UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Autenticación en Windows a través de reconocimiento facial con dispositivo móvil

Trabajo Fin de Máster

presentado por: Rojas Cala, Juan Pablo

Director/a: Blanco, Ramón

Ciudad: Cali, Colombia
Fecha: 20 de junio de 2016

Resumen

En este Trabajo Final de Máster se ha desarrollado un software que permite llevar a cabo un proceso de autenticación en Windows por medio de reconocimiento facial, donde se han usado diferentes tecnologías y librerías ya existentes que han agilizado el proceso de desarrollo. El eje central está dado por el uso del dispositivo móvil inteligente, con el que el usuario realiza las diferentes tareas que permiten llevar a cabo la autenticación en una sesión de Windows. Esto permite que la gestión de acceso sea más fácil tanto para los usuarios, como para el área de tecnología encargada de realizar el soporte respectivo a cuentas de usuarios en directorio activo. Además de lo fácil que puede ser la gestión de acceso, también se tiene que este esquema de implementación es más seguro, debido a las puertas de autenticación requeridas para otorgar acceso al usuario.

Palabras Clave: Autenticación, Reconocimiento facial, Directorio Activo, Dispositivo Móvil, Windows.

Abstract

In this master's project it has developed software that allows to perform an authentication process in Windows using facial recognition, where have used different technologies and existing libraries that have speeded up the development process. The central axis is given by the use of smart mobile device, with which the user performs different tasks that allow you to perform authentication in a Windows session. This makes things easier for both users and for the technology area responsible for carrying out the respective support user accounts in Active Directory. In addition to how easy that can be access management, there is also that this scheme's implementation is more secure, due to the doors of authentication required to grant access to the user.

Keywords: Authentication, facial recognition, Active Directory, Mobile Device, Windows.

Índice de contenido

Índice de figuras

Índice de tablas

1 Introducción

1.1 Motivación

Hoy en día es usual ver que las organizaciones hacen uso del sistema operativo Windows para las terminales de sus usuarios, donde además pueden disponer de 1 o más controladores de dominio tipo Windows server para el ciclo de vida de las cuentas de directorio activo de sus usuarios y demás requerimientos asociados. Sin embargo, el controlar y gestionar adecuadamente los accesos de los usuarios, puede volverse algo incómodo e inseguro para la organización. Esto debido a que es muy frecuente encontrar que los usuarios olvidan su contraseña para ingresar a su respectiva terminal de trabajo, ocasionando problemas tales como el número constante de llamadas a soporte para restablecer la contraseña, y el riesgo de seguridad que esto implica para la organización al disponer analistas con el poder de restablecer contraseñas para cualquier usuario. También los usuarios caen en el error de anotar la clave físicamente en papel, para poder así recordarla fácilmente, y claramente existen unas implicaciones de seguridad que ello implica. Otro de los problemas es que los usuarios acostumbran a prestar su clave, y hasta llegan a olvidar a quien la han prestado, en muchas ocasiones el riesgo puede aumentar, donde las organizaciones que implementen una herramienta de gestión de identidades como Oracle Identity Manager, Novell, que, por lo general, consiste en que los usuarios acceden a las aplicaciones internas de la organización a través de una clave única, por consiguiente otros usuarios pueden hacer uso de las aplicaciones propias de un usuario ajeno, ya que muchas veces el acceso principal corresponde a la clave de directorio activo asociada a la cuenta del usuario.

Todos estos problemas obviamente se originan a raíz del funcionamiento natural que se tienen en sí las cuentas de directorio activo, donde básicamente los usuarios obligadamente tienen que cambiar la contraseña periódicamente y recordarla todos los días para autenticarse en la terminal. Así, por ejemplo, un usuario que no tenga casi buena memoria, caerá continuamente en llamadas a soporte para el restablecimiento de la misma, y también cualquier usuario que llegue al día de cambio de contraseña, existirá una alta probabilidad que la nueva clave sea olvidada y desde luego recurra a soporte.

Adicional, las políticas de contraseña impuestas por la organización pueden llegar a ser complejas, haciendo un tanto difícil para el usuario pensar una nueva clave que no olvide tan fácilmente. Estas políticas pueden ir desde contener ciertos caracteres especiales, como cierta longitud, el uso de mayúsculas y minúsculas o números, que para una persona con perfil informático podría ser trivial, sin embargo, para un usuario ajeno al tema puede resultar difícil.

1.2 Planteamiento del trabajo

El propósito es reevaluar la forma en que los usuarios realizan la autenticación en Windows, buscando atacar los problemas descritos anteriormente, apoyándose en herramientas ya existentes, de modo que la implementación no sea tan costosa y demasiado compleja.

Principalmente, lo que se busca es mejorar el proceso de autenticación de los usuarios en sus respectivas terminales de forma más segura y más fácil de usar. Para lograr esto, de cara al usuario, lo que se plantea es el uso de identificación biométrica de reconocimiento facial, proceso que se realiza mediante un dispositivo móvil asociado al usuario que sirve de llave maestra, sumado a esto, se dispone un servidor biométrico encargado de realizar el cálculo de reconocimiento facial contra la base de datos de imágenes de empleados. Así, el inicio de sesión en Windows pasaría de ser un proceso simple de ingresar una contraseña, a un acceso con un token o clave generado a través del dispositivo móvil con validación de reconocimiento facial, de modo que se gane terreno el campo de la seguridad, dando un valor agregado a los usuarios, ya que tendrían una forma más fácil y cómoda de gestionar su contraseña en Directorio Activo.

En la Figura 1, se muestra un poco más el proceso descrito, en donde se resalta a grandes rasgos los pasos y elementos que componen el proceso:

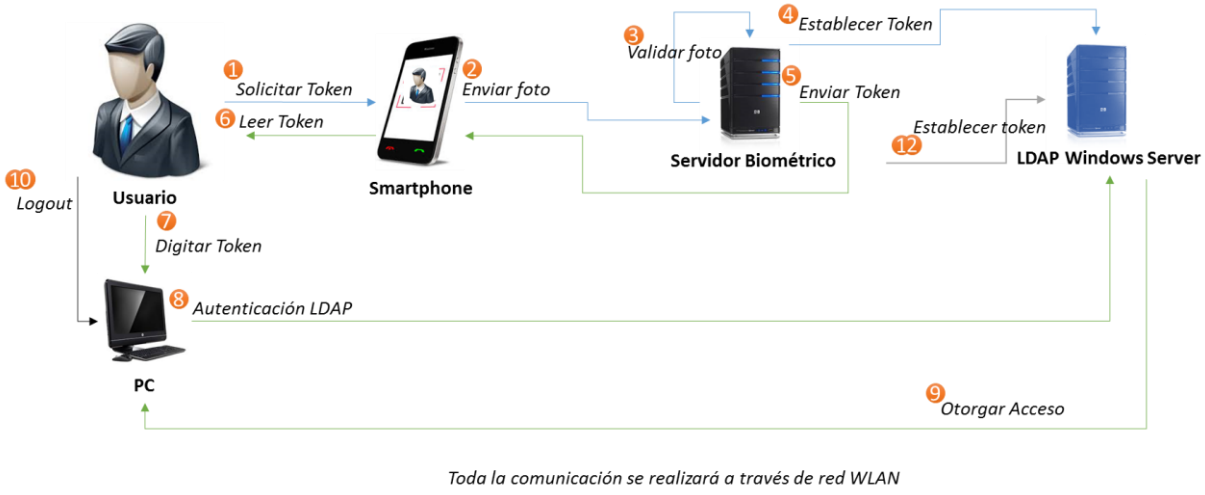


Figura 1: Implementación general sistema de autenticación por reconocimiento facial.

El paso 1 del proceso muestra que el usuario debe solicitar el token a través del dispositivo móvil, luego en el paso 2 la aplicación móvil envía la foto de la persona que ha solicitado el token, posterior, en el paso 3, el servidor biométrico se encarga de validar si la imagen corresponde al usuario que ha solicitado el token por medio del reconocimiento facial, si es así, continua el paso 4 donde el servidor biométrico establece una nueva clave aleatoria y la establece en LDAP al usuario de red correspondiente al usuario, luego en el paso 5 el

servidor biométrico encripta esta clave y la envía al dispositivo móvil donde este se encargará de des encriptarla y mostrarla al usuario, así el usuario con los pasos 6 y 7 leerá la clave que le muestra en pantalla el dispositivo móvil y la ingresará en su estación de trabajo. Los pasos 8 y 9 corresponde a la autenticación y autorización por parte del Directorio Activo, donde validará la credencial ingresada por el usuario.

1.3 Estructura del trabajo

En la sección 2 se encuentra el apartado del contexto y estado del arte, el cual se divide en 6 capítulos, donde el primero trata sobre los dispositivos móviles y sus tipos de seguridad, el segundo es sobre la introducción a los sistemas biométricos, el tercer capítulo se realiza sobre toda la tecnología de biometría de reconocimiento facial, dado que es donde el trabajo se enfoca, el cuarto sobre los algoritmos de reconocimiento facial usados en este trabajo, el quinto trata sobre la tecnología usada para la integración del servidor biométrico con la librería usada de reconocimiento facial, y el sexto trata la arquitectura usada para realizar la comunicación entre el servidor biométrico y el dispositivo móvil.

Luego se encuentra la presentación de los objetivos tanto generales como específicos en la sección 3, para dar el enfoque hacia donde se dirige el trabajo.

También se tiene un apartado del desarrollo específico de la contribución del trabajo realizado en la sección 4, donde se describirán los requerimientos, la descripción, evaluación y resultados del desarrollo del software.

Por último, se tienen las conclusiones y trabajo futuro en la sección 5, donde se explican los diferentes hallazgos y posibles trabajos a realizar a partir del software desarrollado.

2 Contexto y Estado del Arte

2.1 Dispositivo móvil como llave maestra

El auge que han tenido los dispositivos móviles en los últimos años ha sido un tanto considerable, hasta el punto, en que hoy la experiencia que tienen los usuarios es casi dependiente para las actividades que realizan en su vida cotidiana, de estas actividades se encuentran básicamente como el envío de mensajes, las llamadas telefónicas, el uso de redes sociales, la gestión del correo electrónico, realizar compras, revisar sus cuentas bancarias, entre otras (Jaime, Uribe, & Hernández, 2015). Es por esto que los fabricantes de dispositivos móviles ponen esfuerzos en brindar una experiencia al usuario de forma segura, trayendo consigo diferentes métodos de autenticación en los dispositivos móviles, dentro dichos métodos se encuentran: clave simple o passcode, patrón de desbloqueo, Knock Code de LG, reconocimiento de iris, huellas dactilares, entre otros. Seguramente el día de

mañana habrán inventado nuevos métodos y mejorado los ya existentes, esto quiere decir que el grado de seguridad de un dispositivo móvil tiende a ser positivo, dado que las empresas seguirán compitiendo por ofrecer la mejor seguridad a sus usuarios.

Dado lo anterior, cabe la siguiente pregunta, ¿se podría visualizar un dispositivo móvil como una llave maestra?, desde luego, sin embargo, es importante dejar claro que la configuración de seguridad para autenticación en los dispositivos móviles es algo que es opcional para el usuario, lo que quiere decir que la llave maestra puede, por simple negligencia del usuario, ser una llave con una seguridad pobre o nula, dando cavidad a que otra persona acceda a todas las funciones que el usuario realiza, y las correspondientes consecuencias de esto.

Ahora bien, si se acepta finalmente a los dispositivos móviles como llaves maestras, sería interesante poder imaginarlos y aceptarlos como un medio de acceso a sistemas de información. Lo anterior es el punto de partida para empezar a diseñar el método que se implementó para lograr establecer el dispositivo móvil como una llave maestra a través del software desarrollado en este trabajo.

2.2 Sistemas biométricos

El enfoque principal de la solución planteada es sobre el proceso de reconocimiento biométrico, para ello lo primero es conocer las principales etapas generales que plantea este proceso (Jaime, Uribe, & Hernández, 2015):

- Adquisición: obtención del rasgo biométrico por medio de un dispositivo.
- Pre-procesamiento: localización del patrón biométrico a partir de la información obtenida en el proceso de adquisición, se remueve información que no corresponde al dato biométrico.
- Extracción de características: el patrón biométrico obtenido en el pre-procesamiento es llevado a una representación numérica.
- Clasificación: se realiza la comparación de las características extraídas con el patrón biométrico que se encuentra guardado.
- Toma de decisión: de acuerdo a la comparación que se realiza en el proceso de clasificación se decide si se trata de la identidad del usuario.

El detalle de cada proceso en cada etapa, desde luego ya dependerá del sistema biométrico implementado.

Otro aspecto a tener en cuenta es el uso que se tiene de cada sistema biométrico, como se puede ver en la Figura 2, el rasgo biométrico más usado es la huella dactilar, seguido del reconocimiento facial.

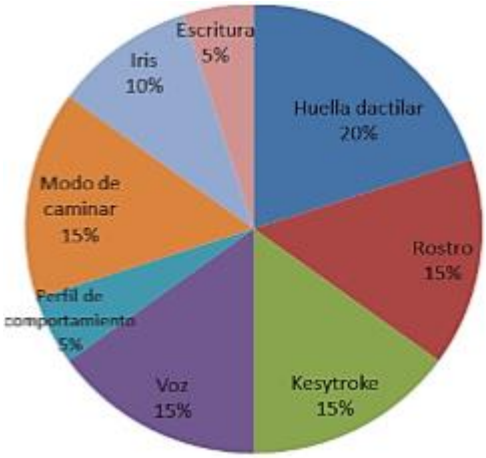


Figura 2: Uso de sistemas biométricos (Jaime, Uribe, Hernández, septiembre, 2015)

Sin embargo, como se ha mencionado anteriormente, lo que se busca también es que, de algún modo, converger a implementar un desarrollo que no sea tan exigente en términos de costos. El problema que traería una implementación haciendo uso de la huella dactilar del dispositivo móvil, es la misma existencia de este mecanismo en sólo la mayoría de dispositivos móviles con un alto costo considerable, donde resultado de una economía social promedio no todo usuario cuenta con un dispositivo móvil con tal característica. En contraste, el reconocimiento facial, gracias a que la adquisición es realizada por la misma cámara del dispositivo móvil, abre las puertas para que todos los usuarios que cuenten con dispositivo móvil se beneficien de este sistema biométrico.

2.3 Sistema biométrico de reconocimiento facial

2.3.1 ¿De qué se trata?

Los humanos por millones de años han desarrollado innatamente cómo diferenciar una persona de otra por su rostro, pues bien, las computadoras actualmente están llegando a realizar esta tarea, y es aquí donde se introduce el concepto de biometría de reconocimiento facial. Este concepto inicialmente se ha basado en diferentes estudios de psicología y neurociencia, que han ayudado a la construcción de diferentes algoritmos computacionales, adicionalmente, en los últimos 30 años se han propuesto diferentes métodos de reconocimiento facial por investigadores quienes han basado su trabajo en áreas como psicología, reconocimiento de patrones, redes neuronales, visión por computador y gráficos de computador, llegando a ser métodos complejos que a veces quedan difíciles de clasificar (Zhao, Chellappa & Phillips, 2003).

Una aproximación a la clasificación de estos algoritmos de reconocimiento facial es la que se presenta en la Tabla 1.

Tabla 1: Clasificación de algoritmos de reconocimiento facial (Zhao et al., 2003)

Approach	Representative work
Holistic methods	
Principal-component analysis (PCA)	
Eigenfaces	Direct application of PCA [Craw and Cameron 1996; Kirby and Sirovich 1990; Turk and Pentland 1991]
Probabilistic eigenfaces	Two-class problem with prob. measure [Moghaddam and Pentland 1997]
Fisherfaces/subspace LDA	FLD on eigenspace [Belhumeur et al. 1997; Swets and Weng 1996b; Zhao et al. 1998]
SVM	Two-class problem based on SVM [Phillips 1998]
Evolution pursuit	Enhanced GA learning [Liu and Wechsler 2000a]
Feature lines	Point-to-line distance based [Li and Lu 1999]
ICA	ICA-based feature analysis [Bartlett et al. 1998]
Other representations	
LDA/FLD	LDA/FLD on raw image [Etemad and Chellappa 1997]
PDBNN	Probabilistic decision based NN [Lin et al. 1997]
Feature-based methods	
Pure geometry methods	Earlier methods [Kanade 1973; Kelly 1970]; recent methods [Cox et al. 1996; Manjunath et al. 1992]
Dynamic link architecture	Graph matching methods [Okada et al. 1998; Wiskott et al. 1997]
Hidden Markov model	HMM methods [Nefian and Hayes 1998; Samaria 1994; Samaria and Young 1994]
Convolution Neural Network	SOM learning based CNN methods [Lawrence et al. 1997]
Hybrid methods	
Modular eigenfaces	Eigenfaces and eigenmodules [Pentland et al. 1994]
Hybrid LFA	Local feature method [Penev and Atick 1996]
Shape-normalized	Flexible appearance models [Lanitis et al. 1995]
Component-based	Face region and components [Huang et al. 2003]

Los algoritmos que se trataron en este trabajo de fin de máster están clasificados como métodos holísticos, los cuales usan toda la región de la cara como entrada al sistema de reconocimiento, adicional, estos algoritmos se encuentran en la librería en la que se basa este proyecto y que se presentará más adelante, junto con el análisis de los algoritmos usados para este trabajo.

2.3.2 Historia y Evolución

En la década de los 60, se presenta un sistema parcialmente semiautomático de reconocimiento facial, funcionaba por medio de un administrador, quien se encargaba de ubicar las coordenadas de las características de la cara, tales como el centro de la pupila, la esquina interior de los ojos, la esquina exterior de los ojos, entre otras, posteriormente, el programa procedía a calcular las distancias entre estos puntos, para finalmente ser comparados con una base de datos, este proyecto se llamó man-machine (hombre máquina) debido a la manualidad de operar el mismo (“Facial recognition system”, s.f).

Años más tarde, en los 70, Goldstein, Harmon, & Lesk, usaron 21 características específicas como color de la cabellera y el grueso de los labios para automatizar el reconocimiento facial (NSTC Subcomité de biometría, 2006, p. 1). Las mediciones y las coordenadas necesarias eran computados manualmente, haciendo que el requiriera de mucho tiempo de trabajo.

En el año 1988, Kirby y Sirobich, aplicaron la técnica de análisis de componentes principales, usualmente referida a la técnica de Eigen faces, su principal ventaja es que demostró que los datos podrían ser reducidos en una escala de 1:1000 para identificar una persona (NSTC Subcomité de biometría, 2006, p. 2).

En el año 1991, Turk y Pentland, descubrieron por medio de las técnicas Eigen faces, que el error excedente podría utilizarse para detectar caras en imágenes. Este descubrimiento permitió sistemas automatizados en tiempo real de reconocimiento facial de manera confiable. Aunque esta técnica estaba limitada por el entorno ambiental, creó un gran interés en promover el desarrollo de tecnologías de reconocimiento facial automatizado (NSTC Subcomité de biometría, 2006, p. 1).

En el año 1993, el programa de la evaluación de tecnología de reconocimiento facial (Face Recognition Technology Evaluation, FERET) fue patrocinado por la Agencia de Investigación de Productos Avanzados de Defensa de los Estados Unidos (Defense Advanced Research Products Agency, DARPA) de 1993 a 1997. Promovió el desarrollo de algoritmos de reconocimiento facial y tecnología mediante la evaluación de los prototipos de sistemas de reconocimiento facial e impulsó el reconocimiento facial a un mercado de productos comerciales (NSTC Subcomité de biometría, 2006, p. 4).

En el año 2000, se comenzaron a realizar pruebas a vendedores de reconocimiento facial (The Face Recognition Vendor Tests, FRVT), las cuales consistieron de 2 componentes: la prueba de rendimiento de reconocimiento y test de usabilidad de productos. El objetivo de la prueba de rendimiento de reconocimiento fue comparar competitivamente las técnicas para reconocimiento facial. Todos los sistemas fueron probados con una base de datos estandarizada. Este estándar aseguraba que todos los sistemas fueran evaluados usando las mismas imágenes, lo cual permitió realizar la comparación la tecnología de reconocimiento facial. La prueba de usabilidad del producto examinó las propiedades de los sistemas para la realización de control de acceso ("Face Recognition Vendor Test", 2016).

2.3.2.1 FVRT 2002

El FVRT en 2002 consistió de 2 pruebas: Alta intensidad computacional (the High Computational Intensity, HCInt) y la prueba de media intensidad computacional (Medium Computational Intensity, MCInt). Ambas pruebas exigían que los sistemas fueran automáticos, la intervención manual no estaba permitida. Los participantes podrían elegir entre los dos tipos de pruebas HCInt o MCInt ("Face Recognition Vendor Test", 2016).

La prueba de alta intensidad computacional (HCInt) fue diseñada para sistemas de vanguardia en imágenes real extremadamente desafiantes. Se trataban de imágenes

frontales fijas. Esta prueba compara las imágenes fijas de la base de datos contra las imágenes fijas de una persona desconocida. Los participantes en HCInt requerían procesar un conjunto de imágenes de aproximadamente 121.000, detectando todas las coincidencias de pares de imágenes posibles del conjunto de imágenes de 121.000. Esto requiere la realización de 15 mil millones de coincidencias en 242 horas. Los resultados del rendimiento de los sistemas de medida HCInt de reconocimientos faciales en grandes bases de datos, examinan el efecto del tamaño de la base de datos sobre el rendimiento y la estimación de la variabilidad en el rendimiento del sistema.

La prueba de mediana intensidad computacional (MCInt) constaba de dos partes separadas: imagen fija y video. MCInt fue diseñado para proporcionar una comprensión de la capacidad de un participante para llevar a cabo tareas de reconocimiento facial con diferentes formatos de imágenes (fijas y de video) bajo condiciones variables. La porción fija de la MCInt es similar a las evaluaciones FERET y FRVT 2000. Se comparó una base de datos de imágenes fijas contra las imágenes fijas de personas desconocidas. La parte fija de la MCInt fue diseñada para medir el rendimiento en diferentes categorías de imágenes. Diferentes efectos que se midieron fueron el tiempo entre las imágenes, los cambios en la iluminación, y la variación en pose. La parte de vídeo ha sido diseñada para proporcionar una evaluación inicial, para determinar si el vídeo ayuda o no a aumentar el rendimiento de reconocimiento facial ("Face Recognition Vendor Test (FRVT) 2002", 2016).

2.3.2.2 FRVT 2006

El objetivo principal de la FRVT 2006 fue medir el progreso de los sistemas prototipo o algoritmos y sistemas de reconocimiento facial comerciales desde el FRVT del 2002. En el FRVT del 2006 el rendimiento evaluado se hizo sobre ("Face Recognition Vendor Test 2006", 2016):

- Imágenes fijas de alta resolución (5 a 6 mega-píxeles).
- Escaneos faciales 3D.
- Multi-muestra de imágenes faciales fijas.
- Pre procesamiento algoritmos que compensan la pose y la iluminación.

Para garantizar una evaluación precisa, el rendimiento medido FRVT 2006 se realizó con datos secuestrados (datos no vistos con anterioridad por los investigadores o desarrolladores). Un conjunto de datos y metodología de prueba estándar se empleó para que todos los participantes fueron evaluados de manera uniforme. El gobierno proporcionó tanto los datos de prueba y el entorno de prueba a los participantes. El entorno de prueba se llama el Ambiente Biométrico de Experimentación (Biometric Experimentation Environment.

BEE). El BEE era la infraestructura del FRVT 2006. Esto permitió que el experimentador se centrase en el experimento mediante la simplificación de la gestión de los datos de prueba, la configuración del experimento, y el procesamiento de los resultados. El FRVT 2006 fue patrocinado por varias agencias del gobierno de los Estado Unidos y se llevó a cabo y gestionado por el Instituto Nacional de Estándares y Tecnología (NIST) ("Face Recognition Vendor Test 2006", 2016).

Uno de los objetivos de la FRVT 2006 fue determinar de forma independiente si se han alcanzado los objetivos del Gran Desafío de Reconocimiento Facial (Face Recognition Grand Challenge, FRGC). El FRGC fue un proyecto separado, sobre el desarrollo de algoritmos diseñado para promover y avanzar en la tecnología de reconocimiento facial, con los esfuerzos de reconocimiento facial en el gobierno de los Estado Unidos. Uno de los objetivos de FRGC fue el desarrollo de algoritmos de reconocimiento facial capaces de un rendimiento mejor que FRVT 2002. El FRGC se llevó a cabo entre mayo de 2004 y marzo de 2006. Los datos FRGC todavía están disponibles para hacer frente a los investigadores de reconocimiento. Para obtener datos del FRGC, los posibles participantes deben firmar las licencias necesarias y seguir las reglas de liberación de datos del FRGC ("Face Recognition Vendor Test 2006", 2016).

El FRVT 2006 realizó un reporte con los resultados obtenidos, donde recibieron algoritmos de 22 organizaciones en 10 países diferentes, donde se subieron múltiples algoritmos. Sin embargo, solo aquellos que completaron satisfactoriamente las pruebas de gran escala se documentaron en el reporte. Algunas de las organizaciones que participaron son Animetrics Inc., Identix Inc., entre otros ("Face Recognition Vendor Test 2006", 2016).

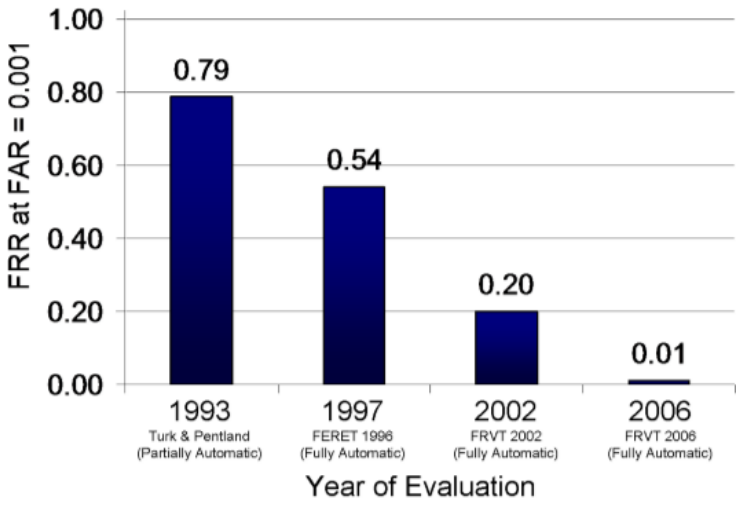


Figura 3: FRVT del estado de la técnica a través de los distintos FRVTs (Phillips et al., 2007)

De la Figura 3 se puede observar que, los resultados de 1993 y 1997 están en el mismo conjunto de datos de prueba y muestran una mejora en la tecnología de los algoritmos en el marco del programa de FERET. La tecnología mejoró, de algoritmos parcialmente automáticos a algoritmos completamente automáticos, mientras que la tasa de error se redujo en aproximadamente un tercio. La mejora en el rendimiento de los algoritmos entre FVRT 2002 y FRVT 2006 se debe a los avances en el diseño de algoritmos, sensores, y la comprensión de la importancia de corregir para variar la iluminación a través de imágenes (Phillips et al., 2007).

Por lo tanto, el programa FERET y FRVT han ayudado a buscar una reducción en tasa del falso negativo (False Rejection Rate o FRR) y la tasa del falso positivo (False Acceptance Rate o FAR) en el campo de la biometría de reconocimiento facial. Una de las claves para lograr esta rápida reducción en esos 13 años, fue el patrocinio del gobierno de los Estado Unidos para las evaluaciones y desafíos planteados en estos programas (Phillips et al., 2007).

Posteriormente en 2010 no se llevó a cabo un programa FRVT, sino que se realizó un programa equivalente llamado still image track of Multiple Biometrics Evaluation MBE 2010, con un objetivo principal de identificar las limitaciones máximas de almacenamiento en las que los sistemas de verificación podrán funcionar bien.

2.3.2.3 FRVT 2013

En 2013 se plantea el FRVT 2013 con el objetivo principal de medir los avances en las destrezas de los sistemas de prototipo y algoritmos de las comunidades académicas y comerciales. El FRVT 2013 tiene como objetivo evaluar el desempeño en varias tareas de reconocimiento, incluyendo ("Face Recognition Vendor Test (FRVT) 2013", 2012):

- Identificación uno-a-muchos de un conjunto de imágenes de tipo policial.
- Uno-a-uno verificación de imágenes de tipo visa.
- Multi-muestra de imágenes faciales fijas.
- El reconocimiento de personas en las secuencias de vídeo.
- Gemelos.
- 90 grados perfil visión.
- La estimación de género (M | F).
- Estimación de la edad (en años).
- Estimación de pose.
- La neutralidad de la expresión

2.3.3 Estandarización

La estandarización es trascendental para avanzar en el estado del arte y a nivel de mercado, por lo que se ha trabajado mucho en los estándares nacionales e internacionales para facilitar la operación entre productos y los formatos de intercambio de datos, lo que lleva a facilitar el avance de la tecnología en un ámbito estandarizado. Los más altos estándares en el área son las normas ANSI/INCITS (M1) 385-2004 e ISO 19794-5 de formato de intercambio de datos de reconocimiento facial, y direcciona a la examinación de imágenes faciales, la verificación de la identidad humana, la identificación de la cara y la verificación automatizada. Estas normas permiten la interoperabilidad entre los vendedores de sistemas de reconocimiento facial. Se está trabajando en los planos nacional e internacional para actualizar las normas para los datos faciales en 3D. La versión de ANSI NIST ITL 1 es la 2011 actualizada al año 2015, se usa en el ámbito internacional para proporcionar orientación en el intercambio automatizado biométrico. Estas normas también facilitan el uso de la información de la cara en las aplicaciones que tienen de almacenamiento limitada (por ejemplo, pasaportes, visados, permisos de conducir). Otras normas, como INCITS 398-2005 marco de formatos de intercambio común (CBEFF), tratan específicamente de los elementos de datos que se utilizan para describir los datos biométricos de una manera común. La Especificación BioAPI INCITS 358-2002 define la interfaz de programación de aplicaciones y la interfaz de proveedor de servicio para una interfaz estándar de tecnología biométrica. Las organizaciones internacionales y nacionales de estandarización (e.g. ISO/IEC/JTC1/SC37 Biometrics) continúan trabajando en las normas en una dirección que facilite el crecimiento, el progreso y la interoperabilidad ("Biometría - Facial", 2016).

2.3.4 Aplicaciones

Uno de los hechos que más se habla es el del SuperBowl del año 2000, donde el departamento de policía de Tampa, probó por primera vez su sistema de reconocimiento facial llamado Faceit, logrando identificar a 19 personas dentro de la multitud que tenían órdenes pendientes. Este hecho despertó el interés en demás personas que querían usarlas para sus sistemas de vigilancia. A continuación, se presentan los usos que se le ha dado a la biometría de reconocimiento facial (Rogers, 2016):

- Aplicación de la Ley y la Seguridad Nacional: En Estado Unidos el FBI usa en su Programa de Identificación de Próxima Generación (Next Generation Identification Program, NGI) una variedad de técnicas biométricas, entre ellas incluye el reconocimiento facial, para la identificación y monitorización de personas de interés ("Automated Facial Recognition in the Public and Private Sectors", 2016).

- Seguridad fronteriza: En Australia, el reconocimiento facial se utiliza en conjunto con las huellas dactilares en las fronteras para identificar si algún sujeto está aplicando con una visa fraudulenta. Los funcionarios de inmigración de Australia también están usando el reconocimiento facial en los esfuerzos para abordar el fraude de visa. Esto es parte de una campaña nacional para acabar con el robo de identidad y el uso de identidades falsas para facilitar la delincuencia ("Automated Facial Recognition in the Public and Private Sectors", 2016).
- Licencias de conducir: Las licencias de conducir en muchas provincias canadienses, como Ontario, Columbia Británica y Manitoba, el reconocimiento facial se utiliza durante el proceso de solicitud de licencia para detectar el robo de identidad y el fraude, así como la identificación de las personas que aplican con diferentes nombres ("Automated Facial Recognition in the Public and Private Sectors", 2016).
- Casinos: Se suele usar para identificar aquellos jugadores que por algún motivo se les ha prohibido la entrada, como por ejemplo los adictos al juego. El software de reconocimiento facial se ejecuta sobre las personas que entran al edificio y compara las imágenes con una base de datos de jugadores que han sido solicitados colocar en un listado de personas prohibidas para los juegos de azar ("Automated Facial Recognition in the Public and Private Sectors", 2016).
- Drones: los drones son cada vez más utilizados por la policía y las agencias gubernamentales internacionalmente. En los EE.UU., habrá un máximo de 30.000 drones a finales de esta década. De acuerdo con un informe del Servicio de Investigación del Congreso de Estados Unidos de 2013, "En un futuro próximo, las organizaciones policiales podrían tratar de equipar aviones no tripulados con el reconocimiento facial o de reconocimiento biométrico suave, que pueden reconocer y realizar un seguimiento de individuos basados en atributos tales como la altura, la edad, el género, y color de la piel ... y pronto tendrá la capacidad de ver a través de paredes y techos." ("Automated Facial Recognition in the Public and Private Sectors", 2016).
- Aplicaciones militares: Según los informes, la Marina de los Estados Unidos ha adoptado el uso de gafas de estilo Robocop, equipadas con una pequeña cámara que puede ver hasta 19 kilómetros. Las gafas pueden capturar 400 imágenes por segundo y compararlas con una base de datos central de 13 millones de rostros. También se trabaja en una cámara integrada al arma de un soldado, junto con una base de datos portátil que podría contener más de un millón de imágenes, lo que permitiría en cuestión de segundos identificar a terroristas sin recurrir a otras conexiones ("Automated Facial Recognition in the Public and Private Sectors", 2016).

- Eventos deportivos: En los juegos olímpicos de China de 2008, se utilizó para someter a todos los que entraran al estadio principal a controles de identidad de reconocimiento facial. Igualmente se hizo para los olímpicos de Londres en 2012 ("Automated Facial Recognition in the Public and Private Sectors", 2016).

2.3.5 Librerías de reconocimiento facial

Actualmente en el mercado existen una variedad de librerías que permiten usar algoritmos ya listos para acoplar a un desarrollo específico de software, algunas son de pago y otras son de licencia libre, de acuerdo a esto veamos las siguientes librerías que se pueden encontrar:

Este listado es tomado de datasciencecentral.com ("50 Face Recognition APIs", 2016)

1. Face Recognition - de Lambda Labs. Este API proporciona el reconocimiento de rostros, detección facial, la posición de los ojos, la nariz posición, posición de la boca, y la clasificación de género.
2. Face (Detection) - API de visión por ordenador para el reconocimiento facial y de detección facial. Actualmente tienen un API libre para la detección de rostros.
3. Animetrics Face Recognition - La API Animetrics de reconocimiento facial se puede utilizar para detectar caras humanas en imágenes. La información sobre los rasgos faciales o "puntos de referencia" se devuelve como coordenadas en la imagen. También detectar y devolver la orientación, o "pose" de caras a lo largo de 3 ejes.
4. Skybiometry Face Detection and Recognition - fácil de utilizar esta API de detección facial y reconocimiento. Se debe tener una aplicación creada en la cuenta de SkyBiometry para utilizarla.
5. ImageVision Face Detection - ImageVision es una empresa de tecnología de visión por ordenador que determina las ubicaciones y tamaños de caras humanas en imágenes arbitrarias (digitales).
6. Face++ - utiliza la tecnología de vanguardia de la visión por ordenador y la minería de datos para proporcionar 3 servicios de visión centrales (detección, análisis y reconocimiento). Esta API proporciona la detección y el análisis de Landmark (23points), Punto de referencia (81 puntos), Atributos: edad, sexo, Gafas, raza, etc.
7. FaceMark - FaceMark es una potente API para la detección de características faciales. Es capaz de detectar 68 puntos para una cara frontal y 35 para un perfil. FaceMark detecta puntos de referencia para las caras en la imagen especificada por el URL o cargados como un archivo y genera una salida JSON que contiene un vector de puntos de referencia faciales y orientación para cada rostro descubierto.

8. EmoVu por Eyeris - software de reconocimiento de la emoción por Inteligencia Artificial, que permite determinar expresiones faciales, sexo y edad de personas.
9. Face and scene recognition by Rekognition.com - Este motor de reconocimiento, rápido, robusto y escalable puede hacer detección facial, reconocimiento facial, y comprensión de una escena. Puede ser entrenado de forma automática con el uso de imágenes y etiquetas en Facebook.
10. FaceRect - es una API potente y libre para la detección de rostros. Encuentra rostros (tanto frontales como de perfil) en una imagen especificada por URL o cargando un archivo, es capaz de encontrar múltiples caras en una sola foto, produciendo la salida JSON con un cuadro de límite para cada rostro descubierto.
11. Infatics Face Detection - API de simple detección de caras.
12. OpenCV Face Recognizer - OpenCV (Open Source Computer Vision Library: es una librería de licencia BSD de código abierto que incluye muchos algoritmos de visión por ordenador.

Para este trabajo se selecciona la librería OpenCV por su amplia gama de algoritmos y además ofrece una forma sencilla de implementación, así como también la reputación que tiene la misma. Esta librería se usa mucho en trabajos de tesis, y ejemplos en internet también por su licencia BSD de código abierto.

2.3.6 OpenCV

Como bien se ha mencionado anteriormente, para este trabajo se ha escogido la librería OpenCV, la cual fue construida para proporcionar una plataforma común para aplicaciones de visión por ordenador y para acelerar el desarrollo de las mismas. Cuenta con más de 2500 algoritmos optimizados entre los cuales se pueden encontrar algoritmos de visión de por ordenador y algoritmos de aprendizaje automático. Este conjunto de algoritmos puede ser usados para detección y reconocimiento de rostros, identificar objetos, clasificar acciones humanas en videos, rastreo de movimiento en videos, rastreo de movimiento de objetos, extracción de modelos de 3D de objetos, entre otras bondades. Esta librería es usada en muchas compañías, grupos de investigación y cuerpos gubernamentales.

El equipo desarrollador es Itseez, la cual recientemente ha sido adquirida por Intel. La comunidad de usuarios está constituida por más de 47 mil personas y más 7 millones de descargas. OpenCV está escrita de forma nativa en C ++, tiene interfaces de C++, C, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. ("ABOUT | OpenCV", 2016)

2.4 Algoritmos

En este apartado se enfocará básicamente en describir los algoritmos usados en este trabajo para la detección y reconocimiento de rostros. Estos dos conceptos son diferentes, ya que mientras la detección se encarga de extraer el rostro de una imagen completa, el reconocimiento se encarga de realizar el todo el proceso de cómo establecer que un rostro corresponde a un individuo. Existen diversos algoritmos, sin embargo, se explicarán aquellos que ayudaron a la construcción de este trabajo.

2.4.1 Esquema general y elección de algoritmos

Un esquema general que puede resumir el proceso de autenticación incluyendo los macro procesos más importantes puede ser el que se muestra en la Figura 4:

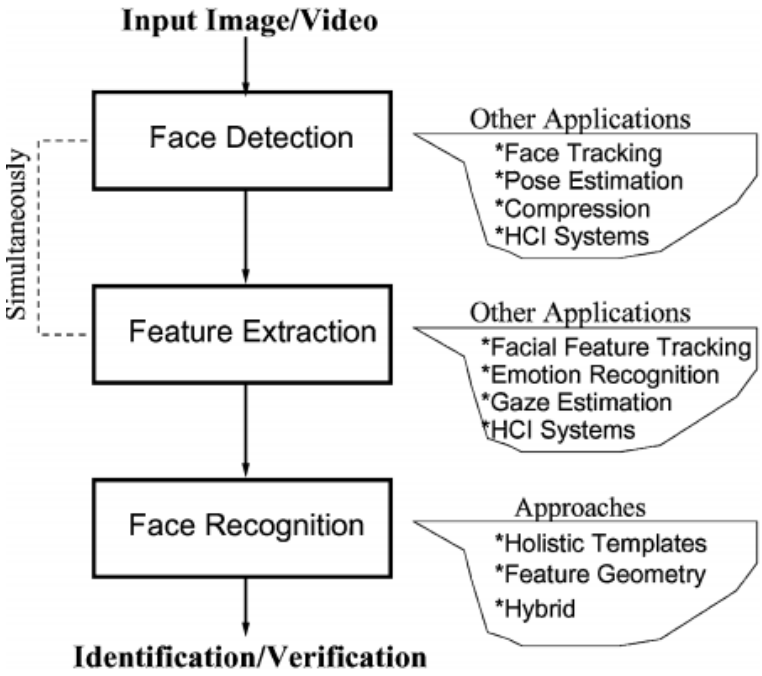


Figura 4: Esquema general del proceso de reconocimiento facial (Zhao et al., 2003)

La librería OpenCV posee un algoritmo para la detección facial llamado “Haar Cascades” y los siguientes algoritmos para el reconocimiento facial ("Face Recognition with OpenCV — OpenCV 2.4.13.0 documentation", 2016):

- EigenFaces
- Fisherfaces
- Local Binary Patterns Histograms

De estos 3 algoritmos, en la sección de “Resultados” se muestra cómo se eligió el mejor, con el objetivo de tener uno solo y el mejor a la hora de realizar la autenticación del usuario a través del dispositivo móvil.

En OpenCV, en común, estos algoritmos trabajan sobre una clase base llamada FaceRecognizer, la cual permite principalmente 2 funciones:

- Crear un modelo basado en el entrenamiento del algoritmo a partir de un conjunto de imágenes.
- Predecir a partir de una imagen de entrada el id correspondiente del modelo creado que se menciona en el punto anterior, y el nivel de confianza, el cual servirá para determinar si es posible con este valor, un umbral de acierto.

2.4.2 Algoritmo de detección de rostro “Haar Cascades”

El algoritmo de detección de rostro ("OpenCV: Face Detection using Haar Cascades", 2016) está basado en el algoritmo de detección de objetos publicado en el artículo "Rapid Object Detection using a Boosted Cascade of Simple Features" (Detección Rápida de objetos utilizando una cascada impulsada por características simples) realizado por Paul Viola y Michael Jones, publicado en el año 2001, en el cual proponen un método efectivo y rápido de detección de objetos usando clasificadores en cascada basados en características Haar. Se trata de una aproximación de aprendizaje automático donde una función en cascada es entrenada a partir de muchas imágenes positivas (imágenes con el objeto a detectar) e imágenes negativas (imágenes sin el objeto).

OpenCV cuenta con un entrenador y detector, dando espacio para que el desarrollador también pueda entrenar su propio clasificador para detectar otros objetos como carros, aviones, etc. También cuenta con clasificadores pre-entrenados para rostro, ojos, sonrisa, etc.

Las principales funciones para detección son:

- CascadeClassifier(XML del clasificador pre-entrenado)
- detectMultiScale(Imagen sobre la que se realiza la detección)

2.4.3 Eigenfaces

Este algoritmo se basa en el Análisis de Componentes Principales en inglés Principal Component Analysis (PCA), el cual fue propuesto por Karl Pearson (1901) y Harold Hotelling (1933) para convertir un conjunto de variables correlacionadas en un conjunto más pequeño de variables no correlacionadas. La idea es, que un conjunto de datos de alta dimensión, es a menudo descrito por variables correlacionadas y por lo tanto sólo unas pocas dimensiones significativas representan la mayor parte de la información. El método PCA encuentra las direcciones con la mayor variación en los datos, llamadas componentes principales.

En la Figura 5 se muestra identificados por color los grupos de valores de escala de grises distribuidos dentro Eigenfaces específicos. En la primera fila imágenes 4 y 5 se puede ver la luz a la izquierda y derecha respectivamente.

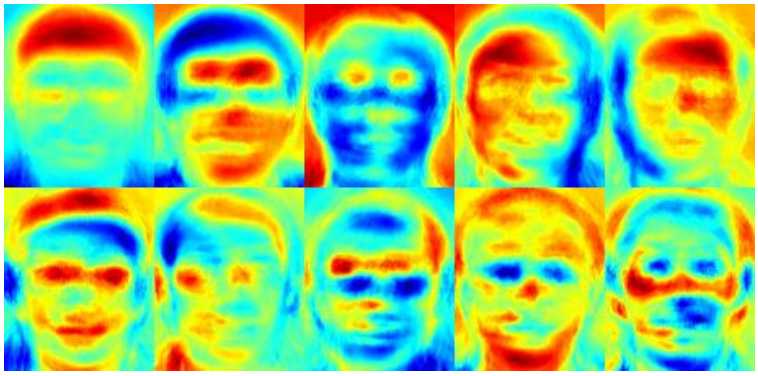


Figura 5: Resultados de agrupación de valores en algoritmo Eigenfaces (OpenCV)

2.4.4 Fisherfaces

En el algoritmo anterior de Eigenfaces, el cual se basa en el uso del método de Análisis de Componentes Principales PCA, presenta un inconveniente. En el evento en que la varianza de los datos se dé por una fuente externa como la luz, los componentes identificados por un PCA no contendrán necesariamente información discriminada alguna, y por consiguiente las muestras proyectadas se mezclen y la clasificación resulte imposible.

Es aquí donde entra el método Análisis Discriminante Lineal en inglés Linear Discriminant Analysis (LDA) propuesto por el estadístico y biólogo Sir R. A. Fisher, el cual usó para la clasificación de flores en su artículo “The use of multiple measurements in taxonomic problems”, 1936. El método consiste básicamente en maximizar la relación entre clases dentro de clases de dispersión, en vez de maximizar toda la dispersión global. Esto quiere decir que las clases iguales deben agruparse estrechamente, mientras clases diferentes están lo más lejos posible una de la otra en la representación lineal inferior.

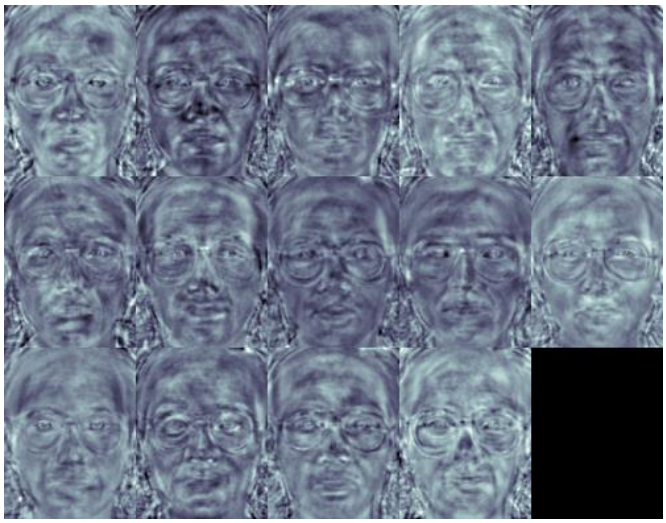


Figura 6: Ejemplos de aplicar algoritmo Fisherface (OpenCV)

En la Figura 7 podemos observar la comparación de los algoritmos anteriores Eigenfaces y Fisherfaces con respecto a la precisión de reconocimiento. A medida que aumentan el número de imágenes por persona, los algoritmos de Eigenfaces y Fisherfaces aumentan la precisión de reconocimiento:

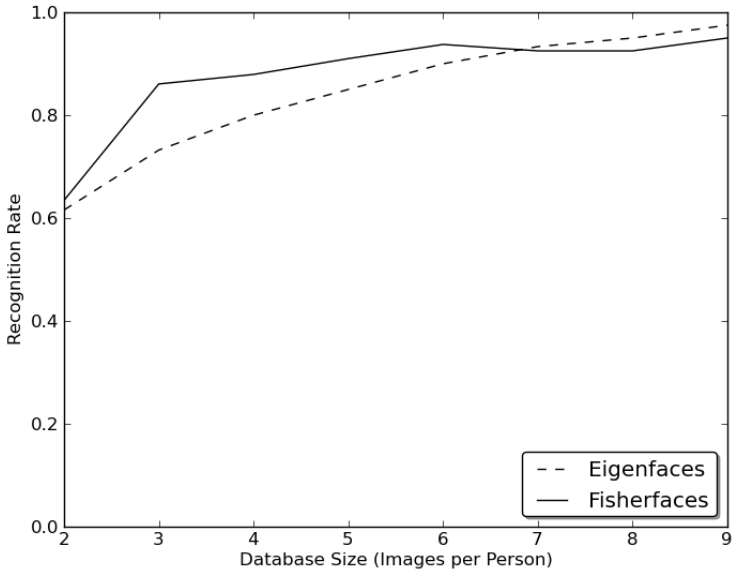


Figura 7: Eigenfaces vs Fisherfaces precisión de reconocimiento (OpenCV)

2.4.5 Local Binary Patterns Histograms

En la vida real los escenarios pueden variar mucho, no se puede garantizar por ejemplo que las imágenes queden con luz perfecta o siempre se puedan tomar 10 diferentes imágenes

de una persona, donde los algoritmos anteriores Eigenfaces y Fisherfaces no se comportan bien en estos escenarios.

La idea del algoritmo de Local Binary Patterns Histograms es la extracción de características locales de las imágenes, se trata de no mirar toda la imagen como un vector de alta dimensión, sino describir solo características locales de un objeto. Estas características tendrán una baja dimensionalidad implícita. Básicamente el método de Local Binary Patterns resume la estructura local en una imagen comparando cada pixel con su vecino, luego tomar un pixel como su centro y el umbral de sus vecinos en contra. Si la intensidad del pixel central es mayo igual que la de su vecino, se denotará con 1 de lo contrario con 0.

En la Figura 8 se puede observar la aplicación del operador LBP a imágenes con transformaciones artificiales monotónicas en escala de grises

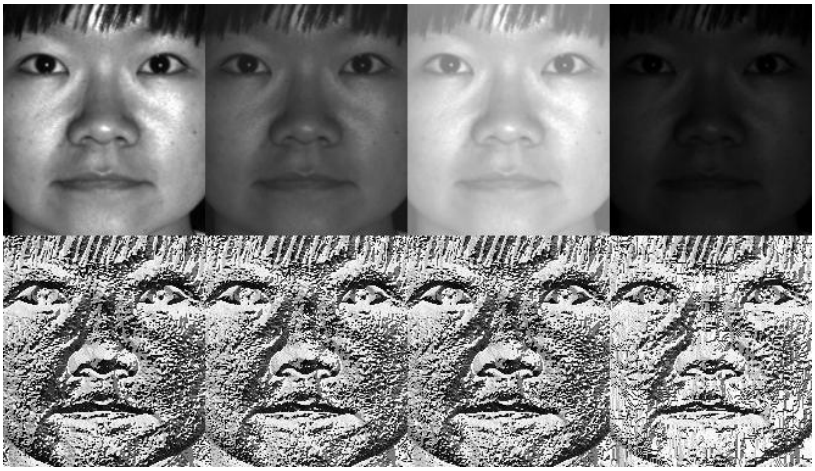


Figura 8: Aplicación del operador LBP (OpenCV)

2.4.6 Identificación y verificación

La importancia de tener claro identificación y verificación, es vital en este trabajo, ya que esto define el modo en que deba realizarse el proceso de autenticación de un usuario ("Explainer: Verification vs. Identification Systems", 2012).

La verificación busca responder a la pregunta ¿es esta persona quien dice ser?, el proceso básicamente consiste en que la persona se presente como una persona específica, y el sistema biométrico compare su dato biométrico contra el perfil biométrico de la persona específica quien dice ser, esto es una comparación de 1 a 1.

La identificación por el contrario busca responder la pregunta ¿quién es esta persona?, o, ¿a quién hace referencia este dato biométrico?, en este caso, la persona solo presenta su dato biométrico y el sistema realiza la comparación 1 a n, donde n es el número de sujetos con datos biométricos en el sistema.

La ventaja con la verificación es que el proceso es más rápido y más precioso por ser una comparación 1 a 1 y no 1 a n, sin importar que la base de datos se incremente. Desde luego en este trabajo se usará el método de verificación. En la Figura 9 se puede observar a la izquierda el proceso de verificación y a la derecha el proceso de identificación.

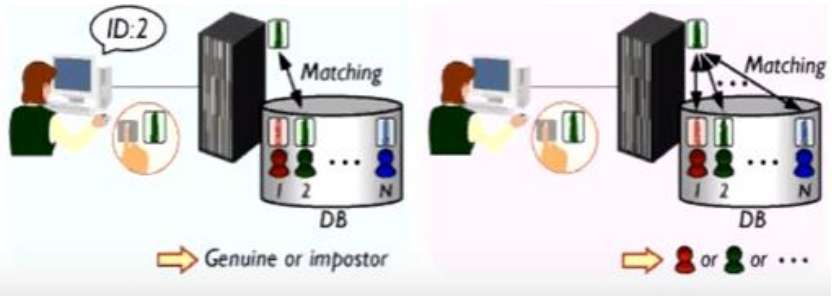


Figura 9: Verificación (izquierda) vs Identificación (derecha) (advancedsourcode, 2013)

Un dato importante, es que la implementación utilizada (OpenCV) de los algoritmos de Eigenfaces y LBPH permiten entrenar el modelo con un conjunto de imágenes de un solo individuo, al contrario, el algoritmo de Fisherfaces, permite a partir de un conjunto de imágenes de 2 individuos.

2.5 Extendiendo la funcionalidad de PHP

El lenguaje para desarrollar el servidor biométrico escogido es PHP, pero uno de los desafíos encontrados para desarrollar este trabajo, es el de la librería OpenCV, ya que esta no tiene nativamente una extensión para PHP, dificultando la integración del servidor biométrico con esta librería.

Sin embargo, hay una pregunta que se debe resolver, ¿por qué es necesario crear una extensión para php?, si, fácilmente se podría pensar, en realizar un llamado por consola a un programa ejecutable que brinde las operaciones requeridas de OpenCV. La respuesta es que pensando en términos de rendimiento no sería óptimo, ya que tendría un consumo más alto de memoria y CPU, dado que el sistema tendría por cada llamado que asignar un nuevo espacio de memoria. En contraste, la extensión de php dll permanece en memoria, haciendo que la página que realice el llamado sea más rápida y los recursos de hardware sean consumidos de una forma más óptima.

Dentro las opciones para realizar la integración, se optó por usar la API Zend de PHP ("PHP: La API Zend: Hackeando el núcleo de PHP - Manual", 2016), existen otras opciones para construir wrappers como SWIG ("SWIG and PHP", 2016) y PHP-CPP ("PHP-CPP - A C++ library for developing PHP extensions", 2016). La decisión de realizar el desarrollo directamente sobre la API Zend, fue solo en que el momento de realizar las pruebas de desarrollo funcionó bien, mientras que con las librerías de construcción de wrappers el

desarrollo no se pudo completar, más por desconocimiento de programación, igualmente desarrollar en la API Zend fue complejo, realmente casi no hay información al respecto.

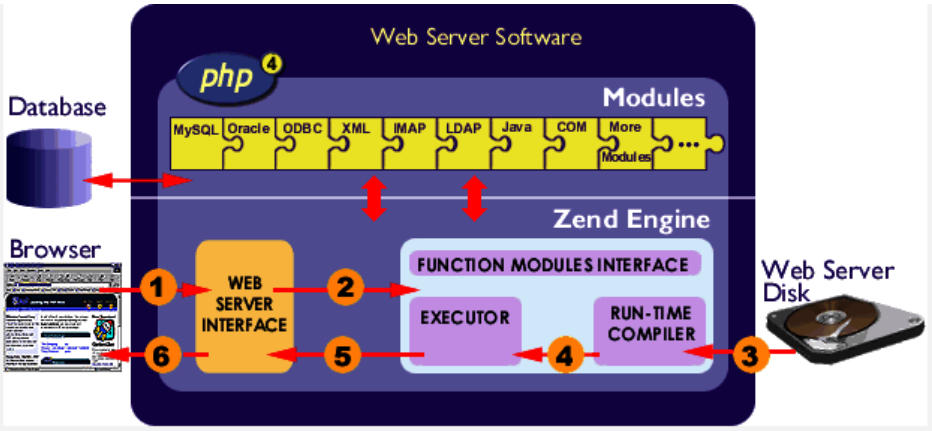


Figura 10: Arquitectura de PHP

2.6 Servicios Web: REST

Es importante conocer el modo en que se realiza la comunicación de los dispositivos móviles con el servidor biométrico. En un inicio se tienen 2 opciones, uso de arquitectura REST o SOAP Web Services, sin embargo, para este trabajo se implementa la comunicación a través de REST.

2.6.1 ¿Qué es REST?

REST (Representational State Transfer) es un tipo de arquitectura de desarrollo web, apoyada en el estándar HTTP, la cual permite crear servicios y aplicaciones que puedan ser usadas por cualquier cliente por medio de HTTP. ("Conceptos sobre APIs REST", 2016)

Consta de los siguientes principios (Navarro, 2006):

- Escalabilidad de la interacción con los componentes
- Generalidad de interfaces
- Puesta en funcionamiento independiente
- Compatibilidad con componentes intermedios

Para cumplir estos principios, cuenta con 3 niveles de calidad ("Conceptos sobre APIs REST", 2016):

- Uso correcto de URIs (Uniform Resource Identifier): Cada página, información en una sección, o archivo, en términos de REST se nombran como recursos. Estos recursos son identificados de forma única por medio de las URLs (Uniform Resource Locator).
- Uso correcto de HTTP: el desarrollador debe dominar aspectos claves como:

  - Métodos HTTP.
  - Códigos de estado.
  - Aceptación de tipos de contenido.
- Implementar Hypermedia: su objetivo es conectar mediante vínculos las aplicaciones clientes con las APIs, así los clientes no tienen que pensar cómo acceder a los recursos.

2.6.2 ¿Por qué REST?

El motivo principal por lo que REST es más viable para implementar en este desarrollo de software, es que REST es más simple y convencional que SOAP Web Services, no requiere configuraciones especiales, como si lo requiere SOAP para suplir cualquier necesidad de comunicación, es una potencia que no se requiere utilizar en este contexto.

3 Objetivos concretos y metodología de trabajo

A partir de la descripción del problema, del contexto y el estado del arte, se establecen los siguientes objetivos.

3.1 Objetivo general

Desarrollar un software que utilice reconocimiento facial diseñado para la autenticación de inicio de sesión de Windows a través de dispositivos móviles como puente entre el terminal del usuario y el sistema central de Directorio Activo.

3.2 Objetivos específicos

- Diseñar y desarrollar el proceso macro y subsistemas que definirán toda la operación del software de reconocimiento facial.
- Elegir el conjunto de algoritmos y herramientas de desarrollo dentro del campo de visión artificial, de modo que permitan el análisis, detección y reconocimiento de rostros en tiempo razonable, junto con la obtención de datos de respuesta para el usuario final que estén acordes con las condiciones presentadas en el escenario de interacción en el mundo real.
- Diseñar y desarrollar un proceso de registro del usuario por medio del dispositivo móvil en el software de reconocimiento facial.
- Establecer un mecanismo propio para permitir integrar otras librerías de reconocimiento facial de modo que sea en lo más posible transparente para los desarrollos implementados.

- Comparar el funcionamiento con los 3 algoritmos de reconocimiento facial planteados anteriormente, de forma que se elija el más adecuado para el desarrollo del software.

3.3 Metodología del trabajo

Se definieron dos etapas para la metodología del trabajo, de modo que ayudaran a lograr el desarrollo de software planteado.

En la primera etapa se definieron aquellos pasos iniciales que permitieran adquirir el conocimiento técnico y conceptual para el desarrollo de los diferentes componentes. Cada tarea se desarrolló independiente una de la otra en diferentes lenguajes pensando siempre en el acople final de los diferentes desarrollos, además también pensando siempre en que el equipo donde se realizaría este desarrollo, es un equipo promedio del común, con características usuales de los equipos de hoy en día, no tipo servidor. Las tareas iniciales serían las siguientes:

- Lograr desarrollar el reconocimiento facial en lenguaje C++, dado que OpenCV está para C++. Se realizarían pruebas con el objetivo de obtener resultados de cada algoritmo.
- Desarrollar aplicación en Android donde consista en primera medida acceder a la cámara del celular y posteriormente enviar una serie de imágenes.
- Preparar ambientes, servidor LDAP Windows y servidor web donde estaría la interface con la aplicación móvil.
- Desarrollar página web que será el servidor biométrico, diseño de las operaciones que realizaría en LDAP, y las operaciones con la aplicación móvil.
- Acoplar el desarrollo en C++ de reconocimiento facial con la página web del servidor biométrico.
- Acoplar la aplicación Android con la página web del servidor biométrico.

En la segunda etapa se analizan los diferentes resultados obtenidos de cada paso anterior, de modo que se pudiera llevar a cabo los pasos que permitieran el desarrollo final del software de reconocimiento facial. Para esto se siguieron las siguientes pautas:

- De acuerdo a los resultados, elegir el mejor algoritmo de reconocimiento facial que haya realizado el mejor match facial de acuerdo a la base de datos planteada.
- Definir las funciones que se implementarán en cada parte del software, que llevarán al desarrollo final del software de reconocimiento facial.
- Realizar las diferentes pruebas necesarias para determinar el comportamiento del software desarrollado.

4 Desarrollo específico de la contribución

4.1 Identificación de requisitos

El objetivo de esta actividad de análisis es identificar los requisitos surgidos a raíz de la delimitación del entorno y del alcance, clasificar tales requisitos en requisitos funcionales y no funcionales.

Dentro del conjunto de requisitos no-funcionales, se encuentran aquellos requisitos referentes al tipo de arquitectura física a usar en el desarrollo del software de reconocimiento facial, en donde se encuentra la plataforma de desarrollo y los lenguajes de programación escogidos, y los dispositivos de entrada y salida a usar. En la Tabla 2 se observan los requisitos no-funcionales detectados para este desarrollo.

En el conjunto de requisitos funcionales, se encuentran aquellos requisitos que definen las funciones y procedimientos a realizar dentro del software de reconocimiento facial que representarían su interfaz de comandos y operaciones que darían soporte al desarrollo de sistemas de reconocimiento facial por medio de dispositivos móviles. Dentro de los requisitos funcionales están expresadas las funciones o proceso principales, junto con la interfaz gráfica de usuario final. En la Tabla 3 se observan los requisitos funcionales detectados para este desarrollo.

Tabla 2: Requisitos No Funcionales del Software

Requisito No.	Tipo	Descripción
RNF1	Rendimiento	Los tiempos de respuesta de la autenticación deben estar dentro de un tiempo razonable
RNF2	Seguridad	Se deben implementar mecanismos básicos de seguridad en la comunicación como protocolo https y encriptación de datos sensibles
RNF3	Seguridad	El acceso al sistema biométrico debe tener un proceso de login y manejo de sesión
RNF4	Interfaz	La aplicación móvil de usuario final debe ser fácil de usar
RNF5	Interfaz	El servidor biométrico debe ser de fácil administración
RNF6	Interfaz	La comunicación entre la aplicación móvil y el servidor biométrico debe realizarse sobre REST debido a que sólo está planteado la comunicación entre estas dos aplicaciones

Tabla 3: Requisitos Funcionales del Software

Requisito No.	Descripción
RF1	La aplicación móvil debe permitir el registro del usuario en el servidor biométrico
RF2	La aplicación móvil debe permitir solicitar un token (password) para su autenticación en windows a través del rostro del usuario
RF3	El servidor biométrico debe ser capaz de validar de que el sujeto que solicite el token de autenticación sea el mismo que realizó el proceso de registro desde el móvil registrado
RF4	El servidor biométrico debe permitir registrar la información de los diferentes usuarios que tienen acceso al sistema
RF5	El servidor biométrico debe permitir realizar un proceso de registro biométrico del usuario (imágenes digitalizadas del usuario)
RF6	El servidor biométrico se debe contar con un mecanismo para que el sistema permita realizar el proceso de registro biométrico al usuario sólo si este posee un token de registro inicial
RF7	El servidor biométrico debe permitir la generación del archivo de conocimiento para el reconocimiento facial

4.2 Descripción de la herramienta software desarrollada

Para este apartado, se describirán los diferentes detalles del proceso, fases y diagramas que describen la herramienta desarrollada.

4.2.1 Hitos del proceso

El primer paso fue abarcar las tareas planteadas anteriormente en el apartado de metodología del trabajo, de esto, se obtuvo el conocimiento inicial para el desarrollo del software y la visualización de los diferentes desarrollos a realizar. El segundo paso fue definir aquellos hitos del proceso que permitieran lograr los objetivos planteados y así la finalización del desarrollo de software. Estos hitos fueron también los puntos más difíciles desarrollar por el alto grado de complejidad y por ende afectarían la finalización del desarrollo de software, así como el cumplimiento de objetivos y requisitos. Los hitos se describen en la Tabla 4:

Tabla 4: Hitos del proceso de desarrollo de Software

Hito No.	**Aplicación(es)**	**Descripción**
H1	Servidor Biométrico	Elegir el mejor algoritmo de reconocimiento facial y su nivel de aceptación
H2	Servidor Biométrico, Servidor LDAP Windows	Desarrollo de clase en php que permitiera realizar la comunicación y operación de cambio de contraseña a un usuario en el servidor LDAP Windows
H3	Servidor Biométrico	Desarrollar extensión dll para php, para poder hacer uso de las operaciones requeridas por el Software que se encuentran en la librería de OpenCV disponible en C++
H4	Móvil	Implementación del algoritmo de detección facial con OpenCV en aplicación móvil que permita la extracción de rostros
H5	Móvil, Servidor biométrico	Generación automática del archivo de índice de imágenes que requiere el proceso de generación de archivo de conocimiento
H6	Móvil, Servidor Biométrico, Servidor LDAP Windows	Incorporar protocolo https en toda la comunicación, diferentes desarrollos y configuraciones de entorno como certificados

4.2.2 Diagrama de arquitectura

El software desarrollado ha recibido el nombre de AFacial, la “A” por la palabra “Autenticación” y “Facial” porque es el medio por el cual se realiza la “Autenticación”. El diagrama de arquitectura que se presenta a continuación, se aclara que, no es el más óptimo y puede ser modificable, ya que el software se adapta a la arquitectura que se desee implementar. En este caso sólo se tiene un computador para correr todos los componentes de software necesarios, y en términos de usar los recursos del computador razonablemente como procesador y memoria, se implementa como se muestra en la Figura 11.

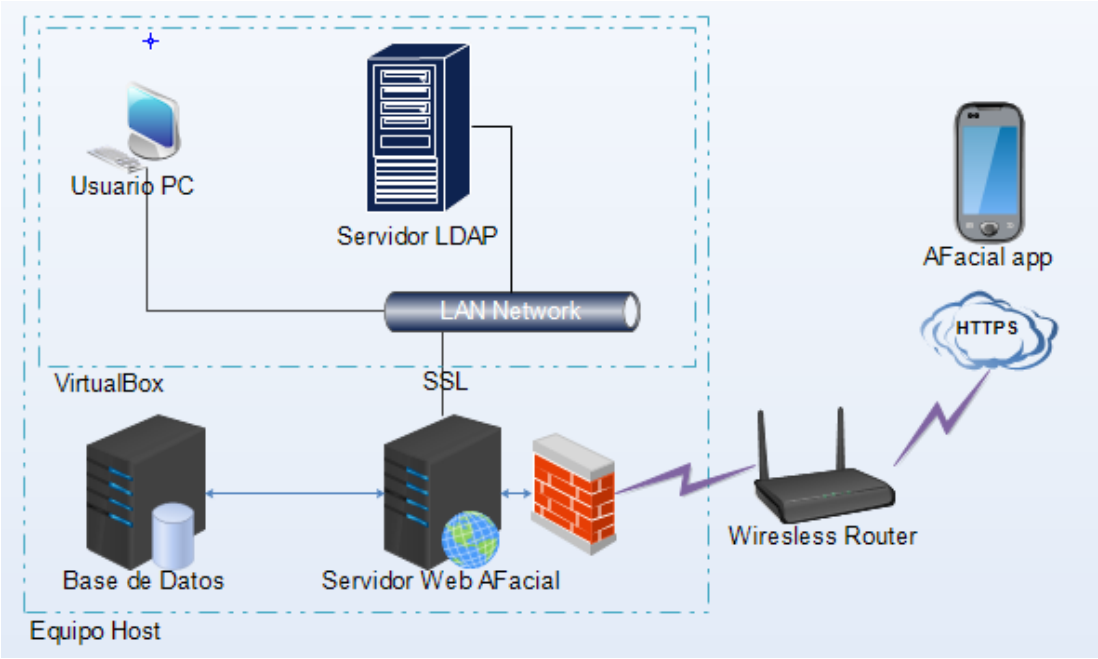


Figura 11: Diagrama de Arquitectura

La Tabla 5 muestra las características de los diferentes componentes de Hardware y Software, usados en este desarrollo de Software.

Tabla 5: Características hardware y software de los componentes del diagrama de arquitectura

Equipo	Características Hardware Software
Smartphone	Android Jelly, Lolipop
Wiresless router	Mifi At&T
Equipo Host	Windows 10 64 bits, procesador Intel icore3 1.80 GHz, Memoria 4 gb.
Servidor Web AFacial	Wamp 2.1 beta 3, Apache/2.2.17 (Win32) mod_ssl/2.2.17 OpenSSL/0.9.8q PHP/5.3.5
Servidor Base de datos	MySQL 5.5.8
VirtualBox	5.0.26
Servidor LDAP	Windows 2008 (64-bit) virtual
Usuario PC	Windows 7 virtual

4.2.3 Diagramas de Casos de Uso

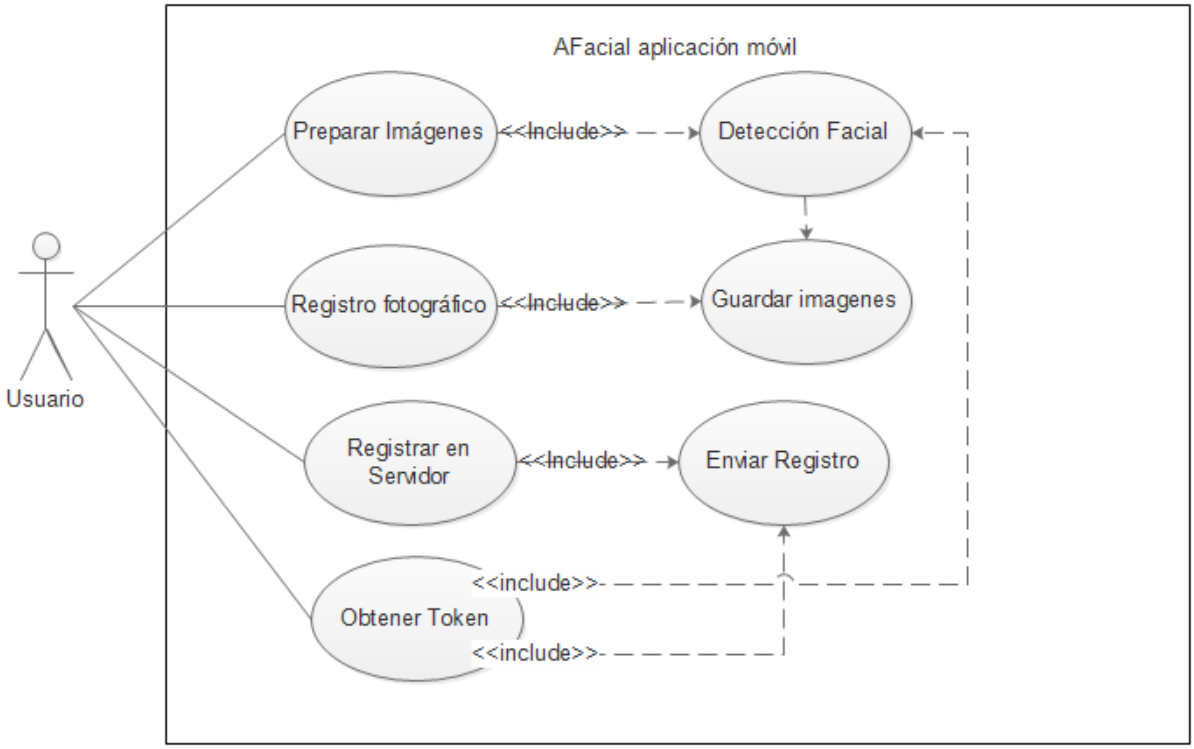


Figura 12: Diagrama de caso de uso para aplicación móvil AFacial

En el diagrama de caso de uso anterior Figura 12, se muestra a grandes rasgos las acciones que el usuario puede ejecutar en la aplicación móvil AFacial. A continuación, en la Figura 13, se muestra las acciones que se pueden realizar en la aplicación servidor web biométrico por parte de un administrador y la aplicación móvil.

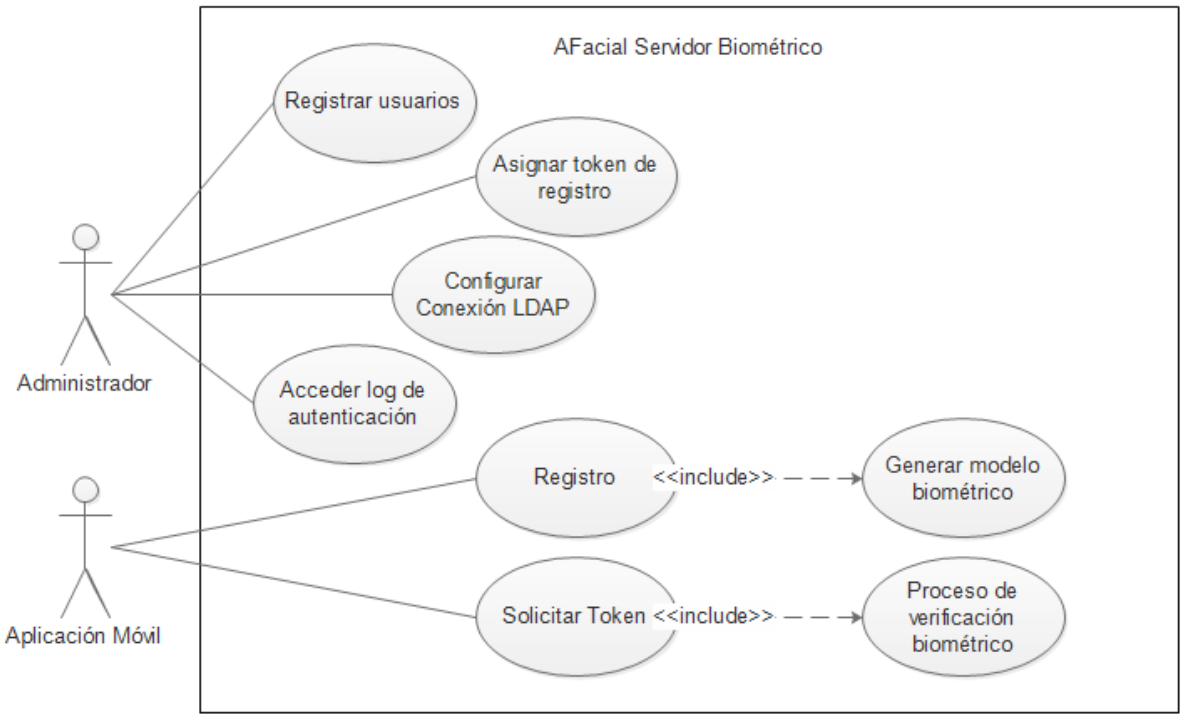
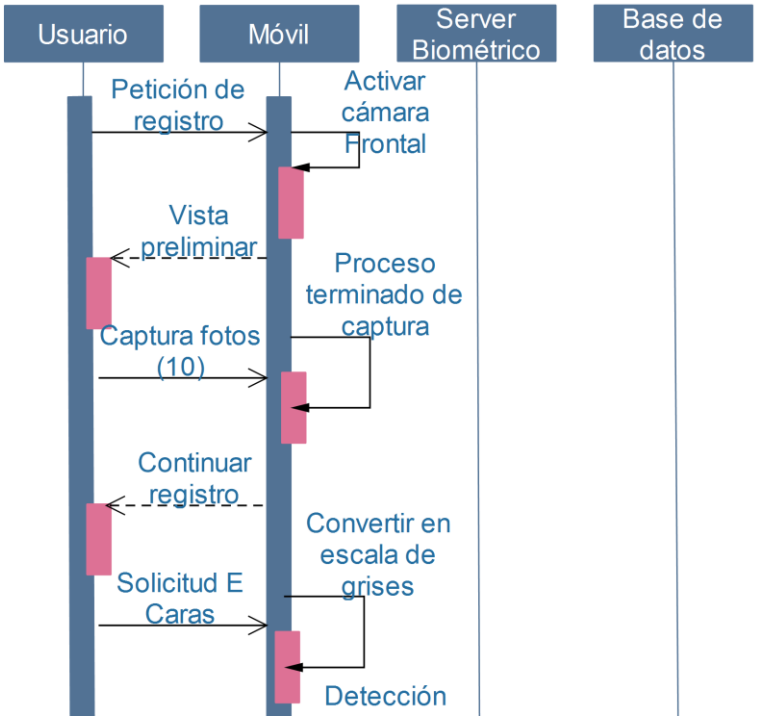


Figura 13: Diagrama de caso de uso de servidor biométrico AFacial

4.2.4 Diagramas de secuencia

En esta sección se incluyen los diagramas de secuencias que más resaltan y también aquellos pasos más importantes que ayudan a describir el proceso.

4.2.4.1 Diagrama de secuencia de Registro Biométrico



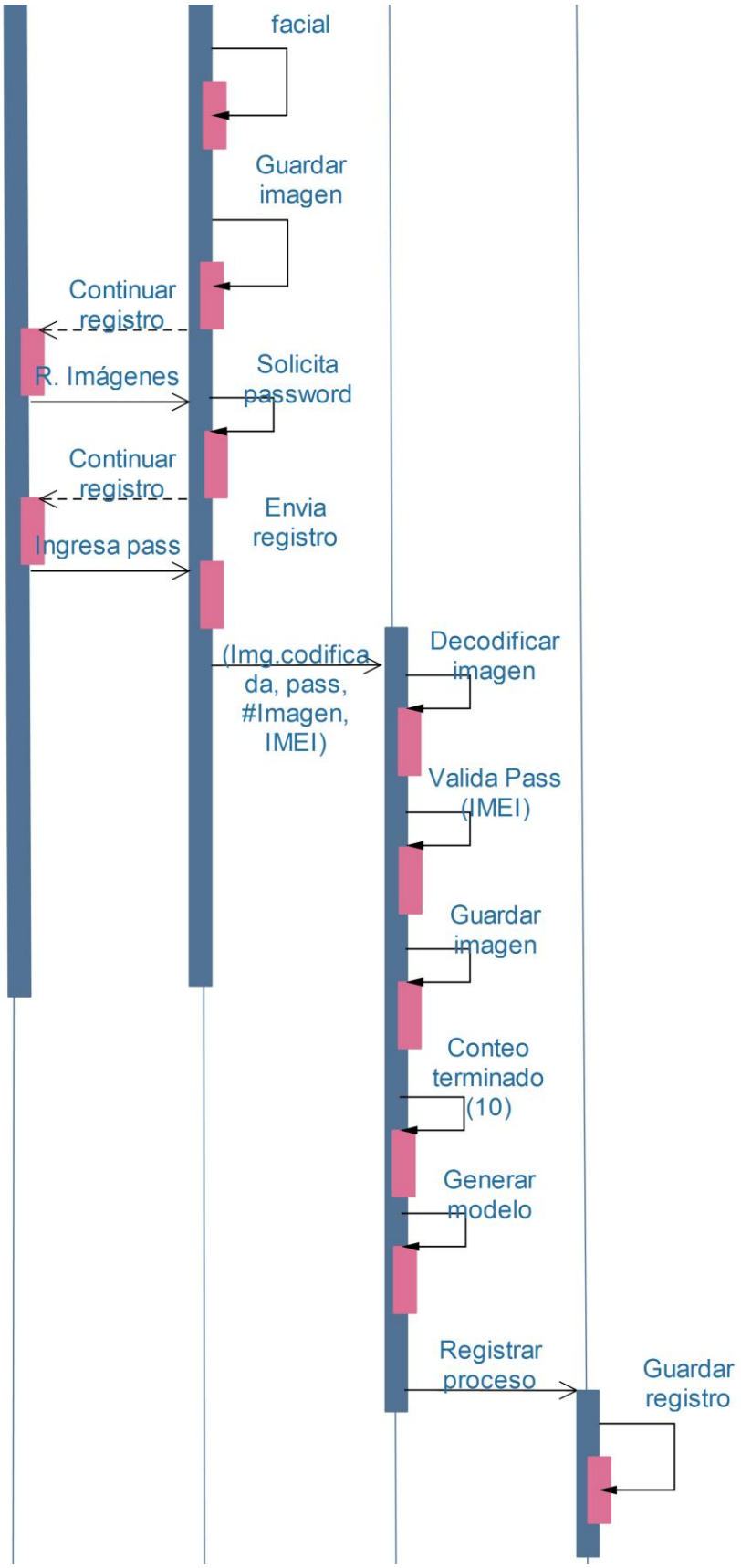
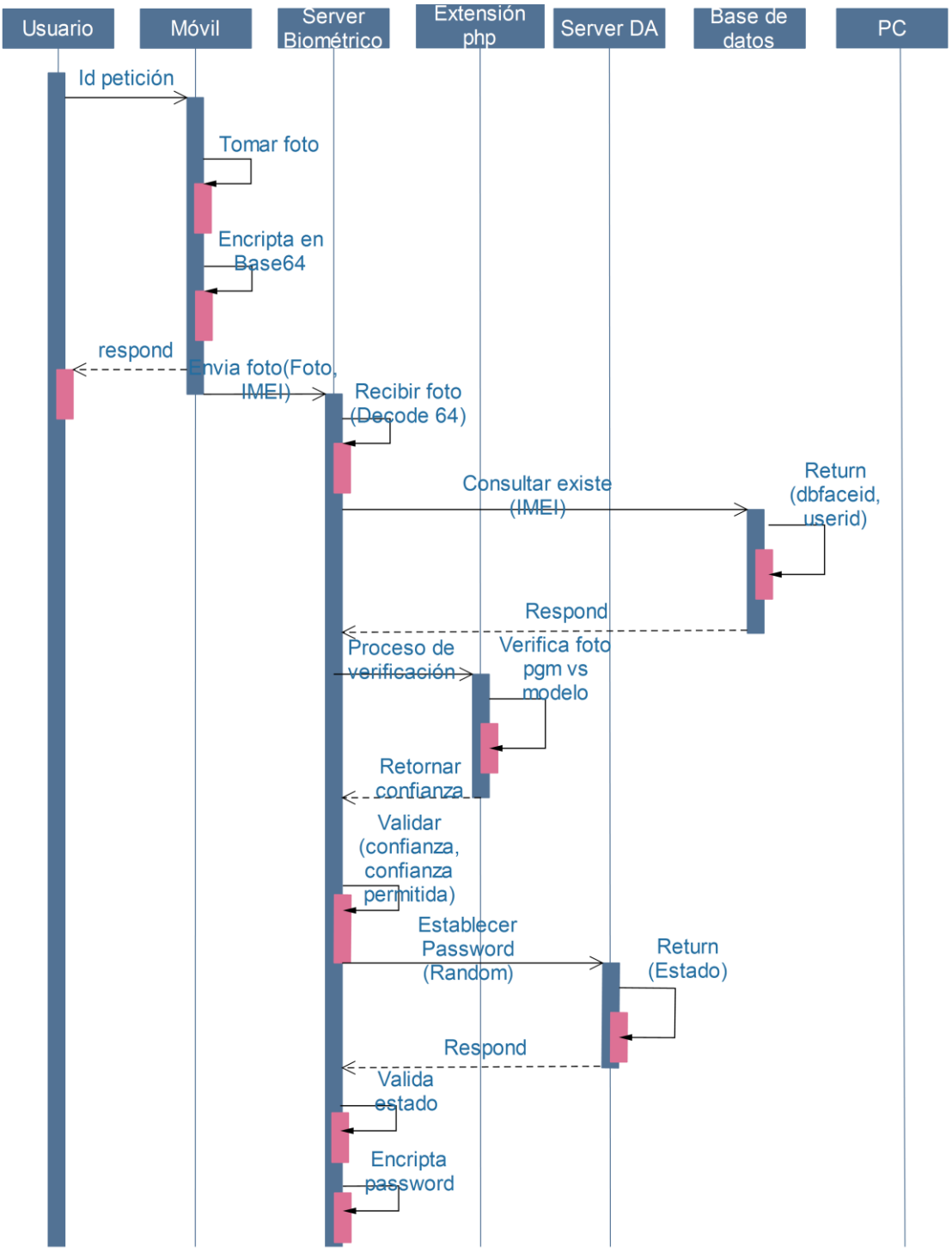


Figura 14: Diagrama de secuencia Registro Biométrico

El proceso anteriormente descrito en la Figura 14, permite al usuario realizar la digitalización del rostro y enviar la información al servidor biométrico para que este almacene dicha información y permita posteriormente realizar el reconocimiento facial.

4.2.4.2 Diagrama de secuencia solicitud de token



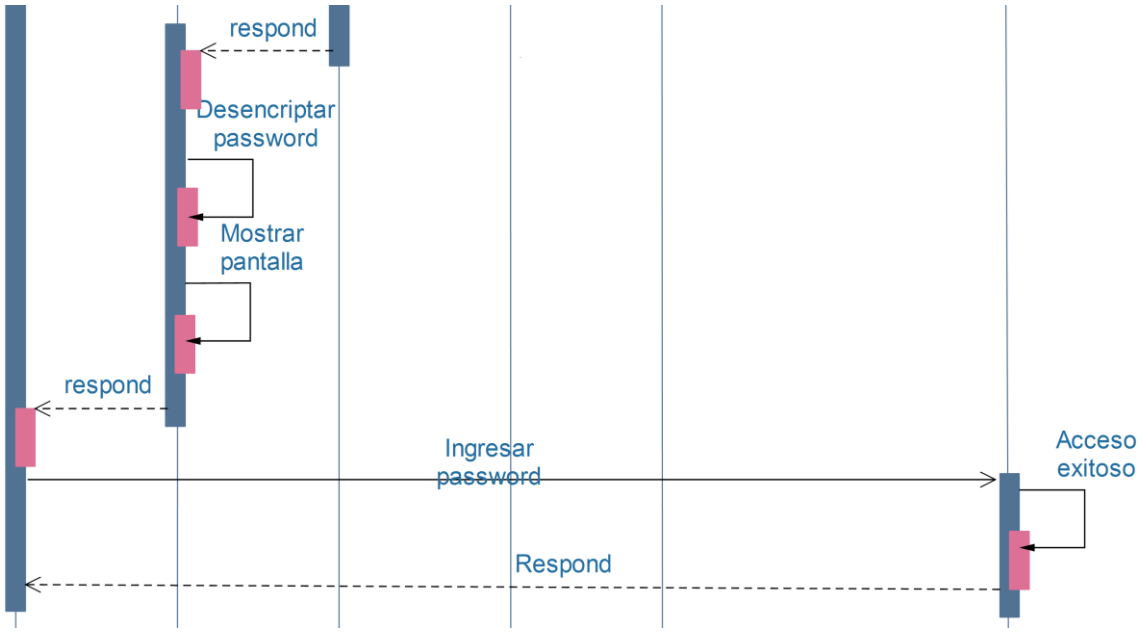


Figura 15: Diagrama de Secuencia de Solicitud de Token por el usuario

El proceso descrito en la Figura 15, permite al usuario solicitar un token o password para el ingreso a una estación de trabajo que este bajo el dominio que se encuentra configurado en el servidor biométrico.

4.2.4.3 Diagrama de secuencia de registro de usuarios en el sistema biométrico

El siguiente diagrama de secuencia Figura 16, describe el proceso de registro de usuarios en el sistema biométrico, es aquí donde se guarda la relación del usuario con respecto al dispositivo móvil que usaría para la solicitud posterior de tokens, así como el userid correspondiente el servidor Ldap.

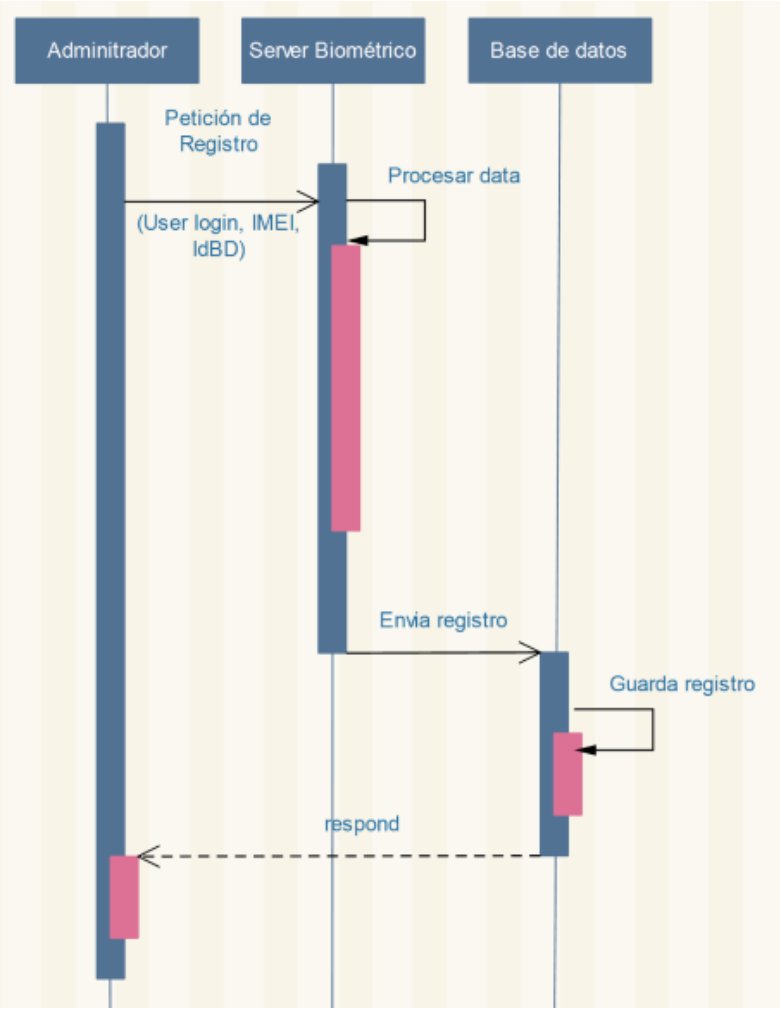


Figura 16: Diagrama de secuencia de registro de usuarios en servidor biométrico

4.2.4.4 Diagrama de secuencia generación de token para registro de usuario

En el siguiente diagrama Figura 17, se describe el proceso de generación de un token para que el usuario pueda realizar un registro biométrico. Este es un mecanismo implementado para que en un primer momento o cuando el usuario lo desee, realice el registro biométrico solo a través de un token o password. Este token sería entregado de forma personal por el administrador del sistema u otra forma que se desee implementar.

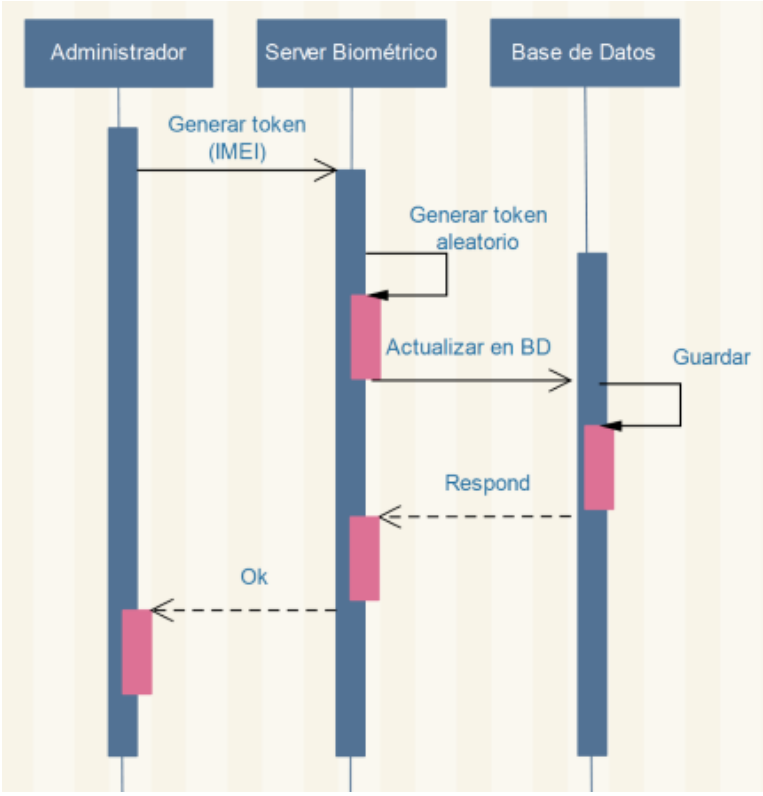


Figura 17: Diagrama de secuencia para generar token de registro biométrico para el usuario

4.2.5 Funcionamiento y flujos de los sistemas desarrollados

A partir de este apartado se encontrarán imágenes de personas diferentes a la del titular de este trabajo de fin de máster, las cuales han sido autorizadas para publicar por sus propietarios.

4.2.5.1 Aplicación AFacial móvil

Esta aplicación funciona para dispositivos Android. En la Figura 18 se puede observar el menú principal del usuario, donde dispone de 4 opciones:

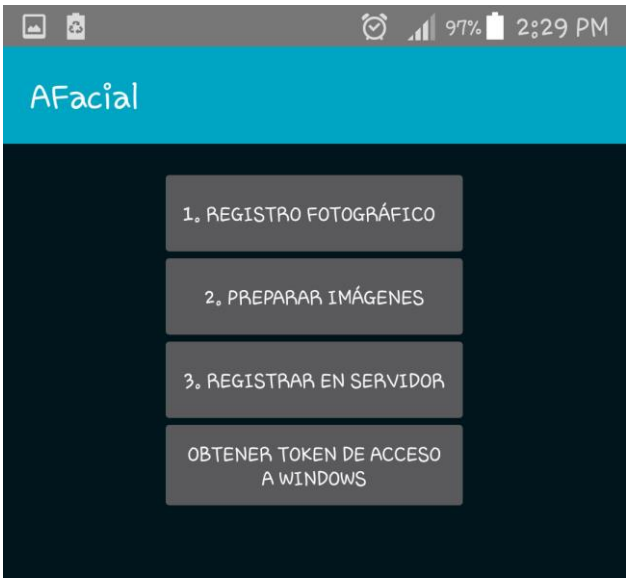


Figura 18: Menú principal de aplicación AFacial para Android

- Registro Fotográfico: Con esta opción el usuario registrará las respectivas fotografías de sí mismo, donde la imagen obtenida tiene la información completa del entorno. En la Figura 19 se puede ver un ejemplo de la interfaz y de la imagen fotográfica tomada.

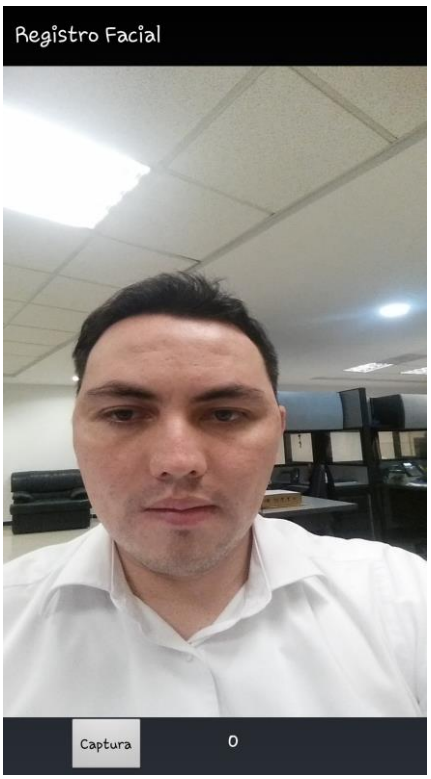


Figura 19: Interfaz de captura de registro fotográfico o facial del usuario

En la Figura 20 se describe el proceso interno que realiza el programa:

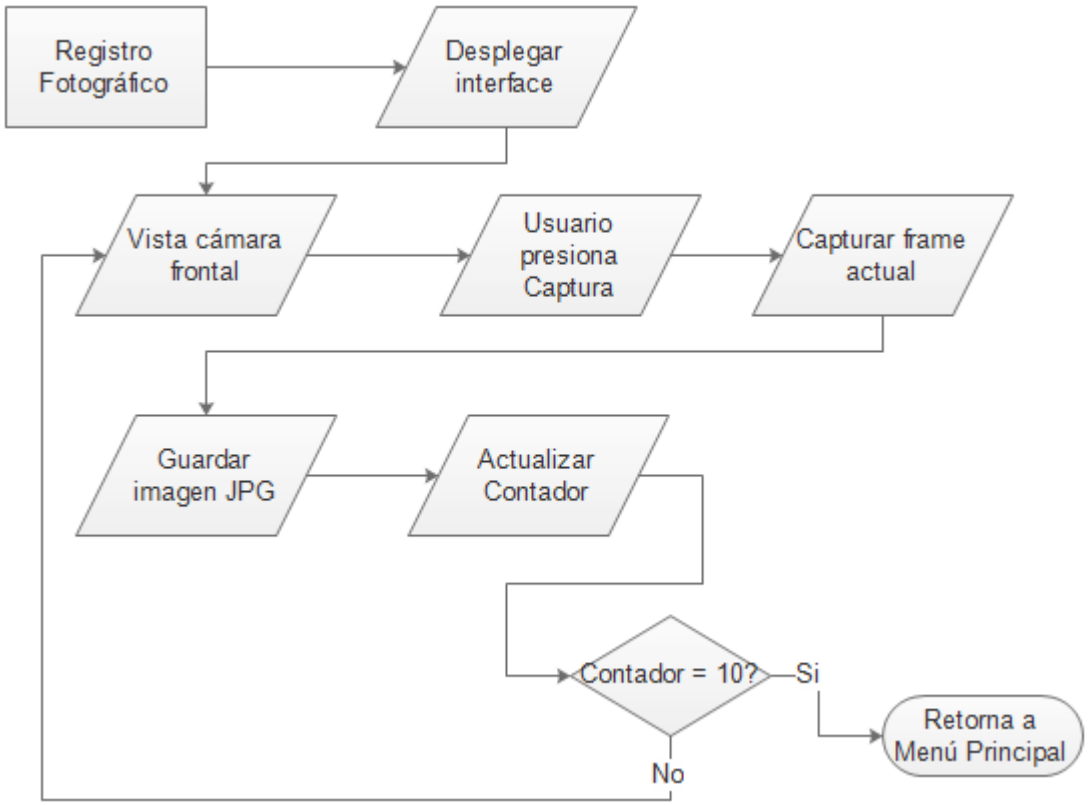


Figura 20: Proceso de registro fotográfico aplicación móvil

- Preparar imágenes: Este paso le permite al usuario extraer de las fotografías tomadas en el paso anterior, su dato biométrico, en este caso el rostro, separando así los demás elementos del entorno. En la Figura 21 se puede ver el resultado de este proceso, donde se ve que sólo queda el rostro de la persona en la imagen.

Figura 21: Resultado detección facial dispositivo móvil

En la Figura 22 se describe el proceso interno que realiza el programa:

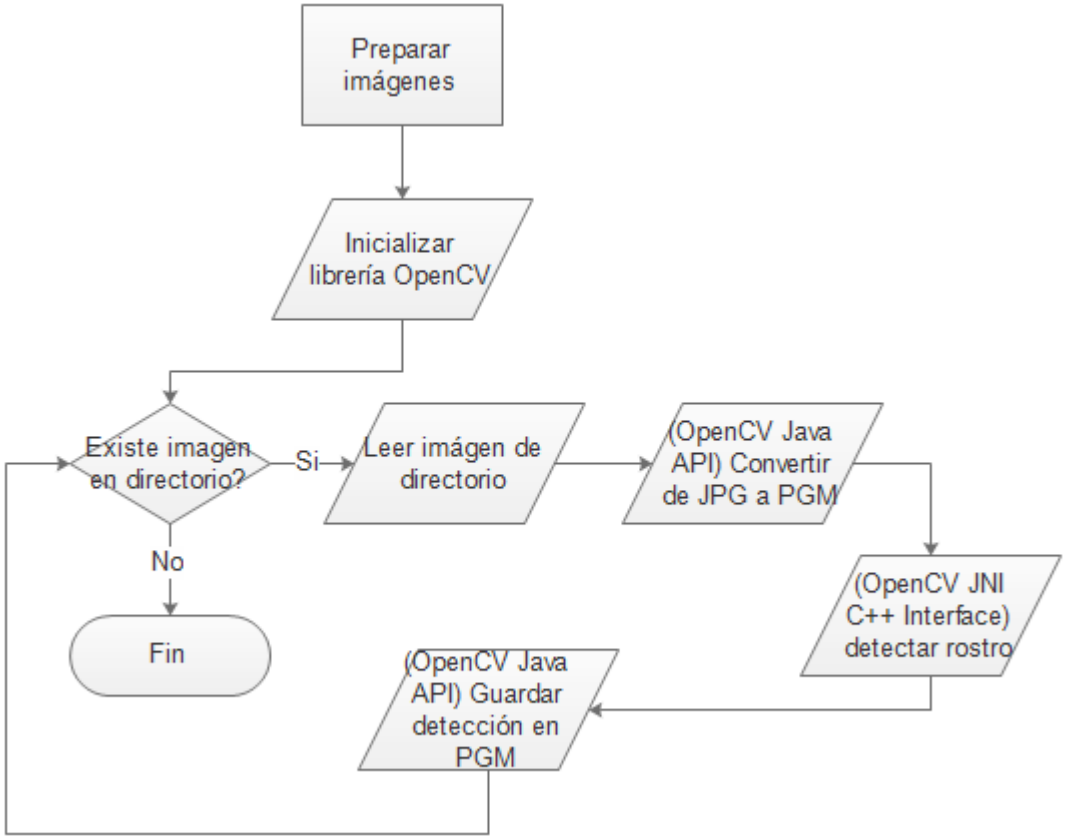


Figura 22: Proceso de preparar imágenes en dispositivo móvil

- Registrar en servidor: Este paso le permite al usuario registrar su dato biométrico en el servidor biométrico. En la Figura 23 se puede observar que la aplicación le solicita al usuario un token (password) para poder ser autorizado por el servidor a realizar dicho proceso, luego de que el usuario digite el token y oprima el botón “Ok” el sistema comenzará a enviar cada imagen al servidor biométrico.

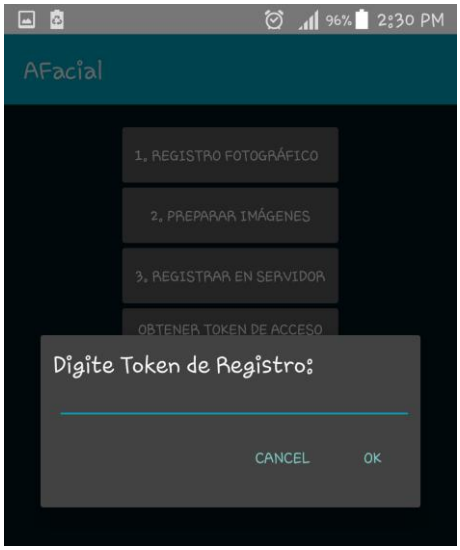


Figura 23: El usuario al registrar en el servidor, debe proveer el token de registro

En la Figura 24 se describe el proceso interno que realiza el programa:

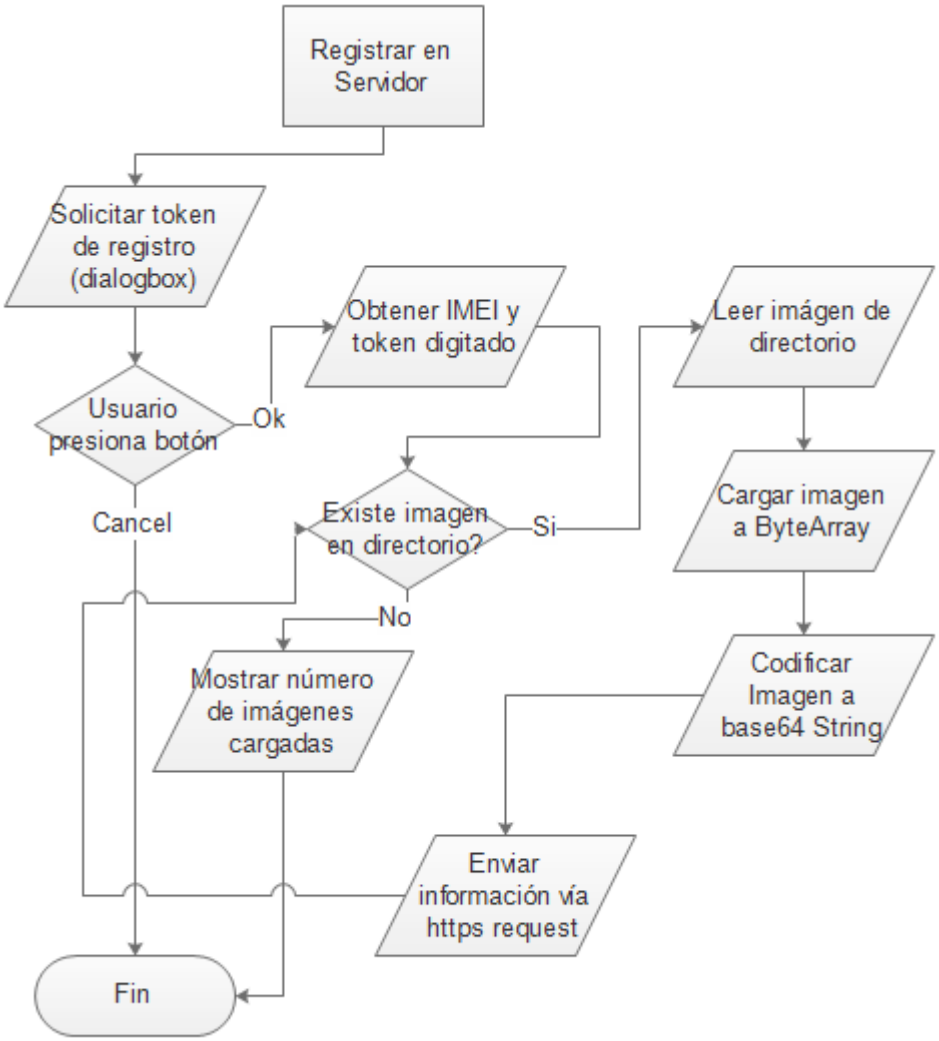


Figura 24: Proceso de registrar en servidor desde dispositivo móvil

- Obtener token de acceso: Esta opción le permite al usuario obtener un token (password) de acceso al equipo o estación de trabajo. En la Figura 25 se puede ver que la interfaz le muestra la vista de la cámara frontal, permitiéndole al usuario realizar una captura de su rostro con el botón “Captura”, para posteriormente con el botón “Token” enviar el dato biométrico (rostro detectado de la captura) al servidor biométrico. En la Figura 26, en la imagen izquierda se ve una autenticación autorizada por el servidor donde le muestra el Token al usuario, y en la imagen derecha se ve una autenticación no autorizada por el servidor, mostrándole un mensaje de “Autenticacion no valida”, se aclara que las tildes del mensaje presentaron problema y por eso se muestra sin tildes.

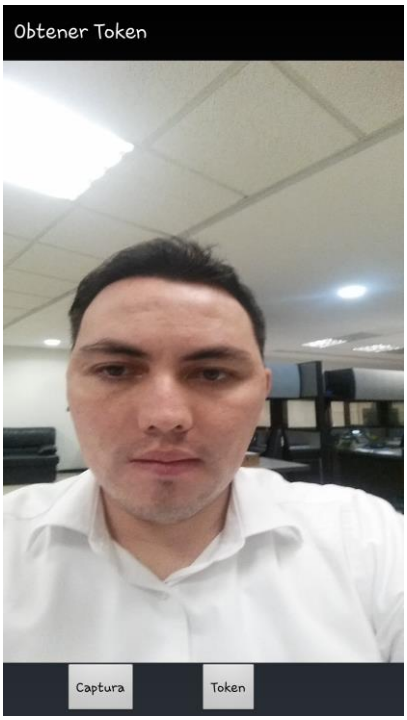


Figura 25: Interfaz para solicitar token de ingreso a Windows.

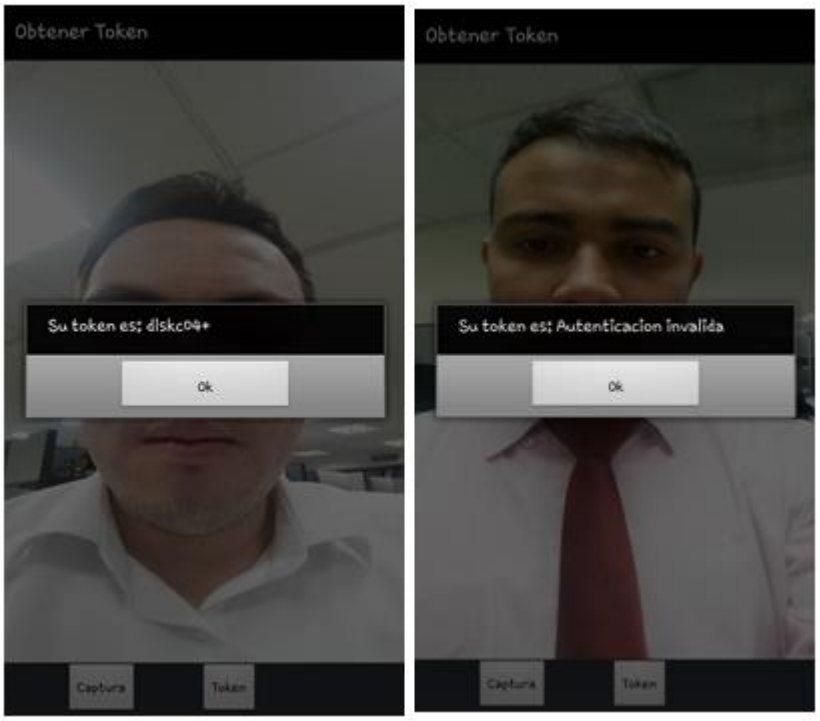


Figura 26: Resultado de autenticación, usuario titular a la izquierda, usuario impostor a la derecha

En la Figura 27 se describe el proceso interno que realiza el programa:

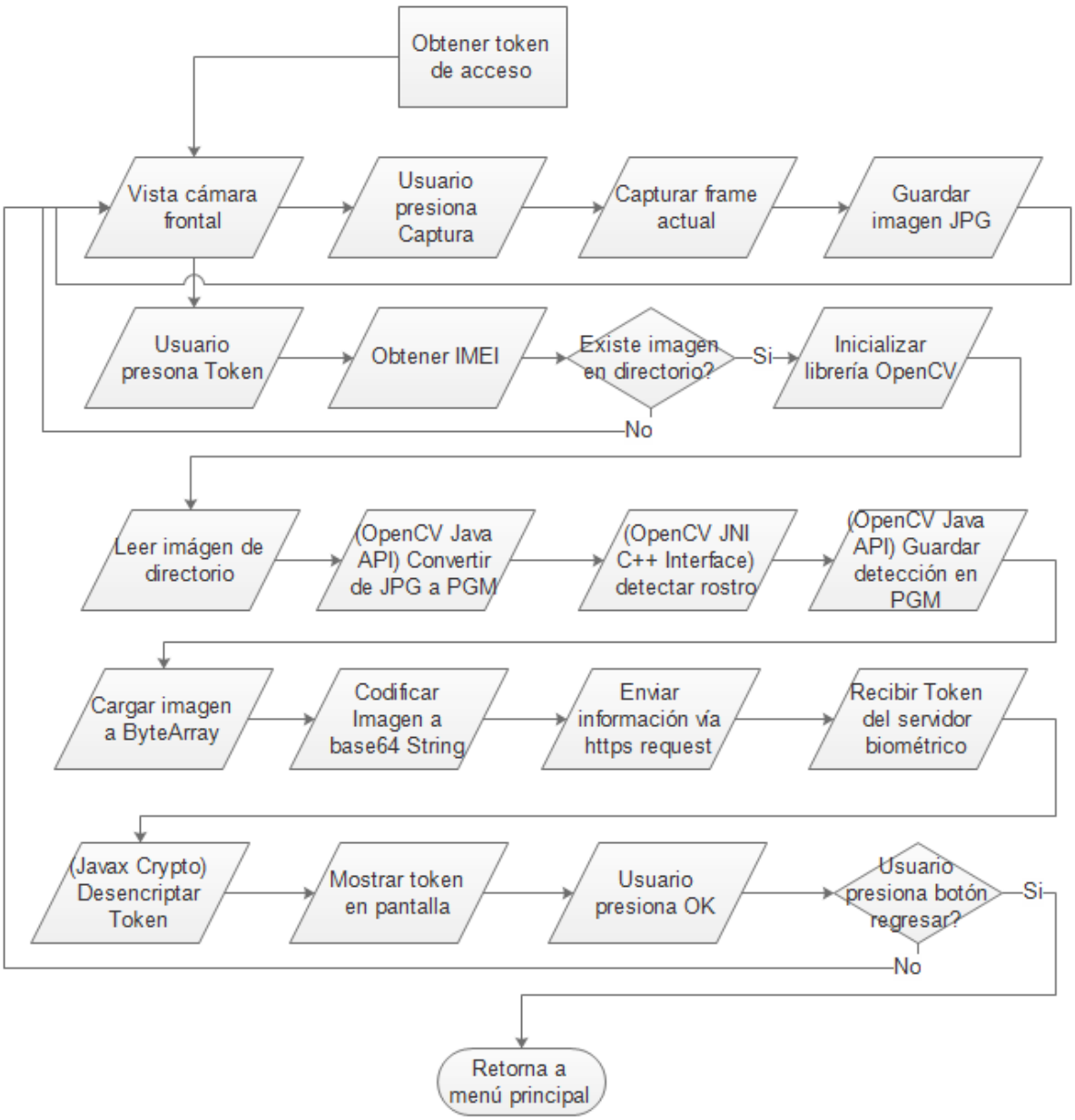


Figura 27: Proceso de obtener token de acceso desde dispositivo móvil

4.2.5.2 Servidor biométrico

El servidor biométrico consta de 4 componentes principales:

- Página de administración nombrada como “Portal de Administración de AFacial”, en donde se realizan las tareas que el administrador puede realizar.
- Página php “syncreg.php” que procesa la petición de registro biométrico invocado por el usuario desde el dispositivo móvil en el paso “Registrar en servidor” visto en la sección anterior 4.2.5.1.

- Página php “servicetoken.php” que procesa la petición de registro biométrico invocado por el usuario desde el dispositivo móvil en el paso “Obtener token de acceso” visto en la sección anterior 4.2.5.1.
- Extensión dll creada para realizar las operaciones de reconocimiento facial en C++ de OpenCV desde php.

4.2.5.3 Portal de Administración de AFacial

Las operaciones que puede realizar el administrador en este portal son las siguientes:

- Login: Ingreso al portal con usuario y password ya que la página cuenta con manejo de sesión. En la Figura 28 se muestra la interfaz de login, y en la Figura 29 se muestra la interfaz de bienvenida.

Figura 28: Interfaz de login en servidor biométrico

Figura 29: Interfaz de inicio al acceder al portal de administración de AFacial

En la Figura 30 se describe el proceso interno que realiza el programa:

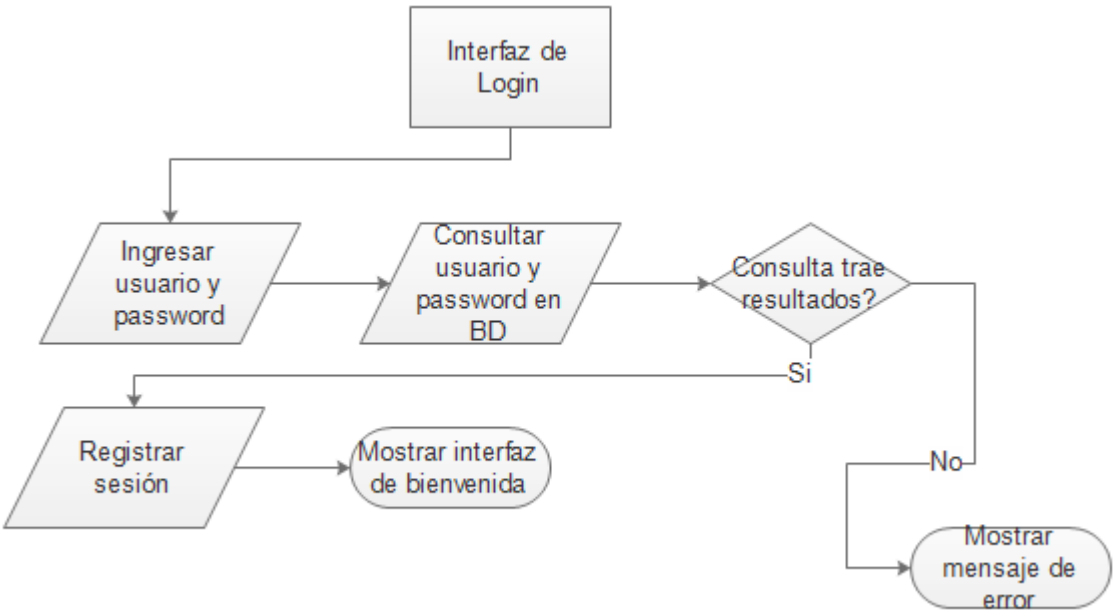


Figura 30: Proceso de login en servidor biométrico

- Registro: En la Figura 31 se muestra la interfaz Registro de usuarios, como su nombre lo indica, le permite al administrador registrar los diferentes usuarios en el sistema biométrico, los campos básicos son:
    - Usr login: el cual corresponde al samAccountName de la cuenta en Windows.

- IMEI: Corresponde al número IMEI del dispositivo móvil del usuario con el que realizará el proceso de registro biométrico y la solicitud de token.
- ID BD: Corresponde al ID con el que quedará almacenado el registro biométrico.

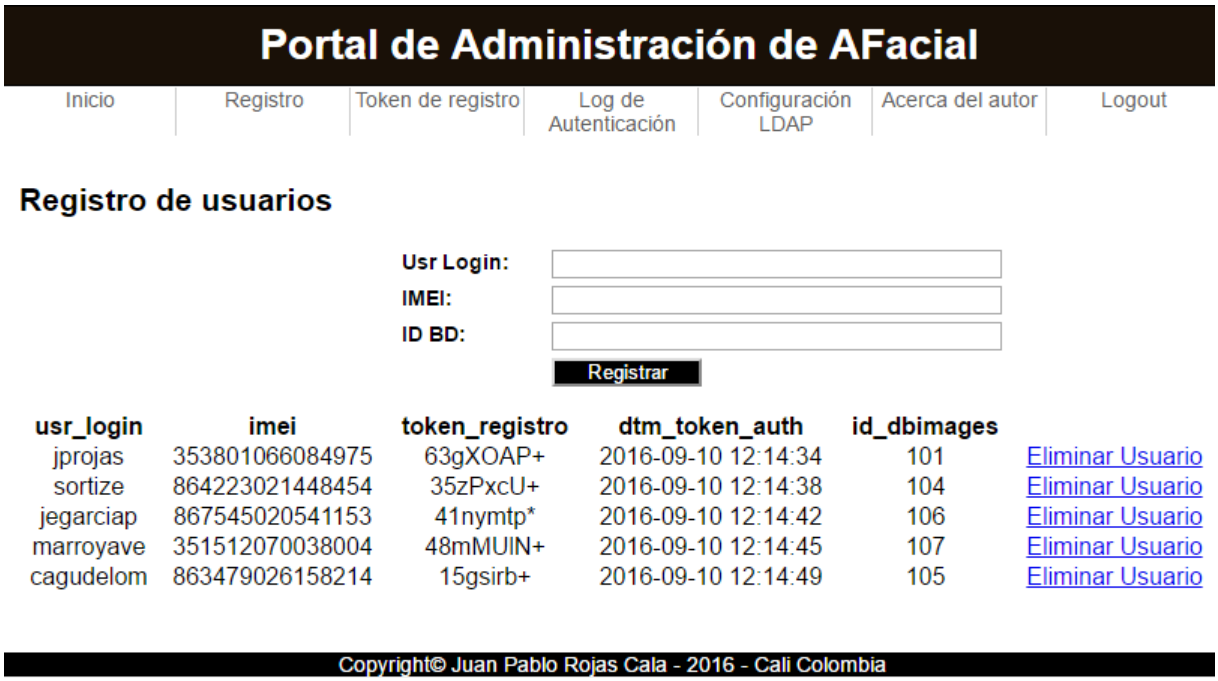


Figura 31: Interfaz de registro de usuarios al sistema biométrico

En la Figura 32 se describe el proceso interno que realiza el programa:

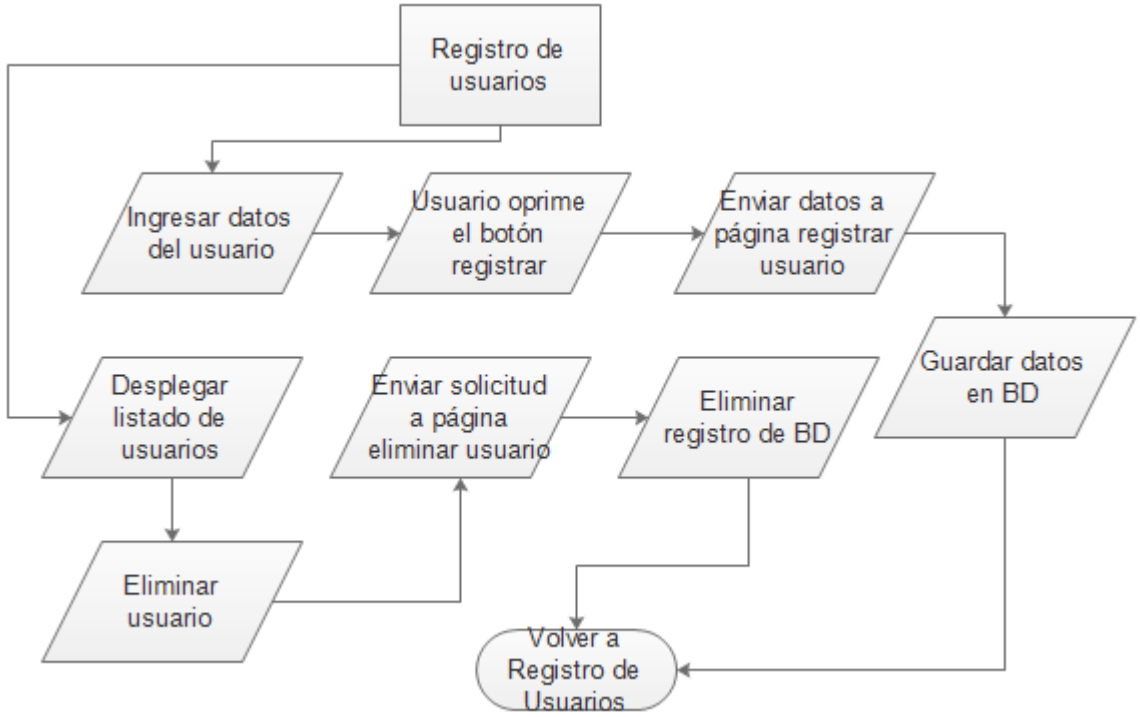


Figura 32: Proceso de registro de usuario en servidor biométrico

- Token de registro: En la Figura 33 se muestra la interfaz que permite al administrador generar un token para que el usuario posteriormente pueda realizar el proceso de registro biométrico desde el dispositivo móvil con el token dado.

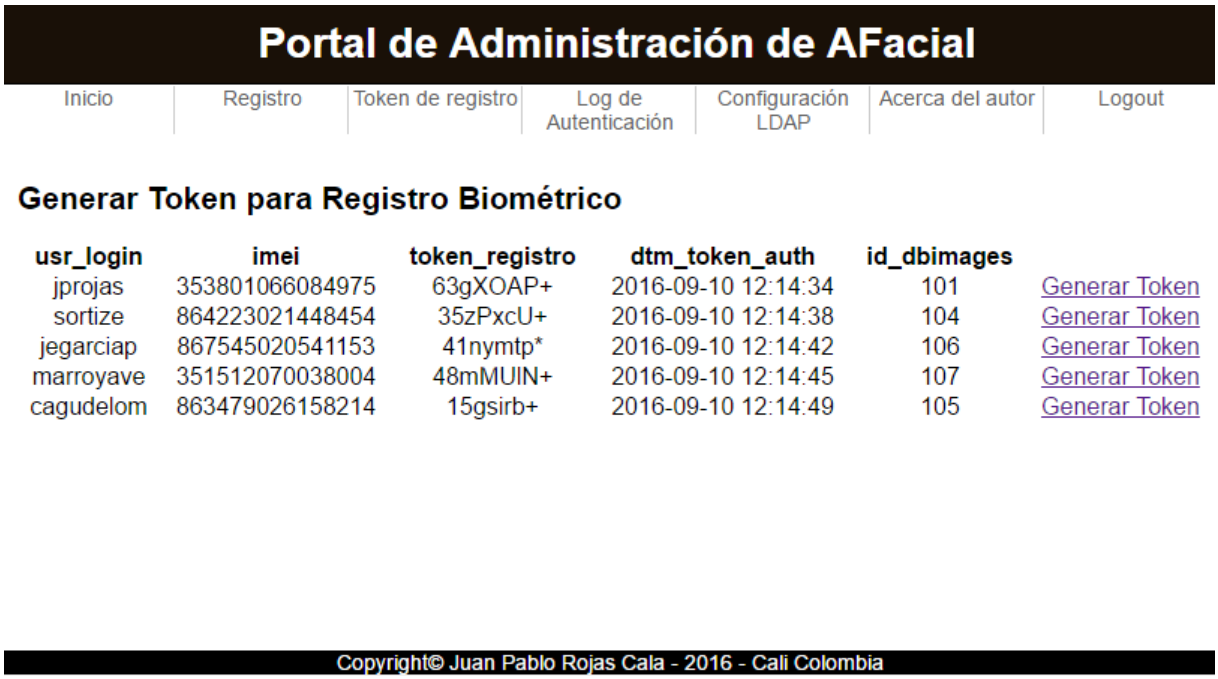


Figura 33: Interfaz de generación de token para registro biométrico

En la Figura 34 se describe el proceso interno que realiza el programa:

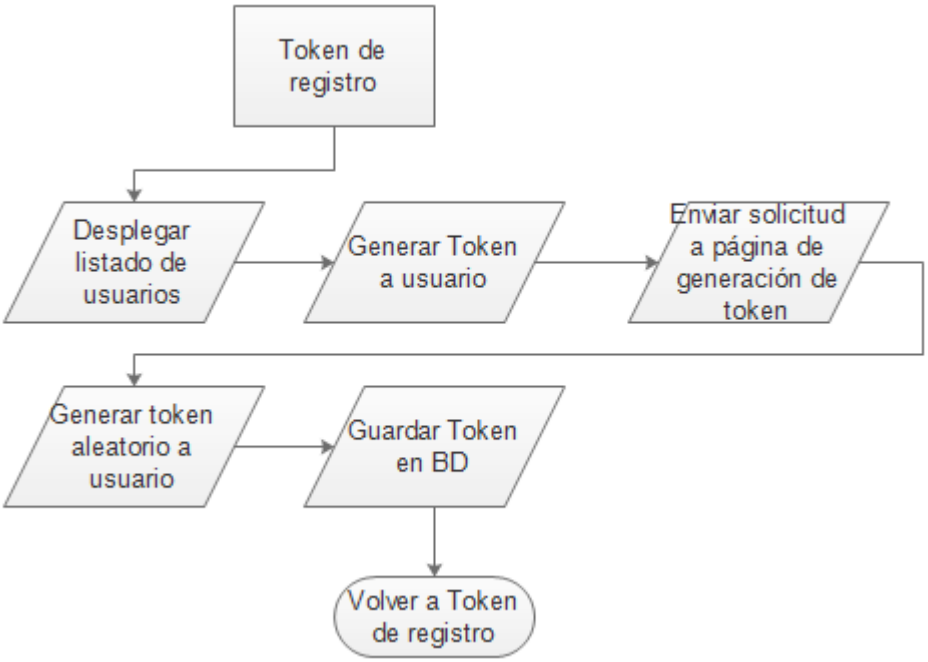


Figura 34: Proceso de crear token de registro en servidor biométrico

- Log de autenticación: En la Figura 35 se muestra la interfaz que le permite al administrador realizar un seguimiento de todas las solicitudes de autenticación. En este log se muestra:
  - Reg_bd: Corresponde a un valor de índice de tabla que se genera automáticamente al ingresar un registro en la tabla. Este campo permite identificar posteriormente la imagen pgm registrada del usuario que realiza la operación de autenticación, que puede corresponder al titular o a un impostor.
  - IMEI: el cual ayuda a identificar el dispositivo móvil desde donde se realiza la operación.
  - Confianza_obtenida: Es el resultado de confianza que proporciona el algoritmo de reconocimiento facial, cuando compara al sujeto titular con el sujeto que realiza la autenticación.
  - Fecha_hora: Fecha y hora en que el servidor biométrico registra el evento.
  - Obtener pgm: le permite al administrador obtener la imagen pgm del usuario que realiza la operación de autenticación, dándole aún más detalle del proceso. De esta forma también el administrador puede evaluar la efectividad del algoritmo de reconocimiento facial. En la Figura 36 se puede observar que se han obtenido dos imágenes pgm que hacen referencia al registro 16 y 22 del log de autenticación, el rostro del registro 16 es el titular, mientras que el rostro del registro 22 es un impostor.

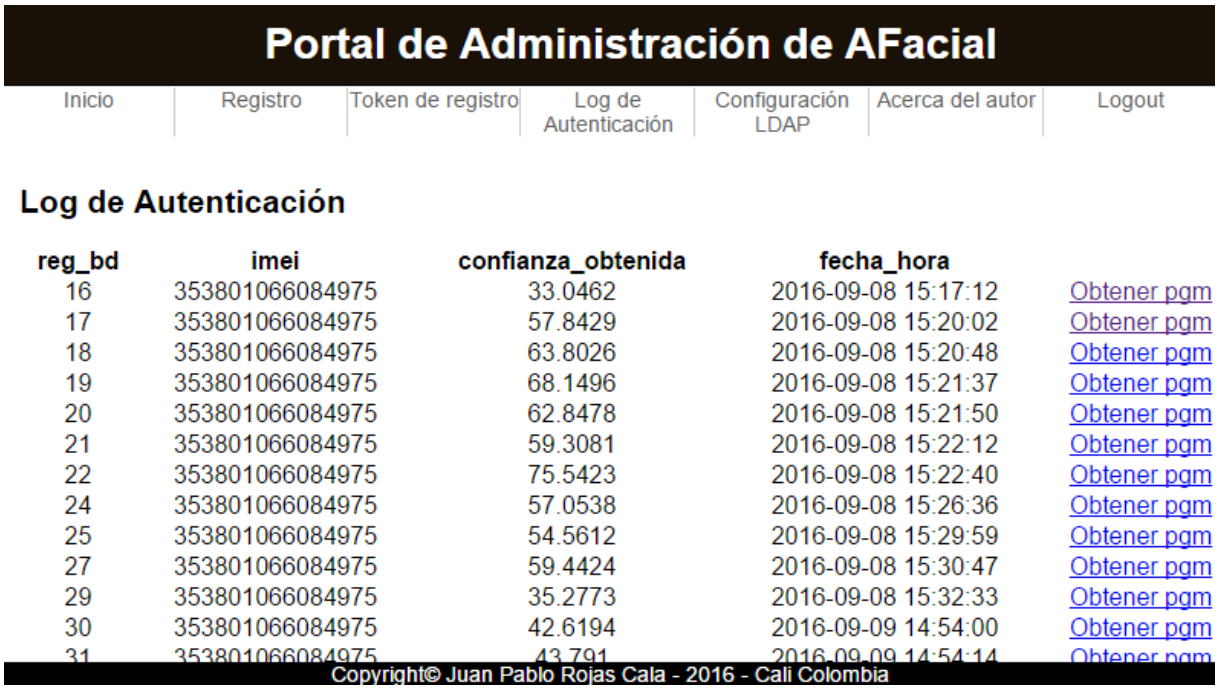


Figura 35: Interfaz que permite al administrador ver el log de autenticación

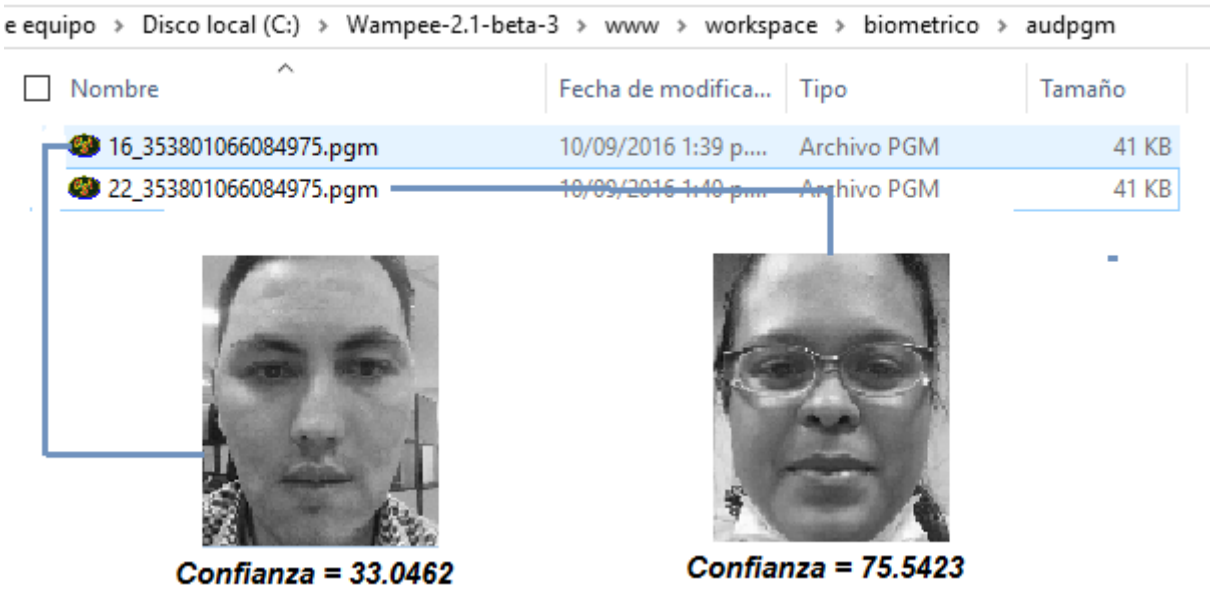


Figura 36: Imágenes pgm del log de auditoría, valores de confianza de acuerdo a Figura 33

En la Figura 37 se describe el proceso interno que realiza el programa:

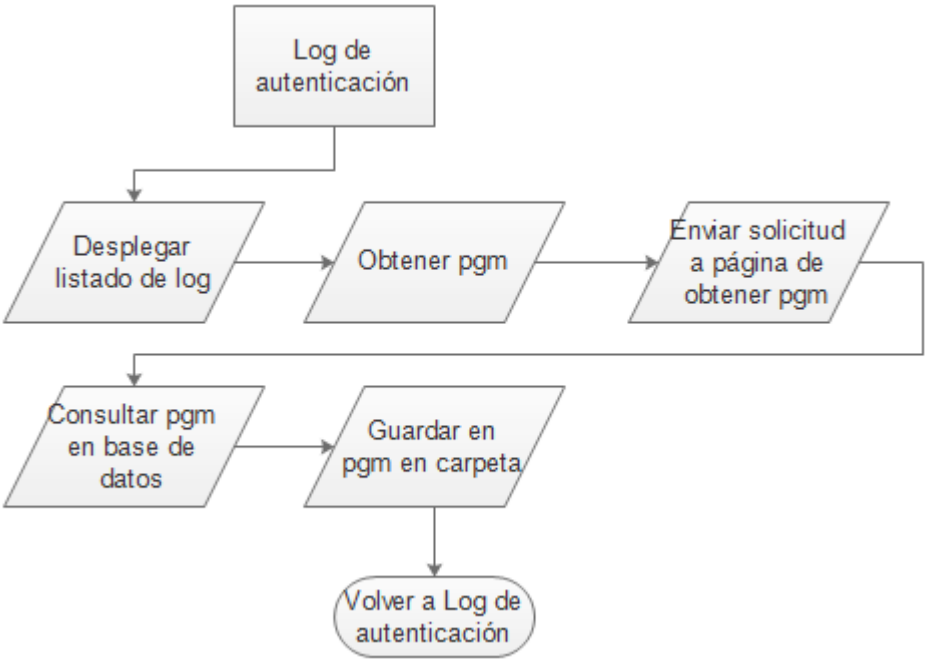


Figura 37: Proceso de log de autenticación en servidor biométrico

- Configuración LDAP: En la Figura 38 se muestra la interfaz de configuración Ldap, la cual le permite al administrador configurar los valores de conexión del servidor de Directorio Activo.

Figura 38: Interfaz de configuración de conexión a servidor LDAP

En la Figura 39 se describe el proceso interno que realiza el programa:

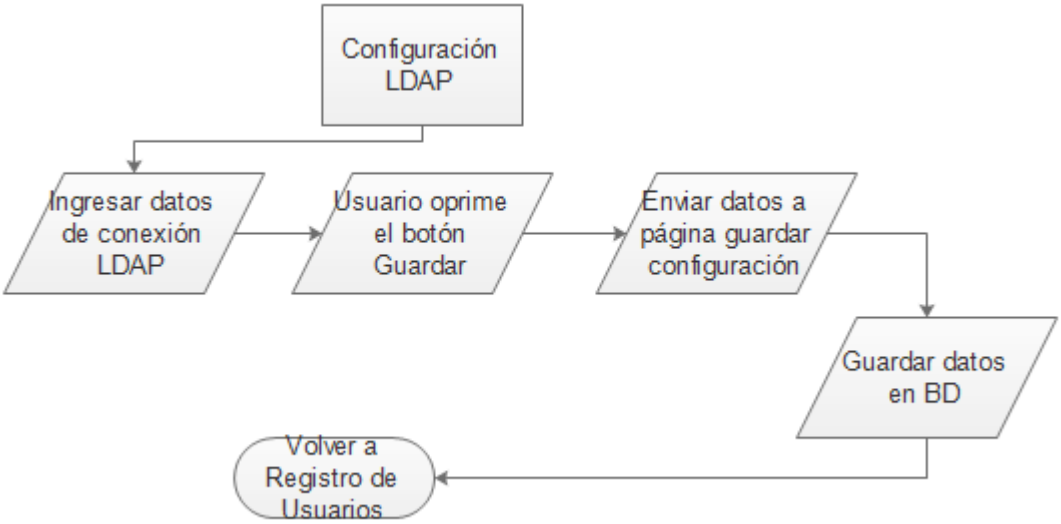


Figura 39: Proceso de configuración LDAP en servidor biométrico

4.2.5.4 Página php "syncreg.php"

Como se mencionó antes, esta página permite llevar a cabo el proceso de registro biométrico que realiza el usuario desde su dispositivo móvil. Esta página tiene implementado un mecanismo de seguridad de autenticación básica, de esta forma sólo la aplicación desarrollada para este software puede acceder a esta operación. En la Figura 40 se muestra el funcionamiento de este proceso de registro biométrico.

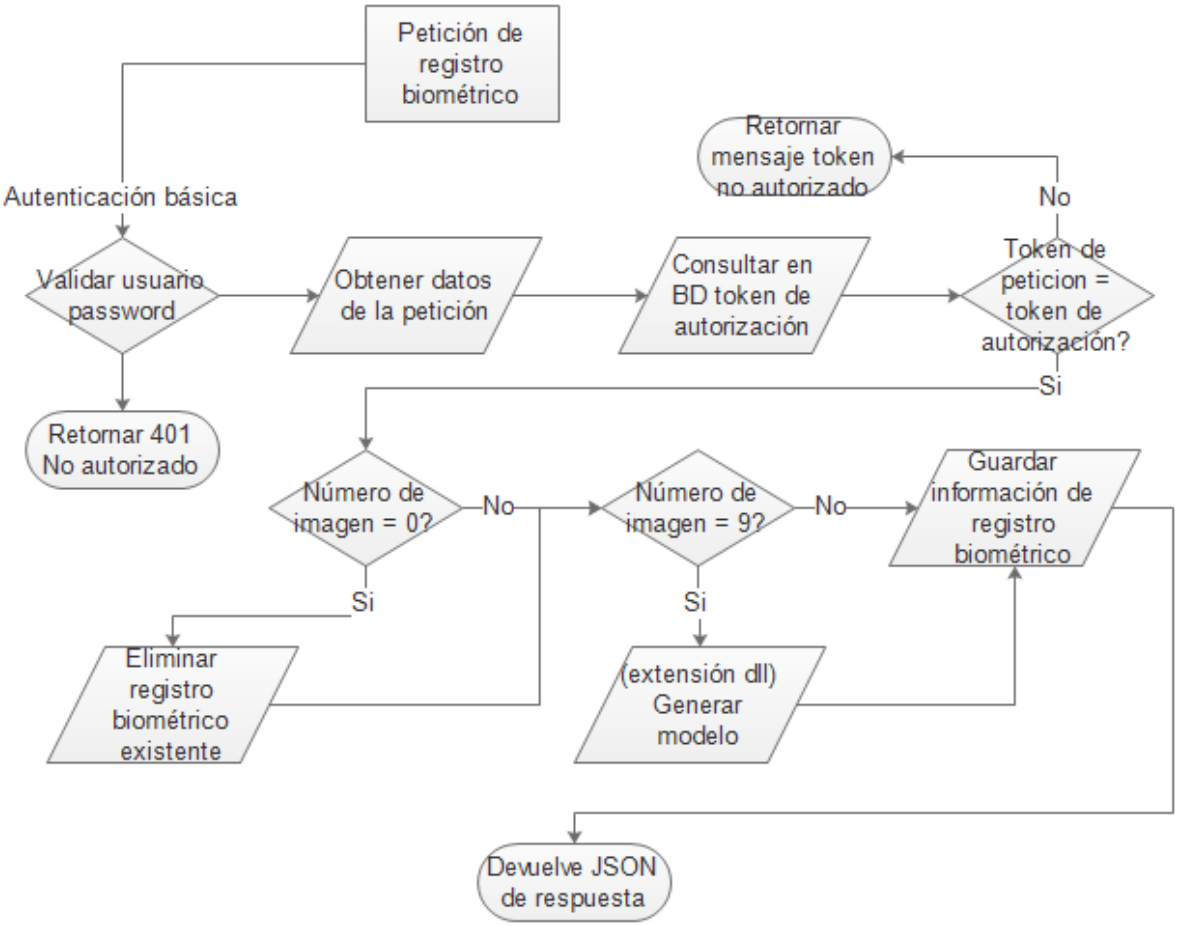


Figura 40: Proceso de petición de registro biométrico en servidor biométrico

4.2.5.5 Página php "servicetoken.php"

Esta página permite llevar a cabo el proceso de obtener token de acceso que realiza el usuario desde su dispositivo móvil. Esta página tiene implementado un mecanismo de seguridad de autenticación básica al igual que la página anterior "syncreg.php". En la Figura 41 se muestra el funcionamiento de este proceso de obtener el token de acceso.

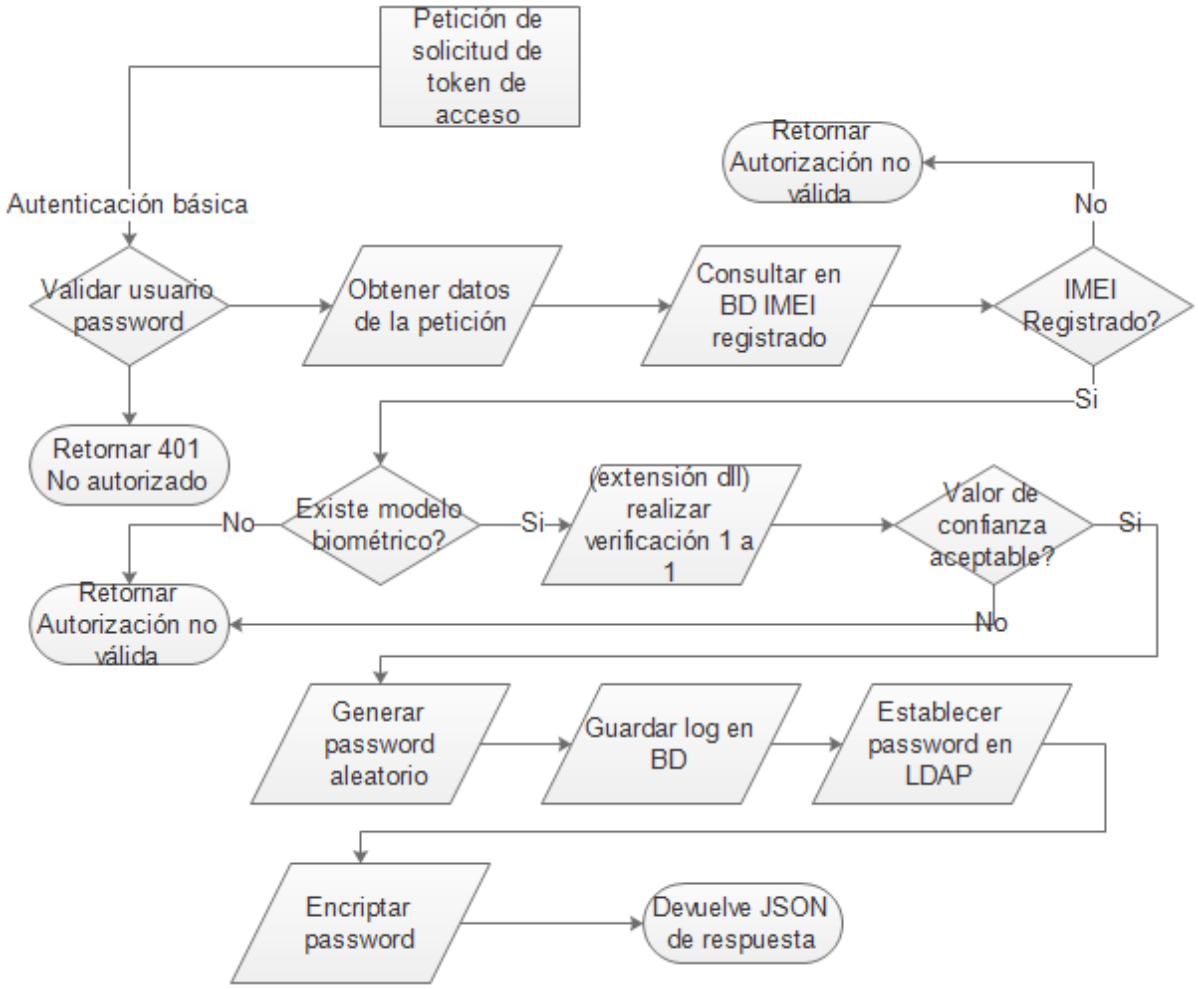


Figura 41: Proceso de solicitud de token en servidor biométrico

4.2.5.6 Extensión dll

Esta extensión permite realizar 2 operaciones que se encuentran en la librería de OpenCV en lenguaje C++:

- Generar Modelo: el cual corresponde al proceso de entrenamiento del algoritmo de reconocimiento facial LBPH a partir de un índice de imágenes.
- Realiza predicción: el cual permite dada una imagen como entrada compararla con un modelo que se encuentra en una ruta dada, esta ruta es parámetro también de la función.

Para poder llevar a cabo este desarrollo se usó la API Zend de PHP ("PHP: La API Zend: Hackeando el núcleo de PHP - Manual", 2016).

En la Figura 42 se encuentra su funcionamiento:

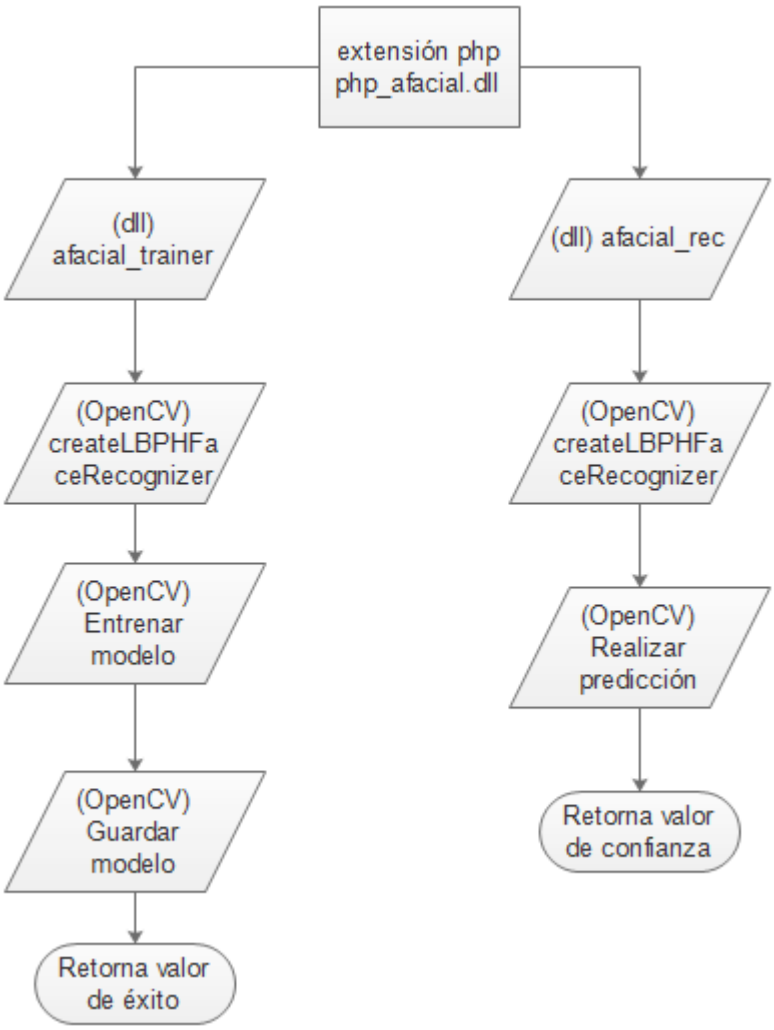


Figura 42: Funcionamiento de extensión dll php_afacial.dll

4.3 Evaluación

En el siguiente apartado se mostrarán los resultados obtenidos, y rendimiento del software desarrollado, para de esta forma demostrar que los objetivos planteados de este trabajo de fin de master, son desarrollados y alcanzados.

4.3.1 Evaluación de los algoritmos de reconocimiento facial

En este apartado se evalúan los algoritmos de reconocimiento facial, Eigenfaces, Fisherfaces y LBPH, como se ha planteado anteriormente, con el objetivo de determinar cuál tiene un mejor comportamiento en cuanto a la falsa aceptación y el falso rechazo.

El proceso de toma de datos consiste en los siguientes pasos:

1. Toma del registro biométrico del usuario que será el titular del registro.
2. Generación del modelo para cada algoritmo a partir del registro biométrico del usuario titular.

3. Con ayuda de 3 programas desarrollados para este trabajo, uno por cada algoritmo, se procede a realizar la comparación facial a través de cada algoritmo de reconocimiento, tanto para el usuario titular, como para otros usuarios quienes serían, usuarios impostores. Para esta prueba se usaron 5 usuarios impostores y 1 titular. Los programas son basados en el programa “gihantharanga/ComputerVision”.
4. Cada programa desarrollado se ejecuta durante 20 segundos, durante este tiempo lo que ocurre es que, por cada frame de video, se extrae el rostro de la persona y se compara con el dato biométrico del usuario titular. De esto, también se guarda el nivel de confianza con el que se realiza el reconocimiento facial para todo usuario y para cada algoritmo. La Figura 43 muestra el funcionamiento de estos programas, a la izquierda se muestra la detección facial, y a la derecha el log.

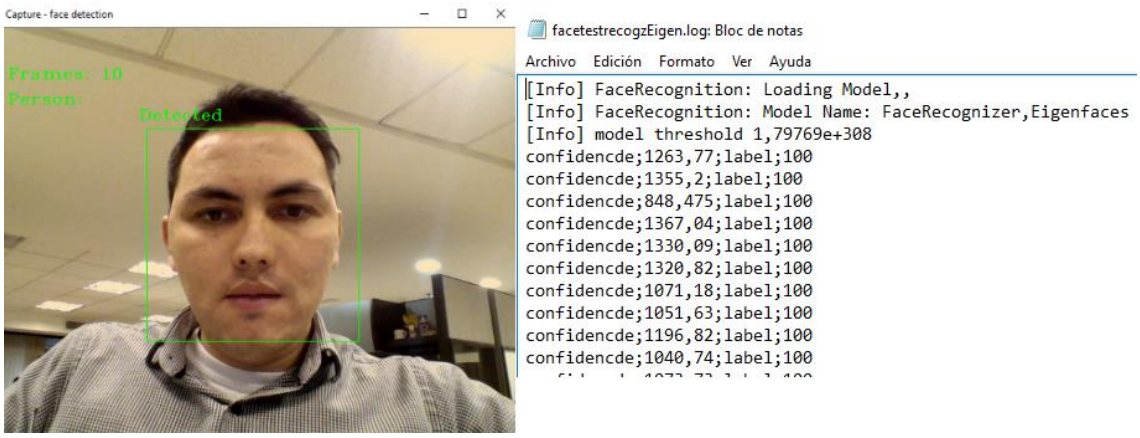


Figura 43: Ejemplo programa de evaluación de algoritmo Eigenfaces.

5. Las variables que determinan la evaluación son obtenidas a partir de esta información, y computadas a través de la herramienta online Alcula.com, las variables son: mínima confianza, máxima confianza, y desviación estándar, calculados a partir de los datos obtenidos en los diferentes logs de cada algoritmo y cada sujeto.

Los criterios para optar por el mejor son:

1. Baja dispersión, con ayuda de la variable desviación estándar, si es baja, los datos tendrán una baja dispersión. Lo que se busca es un algoritmo que dé resultados homogéneos y que estos no sean tan variables, de modo que se tenga un buen nivel de credibilidad con los datos de confianza que da el algoritmo. A pesar de que la base de datos de usuarios con la se cuenta es pequeña y la baja dispersión pueda requerir más datos de prueba, se procede a continuar a evaluar este criterio.

2. Mínima confianza y máxima confianza, permiten establecer el nivel de confianza que se podría manejar con el algoritmo, buscando evitar la falsa aceptación y el falso rechazo.

La Tabla 6, muestra los resultados obtenidos del proceso descrito anteriormente, el usuario titular es jprojas y los demás son usuarios impostores. El tamaño de la muestra es el número de comparaciones realizadas del usuario titular con el usuario impostor:

Tabla 6: Comparación de nivel de confianza entre los algoritmos Eigenfaces, Fisherfaces y LBPH

Usuario	ID detectado	Algoritmo	Tamaño de la muestra	Mínima Confianza	Máxima Confianza	Desviación estándar (s)
jprojas	***100***	***Eigenfaces***	***184***	***603***	***1942***	***253***
adonay	100	Eigenfaces	160	1452	1955	106
fausto	100	Eigenfaces	185	1418	1750	71
jegarciap	100	Eigenfaces	199	1686	2227	143
marroyave	100	Eigenfaces	256	1948	2760	159
sortize	100	Eigenfaces	167	1147	1728	123
jprojas	***100***	***Fisherfaces***	***180***	***7***	***335***	***59***
adonay	100	Fisherfaces	100	1886	2357	116
adonay	101	Fisherfaces	50	2055	2355	81
fausto	101	Fisherfaces	188	52	332	47
jegarciap	100	Fisherfaces	148	4	1038	111
marroyave	100	Fisherfaces	170	1139	3083	235
marroyave	101	Fisherfaces	144	1534	1942	81
sortize	101	Fisherfaces	172	1306	1901	109
jprojas	***100***	***LBPH***	***140***	***30***	***39***	***2***
adonay	100	LBPH	148	48	59	2
fausto	100	LBPH	153	67	88	4
jegarciap	100	LBPH	170	56	88	9
marroyave	100	LBPH	225	62	184	26
sortize	100	LBPH	172	67	89	4

Las observaciones que se tienen de la Tabla 6 son:

- Algoritmo Eigenfaces:
  - Para el usuario titular presenta la mayor desviación estándar de los 3 algoritmos.
  - La mínima y máxima confianza del usuario titular, están dentro de un rango donde los impostores hacen parte, se da falsa aceptación y falso rechazo.
- Algoritmo Fisherfaces:
  - El principal inconveniente es que el algoritmo requiere una base de 2 sujetos para poder generar el modelo, esto implica que no se usa el concepto 1 a 1

de verificación del todo. Sin embargo, se procede a generar la base mínima y generar el modelo para obtener resultados.

- La mínima y máxima confianza no tienen una tendencia clara, no es posible determinar el nivel de confianza.
- Aleatorio id de detección.

- Algoritmo LBPH:
  - Tiene la más baja desviación estándar, lo que implica que tiene una buena homogeneidad de datos, haciendo el algoritmo más creíble.
  - La mínima y máxima confianza, en este caso, distinto a los algoritmos anteriores, tienen una tendencia donde se ve que no se interpone el rango del usuario titular con el rango de cualquier usuario impostor de la prueba.

Por lo tanto, por esto se concluye que el algoritmo LBPH es el mejor de los 3, el cual se puede manejar con una confianza de 43, de acuerdo a los resultados de la Tabla 6.

Ejemplos de muestras capturadas están en la Figura 44, autorizadas para publicar por sus propietarios:

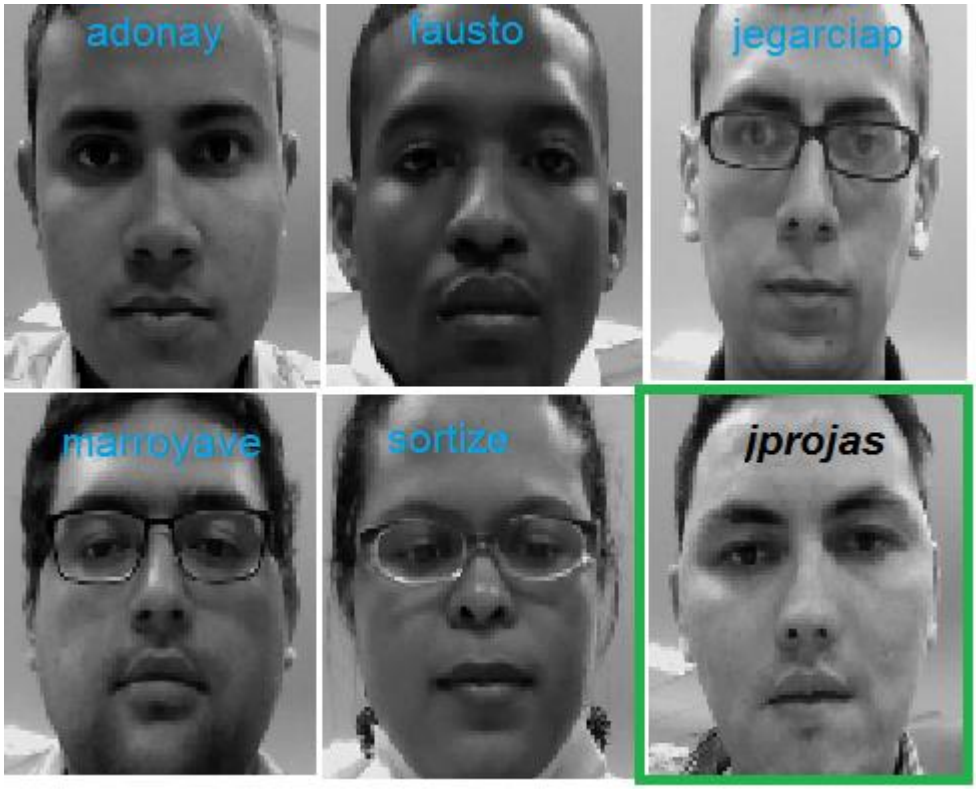


Figura 44: Ejemplos de capturas de detección facial, usadas para evaluar los algoritmos de EigenFaces, Fisherfaces y LBPH

4.4 Resultados

4.4.1 Acceso a Windows

En esta sección se muestra el objetivo principal de este trabajo de fin de master, donde un usuario ingresa a Windows con un token (password) solicitado a través del dispositivo móvil por medio del reconocimiento facial.

La Figura 45 permite visualizar que el usuario con sAMAccountName jprojas no ha asignado contraseña, ya que el campo pwdLastSet se muestra “unspecified”.

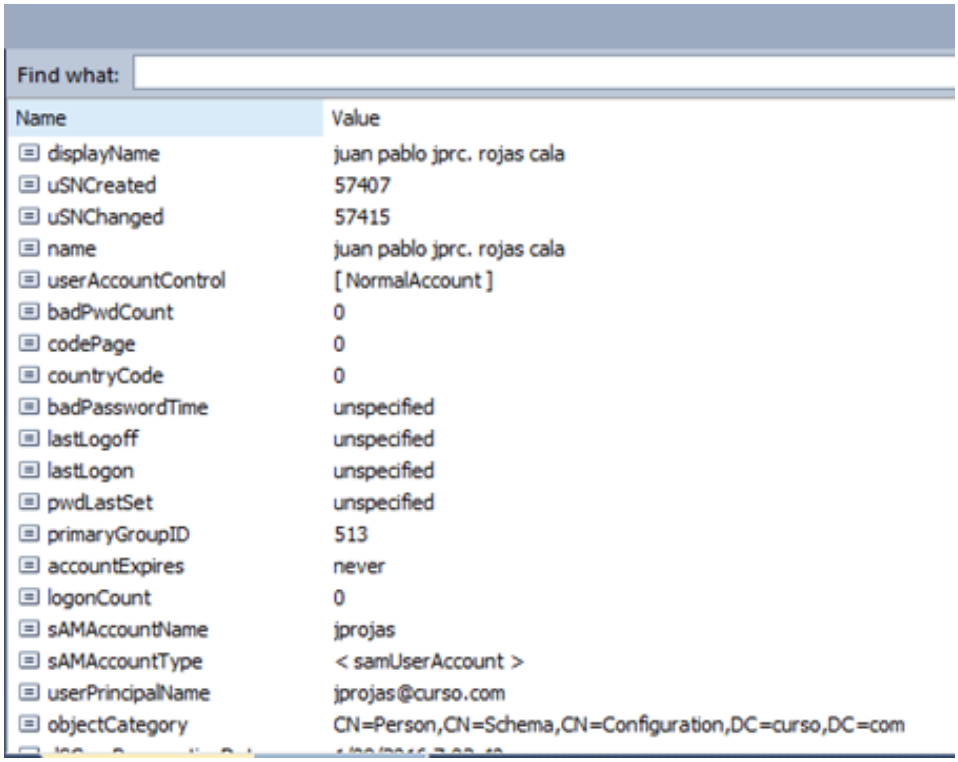


Figura 45: Estado de la cuenta del usuario jprojas antes de realizar el proceso de solicitar token

En la Figura 46, el usuario ha realizado el proceso de solicitud de token satisfactoriamente a través del dispositivo móvil, donde también se puede observar que la cuenta jprojas en Directorio Activo se le ha asignado una clave.

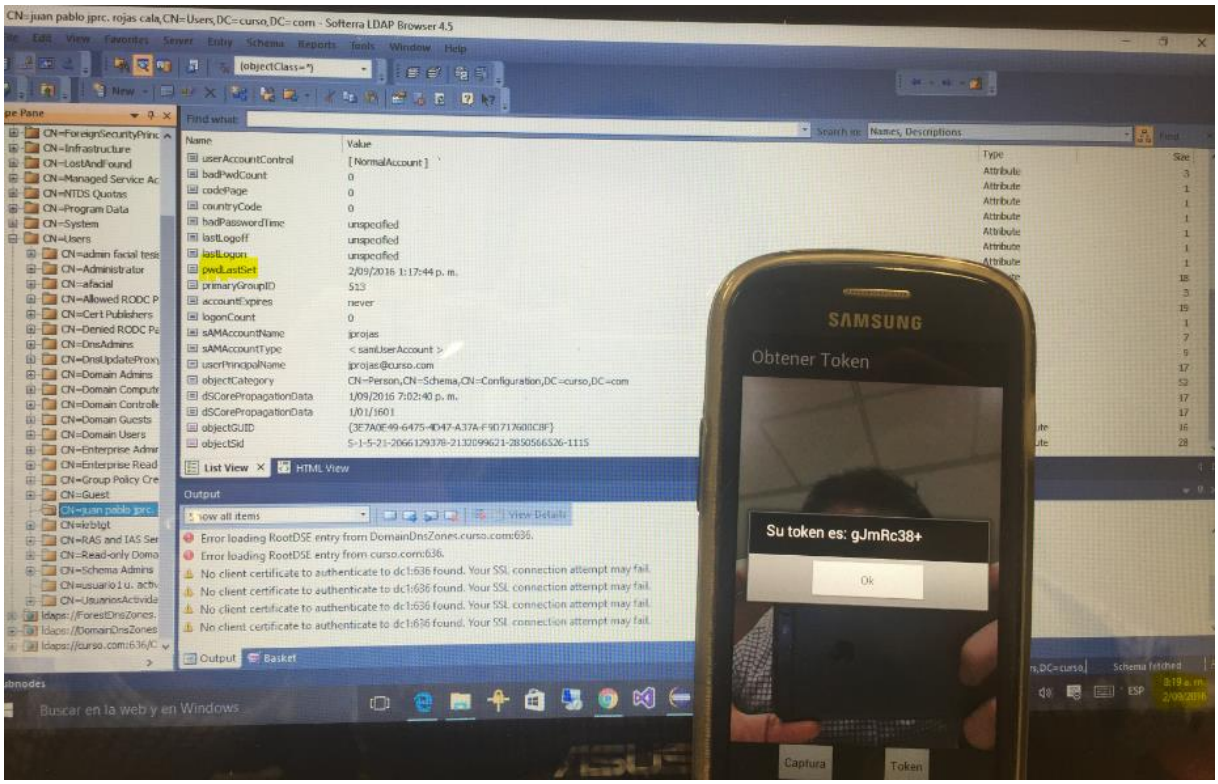


Figura 46: Figura donde se observa el token obtenido gJmRc38+ a través del proceso de solicitud de token

En la Figura 47 se observa que el usuario ingresará a Windows con el token proporcionado en la Figura 46.

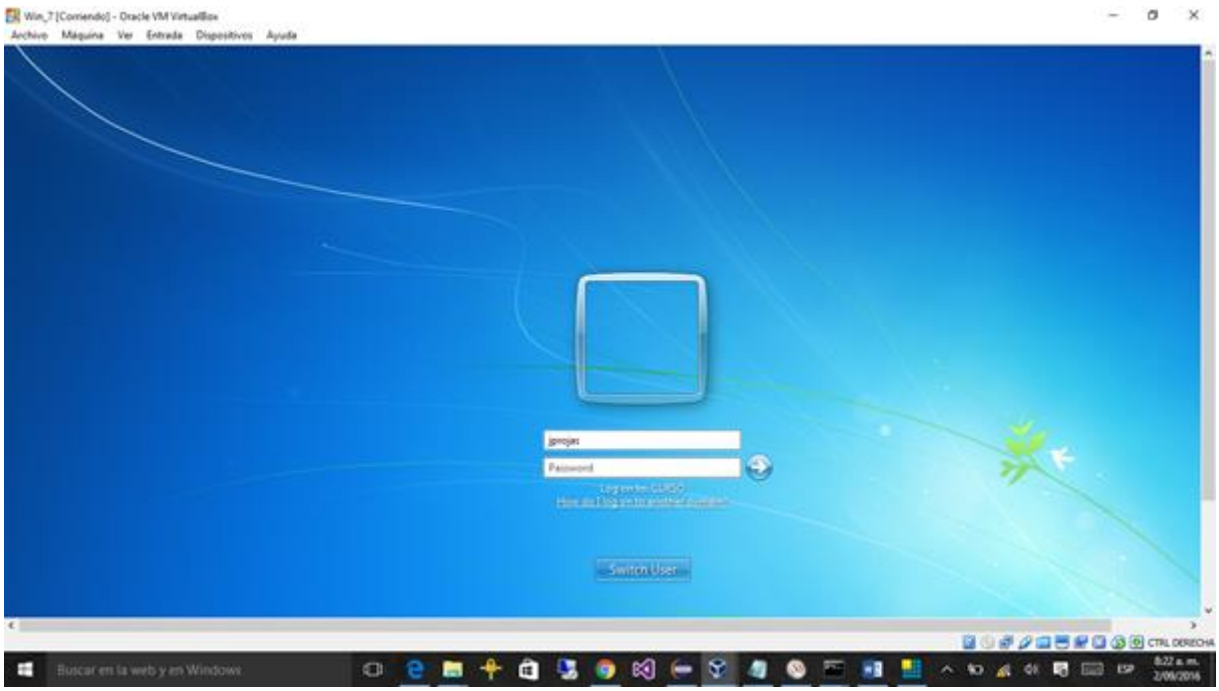


Figura 47: Pantalla de inicio de sesión a Windows, donde el usuario prueba el token

En la Figura 48 se ve el ingreso exitoso del usuario jprojas con el token gJmRc38+.

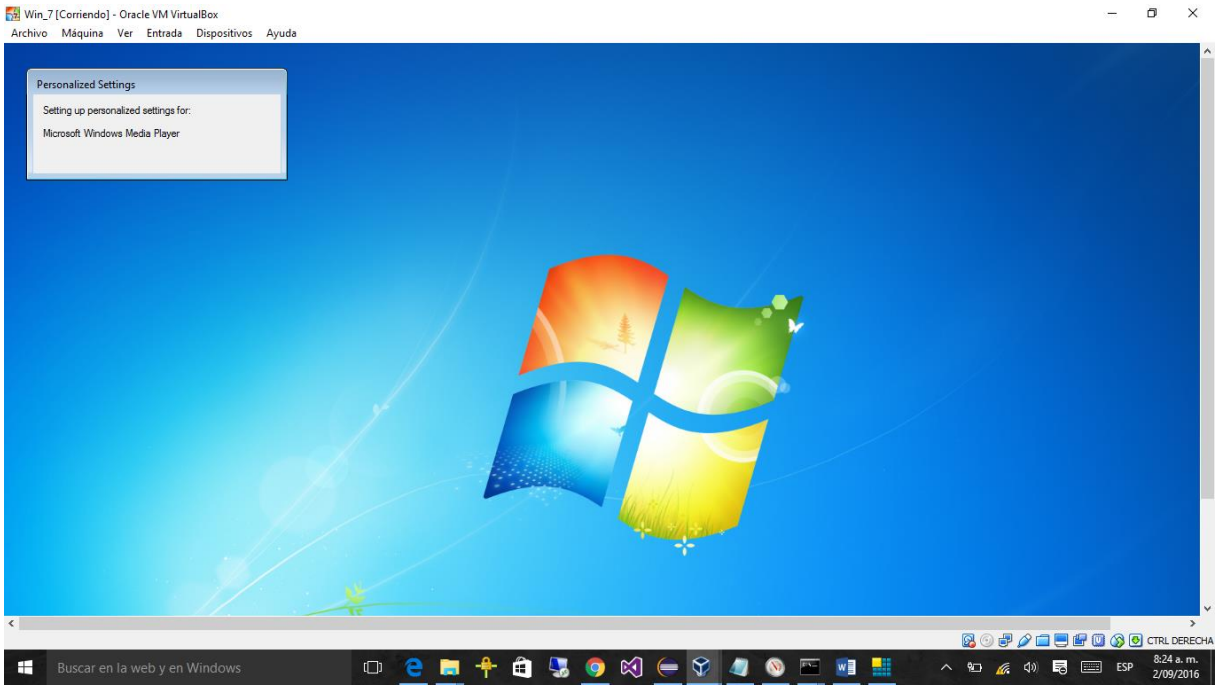


Figura 48: Ingreso exitoso del usuario a Windows

4.4.2 Rendimiento

En este apartado se mostrarán los resultados de rendimiento en términos de procesador y RAM, del proceso de solicitud token, el cual tendría una mayor demanda en el caso que se tuvieran miles de usuarios registrados en el sistema biométrico.

Las características del equipo donde se realizaron las pruebas es el descrito anteriormente, “Equipo Host” en la Tabla 5 “Características hardware y software de los componentes del diagrama de arquitectura”.

Para llevar a cabo esta prueba, se desarrolló una página php, donde se simula una solicitud de token, con datos ya quemados.

4.4.2.1 Prueba con 1 usuario

Primero veamos el rendimiento realizando una sola solicitud:

- Tiempo de respuesta: En la Figura 49 se puede observar a través de la herramienta de desarrollador de Google Chrome el tiempo de respuesta, campo “Time”, el cual da 6.43 segundos.

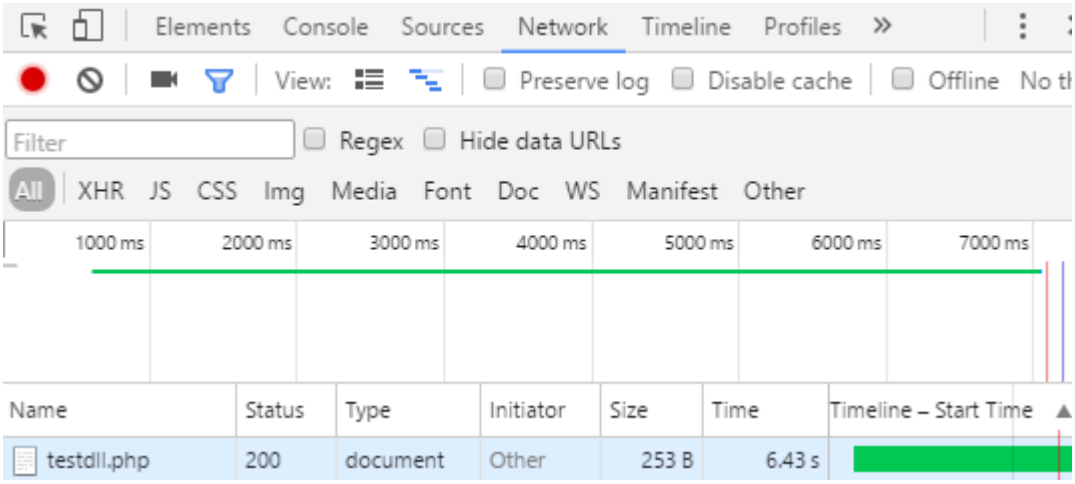


Figura 49: Tiempo de respuesta del proceso de solicitud de token

- Procesador, memoria RAM: En la Figura 50 se puede observar el consumo de CPU y RAM, datos obtenidos a través de la herramienta de Monitor de Rendimiento de Windows 10. El consumo de CPU se mantiene por encima del 60% y menos del 70%, la memoria RAM varía aproximadamente un 6%.

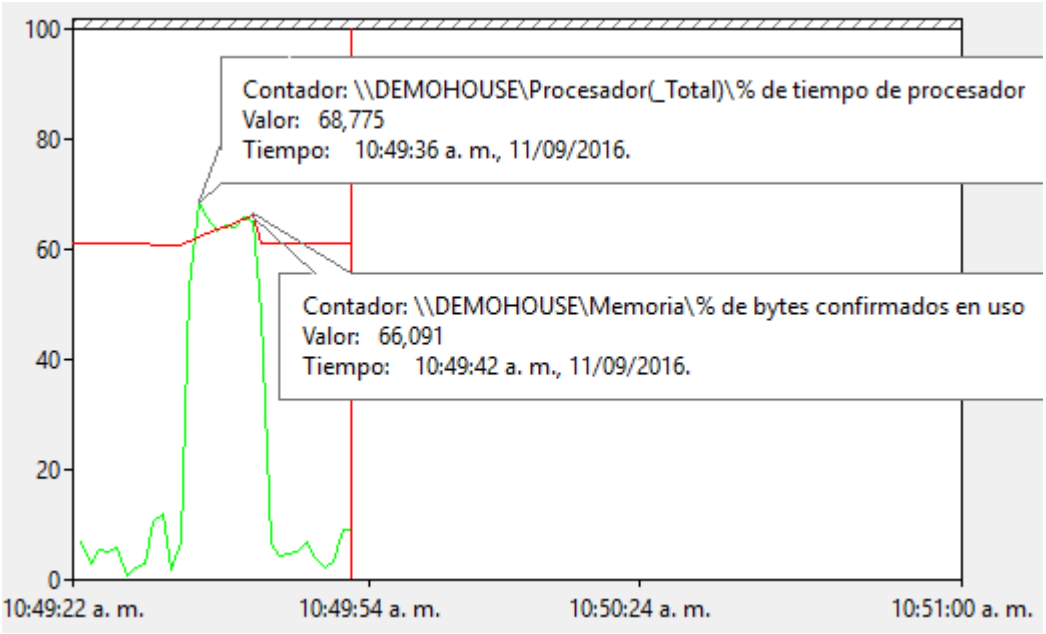


Figura 50: Consumo de CPU (Línea verde) y RAM (Línea roja) para 1 una solicitud de token

4.4.2.2 Prueba de límite de solicitudes

La siguiente prueba tiene como objetivo ver el rendimiento con más usuarios. Para realizar este estudio se usó el programa Apache JMeter, lanzando solicitudes a la vez, desde otro equipo en la misma WLAN, diferente al equipo Host donde se encuentra el servidor biométrico. Se realizaron pruebas con 5, 10, 20, 25, y 30 solicitudes.

De estas pruebas, se encuentra que las 25 solicitudes, serían el límite de procesamiento del servidor biométrico, para que estas sean procesadas satisfactoriamente. Dado que, como se puede observar en la Figura 51, los estados de respuesta (columna “Estado”) se muestran satisfactorios, en contraste, con 30 solicitudes el servidor no tiene la capacidad de responder, como se puede observar en la Figurar 53, donde los estados de respuesta se muestran no satisfactorios desde la petición 8 (columna “Muestra”). Las Figuras 52, 54 muestran el rendimiento de CPU y RAM, para ambos casos, en cuanto a la CPU, se puede observar que se encuentra al límite 100%, mientras que la memoria RAM aumenta de un 42% a un casi 70%, quiere decir que tuvo una variación aproximada de 28%.

De la Figura 51 también se puede decir que el peor tiempo de respuesta es de 70757 ms, correspondiente a la petición número 25 (muestra 25), equivalente a aproximadamente 1 minuto 11 segundos.

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muest...	Estado
1	14:18:40.241	Grupo de usuari...	Petición HTTP	13409	
2	14:18:40.289	Grupo de usuari...	Petición HTTP	13471	
3	14:18:40.203	Grupo de usuari...	Petición HTTP	13589	
4	14:18:40.322	Grupo de usuari...	Petición HTTP	13671	
5	14:18:40.399	Grupo de usuari...	Petición HTTP	37110	
6	14:18:40.370	Grupo de usuari...	Petición HTTP	37178	
7	14:18:40.447	Grupo de usuari...	Petición HTTP	46355	
8	14:18:40.482	Grupo de usuari...	Petición HTTP	47292	
9	14:18:40.660	Grupo de usuari...	Petición HTTP	47601	
10	14:18:40.615	Grupo de usuari...	Petición HTTP	47745	
11	14:18:40.528	Grupo de usuari...	Petición HTTP	47956	
12	14:18:40.582	Grupo de usuari...	Petición HTTP	48117	
13	14:18:40.692	Grupo de usuari...	Petición HTTP	48099	
14	14:18:40.740	Grupo de usuari...	Petición HTTP	54178	
15	14:18:40.817	Grupo de usuari...	Petición HTTP	54101	
16	14:18:40.771	Grupo de usuari...	Petición HTTP	54232	
17	14:18:40.975	Grupo de usuari...	Petición HTTP	67196	
18	14:18:40.895	Grupo de usuari...	Petición HTTP	67558	
19	14:18:40.942	Grupo de usuari...	Petición HTTP	67598	
20	14:18:40.849	Grupo de usuari...	Petición HTTP	67904	
21	14:18:41.054	Grupo de usuari...	Petición HTTP	70659	
22	14:18:41.104	Grupo de usuari...	Petición HTTP	70610	
23	14:18:41.023	Grupo de usuari...	Petición HTTP	70691	
24	14:18:41.134	Grupo de usuari...	Petición HTTP	70581	
25	14:18:41.181	Grupo de usuari...	Petición HTTP	70757	

Figura 51: Resultado de 25 peticiones lanzadas en paralelo al test de solicitud de token

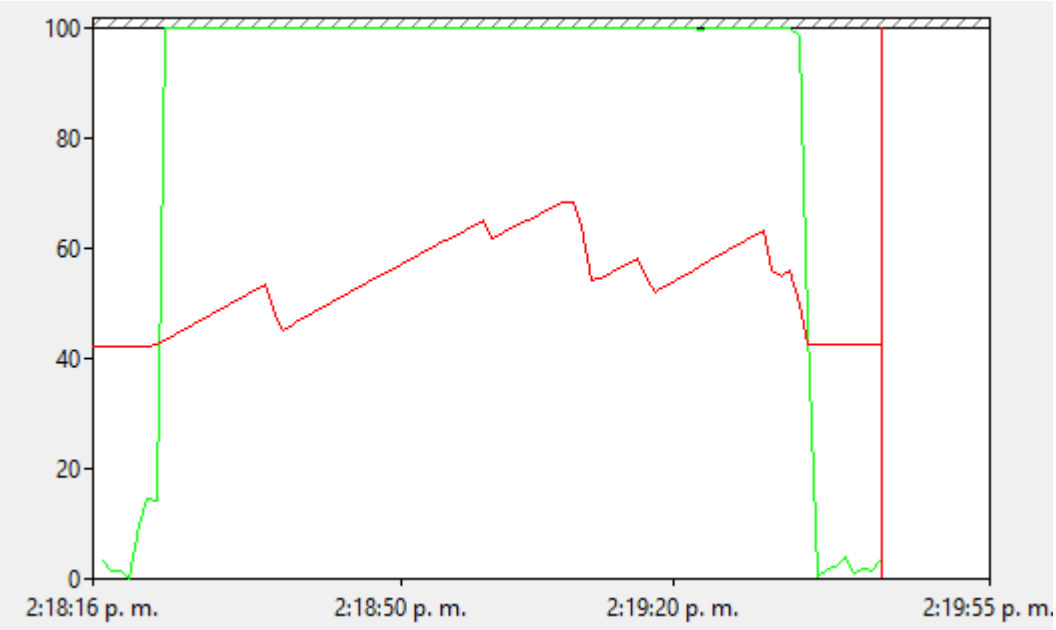


Figura 52: Consumo de CPU (Línea verde) y RAM (Línea roja) para 25 solicitudes de token

Muestra #	Tiempo de comi...	Nombre del hilo	Etiqueta	Tiempo de Muest...	Estado
1	14:21:55.522	Grupo de usuari...	Petición HTTP	22110	
2	14:21:55.481	Grupo de usuari...	Petición HTTP	22213	
3	14:21:55.555	Grupo de usuari...	Petición HTTP	22898	
4	14:21:55.586	Grupo de usuari...	Petición HTTP	23498	
5	14:21:55.621	Grupo de usuari...	Petición HTTP	24773	
6	14:21:55.654	Grupo de usuari...	Petición HTTP	25252	
7	14:21:55.688	Grupo de usuari...	Petición HTTP	27471	
8	14:21:55.925	Grupo de usuari...	Petición HTTP	69833	
9	14:21:56.093	Grupo de usuari...	Petición HTTP	69664	
10	14:21:55.792	Grupo de usuari...	Petición HTTP	69966	
11	14:21:55.895	Grupo de usuari...	Petición HTTP	69863	
12	14:21:56.361	Grupo de usuari...	Petición HTTP	69402	
13	14:21:56.393	Grupo de usuari...	Petición HTTP	69377	
14	14:21:56.026	Grupo de usuari...	Petición HTTP	69744	
15	14:21:55.825	Grupo de usuari...	Petición HTTP	69945	
16	14:21:56.158	Grupo de usuari...	Petición HTTP	69612	
17	14:21:56.326	Grupo de usuari...	Petición HTTP	69448	
18	14:21:55.723	Grupo de usuari...	Petición HTTP	70040	
19	14:21:56.227	Grupo de usuari...	Petición HTTP	69551	
20	14:21:56.194	Grupo de usuari...	Petición HTTP	69584	
21	14:21:56.427	Grupo de usuari...	Petición HTTP	69352	
22	14:21:56.461	Grupo de usuari...	Petición HTTP	69320	
23	14:21:56.293	Grupo de usuari...	Petición HTTP	69489	
24	14:21:55.861	Grupo de usuari...	Petición HTTP	69921	
25	14:21:56.060	Grupo de usuari...	Petición HTTP	69723	
26	14:21:56.127	Grupo de usuari...	Petición HTTP	69657	
27	14:21:55.992	Grupo de usuari...	Petición HTTP	69792	

Figura 53: Resultado de 30 peticiones lanzadas en paralelo al test de solicitud de token

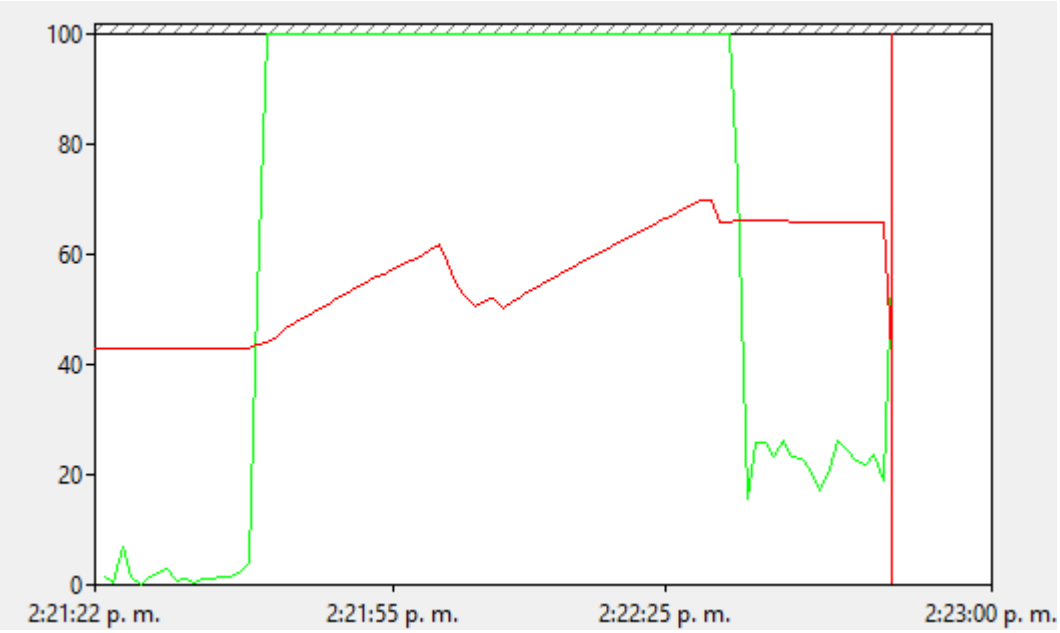


Figura 54: Consumo de CPU (Línea verde) y RAM (Línea roja) para 30 solicitudes de token

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En primera medida como conclusión principal, los resultados que se han mostrado anteriormente, han demostrado el cumplimento del objetivo principal y propósito de este trabajo de fin de master, ya que se comprobó que es realmente viable y factible desarrollar una autenticación en este caso, Ldap Windows, a través del reconocimiento facial, un mecanismo diferente al habitual donde los usuarios siempre tienen que memorizar una contraseña, y en el peor de los casos se olvidan y tienen que contactar al administrador o área que le ayuda a asignar una nueva clave. Hay que tener también en cuenta que las principales razones que hicieron viable este trabajo son:

- Las diferentes librerías disponibles para los algoritmos de reconocimiento facial, como en este caso la librería usada OpenCV, la cual demostró ser robusta y precisa con los algoritmos usados en este desarrollo de software.
- En especial los algoritmos usados de OpenCV no requieren una gran súper computadora para su ejecución, y para este desarrollo de software fue clave que se pudiera usar una computadora con especificaciones normales, es decir, no computadoras tipo servidor con grandes prestaciones.
- La mayoría de Smartphone (dispositivos móviles) cuentan con una cámara frontal, de modo que un usuario casi que por defecto puede tener acceso al software desarrollado.

Las siguientes son las contribuciones de cara a los objetivos específicos anteriormente planteados:

- Se tienen los diseños de los diferentes procesos en el cómo se debe realizar todo el proceso de autenticación por medio de reconocimiento facial, desde el proceso de registro hasta la forma como opera el servidor biométrico.
- Los algoritmos escogidos fueron evaluados en condiciones de ambiente real para determinar la eficacia y veracidad de los mismos.
- La extensión en php que se desarrolló, permite de una forma, visualizar en cómo se puede tener modularmente algoritmos de reconocimiento facial y a su vez otros tipos de algoritmos. Esto quiere decir, que, si se desea cambiar la librería de OpenCV por otra, solo se debería ir a esta extensión y desarrollarla con la otra librería deseada, sin impactar todo el diseño del software desarrollado. Existen muchas librerías y más algoritmos que incluso a nivel de investigación pueden ser fácilmente estudiados a través de este software modificando esta extensión de php.
- Se comprueba que dentro de los 3 algoritmos de reconocimiento facial Eigenfaces, Fisherfaces y LBPH, este último es el de mejor resultado.

Adicional se tienen las siguientes conclusiones:

- Los resultados de rendimiento del proceso de solicitud de token usando el algoritmo LBPH de la librería OpenCV, son muy buenos, ya que nos dice que con 25 usuarios solicitando tokens a la vez, el sistema biométrico puede responder a cada uno su solicitud. Se consideran buenos resultados, dado que desde luego habrá muchos usuarios realizando peticiones al servidor, sin embargo, esto no indica que sean en simultaneo (a la vez).
- La aplicación móvil permite al usuario llevar a cabo el proceso biométrico fácilmente, debido a que la aplicación divide el proceso en 4 pasos sencillos: Registro fotográfico, Preparar imágenes, Registrar en servidor, Obtener token de acceso. Esto también, gracias al algoritmo de detección facial, donde el usuario no tiene la necesidad de estar tratando de ubicar su rostro en alguna cuadrícula, ya que por el contrario el usuario solo tiene que tomarse una foto de sí mismo (selfie) lo cual actualmente hacen a menudo con sus dispositivos móviles.
- Gracias a la centralización del proceso de reconocimiento facial en el servidor biométrico, este puede ser cambiado por otro, siendo transparente para los usuarios registrados en el servidor biométrico, estos no se darían cuenta que se está usando determinado algoritmo.

- Sólo en caso que la toma del registro biométrico que se realiza desde el dispositivo móvil se modifique, es necesario actualizar las respectivas aplicaciones de los usuarios.

5.2 Trabajo Futuro

Las siguientes son los posibles trabajos que se pueden llevar a cabo a partir de este trabajo de fin de master realizado:

- El software que se desarrolló, realmente puede llegar a ser usado a gran escala en una empresa, se tendría que mejorar un poco más en términos de realizar más pruebas, mejorar las interfaces, quizás mejorar los mecanismos de seguridad y de esta forma llegar a tener un software más robusto y confiable.
- Se conoce bien que los algoritmos de reconocimiento facial son susceptibles a la vulnerabilidad de spoofing, donde se pueden hacer reconocimiento sobre imágenes digitales y no precisamente directamente de la persona. Se podría investigar más sobre este tema y ver si es posible implementar algún mecanismo que permita solventar esta vulnerabilidad.
- A nivel de investigación como se mencionó antes, se puede realizar un estudio de más algoritmos, de forma que se pueda siempre llegar a realizar diferentes comparaciones de una forma más ágil al apoyarse en este software desarrollado.
- La autenticación se podría llevar no solo a una autenticación de servidor Ldap Windows, si no a otros sistemas, aplicaciones propias o comerciales de una empresa.
- Involucrar más datos de prueba, esto quiere decir, más sujetos, ya que uno de los temas más complejos es, contar con personal para llevar a cabo todas las distintas pruebas, por lo general no siempre están disponibles. Una ayuda a esto sería, publicar la aplicación AFacial para Android en Google Play, para que de esta forma los sujetos de prueba la descarguen fácilmente y en caso tal, siempre cuenten con la última versión. Una vez los sujetos cuenten con la aplicación en sus equipos y procedan a realizar las diferentes pruebas, el servidor biométrico desarrollado ayudará a través del log de autenticación a ver los diferentes resultados del algoritmo de detección facial.
- Un proyecto interesante sería llevar a un nivel más el reconocimiento facial. Se podría realizar un reconocimiento por una secuencia de gestos, como por ejemplo una secuencia sería: sonrisa, guiño ojo izquierdo, ojo derecho. Esto a través, de los algoritmos de reconocimiento de gestos. La librería de OpenCV en su versión 3.1.0 provee algoritmos de reconocimiento de gestos.

Desde luego hay cosas por mejorar como, por ejemplo:

- En términos de desarrollo, la extensión desarrollada para php en el Zend Engine, es complicada de manejar, debido a que depende mucho de la versión de PHP, dando así problemas de compatibilidad. Lo ideal sería usar algún wrapper con el objetivo de que el desarrollo sea compatible con distintas versiones de php.
- De alguna forma, el nivel de confianza con el algoritmo LBPH al que se llegó en este trabajo, puede ser más bajo, implementando quizás una mejor extracción facial donde exista un menor ruido posible, como por ejemplo en la Figura 21, donde se puede observar que se ve la camisa del sujeto. También podría implementarse una corrección de orientación del rostro detectado, como se explica en la sección "Aligning Face Images" del tutorial "Face Recognition with OpenCV", sería muy interesante ver qué resultados se obtendrían.

Implementar este software a nivel de una gran empresa donde podría haber miles de usuarios se tendrían las siguientes ventajas adicionales a las comentadas anteriormente al inicio del trabajo apartado "Motivación":

- En caso de un intento de intrusión por medio del dispositivo móvil, esta quedaría registrada en el servidor biométrico, donde se obtendría la foto del intruso y se podría llevar a cabo una investigación judicial con un dato veraz. Hoy en día los sistemas de auditoría de ingreso a sistemas carecen de esto, no se cuenta con un dato biométrico como el rostro del intruso.
- El acceso al sistema biométrico puede ser restringido a usar una red WLAN propia de la empresa, o desde la VPN de la empresa, dando así más seguridad en caso de que el dispositivo móvil sea robado, ya que el intruso primero tendría que estar cerca de esta.
- El soporte es fácilmente gestionado, ya que las máquinas o estaciones de trabajo no requieren programa especial alguno, todo se gestiona a través de los dispositivos móviles y el servidor biométrico.

Y por consiguiente las siguientes desventajas:

- Riesgo de intrusión, para que no ocurra, el algoritmo de reconocimiento tiene que estar suficientemente probado en caso que se presente un acceso no deseado por un intruso, igualmente este trabajo muestra cómo se puede evaluar y brinda el log de auditoría para llevar a cabo un estudio más riguroso del algoritmo.
- Aún sigue siendo una desventaja el spoofing en el campo del reconocimiento facial, sin embargo, como es a nivel empresarial, sería raro o sospechoso que un empleado

vea a otro llevando a cabo este tipo de ataque, ya que tendría que hacer toda una labor para lograr su objetivo, en el mejor de los casos sería alertado por otro empleado.

6 Bibliografía

Jaime, V., Uribe, M., Hernández C. (septiembre, 2015). Sistemas Biométricos en los Dispositivos Móviles. En A. Apellido del Presidente del Congreso (Presidencia), Simposio Iberoamericano Multidisciplinario de Ciencias e Ingenierías. Simposio dirigido por Universidad Politécnica de Pachuca, México.

Consejo Nacional de Ciencia y Tecnología (NSTC) Gobierno de los Estados Unidos, Subcomité de biometría. (agosto, 2006). Face Recognition. Recuperado el 7 de julio de 2016 de http://www.biometrics.gov/Documents/facerec.pdf.

Biometría - Facial. (2016). Biometria.gov.ar. Recuperado el 8 de julio de http://www.biometria.gov.ar/metodos-biometricos/facial.aspx.

Facial recognition system, (s.f). En Wikipedia. Recuperado el 10 de julio de 2016 de https://en.wikipedia.org/wiki/Facial_recognition_system.

Face Recognition Vendor Test. (mayo, 2016). Nist.gov. Recuperado el 13 de julio de 2016 de http://www.nist.gov/itl/iad/ig/frvt-home.cfm.

Face Recognition Vendor Test (FRVT) 2002. (2016). Nist.gov. Recuperado el 13 de julio de 2016 de http://www.nist.gov/itl/iad/ig/frvt-2002.cfm.

Face Recognition Vendor Test 2006. (2016). Nist.gov. Recuperado el 14 de julio de 2016 http://www.nist.gov/itl/iad/ig/frvt-2006.cfm.

Phillips, P., Scruggs, W., O'Toole, A., Flynn, P., Bowyer, K., Schott, C., & Sharpe, M. (2007). FRVT 2006 and ICE 2006 Large-Scale Results. Gaithersburg: NIST Internal or Interagency Reports. Recuperado el 15 de julio de http://www.nist.gov/customcf/get_pdf.cfm?pub_id=51131.

FRVT 2013. (2016). Nist.gov. Recuperado el 14 de julio de 2016 http://www.nist.gov/itl/iad/ig/frvt-2013.cfm.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (diciembre, 2003). Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4), 399-458. Recuperado el 13 de agosto

de 2016 http://nichol.as/papers/ZHAO/Face%20Recognition:%20A%20Literature%20Survey.pdf.

Automated Facial Recognition in the Public and Private Sectors. (2016). Priv.gc.ca. Recuperado el 13 de agosto de 2016 de https://www.priv.gc.ca/information/research-recherche/2013/fr_201303_e.asp.

Rogers, K. (febrero, 2016). That Time the Super Bowl Secretly Used Facial Recognition Software on Fans. (2016). Motherboard. Recuperado el 13 de agosto de 2016 de http://motherboard.vice.com/read/that-time-the-super-bowl-secretly-used-facial-recognition-software-on-fans.

50 Face Recognition APIs. (2016). Datasciencecentral.com. Recuperado el 13 de agosto de 2016 de http://www.datasciencecentral.com/profiles/blogs/50-face-recognition-apis.

ABOUT | OpenCV. (2016). Opencv.org. Recuperado el 14 de agosto de 2016 de http://opencv.org/about.html.

Face Recognition with OpenCV — OpenCV 2.4.13.0 documentation. (2016). Docs.opencv.org. Recuperado el 21 de agosto de 2016 de http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html.

gihantharanga/ComputerVision. (2016). GitHub. Recuperado el 22 de agosto de 2016 de https://github.com/gihantharanga/ComputerVision.

Face Recognition Homepage - Databases. (2016). Face-rec.org. Recuperado el 22 de agosto de 2016 de from http://www.face-rec.org/databases/.

OpenCV: Face Detection using Haar Cascades. (2016). Docs.opencv.org. Recuperado el 22 de agosto de 2016 de http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0.

Wang, Y. (2014). An Analysis of the Viola-Jones Face Detection Algorithm. Image Processing On Line, 4, 128-148. http://dx.doi.org/10.5201/ipol.2014.104.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. En Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511). IEEE.

Hu, J., Peng, L., & Zheng, L. (agosto, 2015). XFace: A Face Recognition System for Android Mobile Phones. In Cyber-Physical Systems, Networks, and Applications (CPSNA), 2015 IEEE 3rd International Conference on (pp. 13-18). IEEE.

Explainer: Verification vs. Identification Systems. (2012). BiometricUpdate. Recuperado el 8 de septiembre de 2016 de http://www.biometricupdate.com/201206/explainer-verification-vs-identification-systems.

[advancesourcode]. (diciembre, 2013). Biometric Recognition, Verification, Identification and Authentication: which one?. [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=ue5yRqlle-M.

PHP: La API Zend: Hackeando el núcleo de PHP - Manual. (2016). Php.net. Recuperado el 11 de septiembre de 2016 de http://php.net/manual/es/internals2.ze1.zendapi.php.

SWIG and PHP. (2016). Swig.org. Recuperado el 12 de septiembre de 2016 de http://www.swig.org/Doc1.3/Php.html.

PHP-CPP - A C++ library for developing PHP extensions, (2016). Php-cpp.com. Recuperado el 12 de septiembre de 2016 de http://www.php-cpp.com/.

Marqués, A., (2016) Conceptos sobre APIs REST. Recuperado el 12 de septiembre de 2016 de http://asiermarques.com/2013/conceptos-sobre-apis-rest/.

Navarro, R. (julio, 2006). REST vs Web Services. Universidad de Valencia. Recuperado el 12 de septiembre de 2016 de http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf.