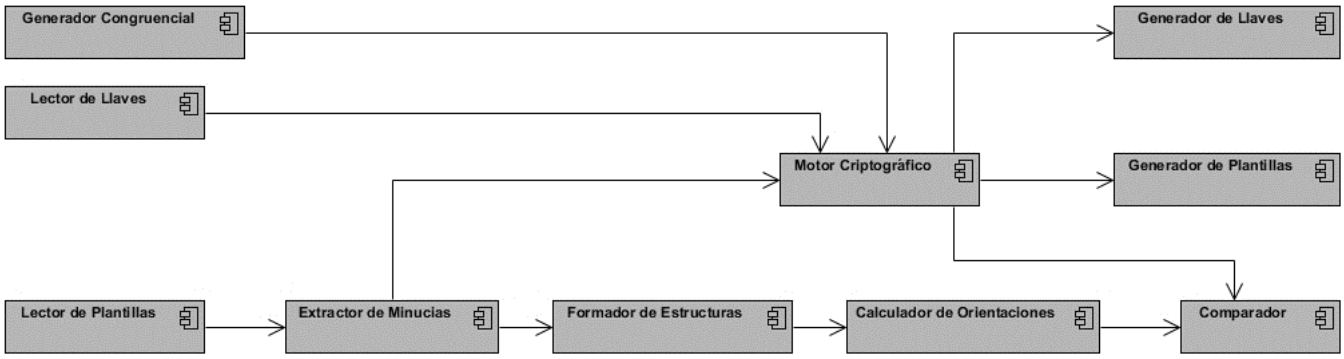

Figura 11: Arquitectura del componente

Elaboración propia.

Donde se evidencian tres flujos principales:

1- Obtención de las plantillas cifradas
2- Obtención del resultado de una comparación
3- Obtención de las llaves

La utilización de estos filtros facilita la simplicidad en la implementación del Componente, por lo que resulta mucho más sencillo cambiar un filtro determinado.

2.5.2 Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular (Pressman, 2002).

- **Experto en información**: Las responsabilidades deben ser asignadas a las clases que poseen la información para realizar dicha responsabilidad. Este patrón se evidencia en la clase que realiza la comparación ya que esta clase es la única que tiene acceso a los umbrales de ajuste y al conjunto de plantillas cargadas.
- **Creador**: Asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro del componente este patrón se evidencia en la clase que realiza la Extracción (*Template*) que es la encargada de crear las estructuras complejas (ISO_SEXTULE).
- **Alta cohesión**: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen

asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases. Este patrón se evidencia en las clases de Cifrado (*CryptographicEngine*) y en la clase encargada de la Comparación.

- **Bajo acoplamiento**: Determina el nivel de dependencia de una clase con respecto a otras. Una clase con bajo acoplamiento no depende de muchas otras. Este patrón es utilizado por cada una de las clases del componente ya que en la arquitectura utilizada facilita la poca dependencia entre clases.
- ***Singleton***: diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de tipo a un único objeto. Este patrón se evidencia en la clase de la Interfaz Visual (*MainWindow*) para realizar una solo instancia de un hilo de ejecución (*QThread*) que atienda los procesos complejos del componente y de esta manera independizar el hilo de la interfaz visual y el hilo de los procesos.

2.6 Tarjetas CRC

La metodología XP propone el uso de tarjetas CRC como alternativa a los diagramas UML de las clases, permitiendo al programador centrarse y apreciar el desarrollo orientado a objetos. En ellas se plasman las responsabilidades que tienen cada uno de los objetos, la clase a la cual pertenece y con las que colaboran con cada responsabilidad. A continuación, se muestra una de las tarjetas CRC generadas, en este caso la correspondiente a las plantillas de minucias (Template) que es la encargada de controlar las operaciones que se realizan a nivel de minucias y estructuras complejas, el resto de las tarjetas CRC se pueden encontrar en el Anexo A.

Tabla 11: Tarjeta CRC clase Template

Elaboración propia.

Tarjeta CRC	
Clase: Template	
Responsabilidad:	Colaborador:
Cargar una Plantilla	
Crear las Estructuras complejas	
Crear las Tripletas	
Calcular la Orientación de las Tripletas (ángulos interiores y longitud de los lados)	CryptographicEngine

Conclusiones parciales

1. El diseño de las tarjetas CRC posibilitó establecer un esquema orientado a objetos, para establecer las jerarquías y dependencias de cada una de las clases que se necesitan para desarrollar el componente.
2. El análisis de las funcionalidades propuestas para el componente facilitó la comprensión de las principales características y capacidades que debe poseer el componente.
3. La planificación del proyecto mejoró el acuerdo entre el cliente y los desarrolladores en cuanto a los tiempos de entregas del componente.
4. La definición de la arquitectura posibilitó definir la relación que existen entre los elementos que integran al componente, así como la interrelación entre los mismos.

Capítulo 3: Implementación y prueba

Como se ha podido constatar la protección de plantillas de minucias de huellas dactilares es uno de los temas investigativos más extensos dentro de la biometría computacional, para continuar con el hilo de esta investigación a continuación se reproducen algunos elementos asociados a la realización computacional del componente y se realiza un análisis de los resultados obtenidos a lo largo de este estudio, lo cual permite validar los objetivos de la investigación y pretende ofrecer algunas consideraciones importantes para continuar indagando en el tema.

3.1 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 este, 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 inglés.
- Las clases tendrán en mismo nombre que los ficheros que las contienen, el inicio de cada palabra que conforme su nombre se escribirá con mayúscula. Ejemplo: CryptographicEngine
- No se utilizará una misma línea para definir más de una variable y siempre que sea posible, estas se inicializarán en su misma línea de declaración.
- Los objetos y atributos se escribirán en minúsculas y utilizando el carácter '_' para los espacios entre las palabras. Ejemplo: "x_orientation".
- Las constantes se definirán en mayúsculas y se utilizará el carácter "_" como separador. Ejemplo: "IMAGE_HEIGHT".
- Los métodos o funciones se escribirán en minúsculas y se utilizará el carácter "_" como separador, los parámetros se escribirán con el mismo formato que los atributos de las clases. Ejemplo: "read_template(QString path)".

- En el caso de las funciones de lectura y escritura de cada parámetro se utilizarán los prefijos “get” y “set” respectivamente: Ejemplo: “getx_orientation()” y “setx_orientation(int new_x_orientation)”

3.2 Tareas de ingeniería

Las tareas de ingeniería son el artefacto enmarcado en la etapa de implementación y prueba que permite puntualizar cada detalle de la realización de las historias de usuario. Cada historia de usuario debe tener definida al menos una tarea de ingeniería y los programadores responsables de materializarla deben establecer además fechas de inicio y fin para dar cumplimiento a las descripciones planteadas, teniendo en cuenta los puntos de esfuerzo estimados para cada tarea de ingeniería. A continuación se muestra la tarea de ingeniería Generar Llaves, el resto se encuentra en el Anexo D.

Tabla 12: Tarea de Ingeniería generar llaves

Elaboración propia.

Tarea de Ingeniería	
Número: 1	Número de la HU: 1
Nombre de la Tarea: Generar llaves	
Fecha de Inicio: 1 de enero 2016	Fecha de Fin: 15 de enero 2016
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Yasmany Cruz Cordero	
Descripción: Debe permitir al usuario generar polinomios a partir de valores aleatorios suministrados por el método congruencial mixto, permitiendo además que el usuario realice ajustes en los parámetros de dicho método, una vez que el polinomio sea generado el usuario podrá guardarlo en un archivo llave para cargarlo con posterioridad.	

3.3 Experimentación

Para estimar el desempeño del componente teniendo en cuenta las tasas de error se decide optar por la utilización de algunas de las bases de datos más representativas a nivel internacional: *FVC2002* y *FVC2004*, las muestras que se encuentran en estas bases de datos se caracterizan por tener una calidad

entre baja y media lo cual constituye un factor importante para ser elegidas como objeto de pruebas de rendimiento biométrico.

Para la extracción de las plantillas de minucias de huellas dactilares de cada una de las bases de datos fue utilizado el SDK provisto por la empresa Innovatrics, líder a nivel internacional. Se utilizó el extractor de este SDK y se almacenaron las plantillas de minucias en texto claro. Como entradas del experimento se utilizan las plantillas de minucias de huellas dactilares y como salida se obtienen la cantidad de falsos aceptados, falsos rechazos, genuinos aceptados, genuinos rechazos y la tasa de error. El proceso sigue las pautas marcadas por la FVC para realizar pruebas de rendimiento biométrico y que se detallan en (Maltoni et al., 2009).

En la siguiente tabla se exponen los resultados alcanzados por el componente sobre las bases de datos:

Tabla 13: Error de FVC2002

Elaboración propia a partir de la experimentación con FVC2002.

Base de datos	Falsos aceptados	Falsos rechazos	Total de comparaciones	Taza de error
DB1_A	198	1298	79800	0.0025319 %
DB2_A	361	1264	79800	0.0046258 %
DB3_A	127	1342	79800	0.0016225 %

Tabla 14: Aceptación de FVC2002

Elaboración propia a partir de la experimentación con FVC2002.

Base de datos	Genuinos aceptados	Genuinos rechazos	Total de comparaciones
DB1_A	102	78202	79800
DB2_A	136	78039	79800
DB3_A	58	78273	79800

Para la realización del experimento se seleccionaron los siguientes parámetros de ajuste:

Grados de libertad de los lados de las tripletas (Gll) = 4%

Grados de libertad de los ángulos de las minucias (Gla): 5%.

Grados de libertad de los ángulos interiores de las tripletas ($Glat$): 5%.

Grados de validez de las tripletas (Gv): 130º.

Umbral de similitud de las estructuras complejas (Us): 30%.

Umbral de similitud de las Plantillas (PUs): 7.59%

Tabla 15: Error de FVC2004

Elaboración propia a partir de la experimentación con FVC2004.

Base de datos	Falsos aceptados	Falsos rechazos	Total de comparaciones	Taza de error
DB1_A	498	648	79800	0.0080265%
DB2_A	242	1370	79800	0.0045872 %
DB3_A	149	1338	79800	0.0027128 %

Tabla 16: Aceptación de FVC2004

Elaboración propia a partir de la experimentación con FVC2004.

Base de datos	Genuinos aceptados	Genuinos rechazos	Total de comparaciones
DB1_A	52	78202	79800
DB2_A	145	78044	79800
DB3_A	62	78251	79800

Para la realización del experimento se seleccionaron los siguientes parámetros de ajuste:

Grados de libertad de los lados de las tripletas (Gll) = 8%
Grados de libertad de los ángulos de las minucias (Gla): 6%.
Grados de libertad de los ángulos interiores de las tripletas ($Glat$): 8%.
Grados de validez de las tripletas (Gv): 130º.
Umbral de similitud de las estructuras complejas (Us): 30%.
Umbral de similitud de las Plantillas (PUs): 5.22%.

Para la ejecución de ambas pruebas se utilizó la siguiente configuración de hardware:

Microprocesador: Intel(R) Core(TM) i3-4000M CPU 2.40 GHz (4 Núcleos).
Memoria RAM: 4 Gb.

Para la estimación del error se utilizó la siguiente expresión:

$$ERR = [\left(\frac{FA}{C}\right) * 100]/[\left(\frac{GR}{C}\right) * 100]$$

Donde:

FA: cantidad de falsos aceptados

GR: cantidad de genuinos rechazos

C: cantidad de comparaciones realizadas

Con el objetivo de comparar el resultado obtenido en la presente investigación con resultados internacionales se realizó una prueba de rendimiento utilizando la base de datos de plantillas de minucias provista por los autores de la investigación. En la tabla 24 se muestran los resultados obtenidos por el componente implementado y los resultados publicados en (Tams, 2012) para la bóveda difusa. Como se puede apreciar el rendimiento biométrico es similar al modelo de bóveda difusa. El rendimiento decae ligeramente debido a los esquemas de comparación que utilizan ambos métodos los cuales son diferentes y en el caso de la bóveda difusa constituye una de las vulnerabilidades del modelo.

Tabla 17: Comparación de los resultados obtenidos

Elaboración propia a partir de los resultados obtenidos en (Tams, 2012).

Componentes de cifrado	Tasa de falso aceptado	Tasa de genuino aceptado
(Tams, 2012)	0.03	81.14
Componente desarrollado	0.04	79.65

3.3.1 Ataques de fuerza bruta

Para comprobar la fortaleza del componente y del método de protección implementados se realizan ataques de fuerza bruta como se proponen en (Tams, 2012). Para ello se toman como datos las bases de datos de la FVC y se realizan las comparaciones como se describe en el epígrafe 3.3. Como objetivo este experimento persigue la obtención de un polinomio que permita calcular los datos originales a partir de los datos cifrados.

El proceso de ataque de fuerza bruta comienza con la generación aleatoria de polinomios en un rango numérico específico. Se cifran las plantillas de minucias de huellas dactilares con un polinomio determinado y se realiza la comparación de las plantillas cifradas utilizando los polinomios generados. Como resultado se obtienen:

1. Porciento de coincidencias (Pc)
2. Cantidad de genuinos rechazados (Gr)

3. Cantidad de falsos rechazados (Fr)

Las plantillas a comparar fueron cifradas con el polinomio que se muestra a continuación:

$$F(X) = 6x^5 + 3x^4 + 4x^3 + x^2 + 2x + 7$$

Los resultados se muestran en la tabla 25.

Tabla 18: Resultados del ataque de fuerza bruta

Elaboración propia a partir de los ataques realizados al modelo.

Polinomios	Base de datos	Total de comparaciones	Tiempo	Pc	Gr	Fr
50	FVC 2002	239400	37 minutos	0	840	840
100	FVC 2004	239400	1 hora 3 minutos	0	840	840
200	FVC 2006	239400	2 horas 12 minutos	0	840	840

Como se puede apreciar en este experimento no fue posible encontrar un polinomio mediante el cual sea posible obtener los datos originales. Se pueden continuar generando polinomios o datos de manera aleatoria o siguiendo alguna distribución con el objetivo de encontrar un conjunto de datos que sean lo suficientemente cercanos a los originales. No se comprueba la seguridad criptográfica con el resto de los ataques ya que el modelo de protección seleccionado no es vulnerable a los mismos.

3.4 Pruebas de Aceptación

Las pruebas de aceptación o pruebas funcionales son definidas por el cliente y son esenciales en su satisfacción con el producto desarrollado. Constituyen el fin de una iteración y el comienzo de la siguiente que asumirá la corrección de los errores encontrados.

Al componente se le realizaron las pruebas a cada historia de usuario, con el objetivo de determinar irregularidades y lograr la estabilidad, a continuación, se muestra la prueba de aceptación realizada a la

historia de usuario: Generar Llaves (Consultar el resto de los casos de prueba en el Anexo Pruebas de Aceptación).

Tabla 19: Caso de prueba HU1CP1

Elaboración propia.

Caso de Prueba de Aceptación	
Código caso de prueba: HU1CP1	Nombre HU: Generar llaves
Responsable de la prueba: Yasmany Cruz Cordero	
Descripción de la prueba: Prueba para verificar que se generen, se guarden y se carguen archivos llaves con polinomios generados	
Condiciones de ejecución: La aplicación no puede tener errores de ningún tipo en su ejecución.	
Entrada / Pasos de ejecución: "Escenario: generar llave." 1. Se accede a la pestaña de "Cifrado". 2. Se inicializa el método generador congruencial de valores aleatorios. 3. Se ejecuta la acción: "Generar Polinomio". "Escenario: guardar llave." 1. Se accede a la pestaña de "Cifrado". 2. Se genera una llave. 3. Se ejecuta la acción: "Guardar Llave". 4. Se selecciona la ruta y el nombre de la llave a guardar. "Escenario: cargar llave." 1. Se accede a la pestaña de "Cifrado". 2. Se ejecuta la acción: "Cargar Llave". 3. Se selecciona la ruta y el archivo llave a cargar.	
Resultado esperado: "Escenario: generar llave." Se genera en la interfaz visual un polinomio del grado seleccionado. "Escenario: guardar llave."	

Se genera un archivo llave que almacena la información de un polinomio. “Escenario: cargar llave.” Se carga el polinomio almacenado en el archivo llave en la interfaz visual
Evaluación de la Prueba: Satisfactoria

3.5 Pruebas unitarias

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los elementos que se someterán a las pruebas unitarias arrojarán un resultado que servirá para valorar la correcta implementación de un elemento determinado dentro del componente. Para realizar las pruebas unitarias el equipo de desarrollo se auxilió de un módulo para la monitorización de la memoria RAM y el microprocesador, estos dos recursos son los más demandados por el componente.

A continuación, se muestra una figura con los resultados obtenidos en cuanto a los recursos al realizar más de 10000 comparaciones:

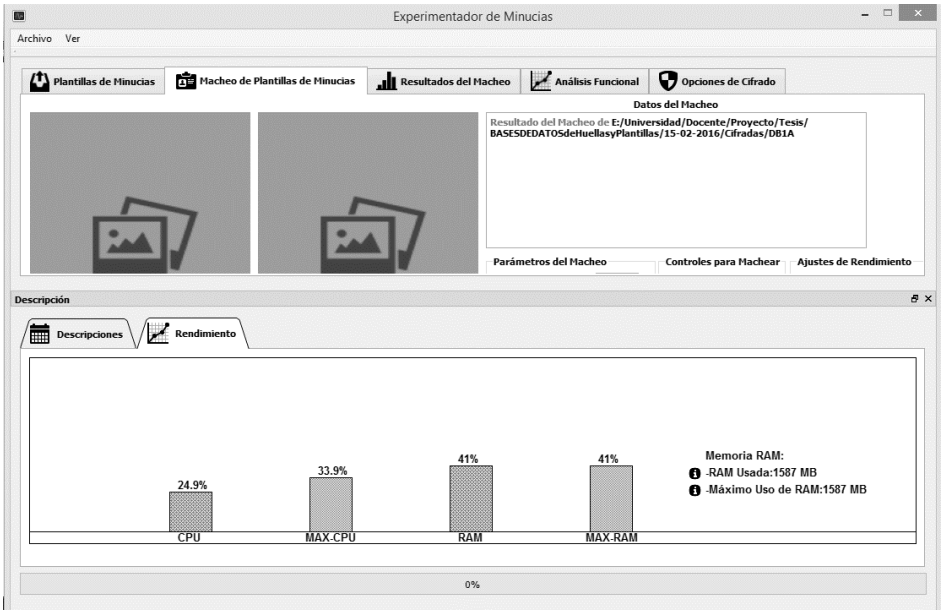

Figura 12: Recursos de hardware consumidos por el componente

Elaboración propia a partir de una vista del componente.

Como se puede apreciar en la figura los resultados de la utilización de los recursos de la computadora donde fue ejecutada la prueba son:

1. Uso de la CPU 24.9 %.
2. Uso de la memoria RAM 41%

Estos valores reflejan el rendimiento necesario para ejecutar comparaciones en bases de datos de mediano tamaño. Las pruebas realizadas al componente de manera general arrojan buenos resultados, facilitando el proceso de protección de plantillas de minucias de huellas dactilares y su comparación en el dominio protegido.

Conclusiones parciales

- La utilización de bases de datos internacionales para la ejecución de experimentos de rendimiento biométrico facilitó la validación de los resultados obtenidos en la investigación.
- Las pruebas de rendimiento realizadas al componente desarrollado mostraron que puede ser utilizado en un sistema automático de identificación de personas mediante huellas dactilares para cifrar los datos biométricos.
- La ejecución de pruebas unitarias al componente desarrollado proporcionó datos del funcionamiento del componente, corroborando el correcto funcionamiento del mismo.

Conclusiones generales

- El análisis de los elementos teóricos-metodológicos más actuales en el campo de la protección de plantillas de minucias de huellas dactilares, facilitó la selección de un modelo de protección libre de vulnerabilidades para ser implementado.
- El análisis de la seguridad criptográfica de los modelos y componentes analizados permitió determinar los ataques a realizar en el componente de protección desarrollado.
- La selección de tecnologías, metodologías y herramientas como parte de la investigación facilitó el proceso de desarrollo del componente y el entendimiento entre los miembros del equipo.
- Las pruebas realizadas al componente mostraron una disminución en el rendimiento biométrico que no se considera considerable comparada con el aumento en la seguridad criptográfica.
- Los ataques de fuerza bruta realizados al componente desarrollado mostraron que no es factible computacionalmente obtener un polinomio mediante el cual calcular los datos originales de la plantilla cifrada.

Recomendaciones

- Integrar el componente con un componente de extracción de características de minucias de huellas dactilares.
- Extender el componente para el cifrado de otros identificadores biométricos.

Referencias Bibliográficas

Ahn, D., Kong, S. G., Chung, Y., & Moon, K. Y. (2008). Matching with Secure Fingerprint Templates using Non-invertible Transforms. In *2008 Congress on Image and Signal Processing Matching* (pp. 29–33). doi:10.1109/CISP.2008.742

Arakala, A., & Jeffers, J. (2006). Minutiae-Based Structures for a Fuzzy Vault. In *Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference* (pp. 1–6).

Arakala, A., Jeffers, J., & Horadam, K. J. (2007). Fuzzy Extractors for Minutiae-Based Fingerprint Authentication. In *Advances in Biometrics* (pp. 760–769).

Beck, K. (1999). Extreme Programming Explained. Embrace Change. *Pearson Education*.

Beedle, M., Beck, K., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifiesto Ágil. *Jim Highsmith*.

Belguechi, R., Cherrier, E., Rosenberger, C., & Ait-aoudia, S. (2013). Operational bio-hash to preserve privacy of fingerprint minutiae templates. *IET Biometrics*, (February), 1–9. doi:10.1049/iet-bmt.2012.0039

Belguechi, R., Rosenberger, C., & Aoudia, S. A. (2010). BioHashing for securing fingerprint minutiae templates. In *2010 International Conference on Pattern Recognition* (pp. 1172–1175). doi:10.1109/ICPR.2010.292

Bolle, R. M., Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001). Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, *40*(3), 614–634.

Cappelli, R., Lumini, A., Maio, D., & Maltoni, D. (2007). Fingerprint Image Reconstruction from Standard Templates. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, *29*(9), 1489–1503.

Clancy, T. C., Kiyavash, N., & Lin, D. J. (2003). Secure Smartcard-Based Fingerprint Authentication.

Coad P., Lefebvre E., D. L. J. (1999). Java Modeling In Color With UML: Enterprise Components and Process. *Prentice Hall*.

Cockbun, A. (2001). Agile Software Development. *Addison-Wesley*.

Dahiya, N., & Kant, C. (2012). Biometrics Security Concerns. In *Second International Conference on Advanced Computing & Communication Technologies Biometrics* (pp. 299–304). doi:10.1109/ACCT.2012.36

Fernández, R. S., Cento, A. A. M., & Sentí, V. E. (2015). Extracción de características identificativas en plantillas de minucias mediante la estructura compleja. *Revista Cubana de Ciencias Informáticas*, *9*(4), 132–141.

Figueroa, R. G. (2004). METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.

Gobi, M., & Kannan, D. (2014). A Secured Public Key Cryptosystem for Biometric Encryption. *International Journal of Computer Science and Information Technologies*, *5*(1), 184–191.

Highsmith, J. (2002). Agile Software Development Ecosystems. *Addison-Wesley*.

Highsmith J., O. K. (2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. *Dorset House*.

Jain, A. K., Nandakumar, K., & Nagar, A. (2008). Biometric Template Security, *2008*. doi:10.1155/2008/579416

Jain, A. K., Nandakumar, K., & Nagar, A. (2013). *Fingerprint Template Protection : From Theory to Practice*. *Security and Privacy in Biometrics* (pp. 1–29).

Jain, A. K., Prabhakar, S., Hong, L., & Pankanti, S. (1999). FingerCode: a filterbank for fingerprint representation and matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 2, p. 8).

Jeffers, J., & Arakala, A. (2007). FINGERPRINT ALIGNMENT FOR A MINUTIAE-BASED FUZZY VAULT. In *Biometrics Symposium*.

Jin, Z., Goi, B.-M., Teoh, A., & Tay, Y. H. (2014). A two-dimensional random projected minutiae vicinity decomposition-based cancellable fingerprint template. *Security and Communication Networks*, *7*(11), 1691–1701. doi:10.1002/sec

Juels, A., & Sudan, M. (2002). A Fuzzy Vault Scheme. In *IEEE International Symposium on Information Theory* (p. 408).

Juels, A., & Wattenberg, M. (1999). A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security* (pp. 28–36).

Kholmatov, A., & Yanikoglu, B. (2008). Realization of correlation attack against the fuzzy vault scheme. In *Electronic Imaging* (pp. 68190–68197).

Kumar, R. (2014). Vulnerability to Fingerprint Biometric Systems- An Overview. *IJCSC*, *5*(1), 109–115.

Lacharme, P., Cherrier, E., & Rosenberger, C. (2014). Preimage Attack on BioHashing. In *International Conference on Security and Cryptography*.

Lee, C., & Kim, J. (2010). Cancelable fingerprint templates using minutiae-based bit-strings. *Journal of Network and Computer Applications*, *33*(3), 236–246. doi:10.1016/j.jnca.2009.12.011

Lee, S., Moon, D., Choi, W. Y., & Chung, Y. (2008). Analysis of Tradeoffs among Verification Accuracy , Memory Consumption , and Execution Time in the GH-based Fuzzy Fingerprint Vault, 0–3. doi:10.1109/SecTech.2008.25

Li, J., Yang, X., Tian, J., Shi, P., & Li, P. (2008). Topological Structure-based Alignment for Fingerprint Fuzzy Vault, (1), 6–9.

Li, Q., Guo, M., & Chang, E.-C. (2008). Fuzzy Extractors for Asymmetric Biometric Representations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08.* (pp. 1–6).

Maltoni, D., Maio, D., Jain, A. K., & Salil Prabhakar. (2009). *Handbook of Fingerprint Recognition* (p. 506).

Marcelo Gabriel, Ana Lía Ramona, Carlos Eduardo, M. A. (2015). Selección de Lenguajes Orientados a Objetos para un estudio comparativo y análisis de rendimiento.

Merkle, J., & Tams, B. (2013). Security of the Improved Fuzzy Vault Scheme in the Presence of Record Multiplicity, (0), 1–40.

Nagar, A., Nandakumar, K., & Jain, A. K. (2010). A hybrid biometric cryptosystem for securing fingerprint minutiae templates q. *Pattern Recognition Letters*, *31*(8), 733–741. doi:10.1016/j.patrec.2009.07.003

Nandakumar, K. (2010). A Fingerprint Cryptosystem Based on Minutiae Phase Spectrum, 2–7.

Nandakumar, K., Jain, A. K., & Pankanti, S. (2007). Fingerprint-Based Fuzzy Vault : Implementation and Performance. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, *2*(4), 744–757.

Poppendieck M., P. T. (2003). Lean Software Development: An Agile Toolkit for Software Development Managers. *Addison Wesley*.

Pressman, R. S. (2002). *Ingienería de Software: Un enfoque práctico*.

Quan, F., Fei, S., Ann, C., & Feifei, Z. (2008). Cracking Cancelable Fingerprint Template of Ratha. In *International Symposium on Computer Science and Computational Technology* (pp. 572–575). doi:10.1109/ISCSCT.2008.226

Ratha, N., Connell, J., & Bolle, R. M. (2006). Cancelable Biometrics : A Case Study in Fingerprints, 18–21.

Ratha, N. K., Chikkerur, S., Connell, J., Bolle, R. M., & Member, S. (2007). Generating Cancelable Fingerprint Templates. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, *29*(4), 561–572.

Rathgeb, C., & Uhl, A. (2011). A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, *2011*(1), 3. doi:10.1186/1687-417X-2011-3

Roja, M. M., & Sawarkar, S. (2013). ElGamel Encryption for Biometric Database Protection. *International Journal of Computer Applications*, *68*(6), 10–14.

Rozsa, A. (2014). *ATTACK ON MINUTIAE-BASED FINGERPRINT AUTHENTICATION SYSTEMS BY USING GENETIC ALGORITHM*.

Scheirer, W. J., & Boult, T. E. (2007). CRACKING FUZZY VAULTS AND BIOMETRIC ENCRYPTION. In *Biometrics Symposium* (pp. 1–6). Baltimore, MD: IEEE.

Schwaber K., Beedle M., M. R. C. (2001). Agile Software Development with SCRUM. *Prentice Hall*.

Security, H. F. (2012). *Hash function security : Cryptanalysis of the Very Smooth Hash and multicollisions in generalised iterated hash functions*.

Stapleton J. (1997). Dsdm Dynamic Systems Development Method: The Method in Practice. *Addison-Wesley*.

Stroustrup, B. (1997). The C+ + Programming Language. *AddisonWesley*.

Tams, B. B. (2012). *Cryptanalysis of the Fuzzy Vault for Fingerprints : Vulnerabilities and Countermeasures*.

Teoh, A., Ngo, D., Ling, C., & Goh, A. (2004). Biohashing : two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, *37*, 2245–2255. doi:10.1016/j.patcog.2004.04.011

Uludag, U. (2006). Securing Fingerprint Template : Fuzzy Vault with Helper Data. In *2006 Conference on Computer Vision and Pattern Recognition Workshop*.

Yang, H., Jiang, X., & Kot, A. C. (2009). Generating Secure Cancelable Fingerprint Templates Using Local and Global Features | not exist otherwise, 0–4.

Yang, W., Hu, J., & Wang, S. (2014). A Delaunay Quadrangle-Based Fingerprint Authentication System with Template Protection Using Topology Code for Local Registration and Security Enhancement. *IEEE Transactions on Information Forensics and Security*, *9*(7), 1179–1192.

Zhang, X., Feng, Q., & He, K. (2014). A New Blind Fingerprint Alignment Algorithm used in Biometric Encryption. In *International Conference on Computer, Communications and Information Technology (CCIT 2014)* (Vol. 1, pp. 231–234).

Zhe, J., & Beng Jin, A. T. (2011). Fingerprint Template Protection with Minutia Vicinity Decomposition. In *International Joint Conference on Biometrics* (pp. 1–7).