Biometría Cancelable basada en funciones físicas inclonables en FPGA para señales ECG

por

Ing. Diana Karen Torres Flores

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE ELECTRÓNICA

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
Diciembre 2020
Tonantzintla, Puebla

Supervisada por:

Dr. José de Jesús Rangel Magdaleno
Investigador Titular INAOE

Agradecimientos

Le agradezco al INAOE por permitirme realizar mis estudios de maestría en sus instalaciones y al Conacyt por la beca otorgada para que esto fuera posible.

Al Dr. José de Jesús por abrirme las puertas desde que realicé mis practicas profesionales, por permitirme ser parte de su grupo de trabajo y por su guía para la realización de esta tesis.

A mis compañeros de laboratorio Carlos, Sergio, Omar y Julio por siempre echarme una mano cuando los necesité y sus consejos.

A mis compañeros de maestría por hacer de esta una experiencia grata pero en especial a Moshis, Salguero, Coto y Tabasco quienes me permitieron conocerlos más allá de lo academico y con quienes hice una bonita amistad.

Y a todas esas personas que formaron parte de mi estancia en el INAOE y de los que me llevó un buen recuerdo.

Dedicatoria

Este trabajo se lo dedico a mis padres, José Luis y Alicia, mi mayor ejemplo de superación y perseverancia. Gracias a ustedes soy lo que soy y estoy donde estoy.

A mi familia por apoyarme en cada decisión que he tomado y recordarme que siempre estan ahí para mi.

A mis amigos y a las personas que me acompañaron y que hicieron posible la realización de esta tesis.

Y a Mayita, por formar parte de esto, por su compañia incondicional en mis días de soledad y de desvelos y por ser la alegria de mis días.

Resumen

Los sistemas basados en rasgos biométricos han tomado ventaja dentro del campo de los sistemas de control de acceso contra los basados en métodos tradicionales tales como claves o tarjetas de identificación debido a la disminución de la probabilidad de falsificación de información. Sin embargo, el robo de información en los sistemas biométricos presenta un problema mayor que en los sistemas de control de acceso convencionales, ya que los rasgos biométricos no pueden ser reemplazables. Para solucionar esto se aplica una distorsión en el patrón de un usuario para proteger la información sensible. Esta transformación se denomina Biometría Cancelable.

En este trabajo se presenta un esquema de biometría cancelable para señales electrocardiográficas (ECG) utilizando funciones físicas inclonables (PUF) implementadas en FPGA. La técnica que se utilizó para encriptar la información fue con la generación de cadenas de bits obtenidas por la PUF en configuración de RingOscillator que es un diseño exclusivo para implementación en FPGA.

Las PUF's se evaluaron en dos FPGA's del mismo modelo para verificar que se obtuvieran respuestas diferentes implementando el sistema bajo las mismas condiciones, se analizaron las respuestas y se aplicaron algoritmos de corrección de errores para corregir los bits de respuesta inestables.

Se emplearon técnicas de extracción de características propuestas en la literatura donde se obtienen características estadísticas y características temporales formando vectores de longitud 7 y como clasificador se utilizó el algoritmo de alineamiento temporal dinámico (DTW).

Se evaluó el sistema biométrico original y posteriormente con las plantillas modificadas. Los resultados obtenidos del sistema propuesto para el mejor test indicaron un AUC de 98.32 %, un error de 7.12 % y una exactitud de 92.8 % superando los resultados de sistemas basados en biometría cancelable para señales ECG utilizando técnicas de Bio-Hash, improved Bio-Hash, operaciones de matrices y redes neuronales.

Tabla de Contenido

Capítulo 1

Introducción

La biometría nació por la necesidad de volver más seguros los sistemas de identificación-autenticación de personas al hacer uso de rasgos biológicos cuyas características son únicas en cada individuo. Sus inicios se presentaron desde siglos atrás cuando se median algunas extremidades de las personas para identificarlas o cuando se realizaba con fotografías o imagenes en credenciales; sin embargo, estos métodos se realizaban de manera manual a través de la memoria fotográfica. Con el tiempo estos procesos se automatizaron gracias al desarrollo tecnológico y en la actualidad se hace uso de varios rasgos biométricos como la huella dactilar, la retina, la palma de la mano, la voz, las señales electrocardiográficas, entre otros.

Un sistema biométrico realiza la identificación-autenticación de un individuo a través de una comparación que realiza con una base de datos. El hecho de almacenar información relevante en una base de datos es un factor de alto riesgo, ya que puede ser robada. A diferencia del uso de credenciales o contraseñas, los rasgos biométricos son permanentes y no pueden ser reemplazados.

Para evitar el robo de patrones biométricos se han hecho investigaciones en el campo de la biometría y la criptoinvestigación para desarrollar esquemas de protección. Entre estos esquemas se encuentra la biometría cancelable.

En la biometría cancelable se busca modificar una plantilla biométrica original con la ayuda de otras plantillas revocables y no invertibles para producir plantillas biométricas cancelables. Esta transformación puede ser en el dominio original o en el dominio de las características.

Un esquema de biometría cancelable proporciona revocabilidad, ya que si algún dato biométrico se ve comprometido puede volver a registrarse mediante otra transforma-

ción. Conserva privacidad ya que es computacionalmente difícil recuperar el sistema biométrico original de uno transformado. Evita las coincidencias cruzadas entre bases de datos, ya que cada aplicación utiliza una transformación diferente. Y no degrada la precisión de un algoritmo de coincidencia, ya que las características estadísticas de las funciones se mantienen aproximadamente después de la transformación. Esto permite utilizar los algoritmos de coincidencia existentes [1]. Existen diversas técnicas para generar estas plantillas que sirven para modificar la plantilla original, entre estas están los generadores de números aleatorios. En este trabajo se propusó utilizar las funciones físicamente inclonables.

Una función físicamente inclonable o PUF *(Physical Unclonable Function)* es una función embebida dentro de una estructura física que es sencilla de evaluar (cuando se tiene acceso al dispositivo que se evalúa) pero casi imposible de predecir o de duplicar. La respuesta que brinda la PUF depende de la complejidad y aletoriedad física del circuito integrado y es dependiente del dispositivo de prueba.

A través de la respuesta de la PUF se logra encriptar la plantilla biométrica realizando una operación lógica entre ellos. Cabe mencionar que en la literatura se suele utilizar una respuesta por dispositivo, en este caso se obtuvieron al menos 50 respuestas por dispositivo obteniendo buenos resultados.

1.1. Justificación

Establecer esquemas de seguridad en los sistemas de control de acceso para proteger los datos biométricos es indispensable debido a la rápida adaptación que han tenido los sistemas biométricos en campos de continuo ataque como bancos, negocios, instituciones entre otros. El hecho de perder tan sólo un rasgo biométrico es suficiente para que el impostor pueda acceder no sólo al sistema que atacó, también a otros sistemas que empleen el mismo rasgo biométrico.

Por esto es importante un sistema que además de brindarle seguridad a la institución que lo implementa le brinde seguridad a los individuos registrados en el mismo.

Debido a esto se propone un esquema de biometría cancelable utilizando las respuestas que generan las funciones físicas inclonables implementadas en FPGA, donde se aprovecha la dificultad de volver a generar la misma respuesta en dispositivos diferentes a pesar de tratarse del mismo modelo.

También se propone el uso de señales ECG debido a la estabilidad y buen rendimiento de sus propiedades comparado con otros rasgos biométricos establecidas en [2].

1.2. Objetivos

1.2.1. Objetivo general

Diseñar un sistema de control de acceso en modo de verificación basado en biometría cancelable para señales ECG utilizando funciones físicas inclonables.

1.2.2. Objetivos especificos

- Implementar en FPGA la función física inclonable en modo configurable ring-oscillator.
- Implementar los algoritmos de filtros digitales FIR y Wavelet para eliminar el ruido externo de la señal.
- Implementar los algoritmos para la extracción de características estadísticas y temporales de la señal ECG.
- Implementar el algoritmo del alineamiento temporal dinámico como clasificador (DTW)
- Encriptar la plantilla biométrica original usando la respuesta de la PUF.
- Mantener o mejorar el rendimiento del sistema biométrico después de la cancelación de plantillas.

Capítulo 2

Estado del arte

En la literatura se pueden encontrar esquemas de biometría cancelable para algunos rasgos biométricos donde se emplean diversas técnicas para la creación de las plantillas de cancelación; entre estos se encuentran los trabajos donde se emplean señales electrocardiográficas.
A continuación se enlistan los trabajos donde se han empleado señales ECG y se describe brevemente su metodología de cancelación de plantillas:

- *BioHash Code Generation from Electrocardiogram Features* [3]: Presentan un enfoque de biometría cancelable llamado BioHashing. El código BioHash se obtiene a través del producto interno de la función de características temporales obtenidas del ECG y un número aleatorio tokenizado, los valores que esten por encima de un umbral se definen como 1 y el resto como 0.

- *A Random Walk-Based Cancelable Biometric Template Generation [4]*: En este artículo utilizan la aleatoriedad del concepto de camino aleatorio (Random Walk) para generar una plantilla de huellas dactilares no invertible. Con esta técnica generan múltiples vectores definiendo el número de objetos y los pasos basandose en el tamaño de la plantilla de características.

- *Cancelable Biometrics Using Deep Learning as a Cloud Service [5]*: Proponen un sistema biométrico cancelable basado en Deep Learning en la nube.

- *Novel approach for multimodal feature fusion to generate cancelable biometric [6]*: En este trabajo se propone un sistema biométrico multimodal basado en

proyecciones. El enfoque propuesto genera una característica biométrica cancelable que se utiliza posteriormente para obtener plantillas revocables y no invertibles a través de proyecciones en un plano aleatorio obtenido mediante una clave específica de usuario. Para medir el rendimiento de este esquema se utilizarón tres bases de datos.

- *Cancelable ECG Biometrics using Compressive Sensing-Generalized Likelihood Ratio Test* [7]: Proponen una técnica que utiliza una prueba de razón de probabilidad generalizada (GLRT) basada en una prueba de hipótesis compuesta en el dominio de detección de compresión (CS). Sus métodos biométricos arrojaron hasta un 93 % de probabilidad de detección con un 2 % de índice de falsas alarmas.

- *Tripe C: A New Algorithm for ECG Cancelable Biometric System [8]*: Este artículo utiliza un algoritmo llamado CCC o Triple C. Se utilizan dos señales que se cambian utilizando corrimientos y se encriptan por el algoritmo Cepstrum, posteriormente se convolucionan. Luego se almacena en una base de datos autorizada. Sus estudios involucran a 46 individuos.

- *A PUF-and Biometric-based Lightweight Hardware Solution to Increase Security at Sensor Nodes [9]*: En este trabajo presentan una técnica de reconocimiento de huellas dactilares ofuscadas con funciones físicamente inclonables y con la identidad del nodo sensor. Esto permite autenticar el origen de los datos detectados con un protocolo de autenticación de doble factor propuesto. Un factor es la identidad física del nodo-sensor físico que mide los datos y la otra es la identidad del usuario. Utilizan QFingerMap 16 (QFM) para hacer el reconocimiento de características basado en texturas.

- *Cancelable Biometric authentication system based on ECG* [2]: En este artículo se proponen dos técnicas biométricas cancelables. El primero es un Bio-Hashing mejorado y el segundo es una técnica de operación matricial. Se utilizan 3 bases de datos distintas y redes neuronales prealimentadas (Feedforward neural network) para verificar a las personas.

- *Cancelable ECG Biometrics using GLRT and Performance Improvement using Guided Filter with Irreversible Guide Signal* [10]: En este trabajo proponen

una biometría de ECG cancelable mediante la derivación de un detector de prueba de razón de verosimilitud generalizada (GLRT) a partir de una prueba de hipotesis compuesta en un dominio proyectado aleatoriamente. Este método se evaluó usando la base de datos ECG-ID con 89 sujetos obtiendo un AUC de 77.5 %.

- *ECG-based biometrics using recurrent neural networks* [11]: En este trabajo evaluan diferentes arquitecturas de redes neuronales recurrentes con varias configuraciones de parámetros, incluyendo las redes tradicionales, de memoria a largo plazo a corto plazo (LSTM), unidad recurrente cerrada (GRU), unidireccionales y bidireccionales. La principal diferencia con este método es que no se extraen características.

- *A system of biometric authentication based on ECG signal segmentation* [12]: Se propone un método basado en convolución para la extracción de latidos y un método basado en en formas de onda para la segmentación de latidos. Se utilizó la base de datos MIT-BIH de sujetos con arritmias no significativas.

En la tabla 2.1 se muestran trabajos donde se han utilizado señales ECG así como las técnicas que utilizaron y su rendimiento, donde AUC indica el área bajo la curva ROC y el EER indica la tasa de igual error.

Tabla 2.1: Estudio del arte de sistemas basados en biométria cancelable.

Autor	Año	Base de datos	Rasgo Biométrico	Técnica	Rendimiento
Pandey	2020	FVC 2004	Huella dactilar	Generación aleatoria de matrices usando Random Walk	EER=0.05
Sudhakar	2020	IIT-D MMU FV-USM	Iris	Deep Learning	IIT-D EER= 0.12 MMU EER= 0.15 FV-USM EER=0.05
Gupta	2020	MCYT bimodal IITD PolyU iris Casia iris FVC2006 DB1-A MMU2 iris	Iris Huella Dactilar	Fusión de características	MCYT & IITD PolyU EER= 0.005 MCYT & Casia iris EER= 0.003 FVC & MMU2 EER= 0.004
Kim	2019	ECG-ID	ECG	GLRT Compressive Sensing	EER=0.054
Shouman	2019	MIT-BIH arrhythmia	ECG	Algoritmo Triple C	AUC= 99
Arjona	2018	FVC 2002 DB1a FVC 2000 DB2a	Huella dactilar	PUF	FVC 2000 DB2a EER= 0.04 FVC 2002 DB1a EER= 0.22
Hammad	2018	MIT-BIH PTM CYBHi	ECG	Improved Bio-Hashing Matrix Operations	MIT-BIH EER=0.34 (Improved) EER=0.06 (Matrix) PTB EER=0.32 (Improved) EER=0.14 (Matrix) CYBHi EER=0.17 (Improved) EER=0.09 (Matrix)
Kim	2017	ECG-ID Database	ECG	GLRT	EER=0.302 AUC=77.5
Karegar	2017	MIT-BIH	ECG	RSA HFD DFA GHE RQA	EER=4.88
Salloum	2017	MIT-BIH	ECG	RNNs	EER=3.5
Keshishzade	2015	MIT-BIH	ECG	FFS	EER=2.34 AUC=99.73
Propuesto	**2020**	**BIDMC PPG and Respiration Dataset**	**ECG**	**Bio-Hashing** **PUF**	**EER=0.071** **AUC= 98.32**

Capítulo 3

Marco teórico

3.1. Biometría

La biometría es una técnica de reconocimiento de personas que aprovecha las características fisiológicas y de comportamiento para identificar positivamente a una persona, basándose en la comprobación científica de que existen elementos en las estructuras vivientes que son únicos e irrepetibles para cada individuo [13].
Cualquier método biométrico debe ser capaz de medir, codificar, comparar, almacenar y reconocer la característica propuesta de un individuo con cierto grado de exactitud.
Estas técnicas se han desarrollado desde hace algunos años como las más efectivas para la identificación humana y el éxito del reconocimiento esta basado en la característica que se mide y el método propuesto. Incluso se han propuesto modelos donde se fusionan dos o más características, de este modo se tiene mejor rendimiento y las probabilidades de usurpación de identidad disminuyen.
Hoy en día se han implementado en múltiples áreas, reemplazando los sistemas de verificación basados en tarjetas y contraseñas.

3.1.1. Características de un sistema biométrico

Entre las características físicas más populares para reconocimiento se encuentran la huella dactilar, la imagen facial, la retina u alguna otra señal fisiológica mientras que las características de comportamiento son la voz, la forma de caminar o la firma. De estos parámetros se obtiene un patrón único que se almacena en una base de datos para posteriormente comparar la información.
La característica que se emplee debe cumplir con las siguientes propiedades:

- Universalidad: todos los individuos la tienen.
- Singularidad o univocidad: distinguen a cada individuo.
- Permanencia en el tiempo y en distintas condiciones ambientales.
- Medibles de forma cuantitativa.

Y el método para medir estas características debe proporcionar:

- Rendimiento: Nivel de exactitud.
- Aceptación: por parte del usuario.
- Resistencia al fraude y usurpación

3.1.2. Funcionamiento de un sistema biométrico

De manera general, un sistema biométrico consta de varias fases que se muestran en la Figura 3.1.

A continuación se describe brevemente en que consiste cada fase:
1. Adquisición de la información: Todos los sistemas deben comenzar con la medida de alguna señal o característica de comportamiento o fisiológica. La adquisición de estos datos normalmente proviene de sensores. Entre mejor este desarrollado el

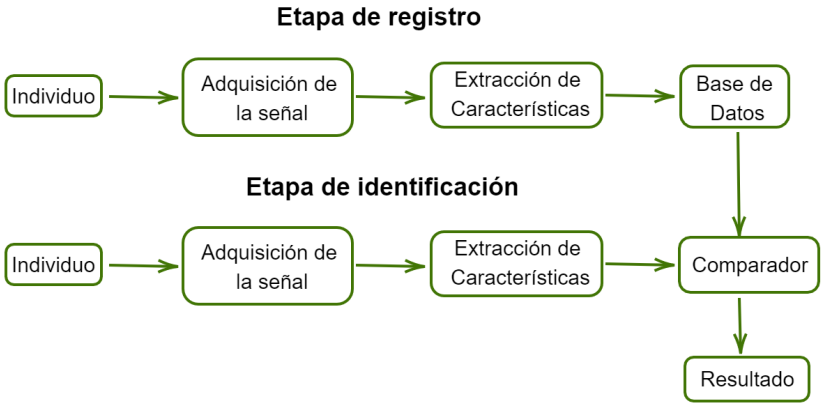

Figura 3.1: Etapas de registro e identificación en un sistema biométrico.

bloque de adquisición podemos evitar que la información adquirida se vea afectada con ruido o componentes indeseables.

2. Transmisión de información: Muchos sistemas implementan los bloques de adquisición de datos en una ubicación mientras el procesamiento de la información lo realizan en otro software. Dichos sistemas requieren transmitir su información mediante diferentes protocolos. La transmisión no sólo puede ocurrir de la adquisición de datos hacía los demás bloques, también durante cualquier punto del proceso.

3. Extracción de características: Aquí es donde se procesa la señal para obtener y conservar la cualidad que se desea analizar y se desecha la información restante.

4. Base de datos: Todos los sistemas deben contar con una base de datos donde se almacenan las características significativas de todos los usuarios para posteriormente acceder a ella.

5. Comparador: Para la decisión se utiliza un comparador, la política del sistema de decisión dirige la búsqueda en la base de datos, y determina los m̈atchingö los n̈o-matching"basándose en las medidas de la distancia recibidas [13].

Este subsistema toma en última instancia una decisión de acepta-rechaza basada en la política del sistema. Tal política podría ser declarar un "matching" para cualquier distancia más baja que un umbral fijo, o la política podría ser declarar un "matching" para cualquier distancia más baja que un umbral dependiente del usuario [13].

Modos de operación

Una vez que se tiene el registro en la base de datos se puede hacer el proceso de autenticación, donde se obtiene la muestra biométrica del usuario que se comparará con las muestras ya almacenadas. Este proceso se puede realizar de dos modos diferentes:

- Modo de identificación: Se basa en comparar la muestra obtenida del individuo contra todas las muestras previamente almacenadas en la base de datos hasta reconocer al usuario que se esta analizando. Este modo requiere de un proceso de cálculo más complejo debido a que se compara con todos los individuos.
- Modo de verificación: Primero, se debe identificar al usuario mediante algún factor externo, puede ser una tarjeta de identificación o algún otro método. Una vez identificado se busca su patrón en la base de datos y posteriormente se le pide al usuario la característica biométrica para compararla con la que se identificó. Este método sólo compara dos muestras y el resultado es positivo o negativo.

3.1.3. Vulnerabilidad de los sistemas biométricos

El uso creciente en distintas áreas y campos de los sistemas biométricos ha generado preocupación acerca de la seguridad y vulnerabilidad de estos datos. Un aspecto importante es que si las características biométricas fueran sustraídas ilegalmente no habría forma de reemplazarlas por otras, como se hace con las contraseñas. Las personas encargadas de la usurpación de información se han dedicado con los años a desarrollar técnicas o métodos que les permitan acceder a los sistemas biométricos. Estos ataques pueden darse de diferentes maneras, puede ser atacando directamente la base de datos o incluso interfiriendo directamente en los protocolos de comunicación cuando se realiza el envió de información.
De manera general los ataques a sistemas biométricos se pueden clasificar en tres grupos:

- Administrativos: Los ataques se realizan por personal del mismo grupo ó gente

de adentro *(insiders)*.

- Infraestructura no segura: El atacante fija como objetivo componentes vulnerables del sistema.

- *Biometric overtness*: Se clonan y se crean modelos biométricos artificiales para acceder al sistema.

Normalmente existe relación entre las tres anteriores, de modo que un mismo ataque suele pertenecer a dos clasificaciones o a las tres.
En el esquema presentado en la Figura 3.2 se muestran detenidamente los ocho puntos más vulnerables en un sistema biométrico.

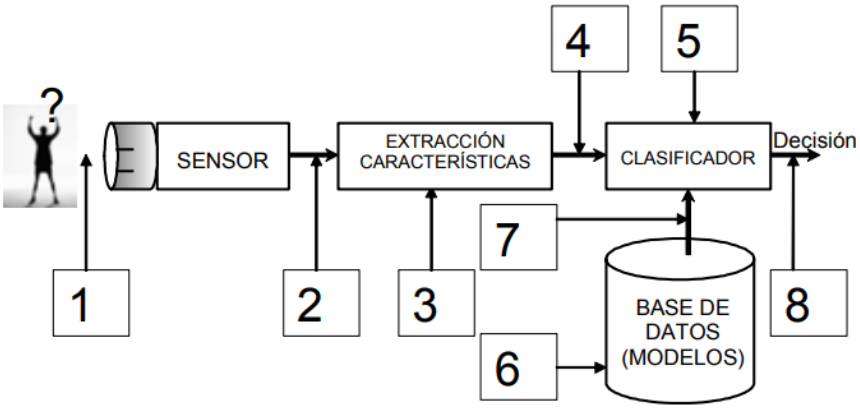

Figura 3.2: Elementos vulnerables en un sistema de reconocimiento biométrico [14]

1. Sensor: Los ataques a nivel sensor consisten en presentar una característica biométrica falsa esperando aceptación del sistema.
2. Transmisión entre el sensor y el extractor de características: Los ataques en este nivel consisten en inyectar datos biométricos almacenados previamente. Esta posibilidad es especialmente importante en aplicaciones remotas, en las que un ordenador cliente proporciona datos biomédicos a un host remoto que lleva a cabo el reconocimiento.
3. Extractor de características: En este tipo de ataques, se obliga al extractor de características a proporcionar valores escogidos por el impostor, en vez de las medidas reales extraídas a partir de la señal original obtenida por el sensor. Este ataque suele hackear el programa principal de extracción de características para reemplazarlo por

otro programa.
4. Transmisión de las características extraídas: Similar al ataque del punto dos, este ataque remoto busca reemplazar las características extraídas por otro conjunto falso. Este tipo de ataques suele ocurrir en sistemas que transmiten únicamente las medidas de interés, lo cual elimina el envió de información no relevante y dificulta que el intruso pueda regenerar la señal biométrica original.
5. Clasificador: Similar al ataque tres, los ataques al clasificador buscan modificarlo para que este entregue puntuaciones alteradas para forzar el sistema a la aceptación o al rechazo.
6. Base de datos: Uno de los ataques más frecuentes y el más importante ya que si se compromete la base de datos el sistema es vulnerable de forma permanente. Estos ataques suelen entrar al sistema para robar toda la información almacenada para poder clonarla e ingresar al sistema posteriormente. Por otro lado, es posible que un intruso se registre en el sistema exitosamente y después acceder al sistema.
7. Transmisión de la base de datos al clasificador: Otro tipo de ataques al host, en este se busca suplantar los datos reales para afectar los modelos (o plantillas) almacenados en la base de datos hacia el clasificador.
8. Decisión: Este ataque sólo intercepta el bloque de decisión después de que se realizó todo el análisis a la señal biométrica.

3.1.4. Biometría Cancelable

Este tipo de biometría busca distorsionar un patrón de características de un usuario en especifico con el objetivo de proteger la información sensible. Esta distorsión debe poder repetirse con el fin de distorsionar el patrón original la cantidad de veces que se requiera. Su fundamento es no almacenar en la base de datos el patrón de características ó plantilla biométrica original, en lugar de esto, se almacena la plantilla distorsionada por un proceso de protección de datos. Así, si una plantilla es comprometida sólo se deberá cambiar las características de distorsión y crear una nueva plantilla modificada.
Los cuatro objetivos al diseñar un sistema biométrico son los siguientes:

- Diversidad: No se pueden usar las mismas características cancelables en varias

aplicaciones, por lo tanto, se requiere una gran cantidad de plantillas protegidas de la misma característica biométrica.

- Reusabilidad-Revocabilidad: Cancelar la plantilla actual y emitir otra en caso de compromiso.
- No invertibilidad: Para evitar la recuperación de datos biométricos originales.
- Rendimiento: El rendimiento del sistema utilizando la plantilla protegida no debe deteriorar el rendimiento del sistema que usa la plantilla sin proteger.

Métodos de biometría cancelable

Actualmente los métodos existentes se dividen en dos categorías:
Biometric Salting: Aquí los datos específicos del usuario (contraseña o cualquier número que se le ha proporcionado) se combinan con los datos biométricos para obtener una versión distorsionada de la plantilla biométrica. Este método depende totalmente de los datos auxiliares externos del usuario y es revocable simplemente cambiando estas contraseñas. La desventaja es que si estos datos son robados el sistema se vuelve vulnerable.
Transformación no invertible: En este método los datos biométricos se modifican usando una función unidireccional no invertible. Los parámetros en la transformación de datos se modifican para desarrollar plantillas actualizadas. La ventaja de este tipo de plantillas es que el impostor no puede reconstruir la plantilla biométrica original incluso si las plantillas protegidas fueron comprometidas.

3.1.5. Rendimiento de un sistema biometríco

El rendimiento de un sistema en modo de verificación se define generalmente en términos de la tasa de falsa aceptación (false acceptance rate o FAR), la tasa de falso rechazo (False Rejection Rate o FRR) y la tasa de igual error (EER). A continuación se detalla en que consiste cada tasa:

- **FRR:** número de veces que el usuario es rechazado a pesar de ser genuino.

- **FAR:** número de veces que el usuario es aceptado a pesar de ser un impostor.
- **EER:** punto de cruce entre el FRR y FAR.

Comunmente se utilizan gráficas representativas del análisis del rendimiento. La curva ROC (Receveing Operating Characteristics) es una gráfica que ayuda a ver la proporción de usuarios impostores que son admitidos en un sistema en modo de verificación.
En una curva ROC la tasa de verdaderos positivos se representa en el eje y y la tasa de falsos positivos se representa en el eje x [15].
En la Fig. 3.3 se muestran tres curvas ROC que representan pruebas excelentes, buenas y sin valor trazadas en el mismo gráfico.
La precisión se mide por el área bajo la curva ROC. Un área de 1 representa una prueba perfecta mientras que un área de 0.5 representa una prueba sin valor [16]. Cuanto más se acerque la curva a la esquina superior izquierda, menor será la probabilidad de fallo y por lo tanto mejor será el resultado.

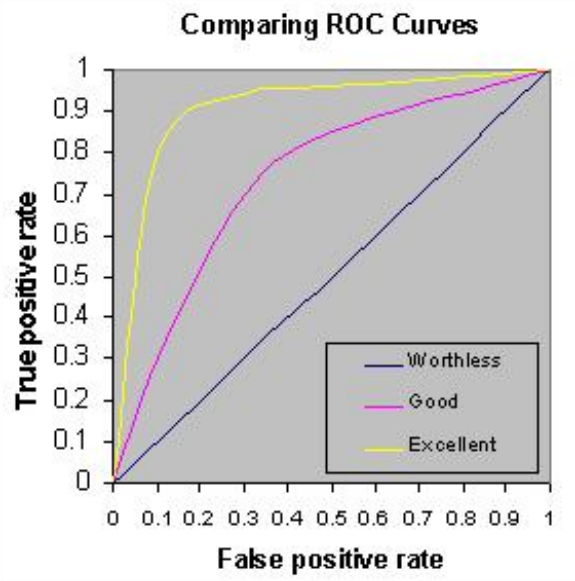

Figura 3.3: Gráfica ROC [16].

Una guía aproximada para clasificar la precisión de un sistema biométrico es la siguiente [16]:

- 0.90-1= Excelente.

Tabla 3.1: Estudio comparativo de diferentes modalidades biométricas [2].

Biometría	Unicidad	Universalidad	Permanencia	Rendimiento	Aceptabilidad
Huella Dactilar	H	M	H	H	M
Palma de la mano	H	M	H	H	M
Iris	H	H	H	H	L
Cara	H	H	M	H	H
Voz	L	M	L	L	M
Firma	L	L	L	L	H
ADN	H	H	H	H	L
Contraseñas escritas	L	L	L	L	M
ECG	**H**	**H**	**H**	**H**	**M**

H-High, M-Medium, L-Low.

- 0.80-0.90= Bueno.
- 0.70-0.80= Justo.
- 0.60-0.70= Pobre.
- 0.50-0.60= Fallo.

3.2. Señales biomédicas

3.2.1. Adquisición y adaptación de la señal biométrica

En la actualidad existen muchos rasgos biométricos para realizar biometría. Cada rasgo es único y algunos ofrecen más beneficios respecto a los otros. De [2] se obtuvo la tabla 3.1 comparativa donde se indica el rendimiento de cada propiedad para cada rasgo biometrico.

Señal electrocardiográfica (ECG)

El electrocardiograma (ECG) representa la actividad eléctrica de las células del corazón. Este impulso produce la contracción rítmica del corazón. A su vez esta actividad electromecánica se produce según un orden estricto y siempre igual latido tras latido [17]. Este tipo de señales muestra buenas propiedades para utilizarse en

un sistema biométrico.

Un individuo sano en condiciones normales debe presentar un diagrama de ECG como el que se muestra en la Fig. 3.4, donde se observa una onda P, un complejo QRS y una onda T.

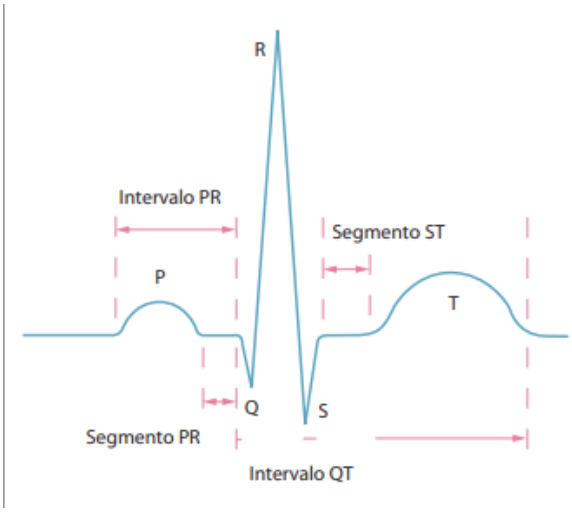

Figura 3.4: Trazado de ECG [18].

Las formas de onda más representativas del ECG son:

- Onda P: Es un evento eléctrico que indica el inicio de un ciclo o periodo cardíaco, representa la despolarización de las aurículas (primero la aurícula derecha, luego la izquierda). Tiene una duración entre 0.09 y 0.11 segundos y su amplitud aproximada es de 0.25 mV.
- Intervalo P-R: Espacio comprendido entre el fin de la onda P y el inicio del complejo QRS. Tiene una duración entre 0.11 y 0.20 segundos.
- Complejo QRS: Representa la despolarización eléctrica de los ventrículos, conocida como depresión u onda Q. Dura entre 0.07 y 0.11 segundos
- Intervalo Q-T: Posee una duración entre 0.35 a 0.44 segundos.
- Segmento ST: Comprendido entre el fin del complejo QRS y el inicio de la onda T. Representa el tiempo durante el que los ventrículos permanecen en estado

activado y se puede iniciar la repolarización ventricular. Tiene un intervalo de tiempo entre 0.05 a 0.15 segundos.

- Onda T: Representa la repolarización de los ventrículos. De igual polaridad al complejo QRS. Alcanza entre 0.1 a 0.5 mV [19].

La forma de onda del ECG varia dependiendo los puntos de referencia donde se coloquen los electrodos. Las posiciones que estos puedan tomar permiten apreciar mejor ciertas propiedades del corazón.

3.2.2. Preprocesamiento de la señal

Todas las señales que provienen de medios físicos o del exterior contienen determinada cantidad de ruido por naturaleza, por eso es importante preprocesar la señal para la eliminación de contenido indeseable.
Existen varias técnicas de filtrado dependiendo de los requerimientos. A continuación se explica la técnica de filtrado digital a través de filtros de Respuesta Finita al Impulso (FIR) y de filtrado a través de la Transformada Wavelet Discreta.

Filtros digitales FIR

Un filtro digital es un sistema lineal e invariante en el tiempo (LTI, por sus siglas en inglés) que modifica los atributos de una señal en el dominio del tiempo o la frecuencia. Un LTI interactúa con la entrada mediante un proceso llamado convolución, denotado por:

$$y = x * f \tag{3.2.1}$$

donde f es la respuesta al impulso del filtro, x es la señal de entrada y y es la salida convolucionada. El proceso de convolución lineal formalmente se define como:

$$y[n] = x[n] * f[n] = \sum_k x[k]f[n-k] = \sum_k f[k]x[n-k] \tag{3.2.2}$$

En 3.2.2 n hace referencia al índice de la muestra y k es el retraso que tiene la

señal original.
Los sistemas LTI se clasifican en FIR (*Finite Impulse Response*) que se caracterizan por ser sistemas no recursivos e *IIR (Infinite Impulse Response)* que se caracterizan por tener retroalimentación en la señal de salida. El filtro FIR permite una implementación más fácil debido a que no tiene la retroalimentación en la entrada, a continuación se explicará de manera más detallada como funciona un filtro FIR.

Filtros FIR. Un filtro FIR de longitud L o de orden N=L-1 se describe mediante la siguiente ecuación:

$$y[n] = x[n] * f[n] = f_0x(n) + f_1x(n-1) + f_2x(n-2) + ... + f_Nx(L-1) \quad (3.2.3)$$

Expresando la ecuación 3.2.3 en el dominio de la frecuencia z tenemos:

$$Y(z) = F(z)X(z) \quad (3.2.4)$$

Donde *F(z)* es la siguiente función de transferencia del filtro en el dominio z :

$$F(z) = \sum_{k=0}^{L-1} f[k]z^{-k} \quad (3.2.5)$$

Entre las propiedades del filtro FIR están:

- Puede diseñarse para tener fase lineal.
- Siempre son estables porque únicamente tienen ceros en el plano complejo.
- Generalmente, la longitud del filtro indicara la cantidad de recursos (sumadores, multiplicadores) que se utilizaran en la implementación del filtro.
- Pueden ser recursivos y no recursivos.
- La salida siempre es una combinación lineal de las valores de entrada pasados y presentes.
- El filtro FIR también se conoce como ”filtro transversal”por su estructura.

En la Figura 3.5 se muestra la representación gráfica de los bloques para la implementación de un filtro FIR, que consta de retardos, sumadores y multiplicadores. Como

ya se ha mencionado anteriormente, la cantidad de recursos y el número de etapas estarán definidos por el orden del filtro.

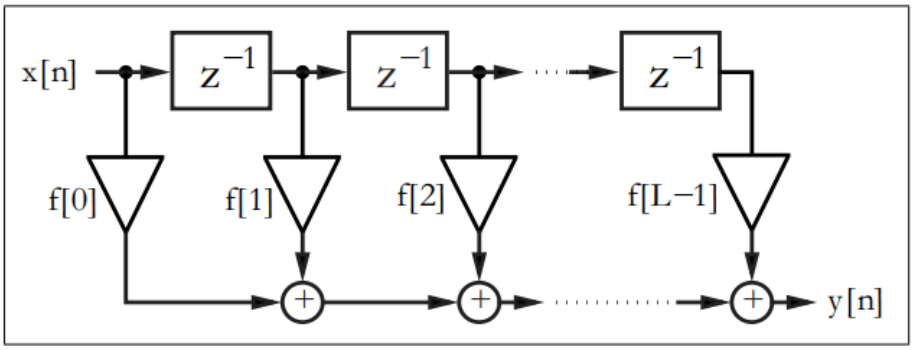

Figura 3.5: Filtro FIR con estructura directa. [20]

La estructura anterior es la más utilizada, sin embargo la estructura cambia si los requerimientos del filtro cambian, por ejemplo al hacerlos de fase lineal o con una estructura transpuesta. Para el caso que se requiera implementar un filtro de fase lineal se deben verificar ciertas condiciones de simetría.[21]

- Un sistema no causal con respuesta impusional conjugada simétrica *(h(n)=h*(-n))* tiene una función de transferencia real.
- Un sistema no causal con respuesta impusional conjugada antisimétrica *(h(n)=-h*(-n))* tiene una función de transferencia imaginaria pura.

Se puede lograr un retardo de grupo constante si la respuesta de frecuencia $F(\omega)$ es puramente real o imaginaria. Esto conlleva a que la respuesta al impulso del filtro tiene simetría par o impar. Esto es:

$$f[n] = f[-n] \;\; o \;\; f[n] = -f[-n] \tag{3.2.6}$$

Una de las ventajas de los filtros de fase lineal en la implementación es que reduce el número de multiplicadores mientras que el número de sumadores se mantiene constante. En la Fig. 3.6 se muestra la estructura de un filtro FIR de fase lineal con

simetría par donde sólo se utiliza un multiplicador por ciclo del filtro, esto es la mitad de la cantidad de multiplicadores que se utilizarian en una arquitectura directa.

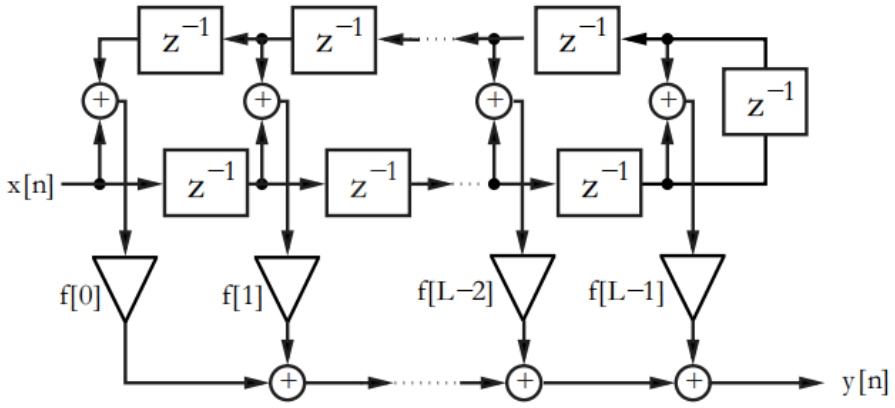

Figura 3.6: Filtro FIR de fase lineal. [20]

Filtrado por Transformada Wavelet Discreta

La trasnformada Wavelet Discreta (DWT) es una herramienta matemática que permite obtener información temporal y frecuencial de una señal a través de una descomposición en bandas. A diferencia de Fourier que expresa la señal en terminos de senos y cosenos a diferentes frecuencias, al análisis Wavelet expresa la señal en función de versiones desplazadas y escaladas de una función Wavelet madre.
En otras palabras la DWT transforma un vector de datos de longitud n en otro vector de coeficientes wavelets de longitud n, usando un conjunto de n funciones bases ortogonales llamadas wavelets[22].

La DTW utiliza filtros pasa-bajas y pasa-altas, $h(n)$ y $g(n)$, denominados filtros de análisis para extraer información de la señal. Se realiza una dilatación para cada escala a través del diezmado. Se obtienen unos coeficientes c_k y d_k a través de la convolución de la señal digital con cada filtro y posteriormente diezmando la salida. Los coeficientes que se producen por el filtrado pasa-bajos reciben el nombre de coeficientes de aproximación y los coeficientes producidos por el filtrado pasa-altos son coeficientes de detalle.

Los coeficientes de aproximación contienen información sobre las bajas frecuencias y los coeficientes de detalle contiene información sobre las altas frecuencias. Ambos coeficientes se producen en multiples escalas utilizando los coeficientes de aproximación en cada escala durante el proceso. Todo el proceso se realiza bajo un esquema de banco de filtros con estructura en árbol como se ve en la Fig. 3.7.

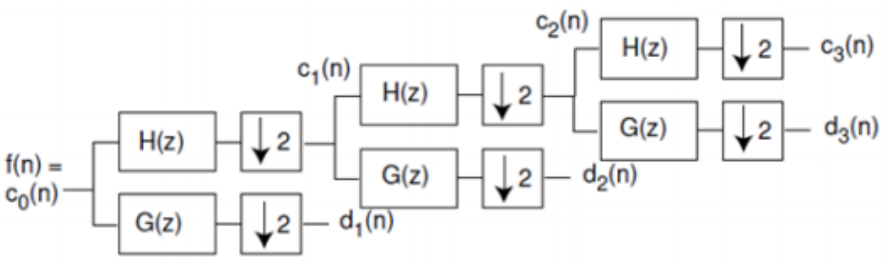

Figura 3.7: Banco de filtros para descomponer la señal. [23]

Después de procesar la señal en el dominio de las Wavelets y de obtener los coeficientes se puede volver la señal a su dominio original a través de filtros de síntesis y expansores. Los coeficientes se aplican a un banco de filtros de síntesis para regresar la señal original como se muestra en la Fig 3.8.

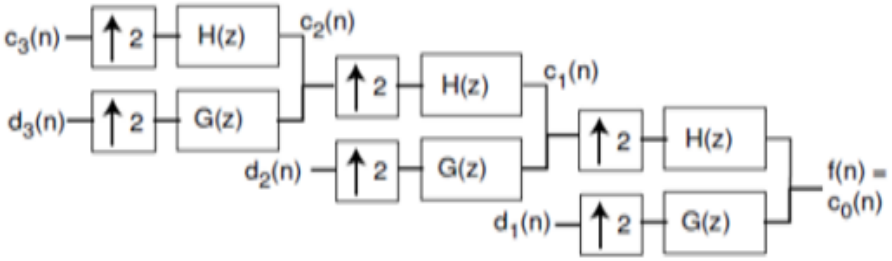

Figura 3.8: Bando de filtros para reestructurar la señal. [23]

Una manera de representar una señal procesada por la DWT es como una función *f(t)* que indica la sumatoria de las funciones Wavelet madre ψ y las funciones de escala ϕ:

$$f(t) = \sum_k \sum_j c_{j,k}\phi(t) + \sum_k \sum_j d_{j,k}\psi(t) \tag{3.2.7}$$

Donde $c_{j,k}$ y $d_{j,k}$ simbolizan los coeficientes de aproximación y de detalle respectivamente, y k es el número de niveles de descomposición.

3.2.3. Técnicas de extracción de características

El bloque de extracción de características recibe la señal pre-procesada y la convierte en un patrón único para cada individuo tomando sólo la información relevante de la señal. Dependiendo el tipo de señal se suelen utilizar distintas características. La selección del tipo de característica varía según la que contenga la mayor información util de la señal.
Los métodos de extracción más utilizados estan basados en las características morfológicas, modelos matemáticos para describir la señal o a través de diversas transformadas para extraer información.
En señales ECG es común hacer uso de las características en el dominio temporal, estos son los intervalos de tiempo del complejo QRS, el tiempo entre picos P, T y la onda U. En la literatura existen un gran número de algoritmos que logran detectar exitosamente este complejo y que presentan diversas características de funcionamiento. Uno de ellos es el algoritmo de Pan-Tomkins.
Otras características que se presentan en este mismo dominio son las estadísticas presentadas en [24], donde se hace uso de la varianza, parámetros Hjorth, asimetría y curtosis.
A continuación se presenta en que consiste cada técnica y como se obtienen.

Algoritmo de Pan-Tompkins

En 1985 se estableció el algoritmo de Pan-Tompkins donde se realiza un análisis de la pendiente, la amplitud y el ancho de los complejos QRS para su detección mediante una serie de etapas.
En la actualidad es el método más robusto y confiable para la detección de complejos QRS, a pesar de la cantidad de ruido que contenga la señal. Para su detección utiliza una serie de filtros y operadores. En la Fig. 3.9 se muestra el orden de los filtros empleados en este algoritmo.

Figura 3.9: Banco de filtros para reestructurar la señal.

A continuación se describen brevemente cada etapa del proceso del algoritmo Pan-Tompkins [25]:

- Filtro pasa-bajas: La señal ECG debe pasar por un filtro pasa-bajas para eliminar las componentes de alta frecuencia, por ejemplo el ruido de la tensión eléctrica.

- Filtro pasa-altas: Este filtro eliminará las componentes de baja frecuencia, por ejemplo el ruido generado por los artefactos al momento de adquirir la señal y también elimina las ondas P y T. En la Fig. 3.10 se muestra una señal ECG después de la etapa de filtrado.

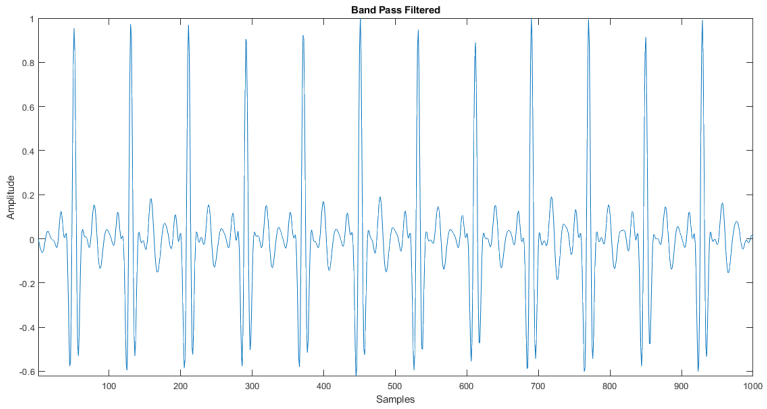

Figura 3.10: Señal ECG después de la etapa de filtrado.

- Derivación: La señal se deriva para detectar las pendientes pronunciadas características de los laterales de la forma de onda del complejo QRS (Ver Fig. 3.11)

- Squaring: Durante este bloque se eleva la señal ECG al cuadrado punto a punto, generando una señal positiva que intensifica las altas frecuencias y atenua las bajas, separando los complejos QRS de las ondas T. El resultado se muestra en la Fig. 3.12.

- Integración: En este punto la señal se integra mediante una ventana móvil. Es recomendable que el ancho de la ventana de integración sea aproximadamente igual al ancho del complejo QRS, ya que si no cumple con esta condición la

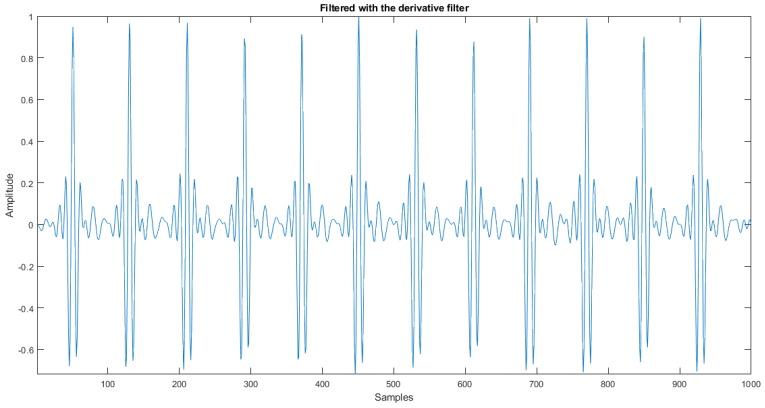

Figura 3.11: Señal ECG después de la etapa de derivación.

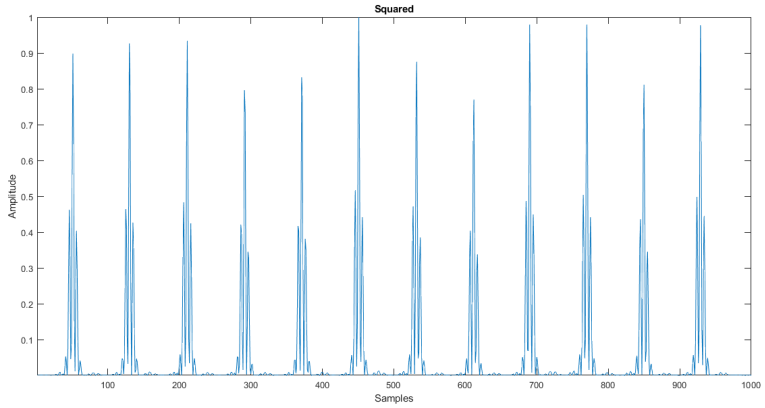

Figura 3.12: Señal ECG después de la elevación al cuadrado.

detección no es óptima. El valor del ancho de la ventana se debe ajustar de acuerdo a la señal ECG. (Ver Fig. 3.13)

- Umbralización: Se emplean umbrales adaptativos para detectar picos de energía, que son cambios de positivo a negativo. El algoritmo a través de un filtro adaptativo determinará si este pico de energía corresponde a un complejo QRS o es una detección erronea. Estos umbrales calculan los valores promedios relacionados con la amplitud de los picos pertenecientes al complejo QRS y del nivel de ruido. En la Fig. 3.14 se muestra el tren de pulsos detectados con el algoritmo y que pertenecen al complejo QRS.

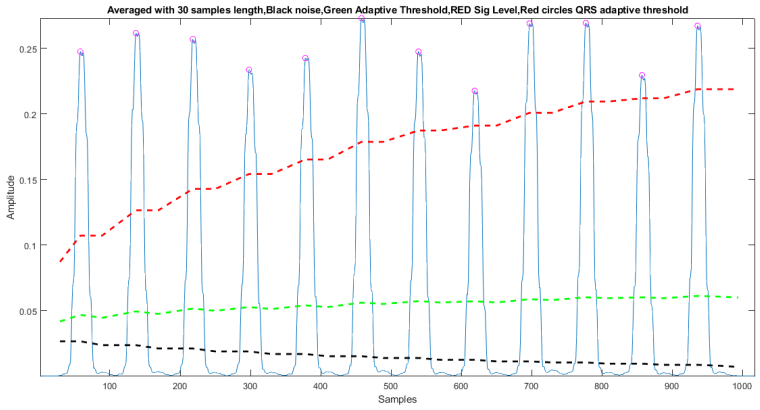

Figura 3.13: Detección del complejo QRS después de la integración.

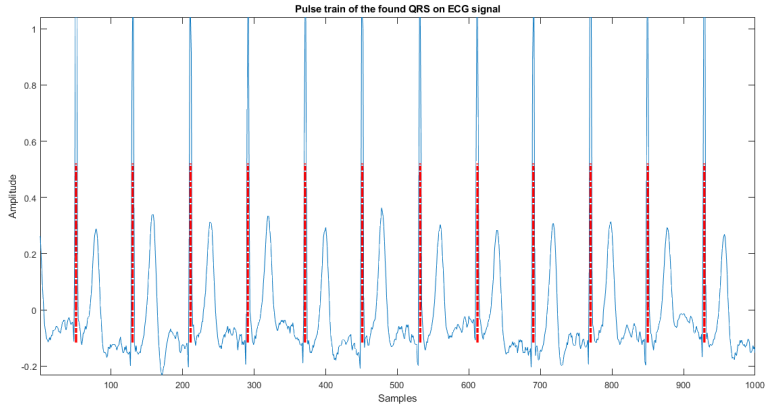

Figura 3.14: Complejos QRS detectados.

Técnicas Estadísticas

Parámetros de Hjorth: Son indicadores de las propiedades estadísticas de una señal, propuestos por Bo Hjorth en [26] inicialmente para señales EEG. Este método involucra la actividad, movilidad y complejidad:

- Actividad: Representa la potencia de la señal x y se calcula con la medida de la varianza de amplitud.

$$Actividad(x) = var(x) \tag{3.2.8}$$

- Movilidad: Representa la frecuencia media o la proporción de desviación estándar del espectro de poder y para obtenerla se calcula la raíz cuadrada de la varianza de la primera derivada de la señal x entre la varianza de la señal x.

$$Movilidad(x) = \sqrt{\frac{var(x')}{var(x)}} \tag{3.2.9}$$

- Complejidad: Representa el cambio de la señal en el dominio de la frecuencia y se calcula con la raíz cuadrada de la primera derivada de la movilidad entre la movilidad.

$$Complejidad(x) = \sqrt{\frac{movilidad(x')}{movilidad(x)}} \tag{3.2.10}$$

Asimetría: Informa sobre el grado de asimetría de la distribución de datos de la señal analizada. Es una medida mediante la cual se puede identificar el modo en que los datos de la señal tienden a agruparse [27]. Se calcula mediante:

$$A = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{N \cdot S_x^3} \tag{3.2.11}$$

Siendo x_i uno de los datos, $\bar{x}$ la media, N el número de muestras y S_x la desviación típica.

Curtosis: Informa sobre el grado de homogeneidad de datos de la señal. Mediante la curtosis se puede saber la proporción de la varianza explicada por la combinación de valores atípicos respecto a la media en contraposición con datos en posiciones más centrales [27]. Se calcula mediante:

$$C = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{N \cdot S_x^4} \tag{3.2.12}$$

Siendo x_i uno de los datos, $\bar{x}$ la media, N el número de muestras y S_x la desviación típica.

3.2.4. Clasificador

El objetivo del clasificador es comparar los datos que se estan ingresando al sistema con los almacenados en la base de datos para obtener la relación que existe entre ellos. Para el modo de identificación se compara la señal contra todas las almacenadas en la base de datos mientras que para el modo de verificación se compara la señal sólo con la del individuo que se indica ser.
Alguna de las técnicas para obtener esta relación es mediante la distancia entre dos conjuntos de observaciones. En la literatura se han establecido métodos como la distancia Euclidiana, la distancia Minkowski, la distancia coseno, entre otros. Pero la que ha reportado mejores resultados es el Alineamiento Temporal Dinámico.

Alineamiento Temporal Dinámico (DTW).
El Alineamiento Temporal Dinámico (DTW por sus siglas en ingles) es una técnica que permite realizar un alineamiento entre dos vectores o conjuntos de datos donde puede existir una variabilidad temporal con el fin de obtener una distancia.
Para obtener la función de alineamiento se parte de dos vectores de características A de longitud m y B de longitud n [28].

$$\text{A= } a_1,a_2,a_3,a_4,...,a_m$$
$$\text{B= } b_1,b_2,b_3,b_4,...,b_n$$

Y de una función de alineamiento 'C':

$$C = \{c(1), c(2), ..., c(k), ..., c(K)\} \tag{3.2.13}$$

Donde c(k) es el par de punteros de los elementos a comparar:

$$c(k) = [i(k), j(k)] \tag{3.2.14}$$

Para cada c(k) se tiene una función de costo que refleja la discrepancia entre los

Tabla 3.2: Distancias locales

	n	d(1,n)	d(2,n)	d(3,n)	...	d(m,n)
	...	...	...	...	...	
b	3	d(1,3)	d(2,3)	d(3,3)	...	d(m,3)
	2	d(1,2)	d(2,2)	d(3,2)	...	d(m,2)
	1	d(1,1)	d(2,1)	d(3,1)	...	d(m,1)
		1	2	3	...	m

a

elementos. Esta descrita por:

$$d\{c(k)\} = \delta(a_{i(k)}, b_{j(k)}) \tag{3.2.15}$$

Una función de costo utilizada comunmente es la siguiente:

$$d\{c(k)\} = |a_{i(k)} - b_{j(k)}| \tag{3.2.16}$$

A partir de esto se obtiene una tabla de distancias como la que se muestra en la Tabla 3.2

En esta tabla de distancias se aplica una función de recursión descrita por 3.2.17, donde a y b son los indices de la matriz.

$$D(a,b) = d(a,b) + min[D(a-1,b), D(a-1,b-1), D(a,b-1)] \tag{3.2.17}$$

Y finalmente se obtiene el valor de DTW(a,b) con 3.2.18

$$DTW(a,b) = \frac{D(M,N)}{L} \tag{3.2.18}$$

Donde:

- D(M,N) es la función de recursión evaluada en (M,N).

- L es la longitud del camino óptimo desde D(0,0) a D(M,N).

3.2.5. Decisión

El módulo de toma de decisiones se usa para determinar si el grado de similitud devuelto es suficiente para determinar la identidad de un individuo[29]. A partir de esto se puede conocer si el individuo que intenta acceder al sistema es un usuario autentico o un impostor.
Para el modo de identificación se tiene un conjunto de puntuaciones obtenidas del clasificador y a partir de estas se puede determinar a cuál de todos los individuos pertenece la identidad. En el modo de verificación se debe establecer un umbral para validar la identidad del usuario que se dice ser. Si el puntaje esta por debajo del umbral entonces se comprueba que la identidad es correcta, de lo contrario es un impostor.

3.3. Funciones Físicamente Inclonables (PUF)

Una PUF es una función física aleatoria que extrae una única ”firma”de un circuito integrado, basándose en la aleatoriedad durante el proceso de fabricación. Esta firma puede ser usada como una llave que depende del dispositivo o como un código de identificación del dispositivo.
El comportamiento del PUF se basa en crear una función que mapea un conjunto de entradas a un conjunto de respuestas que se rigen por las propiedades físicas que son difíciles de predecir, controlar o reproducir. Por lo tanto, el comportamiento de la función puede ser sólo evaluado por el sistema físico y este es único para cada dispositivo en el que se implemente. La respuesta que arroje el dispositivo puede cambiar debido a las variaciones del entorno o el envejecimiento que pueda afectar al dispositivo, sin embargo, debe existir cierto grado de similitud entre las respuestas de modo que pueda identificarse que esas salidas pertenecen a la misma entrada.
Entre las ventajas que tienen las PUFs esta que se pueden obtener cualidades que no pueden ser obtenidas por criptografía pero requieren una base física para establecerse, la más notable es la imposibilidad de clonación física, ya que se ha

demostrado a través de razonamiento físico que producir un clon físico de una PUF es extremadamente difícil o imposible [30].
Otra de sus principales cualidades es que no se requiere el uso de memorias para almacenar claves lo que lo hace menos vulnerable a ataques.

3.3.1. Proceso de Construcción

La razón por la cual se asegura que una PUF es casi imposible de clonar se basa en las pequeñas variaciones de fabricación. Incluso con un control extremo sobre un proceso de fabricación, no se pueden crear dos dispositivos, idénticos por diseño, exactamente iguales debido a la influencia de efectos aleatorios e incontrolables. Las diferencias en las características de estos son impredecibles y sólo pueden medirse con técnicas de escala microscópica.
Lograr una medición de este tipo con una precisión lo suficientemente alta como para distinguir estas características es el objetivo principal en el estudio de construcciones de PUF

3.3.2. Propiedades

La eficiencia y la funcionalidad de una PUF se basan en cumplir con ciertas propiedades y en sus distancias de Hamming intraclase e interclase.
Para un mismo dispositivo la distancia intraclase debe tener pequeñas variaciones y ser menor a un umbral de identificación óptimo, mientras que para salidas de diferentes dispositivos (distancias interclase) estas salidas deben ser lo suficientemente distintas o mayores al umbral.
A continuación se mencionan una serie de propiedades que deben estar presentes también al evaluar una PUF.

- Evaluable: Se considera que una PUF es evaluable, si para una PUF y una entrada aleatoria es fácil evaluar la salida correspondiente.
- Único: La salida de la entrada evaluada en el PUF contiene alguna información

acerca de la identidad física en la que esta embebida la PUF ó si la distancia interclase es mayor que el umbral.

- Reproducible: Si la salida a determinada entrada dada es reproducible hasta con un pequeño error.
- Inclonable: Dada determinada PUF es muy difícil fabricar un clon idéntico que genere las mismas respuestas a las entradas dadas hasta con determinado error.
- Impredecible: Para un conjunto de entradas y salidas dadas debe ser imposible predecir cual sera la salida a una nueva entrada dada.
- De una sola mano: Para una salida y una PUF dadas debe ser difícil encontrar la entrada con la cual generar esa salida.

Desafíos y Respuestas (CRP)

Típicamente se le ha denominado desafío a la entrada de una PUF y respuesta a la salida. Un desafío genera una respuesta y a este par se le denomino par desafío-respuesta o CRP (*Challenge-Response Pair*). Generalmente en el proceso de registro se recolecta un grupo de CRP's por cada PUF que son almacenados en su base de datos mientras que en el proceso de verificación se debe asegurar que algún CRP's almacenado coincida con el dispositivo que se solicita.

3.3.3. Aplicaciones

A continuación se presentan los diferentes escenarios donde se pueden emplear las PUF.

Identificación de sistemas

Las respuestas de las PUFs se pueden usar directamente para identificación muy similar al funcionamiento de un sistema de identificación biométrico.
Este proceso se divide en varias etapas, durante la etapa de inscripción un numero de

CRP's de cada PUF se almacenan en una base de datos junto con el sistema físico en el que esta embebido cada PUF. Durante la verificación se obtiene la salida de una PUF y se compara con las salidas almacenadas en una base de datos. Debido a que no se tiene control sobre la funcionalidad de la PUF no se asegura que la respuesta generada va a coincidir completamente con alguna respuesta almacenada, para medir la similitud del dispositivo en prueba se usa la distancia de Hamming y se deben cumplir las siguientes condiciones:

- La distancia de Hamming entre la salida de la PUF y la salida de la base de datos almacenada debe ser menor que la distancia límite fijada, con lo cual se asegura que las distancias son lo suficientemente similares.
- Las distancias de Hamming entre la salida de la PUF y el resto de las salidas almacenadas en la base de datos deben ser mayores a la distancia límite asegurando que las salidas son distintas.

El umbral se define en base a la separación que hay entre el histograma de la intra-distancia y la inter-distancia. En la Figura 3.15 se muestran dos curvas, la roja representa la distancia de Hamming de las salidas de una PUF generadas por el mismo dispositivo, mientras que la azul representa la distancia de Hamming de salidas de dispositivos distintos. Si las curvas no se sobreponen se puede lograr una identificación sin problemas definiendo el umbral en algún punto entre ambas curvas, pero en caso contrario, cuando las curvas se sobreponen se define el umbral al encontrar un equilibrio entre la tasa de aceptación falsa y la de rechazo falso (FRR).

Generación de llaves criptográficas

En muchos sistemas de seguridad se requiere el uso de llaves secretas. Generalmente estas llaves deben estar almacenadas en memorias haciendo vulnerable el sistema a ataques. Aquí es donde se presenta una de las mayores ventajas de las PUF's gracias a que se puede generar la llave con una PUF cuando se requiera, evitando almacenarla en una memoria.
Para este tipo de aplicaciones, contrario a la de identificación de sistemas, si se debe asegurar que la salida es estable antes de usarse como llave, por lo que se añade un

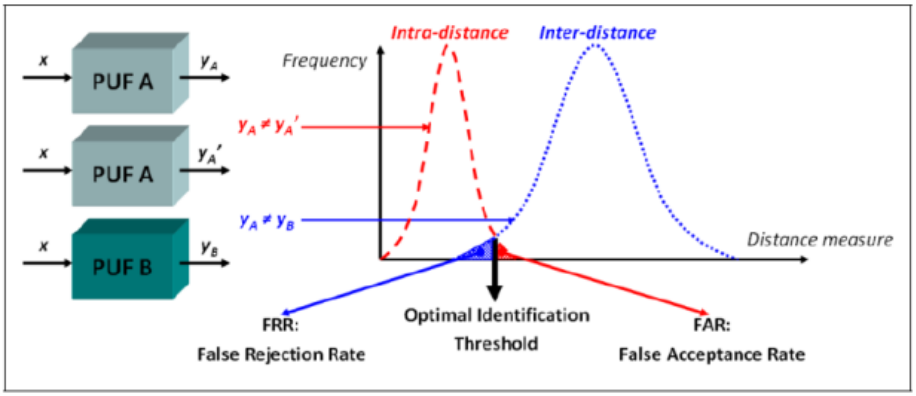

Figura 3.15: Proceso de identificación [31]

paso intermedio. Durante la etapa de generación, se obtiene la salida de la PUF y con ayuda de códigos de corrección de errores (ECC) se obtienen los *"helper data"* con los que se corrigen los bits erróneos en la salida

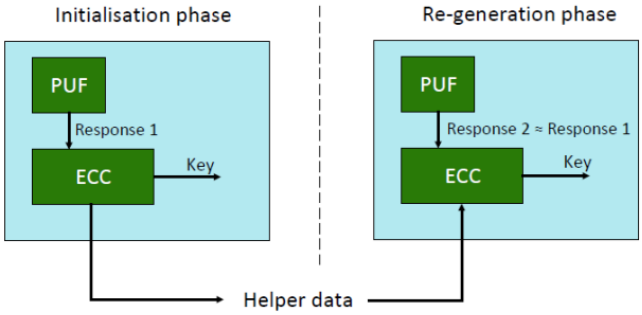

Figura 3.16: Proceso para la generación de llaves criptográficas [32].

Autenticación bajo costo

Debido a que la salida de cada PUF es única e impredecible para cada circuito integrado, siempre que la longitud de la respuesta sea lo suficientemente larga, se puede identificar a un circuito integrado con la respuesta que genere. Para esto se utilizan los CRP y el esquema de autenticación que se muestra en la Figura 3.17 consta de los siguientes pasos:

- Por cada dispositivo, se toma un grupo suficiente de muestras de CRP's y se

almacenan en una base de datos junto con la etiqueta del dispositivo del cual se obtuvieron las muestras.

- Cuando se requiere hacer el proceso de autenticación, primero, se verifica que la etiqueta del dispositivo que se quiere validar se encuentre en la base de datos. Una vez encontrado, se toma una muestra aleatoria de CRP correspondiente a ese dispositivo y se envía a la entrada del mismo. El dispositivo debe generar una respuesta lo suficientemente similar a la respuesta almacenada y , de ser así, el dispositivo se logra autentificar exitosamente. Cuando termina este proceso el sistema borra la muestra utilizada para esta validación de la base de datos.

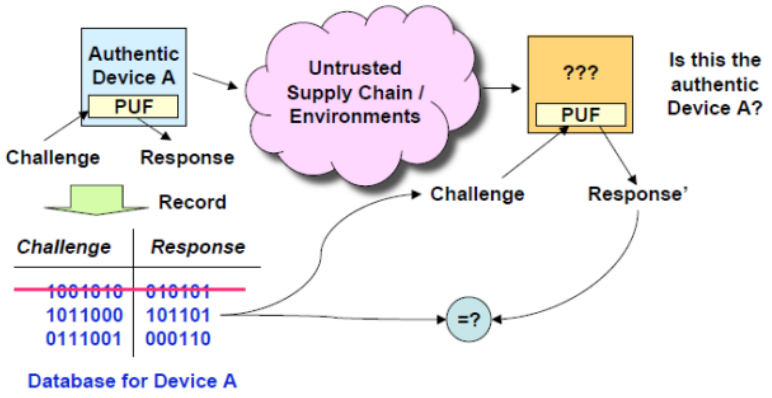

Figura 3.17: Proceso para autenticación [32].

Tipos de PUFs

Las funciones físicas inclonables tienen diferentes clasificaciones en base a diferentes parámetros, debido a la cantidad de PUF's que se han propuesto no se explicaran todos con detalle.

Una clasificación se basa en los materiales y la tecnología con la que se fabrican dividiendolas en PUF's no electrónicas, electrónicas y las PUF basadas en silicio, que es una categoría mayor de las PUF's electrónicas.

Otra clasificación importante es la basada en sus propiedades de construcción que los divide en PUF's intrínsecas y no intrínsecas, propuesto por Guarjardo en [33]. Para que una PUF pueda considerarse intrínseca debe cumplirse que:

- Su evaluación debe realizarse internamente mediante equipo de medición incrustado.
- Sus características aleatorias especificas de instancia son implícitamente introducidas durante su proceso de fabricación.

La última clasificación esta basada en las propiedades de seguridad del comportamiento reto-respuesta. Estos se dividen en *Fuertes* y *Débiles*. Una PUF es considerada fuerte si, incluso después de haber otorgado acceso a un usuario a la instancia de la PUF durante un periodo de tiempo, aún es posible que exista un reto para el cual el adversario no pueda conocer su respuesta. Para que esto se cumpla son necesarias dos cosas: 1)la PUF considerada tiene un gran conjunto de retos y 2) es casi imposible construir un modelo aproximado de la PUF basándose en la observación de los pares retos-respuestas, en otras palabras la PUF es impredecible. Una PUF que no cumpla con estas características es llamada PUF débil [30].

A continuación sólo se mencionarán los tipos de PUF's digitales existentes basados en construcciones intrínsecas que aprovechan la aleatoriedad de los procesos de fabricación de chips de silicio.

Estos a su vez se clasifican en base a su principio de operación:

- La primera clase se basa en la variación del retardo aleatorio del circuito digital, estos son llamados PUF's de silicio basados en retardos.
- La segunda clase usa las variaciones de parámetros aleatorios entre dispositivos de silicio coincidentes, en los elementos de memoria bi-estables. Estas son llamadas PUF's de silicio basadas en memoria.
- La ultima clase es para las PUF's de circuitos mixtos. Son analógicos por lo que arrojaran una salida analógica y se requiere de un convertidor analógico-digital para poder representar su salida digitalmente.

Algunos de las tipos de PUF's pertenecientes a estas clases se observan en el diagrama de la Figura 3.18 y sólo se describen los tres modelos más útiles basados en retardos.

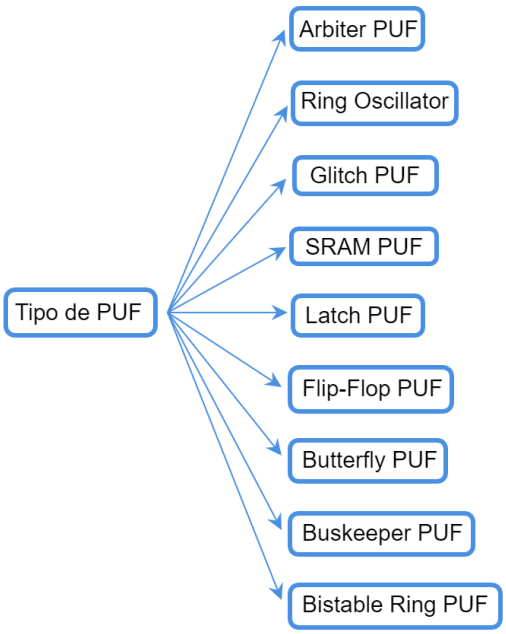

Figura 3.18: Modelos de PUF's intrínsecos basados en silicio

Arbiter PUF : Fue propuesto por Lee en [34] y [35], es una PUF de silicio basada en retardos. La base de su funcionamiento esta en ingresar una condición en una carrera digital que tiene dos caminos diseñados simétricamente en un chip y hacer que el circuito, que se ha denominado *arbitro* (arbiter) decida cual de los dos caminos ganó la carrera, es decir, este circuito decide cual de los caminos fue el más rápido y con esto se genera una cadena binaria como salida. Idealmente, si ambos caminos son simétricos deberían tener el mismo retardo, sin embargo, debido a las variaciones de fabricación, el retardo en cada chip involucrado será diferente, con lo que habrá un camino que tomara ventaja en cuestión de tiempo.

En la Figura 3.19 se muestra el esquema de funcionamiento de un Arbiter PUF, cada bloque de conmutación (switch) consta de dos multiplexores conectados los cuales tendrán un bit de control. Este bit de control será definido por la condición de carrera que se ingresa externamente (ó entrada de la PUF). Si el bit de control es 0 la entrada se conecta directamente con la salida correspondiente, en caso contrario las salidas se conectan de forma cruzada.

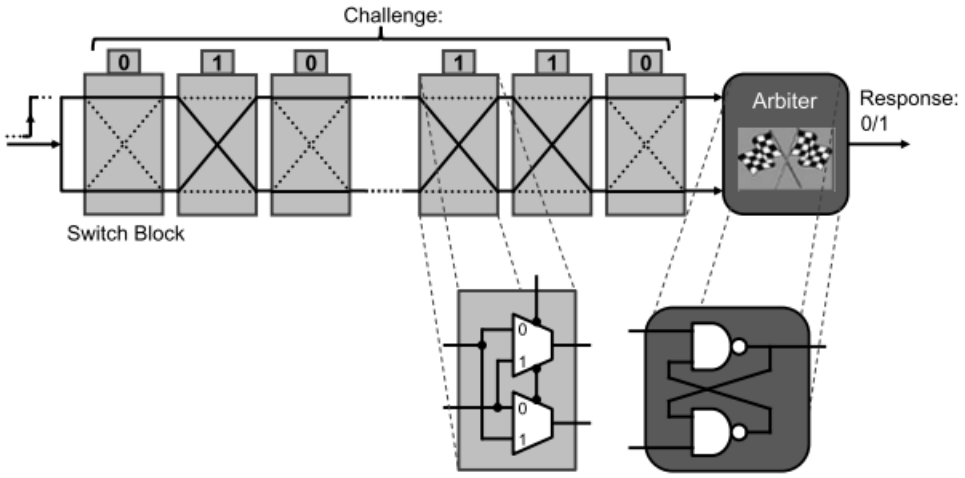

Figura 3.19: Modelo del Arbiter PUF[30].

Todos los bloques de conmutación están conectados en serie, logrando así que exista un retardo acumulativo. Al final del camino, este bloque de arbitro detecta cual de los dos caminos fue el más rápido, arrojando un 0 ó 1 como salida. Una desventaja de este modelo es que sólo se arroja un único bit como salida, si se desean obtener más se deben hacer modificaciones para poder generar una cadena de bits.

Ring Oscillator (RO PUF): Otra PUF basada en retardos es el oscilador de anillo, propuesta por Gassend en [36] y [37]. El Ring Oscillator hace uso de los retardos en las compuertas lógicas y las interconexiones que hay entre estas. Se construye con dos componentes básicos, el oscilador de anillo y un contador de frecuencia. El oscilador se compone de una compuerta nand y una serie de compuertas not conectadas en serie, retroalimentando la salida a la entrada para conseguir que oscile (ver Figura 3.20), mientras que el contador de frecuencia es un contador asíncrono que servirá para medir la frecuencia de ese oscilador. Una característica importante en esta PUF que se debe cumplir para su correcto funcionamiento es que el numero de compuertas not implementadas debe ser impar.

En la Figura 3.21 se observa un sistema convencional con este tipo de PUF's. Se compone de un conjunto de RO-PUF's, conectados a un multiplexor, cuyo selector se controla externamente y selecciona el RO-PUF que se desee comparar. La salida del multiplexor se conecta a dos contadores asíncronos que también se ven afectados por retardos aleatorios en su fabricación, por lo cual no tendrán la misma cuenta.

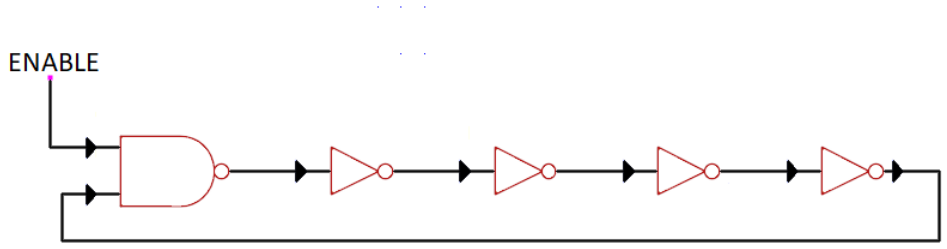

Figura 3.20: Modelo del Ring Oscillator con 5 etapas

Estos contadores se activan durante un periodo de tiempo, al final se comparan ambas cuentas y se seleccionara la mayor. El bit de salida será 0 ó 1 dependiendo del contador con la cuenta mayor. Este tipo de PUF ha tenido varias modificaciones en su esquema de funcionamiento para generar una mayor aleatoriedad.

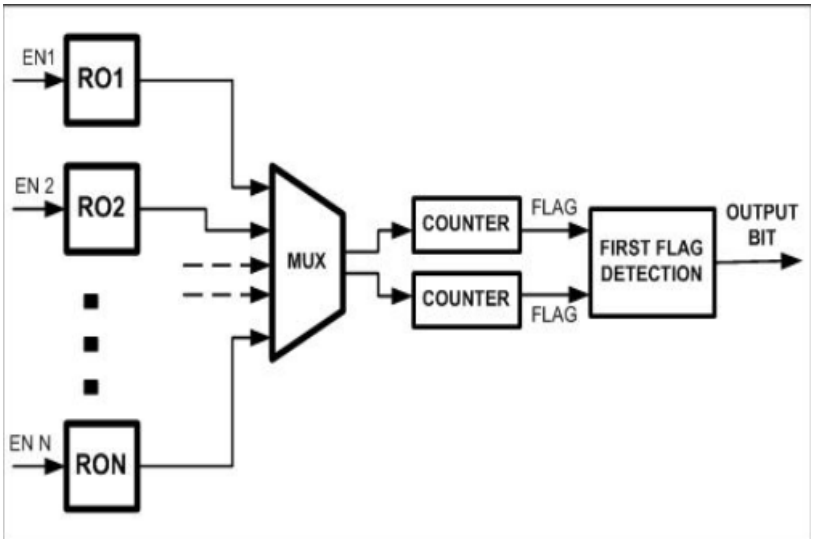

Figura 3.21: Sistema convencional con salida de un bit con un conjunto de RO-PUF's [38].

Glitch PUF: Este tipo de PUF fue propuesto por Anderson [39]para su implementación en FPGA y esta basado en la falla de comportamiento de los circuitos lógicos combinacionales. Un circuito puramente combinatorio no tiene estado interno lo que significa que su salida esta definida por sus señales de entrada. En algunos casos, al cambiar la señal de entrada, la salida no cambia inmediatamente debido a efectos de transición y debe transcurrir un tiempo antes de que esta salida se vuelva estable. Estos efectos se conocen como fallas y se producen por los retardos en las interconexiones que hay desde la señal de entrada hasta la señal de salida. Si esta

falla se puede medir con precisión se puede usar como salida de la PUF.

Con el tiempo se ha tratado de mejorar el comportamiento de las PUF y se han combinado con otras técnicas para diferentes aplicaciones. Entre estos nuevos modelos existen las *PUF's controladas, PUF's reconfigurables y las PUF's publicas.* A continuación sólo se hablara de las PUF's configurables, que fueron las que se implementaron en este trabajo.

CROPUF o Configurable Ring Oscillator. Se introdujeron por Maiti and Schaumont en [40]. El diseño esta basado en PUF's con configuración Ring-Oscillator pero con ciertas modificaciones.
La arquitectura de un CROPUF se muestra en la Fig. 3.22. Se compone de una compuerta AND, seis compuertas NOT y tres multiplexores. Dependiendo el tipo de FPGA que se utilice para saber que espacio utiliza de una CLB, ya que las CLB varian según el FPGA. Las entradas C_1, C_2 y C_3 permiten establecer una llave generando una respuesta diferente en la salida. La ventaja de esta configuración es poder seleccionar la salida más estable o poder seleccionar un bit de salida específico si es necesario.

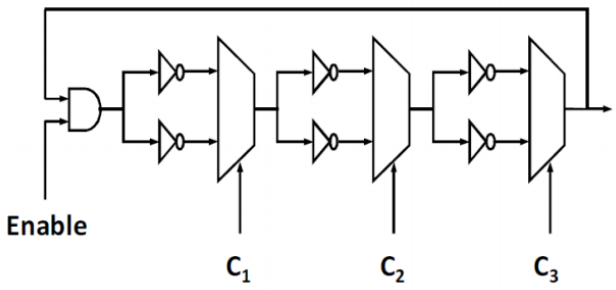

Figura 3.22: Diseño de un RO configurable [40].

Anderson PUF: Una de las estructuras propuestas recientemente fue el Anderson PUF [39] que consta de una serie de multiplexores conectados en serie con unas LUTs y un Flip-Flop que definirá la salida. Todos estos elementos van implementados en un slide tipo L y un slide tipo M. Se han propuesto trabajo como en [41] donde se han realizado modificaciones en el diseño original para poder implementarlo sólo en slides tipo L, ya que son las que más predominan dentro de los FPGA. En la Fig. 3.23.

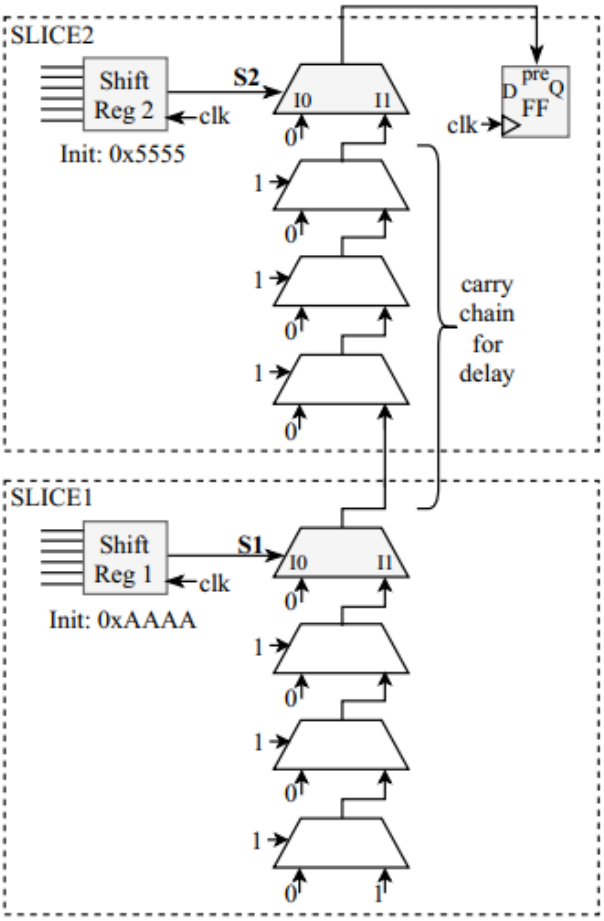

Figura 3.23: Diseño de un Anderson PUF [41].

3.4. FPGA

Los FPGA (*Field Programmable Gate Array - Arreglo de compuertas Programable en Campo*) son dispositivos semiconductores programables que se basan en una matriz de bloques lógicos configurables (CLB) conectados a través de interconexiones programables[42]. Estos dispositivos se componen de un grupo de módulos lógicos, estos módulos son independientes entre sí y se pueden interconectar de acuerdo al diseño requerido. La arquitectura de cada FPGA dependerá del fabricante, por ejemplo, la arquitectura de los dispositivos de Xilinx dependen de bloques configurables mientras que los dispositivos de Intel se basan en elementos de función fija. Además, los FPGA incluyen elementos como memorias RAM ó procesadores digitales de señales (DSP).

La Figura 3.24 muestra una estructura simplificada de un PFGA desarrollada por Xilinx, donde se puede observar que se compone principalmente de bloques lógicos, bloques de entrada/salida y las interconexiones.

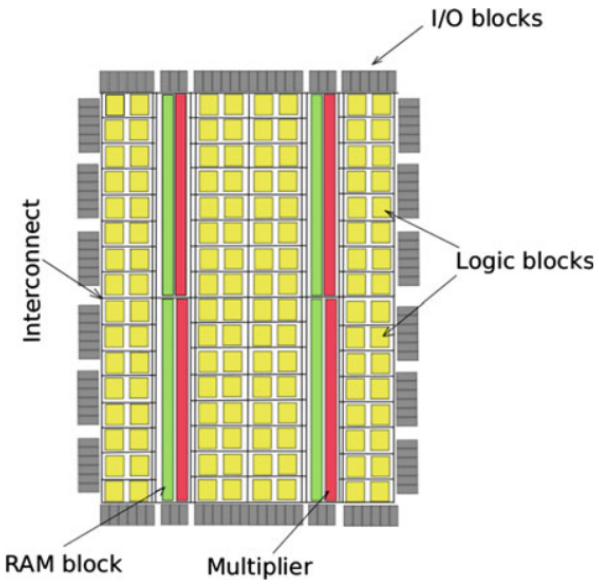

Figura 3.24: Estructura de bloques simplificada de un FPGA Spartan-3 de Xilinx [43].

3.4.1. Componentes de un FPGA

Se debe hacer una descripción personalizada de los componentes de cada FPGA según el fabricante. En este trabajo sólo se mencionarán los componentes de FPGAs diseñadas por Xilinx debido a que esas fueron las que se usaron en el presente.

Bloques lógicos

Xilinx denomina *Bloque lógico configurable* a sus bloques lógicos mientras que Intel los denomina *Modulo lógico adaptativo*

Bloques Lógicos Configurables. Un CLB es el componente principal de un FPGA y le permite al usuario implementar cualquier función lógica dentro del chip. Cada CLB se compone internamente de *slices*, la cantidad que tenga de estas

dependerá del modelo de FPGA. Para el caso de las series-7 de Xilinx, cada CLB se compone de 2 slices. Un ejemplo de la distribución interna de CLBs y Slices se muestra en la Figura 3.25.

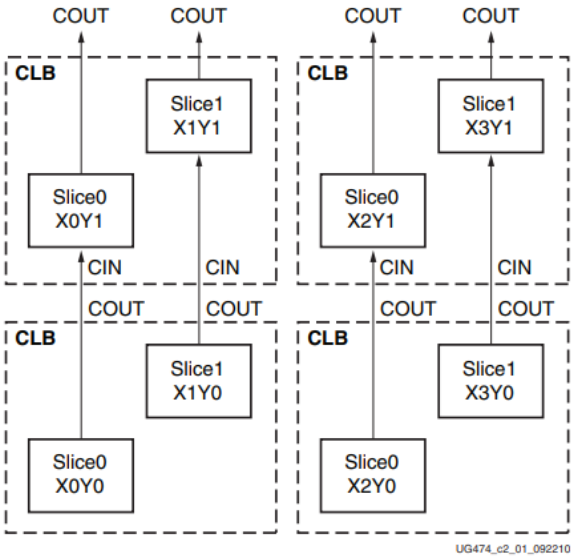

Figura 3.25: Distribución interna de CLBs y Slices [44].

Estos dos *slices* no tienen conexiones directas entre sí. Las herramientas de Xilinx permiten ubicar cada *slice* con estas definiciones [44]:

- Una X seguida de un número identifica la posición de cada slice en un par, así como la posición de la columna en el slice. El número X cuenta los cortes comenzando desde la parte inferior en la secuencia 0, 1 (para la primera columna CLB); 2, 3 (para la segunda columna CLB); etc.
- Una Y seguida de un número indica la fila del slice. El número sigue siendo el mismo dentro de un CLB, pero cuenta una secuencia desde una fila CLB a la siguiente fila CLB, empezando por abajo.

Cada Slice se compone de 4 LUTs de 6 entradas y 8 elementos de almacenamiento, multiplexores en consecuencia y *carry logic* (sirven para soportar de manera eficiente implementaciones de operaciones matemáticas). Todos los elementos mencionados anteriormente son los que proveen la lógica, aritmética y las funciones de la ROM.

Algunas Slices, además, soportan otras funciones como almacenar datos usando una RAM distribuida y corrimiento de datos con registros de 32 bits. Las Slices capaces de realizar estas dos funciones extras se llaman SliceM, mientras que las otras son SliceL. Cada CLB puede contener dos SliceL y una SliceM.

Look-Up Table (LUT). Una LUT es un generador de funciones basado en RAM y es el principal recurso para implementar funciones lógicas[43]. En el caso de las series-7, cada generador de funciones es implementado en LUTs de 6 entradas. Estas 6 entradas son independientes (Entradas A - A1 hasta A6) y tiene dos salidas independientes (O5 y O6) para cada uno de los cuatro generadores de funciones en una Slice (A,B,C y D). Ademas de los LUTs, cada Slice contiene tres multiplexores (F7AMUX, F7BMUX y F8MUX).

Bloques de entrada/salida Estos bloques tienen tres rutas de señal principales: la ruta de salida, la ruta de entrada y 3 rutas de estado. Cada ruta tiene su propio par de elementos de almacenamiento que pueden actuar como registros. Los bloques de entrada/salida proporcionan una interfaz programable, unidireccional o bidireccional entre el pin del paquete y la lógica interna del FPGA[43].

3.4.2. Hard Macro

Una *Hard macro* física es una función lógica que ha sido creada para componentes de una familia de dispositivos específica. Con una hard macro se puede asignar relativamente la ubicación en la FPGA de un grupo de celdas con diferentes objetivos como: establecer la ubicación de celdas para que queden más unidas, reducir las rutas y optimizar tiempo, entre otros. Se establecen mediante comandos en la consola Tcl o añadiendo la línea de comando correspondiente en el archivo de *XDC Constraints*. Una de sus principales ventajas es que permite preservar todas las características de un bloque diseñado, desde el rutado hasta la configuración de LUTs. Este permite replicar el circuito en diferentes regiones del FPGA asegurando que todos tienen las mismas características.
Los comandos para generar una Hard macro son los siguientes:

- *create_macro*: Este comando crea un nuevo objeto macro. Los nombres de las

macros deben ser únicos, de lo contrario arroja un error. La sintaxis es la siguiente:

create_macro <name>

- *update_macro*: Este comando agrega las celdas en una hoja y las ubicaciones relativas (RLOC) al macro. Todas las celdas deben especificarse a la vez ya que no se permiten definiciones parciales. La sintaxis es la siguiente:

  update_macro <name_macro><cell RLOC list>

  Donde: *cell-RLOC list* hace referencia a la lista de los pares relativos junto con las celdas. Todas las celdas de las macros deben especificarse sólo una vez.

- *delete_macros*: Este comando elimina determinada macro. Su sintaxis se muestra a continuación:

  delete_macro <name_macro>

Todos los comandos antes mencionados se deben ingresar mediante la consola TCL que se encuentra en el software *Vivado* como se muestra en la Fig. 3.26 donde se crea una hard macro con el nombre *m0*.

Figura 3.26: Comando para la creación de una hard macro en la consola TCL.

Capítulo 4

Metodología

En este capítulo se describe el esquema propuesto basado en biometría cancelable para señales ECG, dicho esquema se ilustra en la Fig 4.1.

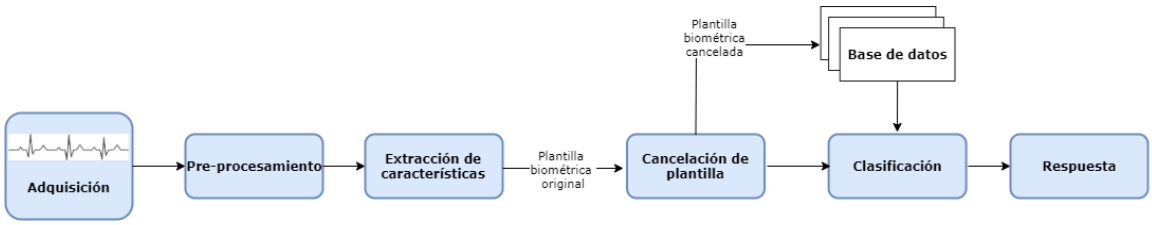

Figura 4.1: Diagrama a bloques del esquema propuesto.

El orden en el que se presenta la información es el siguiente:

- Adquisición de datos: Muestra el banco de datos utilizado y las características de las señales propuestas.
- Pre-procesamiento: Se describen las técnicas de filtrado para eliminar el ruido de la señal y el acondicionamiento que se realizó antes de la extracción de características.
- Extracción de características: Muestra como se obtuvieron las características.
- Cancelación de plantilla biométrica: Explica el procedimiento que se siguió para la cancelación de las características originales y la creación de la plantilla biométrica modificada.

- Generación de la base de datos: Detalla como se realizó la selección para la base de datos.
- Clasificación: Describe el algoritmo utilizado para la clasificación de los individuos respecto la base de datos.
- Decisión: Define como se realiza la toma de decisiones sobre si el usuario es genuino o un impostor.

4.1. Adquisición de datos

Para este trabajo se consideró el uso de señales ECG debido al buen rendimiento. Se utilizó el banco de datos *"BIDMC PPG and Respiration Dataset"* de la página web *PhysioNet* que contiene señales EGC y PPG de 53 individuos. Cada señal tiene una duración aproximada de 8 minutos a una frecuencia de muestreo de 125 Hz.
Sólo se tomaron 50 individuos de todo el conjunto, de cada uno se tomó la señal etiquetada como ÏI"que corresponde a la segunda derivación del ECG. La información de todo el conjunto se obtuvo en formato .mat para ser procesada en $MATLAB$. En la Fig. 4.2 se muestra un segmento de la señal ECG del individuo 1.

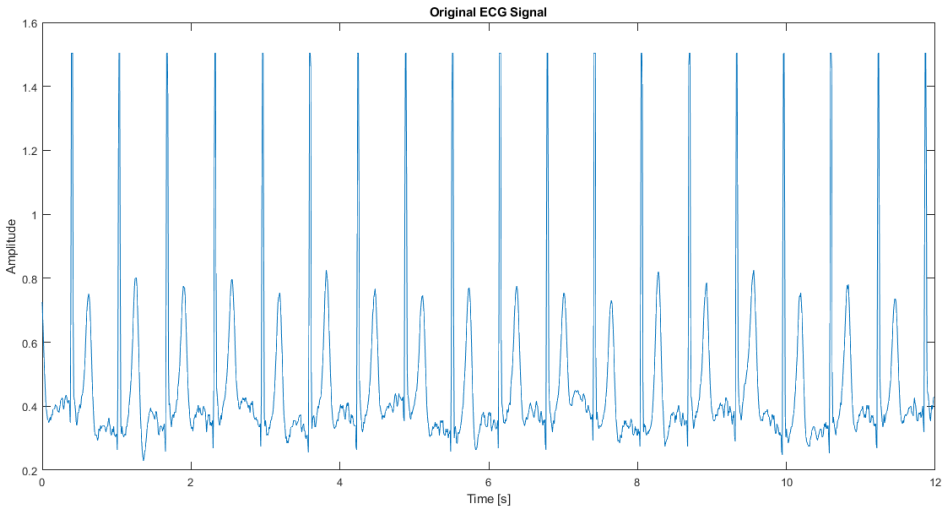

Figura 4.2: Señal ECG del individuo 1 tomada del banco de datos.

4.2. Pre-procesamiento

Después de que la señal es adquirida aún no esta lista para su uso en diagnóstico ya que, desafortunadamente, se ve alterada por varios factores, por lo que, durante esta etapa se realiza un filtrado y acondicionamiento previo para posteriormente extraer las características propuestas.
En la Fig. 4.3 se muestran las etapas que componen el pre-procesamiento y posteriormente se describe en que consiste cada una.

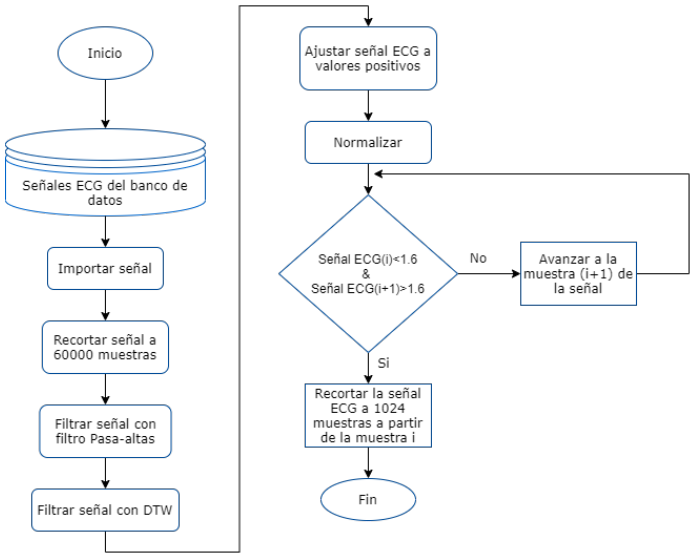

Figura 4.3: Diagrama de flujo correspondiente al pre-procesado de la señal.

El primer paso es seleccionar la señal ECG correspondiente de cada individuo y recortarla a 60000 muestras, ya que es el número máximo de muestras que tienen todas las señales, lo que corresponde a 7.99 minutos. Luego se realiza el filtrado de la señal que se detalla en la siguiente sección.

4.2.1. Etapa de filtrado

El ruido de la base de línea y el de la interferencia eléctrica son los componentes que más afectan las señales ECG y por lo tanto deben ser removidos.
El ruido de la base de línea se genera debido a la respiración del paciente. Su rango

de frecuencia es bajo, comprende entre 0.05 y 0.5 Hz. Los filtros digitales FIR e IIR son capaces de remover este ruido con un diseño pasa altas , sin embargo, los filtros IIR introducen una distorsión en la señal debido a que este tipo de filtros tienen una respuesta de fase no lineal que no es tolerada por las señales ECG. Dejando así los filtros FIR como la mejor opción.
Para eliminar el ruido mencionado anteriormente se propuso un filtro lineal Equiripple pasa-altas de orden 200 con una frecuencia de corte de 2 Hz mediante el toolbox de filtros digitales de MATLAB.

Posteriomente, se debe eliminar el ruido de la tensión eléctrica. Este ruido suele presentarse entre los 50 a los 60 Hz (dependiendo la frecuencia estándar que se usa). Esta componente, al ser de alta frecuencia, puede eliminarse con filtrado mediante la Transformada Wavelet Discreta.
Para elegir la función Wavelet madre se consideró la familia de las waveletes ortogonales Daubechies (db) y se tomó la función daubechies 4 por ser la que presenta mayor similitud con la señal ECG. Posteriormente, se tomó el nivel 1 de descomposición que corresponde a las frecuencias entre 0-62.5 Hz debido a que el nivel 2 genera distorsión en la señal y se tomaron los coeficientes de aproximación para reconstruir la señal eliminando las componentes de alta frecuencia.
En la 4.4 se muestra la señal original, luego la señal filtrada con el filtro pasa-altas Equiripple y finalmente la señal procesada por el filtrado mediante DWT.

4.2.2. Acondicionamiento

Durante el acondicionamiento se procesó la señal para poder segmentarla y ajustarla para extraer sus características. A continuación se enlistan las etapas que componen este bloque.

- Se obtiene el mínimo absoluto de la señal y se le suma para obtener una señal con valores positivos, como se describe en 4.2.1

$$\text{Señal con valores positivos= señal original + mínimo de la señal original} \tag{4.2.1}$$

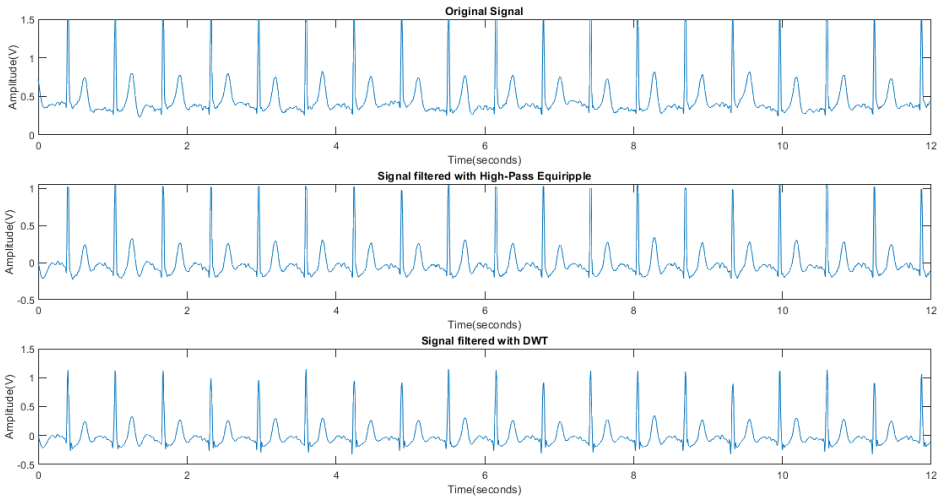

Figura 4.4: Señal ECG después de la etapa de filtrado.

- Se normaliza la señal obtenida con valores positivos siguiendo 4.2.2, donde el umbral máximo deseado se estableció a 1 V.

$$\textit{señal normalizada} = \frac{\textit{señal con valores positivos} \ * \ \textit{umbral máximo deseado}}{\textit{valor máximo de la señal con valores positivos}} \tag{4.2.2}$$

- Adicionalmente se suma un valor de DC de 1.5 V siguiendo 4.2.3.

$$\text{señal final} = \text{señal normalizada} + \text{nivel de DC} \tag{4.2.3}$$

- Para la extracción de características la señal se recortó en segmentos de 1024 muestras que corresponden a 8.192 segundos. Para realizar este corte se realiza la detección de cruce de nivel a 1.6 V para asegurar la similitud de todos los segmentos.

Al final del acondicionamiento se obtuvieron 57 segmentos por individuos como el que se muestra en la Fig. 4.5.

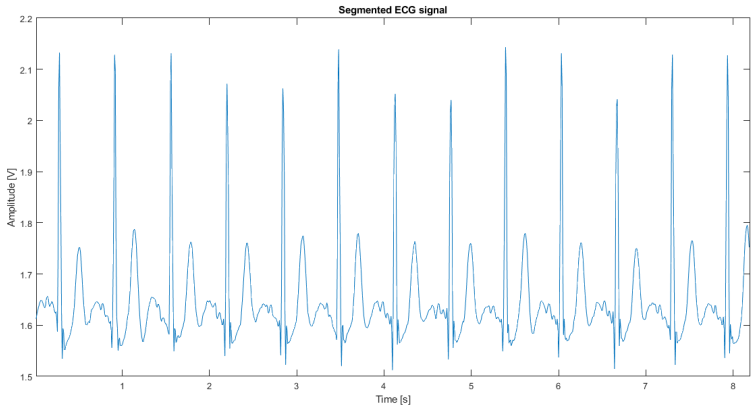

Figura 4.5: Segmento de la señal ECG.

4.2.3. Extracción de características

En este trabajo se consideró el uso de características temporales y estadísticas como se mencionó anteriormente. A continuación se muestra una lista de las características extraídas.

- **Amplitud y diferencia de instantes entre ondas R**
  Para la obtención de estos valores se utilizó el algoritmo Pan-Tompkins en *MATLAB* proporcionado por [45] que arrojá la amplitud de cada onda R y el valor de la muestra donde se encuentra cada onda.
  En el siguiente diagrama de la Fig. 4.6 se muestra el orden del algoritmo utilizado.

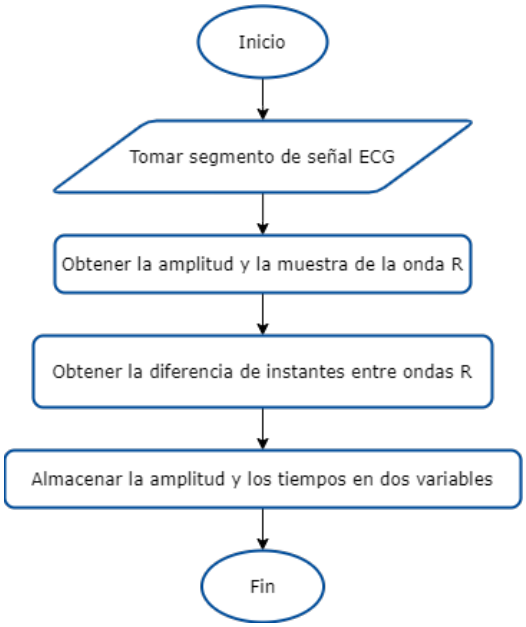

Figura 4.6: Diagrama de flujo para extraer la amplitud y la diferencia de instantes entre ondas R.

- **Actividad (Parámetro Hjorth)**
  Esta característica se obtiene calculando la varianza de la señal y su diagrama de flujo se muestra en la siguiente Fig. 4.7.

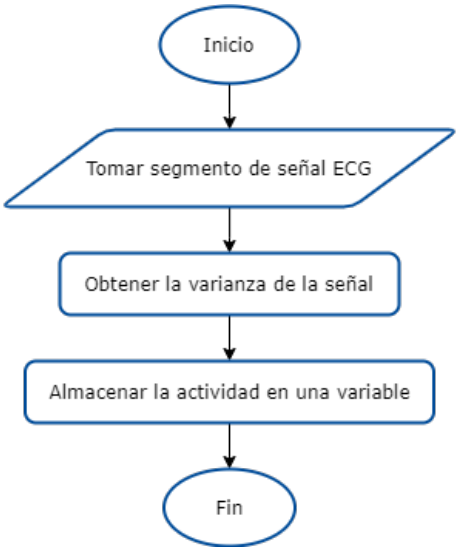

Figura 4.7: Diagrama de flujo para extraer la actividad de la señal.

- **Movilidad (Parámetro Hjorth)**

Para obtener esta característica se debe obtener la varianza de la señal original, la primer derivada de la señal y luego la varianza de la primer derivada. Posteriormente se debe calcular la raíz cuadrada de la razón de la varianza de la primer derivada sobre la varianza de la señal original. El diagrama de flujo se puede observar en la Fig. 4.8.

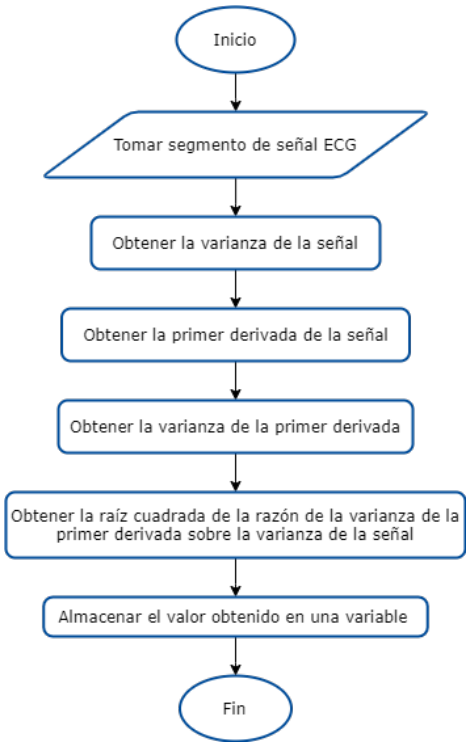

Figura 4.8: Diagrama de flujo para extraer la movilidad de la señal.

- **Complejidad (Parámetro Hjorth)**
  Se obtiene calculando la razón de la primer derivada de la movilidad sobre la movilidad original. El diagrama de flujo se observa en la Fig. 4.9.

- **Asimetría**
  La asimetría se puede obtener mediante el comando *skewness* en $MATLAB$. El diagrama de flujo para obtenerla se muestra en la Fig. 4.10.

- **Curtosis**
  Para obtener la curtosis se utiliza el comando *kurtosis* en $MATLAB$ y su diagrama de flujo se muestra en la Fig. 4.11.

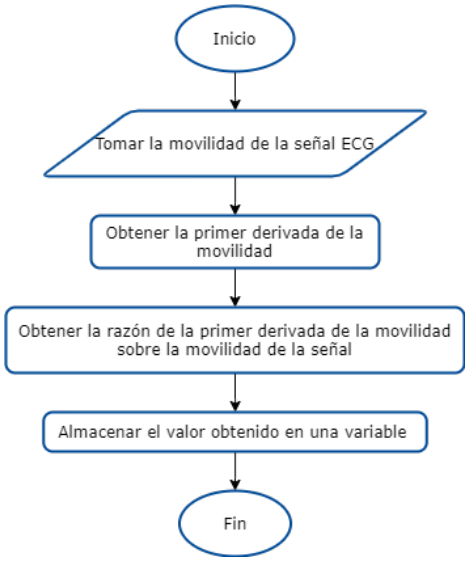

Figura 4.9: Diagrama de flujo para extraer la complejidad de la señal.

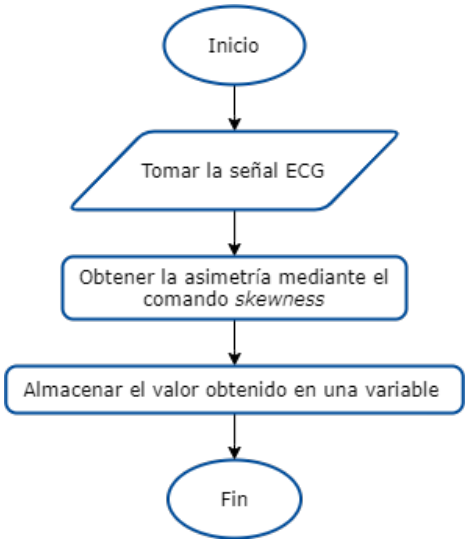

Figura 4.10: Diagrama de flujo para extraer la asimetría de la señal.

Finalmente estas características se concatenan formando un vector de longitud 7, obteniendo en total 57 vectores de características.

Para la cancelación de plantillas es necesario tomar al menos 9 vectores de características, por lo que, se seleccionaron 12 vectores de características para formar una matriz de 12x7 que posteriormente será modificada por la respuesta de la función física inclonable.

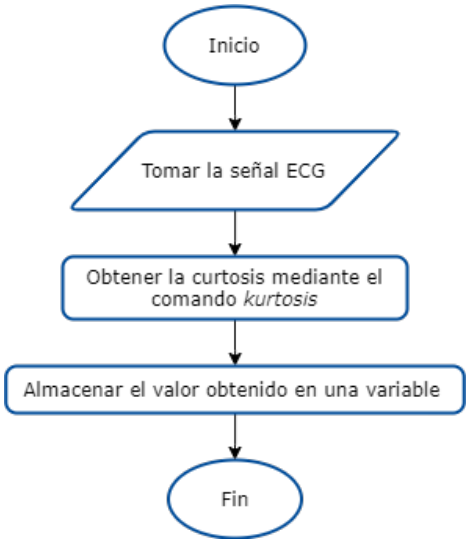

Figura 4.11: Diagrama de flujo para extraer la curtosis de la señal.

4.3. Cancelación de plantilla biométrica

La cancelación o modificación de la plantilla biométrica se logra mediante una serie de transformaciones que involucran una plantilla de cancelación. Esta plantilla se puede obtener por diferentes técnicas. En este trabajo se propone la obtención de dicha plantilla a través de funciones físicamente inclonables.

4.3.1. Implementación de la función físicamente inclonable

Las funciones físicamente inclonables pueden implementarse en diferentes dispositivos de acuerdo a la topología que se desee. En este trabajo se propuso el diseño de *Configurable Ring-Oscillator (CROPUF)* debido a las facilidades que brinda de poder modificar la salida del dispositivo variando el retardo de la ruta y su fácil implementación.

Este diseño se compone de 6 compuertas NOT, 3 multiplexores y una compuerta AND que retroalimenta la salida del ciclo y habilita la PUF. Tiene 2 entradas, un habilitador y una llave que se conecta a los selectores de los multiplexores y permitirá configurar el retardo, y sólo cuenta con una salida que corresponde a la respuesta. En la Fig. 4.12 se muestra el diseño de un CROPUF.

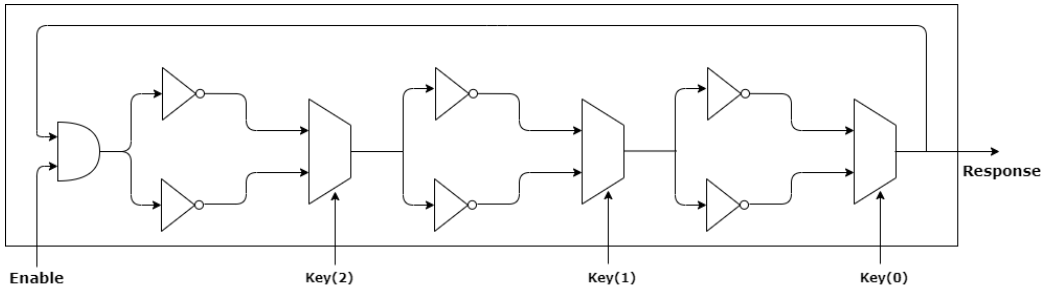

Figura 4.12: Arquitectura de un CROPUF.

Para poder generar el bit de respuesta es necesario comparar las salidas de dos estructuras CROPUF. Los dos CROPUF oscilaran a frecuencias diferentes debido a los retardos generados por las compuertas lógicas y la ruta, la estructura que oscile a una frecuecia mayor determinara si el bit de salida es 1 o 0.
Se consideraron 128 estructuras CROPUF, para seleccionar el CROPUF a evaluar se utilizó un multiplexor cuya salida va conectada a un contador que se activa durante determinado tiempo. Al finalizar este tiempo se almacena temporalmente la frecuencia medida y después comienza a oscilar el siguiente CROPUF, cuando este termina se comparan ambas frecuencia y se define una salida. En la Fig. 4.13 se muestra la arquitectura de esta etapa.

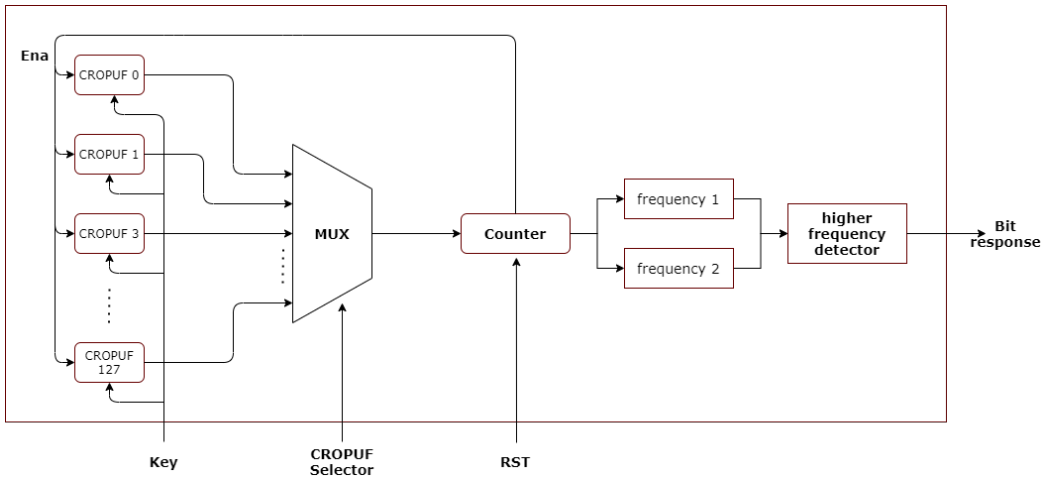

Figura 4.13: Arquitectura para determinar el bit de respuesta.

Se pueden hacer multiples comparaciones e ir concatenando las respuestas para generar una cadena de bits. De todo el conjunto de CROPUF se pretende obtener 50 cadenas de bits diferentes, una para cada individuo, que servirán para la cancelación

de la plantilla de características.
La plantilla de características es una matriz de 12x7 y contiene 84 datos por individuo, sin embargo, sólo se toman en cuenta matrices de 6x7, es decir sólo se toman en cuenta 6 periodos de la señal ECG por si hay segmentos de la señal dañados.
Se necesita una matriz de cancelación de las mismas dimensiones, por lo que, se deben obtener cadenas de 42 bits provenientes de la función física inclonable que posteriormente se distribuiran en una matriz. Para obtener dichas cadenas se necesitan realizar 43 comparaciones.
Se realizó una selección aleatoria de 43 datos entre 0 y 127 por individuo para formar un vector de posiciones que indicarán el número de CROPUFs que se van a comparar. En total se obtuvieron 50 vectores de posiciones, uno por cada individuo, y se almacenaron en una memoria ROM. Cada vector de posiciones se asignó a cada usuario en el orden en el que se va registrando en el sistema, por consiguiente se debe indicar el número de individuo al momento de leer la memoria para obtener el vector de posiciones correspondiente. En la Fig. 4.14 se muestran los pasos para obtener la cadena de bits para la cancelación.

En la Fig. 4.15 se observa el diagrama de flujo que detalla la comparación que se realiza para obtener un bit de respuesta.

El usuario indicará su identidad a través de una PC utilizando el protocolo RS-232 para comunicar el software con el FPGA y retornará la cadena de bits para ser procesada después. El control de todos los bloques se logra a través de una máquina de estados. En la Fig. 4.16 se muestra la arquitectura final y relevante del sistema, el bloque *CROPUF* hace referencia al diseño de la Fig. 4.13.

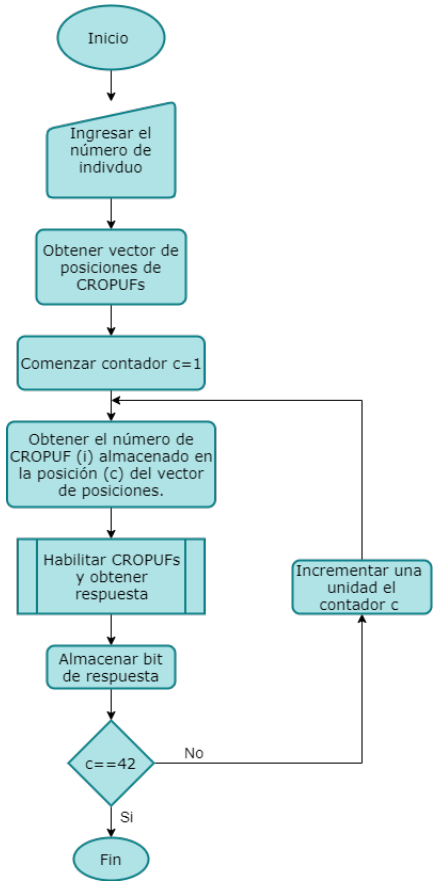

Figura 4.14: Diagrama de flujo para obtener la cadena de bits.

Una de las principales consideraciones al momento de implementar varias PUF es asegurar que todas sean identicas, en este caso es necesario que todos los CROPUF sean identicos emplazados en el FPGA, deben tener el mismo rutado y deben tener las mismas configuraciones de LUTs. De esta manera se asegura que los retardos generados durante la ruta se deben unicamente a los retardos inducidos por las compuertas lógicas debido a las variaciones de los procesos de fabricación.
Para lograr esta simetría de los CROPUF se debe generar una *Hard macro* y establecer manualmente en que CLB y en que LUT se va a emplazar cada elemento. La distribución interna de los FPGA de la serie 7 se compone en su mayoria de *slices* de tipo L y tipo M. Los slice L hacen referencia a la lógica y cuenta con recursos de almacenamiento y LUT, lógica de acarreo y multiplexores, mientras que los slice tipo M hacen referencia a memoria y en estas las LUT se pueden usar como RAM distribuidas. La mayoria de las Slices disponibles son del tipo L.

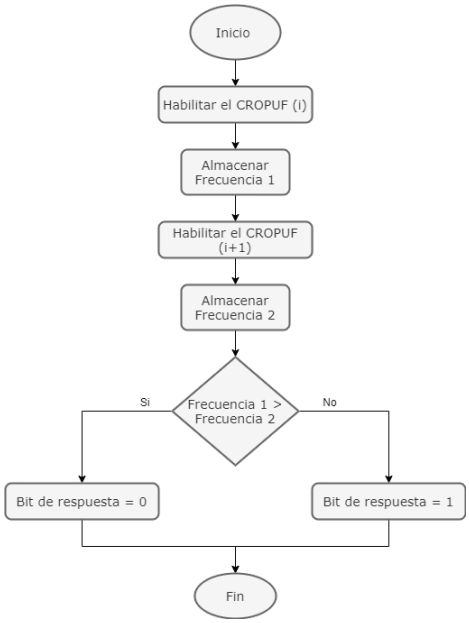

Figura 4.15: Diagrama de flujo para obtener un bit de respuesta.

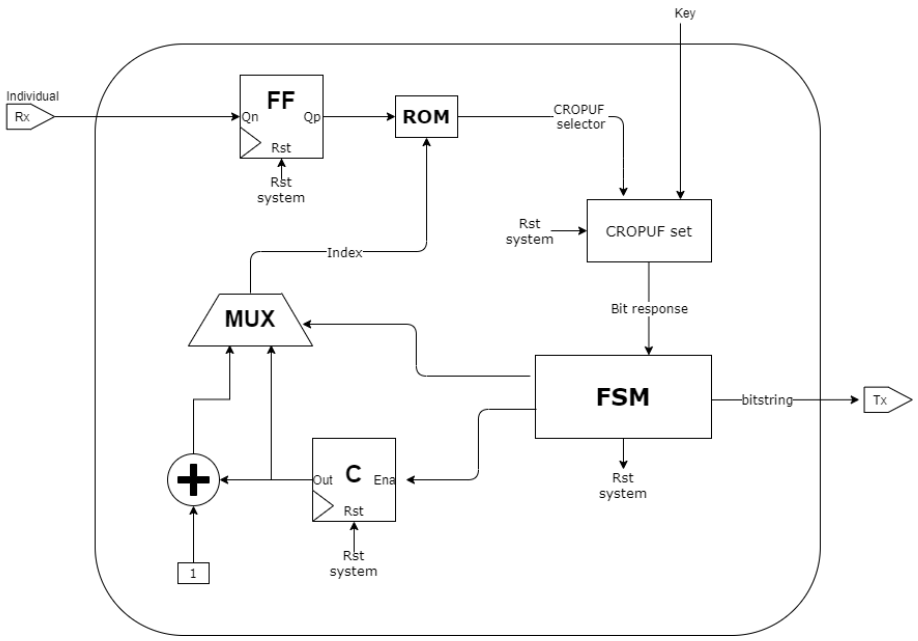

Figura 4.16: Diagrama a bloques del sistema general.

La generación de la Hard Macro se puede lograr mediante el ingreso de comandos en la consola TCL o se puede hacer directamente en el archivo xdc. Se deben agregar dos instrucciones, primero, se debe crear la macro con el comando *create_macro*

seguido del nombre que se le asignará a la misma, en este caso se deberán crear 128 macros, una por cada estructura CROPUF. Un ejemplo para la creación de una macro es el siguiente:

create_macro nombre_macro.

Posteriormente, con el comando *update_macro* se asigna una ubicación relativa a cada elemento de la macro. Aquí también es necesario indicar el nombre de la macro seguido del nombre de la instanciación como aparece en el Netlist y de la ubicación de cada LUT como se describe a continuación:

update_macro nombre_macro {instanciación_LUT CoordenadaXCoordenadaY}

Posteriormente hay que establecer la posición final de las LUT dentro de cada Slice, para lograr esto se utilizan dos comandos. El primero se utiliza para indicar el tipo de LUT en el que se va a emplazar el diseño seguido del nombre de la instanciación, tal como se encuentra en el Netlist como se indica a continuación:

set_property BEL TIPO_LUT [get_cell nombre_instanciación]

El segundo comando sirve para indicar la slice donde se colocará, basta con indicar sus coordenadas seguido del nombre de la instanciación como se muestra a continuación:

set_property LOC SLICE_XcoordenadaYcoordenada [get_cells nombre_instanciación]

Para implementar la Hard Macro se propuso utilizar una LUT por cada elemento lógico del diseño, utilizando así 10 LUTs por CROPUF. Para evitar agregar retardo

durante las rutas se emplazaron lo más cerca posible formando una matriz de Slices de 2x2, utilizando 3 LUTs en el caso de 3 CLBs y en el último CLB sólo 1 LUT. En la Fig. 4.17 se puede observar la distribución interna de cada CROPUF ya emplazada en el FPGA.

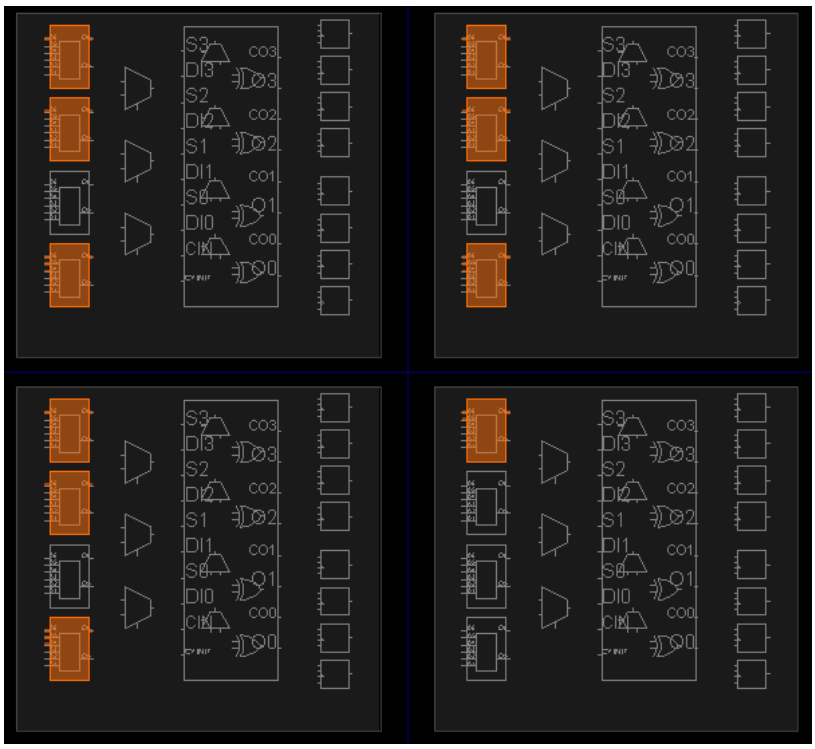

Figura 4.17: Distribución interna de un CROPUF en el FPGA.

La distribución mostrada en la Fig. 4.17 se replico 128 veces para implementar los 128 CROPUFs que se habían mencionado anteriormente. Se seleccionó una región en el FPGA donde permitiera emplazar los CROPUF lo más cerca posible y que estuviera cerca en relación con los puertos de entrada-salida del diseño final. En la Fig. 4.18 se puede visualizar que los CROPUF se emplazaron en la región $X0Y2$.

En la siguiente Fig. 4.19 se muestra la distribución de los 128 CROPUF distribuidos en una matriz de 32x8 slices.

Sólo se consideró asignar manualmente las ubicaciones de los CROPUF, el routeado del diseño restante que incluye multiplexores, Flip-flops, máquinas de estado, etc se asignó automáticamente por el sintetizador de Vivado, sin perjudicar o alterar la distribución ya establecida. El resultado de las conexiones finales se pueden visualizar

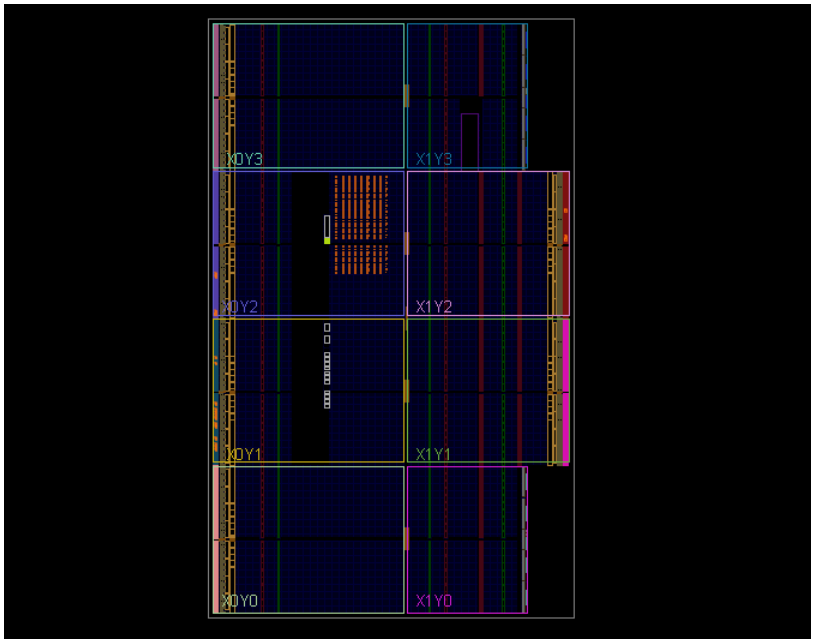

Figura 4.18: Vista general interna del FPGA.

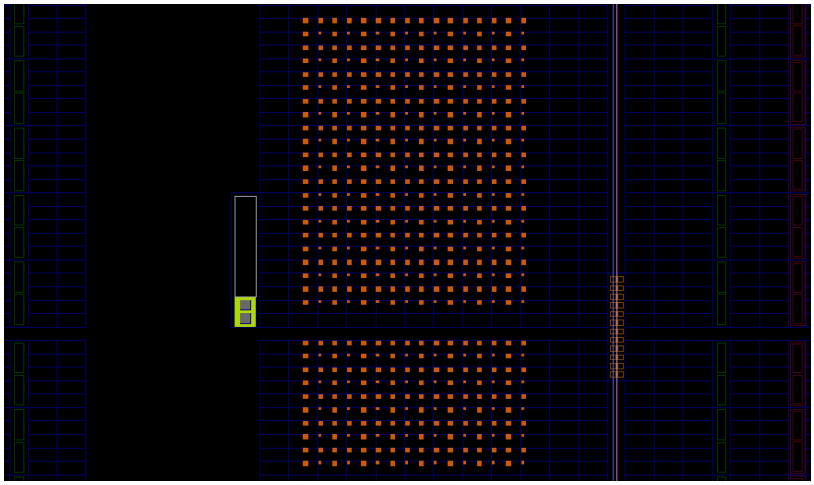

Figura 4.19: Distribución de los CROPUF en el FPGA.

en la Fig. 4.20.

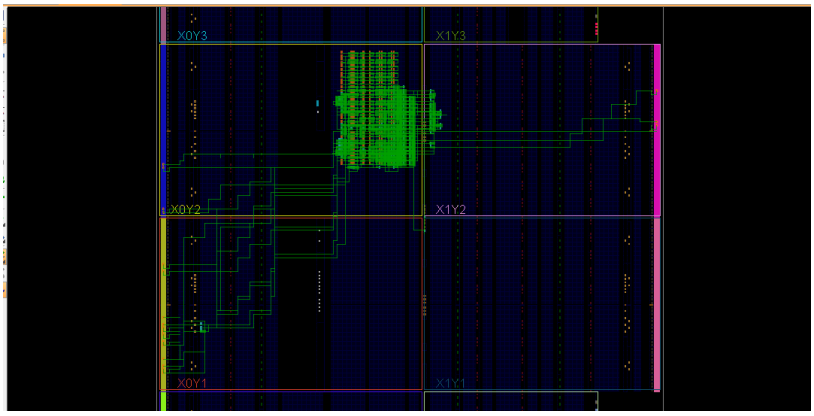

Figura 4.20: Conexiones finales de la implementación.

4.3.2. Algoritmo de corrección de errores

La implementación de algoritmos de corrección de errores es fundamental cuando se emplean PUF para la generación de llaves, ya que es necesario obtener la misma respuesta. La mayoría de las veces las cadenas de bits obtenidas por una PUF varían en algunos bits, esto ocurre cuando se comparan dos estructuras PUF que oscilan a frecuencias muy similares.

El algoritmo de corrección de errores utilizado funciona como un código de corrección de errores de repetición y consiste en dos etapas, la primera es la generación de los Helper-Data y la segunda es la reconstrucción de la cadena de bits. Todo este proceso se realiza en software fuera del FPGA.

Durante la primer etapa se clasifica cada bit de una cadena como bit estable o inestable. Para lograr esto es necesario contar con varias respuestas del mismo PUF y detectar los bits inestables en cada una. Para la detección se toma como base determinada respuesta del PUF, en este caso se consideró la respuesta que se repitió más de un grupo de 50 respuestas. Se realiza una comparación bit a bit de esta respuesta modelo con otra respuesta del conjunto y se aplica una máscara binaria, si los bits coinciden se asigna un 1 en esa posición de la máscara y de lo contrario se asigna un 0, dando como resultado máscaras binarias de 42 bits que deben almacenarse en una memoria no volátil.

Con esta máscara binaria se desprecian los bits inestables de cada cadena de bits y se obtiene una respuesta estable de longitud menor si se presentaron bits inestables. Para obtener esos bits “faltantes”se consideran los bits laterales a esa posición del

vector y se combinan con la función lógica XOR, el resultado se coloca en la posición del bit faltante como se muestra en la Fig. 4.21 (a) obteniendo una nueva cadena de bits denominada *Respuesta ID*.

El siguiente paso es obtener los Helper-Data, para esto se utiliza el código de repetición. Primero se establece un código K para codificar de n bits $K = [k_1, k_2, ..., k_n]$. Cada bit se repite r veces para utilizarlo como un código de corrección de errores de repetición $K_{coded} = [k_{11}, ..., l_{1r}, ..., k_{a1}, ..., k_{ar}]$.

Mediante la operación lógica XOR la respuesta ID se combina con K_{coded} para obtener los Helper-Data como se indica en la Fig. 4.21 (b). Los Helper-Data deben obtenerse para cada usuario y también se almacenan en una memoria no volátil. Todo este proceso se realiza sólo una vez.

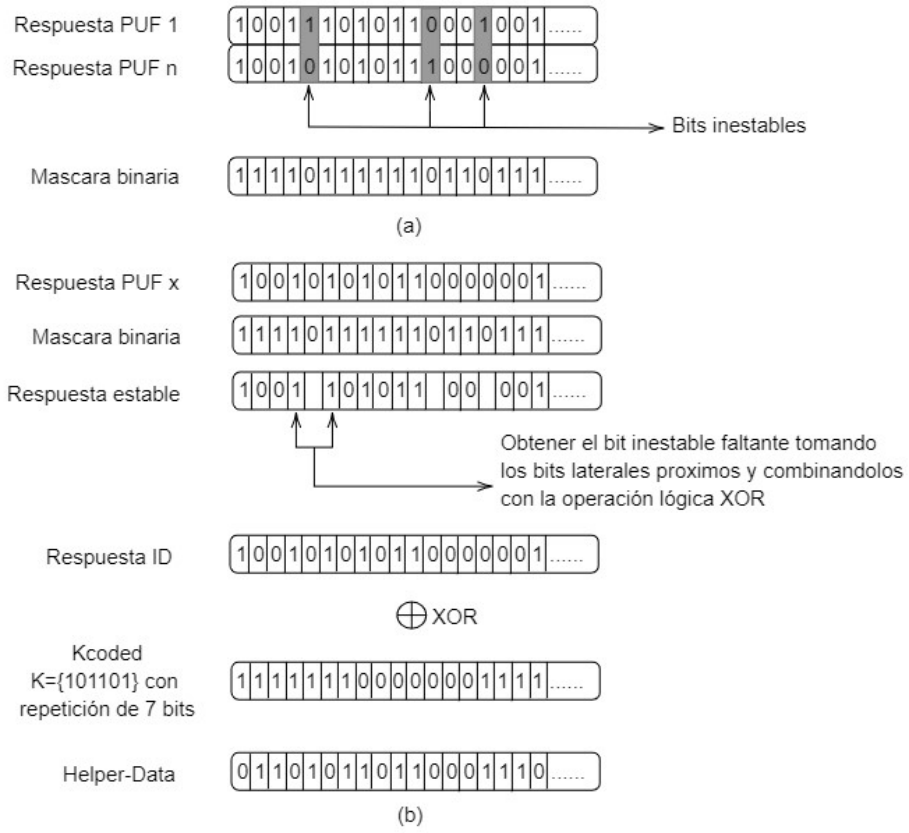

Figura 4.21: Etapa de registro para cada cadena de bits.

Existen otros algoritmos de corrección de errores como el BCH o Reed-Muller que son más complejos de implementar. La ventaja de utilizar este algoritmo propuesto

es que es fácil y es posible implementarlo ya que se clasifican los bits en bits estables e inestables, cosa que no se realiza para los otros algoritmos de corrección de errores.

Para la segunda etapa que consiste en la reconstrucción de la cadena de bits se siguen los siguientes pasos y se muestran en la Fig. 4.22.

- Obtener la cadena de bits original de acuerdo a cada individuo.
- Aplicar la máscara binaria.
- Obtener la nueva cadena de bits reemplazando los bits inestables con ayuda de la máscara binaria.
- Aplicar la operación lógica XOR a la respuesta estable y los Helper-Data para obtener la respuesta ID.
- Extraer K del Kcoded obtenido y verificar que coincida con la llave K utilizada en el registro. Si no coincide se debe repetir el proceso ya que indica que algún bit en la cadena es erróneo.

4.3.3. Método de cancelación

Se puede obtener la plantilla cancelada de cada individuo una vez que se tiene la plantilla de carácteristicas original y la plantilla de cancelación. En la Fig. 4.23 se muestra el diagrama de flujo de este proceso y más adelante se describe brevemente en que consiste cada paso.

- Se toma la respuesta ID obtenida después del algoritmo de corrección de errores.
- Se asignan los bits en una matriz de 6x7 ordenando los datos primero por filas.
- La matriz resultante del paso anterior se multiplica por la matriz de características.
- Se toma cada columna de la matriz resultante del producto y se suman todos los datos que se encuentren verticalmente.

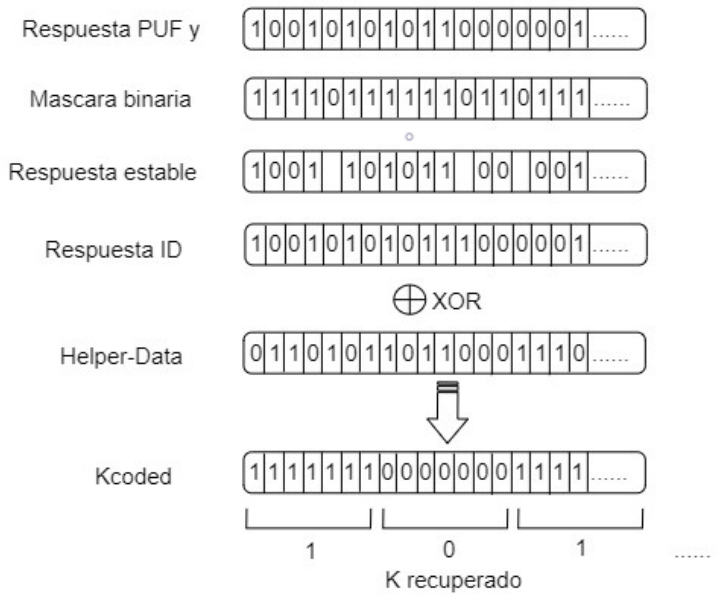

Figura 4.22: Etapa de reconstrucción de la cadena de bits por sujeto.

- Se obtiene un único vector de longitud 7 con las características canceladas.
- Dicho vector se almacena para posteriormente generar la base de datos.

Por individuo se obtienen 50 vectores de características canceladas utilizando determinada llave para la configuración del CROPUF. Dicha llave es de 3 bits por lo que se pueden realizar 8 diferentes combinaciones, esto quiere decir que, por sujeto se pueden obtener 8 vectores de características canceladas distintos.

4.4. Generación de la base de datos

Durante la etapa de registro se debe almacenar un conjunto de vectores que serviran como base para poder decidir si el individuo que desee ingresar al sistema es genuino o un impostor.
Para la creación de dicha base de datos se tomaron 10 vectores por individuo de todo el conjunto de vectores de cancelación y se almacenaron en una nueva matriz como se muestra en la Fig. 4.24, resultando así una matriz final de 500x7.

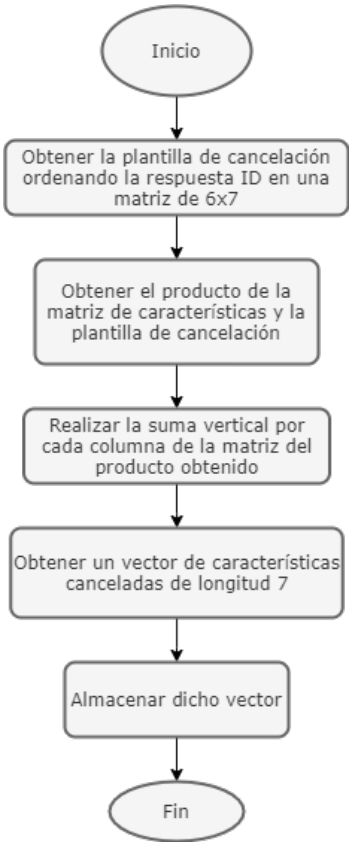

Figura 4.23: Diagrama de flujo de la etapa de cancelación de plantilla biométrica.

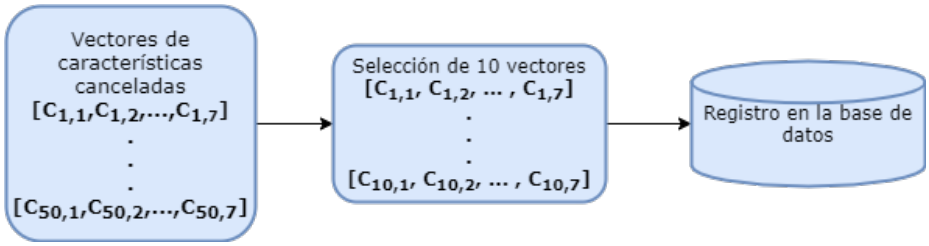

Figura 4.24: Diagrama a bloques para la creación de la base de datos por individuo.

4.5. Clasificación

La clasificación en los sistemas que trabajan en el modo de verificación se debe realizar mediante una comparación entre el vector de características canceladas del usuario que desea ingresar al sistema contra los vectores almacenados en la base de datos correspondientes a ese usuario a través de las distancias obtenidas por el algo-

ritmo de Alineamiento Temporal Dinámico (DTW). Como se mencionó anteriormente la base de datos contiene 10 vectores por sujeto por lo que se realizarán 10 comparaciones por individuo.
El diagrama de flujo de esta etapa se muestra en la Fig. 4.25.

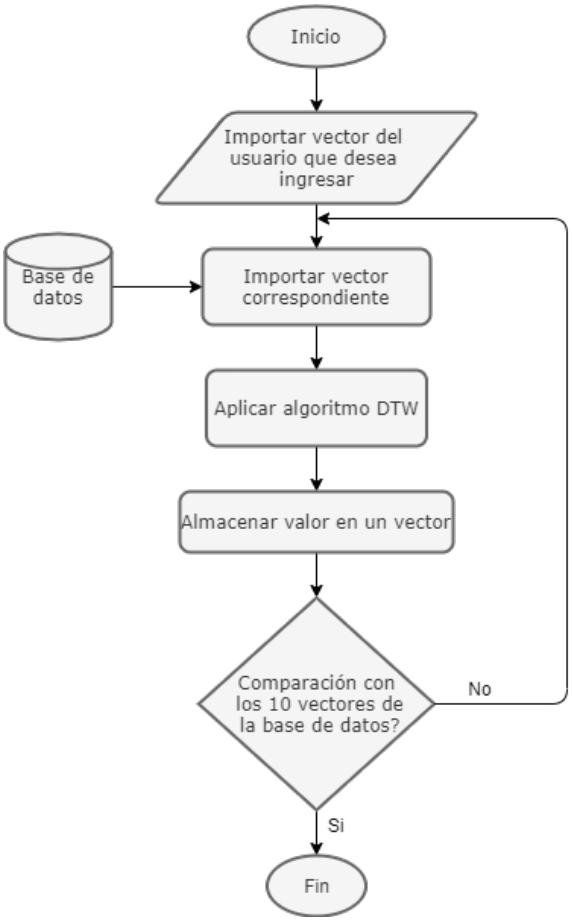

Figura 4.25: Diagrama de flujo de la clasificación.

Para la obtención del DTW se utilizó el comando *pdist2* que realiza la comparación entre dos vectores siguiendo dicho método. El diagrama de flujo del DTW se muestra en la Fig. 4.26.

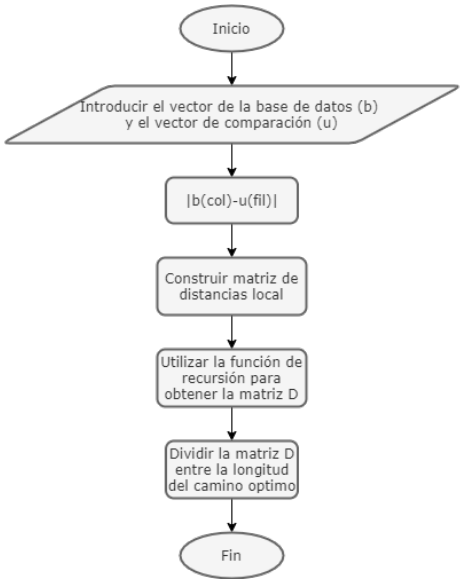

Figura 4.26: Diagrama de flujo del algoritmo DTW.

4.6. Decisión

Durante esta etapa se obtiene la distancia uno a uno de todos los vectores almacenados en la base de datos y se normalizan para obtener distancias sólo entre 0 y 1. Luego se propone cierto umbral, si el valor de distancia esta por debajo del umbral entonces el usuario es genuino y corresponde con el usuario que dice ser de lo contrario es un impostor.

Seleccionar el umbral que arroje mejores resultados va a depender de la tasa de usuarios que se reconocieron correctamente. Para cada umbral se obtiene una matriz de confusión y una curva ROC que indica el rendimiento del sistema.

Una vez determinado el umbral, cuando un usuario ingrese al sistema se comparará su vector con los diez almacenados en la base de datos y se compararán con el umbral, si almenos cinco de los diez vectores almacenados dan una respuesta positiva entonces el usuario es aceptado.

Capítulo 5

Resultados

En este capítulo se muestran los resultados de la implementación de las funciones físicamente inclonables y el rendimiento del sistema biométrico en modo de verificación. Se puede medir el rendimiento del sistema con cada configuración del CROPUF ya que la cancelación de características cambia los vectores modificados en cada caso, cambiando así el rendimiento.
También se implementó el diseño en dos FPGA del modelo Nexys 4 DDR de las series 7 de Xilinx. Durante este capítulo se hace referencia a dichos FPGA como *FPGA 1* y *FPGA 2*.
Los segmentos de señal ECG utilizados después del filtrado y acondicionamiento son como los que se muestran en la siguiente Fig. 5.1, donde se pueden observar al menos 11 periodos de los 6 que se necesitan para la extracción de características.

A continuación se muestran los resultados obtenidos para el primer sujeto registrado en la base de datos. Cabe mencionar que se realizaron 100 pruebas por sujeto y por configuración para verificar que las frecuencias obtenidas se mantuvieran constantes dentro de determinado rango y se generaran las mismas respuestas.

Un ejemplo de las frecuencias obtenidas por los CROPUF seleccionados para sujeto 1 durante el proceso de generación de las cadenas de bits son las que se muestran en la Fig. 5.2, donde se observa que las frecuencias oscilan entre 250 y 300 MHz para el FPGA 1.

Mientras que para el FPGA 2 las frecuencias obtenida fueron las que se muestran

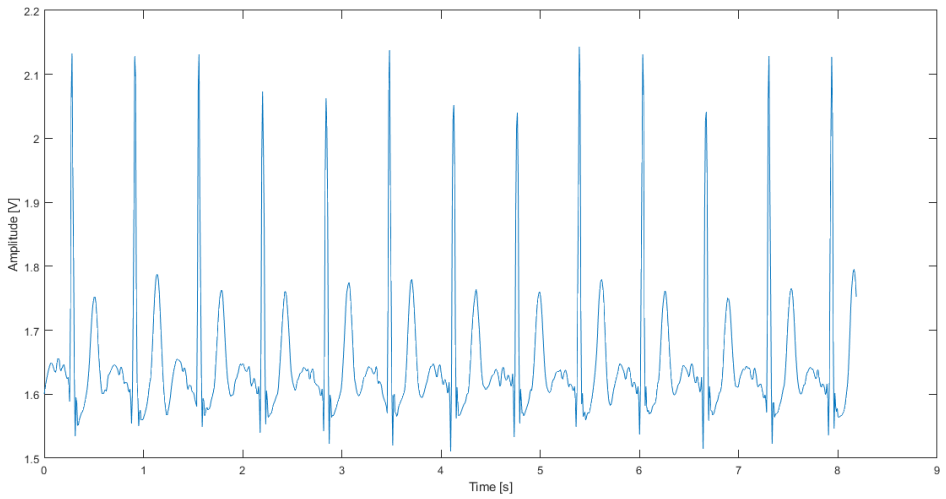

Figura 5.1: Segmento de señal ECG después del preprocesamiento.

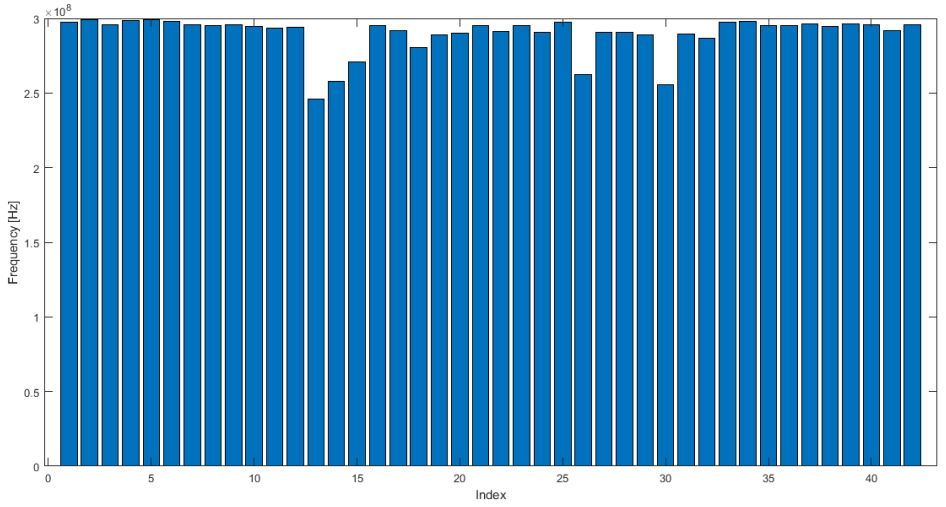

Figura 5.2: Frecuencias de la cadena de bits generadas para el sujeto 1 en el FPGA 1.

en la Fig. 5.3 y se encuentran dentro del mismo rango pero con algunas variaciones.

En general, todos los CROPUF oscilan dentro del mismo rango de frecuencias con las 8 configuraciones disponibles, debido a que las diferencias entre los retardos generados en las rutas son mínimos. Sin embargo, si existe diferencia entre las cadenas de bits generadas.

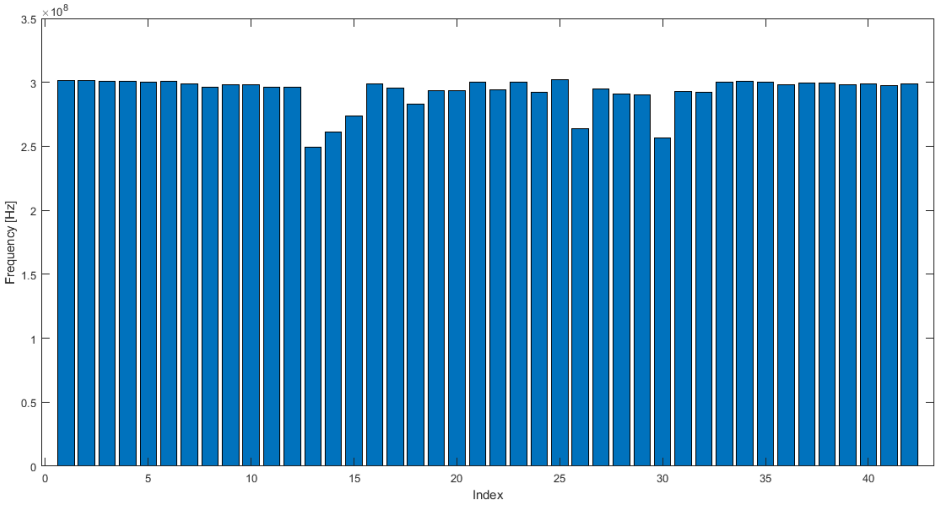

Figura 5.3: Frecuencias de la cadena de bits generadas para el sujeto 1 en el FPGA 2.

Un parámetro común que permite detectar las variaciones en las cadenas de bits es la distancia de hamming que nos indica la cantidad de bits que difiere una cadena de bits respecto a otra. Se realizaron pruebas intra-clase e inter-clase, las primeras para medir la variabilidad entre respuestas generadas por el mismo FPGA y la segunda para detectar la variabilidad entre respuestas de FPGA diferenes utilizando la misma configuración en ambos casos.

En la Fig. 5.4 se muestra la distancia intra-clase de las respuestas de 100 pruebas realizadas por 50 sujeto, considerando un total de 5000 pruebas, utilizando la configuración de CROPUF 001 y el FPGA 1. En esta misma Fig. 5.4 se puede observar que la variación máxima fue de 2 bits.

Mientras que la distancia inter-clase se muestra en la Fig. 5.5 donde se utilizó la misma configuración de CROPUF del ejemplo anterior utilizando el FPGA 1 y FPGA 2.

Los resultados de utilizar una configuración diferente se muestran en la Fig. 5.6 para la distancia intra-clase utilizando la configuración 111.

La distancia inter-clase para la misma configuración 111 se muestra en la Fig. 5.7.

Figura 5.4: Distancia de Hamming intra-clase para la configuración 001.

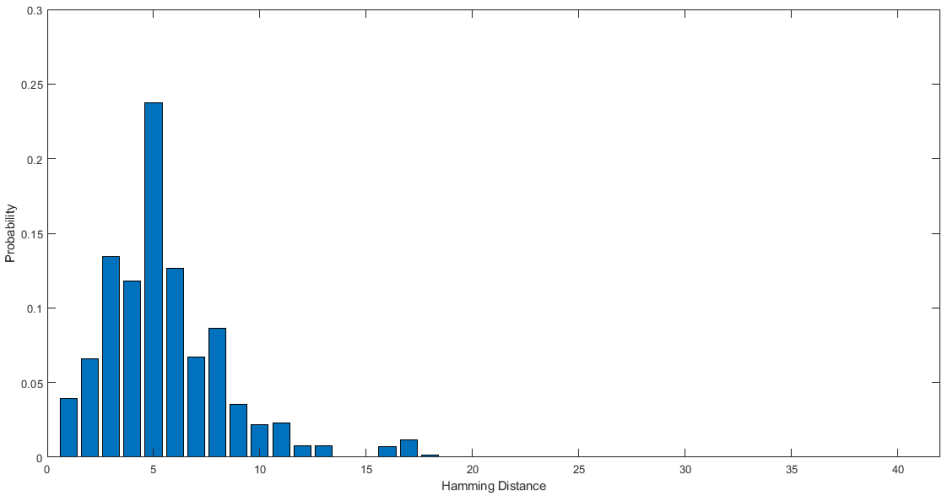

Figura 5.5: Distancia de Hamming inter-clase para la configuración 001.

La respuesta o cadena de bits obtenida para cada sujeto es inestable cuando se adquiere del FPGA, por eso debe ser procesada por el algoritmo de corrección de errores. En la siguiente lista se observan las respuestas normales arrojadas por el FPGA para un determinado individuo:

Muestra 1: 000100100110010001001010110100110001110101

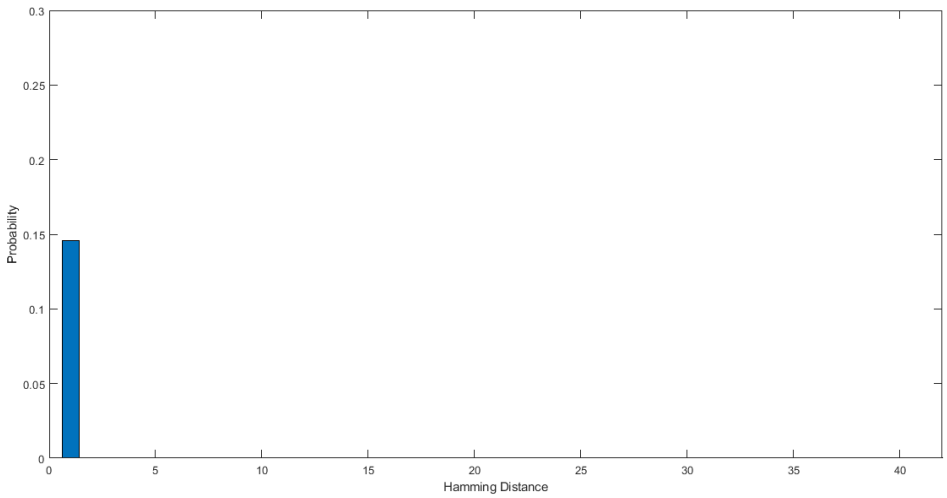

Figura 5.6: Distancia de Hamming intra-clase para la configuración 111.

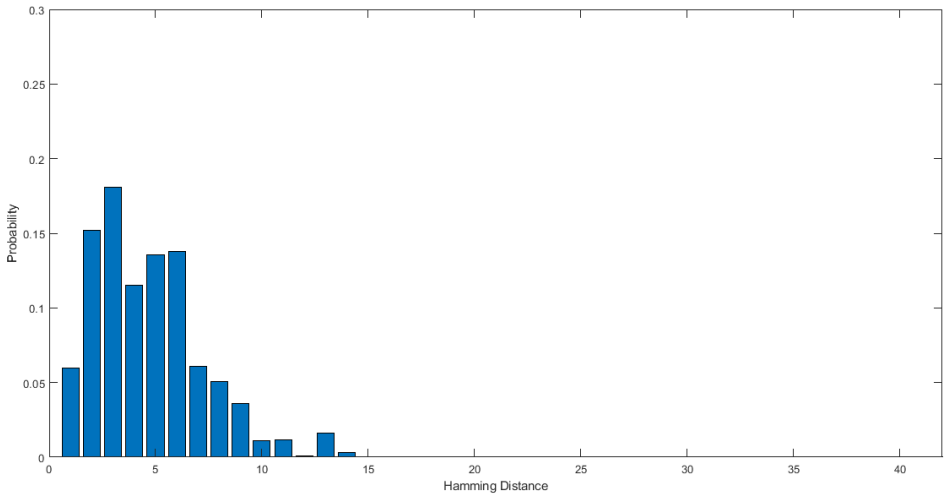

Figura 5.7: Distancia de Hamming inter-clase para la configuración 111.

Muestra 2: 00011010011001000100101011010011000111010 1

Se puede apreciar que no existe una diferencia significativa entre ambas respuestas y se puede observar en la Fig. 5.4 y 5.6 que la cantidad de bits inestables es mínima ya que sólo varían entre 2-3 bits. Sin embargo se necesita exactitud en este tipo de aplicaciones por lo que se le aplico el algoritmo de corrección de errores a las

respuestas anteriores.

Para determinar la cantidad de bits a corregir se realizó una comparación entre las 100 muestras que se tenían por individuo y se consideraron todos los bits inestables detectados al menos una vez. Después del procesamiento mencionado en el capítulo anterior la respuesta corregida es la siguiente:

00011010011001100100101011010011000111010 1

Con lo que se concluye que el algoritmo de corrección no introduce cambios significativos dentro de las respuestas y se mantiene la respuesta o comportamiento del CROPUF.
A continuación se enlistan las cadenas de bits obtenidas para el mismo individuo con cada configuración después de ser procesadas por el algoritmo de corrección de errores.

- Configuración 000

  FPGA 1: 00110010001011100101010101100101101101001 0
  FPGA 2: 00100101100011100101010101000101100100001 0

- Configuración 001

  FPGA 1: 10110011011011100111010101000101101101001 0
  FPGA 2: 10100001100011100101010101000111100010001 0

- Configuración 010

  FPGA 1: 01010011001011100101010101100101101001001 1
  FPGA 2: 00100101100011000101010101000101110011001 0

- Configuración 011

  FPGA 1: 10110011011011100111010101100101101011001 0
  FPGA 2: 00100001100011100101010101000111110010001 0

- Configuración 100

  FPGA 1: 10110000001011100111010101100101101101001 0
  FPGA 2: 00100001100011100101010101000101100101001 1

- Configuración 101

  FPGA 1: 10110001001011100111010101000101101101001 0
  FPGA 2: 00101001000011100101010101000101100110101 1

- Configuración 110

  FPGA 1: 00110011001001100101010101100101101001001 1
  FPGA 2: 00110001100001100101010101000101100101001 1

- Configuración 111

  FPGA 1: 10110011001001100111010101100101100111001 0
  FPGA 2: 01011001100001100101010101000101000010101 1

Se puede observar que existe una variación de bits entre las respuestas de ambas FPGA a pesar de implementar el mismo diseño, lo cual es una ventaja y característica que le da el dispositivo ya que no podrá replicarse la salida al menos de contar con el mismo dispositivo que se utilizó durante la etapa de registro.

En la Tabla 5.1 se muestran las características originales y canceladas para un sujeto pero con diferente configuración de CROPUF en el mismo FPGA.

La evaluación de los sistemas en modo de verificación se hace a través de las curvas ROC, en el eje x se grafica la tasa de falsos positivos y la tasa de verdaderos positivos en el eje y. De las curvas ROC se puede obtener el área bajo la curva (AUC), la tasa de igual error (EER) y la exactitud del sistema.

A continuación se muestran los resultados obtenidos relacionados con el rendimiento del sistema utilizando la misma configuración de CROPUF y el mismo FPGA

Tabla 5.1: Características obtenidas con diferentes configuraciones

Llave de configuración	Característica						
	Actividad	Movilidad	Complejidad	Asimetría	Curtosis	Tiempo R	Amplitud R
Original	0.010109	0.505024	0.861488	2.6505606	10.331916	0.6400	2.072112
000	0.027097	0.573666	2.811595	11.313295	21.042516	2.5280	6.191077
001	0.044346	0.574762	3.960895	11.732176	37.379220	1.9680	6.211911
010	0.025473	1.162205	1.948337	11.634390	25.044233	2.6320	8.244357
011	0.036338	1.140663	3.894524	11.333592	32.552825	2.5440	6.343160
100	0.034778	0.574514	2.973997	11.790378	37.967769	2.6240	4.188545
101	0.043541	0.595287	2.941598	11.839260	37.108482	2.6560	4.225001
110	0.026937	0.574598	2.974362	11.683766	21.839633	1.9280	8.411214
111	0.043755	1.175824	2.936977	11.512556	36.151851	1.9680	4.217907

Tabla 5.2: Parámetros de rendimiento del sistema.

Prueba	Medidas (%)		
	AUC	EER	Exactitud
M1	98	7.51	92.5
M2	98.09	8.15	91.8
M3	98.32	7.14	92.9
M4	97.96	7.63	92.3
M5	98.21	7.35	92.6
Promedio	98.116	7.5	92.42

pero tomando diferente conjunto de muestras para la creación de la base de datos. Dichos conjuntos estarán etiquetados como M1, M2, M3, M4 y M5. Los resultados se muestran en la Tabla 5.2

En las Fig. 5.8, 5.9, 5.10, 5.11, 5.12 y 5.13 se muestran las curvas ROC y los gráficos de error de las tres primeras pruebas M1, M2 y M3 realizadas. No se añadieron las restantes debido a que el comportamiento en todas las pruebas fue muy similar.

Después de medir el rendimiento del sistema con la curva ROC se estableció un umbral óptimo con el cual se validará al usuario en cuestión. Dicho umbral se obtiene de la misma curva ROC. Se realizaron 20 pruebas por sujeto con la configuración de CROPUF 101 eligiendo muestras diferentes a las utilizadas durante la etapa de registro y generación de la base de datos. En la Fig. 5.14 se muestra la relación entre los usuarios y la cantidad de veces que el sistema logró verificarlos.

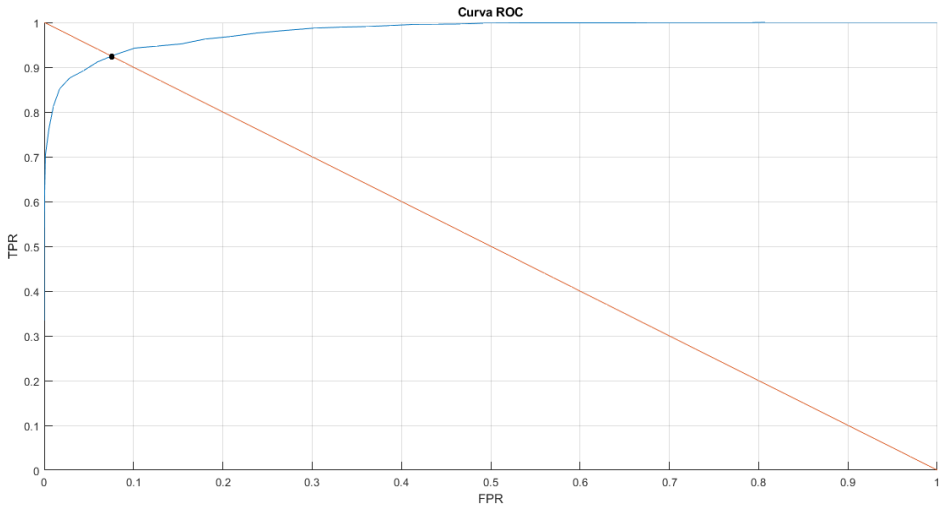

Figura 5.8: Curva ROC de la prueba M1.

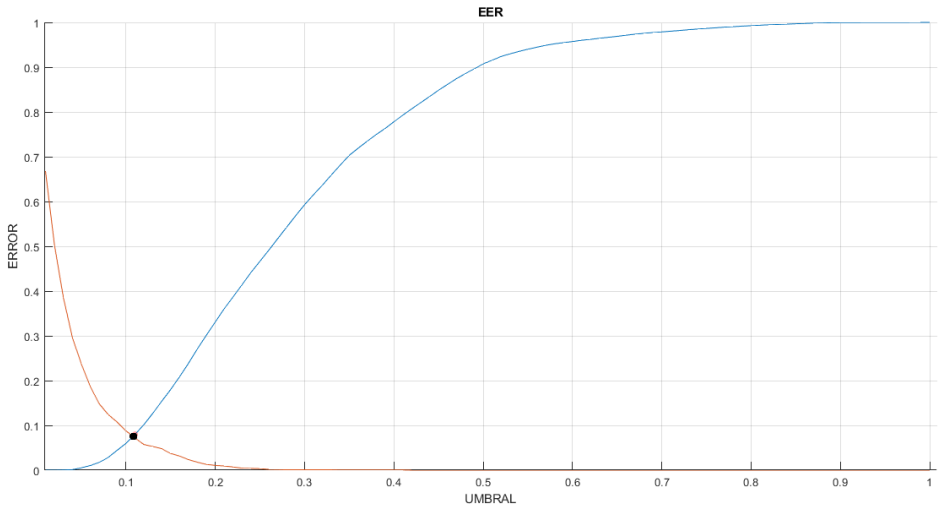

Figura 5.9: Intersección de la tasa de falsos rechazos y la tasa de falsas aceptancias para la prueba M1.

Adicionalmente, se realizó la implementación de otros modelos de PUF como el Arbiter PUF y el Anderson PUF. Sin embargo, el Arbiter PUF presenta muchas desventajas frente al CROPUF y no se obtuvo una respuesta positiva del Arbiter PUF por lo que ambos casos fueron desechados.

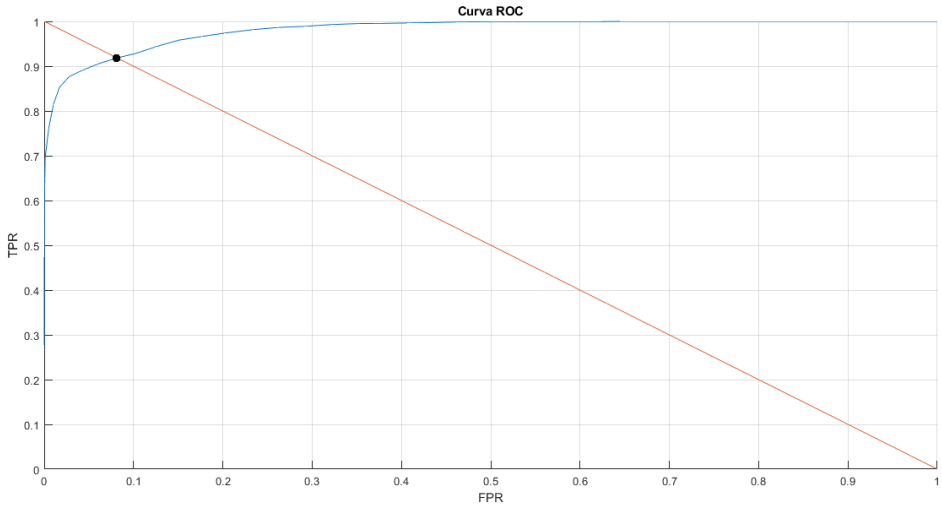

Figura 5.10: Curva ROC de la prueba M2.

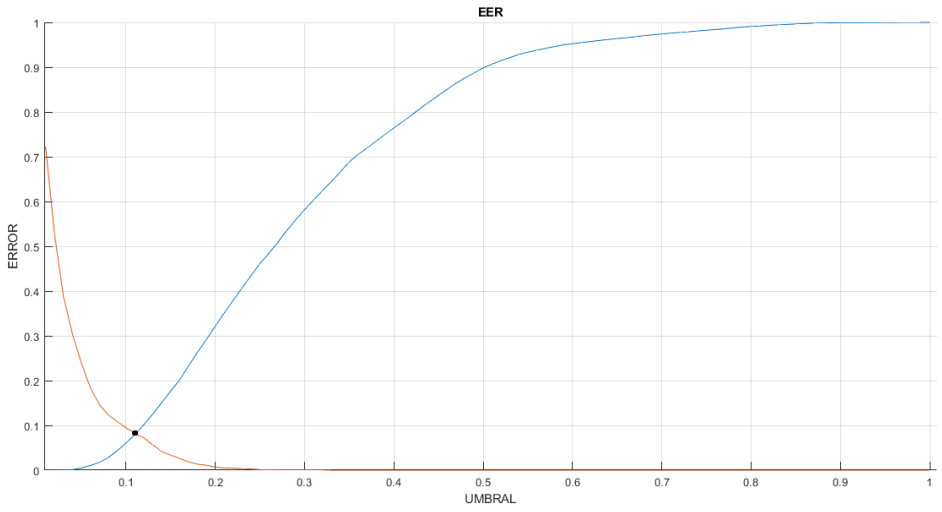

Figura 5.11: Intersección de la tasa de falsos rechazos y la tasa de falsas aceptancias para la prueba M2.

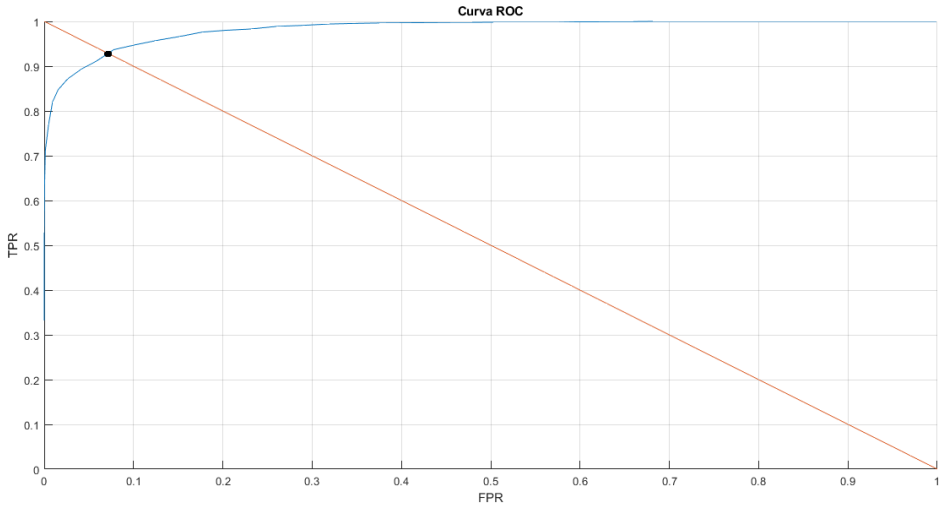

Figura 5.12: Curva ROC de la prueba M3.

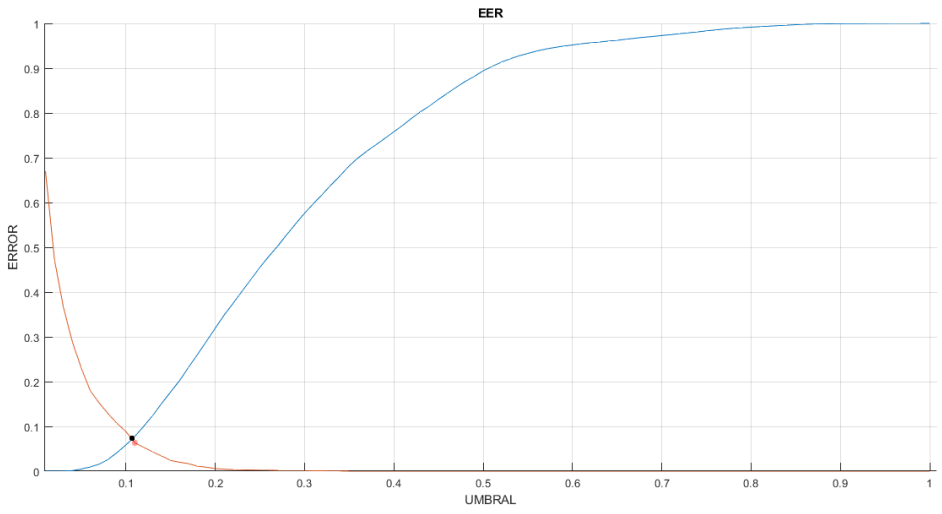

Figura 5.13: Intersección de la tasa de falsos rechazos y la tasa de falsas aceptancias para la prueba M3.

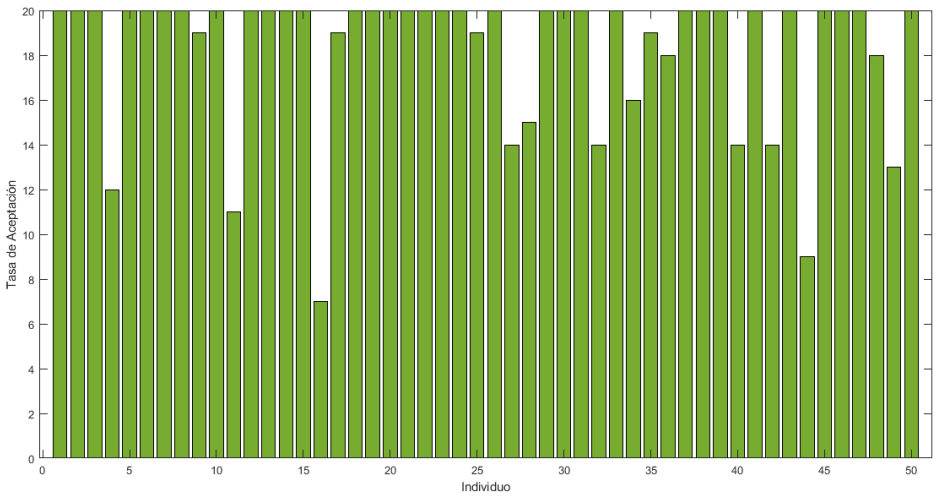

Figura 5.14: Relación entre los usuarios y la aceptación del sistema.

Capítulo 6

Conclusiones

El uso de funciones físicas inclonables es una alternativa para extraer llaves criptográficas y aplicarlas en la transformación de otras variables. Su principal ventaja, como se ha mencionado anteriormente, es el bajo costo de implementación en hardware debido a que se aprovechan los recursos del dispositivo y que dichas respuestas no se almacenan.

Generalmente se suele extraer una respuesta o llave por dispositivo, sin embargo, en este trabajo se propusó extraer varias respuestas de uno sólo obteniendo buenos resultados. Se comprobó también que dichas respuestas son únicas y características del dispositivo en las que se implementen y que las variaciones intraclase son mínimas respecto a las variaciones interclase con otros dispositivos.

Se demostró que el uso de características estadísticas en un sistema biométrico cancelable también da buenos resultados, ya que la mayoría de los trabajos hace uso solamente de las características temporales de una señal ECG.
Respecto a la eficiencia del rendimiento biométrico se comprobó que la modificación de plantillas a través de cadenas binarias obtenidas por diferentes medios es una técnica sencilla de cancelación de plantillas que arroja buenos resultados y que requiere menor costo computacional comparado con técnicas que implementan redes neuronales o que requieren de muchas iteraciones durante la generación de dichas cadenas.

Los resultados obtenidos en este trabajo estan por encima de varias técnicas propuestas en la literatura.

Capítulo 7

Trabajo a futuro

Este trabajo puede ser la base para introducir el uso de diferentes funciones físicamente inclonables como técnica de cancelación de plantillas en el área de la biométria cancelable, ya que existen muchas configuraciones disponibles y con estas se puede mejorar el sistema disponible para obtener aún mejores resultados. Un punto importante es conseguir implementar un sistema en el que se obtenga acceso a varios dispositivos de uso personal como celulares o computadoras donde se puedan implementar PUF para obtener un mejor control de acceso en el sistema.

Esto podría verse aplicado también a la seguridad de desbloqueo en dispositivos moviles, ya que muchos de los celulares o tabletas hacen uso de rasgos biométricos como la huella dactilar o el reconocimiento facial. Estos pueden utilizarse y se les puede aplicar una transformación con la respuesta de alguna PUF implementada previamente en el móvil para almacenar una plantilla de características cancelada y no la plantilla original previniendo así el robo de información personal importante como son los rasgos biométricos y a la vez evitando el desbloque de dicho dispositivo.

Un cambio importante que se puede realizar es la manera en la que se relacionan o combinan las plantillas cancelables con las plantillas de características originales. En este trabajo se transformaron a través del producto entre matrices y posteriormente con la suma de las columnas, sin embargo, existen métodos más complejos que pueden mejorar los resultados obtenidos previamente.

Otra modificación que se puede realizar es cambiar la técnica de clasificación debido a que se suele utilizar la distancia de Hamming como clasificador durante el uso de las PUF. Para esto sería necesario obtener una plantilla de cancelación binaria que puede conseguirse a través de umbralización como se ha realizado en otros trabajos.

Por último, la implementación total del sistema puede realizarse en un FPGA para conseguir un funcionamiento en tiempo real o se puede hacer uso de otros rasgos biometricos.

Bibliografía

[1] Vishal M Patel, Nalini K Ratha, and Rama Chellappa. Cancelable biometrics: A review. *IEEE Signal Processing Magazine*, 32(5):54–65, 2015.

[2] Mohamed Hammad, Gongning Luo, and Kuanquan Wang. Cancelable biometric authentication system based on ecg. *Multimedia Tools and Applications*, 78(2):1857–1887, 2019.

[3] Nilanjan Dey, Bijurika Nandi, Monalisa Dey, Debalina Biswas, Achintya Das, and Sheli Sinha Chaudhuri. Biohash code generation from electrocardiogram features. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 732–735. IEEE, 2013.

[4] Fagul Pandey, Priyabrata Dash, and Divyanshi Sinha. A random walk-based cancelable biometric template generation. In *Innovations in Computational Intelligence and Computer Vision*, pages 423–429. Springer.

[5] Tanuja Sudhakar and Marina Gavrilova. Cancelable biometrics using deep learning as a cloud service. *IEEE Access*, 8:112932–112943, 2020.

[6] Keshav Gupta, Gurjit Singh Walia, and Kapil Sharma. Novel approach for multimodal feature fusion to generate cancelable biometric. *The Visual Computer*, pages 1–13, 2020.

[7] Hanvit Kim and Se Young Chun. Cancelable ecg biometrics using compressive sensing-generalized likelihood ratio test. *IEEE Access*, 7:9232–9242, 2019.

[8] Amir E El-Refaey, Marwa A Shouman, Ezz El-din Hemdan, Adel EL-Fishawy, Abd El-Samie, et al. Triple c: A new algorithm for ecg cancelable biometric

system. *Menoufia Journal of Electronic Engineering Research*, 28(ICEEM2019-Special Issue):43–50, 2019.

[9]

[10] Hanvit Kim, Minh Phuong Nguyen, and Se Young Chun. Cancelable ecg biometrics using glrt and performance improvement using guided filter with irreversible guide signal. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 454–457. IEEE, 2017.

[11] Ronald Salloum and C-C Jay Kuo. Ecg-based biometrics using recurrent neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2062–2066. IEEE, 2017.

[12] Sarineh Keshishzadeh and Saeid Rashidi. A system of biometric authentication based on ecg signal segmentation. In *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, pages 1873–1877. IEEE, 2014.

[13] César Tolosa Borja and Álvaro Giz Bueno. Sistemas biométricos. *Disponible en internet: http://www. dsi. uclm. es/asignaturas/42635/web_BIO/Documentacion/Trabajo s/Biometria/Trabajo % 20Biometria. pdf*, 2010.

[14] Marcos Faúndez Zanuy. Experimentos prácticos sobre la vulnerabilidad de sistemas biométricos. Año Desconocido.

[15] Sergio Sánchez Martín. Estudio del rendimiento biométrico de sistemas de huella dactilar: análisis de diferentes sensores y algoritmos. B.S. thesis, 2015.

[16] University of Nebraska Medical Center. The area under an roc curve. http://gim.unmc.edu/dxtests/roc3.htm, Desconocido.

[17] OSCAR EDUARDO VERA, EDISON DUQUE CARDONA, and JORGE RIVERA PIEDRAHITA. Extracción de características de la señal electrocardiográfica mediante software de análisis matemático. *Scientia et technica*, 12(31):59–64, 2006.

[18] Luis Azcona. El electrocardiograma. *López Farré A, Macaya Miguel C, directores. Libro de la salud cardiovascular del Hospital Clínico San Carlos y la fundación BBVA. 1ª ed. Bilbao: Fundación BBVA*, pages 49–56, 2009.

[19] E. J Berbari. *Biomedical Engineering Handbook*, volume 2. Bronzino, Joseph D. Boca Raton, 2000.

[20] Uwe Meyer-Baese and U Meyer-Baese. *Digital signal processing with field programmable gate arrays*, volume 65. Springer, 2007.

[21] Marcelino Martínez Sober, Juan Gómez Sanchis, Luis Gómez Chova, Antonio J Serrano López, and Joan Vila Francés. Filtros digitales (2009/2010). 2009.

[22] Jesús Rubén Azor Montoya. La transformada wavelet. *Revista de la Universidad de Mendoza*, 2001.

[23] Dr Mistry. Discrete wavelet transform using matlab. *International journal of Computer Engineering Technology (IJCET)*, 4:252–259, 01 2013.

[24] Denisse Escarlette Mancilla Palestina. Análisis de características estadísticamente significativas en el dominio temporal de señales ecg y ppg para identificación biométrica bimodal. Master's thesis, 2019.

[25] Alexis Martín Cruz. Clasificación de latidos según estándar aami mediante red neuronal sobre plataforma intel edison. B.S. thesis, 2015.

[26] Bo Hjorth. Eeg analysis based on time domain properties. *Electroencephalography and clinical neurophysiology*, 29(3):306–310, 1970.

[27] Marcos Raphael Benítez Aldás et al. Estudio y análisis de métodos para la extracción de características y clasificación de emociones basados en eeg. Master's thesis, 2018.

[28] Universidad Autonoma de Madrid. Dtw. alineamiento temporal dinámico. 2017.

[29] Jucheng Yang and Shan Juan Xie. *New trends and developments in biometrics*. BoD–Books on Demand, 2012.

[30] MAES Roel. Physically unclonable functions: Constructions, properties and applications. *Katholieke Universiteit Leuven, Belgium*, 2012.

[31] Roel Maes and Ingrid Verbauwhede. *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. Towards Hardware-Intrinsic Security, 2010.

[32] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14. IEEE, 2007.

[33] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In *International workshop on cryptographic hardware and embedded systems*, pages 63–80. Springer, 2007.

[34] Blaise Gassend, Daihyun Lim, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Identification and authentication of integrated circuits. *Concurrency and Computation: Practice and Experience*, 16(11):1077–1098, 2004.

[35] Jae W Lee, Daihyun Lim, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, pages 176–179. IEEE, 2004.

[36] Blaise Laurent Patrick Gassend. *Physical random functions*. PhD thesis, Massachusetts Institute of Technology, 2003.

[37] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 148–160, 2002.

[38] Giray Kömürcü, Ali Emre Pusane, and Günhan Dündar. A ring oscillator based puf implementation on fpga. *IU Journal of Electrical and Electronics Engineering*, 13(2):1647–1652, 2013.

[39] Jason H Anderson. A puf design for secure fpga-based embedded systems. In *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 1–6. IEEE, 2010.

[40] Abhranil Maiti and Patrick Schaumont. Improving the quality of a physical unclonable function using configurable ring oscillators. In *2009 International Conference on Field Programmable Logic and Applications*, pages 703–707. IEEE, 2009.

[41] Mohammad A Usmani, Shahrzad Keshavarz, Eric Matthews, Lesley Shannon, Russel Tessier, and Daniel E Holcomb. Efficient puf-based key generation in fpgas using per-device configuration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(2):364–375, 2018.

[42] Alejandra Miguel Vargas Mandujano. *Metodología para la implementación de métodos numéricos en hardware*. PhD thesis, 2013.

[43] TC Esteban, M Rangel, and L de la Fraga. Engineering applications of fpgas: chaotic systems, artificial neural networks, random number generators, and secure communication systems. *Switzerland: Springer*, 2016.

[44] I Xilinx. series fpgas configurable logic block: user guide. *San Jose, CA: Xilinx*, 2014.

[45] Hooman Sedghamiz. Matlab implementation of pan tompkins ecg qrs detector. *Code Available at the File Exchange Site of MathWorks*, 2014.