Universidad Zaragoza
1542

Trabajo Fin de Máster

Desarrollo de un sistema de identificación biométrico utilizando la señal pletismográfica (PPG), en un entorno de cloud computing

Development of a biometric identification system using the photoplethysmogram (PPG) in a Cloud Computing environment

Autor

Jorge Sancho Larraz

Director

Álvaro Alesanco Iglesias

Escuela de Ingeniería y Arquitectura

2016

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Jorge Sancho Larraz,

con nº de DNI 73087078R en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster, (Título del Trabajo) Desarrollo de un sistema de identificación biométrico utilizando la señal pletismográfica (PPG), en un entorno de cloud computing.,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 24 de junio de 2016

Fdo: Jorge Sancho Larraz

Agradecimientos

En primer lugar quería dar las gracias a ***Álvaro****, ya que sin él este proyecto no habría sido posible, y sobretodo agradecerle la confianza que ha depositado en mí en todo momento.*

A mis padres y mi hermana, quienes realmente han hecho posible que haya llegado hasta aquí. Gracias por estar siempre a mi lado, animándome en los malos momentos y aguantándome en aquellos de mayor estrés.

A todos los compañeros de la universidad por hacer que todas esas largas horas en el aula y en los laboratorios, así como las interminables semanas durante el periodo de exámenes en la sala de estudios hayan transcurrido de la forma más amena posible.

A mis amigos de Tauste, por el interés y apoyo que me habéis mostrado en todo momento y por todos esos momentos que hemos compartido a lo largo de los años.

A todos los miembros de la banda, por soportar mis repetidas ausencias a los ensayos en los periodos de exámenes.

A la charanga, por todos los buenos momentos que compartimos tanto mientras tocamos como estando de fiesta.

En definitiva, a todas las personas importantes para mí, simplemente gracias.

Desarrollo de un sistema de identificación biométrico utilizando la señal pletismográfica (PPG), en un entorno de cloud computing.

RESUMEN

Actualmente se está produciendo un cambio en el paradigma de la prestación de los servicios sanitarios en todo el mundo, pero es imprescindible realizar esta transición garantizando la seguridad de la información transmitida y almacenada en los archivos de señales biomédicas.

Durante este proyecto se ha analizado la capacidad identificativa del PPG en entornos con un número reducido de usuarios. Para ello se han evaluado las prestaciones obtenidas aplicando varios esquemas de procesado, utilizando MATLAB.

A continuación se ha diseñado e implementado una arquitectura de almacenamiento y procesado basada en la computación en la nube, que permita realizar la identificación de usuarios en tiempo real aprovechando las técnicas que mejores prestaciones presentaron en el análisis anterior. Para ello, se ha implementado un servidor FHIR para el almacenamiento de las muestras del PPG de los diferentes usuarios y un servicio capaz de realizar la generación de los modelos de los usuarios y el proceso de identificación de forma eficiente utilizando una librería propia escrita en C.

Para utilizar los servicios provistos por la arquitectura se han definido las interacciones que deberán seguirse y se han implementado dos clientes, uno para PC, utilizando Python y otro para terminales móviles, utilizando HTML5, CSS3 y JS, capaces de recoger los datos procedentes de los pulsioxímetros y comunicarse con los elementos de la arquitectura.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1 Ciberseguridad e identificación en entornos de telemonitorización domiciliaria.

Actualmente se está produciendo un cambio en el paradigma de la prestación de los servicios sanitarios en todo el mundo, impulsado por los rápidos avances en el ámbito de las Tecnologías de la Información y Comunicaciones (TIC) y más concretamente gracias al desarrollo de aplicaciones y dispositivos móviles y su utilización en el campo de la salud. Dentro de los diversos tipos de servicios que se han puesto en marcha en los últimos años (e.g. recordatorios de citas, historia clínica, promoción de la salud, acceso y búsqueda de información, etc.) se espera que el seguimiento de pacientes con enfermedades crónicas sea un área de gran crecimiento en los próximos años. Sin embargo, aspectos como la seguridad y privacidad de la información del paciente, la usabilidad del sistema y la interoperabilidad y estandarización, constituyen retos y barreras a la implantación y adopción a mayor escala de este tipo de servicios.

Teniendo en cuenta todo esto, puede resultar interesante el diseño de un sistema que aproveche las señales monitorizadas durante el seguimiento de enfermedades crónicas para realizar una identificación transparente del usuario que permita una mayor comodidad y facilidad de uso de los sistemas en entornos médicos garantizando la seguridad de la información transmitida y almacenada en los

archivos de señales biomédicas, lo que resulta de vital importancia desde el punto de vista ético y legal.

1.2 Objetivos

El objetivo principal de este proyecto es el desarrollo de un sistema de identificación biométrico utilizando la señal pletismográfica (PPG) que permita realizar una identificación transparente de la persona en entornos con un número reducido de usuarios. Se plantean además una serie de objetivos específicos para el correcto funcionamiento del sistema:

- Análisis de diferentes esquemas de procesado y evaluación de sus prestaciones en términos de identificaciones correctas y erróneas para diferente número de usuarios.
- Implementación de los algoritmos utilizados para llevar a cabo la identificación utilizando un lenguaje de programación de bajo nivel, de forma que se maximice la eficiencia del sistema y se minimice el tiempo necesario para realizar la identificación.
- Diseño y desarrollo de una arquitectura basada en microservicios que permita trasladar la complejidad de cálculo a la nube.
- Utilización de estándares de comunicación médicos, prestando especial atención a HL7 FHIR , para soportar todos los intercambios de información que incluyan datos médicos, como por ejemplo las muestras del PPG.

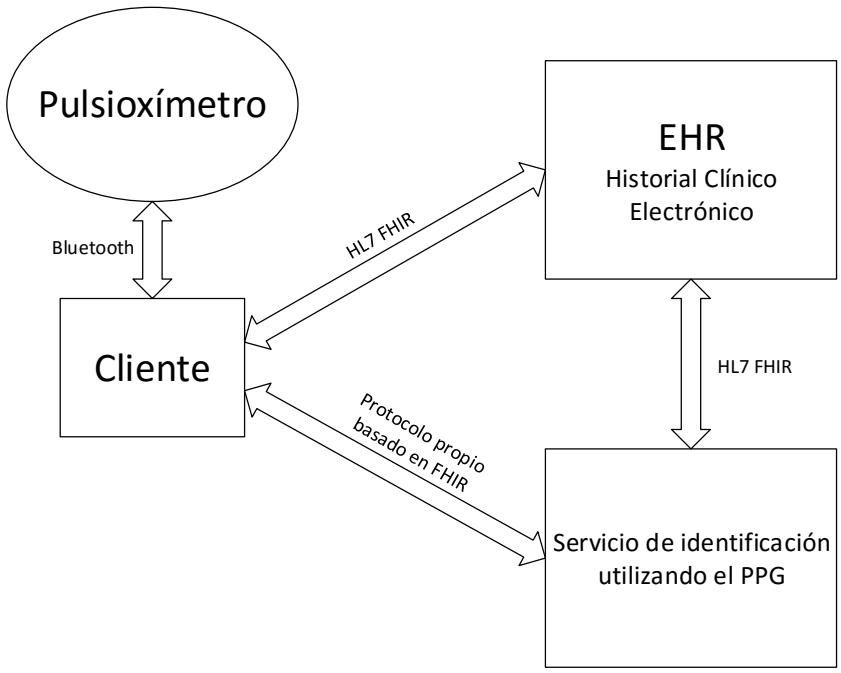


Figura 1.1: Relación entre los diferentes elementos del sistema.

1.3 Materiales y herramientas utilizadas

Para la realización de este proyecto se han utilizado los siguientes recursos hardware y software.

Hardware:

- **Ordenador:** Ordenador personal con conectividad Bluetooth para realizar la adquisición de la base de datos de señales pletismográficas y desarrollar los algoritmos necesarios para realizar la identificación.
- **Pulsioximetro Berry:** Sistema de adquisición de la señal pletismográfica de bajo coste con conexión Bluetooth. La señal obtenida con este equipo se utilizará para realizar la identificación biométrica.
- **Pulsioximetro Nonin Wrist Ox2:** Sistema de adquisición de la señal pletismográfica semi-profesional con conexión Bluetooth. La señal obtenida con este equipo se utilizará para realizar la identificación biométrica.
- **Terminal móvil:** Dispositivo móvil con sistema operativo Android y

conectividad Bluetooth y Wifi.

Software:

- **Debian:** Sistema operativo sobre el que se ha realizado el desarrollo. Es una distribución de Linux gratuita, fácil de instalar y cuenta con una ingente cantidad de software. Muestra una de las mejores relaciones entre funcionalidad y recursos empleados.
- **Matlab:** Lenguaje de desarrollo rápido utilizado para realizar la comparativa de prestaciones entre los diferentes algoritmos desarrollados para realizar la identificación.
- **C:** Lenguaje de programación de bajo nivel y alta eficiencia utilizado para la realizar la versión definitiva del algoritmo.
  - **GSL (GNU Scientific Library):** Librería matemática gratuita escrita en c.
  - **gcc:** conjunto de compiladores c creados por el proyecto GNU. Estos compiladores se consideran estándar para los sistemas operativos derivados de UNIX.
- **Python:** Lenguaje de programación interpretado, multiparadigma y de código abierto. Se ha utilizado el entorno de desarrollo por defecto, IDLE, por venir ya instalado con Python y por contar con una interfaz muy sencilla de utilizar.
  - **Hashlib:** Librería para el cálculo de hashes.
  - **PyBluez:** Librería para la conexión Bluetooth desde Python.
  - **WxPython:** Librería para la realización de interfaces graficas en Python.
  - **Matplotlib:** Librería para la generación de gráficas de datos.
  - **BaseHTTPSerber:** Librería que implementa un servidor HTTP.

  - **Requests:** librería para la generación de peticiones HTTP
  - **JSON:** Librería para codificar y decodificar objetos JSON
  - **pyMongo:** librería para realizar las interacciones necesarias con una base de datos MongoDB desde Python

- **MongoDB:** Base de datos NoSQL de código libre.
- **VMWare:** entorno de virtualización de uso profesional.
- **Android:** Sistema operativo basado en el núcleo Linux diseñado principalmente para dispositivos móviles con pantalla táctil.
- **Apache Cordova:** Marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo cada plataforma móvil.
  - **HTML 5:** Última versión del estándar HTML, el cual es un lenguaje markup usado para estructurar y presentar el contenido para la web.
  - **CSS 3:** Es la información que define como va a ser la presentación de una web. Entendiendo por presentación los colores, efectos, tipos de letra, etc.
  - **JavaScript:** Lenguaje de programación interpretado, de lado del cliente que permite añadir dinamismo a las páginas web.

1.4 Organización de la memoria

La memoria está estructurada de la siguiente manera:

- **Capítulo 1: Introducción.** Es el capítulo actual y contiene una breve descripción del trabajo realizado, así como sus principales objetivos.
- **Capítulo 2: Estado del arte.** En este capítulo se describen tanto las principales soluciones existentes actualmente para autenticar a un usuario como las principales características del estándar FHIR.

- **Capítulo 3: Sistema de identificación biométrico.** En este capítulo se presentan las diferentes soluciones analizadas para realizar la identificación del usuario, así como los resultados obtenidos.
- **Capítulo 4: Arquitectura del sistema.** En este capítulo se presenta la arquitectura desarrollada y los diferentes elementos que se han implementado.
- **Capítulo 5: Conclusiones y líneas futuras.** Este es el último capítulo de la memoria y contiene las conclusiones que se han sacado en este proyecto y las posibles líneas futuras que se podrían seguir.

También se han añadido los siguientes anexos:

- **Anexo A. Acrónimos.**
- **Anexo B. Características del PPG.**
- **Anexo C. Recursos FHIR utilizados: Patient y Observation.**
- **Anexo D. Interfaz RESTful FHIR.**
- **Anexo E. Algoritmo genético de alineamiento.**
- **Anexo F. Fundamentos matemáticos: KLT y LDA.**
- **Anexo G. Métodos desechados para la generación de modelos y la identificación.**
- **Anexo H. Documentación de la librería PPG_lib.**
- **Anexo I. Documentación de la API pyPPG.**

Capítulo 2

Estado del arte

2.1 Biometría

Las técnicas de identificación o autenticación son la primera línea de fuego para proteger un sistema informático contra las actividades de usuarios sin autorización. Conocer la identidad de un sujeto es imprescindible para realizar una correcta autorización y contabilidad. Estos tres conceptos juntos se denominan AAA (Authentication, Authorization and Accountability).

Tradicionalmente se ha considerado que existen tres formas de autenticar a un usuario, por algo que sabe (e.g. contraseñas, PINs, patrones gráficos o preguntas), por algo que tiene (e.g., Smart cards, certificados, tarjetas SIM o teléfono móvil, hardware o software tokens), y por algo que es (biometría).

La biometría utiliza características medibles y distintivas de un individuo para realizar una autenticación robusta. Estas características pueden ser fisiológicas y comportamentales. Las características fisiológicas son aspectos únicos del cuerpo tales como la huella dactilar, la forma de la cara o el iris. Por su parte, las características comportamentales son aquellas adquiridas a lo largo de la vida como la voz, el patrón de pulsaciones en un teclado o la firma.

En los últimos años, la biometría está complementando y/o sustituyendo al resto de métodos de identificación en muchos sistemas informáticos, en especial en aquellos de alta seguridad, debido a que estas características no pueden olvidarse,

como las contraseñas, no se pueden prestar o ser robadas fácilmente, como los tokens, y son difíciles de reproducir, cambiar o esconder, ofreciendo de este modo, una buena defensa frente al repudio.

Todo sistema biométrico puede funcionar de dos modos diferentes, verificación e identificación. En el proceso de verificación los rasgos biométricos se comparan solamente con los de un único modelo. Este proceso implica conocer la identidad del individuo a autentificar, por lo tanto, dicho individuo ha de presentar algún tipo de credencial, que después del proceso de autentificación biométrica será validada o no. En el proceso de identificación no es necesario conocer la identidad del sujeto previamente. La nueva muestra de datos biométricos es tomada del usuario y comparada una a una con todos los patrones existentes en la base de datos. El resultado de este proceso es la identidad del individuo.

Mientras que el resultado de una autenticación con contraseña o token es booleano, es decir, la contraseña puede ser correcta o no, en biometría el resultado es una medida de confianza, y en base a umbral de decisión se determina si la confianza es suficiente para aceptar o rechazar el acceso. Consecuentemente, existe un margen para el cual la decisión será errónea, tanto dando acceso a un intruso como denegándoselo a un usuario legítimo.

Típicamente, el rendimiento de un sistema de autenticación biométrico se mide en función de varias tasas de error, como la tasa de falsa aceptación (FAR), tasa de falso rechazo (FRR) y la tasa de igual error (EER). Modificando el umbral de decisión podremos reducir la FAR a costa de aumentar la FRR y viceversa. Así pues, en un sistema de alta seguridad será preferible que el número de intrusos que consigan acceso sea prácticamente nulo, aun a costa de que los usuarios autorizados no consigan acceder al sistema en ciertas ocasiones, mientras que en el desbloqueo de un terminal móvil será preferible que el dueño sea capaz de desbloquearlo a la primera, aunque la probabilidad de que otra persona pueda desbloquearlo sea relativamente alta, produciéndose un compromiso entre seguridad y usabilidad.

Los estudios más recientes en biometría se centran en el uso de señales médicas, como el electrocardiograma (ECG), el electromiograma (EMG) o el electroencefalograma (EEG) para realizar la autenticación. Estas señales presentan

una ventaja frente al resto de técnicas existentes, y es que aportan de forma intrínseca una prueba de vida del usuario. Por el contrario, estas señales son muy complejas de capturar por la cantidad de sensores que hay que colocar al paciente y el coste económico de los equipos.

En la línea de las señales anteriormente nombradas (ECG, EMG y EEG), existe una señal denominada foto-pletismograma (PPG), la cual es monitorizada normalmente durante el seguimiento de pacientes y durante el desarrollo de actividad física por su sencillez de adquisición, ya que se puede capturar con un único sensor situado en un dedo, con un bajo coste económico.

2.2 PPG

La señal PPG representa el movimiento de la sangre, que sale del corazón hacia la periferia a través de los vasos sanguíneos.

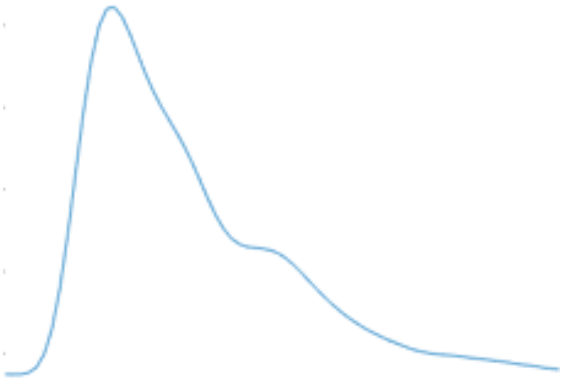


Figura 2.1: Ciclo extraído del "PPG".

La calidad de la señal PPG depende de la localización del sensor, de las propiedades de la piel del sujeto en el momento de la medida, de la estructura de la piel, la saturación de oxígeno en sangre, la presión sanguínea, la temperatura de la piel y el entorno de medida. Estos factores generan varios tipos de errores, también denominados artefactos, que pueden reflejarse en la medida del PPG.

El sensor utilizado para la adquisición del PPG consiste en un emisor de luz infrarroja (típicamente un fotodiodo emitiendo una longitud de onda cercana a

los 900 nm) y un fotodetector. La luz procedente del emisor ilumina el tejido (por ejemplo, la piel), y el fotodetector mide las pequeñas variaciones en la intensidad de luz recibida. Estas pequeñas variaciones en la intensidad, están asociadas a cambios en el volumen de sangre en los vasos sanguíneos. Un incremento en el volumen de sangre, se representa en el sensor como un decremento en la intensidad de luz recibida, y viceversa.

La forma del PPG se divide en dos fases, la fase anacrótica o flanco ascendente del pulso y la fase catacrotica o flanco descendente. La primera fase esta principalmente relacionada con la sístole, mientras que la segunda fase lo está con la diástole y las ondas reflejadas desde la periferia. Un notch dicrótico suele aparecer en la fase catacrotica en sujetos con arterias sanas. Se definen una serie de características básicas de la forma de onda.

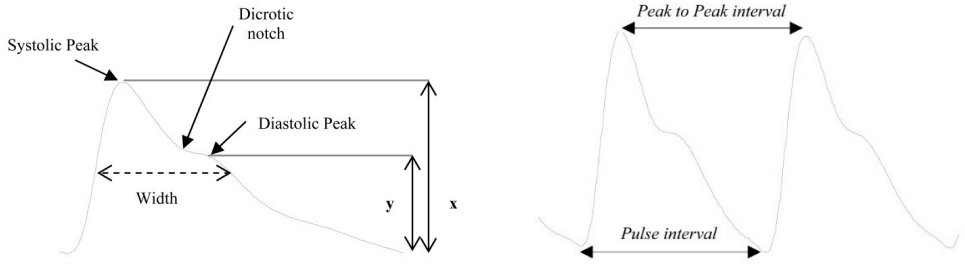


Figura 2.2: Características del "PPG".

- **Amplitud sistólica:** es un indicador de los cambios en el volumen sanguíneo causados por el flujo de sangre en las arterias próximas al sitio de medida. La amplitud sistólica es una buena medida para estimar la presión sanguínea.
- **Anchura del pulso:** anchura del pulso calculada cuando la altura es la mitad del pico sistólico. La anchura del pulso esta corralada con la resistencia vascular sistémica.
- **Área del pulso:** mide el área total bajo la curva del PPG.
- **Intervalo pico a pico:** es la distancia entre dos picos sistólicos consecutivos. Esta medida esta corralada con el intervalo R-R del ECG.

- **Intervalo del pulso:** es la distancia entre el comienzo y el fin de la onda PPG. Este intervalo se utiliza frecuentemente en lugar del intervalo pico a pico cuando los picos diastólicos son más claros y sencillos de detectar que los sistólicos.

El resto de características del PPG y de sus derivadas VPG y APG se encuentran detalladas en el anexo B.

2.3 La arquitectura HL7

HL7 (Health Level Seven) es una organización sin ánimo de lucro fundada en 1987 y acreditada por el ANSI (American National Standards Institute) para el desarrollo de estándares médicos. El trabajo de esta organización está enfocado a proveer de un framework completo y de los estándares necesarios para el intercambio, integración, compartición y recuperación de información clínica electrónica que de soporte a la práctica clínica y a la gestión, provisión y evaluación de servicios sanitarios.

La primera versión, HL7 V1.0 fue publicada solo unos meses después de la fundación de la organización. Esta versión se centraba en el intercambio de información relativa a ingresos, altas y traslados (ADT) dentro del hospital.

En 1988 se publicó la versión HL7 2.0, la cual introdujo principalmente una serie de mensajes para el intercambio de solicitudes e informes relativos a pruebas clínicas y tratamientos. Dos conceptos clave para entender esta versión del estándar son la sintaxis de los mensajes y los tipos de datos. La sintaxis de los mensajes define la estructura general y como se organizan las diferentes partes de los mismos. Cada mensaje está compuesto de varios segmentos, cada uno de los cuales contiene una serie de campos, que a su vez están definidos por un tipo de dato. Los tipos de dato son los bloques con los que se construyen los campos, y pueden ser simples, con un único valor, o complejos, con varios componentes. Estos componentes también están definidos por un tipo de dato, que de nuevo podrá ser simple o complejo, generando subcomponentes.

Este estándar ha estado en desarrollo durante más de 25 años. Actualmente, la versión más reciente es la Versión 2.8, la cual fue publicada en 2014. El 95 % de las organizaciones de la salud americanas utilizan HL7 V2.x y más de 35 países cuentan con implementaciones de dicho estándar.

A pesar de la gran adopción de este estándar, hay que tener en cuenta que su desarrollo se realizó sin una buena planificación, solucionando problemas específicos. Por ejemplo, cuando se necesitaba un elemento adicional, este se añadía en la siguiente actualización, por eso la Versión 2 proporciona varias formas de hacer la misma cosa. Todo esto permite una gran flexibilidad, pero al mismo tiempo hace imposible especificar un procedimiento bien definido para comprobar la conformidad de una implementación con el estándar y obliga a los implementadores a invertir una gran cantidad de tiempo en analizar y planear las interfaces, para asegurar que sus equipos soportan las mismas opciones que el equipo situado en el otro extremo de la comunicación.

El trabajo sobre HL7 Versión 3 comenzó en 1992, y se centró en solucionar estos problemas utilizando una metodología bien definida basada en un modelo de información de referencia (RIM). Utilizando unas técnicas rigurosas de análisis y construcción de mensajes, y añadiendo nuevos eventos y formatos de mensaje, con muy poca opcionalidad, el principal objetivo de HL7 Versión 3 es ofrecer un estándar testeable y que permita a los fabricantes certificar la conformidad de sus productos con el estándar.

El RIM es una parte esencial en la metodología de desarrollo de HL7 Versión 3 que especifica la gramática de los mensajes de la Versión 3, proveyendo de una representación explicita de las conexiones semánticas y léxicas que existen entre la información transportada en los campos de los mensajes.

La aplicación más ampliamente utilizada de HL7 Versión 3 es “Clinical Document Architectura” (CDA), la cual comenzó a desarrollarse en 1997 y ofrece un formato XML estándar para la representación de documentos clínicos. CDA define una estructura de tres niveles de documentos. El nivel 1 que incluye una cabecera con los metadatos básicos y un cuerpo con texto o imágenes. El nivel 2 incluye la misma cabecera que el nivel 1 y el cuerpo puede ser un elemento sin

estructura o un número cualquiera de secciones, las cuales pueden estar anidadas, conteniendo cada una de estas secciones un bloque de texto, lo que permite generar formularios. El nivel 3 permite que, en cada una de las secciones anteriores, además del bloque de texto se incluya una estructura de datos.

"Fast Healthcare Interoperability Resources" (FHIR), es un marco de trabajo de nueva generación creado por HL7 que combina las mejores características de HL7 V2, HL7 V3 y CDA, utilizando los últimos estándares WEB (XML, JSON, HTTPs, OAuth, etc.), centrándose en permitir una implementación rápida y sencilla.

FHIR es apropiado para su uso en aplicaciones móviles, comunicaciones en la nube, intercambio de datos basados en el historial clínico electrónico (EHR) y comunicación entre servidores de instituciones proveedoras de servicios sanitarios entre otros.

Las soluciones basadas en FHIR están construidas a partir de un conjunto de componentes modulares llamados "Recursos". Estos recursos son modelos de información que podrían asemejarse a los documentos en CDA. Ha de prestarse especial atención a los recursos "Patient" y "Observation" cuya estructura, representación UML y serialización XML y JSON se muestran con más detalle en el anexo C.

Para el intercambio de dichos recursos entre sistemas, FHIR define una interfaz basada en REST que permite realizar las siguientes operaciones:

- Interacciones a nivel de instancia de un recurso:
  - **Read:** solicita una copia del estado actual de un recurso.
  - **vRead:** solicita una copia de una versión específica de un recurso.
  - **Update:** actualiza un recurso existente, o lo crea si no existe, con la id aportada por el usuario.
  - **Delete:** elimina un recurso.
  - **History:** solicita el historial de actualizaciones de un recurso.
- Interacciones a nivel de tipo de recurso:

  - **Create:** crea una nueva instancia de un recurso con la id asignada por el servidor.
  - **Search:** busca en todos los recursos de cierto tipo en base a unos criterios de filtrado.
  - **History:** solicita el historial de actualizaciones para un tipo de recurso

- Interacciones a nivel de sistema:

  - **Conformance:** solicita el certificado de conformidad del sistema.
  - **Batch/transaction:** actualizar, crea o elimina un conjunto de recursos en una única interacción.
  - **History:** obtiene el historial de actualizaciones de todos los recursos.
  - **Search:** busca en todos los recursos del servidor en base a unos criterios de filtrado.

El formato de cada uno los mensajes y posibles respuestas esperadas puede encontrarse en el anexo D.

Capítulo 3

Sistema de identificación biométrico

Para analizar la capacidad identificativa de la señal PPG se han desarrollado varios esquemas de procesado utilizando un lenguaje de desarrollo rápido como MATLAB. Para garantizar la confiabilidad de los resultados todas las pruebas se han realizado sobre las señales capturadas con dos dispositivos diferentes.

3.1 Generación de la base de datos

Para evaluar las prestaciones de los diferentes algoritmos de identificación desarrollados se consideró la opción de utilizar una base de datos ya existente, pero se descartó por no encontrar ninguna con un número suficiente de sujetos y capturada en condiciones reales.

Para garantizar la privacidad de los usuarios será muy importante no almacenar las señales con un identificador que permita asociar fácilmente la señal con su propietario, por lo que las señales capturadas serán almacenadas con un identificador generado aleatoriamente.

Teniendo en cuenta lo anterior, en términos de la Ley Orgánica de Protección de Datos (LOPD) podemos considerar las señales capturadas como «Dato anonimizado o irreversiblemente disociado» tal y como se define en la letra i) de la Ley 14/2007, de 3 de julio, de Investigación biomédica, ya que se considera

que revertir una asignación aleatoria exige un esfuerzo no razonable, entendiendo por tal el empleo de una cantidad de tiempo, gastos y trabajo desproporcionados. En este caso, la base de datos quedaría excluida de la aplicación de la Ley Orgánica 15/1999 y, en consecuencia, no deberá de notificarse el fichero al Registro General de Protección de Datos, ni obtenerse el consentimiento informado ni adoptar las medidas de seguridad previstas en el Real Decreto 1720/2007, de 21 de diciembre.

3.2 Preprocesado

Para poder realizar un correcto procesado de la señal es necesario realizar un acondicionado previo. En primer lugar, será necesario eliminar la componente de continua que contiene la señal, para lo que se realizará una fase de filtrado. A continuación, se segmentará en ciclos y por último se realizar una normalización tanto en amplitud como en el eje temporal.

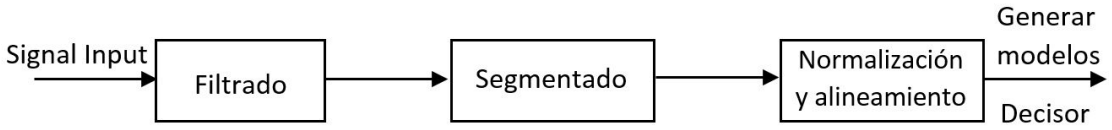
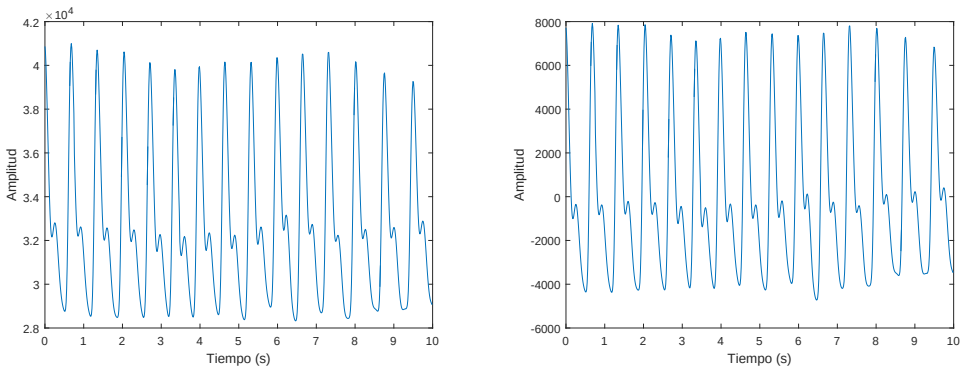


Figura 3.1: Cadena de preprocesado.

3.2.1 Filtrado

La señal capturada presenta una componente de continua, correspondiente al volumen de sangre que se encuentra en los vasos sanguíneos de forma estacionaria. Para eliminar esta línea de base, se realiza un filtrado paso alto con una frecuencia de corte de 0.5 Hz. Se ha utilizado un filtro de tipo Butterworth por producir una respuesta máximamente plana en la banda de paso. Para corregir la distorsión de fase introducida por el filtro se realiza el filtrado en sentido directo (forward) y en sentido inverso (backward) obteniendo así un filtro de fase cero. En este caso se utilizará un filtro IIR de tipo Butterworth de orden 3.



(a) PPG antes del filtrado (b) PPG después del filtrado

Figura 3.2: Señal PPG antes y después del filtrado.

3.2.2 Segmentado

Para segmentar la señal se ha decidido utilizar un algoritmo que marcará como punto de separación entre ciclos aquel mínimo local que preceda a un máximo local cuya diferencia de amplitudes sea mayor que un umbral.

Aplicando únicamente este algoritmo, existe la posibilidad de que no se detecte el punto de separación entre alguno de los ciclos. Para evitar este problema se ha incluido predictor basado en el "Filtro de Kalman" sin considerar la presencia de ruido en el sistema, o lo que sería equivalente, en el "Observador de Luenberger". Estos predictores permiten estimar la posición de un punto de separación entre ciclos en el caso de que no sea detectado por el algoritmo de segmentado explicado anteriormente.

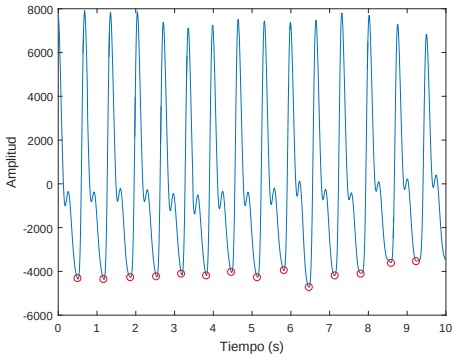
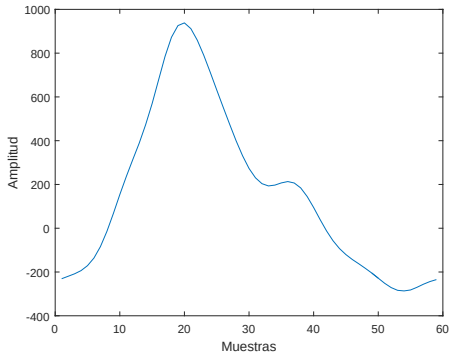


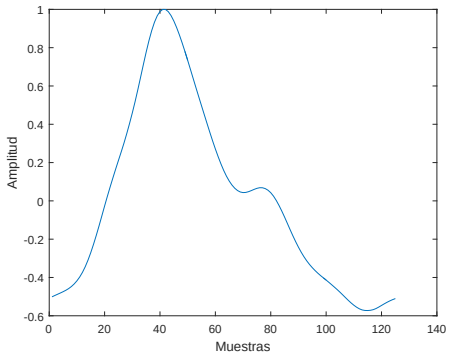
Figura 3.3: Señal PPG segmentada.

3.2.3 Normalización y alineamiento

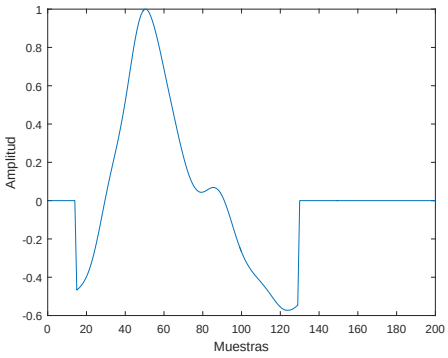
En primer lugar, se implementó un algoritmo de normalización y alineamiento determinista, es decir, la amplitud se escala para que el máximo de cada ciclo sea la unidad y se interpola para que cada uno de ellos esté formados por 128 muestras. La interpolación se ha realizado utilizando splines cúbicos. El alineamiento se ha llevado a cabo situando el máximo del ciclo en la muestra número 50.



(a) Ciclo segmentado



(b) Ciclo normalizado



(c) Ciclo alineado

Figura 3.4: Señal PPG antes y después de la normalización y alineamiento.

A pesar de la alta eficiencia, computacionalmente hablando, de este algoritmo de normalización y alineamiento, se analizó la posibilidad de utilizar otros algoritmos que ofreciesen un mejor emparejamiento entre ciclos, ya que las prestaciones de todas las técnicas de identificación posteriores van a depender en gran medida de esta fase del preprocesado.

Para realizar esta tarea se implementó y probó, también, un algoritmo genético cuyo funcionamiento se detalla en el Anexo E. Este algoritmo no se ha sido utilizado en la solución final debido a que el incremento en el tiempo de procesado que conllevaba era insostenible, incluso para un número reducido de usuarios.

3.3 Generación de modelos e identificación

Todo sistema biométrico pasa por dos fases bien diferenciadas. Una primera, en la que se generan los modelos que representarán a los usuarios del sistema a partir unas señales de entrenamiento, y una segunda, en la que el sistema realizará la verificación/identificación de un individuo en base a sus rasgos biométricos, representados por una señal de test y unos criterios de decisión.

Tras la generación de los modelos, en la fase de verificación/identificación, el sistema podrá seguir dos modos de funcionamiento, el modo de verificación y el modo de identificación. En el modo de verificación, el individuo aporta al sistema una identidad y una señal de test, que representará sus rasgos biométricos. El sistema comparará la señal de test con el modelo correspondiente a la identidad aportada por el individuo, obteniendo una medida de la similitud entre ambos. Finalmente la decisión de si el individuo es quien dice ser se toma en base a si la medida de similitud obtenida supera un umbral o no, siendo dicho umbral un parámetro del sistema que determinará sus prestaciones, tal y como se explica en el apartado de Resultados. En el modo de identificación, el individuo aporta al sistema únicamente una señal de test, la cual se compara con todos los modelos existentes en el sistema, obteniendo una medida de la similitud con cada uno de ellos. El sistema asignará al individuo aquella identidad cuyo modelo obtenga la máxima similitud con la señal de test.

Se han analizado diferentes alternativas para la realización tanto la generación de los modelos como el proceso de verificación/identificación, entre las cuales destacan una en el dominio temporal y otra en el dominio transformado. El resto de métodos analizados se describen en el anexo G.

3.3.1 Dominio temporal

En el dominio temporal las mejores prestaciones se obtienen generando el modelo como la media temporal de todos los ciclos pertenecientes a un usuario tal y como se muestra en la figura 3.5.

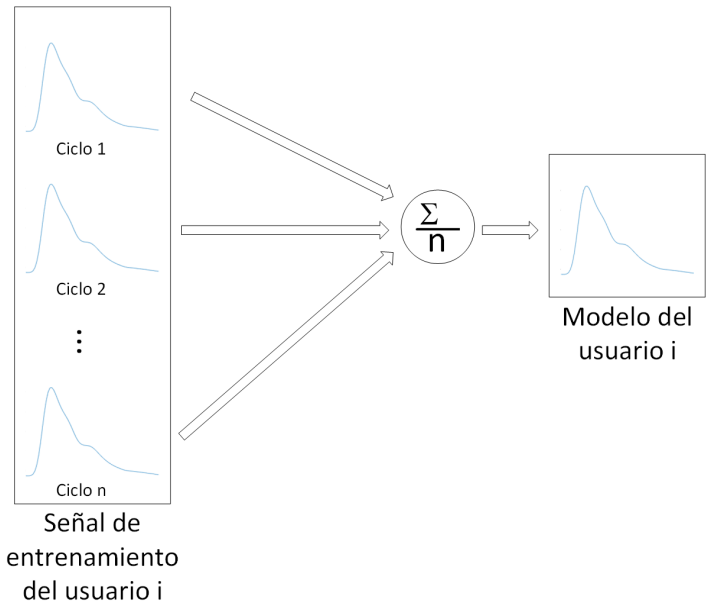


Figura 3.5: Generación del modelo en el dominio temporal.

En la fase de verificación/identificación, la medida de similitud utilizada es la correlación cruzada normalizada, evaluada en cero, entre la media temporal de todos los ciclos de la señal de test y el modelo.

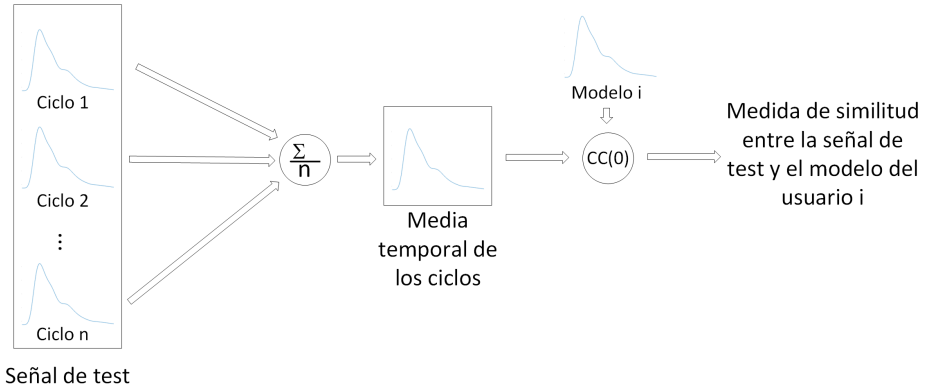


Figura 3.6: Cálculo de la medida de similitud en el dominio temporal.

3.3.2 Dominio transformado

En el domino transformado se han analizados dos variantes, una basada en la transformada de Karhunen-Loève (KLT) y otra en un análisis discriminante lineal (LDA), el cual es una generalización del método discriminante lineal de Fisher, cuyos fundamentos se explican en el anexo F.

Considerando la proyección de un ciclo sobre la base de transformación como un punto de R^n, el modelo de un usuario estará formado por un conjunto de puntos correspondientes a las proyecciones de los ciclos de la señal de entrenamiento sobre la base de transformación.

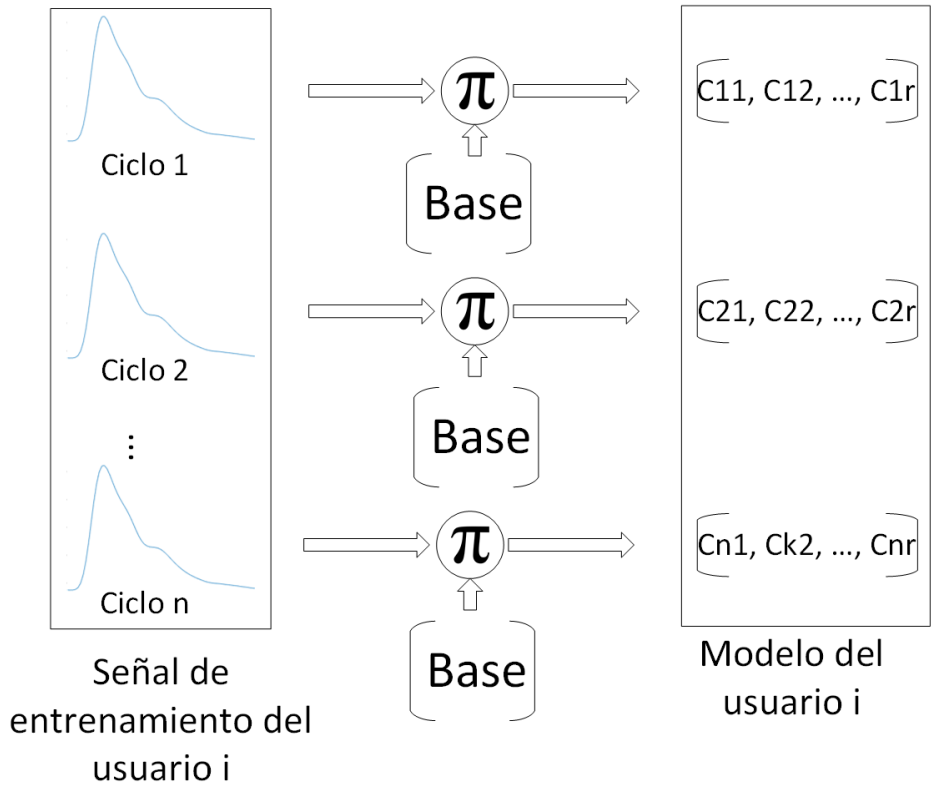


Figura 3.7: Generación del modelo en el dominio transformado.

Denominando a la proyección de los ciclos de la señal de test sobre la base de transformación como "puntos de test", la medida de similitud utilizada en la fase de verificación/identificación se obtiene como la suma de las distancias al vecino más cercano de cada uno de los puntos de test. La distancia al vecino más cercano entre un punto de test y el modelo de un usuario se define como la distancia mínima

entre el punto de test y uno de los puntos contenidos en el modelo del usuario, entendiéndose por distancia una de las normas de R^n, en este caso la norma 1.

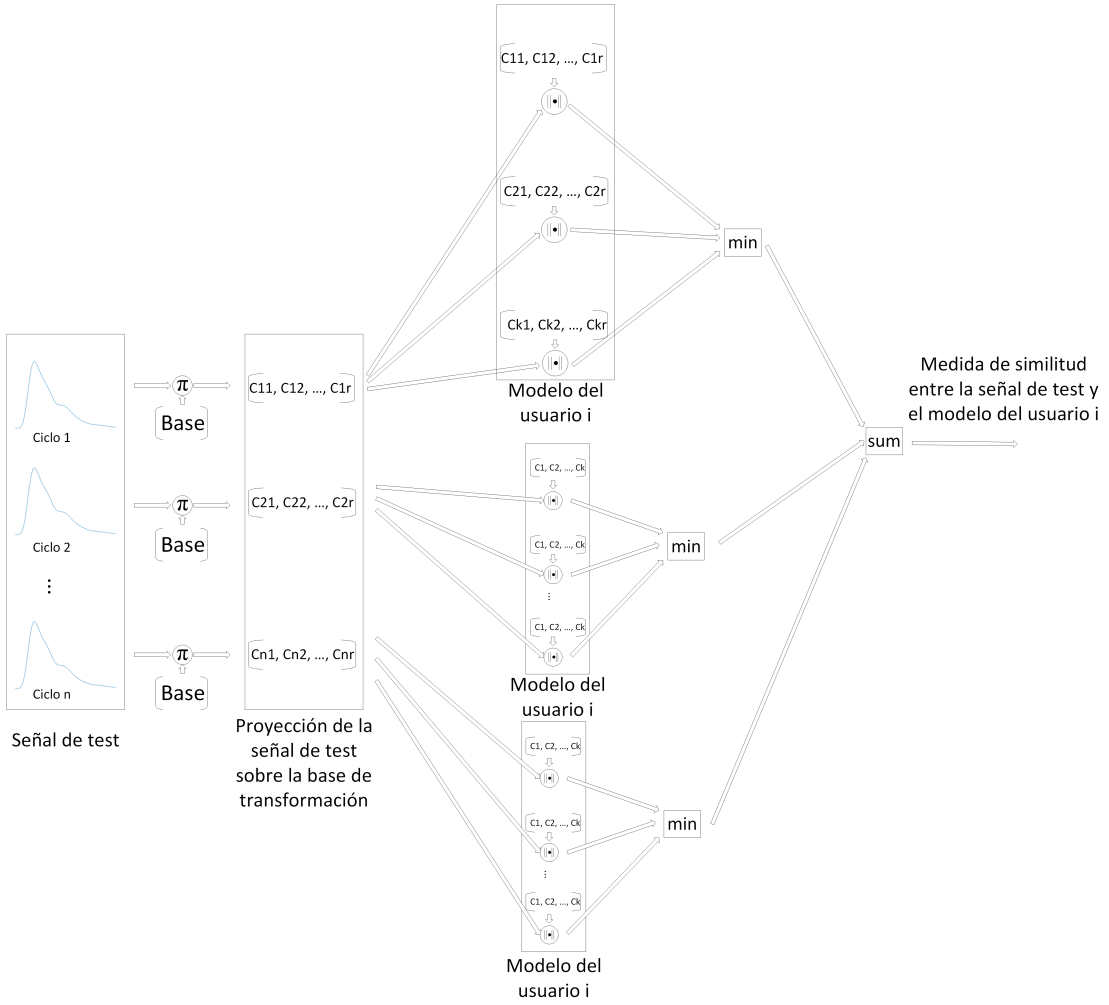


Figura 3.8: Cálculo de la medida de similitud en el dominio transformado.

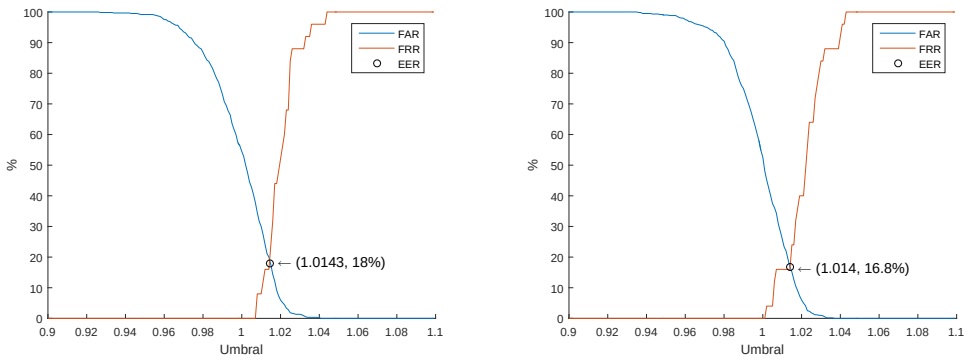
3.4 Resultados

A diferencia de los sistemas de autenticación tradicionales, como aquellos basados en contraseña o token, cuyo resultado es booleano, en biometría el resultado es una o varias medidas de similitud entre una señal de test y uno o varios modelos, en base a las cuales se decide si un usuario es quien dice ser, si el sistema esta trabajando en modo verificación, o la identidad del individuo, si trabaja en modo identificación. Consecuentemente, existe un margen para el cual

la decisión es errónea, por lo que la bondad de un sistema se mide en términos estos errores.

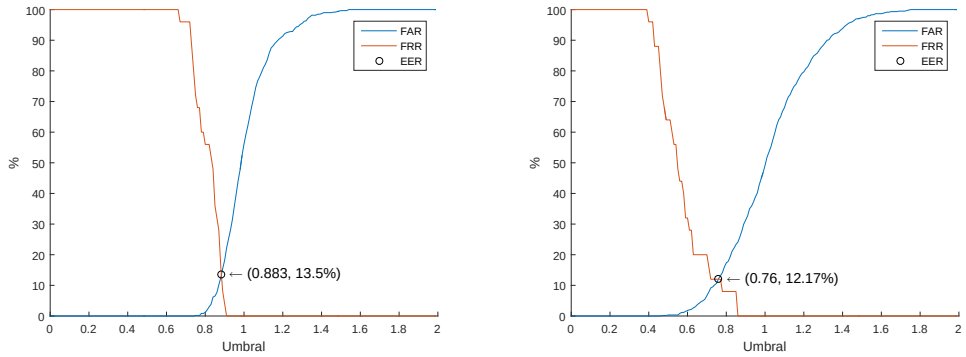
Típicamente las prestaciones de cualquier sistema biométrico se evalúan en términos de FAR, FRR y EER las cuales caracterizan al sistema trabajando en modo verificación. Se define el FAR o tasa de falsa aceptación como el porcentaje de verificaciones en las que el sistema da acceso a un usuario sin autorización y el FRR o tasa de falso rechazo como el porcentaje de verificaciones en las que el sistema niega el acceso a un usuario con autorización. El punto de cruce de las curvas de FAR y FRR se denomina EER o tasa de igual error, el cual es un valor ampliamente utilizado para comparar sistemas biométricos entre sí. Para homogeneizar los resultados, en todos los casos la distancia de una señal a un modelo se ha normalizado por la distancia de esa misma señal a todos los modelos antes de comparar con el umbral.

Para cada uno de los métodos analizados se han obtenido las gráficas de FAR y FRR en función del umbral y se indica el valor del EER en una población de 25 usuarios. Para garantizar la confiabilidad de los resultados, todos los métodos se han analizado por duplicado con las señales adquiridas con dos pulsioxímetros diferentes. En las figuras 3.9, 3.10 y 3.11 se pueden observar los valores de FAR, FRR y EER obtenidos en el dominio temporal y en el dominio transformado utilizando la KLT y LDA con ambos pulsioxímetros.



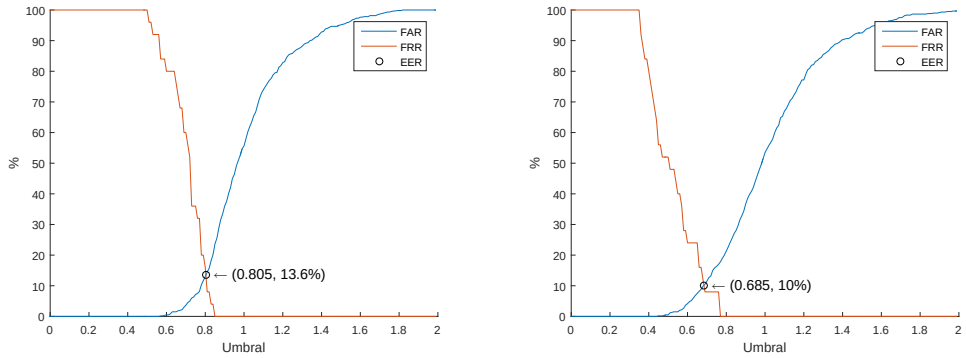
(a) Berry (b) Nonin

Figura 3.9: FAR, FRR y EER en el dominio del tiempo.



(a) Berry (b) Nonin

Figura 3.10: FAR, FRR y EER en el dominio transformado utilizando la KLT.



(a) Berry (b) Nonin

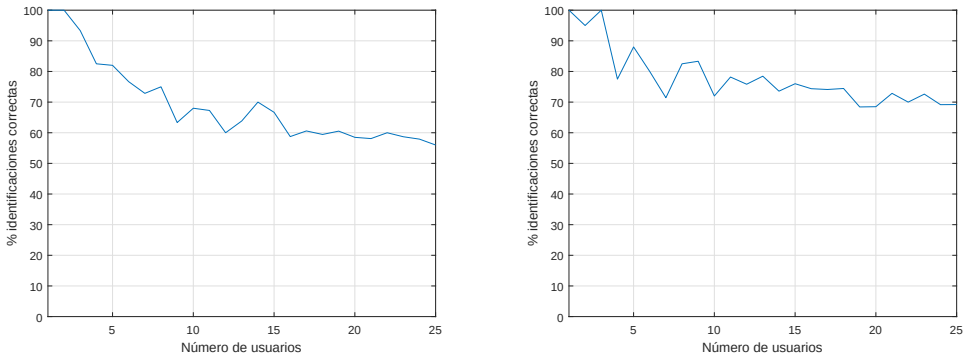
Figura 3.11: FAR, FRR y EER en el dominio transformado utilizando LDA.

Aunque típicamente las prestaciones de los sistemas biométricos se evalúan utilizando las gráficas anteriores, estas prestaciones representan al sistema trabajando en modo verificación. Como aproximación de los resultados que podrían esperarse del sistema trabajando en modo identificación pueden observarse las siguientes gráficas que muestran el porcentaje de identificaciones correctas en función del número de usuarios en el sistema.

Para obtener estas gráficas se realiza el proceso de identificación sobre un subconjunto de usuarios generado aleatoriamente para cada número de usuarios. Si se obtuviesen los resultados con una sola realización para cada número de usuarios, un único error en un subconjunto pequeño supondría una gran reducción

en el porcentaje de identificaciones correctas. Sin embargo ese resultado no sería realista. Para poder considerar los resultados como representativos, se debe realizar el proceso de identificación en varios subconjuntos formados por diferentes usuarios para cada número de usuarios y obtener el valor medio de identificaciones correctas. A este tipo de simulación se la conoce como análisis de Montecarlo y en este caso se ha ejecutado con 10 realizaciones.

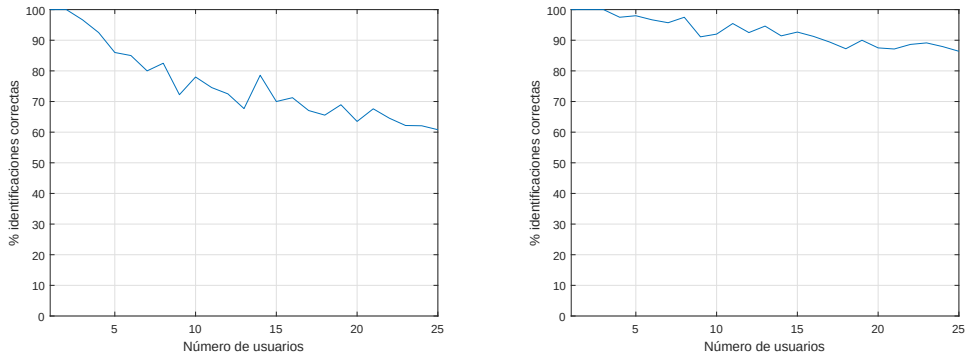
En las figuras 3.12, 3.13 y 3.14 se puede observar el porcentaje de identificaciones correctas obtenidas en el dominio temporal y en el dominio transformado utilizando la KLT y LDA tanto con el pulsioxímetro Berry como con el Nonin.



(a) Berry (b) Nonin

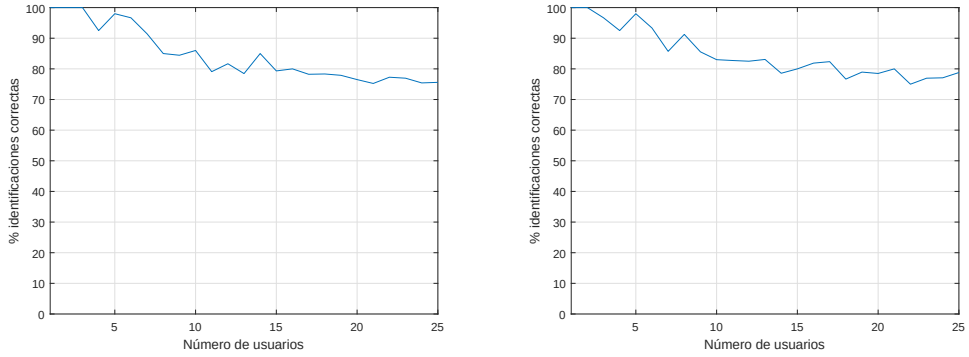
Figura 3.12: Porcentaje de identificaciones correctas en el dominio del tiempo.

Se han analizado las prestaciones obtenidas con tres algoritmos diferentes, en modo verificación e identificación y utilizando dos pulsioxímetros. En modo verificación, las mejores prestaciones se obtienen en el dominio transformado, utilizando LDA, con el pulsioxímetro de Nonin. En este caso se alcanza un EER del 10 %, el cual se encuentra en el orden de magnitud de los resultados obtenidos con la huella dactilar, que según se indica en el articulo "Performance Evaluation of Fingerprint Verification Systems"[10] oscila entre el 2 y 4 % para los 10 algoritmos que mejores prestaciones presentan. Como puede observarse, a pesar de que el pulsioxímetro de Nonin ofrece mejores prestaciones en todos los casos analizados, las diferencias obtenidas entre ambos equipos en términos de EER únicamente



(a) Berry (b) Nonin

Figura 3.13: Porcentaje de identificaciones correctas en el dominio transformado utilizando la KLT.



(a) Berry (b) Nonin

Figura 3.14: Porcentaje de identificaciones correctas en el dominio transformado utilizando LDA.

oscilan entre el 1.5 y 3 %.

El mejor resultado obtenido en modo de identificación se consigue en el dominio transformado, utilizando la KLT, con el pulsioxímetro de Nonin. En este caso, el porcentaje de identificaciones correctas se sitúa muy próximo al 90 % incluso con 25 usuarios en el sistema. Sin embargo este método muestra una alta disparidad en los resultados obtenidos, ya que al aplicarlo a los datos obtenidos con el pulsixímetro de Berry únicamente se obtiene un 60 % de identificaciones correctas. Analizando el algoritmo en el dominio transformado, utilizando LDA, se obtienen unos resultados más uniformes, con unos valores entre el 75 y 80 % de identificaciones correctas

con ambos pulsioximetros para el número máximo de usuarios. También con ambos pulsioxímetros, se obtiene más de un 85 % de aciertos siempre que haya 10 o menos usuarios en el sistema, situación que se contemplaba en el planteamiento inicial del sistema.

Para la implementación en la arquitectura final se ha decidido utilizar el algoritmo en el dominio transformado con LDA por ser el que ofrece las mejores prestaciones para el dispositivo de Berry, que debido al coste económico asociado a ambos dispositivos sería el elegido en un despliegue real.

Capítulo 4

Arquitectura del sistema

Se ha diseñado e implementado una arquitectura de almacenamiento y procesado, basada en la computación en la nube, que permita realizar la identificación de usuarios en tiempo real aprovechando las técnicas que mejores prestaciones presentaron en el análisis del capítulo anterior. Los elementos que forman dicha arquitectura pueden observarse en la figura 4.1 y serán explicados a continuación con mayor grado de detalle.

4.1 Client

Es la aplicación que corre en el equipo del usuario y está compuesta por 4 componentes, el Bluetooth handler, el proxy Bluetooth-FHIR, el generador de peticiones y el visualizador.

- El **Bluetooth handler** deberá interactuar con el driver de Bluetooth realizando un escaneo de los dispositivos próximos y estableciendo comunicación con el pulsioximetro en el caso de que se encuentre disponible. La comunicación se realizará mediante el protocolo de transporte RFCOMM, el cual es transmitido sobre L2CAP y sirve para el intercambio de datos vía serie de forma inalámbrica emulando el comportamiento de un puerto RS-232. Finalmente, extraerá los campos de datos de los paquetes recibidos y se los entregará al proxy Bluetooth-FHIR.

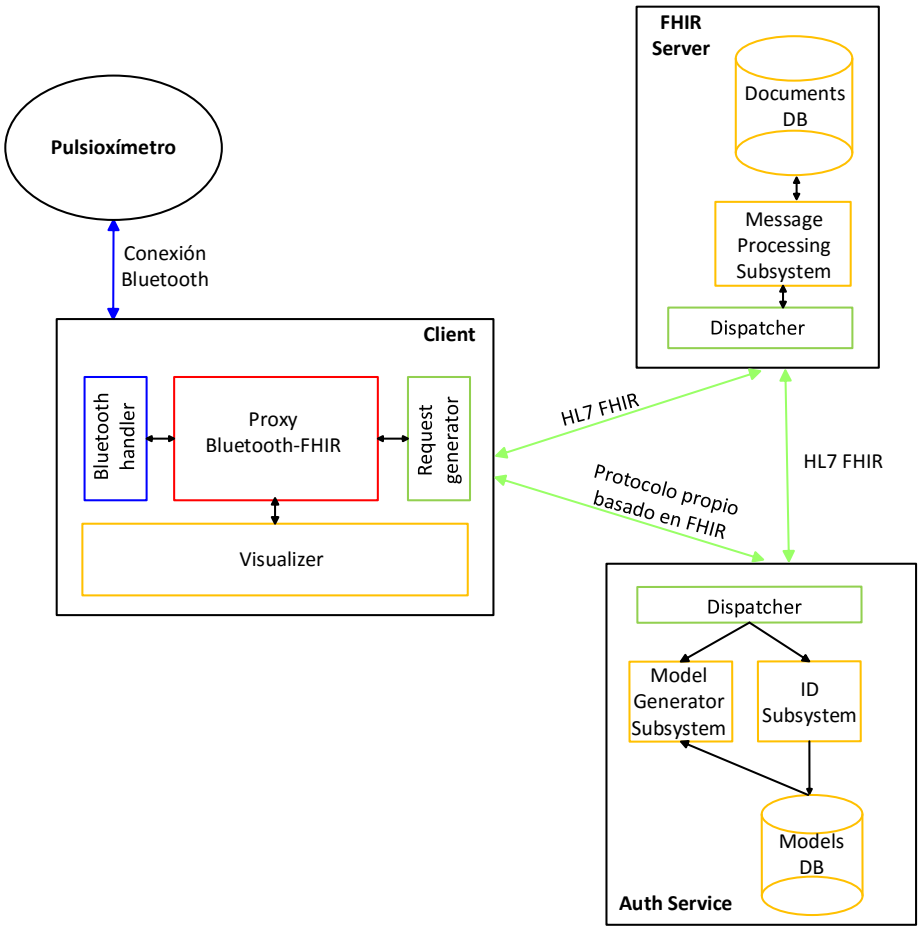


Figura 4.1: Arquitectura del sistema.

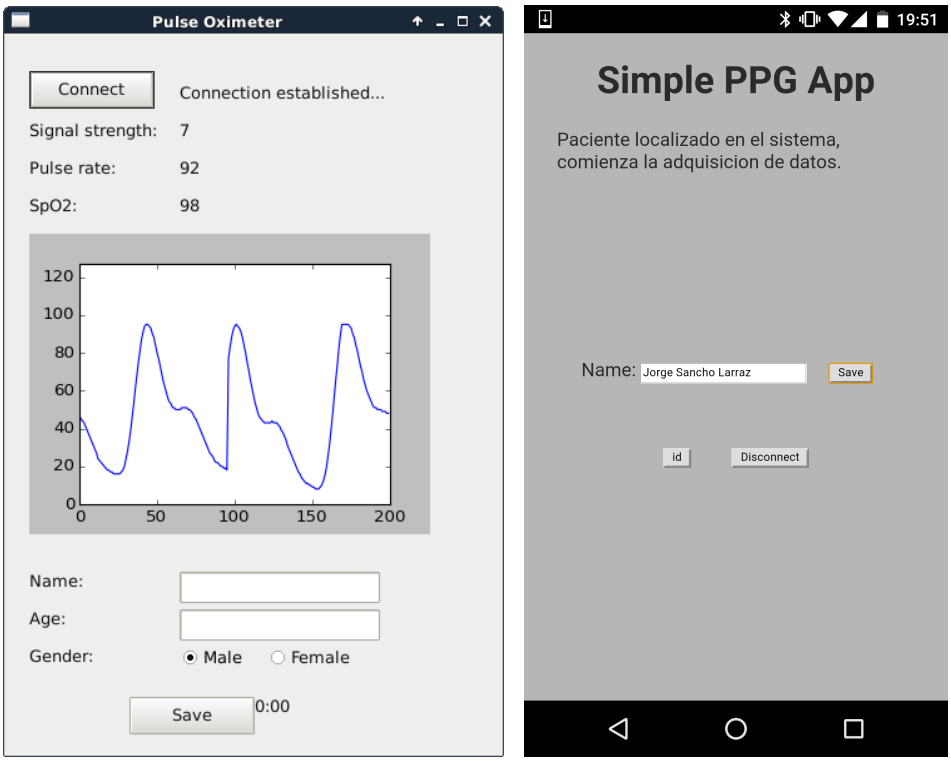
- El **proxy Bluetooth-FHIR** interpretará la información recibida y seleccionará aquella que se considere relevante, en este caso, la frecuencia cardiaca, la saturación de oxígeno en sangre, la señal PPG y los campos de sincronización. A continuación, codificará dicha información en instancias de recursos FHIR utilizando JSON y se los entregará al request generator.

- El **generador de peticiones** tiene dos modos de funcionamiento. Uno en el que la comunicación se establece con el "FHIR Server" con el objetivo de almacenar los datos médicos recibidos del proxy y un segundo el que la comunicación se establece con el "Auth Service" con el fin de determinar la identidad del usuario.

- El **visualizador** es el encargado de gestionar la interfaz gráfica, este

elemento esta implementado como un proceso independiente que accederá a la información decodificada por el proxy y la mostrará por pantalla.

Se han implementado dos versiones de esta aplicación, una para PC utilizando Python, por contar con soporte para el uso de todas las tecnologías necesarias, y otra para terminales móviles utilizando Apache Cordova, el cual es un marco de desarrollo que permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo cada plataforma móvil.



(a) Python (b) Android

Figura 4.2: Interfaces gráficas de las aplicaciones cliente.

4.2 FHIR Server

El servidor FHIR es el encargado de almacenar e interactuar con las instancias de los recursos FHIR, actuando a modo de EHR (historial clínico electrónico).

Este servidor está formado por tres componentes principales, el dispatcher, el subsistema de procesado de mensajes y la base de datos de documentos.

- El **dispatcher** es la encargada de recibir las peticiones HTTP, desempaquetarlas y entregárselas al subsistema de procesado de mensajes. Del mismo modo, se encarga de empaquetar y enviar los mensajes de respuesta generados por el subsistema de procesado de mensajes. Para la implementación de este elemento se ha decidido utilizar la librería estándar "BaseHTTPserver" de Python, debido a la facilidad que ofrece este lenguaje de programación para el manejo de datos codificados utilizando JSON.

- El **subsistema de procesado de los mensajes** es la parte central del servidor. Se encarga de realizar las operaciones indicadas en los mensajes sobre los recursos FHIR. Las operaciones soportadas son "read", "search", "create" y "delete" sobre los recursos "patient" y "observation". En el caso de recibir una petición mal formada, no soportada o sobre un recuso no soportado, generará el mensaje de error correspondiente y en el caso de que la petición sea correcta, realizará las consultas necesarias a la base de datos y generará la respuesta pertinente con los datos obtenidos. El mensaje de respuesta será entregado al dispatcher para su envió al cliente.

- La **base de datos de documentos** es el elemento en el cual se almacenará toda la información referente tanto a los pacientes como a las pruebas. Se decidió utilizar una base de datos NoSQL (not only SQL) por ajustarse perfectamente al escenario de aplicación y a los datos que se almacenarán. Estas bases de datos permiten almacenar documentos completos especialmente aquellos con formato JSON, lo cual encaja a la perfección con la aplicación desarrollada, ya que los recursos FHIR pueden serializarse utilizando este formato. Además, si la información esta codificada con JSON, se pueden realizar búsquedas en el interior del documento especificando una serie de filtros con formato {"key" : "value"}. La implementación de NoSQL que se ha decidido utilizar es "MongoDB" por ser una de la implementaciones más extendida, muy ligera, escalable y

de código abierto. Para realizar la interacción con la base de datos desde el subsistema de procesado de mensajes se ha utilizado la librería pyMongo. A modo de ejemplo, a continuación puede observarse un recuso de tipo Observation codificado con JSON:


```
{
    “resourceType”: “Observation”,
    “identifier”: “ppg”,
    “status”: “final”,
    “category”: {
        “coding”: [
            {
                “system”: “http://hl7.org/fhir/observation-category”,
                “code”: “procedure”,
                “display”: “Procedure”
            }
        ]
    },
    “code”: {
        “coding”: [
            {
                “system”: “urn:oid:2.16.840.1.113883.6.24”,
                “code”: “131328”,
                “display”: “MDC_ECG_ELEC_POTL”
            }
        ]
    },
    “subject”: {
        “reference”: “Patient/f001”,
        “display”: “P. van de Heuvel”
    },
```

“effectiveDateTime”: “2015-02-19T09:30:35+01:00”,
“performer”: [
{
“reference”: “Practitioner/f005”,
“display”: “A. Langeveld”
}
],
“device”: {
“display”: “12 lead EKG Device Metric”
},
“valueSampledData”: {
“origin”: {
“value”: 2048
},
“period”: 10,
“factor”: 1,
“lowerLimit”: -3300,
“upperLimit”: 3300,
“dimensions”: 1,
“data”: “2041 2043 2037 2047 2060 2062 2051 2023 2014 2027 2034 2033 2040 2047 2047 2053 2058 2064 2059 2063 2061 2052 2053 2038 1966 1885 1884 2009 2129 2166 2137 2102 2086 2077 2067 2067 2060 2059 2062 2062 2060 2057 2045 2047 2057 2054 2042 2029 2027 2018 2007 1995 2001 2012 2024 2039 2068 2092 2111 2125 2131 2148 2137 2138 2128 2128 2115 2099 2097 2096 2101 2101 2091 2073 2076 2077 2084 2081 2088 2092 2070 2069 2074 2077 2075 2068 2064 2060 2062 2074 2075 2074 2075 2063 2058 2058 2064 2064 2070 2074 2067 2060 2062 2063 2061 2059 2048 2052 2049 2048 2051 2059 2059 2066 2077 2073”
}
}

4.3 Auth Service

El servicio de identificación es el da sentido a la arquitectura propuesta. Como se explica en los resultados del capítulo anterior, entre todas las técnicas de identificación analizadas, se ha decidido implementar en el servidor una técnica basada en LDA utilizando la distancia al vecino más cercano como criterio de decisión. Este servicio está formado por cuatro elementos, el dispathcer, el subsistema de generación de modelos, el subsistema de identificación y la base de datos de modelos.

- El **dispatcher** es el elemento encargado de recibir las peticiones, desempaquetarlas y entregárselas al subsistema correcto. También, se encarga de empaquetar y enviar los mensajes de respuesta. Las principales peticiones soportadas son la solicitud de actualización de los modelos que se encuentran almacenados y la solicitud de iniciar, detener o consultar el estado actual de un proceso de identificación. Al igual que en el servidor FHIR, se ha decidido utilizar la librería estándar "BaseHTTPserver" de Python.

- El **subsistema de generación de modelos** se ejecuta cuando el dispatcher recibe una solicitud de actualización de los modelos. Este módulo establece comunicación con el servidor FHIR y obtiene todos los documentos que contengan señales PPG. A partir de dichas señales se realiza un análisis LDA, obteniendo una matriz de proyección y un conjunto de coeficientes para cada usuario que formarán su modelo. Tanto la matriz de proyección como los modelos calculados, se formatearán en JSON y se almacenan en la base de datos de modelos.

- El **subsistema de identificación** iniciará un nuevo proceso de identificación cuando el dispatcher reciba la petición correspondiente. En primer lugar, recuperará la matriz de proyección y los modelos de los usuarios de la base de datos de modelos. Utilizando esta información junto a la señal incluida en la petición del cliente, se procederá a realizar la identificación, hasta que la distancia entre el modelo más probable y el siguiente supere

un umbral. Dicho umbral regula el tiempo de convergencia del sistema, de manera que si transcurrido un periodo de tiempo no existe ningún modelo cuya distancia sea razonablemente más pequeña que la del resto, el sistema indicará que no ha sido capaz de determinar la identidad del usuario. Finalizado el proceso de identificación, se le pasará al dispatcher la identidad del usuario para que la envía al cliente en el mensaje de respuesta.

- La **base de datos de modelos** es el elemento en el cual se almacenará toda la información referente tanto a los modelos de los usuarios como información auxiliar utilizada para su calculo. Al igual que en el caso anterior, será una base de datos NoSQL "MongoDB".

Se ha desarrollado una librería para realizar todas las tareas relacionadas con la generación de modelos y con el proceso de identificación. Dicha librería se ha escrito en lenguaje C para maximizar la eficiencia y minimizar el tiempo de procesado necesario, debido al alto coste computacional que suponen estas tareas. La documentación completa de la librería puede encontrarse en el anexo H.

Para la correcta integración de la librería con el resto de la aplicación, se desarrolló un API escrita en Python que permite utilizar las funciones principales de la librería como si se tratase de código Python nativo. La documentación completa de la API puede encontrarse en el anexo I.

Para el envío de datos desde la aplicación cliente durante un proceso de identificación se plantearon dos posibilidades. Una primera en la que el cliente almacenaría los datos durante un pequeño periodo de tiempo y los enviaría por bloques en el cuerpo de un mensaje HTTP. En la segunda, los datos se enviarían al servidor como un flujo de tiempo real, bien como un streaming sobre HTTP o bien con RTP/RTCP para dotar de mayor interactividad a la aplicación. Finalmente se decidió que no tenía sentido utilizar esta última opción ya que aportaba una complejidad extra y realmente no aportaría la interactividad deseada, puesto que el filtrado realizado en el servidor es de tipo forward-backward y únicamente puede aplicarse sobre un bloque de datos y no sobre un flujo.

4.4 Flujo de la información

Se han diseñado dos flujos de información en el funcionamiento normal de la arquitectura. Uno en el que el cliente establece la comunicación con el servidor FHIR con el objetivo de almacenar los datos médicos obtenidos del pulsioximetro y un segundo el que la comunicación la establece el cliente con el servidor de identificación con el fin de determinar la identidad del usuario.

4.4.1 Interacciones Client-FHIR Server

En este caso, el cliente establece la comunicación con el servidor FHIR con el objetivo de almacenar los datos médicos obtenidos del pulsioximetro. Para ello almacena todos los datos recibidos del pulsioximetro durante un cierto periodo de tiempo, pasado el cual inicia el procedimiento que puede observarse en la figura 4.3 para enviarlos al servidor FHIR para su almacenaje.

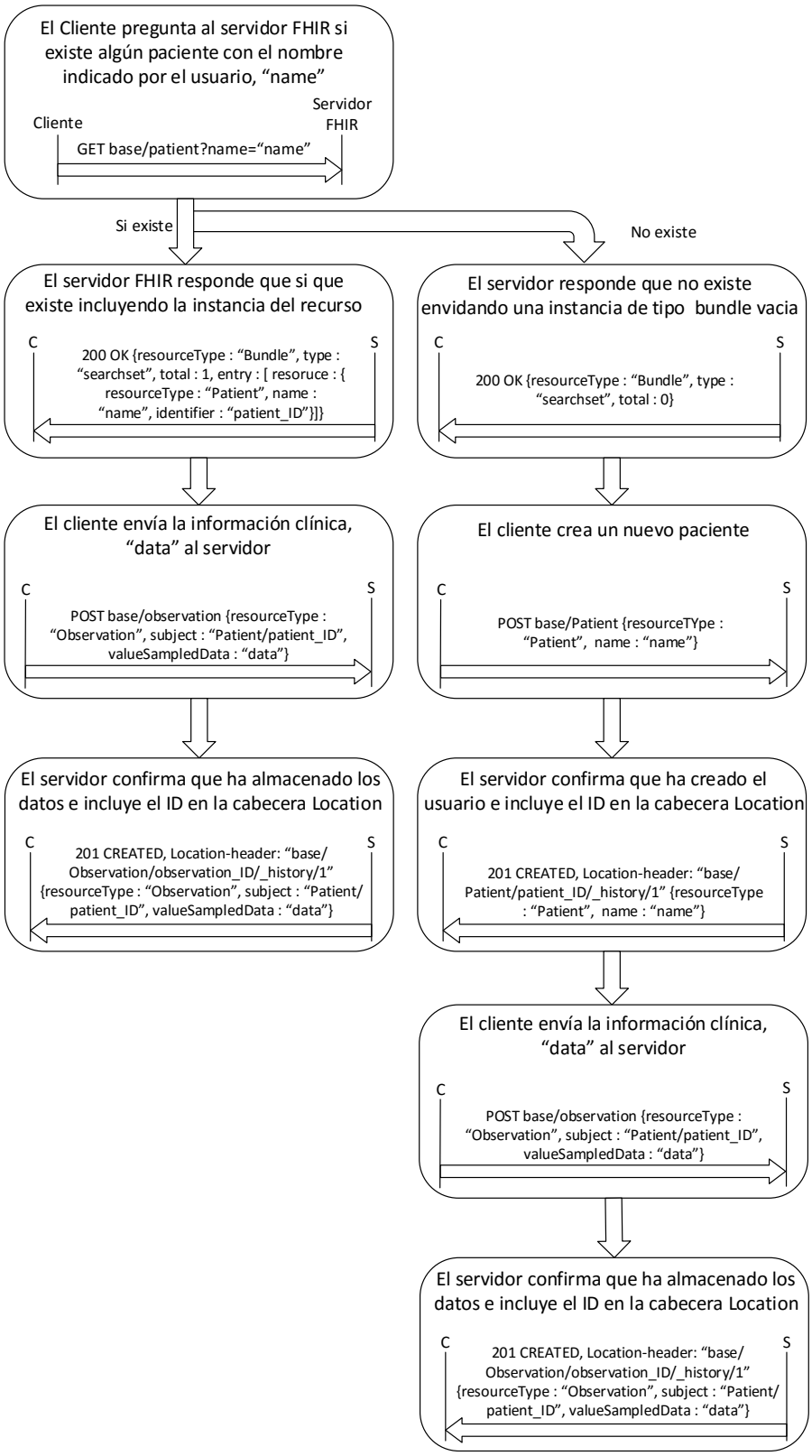


Figura 4.3: Flujo de mensajes para para el almacenamiento del PPG.

4.4.2 Interacciones Client-Auth Service

En este caso, el cliente se comunica con el servidor de identificación con el fin de determinar la identidad del usuario. Para ello los datos procedentes del pulsioximetro se enviarán al servidor de identificación cada 10 segundos siguiendo el flujo de mensajes que se muestra en la figura 4.4 hasta que se completa el proceso de identificación y el cliente recibe la identidad del usuario.

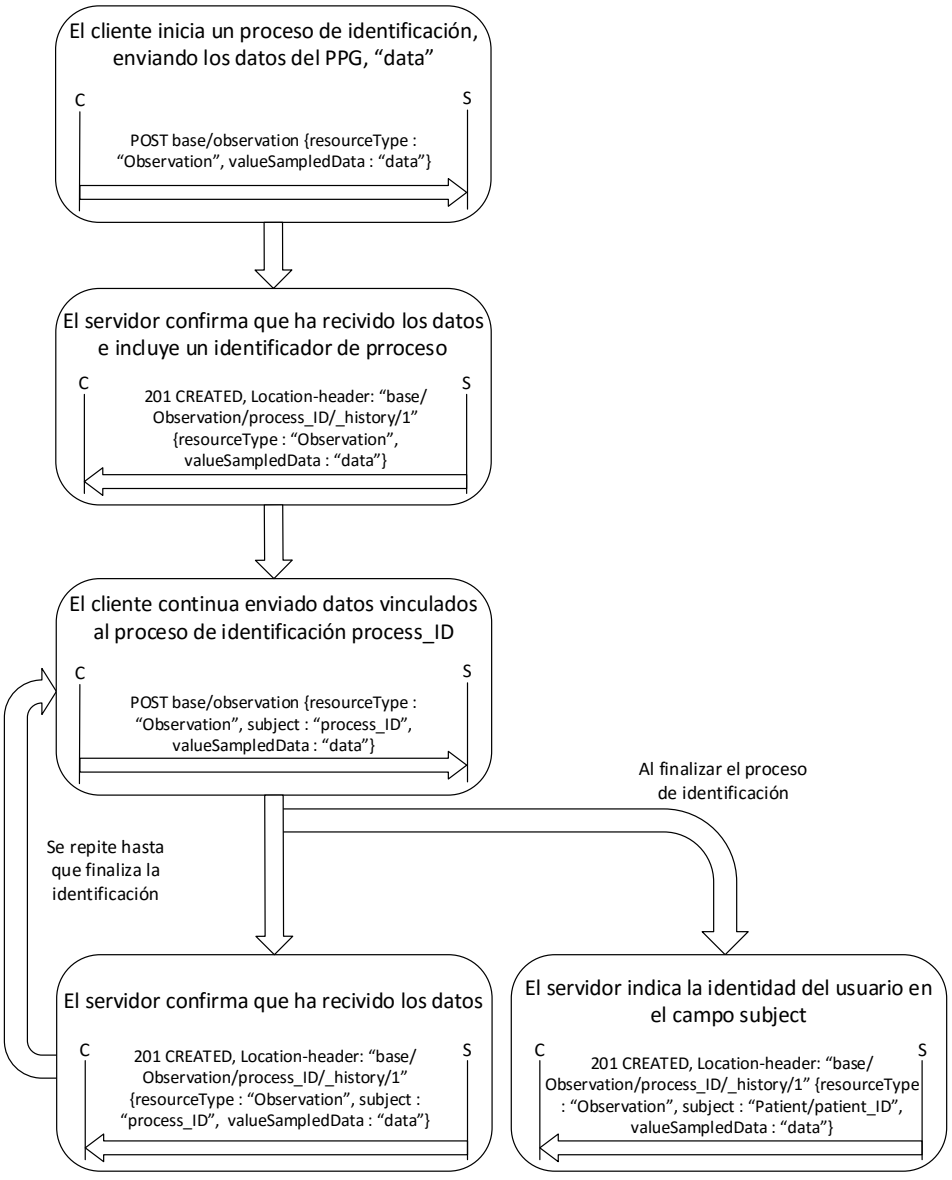


Figura 4.4: Flujo de mensajes para iniciar un proceso de identificación.

4.5 Implementación de la arquitectura global

La aplicación cliente, se distribuye como un único ejecutable en el caso de Python y como un fichero apk en el caso de Android. Para el correcto funcionamiento de la aplicación será necesario que el equipo que la ejecute cuente con una interfaz Bluetooth y los drivers correctamente configurados. Respecto al servidor FHIR y al servidor de identificación, se han configurado como dos instancias t2.micro en la nube de Amazon, AWS (Amazon Web Services). Para los usuarios que desean ejecutar estos servidores en sus propios equipos se han creado dos máquinas virtuales utilizando VMWare con todos los componentes de los servidores instalados y configurados.

Capítulo 5

Conclusiones y líneas futuras

5.1 Conlusiones

En este proyecto se han analizado las prestaciones de varios esquemas de procesado para su uso en biometría utilizando el PPG. Para dar soporte al proceso de identificación, se ha diseñado e implementado una arquitectura de procesado en la nube, en la que todas las comunicaciones se realizan sobre estándares de comunicación médicos, que permita realizar la identificación en tiempo real.

En las diferentes partes del proyecto, se han analizado las principales herramientas para realizar cada una de las tareas, y se han seleccionado las opciones más interesantes, siempre tratando de optimizar el rendimiento para obtener un resultado funcional que pudiese ser utilizado incluso en escenarios con un número elevado de usuarios. Además, se ha desarrollado todo de forma modular para permitir que cualquier parte de este trabajo pudiese ser reaprovechada en futuros proyectos.

En primer lugar, se implementaron diversos algoritmos que permitiesen explotar la capacidad identificativa la señal PPG, aprovechando características tanto del dominio temporal como varios dominios transformados y se analizaron las prestaciones ofrecida por cada uno de ellos. Para realizar esta tarea se utilizo MATLAB por permitir realizar un rápido desarrollo de los algoritmos. Se encontró que el método que presentaba unos resultados más homogéneos entre los dos

pulsioxímetros es LDA utilizando la distancia al vecino más cercano como medida de similitud.

A continuación, se desarrollaron varios elementos que conjuntamente componen una arquitectura que permite realizar la identificación de un usuario aplicando la técnica nombrada en el párrafo anterior. El primero de estos elementos es una aplicación capaz de establecer comunicación vía Bluetooth con el pulsioximetro, recibir, procesar y reformatear los datos, representarlos gráficamente y enviarlos a la nube para su almacenamiento y procesado.

Otro de estos elementos es un servidor para el almacenamiento de los datos médico basado en el estandar HL7 FHIR, el cual utiliza una arquitectura RESTful para el intercambio de datos representados por unos modelos de información especificados por el estándar y serializados utilizando JSON.

El último de los elementos de la arquitectura es el servicio de identificación, el cual es el encargado de realizar la generación de los modelos de los usuarios y realizar la identificación propiamente dicha.

Dada la alta complejidad computacional de los algoritmos de identificación analizados, y para facilitar la escalabilidad del sistema, se ha desarrollado una librería escrita completamente en C, que aporta toda la funcionalidad necesaria para implementar de forma eficiente el algoritmo de identificación.

Se ha desarrollado una API (interfaz de programación de aplicaciones) que permita utilizar la librería desarrollada desde un lenguaje de alto nivel, en este caso Python, debido a que la decodificación y gestión de los datos recibidos utilizando el estándar HL7 FHIR se ha realizado utilizando este lenguaje.

Para la comunicación con el servidor de identificación se ha definido un protocolo basado en FHIR para realizar las tareas de actualización de modelos e iniciar, detener y consultar el estado de un proceso de identificación.

Por último, se han creado dos instancias en la nube de Amazon, AWS, donde se han lanzado servidor FHIR y el servidor de identificación. Además, se han creado dos máquinas virtuales con los servidores instalados y configurados

5.2 Lineas futuras

Aunque todos los objetivos planteados se han cumplido en el presente TFM, se plantean unas posibles mejoras que aportarían funcionalidades extra al sistema:

- **Analizar resultados con señales de días diferentes:** Todos los resultados presentados en el capítulo 3 se han generado capturando una única señal para cada usuario y utilizando la mitad para generar los modelos y la otra mitad para realizar la identificación. Resultaría muy interesante analizar en qué medida varían los resultados aquí obtenidos al utilizar señales adquiridas en días diferentes para generar los modelos y realizar la identificación.

- **Utilizar varias longitudes de onda para la identificación:** Como se explica en el capítulo 2.1, el PPG corresponde a las variaciones en la intensidad de luz absorbida por el cuerpo en una longitud de onda cercana a los 900 nm. Podría analizarse la capacidad identificativa aportada por diferentes longitudes de onda.

- **Utilizar la video-pletismografía para la identificación:** Se podría estudiar la opción de utilizar una cámara CCD infraroja para capturar la señal en lugar de un fotodetector. Esto permitiría analizar el PGG en varios puntos del cuerpo al mismo tiempo y su correlación.

- **Evolución temporal de los modelos:** Recoger estadísticas sobre qué puntos de cada modelo aparecen un mayor número de veces en las identificaciones correctas y erróneas. Se podrían mejorar los modelos añadiendo y quitando puntos dinámicamente en base a estas estadísticas.

- **Modificar la librería PPG_lib para soportar el procesado en GPU:** Analizar cuáles de las funciones implementadas en la librería de procesado de la señal PPG desarrollada muestran una mayor criticidad en tiempo de computación y modificarlas para soportar el procesado paralelo basado en GPU utilizando, por ejemplo, OpenCL, CUDA o Stream.

- **Integración de la arquitectura con el framework XACML:** Para garantizar la privacidad de los usuarios, sería conveniente incluir un sistema de control de acceso que utilizando una correcta política de autorización impidiese que un individuo sin los permisos adecuados pudiese acceder a la información de un usuario. Para realizar esta tarea se propone integrar el framework XACML en el servidor FHIR.

Bibliografía

[1] Angelo Bonissi, Luca Perico Roberto Sassi Fabio Scotti Luca Sparagino, Ruggero Donida Labati (Septiembre del 2013). «A Preliminary Study on Continuous Authentication Methods for Photoplethysmographic Biometrics». *IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*.

[2] Elgendi, Mohamed (Agosto del 2012). «On the Analysis of Fingertip Photoplethysmogram Signals». *Current Cardiology Reviews*.

[3] Girish Rao Salanke N S, Dr. Maheswari N S.Sadhasivam, Dr. Andrews Samraj (Marzo del 2013). «Enhancement in the design of Biometric Identification System based on Photoplethysmography data». *International Conference on Green High Performance Computing*.

[4] Gu, Y. Y. y Zhay, Y. T. (Octubre del 2003). «Photoplethysmographic Authentication through Fuzzy Logic». *Conference on Biomedical Engineering*.

[5] K. F. Man, K. S. Tang y Kwong, S. (Octubre del 1996). «Genetic Algorithms: Concepts and Applications». *IEEE Transactions on Industrial Electronics*.

[6] Mike Banahan, Declan Brady y Doran, Mark (1991). *The C Book*. Addison Wesley.

[7] Newman, Sam (2015). *Building Microservices*. O'Reilly Media.

[8] Nur Azua Liyana Jaafar, Khairul Azami Sidek y Azam, Siti Nurfarah Ain Mohd (Mayo del 2015). «Acceleration Plethysmogram Based Biometric

Identification». *International Conference on BioSignal Analysis, Processing and Systems*.

[9] P. Spachos, Dimitrios Hatzinakos, Jiexin Gao (Enero del 2011). «Feasibility study of photoplethysmographic signals for biometric identification». *17th International Conference on Digital Signal Processing*.

[10] Raffaele Cappelli, Davide Maltoni James L. Wayman, Dario Maio y Jain, Anil K. (Enero del 2006). «Performance Evaluation of Fingerprint Verification Systems». *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[11] Singh, Dr. Mandeep y Gupta, Spiti (Enero del 2012). «Correlation studies of PPG finger pulse profiles for biometric system». *International Journal of Information Technology and Knowledge Management*.

[12] Tim Benson, Grahame Grieve (2016). *Principles of Health Interoperability: SNOMED CT, HL7 and FHIR*. Springer.

[13] Y. Y. Gu, andY. T. Zhang, Y. Zhang (Abril del 2003). «A Novel Biometric Approach in Human Verification by Photoplethysmographic Signals». *IEEE Transactions on Information Technology Applications in Biomedicine*.