AUTENTICACIÓN POR RECONOCIMIENTO FACIAL PARA APLICACIONES WEB, UTILIZANDO SOFTWARE LIBRE.

CARLOS GUILLERMO NOGUERA ANDRADE

UNIVERSIDAD PONTIFICIA BOLIVARIANA

FACULTAD DE INGENIERÍA INFORMÁTICA,

ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA,

BUCARAMANGA

2012

AUTENTICACIÓN POR RECONOCIMIENTO FACIAL PARA APLICACIONES WEB, UTILIZANDO SOFTWARE LIBRE.

CARLOS GUILLERMO NOGUERA ANDRADE

Monografía

Directora:

Dra. Isabel Cristina Satizábal Echavarría

UNIVERSIDAD PONTIFICIA BOLIVARIANA

FACULTAD DE INGENIERÍA INFORMÁTICA,

ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA,

BUCARAMANGA

2012

AGRADECIMIENTOS

A Dios, quien me ha brindado de todas
las bendiciones en la vida

A mis padres, quienes me han apoyado
incondicionalmente y formado como
una persona íntegra

Familiares, amigos y compañeros quienes
de manera directa o indirectamente
me han brindado su apoyo

A la directora de proyecto, profesores y a la universidad que
gracias a su trabajo, esfuerzo, paciencia y dedicación
han contribuido a mi formación
académica y personal

Y a todos aquellos quienes no están
ahora con nosotros les recordare
siempre por su apoyo y cariño

Mis más sinceras gracias.

TABLA DE CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: AUTENTICACIÓN POR RECONOCIMIENTO FACIAL PARA APLICACIONES WEB, UTILIZANDO SOFTWARE LIBRE.

AUTOR(ES): CARLOS GUILLERMO NOGUERA ANDRADE

FACULTAD: Facultad de Ingeniería Informática

DIRECTOR(A): Isabel Cristina Satizábal Echavarría

RESUMEN

En la Web, la autenticación juega un papel importante ya que permite identificar adecuadamente a los usuarios antes de que estos accedan a los recursos y servicios. Sin embargo, el mecanismo de autenticación más usado, actualmente, es usuario/contraseña, un mecanismo que al basarse en un único factor de autenticación (algo que se conoce) es débil, por lo que se puede suplantar fácilmente la identidad de los usuarios y realizar acciones en su nombre. El presente trabajo consiste en el desarrollo de un prototipo de autenticación fuerte, usando software libre, basado en dos factores: algo que se conoce (contraseña) y algo que se es (rasgos faciales) para aplicaciones Web desarrolladas en PHP, con el fin de evitar la suplantación de identidad. Los datos entre el cliente y el servidor viajan de manera cifrada. Se implementan dos métodos conocidos de reconocimiento de patrones *eigenfaces* y *fisherfaces*, se demostró la robustez y rendimiento del sistema propuesto a diferentes condiciones de iluminación y resolución de cámara. Se realizaron diferentes ataques al modulo de autenticación biométrico y se demostró sus deficiencias como método de autenticación web, finalmente se analizaron la importancia del uso de la biometría multimodal y el uso de buenas prácticas para el aseguramiento de equipos de computación.

PALABRAS CLAVES:

Autenticación biométrica, eigenfaces, fisherfaces, seguridad Web, reconocimiento facial, software libre.

GENERAL SUMMARY OF WORK OF GRADE

TITLE: FACIAL RECOGNITION AUTHENTICATION FOR WEB APPLICATIONS, USING OPEN SOURCE.

AUTHOR(S): CARLOS GUILLERMO NOGUERA ANDRADE

FACULTY: Facultad de Ingeniería Informática

DIRECTOR: Isabel Cristina Satizábal Echavarría

ABSTRACT

On the Web, authentication plays an important role since it allows to properly identifying users before accessing these resources and services. However, the most widely used authentication mechanism is currently a username / password, a mechanism to be based on a single factor authentication (something you know) is weak, so that you can easily impersonate the identity of users and perform actions on their behalf. The present work is the development of a prototype of strong authentication, using open source, based on two factors: something you know (password) and something that is (facial features) for web applications developed in PHP, to prevent spoofing. The data between the client and the server in encrypted traveling. Implemented two methods known pattern recognition eigenfaces and fisherfaces, demonstrated the robustness and performance of the proposed system to different lighting conditions and camera resolution. There have been various attacks on biometric authentication module and demonstrated its shortcomings as web authentication method, finally analyzed the importance of the use of multimodal biometrics and the use of best practices for securing computer equipment.

KEYWORDS:

Biometric authentication, eigenfaces, fisherfaces, Web security, facial recognition, open source.

INTRODUCCIÓN

Las tecnologías Web han tenido un gran crecimiento en los últimos años, al punto de convertirse en parte del día a día. Las tecnologías de la información brindan la posibilidad de acceder remotamente, y de manera virtual, desde cualquier lugar del mundo, a Internet, permitiendo la comunicación en tiempo real y a bajo costo. Sin embargo, dado que Internet es una red pública, se deben garantizar los servicios básicos de seguridad, como son: disponibilidad, confidencialidad, integridad y autenticación, pero dadas las limitaciones en las tecnologías Web actuales, es una tarea ardua el poder garantizar la completa seguridad en ellas. La autenticación permite validar la identidad de un usuario que desee ingresar a un sistema. Este proceso es vital en plataformas Web, ya que permite asegurar el correcto ingreso de usuarios autorizados, pues si algún tercero ingresa usurpando la identidad de un usuario autorizado puede comprometer la seguridad de todo el sistema. Existen diversas tecnologías que emplean diferentes factores para verificar la identidad de los usuarios y hacen más difícil la tarea de la usurpación. Una de las más fiables es la biometría. La tecnología biométrica permite identificar a un usuario por sus características físicas y/o de comportamiento, y al ser usada en plataformas Web permite que la autenticación de usuarios sea más fiable, ya que a través de Internet, se puede suplantar fácilmente a otra persona.

El presente trabajo se refiere al desarrollo de un prototipo de autenticación fuerte basado en dos factores: algo que se conoce (contraseña) y algo que se es (rasgos faciales), donde los datos entre el cliente y el servidor viajan de manera cifrada, con el fin de evitar la suplantación de identidad en aplicaciones Web, haciendo uso de software libre.

La investigación se estructura de la siguiente manera: El primer capítulo define el problema y la justificación. En el segundo capítulo se presentan los objetivos del proyecto. El tercer capítulo contiene las bases teóricas que sustentan la investigación. El cuarto capítulo presenta el diseño del prototipo. En el quinto capítulo se muestran los resultados obtenidos tras las pruebas y en el sexto capítulo se presentan las conclusiones y recomendaciones.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 ESPECIFICACIÓN DE LA SITUACIÓN PROBLEMÁTICA

En la Web, la autenticación juega un papel importante ya que permite identificar adecuadamente a los usuarios antes de que estos accedan a los recursos y servicios. Sin un mecanismo de autenticación apropiado, se puede suplantar fácilmente la identidad de los usuarios y realizar acciones en su nombre.

Las técnicas de autenticación de usuario pueden clasificarse en tres factores, dependiendo de lo que se deba presentar para demostrar la identidad: algo que se conoce (contraseña, PIN, etc.), algo que se tiene (tarjeta inteligente, *token*, etc.), algo que se es o característica biométrica (huella digital, voz, retina, iris, etc.) [1]. Si el usuario utiliza dos o más de estos factores para identificarse ante algún sistema, la autenticación es fuerte y si se utiliza sólo uno de ellos, la autenticación es débil.

Cada uno de los factores de autenticación tiene sus consideraciones para poder ser usado. El primero de ellos (algo que se conoce), es el más simple de implementar y el más usado (usuario/contraseña), pero también el más vulnerable, ya que, incluso haciendo uso de buenas prácticas, como contraseñas largas y complejas, no se evita que el usuario sea víctima de suplantación porque algún atacante obtuvo su contraseña [2]. El segundo de ellos (algo que se tiene), brinda una mejor protección al hacer uso de un elemento externo que identifica al usuario [3]. Su implementación es más compleja que el método anterior y es usado ampliamente en la actualidad, como en el caso de las tarjetas de crédito. Este se suele combinar con el primer factor, para ofrecer mayor seguridad. Sin embargo, los usuarios pueden perder este elemento externo, no cargarlo siempre con ellos o ser víctimas de algún robo.

El último factor de autenticación (algo que se es) es el que presenta mayor complejidad de implementación y es usado principalmente para el control de acceso a nivel local, debido al requerimiento de hardware especial para la captura de los datos biométricos y además, los datos biométricos no pueden viajar en claro por la red, pues si un atacante obtuviera esta información, al ser estas características inherentes a cada ser humano, no se podrían

cambiar con facilidad y la autenticación ya no sería segura. Esto hace que no sea usual su uso a nivel Web. No obstante, el uso de este método de autenticación en la Web permitiría asegurar que al otro lado de la comunicación se encuentra un ser humano y no una máquina, y se haría más difícil suplantar a alguien. Además, no existen los problemas mencionados en el segundo factor (algo que se tiene). Para evitar el problema de adquirir hardware muy especializado para la captura de datos biométricos, se ha pensado en el uso de las cámaras Web, que son habituales en los computadores y dispositivos móviles actuales. La pregunta de investigación es: ¿Cómo implementar un mecanismo de autenticación biométrica por reconocimiento facial para aplicaciones Web, utilizando únicamente software libre?.

La autenticación biométrica es un campo de investigación que lleva más de 30 años de desarrollo y se han realizado importantes avances en esta área [4][5][6]. La autenticación biométrica orientada específicamente al ambiente Web aún se encuentra en desarrollo, pero se han presentado diferentes propuestas en este campo, entre algunas de ellas están:

- Otero et al. [7] presentan un modelo de seguridad para el control de acceso a servicios Web, en plataformas java, mediante la implementación de autenticación biométrica usando BioApi[8] y *Central Authentication Service* (CAS). Se plantea un esquema cliente – servidor, cuyo software cliente captura datos biométricos, como rasgos faciales, voz, entre otros.

- Franco et al. [9] presentan una solución de autenticación biométrica para resolver el problema de suplantación de identidad de los *Learning Management Systems* (LMS), mediante el diseño de una arquitectura de software libre para la validación de identidad de los estudiantes que usan el LMS MOODLE, utilizando tecnología de Macromedia FLEX del lado del cliente y de PHP del lado del servidor. Plantea una arquitectura de reconocimiento facial, distribuida entre el cliente embebido y el servidor Web. Esta permite que el cliente se encuentre embebido dentro de la página de inicio y no haya necesidad de instalar ningún software adicional para lograr la autenticación. Su inconveniente es que a pesar de ser un proyecto de investigación de software libre, al momento de realizar esta investigación no se logró encontrar el

código fuente del mismo.

- En el trabajo de investigación realizado por González et al. [10] se presenta una solución orientada a plataformas Web de tele-enseñanza en la cual, mediante el uso de autenticación biométrica multimodal (rasgos faciales, voz, comportamiento y otras), se realiza el control de acceso y monitoreo de los usuarios, a fin de evitar fraude en la acreditación de cursos. Esta propuesta usa la librería OpenCV[11], Java, BioApi y CAS. La investigación hace parte del proyecto PRESA[12], que se centra en el reconocimiento de personas en escenarios reales, mediante computación distribuida con tecnología biométrica multimodal[1], brindando una completa referencia de una arquitectura de control de acceso biométrico orientado a plataformas Web. El inconveniente de esta solución es que depende de la descarga de un software cliente para su funcionamiento.

- C. Ruchir [13] presenta un modelo de autenticación biométrica para plataformas Web J2EE haciendo uso de CAS, *Java Authentication and Authorization Service* (JAAS) y BioAPI. Este modelo permite el uso de dispositivos biométricos del lado del cliente para capturar las características faciales del usuario y lograr la autenticación. El autor enfoca este modelo de seguridad para que se use ampliamente a nivel global, en aplicaciones Web que implementen control de acceso robusto.

- La compañía bioID[14] ofrece un servicio propietario para la autenticación biométrica en servicios Web y plataformas móviles, por medio de una foto personal o la captura de la *Webcam*, a través de un cliente embebido en el navegador Web. Hace uso de los rasgos faciales para lograr identificar al usuario e implementa tecnología OpenID, por lo que cualquier sitio Web que implemente OpenID puede hacer uso de este servicio de autenticación biométrica.

Las propuestas de software libre encontradas son para plataformas Java y dependen de la descarga de algún software cliente para su operación; otras son propuestas propietarias y no

[1] Cuando se usan dos o más características biométricas para identificar a una persona se denomina biometría multimodal.[1]

brindan información adicional referente a sus servicios.

En este proyecto se pretende implementar un prototipo de autenticación fuerte, basado en los factores: algo que se conoce (contraseña) y algo que se es (rasgos faciales), donde los datos entre el cliente y el servidor viajen de manera cifrada, con el fin de evitar la suplantación de identidad en aplicaciones Web, todo esto a través de software libre, lo que posibilitará su uso extendido en Internet.

1.2 JUSTIFICACIÓN

Estadísticas en el campo de la seguridad informática muestran un incremento sustancial de incidentes relacionados al robo de información a través de los últimos años. Symantec en su reporte de amenazas de seguridad en Internet, publicado en abril de 2011, informa que la cantidad de ataques a sitios Web, durante el 2010, aumentó en un 93% con respecto al 2009 [15]; por su parte RSA, en su reporte de fraude en línea de 2012 [16], muestra que ataques Web como *phishing* incrementaron en el 2011 cerca de un 37% en relación al año 2010. Observando esta tendencia, cientos de sitios Web fraudulentos son creados a diario, por atacantes, con el propósito de robar las credenciales de los usuarios y así poder suplantar su identidad. Cerca de un promedio de 500 ataques diarios de *phishing* son reportados públicamente a través de la comunidad *Phishtank* y entre los objetivos más comunes se encuentran *PayPal*, *Facebook*, *Mastercard*, *Aol*, entre otros [17]. Este tipo de ataque genera pérdidas estimadas en millones de dólares, junto con la pérdida de reputación y deterioro de la confianza en la red. Entre los sectores más afectados dentro de la industria se encuentran la banca, sistemas de pago en línea [18] y redes sociales.

Estadísticas internacionales [19] indican que entre el 2006 y el 2009, la cantidad de casos reportados, referentes al robo de identidad, en Estados Unidos, se incrementó de 8.4 a 11.1 millones de personas, y se estimaron pérdidas que ascienden a los 54 mil millones de dólares. Estadísticas realizadas en América Latina [20], estimaron que en Argentina, durante el año 2005, se realizaron más de 1700 denuncias por robo de identidad; en Ecuador, en el año 2009, se realizaron 891 denuncias asociadas a este delito; y en México,

durante el año de 2010, se registraron cerca de 300 mil denuncias [21]. Debido a este emergente problema, se debe contar con mecanismos que complementen a los sistemas actuales de autenticación, a fin de minimizar los incidentes asociados al robo de identidad.

Este proyecto busca proponer un mecanismo que permita garantizar el ingreso de usuarios legítimos en plataformas Web, es decir, que los usuarios sean realmente quienes dicen ser, por medio de la implementación de biometría de reconocimiento facial. Este tipo de autenticación biométrica es un método pasivo y no invasivo, ya que es una característica común de cualquier ser humano y no requiere de hardware especial para la captura de imágenes, puesto que, sólo es necesaria una Webcam, dispositivo que es accesible y tiene un bajo costo de adquisición, además que es cada vez más común que los computadores personales traigan incorporado este dispositivo.

Al hacer uso de la biometría, se asegura que al otro lado de la comunicación se encuentre realmente un ser humano y no una máquina, y al ser características propias de los seres humanos, no serán imitadas de manera fácil.

En este proyecto, se hará uso de software libre, lo que permitirá que sea accesible a toda la comunidad de Internet y pueda servir como punto de referencia a futuros trabajos en el área de la seguridad informática, a fin de brindar una solución para minimizar el problema del robo y suplantación de identidad en plataformas Web.

2. OBJETIVOS

2.1 Objetivo General

Implementar un prototipo de autenticación biométrica por reconocimiento facial, fuerte y seguro, utilizando software libre, que pueda ser usado en aplicaciones Web.

2.2 Objetivos Específicos

- Realizar un análisis comparativo de las diferentes propuestas existentes en el área de la autenticación biométrica para aplicaciones Web, a fin de contar con referentes claros en el tema de investigación y obtener un conjunto de características deseables en el prototipo a implementar.
- Diseñar un mecanismo de autenticación biométrica por reconocimiento facial para aplicaciones Web, que incluya al menos dos factores de autenticación y que ofrezca una comunicación segura entre cliente y servidor.
- Implementar un prototipo del mecanismo de autenticación diseñado, utilizando software libre.
- Evaluar la usabilidad, nivel de seguridad y resistencia a ataques de robo de identidad del prototipo implementado, para detectar posibles fallos y realizar las mejoras necesarias.

3. MARCO TEÓRICO

3.1 SEGURIDAD EN APLICACIONES WEB

Dada la importancia de los datos transmitidos durante las comunicaciones entre los diferentes servicios Web, como: datos personales (redes sociales), credenciales de acceso (banca electrónica), etc., se deben garantizar los servicios de seguridad: disponibilidad, autenticación, confidencialidad e integridad. Existen protocolos que especifican la forma en que debe garantizarse la seguridad en los servicios Web como *Web Service Security* (WSS) y proyectos de software libre como *Open Web Application Security Project* (OWASP).

3.1.1 WSS

Originalmente desarrollado por IBM, Microsoft, y VeriSign, es un protocolo de comunicaciones que contiene especificaciones sobre cómo debe garantizarse la integridad y seguridad en mensajería de servicios Web. Es una extensión flexible y enriquecida para *Simple Object Access Protocol* (SOAP) para brindar seguridad a servicios Web. El protocolo WSS incluye detalles en el uso de *Security Assertion Markup Language* (SAML) y Kerberos, y formatos de certificado tales como X.509. Se enfoca principalmente en el uso de firma digital y cifrado de XML para proveer seguridad extremo a extremo.[22]

WSS describe tres mecanismos principales:

- Como se firman mensajes SOAP para asegurar integridad. Adicionalmente los mensajes firmados proveen no repudio.
- Como se cifran mensajes SOAP para asegurar confidencialidad
- Como adjuntar *tokens* de seguridad para determinar la identidad del remitente.

La especificación permite una variedad de formatos de firma, algoritmos de cifrado y varios dominios de confianza, además es abierto a varios modelos de seguridad de *token* como: certificados X.509, Kerberos, credenciales de usuario y contraseña y SAML.

La semántica y formatos de los *tokens* son definidos en los documentos asociados al modelo. WSS incorpora características de seguridad en la cabecera del mensaje SOAP, trabajando en la capa de aplicación del modelo OSI.

Estos mecanismos por si mismos no proveen una completa solución de seguridad para servicios Web, en lugar de esto, esta especificación es una construcción en bloque que permite ser usada en conjunto con otras extensiones de servicios Web y protocolos de aplicaciones especificas de alto nivel para dar cabida a una amplia variedad de modelos de seguridad y tecnologías de seguridad. En general, WSS por si mismo no provee ninguna garantía de seguridad. Cuando se implementa y se usa la sintaxis y el *framework*, le corresponde a quien lo implementa proporcionar los mecanismos necesarios para asegurar que el resultado no sea vulnerable. La gestión de claves, confianza de la secuencia de inicio y detalles técnicos (cifrado, formatos, algoritmos) están fuera del alcance de WSS.

3.1.2 OWASP

Es un proyecto abierto de seguridad en aplicaciones Web dedicado a determinar y combatir las causas que hacen que el software sea inseguro[23]. Brinda información sobre vulnerabilidades conocidas y metodologías para pruebas de penetración específicamente orientadas a la Web. La comunidad de OWASP incluye corporaciones, organizaciones educativas e individuos de todo el mundo. Es una comunidad que trabaja para crear artículos, metodologías, documentación, herramientas, y tecnologías de acceso libre orientados a la seguridad en aplicaciones Web.

Los proyectos de OWASP son divididos en dos principales categorías, proyectos de desarrollo y proyectos de documentación. Entre algunos de los proyectos de documentación se encuentran:

- OWASP *Application Security Verification Standard* (ASVS): Estándar para realizar verificaciones de seguridad a nivel de aplicación
- Guía OWASP: Documento que proporciona una guía detallada sobre la seguridad de aplicaciones Web
- OWASP Top 10: Documento de alto nivel que se centra sobre las vulnerabilidades más críticas en aplicaciones Web

Y algunos de los proyectos de desarrollo creados en esta comunidad son:

- Enigform: Conjunto de aplicaciones cliente y aplicaciones de servidor para

implementar características de OpenPGP a HTTP, como administración de sesiones seguras, firmas de peticiones y respuesta HTTP, y OpenPGP cifrado en HTTP.

- OWASP *Enterprise Security* API (ESAPI) *Project*: Es una colección de métodos de seguridad libres necesarios para construir aplicaciones Web seguras.
- AntiSamy: Aplicación para validar datos de entrada y codificación de salidas de impresión
- WebScarab: Aplicación que sirve de servidor proxy HTTP y HTTPS que puede ser usada para interceptar, examinar y modificar contenido de paquetes.
- Webgota: Una aplicación Web insegura creada por OWASP con el propósito de servir como una guía para prácticas de programación segura. Esta aplicación provee documentación y tutoriales sobre como explotar vulnerabilidades con la intensión de enseñar al usuario como escribir código seguro.

3.2 BIOMETRÍA

La biometría (del griego bios, que significa vida, y metron, que significa medida) es la ciencia de reconocer un individuo basado en sus características fisiológicas y/o de comportamiento. Los sistemas biométricos han sido empleados en varias aplicaciones comerciales, civiles y forenses. Entre los diferentes tipos de acceso biométrico, según la fisiología, se encuentran: huella dactilar, escaneo de iris, escaneo de retina, reconocimiento facial, escaneo de la geometría de la mano, etc., y según el comportamiento: firma personal, comportamiento en el uso del computador, etc.[24]. En la Tabla 1 se comparan diferentes características de estos métodos biométricos.

Tabla 1 Comparación de métodos biométricos

	Ojo - Iris	**Ojo - Retina**	**Huellas dactilares**	**Geometría de la mano**	**Escritura - Firma**	**Voz**
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta
Prevención de ataques	Muy Alta	Muy alta	Alta	Alta	Media	Media

Aceptación	Media	Media	Media	Alta	Muy alta	Alta
Estabilidad	Alta	Alta	Alta	Media	Media	Media
Identificación y autenticación	Ambas	Ambas	Ambas	Autenticación	Ambas	Autenticación
Estándares	-	-	ANSI/NIST, FBI	-	-	SVAPI
Interferencias	Gafas	Irritaciones	Suciedad, heridas, asperezas ...	Artritis, reumatismo ...	Firmas fáciles o cambiantes	Ruido, resfriados ...
Utilización	Instalaciones nucleares, servicios médicos, centros penitenciarios	Instalaciones nucleares, servicios médicos, centros penitenciarios	Policía, industrial	General	Industrial	Accesos remotos en bancos o bases de datos

Fuente: Red académica y de investigación española – RedIRIS[1]

3.2.1 Historia

Los sistemas de identificación biométricos automatizados sólo han estado disponibles durante las últimas décadas, y esto debido a los avances significativos en el campo del procesamiento por computador. Sin embargo, muchas de estas nuevas técnicas automatizadas fueron basadas en ideas que fueron originalmente concebidas cientos, incluso, miles de años atrás.

Uno de los más antiguos y más básicos ejemplos de una característica que es usada para el reconocimiento de personas son los rasgos faciales. Desde el inicio de la civilización, los humanos han usado los rostros para identificar individuos conocidos (familiares) o desconocidos (no familiares). Esta simple tarea fue incrementado cada vez más su dificultad a la medida que la población aumentaba y se introducían nuevos individuos en poblaciones que antes eran pequeñas. El concepto de reconocimiento humano a humano es también visto como biometría de comportamiento predominante como reconocimiento por voz y forma de caminar. Los individuos usan estas características, de algún modo inconsciente, para reconocer individuos conocidos en el día a día.

Entre otras características que han sido usadas, a través de la historia de la civilización, como métodos de reconocimiento más formales, se nombran algunos ejemplos [25]:

- En una caverna se encontró evidencia de los primeros usos de la biometría por parte de los humanos. Este hallazgo se estima que tiene, posiblemente, una edad de 31.000 años de antigüedad. Las paredes estaban adornadas con pinturas que se cree fueron creadas por hombres prehistóricos que vivieron allí. Alrededor de estas pinturas fueron encontradas numerosas huellas de manos, que en algún sentido, se piensa son la firma del creador de las pinturas.
- También existe evidencia de que las huellas digitales fueron usadas como una marca personal a principios de los años 500 D.C. En babilonia las transacciones de negocios eran grabadas en tablas de arcilla que incluían huellas dactilares.
- Joao de Barros, un explorador y escritor español, escribió que los primeros comerciantes chinos usaban huellas dactilares para firmar transacciones de negocios. Además, los padres chinos también usaban huellas dactilares y huellas de pies para diferenciar sus hijos de otros.
- En el principio de la historia egipcia, los comerciantes eran identificados por sus descripciones físicas, para diferenciar los comerciantes de confianza con buena reputación de aquellos que no lo eran.

A mediados de 1800, con el rápido crecimiento de las ciudades, debido a la revolución industrial y a cultivos más productivos, había una necesidad más formal de identificar a las personas. Comerciantes y autoridades se enfrentaban a una población creciente y cada vez más móvil, y no podían confiar solamente en sus propias experiencias y conocimientos. Influenciados por los escritos de Jeremy Betham y otros pensadores, las cortes de este periodo comenzaron a escribir conceptos de justicia que lograran perdurar hasta los días de hoy. Más notablemente, el sistema de justicia buscó tratar, por primera vez, que los delincuentes tuviesen mayor indulgencia y que aquellos delincuentes que volvieran a cometer crímenes fueran tratados más duramente.

Esto creó la necesidad de un sistema formal que registrara delitos, junto con las

características de rasgos de identidad del delincuente. El primero de dos enfoques fue el sistema Bertillon, que caracterizaba varias dimensiones del cuerpo, originalmente creado en Francia. Estas características estaban escritas en tarjetas que podrían ser ordenadas por altura, longitud de brazos o cualquier otro parámetro. Este campo fue llamado antropometría. El otro enfoque fue el uso formal de huellas dactilares por los departamentos de policía. Este proceso surgió en Sur América, Asia y Europa. A finales de 1800, un método fue desarrollado, como índice de huellas dactilares, que proveía la habilidad de obtener registros como el método Bertillon pero basado en una métrica más individualizada (patrones de huellas dactilares). El primer sistema robusto para clasificación de huellas dactilares fue desarrollado en la india por Azizul Haque para Edward Henry, Inspector General de la Policía de Bengala, India. Este sistema fue llamado el sistema Henry, y con algunas variaciones en él, es todavía usado actualmente para la clasificación de huellas dactilares.

Los verdaderos sistemas biométricos comenzaron a surgir a finales de la mitad del siglo veinte, coincidiendo con la aparición de los sistemas de computación. El naciente campo experimentó una explosión de desarrollo en 1990 y a principios del año 2000 comenzó a surgir en aplicaciones del día a día.[25]

3.3 BIOMETRÍA DE RECONOCIMIENTO DE CARAS

Detectar y reconocer rostros humanos, en imágenes y video, es un problema, cada vez con más auge, en el campo de la visión por computador. Entre algunas de sus soluciones entran el campo de la minería de datos y el reconocimiento de patrones. Existe una gran diversidad de aplicaciones prácticas, algunas de ellas son [26]:

- **Vigilancia:** Buscar un individuo perseguido por la justicia. Localizarle en ámbitos públicos, aduanas, aeropuertos, terminales. Vigilancia doméstica, verificar que las personas que entran son conocidas o desconocidas, detección de intrusos.
- **Videoconferencia:** Localizar la imagen de un individuo en la secuencia de video a través de la *Webcam* para poder hacer seguimiento, además para saber si realmente se encuentra presente o no.

- **Detección de expresiones faciales:** Considerado un subcampo del reconocimiento facial en general. Utilizada en interfaces inteligentes y aplicaciones médicas.
- **Control de acceso:** En tarjetas de identificación, detección de fraudes (usurpación de identidad), interfaces hombre-máquina, etc.

El punto de partida es poseer un conjunto previo de imágenes etiquetadas y almacenadas en una base de datos, a partir de las cuales se pueda entrenar el sistema, por medio de algún tipo de mecanismo de aprendizaje, que permita resolver el problema concreto. El problema de reconocimiento de caras considera dos escenarios, dependiendo del objetivo que se tenga: identificación de caras ¿quién soy? (test uno a muchos) y verificación o autentificación de caras ¿soy quien digo ser? (test de uno a uno)[26] .

3.3.1 Historia

El primer método formal de clasificación de caras fue propuesto en 1888 por Francis Galton. Se recogían perfiles faciales como curvas, encontrando su norma, después se clasificaban perfiles por sus desviaciones con respecto a la norma. El resultado era un vector que podía ser comparado con otros vectores de la base de datos.

En 1960, se introdujeron sistemas semiautomáticos que hacían marcas en las fotografías para localizar los rasgos principales: ojos, orejas, nariz y boca. Las distancias y radios se calculaban a través de las marcas para construir un sistema de referencia y poder comparar los datos. A principios de los 70, Goldstein, Harmon y Lesk crearon un sistema con 21 marcadores que incluían color del pelo y grosor de labios. Sus pruebas eran también difíciles de automatizar porque muchas de estas medidas se tomaban a mano.

Pocos años después, Fisher y Elschlagerb introdujeron un sistema más automático que utilizaba plantillas para medir los rasgos de diferentes partes de la cara, con esas medidas se construía un mapa global. Tras una continuada investigación, resultó que estas medidas no contenían suficientes datos únicos como para representar una cara de adulto.

Otra aproximación intenta clasificar la cara humana, usando una combinación de la gama de gestos como un juego de marcadores a identificar. En general, se pone en práctica usando reconocimiento de modelos en 2D y principios de redes neuronales. La mayoría de

veces esta técnica requiere un enorme número de caras a entrenar, para alcanzar una exactitud respetable.

El primer método completamente automatizado, comparaba las caras en un modelo genérico de rasgos esperados y creaba una serie de patrones para una imagen. Esta estrategia, era principalmente estadística, se basaba en histogramas y en el valor de escala de gris. Por otro lado, a partir de 1988 los trabajos –primero de Kirby y Sirovich (Brown University) y después de Turk y Pentland (MIT), basados en el método *Eigenface*, abrieron un camino al que ha ido contribuyendo mucha gente.

Desde 1990, al interés por el reconocimiento de caras automático se han sumado un creciente número de grupos de investigación. Apoyándose también en las mejoras técnicas que han supuesto el desarrollo de las redes neuronales, análisis de *wavelet*, infografía y visión por computador. [26]

3.3.2 Detección de Caras

La detección de caras es considerada un paso previo al reconocimiento de caras, ya que son aquellas técnicas que se emplean para localizar un rostro válido dentro de una imagen y así extraer dicho rostro. Además, la mayoría de las técnicas de reconocimiento facial tienen sentido si esta fase se realiza adecuadamente.

El enfoque del problema para detectar caras es lograr determinar si existe o no una cara válida en una imagen arbitraria, y si se encuentra alguna cara válida, retornar su ubicación en la imagen y la extensión de cada cara. Entre los retos asociados con la detección de caras pueden ser atribuidos los siguientes factores[27]:

- **Pose**: Las imágenes de una cara varían debido a la pose de la cara con respecto a la cámara (frontal, 45 grados, arriba, abajo, etc.), y algunas características faciales como los ojos o la nariz pueden obstruir la imagen parcial o totalmente.
- **Presencia o ausencia de componentes estructurales**: Características faciales como barba, bigote, y lentes pueden o no estar presentes y hay una gran variabilidad entre estos componentes incluyendo formas, color y tamaño.

- **Expresiones fáciles**: La apariencia de las caras están directamente afectadas por la expresión facial de la persona.
- **Obstrucción**: Las caras pueden estar parcialmente obstruidas por otros objetos. En una imagen con un grupo de personas, algunas caras podrían estar parcialmente obstruidas por otras.
- **Orientación de la imagen**: Las imágenes de las caras varían directamente por las diferentes rotaciones del eje óptico de la cámara.
- **Condiciones de imágenes**: Cuando las imágenes son capturadas, factores como la iluminación (espectro, fuente de distribución e intensidad) y las características de la cámara (sensor, respuesta, lentes) afectan la apariencia de una cara.

Y. Ming-Hsuan et al. [27]clasifican en 4 categorías las diferentes técnicas usadas para el problema de detección de caras en una imagen:

1. **Métodos basados en conocimiento**

   Estos métodos son basados en reglas que codifican el conocimiento humano que constituye una cara típica. Usualmente, las reglas capturan las relaciones entre características faciales, como ejemplo que existan en la imagen de una cara dos ojos que sean simétricos el uno con el otro, una nariz y una boca. Las relaciones entre estas características pueden ser representadas por sus distancias y posiciones relativas. Las características faciales en una imagen de entrada son extraídas primero y las caras son identificadas basadas en las reglas codificadas. Adicionalmente, se finaliza con un proceso de verificación para reducir las detecciones erróneas. Un ejemplo de esta técnica es el método basado en reglas de multi-resolución realizado por G. Yang y T. S. Huang [28].

2. **Métodos basados en enfoque de características invariables**

   Estos algoritmos pretenden encontrar características estructurales que existan incluso cuando la pose, punto de vista, o condiciones de luz varíen, y usan estas características para localizar caras. En contraste con los métodos basados en

conocimiento, este enfoque trata de encontrar aquellas características invariables en las caras para su detección. La idea fundamental nace en la observación que los humanos pueden detectar caras y objetos casi sin esfuerzo en diferentes poses y condiciones de luz y, así, deberían existir propiedades o características que sean invariantes sobre estas variabilidades. Características faciales como las cejas, ojos, nariz, boca, y cabello son extraídas usando técnicas de detección de bordes. Basado en las características extraídas, un modelo estadístico es construido para describir sus relaciones y para verificar la existencia de una cara. El problema con este tipo de algoritmos basados en características es que las propiedades de la imagen pueden ser modificadas severamente debido a la iluminación, ruido y obstáculos.

3. **Métodos de comparación de plantillas**

   Estos métodos usan una gran cantidad de patrones estándar de una cara (con pose frontal), previamente guardados, que describen la cara como un todo. Los patrones estándar de una cara son manualmente predefinidos o parametrizados por una función. Dado una imagen de entrada, los valores de correlación con los patrones estándar son procesados por el contorno de la cara, ojos, nariz y boca, independientemente. La existencia de una cara es determinada por los valores de correlación. Este enfoque tiene la ventaja de ser simple de implementar. Sin embargo, ha sido probado que es inadecuado para la detección de caras, ya que no puede procesar efectivamente variaciones en escala, pose, y formas. Multi-resolución, multi-escala, subplantillas y plantillas deformables han sido propuestas, posteriormente, para lograr un mejor rendimiento con escala e invariancias de forma.

4. **Métodos basados en apariencias**

   En contraste con el método de comparación de plantillas, donde los modelos (o plantillas) son predefinidos por expertos, las plantillas en los métodos basados en apariencias son aprendidas de ejemplos, desde un conjunto de imágenes de entrenamiento, las cuales deberían capturar la variabilidad representativa de la

apariencia de una cara. En general, los métodos basados en apariencias se fían en técnicas de análisis estadístico y máquinas de aprendizaje, para encontrar características relevantes en una imagen que posea o no cara. Las características aprendidas están en la forma de modelos de distribución o funciones discriminantes, que son, por consiguiente, usadas en la detección de caras. Mientras tanto, la reducción de la dimensionalidad es, usualmente, llevada a cabo por la eficiencia en cómputo y eficacia en la detección.

3.3.3 Métodos de Reconocimiento Facial

El reconocimiento de caras es una de las características biométricas que más ha llamado la atención de los investigadores, debido a que ofrece un método no invasivo y pasivo para la captura de datos biométricos. Aunque es claro que las personas son buenas reconociendo rostros, no es del todo obvio como los rostros son codificados o decodificados por el cerebro humano. El reconocimiento facial de humanos ha sido estudiado por más de veinticinco años. Desafortunadamente desarrollar un modelo computacional para reconocimiento de rostros es algo difícil, porque las rostros son complejos, multidimensionales y dependen de otros estímulos visuales como el uso de lentes, corte de cabello, etc. Por lo tanto, el reconocimiento facial es una tarea de alto nivel para la visión de computadora, en la cual se ven involucradas muchas técnicas para visión.

El primer paso para identificación de rostros humanos es extraer las características más relevantes de las imágenes faciales. Investigaciones en el campo buscan, principalmente, generar suficientes similitudes familiares de rostros humanos para que otro humano pueda correctamente identificar el rostro. La pregunta que naturalmente se da es: ¿qué tan bien pueden ser cuantificadas las características faciales? Si existe tal cuantificación es posible que un computador pueda ser capaz de reconocer una cara, dado un conjunto de características. Numerosas investigaciones realizadas han demostrado que ciertas características faciales pueden ser usadas por los seres humanos para identificar rostros.

Existen diferentes métodos desarrollados para el reconocimiento facial en sistemas de computación. Entre los métodos más básicos se encuentran *Eigenfaces* o *Principal*

Component Analysis (PCA) [29][30], *Fisherfaces* o *Linear Discriminant Analysis* (LDA) [31][32] y otros métodos más avanzados como *Kernel Methods* [33], *3D face recognition methods* [34], *Gabor Wavelets method* [35], *Hidden Markov Models* [36] y *Active Appearance Models* [37].

3.3.3.1 *Eigenfaces*

Eigenfaces o *Principal Component Analysis* (PCA) es un método para el reconocimiento de rostros desarrollado por Turk Matthew y Pentland Alex, basado en los conceptos de la teoría de la información, en otras palabras, en los métodos de análisis de componentes principales. En este enfoque, tratan al problema de reconocimiento de rostros como un problema de reconocimiento intrínsecamente de dos dimensiones (2D), en lugar de requerir de la geometría en tres dimensiones, tomando como ventaja el hecho de que los rostros se encuentran normalmente en posición vertical y por lo tanto pueden ser descritos como un pequeño conjunto de características de 2D.

Basados en la expansión de patrones de reconocimiento, de Karhunen-Loeve, M. Kirby y L. Sirovich, han creado un sistema que funciona proyectando imágenes de rostros en un espacio de características, que abarca las variaciones significativas entre las imágenes de rostros conocidos. Estas características significativas son conocidas como "*eigenfaces*", debido a que son los *eigen*-vectores (componentes principales) del conjunto de rostros; así que no necesariamente corresponden a características como ojos, orejas o nariz. La operación de proyección caracteriza un rostro individual por la suma de la ponderación de las características de la *eigenface*, y así para reconocer un rostro en particular sólo es necesario comparar su ponderación con las de los individuos conocidos.[29].

El problema con la presentación de imágenes se debe a su alta dimensionalidad. Imágenes de escala de grises de dos dimensiones $p \times q$ se extienden a un vector de espacio $m = pq$-dimensional, así que una imagen de 100×100 pixeles resulta en un espacio dimensional de imagen de 10.000 pixeles. Este es un espacio dimensional muy grande para ser procesado, por lo que cabe preguntar si todas las dimensiones son realmente útiles. Sólo se puede tomar una decisión si hay alguna varianza en los datos, así que realmente se están

buscando los componentes que contienen la mayor información posible.

El análisis de componentes principales o en ingles *Principal Component Analysis* (PCA) fue independientemente propuesto por Karl Pearson y Harold Hotelling para convertir un conjunto de variables, posiblemente correlacionadas, en un pequeño conjunto de variables no correlacionadas. La idea es que un conjunto de datos de grandes dimensiones es, a menudo, descrito correlacionando variables y, por lo tanto, sólo pocas dimensiones significativas cuentan con la mayor cantidad de información. El método PCA encuentra las direcciones con mayor varianza en los datos, llamados componentes principales.[30]

Descripción del Algoritmo

Sea el conjunto $X = \{x_1, x_2, \dots\dots, X_n\}$ un vector aleatorio con observaciones $x_i \in \mathbb{R}$,

1. Calcular la media μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

2. Calcular la matriz de covarianza S

$$S = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T \tag{2}$$

   Donde T indica que se utiliza la matriz transpuesta.

3. Calcular los valores propios λ_i y vectores propios v_i de S

$$Sv_i = \lambda_i v_i, i = 1,2, \dots\dots, n \tag{3}$$

4. Ordenar los vectores propios descendentemente por su valor propio. Los k componentes principales son los vectores propios v_i correspondientes a los k valores propios λ_i más grandes.

Los k componentes principales del vector x observado son dados por:

$$y = W^T(x - \mu) \tag{4}$$

Donde $W = (v_1, v_2, \dots\dots, v_k)$. La reconstrucción de la base de PCA está dada por:

$$x = W_y + \mu \tag{5}$$

El método de *Eigenfaces* realiza el reconocimiento de caras siguiendo estos pasos:

1. Proyecta todas las muestras de entrenamiento en el sub-espacio PCA (usando la ecuación 4)
2. Proyecta la imagen de entrada en el sub-espacio PCA (usando las ecuaciones 1,2 y 3)
3. Se encuentra el vecino más cercano entre las imágenes de entrenamiento proyectadas y la imagen de entrada proyectada.

3.3.3.2 *Fisherfaces*

Fisherfaces o *Linear Discriminant Analysis* (LDA) es una técnica para el reconocimiento de patrones, basada en el trabajo desarrollado por Robert Fisher en 1936 para clasificación taxonómica. Dependiendo de las características que se utilicen, puede ser aplicada de diferentes formas en visión por computadora e incluso para el reconocimiento de rostros. Cuando LDA es usado para encontrar la representación del sub-espacio de un conjunto de imágenes de rostros, los vectores base resultantes, que definen ese espacio, se conocen como *Fisherfaces.* Cheng et al. [31] presentaron un método que hacía uso del discriminador de Fisher para reconocimiento de rostros, más tarde Belhumeur et al. [32] presentaron un algoritmo basado en *Fisherfaces* para el reconocimiento de rostros, en el cual mostraban que tenía un mejor rendimiento comparado con *Eigenfaces*, al reducir la tasa de error ocasionado por las variaciones de iluminación y expresiones faciales.

El método PCA, descrito anteriormente, encuentra una combinación lineal de características, que maximiza la varianza total en los datos. Aunque esta es, claramente, una poderosa manera de representar los datos, no considera ninguna clase y así, mucha de la información discriminatoria puede perderse cuando existen componentes lejanos. Un ejemplo de una situación donde la varianza puede verse afectada por una fuente externa, puede ser la iluminación. Los componentes identificados por PCA no necesariamente pueden contener información discriminatoria, por lo que las muestras proyectadas pueden encontrarse distorsionadas y una clasificación se vuelve imposible. Mientras PCA busca los vectores que mejor describen los datos, LDA busca los vectores que proporcionan mejor discriminación entre clases después de la proyección.

Con el fin de encontrar la combinación de características que mejor separe las clases, el LDA maximiza la relación entre las clases dentro de las clases de dispersión. La idea es simple: muestras de la misma clase deberían agruparse juntas, mientras que muestras de diferentes clases deberían estar tan alejadas como fuese posible una de otra. Esto fue también reconocido por Belhumeur, Hespanha y Kriegman, así que aplicaron un análisis discriminante para reconocimiento de caras.[30]

Descripción del Algoritmo

Sea X un vector aleatorio con muestras de patrones etiquetados en c clases y cada clase cuenta con X_i patrones:

$$X = \{X_1, X_2, \dots., X_c\} \tag{6}$$

$$X_i = \{x_1, x_2, \dots., x_n\} \tag{7}$$

Las matrices de dispersión S_B y S_W son calculadas como:

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{8}$$

$$S_w = \sum_{i=1}^{c} \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T \tag{9}$$

Siendo μ_i la media de cada clase, N_i la cantidad de patrones de la clase i, μ la media de todos los datos:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{10}$$

Y μ_i es la media de la clase $i \in \{1, \dots \dots, c\}$:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j \tag{11}$$

El algoritmo clásico de Fisher, busca luego una proyección W, que maximice el criterio de separabilidad de clases:

$$W_{opt} = argmax_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{12}$$

Una solución, para este problema de optimización, se da resolviendo el problema general de valores propios:

$$\begin{aligned} S_B v_i &= \lambda_i S_w v_i \\ S_W^{-1} S_B v_i &= \lambda_i v_i \end{aligned} \tag{13}$$

Solo queda un problema que resolver: La clasificación de S_w es a lo sumo $(N - c)$, con N muestras y c clases. En problemas de reconocimiento de patrones el número de muestras N es casi siempre más pequeño que la dimensión de los datos de entrada (el número de pixeles), y así la matriz de dispersión S_w siempre es singular, es decir, no tiene inversa. Esto es resuelto realizando un PCA en los datos y proyectando las muestras en el espacio dimensional $(N - c)$. Se realiza, entonces, un LDA en la data reducida, debido a que S_w deja de ser singular [30].

El problema de optimización puede ser reescrito como:

$$W_{pca} = argmax_W |W^T S_T W| \tag{14}$$

$$W_{fld} = argmax_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \tag{15}$$

La matriz de transformación W, que proyecta la muestra en el espacio dimensional $(c - 1)$ está dada por:

$$W = W_{fld}^T W_{pca}^T \tag{16}$$

El método de *Fisherfaces* realiza el reconocimiento de caras siguiendo estos pasos [26]:

1. Construye la matriz de dispersión con los vectores propios generalizados de S_B y S_W correspondientes a los valores propios no nulos. (usando las ecuaciones 8 y 9,

respectivamente)

2. Para cada clase se construye un vector de características proyectando la matriz anterior. Se puede elegir uno por cada clase o promediarlos.
3. Previamente, puede ser necesaria una reducción de dimensionalidad para poder tratar con S_B y S_W, dicha reducción debería hacerse a los vectores de características y a los de prueba para la clasificación.
4. La clasificación se resuelve por la distancia a los vectores de características.

4. DESARROLLO

4.1 ANÁLISIS DE REQUISITOS Y ELECCIÓN DEL SOFTWARE

En esta sección se realiza el análisis de la información referente al problema, el establecimiento de los requisitos que debe cumplir la aplicación y la elección de las diferentes herramientas que se usan para el desarrollo de la misma.

4.1.1 Comparación de Soluciones de Autenticación Biométrica para Aplicaciones Web

Se compararon cuatro soluciones de autenticación biométrica para aplicaciones Web (ver Tabla 2). Al analizar esta tabla se determinó que los requisitos más relevantes para el diseño de un esquema de autenticación por reconocimiento facial vía Web son:

- Comunicación segura entre el cliente y el servidor Web
- Algoritmo robusto para el reconocimiento de rostros conocidos
- Algoritmo para detección de rostros válidos

Otros elementos que se tomaron en consideración para el diseño de la aplicación fueron:

- Uso de software libre para el desarrollo de la misma, a fin de que pueda ser distribuida y utilizada ampliamente en Internet.
- Aplicación ejecutable en cualquier plataforma con una mínima configuración.
- Datos y configuraciones separados de la aplicación.
- Implementación y uso de buenas prácticas de programación, así como patrones de diseño.
- Diseño de la interfaz de gestión de usuarios.

Tabla 2 Análisis Comparativo de Soluciones de Autenticación Biométrica para Aplicaciones Web

Solución	**Referencia**	**Código abierto**	**Pago**	**Lenguaje de programación**	**Descripción**	**Biometría multimodal**	**Características biométricas medidas**	**Algoritmo de reconocimiento Facial**	**Detección de caras**	**Comunicación cifrada entre cliente/servidor**
BioWebAuth	[7][10][12] [13]	Si (Código fuente disponible)	No	Java	Aplicación de Software Libre que hace uso de OpenCV[11] , BioApi[8] y CAS para brindar autenticación biométrica multimodal a servicios Web basados en Java	Si	Reconocimiento facial, Reconocimiento de voz, Reconocimiento de huella digital y Firma digital	Gabor Wavelets [35]	Si	Si
Reconocimiento facial en línea, una arquitectura Open Source para validación de identidad de estudiantes para el LMS MOODLE	[9]	Si (Código fuente no disponible)	No	Flex, Php	Solución de autenticación biométrica para plataformas LMS MOODLE, no se encuentra disponible para descargas.	No	Reconocimiento facial	Principal Component Analysis (PCA) [29][30]	Si	No
BioID	[14]	No	Si	-	Servicio propietario para la autenticación biométrica facial en servicios Web y plataformas móviles que usen OpenID	No	Reconocimiento facial	-	Si	Si

VeriLook	[47]	No	Si	Múltiples Lenguajes	Provee una SDK con licencia de pago, permite usar biometría multimodal para aplicaciones, servicios Web y plataformas móviles	No	Reconocimiento facial	-	Si	Si

4.1.2 Herramientas de Software Libre a Utilizar

Para el desarrollo del proyecto se decidió seleccionar un conjunto de herramientas que permitieran facilitar la tarea del desarrollo del sistema de reconocimiento facial para ambiente Web. Fueron seleccionadas aquellas con código fuente disponibles y con licencia de software libre para su uso sin restricciones, entre las herramientas seleccionadas se presentan a continuación:

- **OpenSSL:** Es un paquete de herramientas administrativas y bibliotecas de código abierto, de funciones criptográficas, que permite implementar los protocolo SSL[2] y TLS[3] y puede ser usado en diversos sistemas operativos [42].
- **OpenCV:** Es una librería multiplataforma de código abierto para visión artificial con licencia BSD[4], lo que permite su uso particular o comercial. Contiene más de 2500 algoritmos optimizados para el procesamiento de imágenes, aprendizaje de máquina, entre otros. [38]
  - **Libfacerec*:*** Es una librería, creada por Philipp Wagner, para el reconocimiento de caras, que implementa métodos conocidos como Eigenfaces, Fisherfaces y LBPH[5]. Forma parte de la API de C++ de OpenCV. [30]
- **Marilena:** Es una librería AS3[6] creada por Ohtsuka Masakazu, que implementa una versión del algoritmo basado en *Haar-like feature*[7] para la detección de objetos, usado para la detección de caras de la librería OpenCV en AS3 [40]. Además, se hace uso de una versión modificada y optimizada por Mario Klingemann [41], que permite detectar caras sin la necesidad de instalar algún elemento adicional, facilitando la tarea de detección de caras en aplicaciones embebidas a navegadores Web.

---

[2] *Secure Sockets Layer* (SSL) protocolo criptográfico para comunicación segura en red
[3] *Transport Layer Security* (TLS) protocolo criptográfico sucesor de SSL, incorpora una serie de mejoras de seguridad para comunicación segura en red
[4] *Berkeley Software Distribution* (BSD) es una licencia de software libre que mantiene la protección de los derechos de autor únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, permitiéndose el uso en software comercial.
[5] *Local Binary Patterns Histograms* (LBPH) método para reconocimiento facial que hace uso de descriptores de textura
[6] Adobe ActionScript v3, lenguaje de programación para la plataforma Adobe Flash
[7] *Haar-like feature* son características de imágenes digitales usadas en reconocimientos de objetos.

- **JQuery:** Es una librería de *javascript* que permite simplificar la manera de interactuar con documentos html, manipular hojas de estilos CSS, manejar eventos, desarrollar efectos y animaciones, seleccionar elementos Dom y agregar la tecnología AJAX[8]. [39]

4.1.3 Plataforma de Sistema Operativo.

Existen diferentes sistemas operativos, pero se hace necesario elegir uno para el desarrollo de la aplicación. Para fines prácticos de la investigación se optó por usar la distribución de GNU/Linux Debian 6.0 por las siguientes razones:

- Es una distribución de Linux ampliamente reconocida, que posee una activa y extensa comunidad de usuarios.
- Es una plataforma compatible con las diferentes librerías y herramientas de software libre seleccionadas para el proyecto.
- Sus requisitos de funcionamiento son mínimos (Procesador de 1 GHz, 256 MB de RAM y 5 GB de Disco Duro), lo que lo hace perfecto para funcionar en una máquina virtual.

A pesar que se seleccionó este sistema operativo, las librerías y herramientas de software libre son compatibles con una gran diversidad de sistemas operativos disponibles, lo que las hace portables a otras plataformas.

4.1.4 Lenguajes de Programación

Para el desarrollo y ejecución de esta investigación se decidió elegir los lenguajes de programación apropiados que fueran compatibles con el conjunto de librerías y herramientas de software libre así como con el sistema operativo seleccionado.

- Para el diseño del servicio Web se seleccionó PHP ya que es un lenguaje ampliamente usado para desarrollo Web y ofrece una gran versatilidad de interacción con gestores de base datos.
- Para el diseño y desarrollo de los módulos de detección de caras y reconocimiento facial se selecciono C++ por su compatibilidad con la librería OpenCV y velocidad de procesamiento.

[8] AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML)

- Para mejorar el rendimiento y el diseño de la interfaz gráfica del diseño Web se optó por el uso de Javascript junto con la Librería JQuery para procesar las peticiones AJAX entre cliente/servidor.
- Finalmente para el desarrollo del componente de captura de datos biométricos a través de Webcam se selecciono Flex3, ya que permite crear componentes Flash que pueden ser incrustados dentro de la página Web, gracias a esto no depende de la instalación de archivos adicionales para su funcionamiento.

Cabe destacar que todos los lenguajes de programación seleccionados son *Open Source* lo que permite su libre uso e implementación sin restricciones de ningún tipo.

4.1.5 Patrón de diseño

Se puede afirmar que el uso de buenas prácticas en el desarrollo de aplicaciones es una necesidad fundamental para desarrollar software de calidad, es por ello que se optó por la implementación de los patrones de diseño Modelo Vista Controlador (MVC) [44] y Singleton [45] por las siguientes razones:

El patrón de diseño MVC permite separar la interfaz gráfica de su correspondiente programación y acceso a datos, lo que supone una mejora en el desarrollo y mantenimiento de la aplicación. El patrón de diseño Singleton, por su parte, garantiza la existencia de una única instancia para cada clase y la creación de un mecanismo de acceso global a dicha instancia, lo que supone una mejora en el rendimiento de la memoria.

4.1.6 Base de datos

En esta investigación uno de los principales puntos a tener en cuenta es el tipo de base de datos a utilizar, debido a que es necesario guardar la información de autenticación para reconocimiento de los usuarios. Se eligió el gestor de base de datos MYSQL como sistema de almacenamiento de datos persistentes debido a que:

- Su código es de dominio público y puede ser incorporado en el desarrollo de cualquier aplicación sin ningún tipo de restricción.
- Es autónomo, no necesita de ningún software adicional para funcionar.
- Es compatible con los lenguajes de programación elegidos.

4.2 DISEÑO DEL SISTEMA

4.2.1 Funcionalidad del Sistema

El sistema de autenticación por reconocimiento facial es un sistema que utiliza tecnología biométrica para garantizar el ingreso legítimo de los usuarios, previamente registrados en la base de datos, y disminuir la posibilidad de una suplantación de identidad.

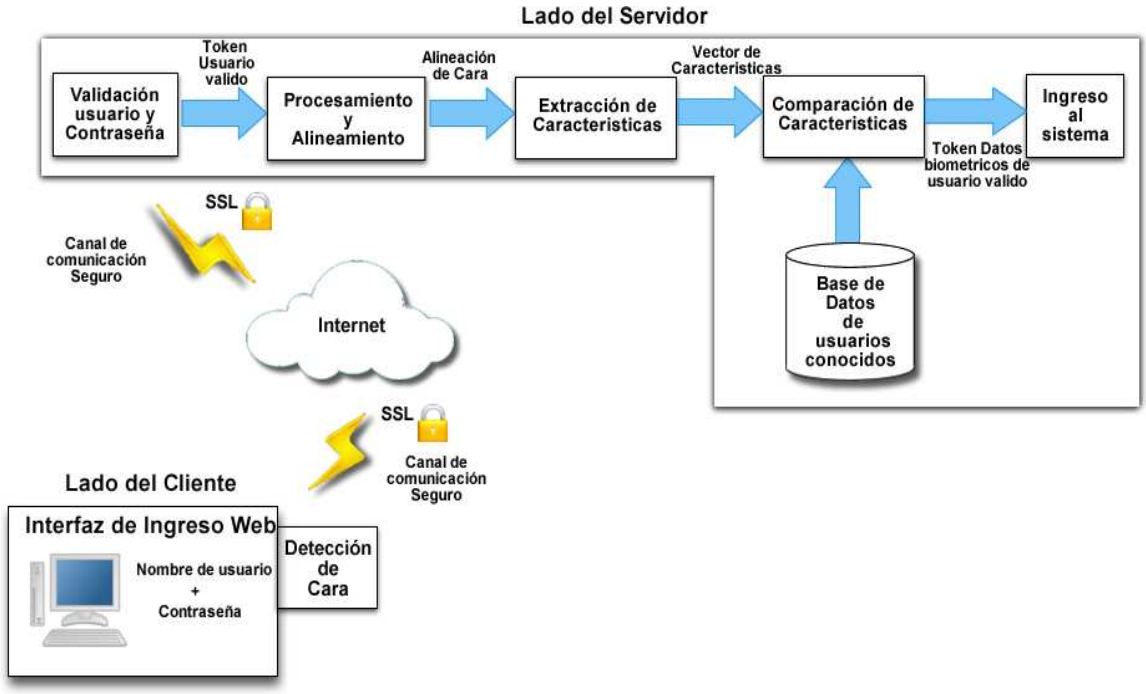
Además de autenticar a los usuarios, el sistema realiza las siguientes funciones: crear y modificar cuentas de usuario, agregar o modificar fotos de los usuarios para el entrenamiento de la base de datos, verificar que las muestras contengan caras válidas, procesar las imágenes obtenidas aplicando escala de grises, detectar rostros y recortar las caras detectadas para tener sólo imágenes con caras válidas, y permitir el uso de dos tipos de métodos para el reconocimiento facial: *Eigenfaces* y *Fisherfaces*.

Adicionalmente, el sistema usa un archivo de configuración, con el cual se puede personalizar el comportamiento del mismo. También, tiene un diseño orientado a servicios Web, implementado a través de una API para el procesamiento de funciones y hace uso de un canal de comunicación cifrado por medio de SSL con certificados digitales.

4.2.2 Diseño del Esquema de Autenticación

En la Fígura 1 se detalla el esquema de autenticación fuerte propuesto. El usuario accede al sistema por medio de un navegador Web e ingresa sus credenciales (nombre de usuario y contraseña). Una vez validados estos datos, el sistema solicita al usuario, que desea ingresar, sus datos biométricos. Un objeto Flash embebido en el navegador inicia la captura de datos biométricos a través de la cámara Web (para validar que se encuentre una cara válida, en el momento de la adquisición de imágenes, se hace uso del detector de rostros implementado en Marilena [40][41]). Terminado este proceso, los datos se envían a través de un canal seguro, para evitar el robo de los mismos, y en el lado del servidor se procesan las muestras de imágenes capturadas por la aplicación cliente, se extraen las características y se comparan con las contenidas en la base de datos de usuarios conocidos. Si el valor de coincidencia se encuentra en el rango aceptado, se permite el ingreso al usuario.

Fígura 1 Esquema de Autenticación Cliente - Servidor



Visto de una manera más simple los pasos para el ingreso de un usuario al sistema son:

1. El usuario ingresa al portal Web a través de SSL (HTTPS)
2. El servidor valida que esté habilitado Javascript y Flash, en caso contrario lanza un mensaje de error
3. El servidor solicita las credenciales de acceso al usuario (nombre de usuario y contraseña)
4. El servidor valida las credenciales. En caso de error devuelve la página de inicio. En caso de éxito, devuelve un token de validación y se carga el componente flash (ver paso 5).
5. Se carga el componente flash para la captura de imágenes por *Webcam*, a fin de identificar al usuario
    i. Se inicia la captura de imágenes con el botón de autenticar
    ii. Se captura una serie de imágenes (según se ha definido en el archivo de configuración) y se envían al servidor

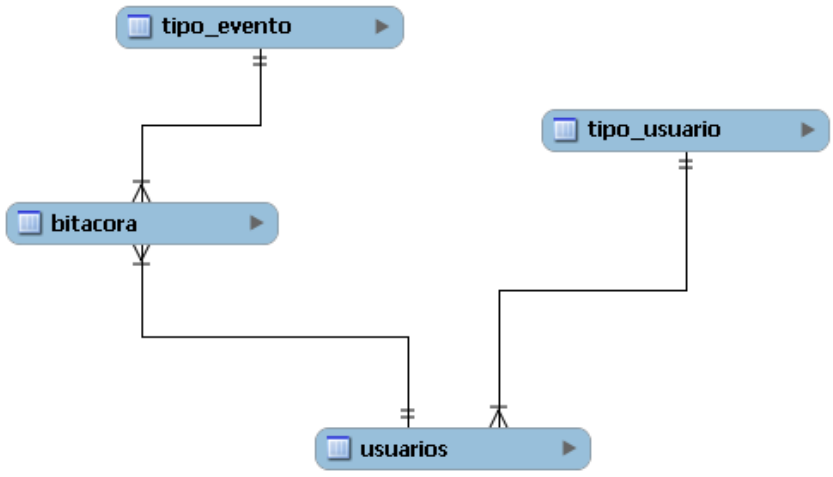
iii. El servidor valida las imágenes. En caso de éxito, devuelve un *token* de validación y se continúa con el paso iv. En caso de error devuelve un mensaje de error (mensaje enviado en formato JSON[9] a través de SSL) (el componente Flash re-direcciona a la página de inicio, eliminando la sesión)

iv. Se verifica el tipo de cuenta del usuario y se re-direcciona a su página correspondiente.

4.2.3 Diseño de la Base de Datos

En la Fígura 2, se muestra el modelo entidad – relación de la base de datos a usarse en el prototipo:

Fígura 2 Modelo Entidad - Relación de la Base de datos



- La entidad **usuarios** representa a los miembros que pueden acceder al sistema, contiene los atributos: nombre del usuario, apellido, email, pseudónimo del usuario para ingresar al sistema, *password*, ruta del directorio de fotos del usuario.
- La entidad **tipo_usuario** se refiere al tipo de cuenta a la cual está asociado un usuario. Permite clasificar a un usuario como administrador o usuario del sistema.

[9] *JavaScript Object Notation* (JSON) formato ligero de intercambio de datos.

Tiene los atributos nombre del tipo de cuenta y descripción.

- La entidad **bitácora** se refiere a los eventos realizados por el usuario. Contiene los atributos: dirección IP, hora del evento, y un campo opcional de información adicional.
- La entidad **tipo_evento** representa al conjunto de tipos de eventos definidos dentro del ámbito del sistema. Contiene los atributos: nombre del evento y descripción.

4.3 IMPLEMENTACIÓN DEL SISTEMA

4.3.1 Implementación del Módulo de Autenticación Biométrico

Para la implementación del mecanismo de reconocimiento facial se crearon un conjunto de herramientas para el procesamiento de imágenes (Tabla 3), debido a que, los algoritmos de reconocimiento facial sólo funcionan correctamente con imágenes debidamente procesadas con rostros válidos.

Se hizo uso del lenguaje de programación C++ junto con la librería OpenCV para las funciones de procesamiento de imágenes, entrenamiento de la base de datos de rostros y reconocimiento de rostros conocidos. Para la extracción de rostros válidos de imágenes de entrada se hizo uso del clasificador Haarcascade_frontalface_alt.xml de la misma librería OpenCV.

A fin de determinar el tamaño ideal para las imágenes extraídas se realizaron diferentes pruebas y se obtuvieron los mejores resultados con muestras de imágenes de rostros de 80x80 pixeles, por lo que imágenes de mayor tamaño necesitaban mayor nivel de procesamiento e imágenes de menor tamaño perdían información y eran más propensas a errores.

Tabla 3 Herramientas implementadas para el modulo de autenticación

Herramienta	Uso	Descripción
detectar_cara	Uso: procesar_caras <archivo haarcascade faces> <opción> <imagen> [dirección nueva imagen] -d -> sólo detecta cara en la foto, imprime mensaje -s -> salva la región de interés de la cara detectada	Programa para detectar y extraer caras alineadas de imágenes. Extrae la cara y procesa la imagen a escala de grises para ser usada en el reconocimiento

entrenar_bd_rostros	Uso: entrenar_bd_rostros <método> <archivo de entrenamiento> <nombre de archivo de entrenamiento>	Programa para el entrenamiento de caras. Genera un archivo de base de datos usado en el reconocimiento
procesar_imagen	Uso: procesar_imagen <imagen a procesar> <nombre de archivo de salida>	Programa para procesar una imagen de escala de colores a escala de grises
reconocer	Uso: ./reconocer <método> <datafaces.yml> <testfile>	Programa para reconocer una imagen procesada de cara, usando una base de datos de caras conocidas

Los procesos de entrenamiento de base de datos de rostros conocidos y autenticación de rostros, usados en este módulo, son descritos a continuación.

Entrenamiento de la Base de Datos de Rostros Conocidos

1. Usando la herramienta **detectar_cara** se extrae un rostro válido de una imagen de entrada usando el clasificador haarcascade_frontalface_alt.xml de la librería OpenCV. Luego, se alinea el rostro resultante, se convierte a escala de grises y de ser necesario se agranda o reduce de tamaño. Así, se extrae una imagen de rostro válido de dimensiones 80x80 pixeles.
2. La herramienta **entrenar_bd_rostros** recibe un archivo de entrada llamado archivos_entrenamiento.csv, el cual contiene información sobre la ubicación de los rostros conocidos para entrenar la base de datos, se especifica el método de reconocimiento facial a usar (*Eigenfaces* o *Fisherfaces*), y se genera el archivo de la base de datos entrenada llamada bd_rostros.yml.

Reconocimiento de Rostros Conocidos

1. Una vez validados las credenciales de ingreso de un usuario (nombre de usuario y contraseña) se realiza una captura de imagen vía *Webcam* y se procesa con la herramienta **detectar_cara** para extraer un rostro válido.
2. Luego de procesar la imagen de entrada, se invoca a la herramienta **reconocer** que

revisa en la base de datos de rostros conocidos, y genera la información sobre el resultado de algoritmo usado (Como fue descrito en la sección 5.2.2). Esta herramienta necesita especificar el método de reconocimiento facial usado en la base de datos entrenada (*Eigenfaces* o *Fisherfaces*).

4.3.2 Implementación del Archivo de Configuración del Sistema

Para fines prácticos de la investigación se implementó un archivo de configuración global, que permite personalizar cada una de las variables del sistema, así como seleccionar el tipo de algoritmo de reconocimiento a usar. Estos valores pueden ser consultados en pantalla del sistema (ver Fígura 9). El contenido del archivo de configuración es el siguiente:

```
/**
* Realiza una instancia a la clase Config::singleton y carga variables de configuración
*/
$config = config::singleton();

/**
Configuración global del sistema
*/
$config->set("url_base", "https://192.168.153.128/");
/** Nota: si activa la opción SSL se debe cambiar la url_base de http:// a https:// para que funcionen
correctamente los componentes del cliente flash */

/**
Activa la función del sistema para obligar a que todas las peticiones sean hechas a través de SSL
*/
$config->set("ssl", "on"); //on / off
/** Nota: si activa esta opción se debe cambiar la url_base de http:// a https:// para que funcionen
correctamente el componente del cliente flash */

/**
Activa el modo de depuración de errores, este permite imprimir todos los mensajes de error
*/
$config->set("modo_depuracion", "off"); //on / off
/** Nota: de forma predeterminada el modo de depuración está deshabilitado, para el correcto funcionamiento
del sistema debido a los mensajes de error
*/

/**
Ruta de directorios de uso del sistema
*/
$config->set("controllersFolder", "controllers/");
$config->set("modelsFolder", "models/");
$config->set("viewsFolder", "views/");
```

```
/**
Configuración de la conexión a la base de datos
*/
$config->set("dbhost", "127.0.0.1");
$config->set("dbname", "biometrico");
$config->set("dbuser", "root");
$config->set("dbpass", "18419356");

/**
Configuración base para aceptar nombres de usuarios
*/
//Tamaño mínimo de caracteres de nombre de usuario
$config->set("user_tam_min", "2");

// Tamaño máximo de caracteres en nombre de usuario
$config->set("user_tam_max", "20");

/**
Configuración base para aceptar contraseñas de usuarios
*/
//Tamaño mínimo de contraseña de usuario
$config->set("user_pass_min", "2");
//Tamaño máximo de contraseña de usuario
$config->set("user_pass_max", "10");

/**
Configuración del Modulo de Autenticación Biométrica
*/
//Activa el modulo de autenticación de reconocimiento facial
$config->set("face_recognition", "off"); //on / off

//Define el método de reconocimiento facial usado para el entrenamiento de la base de datos y reconocimiento de rostros
$config->set("metodo_reconocimiento", "eigenfaces"); // eigenfaces / fisherfaces

//Ruta de directorio para subir archivos temporales de imágenes para reconocimiento de usuarios y creación de cuentas
$config->set("uploaddir_img_test", "tmp/");

//Ruta de directorio para la base de datos de rostros
$config->set("base_datos_rostros", "base_datos_rostros/");

//Ruta de directorio para los programas de reconocimiento y procesamiento de rostros
//$config->set("bin", "bin/");

//Programa para el reconocimiento y procesamiento de rostros
$config->set("detectar_cara", "bin/detectar_cara"); // Limitación: Solo detecta y procesa un solo rostro en la imagen

//Programa para el entrenamiento de la base de datos de rostros
$config->set("entrenar_bd_rostros", "bin/entrenar_bd_rostros");

//Programa para el reconocimiento de rostros
```

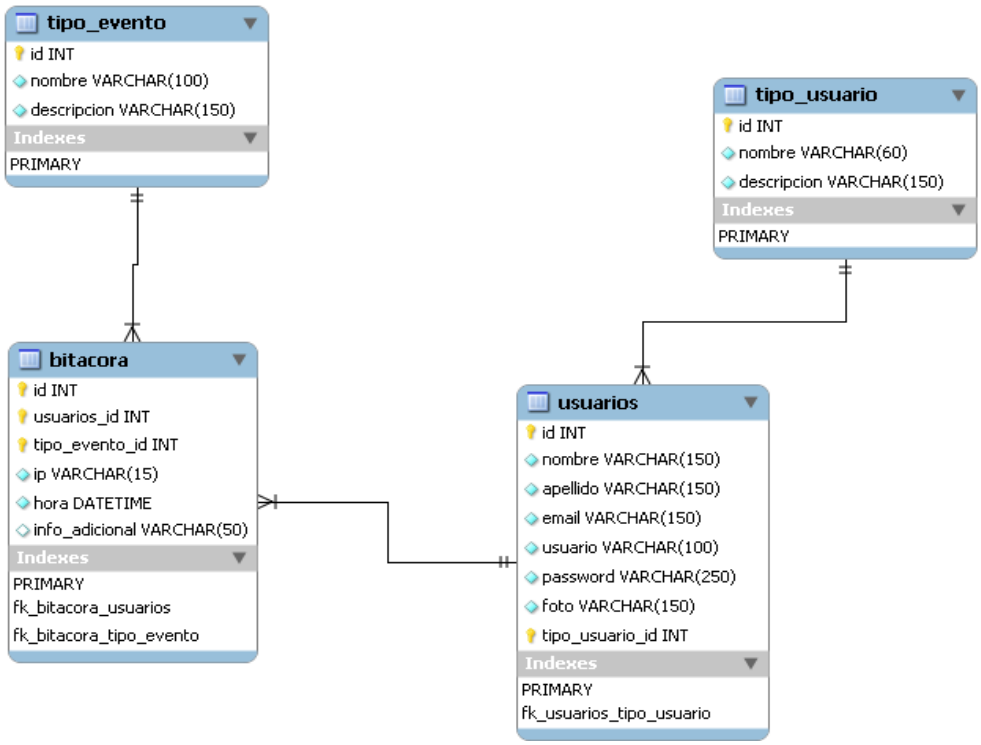
```
$config->set("reconocer_rostro", "bin/reconocer");  // Limitación: Sólo se limita a encontrar similitudes en imágenes con el mismo tamaño salvados en la bd 80x80

//Programa para procesamiento de imágenes - Escala de grises y procesamiento de iluminación
$config->set("procesar_imagen", "bin/procesar_imagen");

//Programa para el reconocimiento y procesamiento de rostros
$config->set("haarcascade", "bin/haarcascade_frontalface_alt.xml");

//Nombre de archivo para entrenamiento de rostros
$config->set("train_list", "bin/archivos_entrenamiento.csv");

//Nombre de archivo para guardar la base de datos del entrenamiento
$config->set("bd_reconocimiento", "bin/bd_rostros.yml");

//Valor mínimo de confianza para aceptar reconocimiento
$config->set("min_confianza", "0.30"); // : Valor de distancia más cercanos a imagen autenticación correcta,

//Cantidad de muestras de capturas de imágenes por usuario para entrenamiento de la bd
$config->set("cant_muestras", "10");

//Configuración para Aceptar Usuarios por Reconocimiento Facial

//Cantidad de muestras de capturas de imágenes para pruebas de reconocimiento
$config->set("max_capturas", "5");

//Porcentaje mínimo de aceptación para las capturas
$config->set("porcentaje_aceptacion", "20");
```

Para asegurar la confidencialidad del sistema, este archivo de configuración debe ser cifrado, ya que contiene información sensible, usada durante la conexión a la base de datos, como nombre de usuario y contraseña. Por tanto, se debe proteger el acceso a esta información, en caso de que el sistema sea comprometido de alguna forma. Para ello, se cifró simétricamente el archivo de configuración, usando la clave contenida en un archivo llamado: **mva-pkcs12.semilla.ssl**. Este archivo, a su vez, se cifró asimétricamente con la clave privada del servidor. Usando esta combinación de cifrado simétrico y asimétrico se puede brindar una protección robusta a la información contenida en este archivo.

4.3.3 Implementación de la Base de Datos

Aquí se presenta la estructura de las tablas que contendrán los datos almacenados por el prototipo. En la Fígura 3 se muestra el modelo de datos específicos para el gestor de base de datos MySQL.

Fígura 3 Modelo físico de datos



En la Fígura 3 se expanden las entidades del modelo entidad-relación en tablas físicas con los datos especificados, su tipo, opcionalidad, claves primarias (PK) y claves foráneas (FK). Se redujo la cantidad de tablas al mínimo con el fin de simplificar las consultas SQL, que internamente maneja el sistema.

4.3.3.1 Diccionario de Datos

El diccionario de datos es la explicación formal del modelo físico de datos, donde se detallan las tablas y campos con sus respectivas descripciones (ver Tablas 4, 5, 7 y 9)

Tabla 4 Tabla de Usuarios

Tabla	Usuarios	**Alias**	USR	**Descripción**	Los usuarios son entidades o personas que se crean para ingresar al sistema

Clave	**Campo**	**Tipo dato**	**Descripción**	**Ejemplo**
PK	ID	INT	Identificador primario de la tabla de usuarios	1
FK	TIPO_USUARIO_ID	INT	Hace referencia a un tipo de usuario	1
	NOMBRE	VARCHAR	Nombre del usuario	Juan
	APELLIDO	VARCHAR	Apellido del usuario	Pérez
	EMAIL	VARCHAR	Correo electrónico del usuario	Juan.perez@mail.com
	USUARIO	VARCHAR	Nombre de usuario que va a usar para ingresar al sistema	Juan.perez
	PASSWORD	VARCHAR	Contraseña cifrada del usuario	
	FOTO	VARCHAR	Ruta del directorio de muestras de fotos para el entrenamiento del algoritmo para el reconocimiento biométrico facial del usuario	Base_datos_foto/1

Tabla 5 Tabla de Tipo_usuario

Tabla	Tipo_usuario	**Alias**	TPU	**Descripción**	Los tipos de usuario son la forma para clasificar a los usuarios registrados en el sistema

Clave	**Campo**	**Tipo dato**	**Descripción**	**Ejemplo**
PK	ID	INT	Identificador primario de la tabla tipo_usuario	1
	NOMBRE	VARCHAR	Nombre del tipo de usuario. Para fines prácticos del sistema, se definieron dos tipos de usuario, que se describen en la tabla 5	Administrador
	DESCRIPCION	VARCHAR	Descripción del tipo de usuario	Usuario administrador del sistema, tiene permitido el ingreso al sistema de gestión de usuarios del sistema
Para ver los valores definidos para esta tabla ver Tabla 6				

Tabla 6 Valores definidos para la tabla Tipo_usuario

ID	Nombre	Descripción
1	administrador	Usuario administrador del sistema. Tiene permitido el ingreso al sistema de gestión de usuarios del sistema
2	usuario del sistema	Usuario del sistema. Tiene permitido el ingreso al sistema

Tabla 7 Tabla de Tipo_evento

Tabla	Tipo_evento	Alias	TPE	Descripción	Los tipos de eventos son para clasificar los eventos realizados por el usuario.

Clave	Campo	Tipo dato	Descripción	Ejemplo
PK	ID	INT	Identificador primario de la tabla de tipo_evento	1
	NOMBRE	VARCHAR	Nombre del tipo de evento. Para fines prácticos del sistema, se definieron seis tipos de eventos de usuario, que se describen en la tabla 7	ingreso al sistema
	DESCRIPCION	VARCHAR	Descripción del tipo de evento	registra el ingreso correcto al sistema
Para ver los valores definidos para esta tabla ver Tabla 8				

Tabla 8 Valores definidos para la tabla Tipo_evento

ID	Nombre	Descripción
1	ingreso autenticación básica de usuario	Registra el ingreso correcto de un usuario al sistema con autenticación básica
2	rechazo autenticación básica de usuario	Registra el rechazo de un usuario por parte del sistema en la autenticación básica
3	ingreso autenticación facial de usuario	Registra el ingreso correcto de un usuario al sistema con autenticación biométrica facial
4	rechazo autenticación facial de usuario	Registra el rechazo de un usuario por parte del sistema en la autenticación biométrica facial
5	ingreso al sistema	Registra el ingreso correcto al sistema

6	salida del sistema	Registra la salida del sistema

Tabla 9 Tabla de Bitácora

Tabla	Bitacora	**Alias**	BCA	**Descripción**	La bitácora son los eventos registrados en el sistema por un usuario.

Clave	**Campo**	**Tipo dato**	**Descripción**	**Ejemplo**
PK	ID	INT	Identificador primario de la tabla de usuarios	1
FK	USUARIOS_ID	INT	Hace referencia a un usuario	2
FK	TIPO_EVENTO_ID	INT	Hace referencia a un tipo de evento	4
	IP	VARCHAR	Dirección IP externa que registra el evento	192.168.1.9
	HORA	DATETIME	Hora en la cual se registra el evento	2004-09-01 21:12:36
	INFO_ADICIONAL	VARCHAR	Campo opcional de información adicional del registro del evento	modo_depuracion=on

4.3.4 Estructura de Directorios y Archivos Generados

En la Tabla 10 se muestra la estructura, en forma árbol, de los directorios de los recursos creados para el funcionamiento del sistema:

Tabla 10 Estructura de directorios y archivos generados

base_datos_rostros/	Directorio de la base de datos de imágenes de usuarios, tiene una estructura de archivos con el código de identificación de usuario, como se muestra a continuación:	
	1/	1.jpg
		2.jpg
		3.jpg
		4.jpg
		5.jpg
		6.jpg
		7.jpg

	8.jpg	
	9.jpg	
	10.jpg	
bin/	Contiene las aplicaciones creadas en C++ con la librería OpenCV para el procesamiento de imágenes y reconocimiento de caras usados por el servidor	
	archivos_entrenamiento.csv	Archivo generado por el sistema para el entrenamiento de caras de usuarios, contiene una lista en formato csv de archivos ordenados por usuario
	bd_rostros.yml	Base de datos del entrenamiento de caras conocidas
	detectar_cara	Programa para detectar y extraer caras alineadas de imágenes. Extrae la cara y procesa la imagen a escala de grises para ser usada en el reconocimiento
	entrenar_bd_rostros	Programa para el entrenamiento de caras, genera un archivo de base de datos usado en el reconocimiento
	haarcascade_frontalface_alt.xml	Archivo para la clasificación de objetos usado para la detección de caras
	procesar_imagen	Programa para procesar una imagen de escala de colores a escala de grises
	reconocer	Programa para reconocer una imagen procesada de cara, usando una base de datos de caras conocidas
cliente/	Contiene los componentes flash, creados con flex3, usados en el navegador Web del cliente	
	actualizar_foto.swf	Componente flash para actualizar fotos del usuario en el sistema
	detectar_cara.swf	Componente flash para pruebas de detección de caras
	login_biometrico.swf	Componente flash para la autenticación por biometría de reconocimiento facial
	reconocer_cara_tiempo_real.swf	Componente flash para pruebas de reconocimiento facial en tiempo real
controllers/	Contiene los archivos de los controladores del patrón de diseño MVC	
	basicoController.php	Procesa peticiones de usuarios sin privilegios administrativos del sistema.
	indexController.php	Procesa peticiones predeterminadas del sistema
	loginController.php	Procesa peticiones de ingreso al sistema, valida la pasarela de autenticación de dos factores
	sistemaController.php	Procesa peticiones de usuarios con privilegios administrativos del

	sistema, gestión de usuarios.
core/	Contiene las librerías, complementos y archivos de sistema usados en el patrón de diseño MVC
biometrico.php	Clase que implementa métodos para reconocimiento facial, hace uso de los programas contenidos en el directorio bin/
cifrado.php	Clase que implementa métodos para descifrar el archivo config.php usando la clave privada y certificado digital del servidor
config.php	Clase para agregar y obtener variables ya definidas en un archivo de configuración, hace uso del patrón de diseño singleton
controllerBase.php	Clase abstracta, que implementa el constructor base, para todas las clases de controladores
db.php	Clase que implementa métodos para la conexión de la base de datos
frontController.php	Clase para el manejo de peticiones de las clases controladores, archivo principal del patrón de diseño MVC
modelBase.php	Clase abstracta, implementa el constructor base para todas las clases de modelo
spdb.php	Clase que se extiende de db.php, implementa el patrón de diseño singleton
view.php	Clase para el manejo de vistas
css/	Directorio de la plantilla de estilo CSS
style.css	Plantilla de estilo css usada para el maquetado de la página Web
images/	Directorio de imágenes del sistema
adduser.png	Imagen de agregar usuario
article.png	Imagen de artículo
bg.gif	Imagen usada para fondo de la página
construccion.jpg	Imagen de construcción
folder.png	Imagen en carpeta
logout.png	Imagen de salida de usuarios
note_add.png	Imagen de nota
skull.png	Imagen de calavera
star.png	Imagen de estrella
top.gif	Imagen con letra Top
user.png	Imagen de usuario
warning.png	Imagen de advertencia
Webcam.jpg	Imagen de *Webcam*
x.png	Imagen de X
js/	Directorio de archivos Javascript

	jquery-1.7.2.min.js	Versión 1.7.2 de librería JQuery
models/	Contiene los archivos de los modelos del patrón de diseño MVC	
	bitacoraModel.php	Clase modelo que implementa métodos para registrar eventos del sistema
	loginModel.php	Clase modelo que implementa métodos para la consulta de datos de usuario del sistema
	sistemaModel.php	Clase modelo que implementa métodos para la gestión de usuarios del sistema
views/	Contiene los archivos de las vistas (código fuente HTML) del patrón de diseño MVC	
	face_loginView.php	Vista de objeto Flash para la autenticación biométrica facial
	footer.php	Vista de pié de página
	header.php	Vista cabecera de página de inicio
	indexView.php	Vista de cuerpo de página de inicio
	basico	Directorio de vistas de usuarios del sistema
	header.php	Vista cabecera de página de inicio de usuarios del sistema
	ver_datos_usuario.php	Vista de cuerpo de página de inicio de usuarios del sistema
	sistema	Directorio de vistas de usuarios administradores
	formulario_buscar_usuario.php	Vista que imprime formulario para buscar usuarios en el sistema
	formulario_modificar_usuario.php	Vista que imprime formulario para modificar usuarios al sistema
	formulario_pruebas.php	Vista que contiene los objetos Flash para realizar pruebas de reconocimiento facial
	formulario_usuario.php	Vista que imprime formulario para agregar nuevos usuarios al sistema
	header.php	Vista cabecera de página de inicio de usuarios administradores del sistema

		info.php	Vista que muestra información de archivo de configuración
		listado_usuarios.php	Vista de listado de usuarios del sistema
		ver_datos_usuario.php	Vista de datos de usuario del sistema
tmp/	Directorio de archivos temporales		
index.php	Archivo de inicio de la aplicación		
config.php	Archivo de configuración de la aplicación		

4.3.5 Implementación de la Entidad Certificadora

Para garantizar la comunicación segura entre el servidor y el cliente se hace uso de certificados digitales. Para ello, se instaló en la máquina de pruebas OpenSSL y se creó una autoridad certificadora local. A continuación, se describe el proceso de configuración de la misma.

4.3.5.1 Entidad Certificadora Local

Para la creación de la llave privada de la autoridad certificadora local se introdujeron los siguientes comandos (ver Tabla 11):

Tabla 11 Comandos para la creación de la entidad Certificadora

	Comando	**Descripción**
1	mkdir –p /usr/local/ssl	Se crea el directorio contenedor de certificados digitales de la entidad certificadora
2	cd /usr/local/ssl	Se cambia al directorio /usr/local/ssl
3	openssl req –newkey rsa:2048 –x509 –keyout cakey.pem –out cacert.pem –days 3650 –outform PEM	Se genera la llave privada de 2048 bits y se genera la llave pública de la entidad certificadora local por 3650 días
4	Country Name: VE State o Province Name: Tachira Locality Name: San Cristobal Organization Name: Biometrico Organizational Unit Name: Reconocimiento Common Name: Admin Email Address: admin@localhost	Datos de la entidad certificadora local
5	mkdir –p /usr/local/ssl/private	Se crea el directorio contenedor de la clave privada

6	mkdir –p /usr/local/ssl/newcerts	Se crea el directorio contenedor para la emisión de nuevos certificados
7	cp cakey.pem /usr/local/ssl/demoCA/private	Se copia la clave privada cakey.pem en la carpeta private
8	touch /usr/local/ssl/index.txt	Se genera el archivo index.txt, que contendrá una lista de los certificados públicos generados
9	echo 01 > /usr/local/ssl/serial	Se genera el archivo de nombre serial, que contendrá el consecutivo de certificados públicos generados

4.3.5.2 Llave Privada y Certificado Digital del Servidor Web

Para la creación de la llave privada y certificado digital del servidor Web se introdujeron los siguientes comandos (ver Tabla 12):

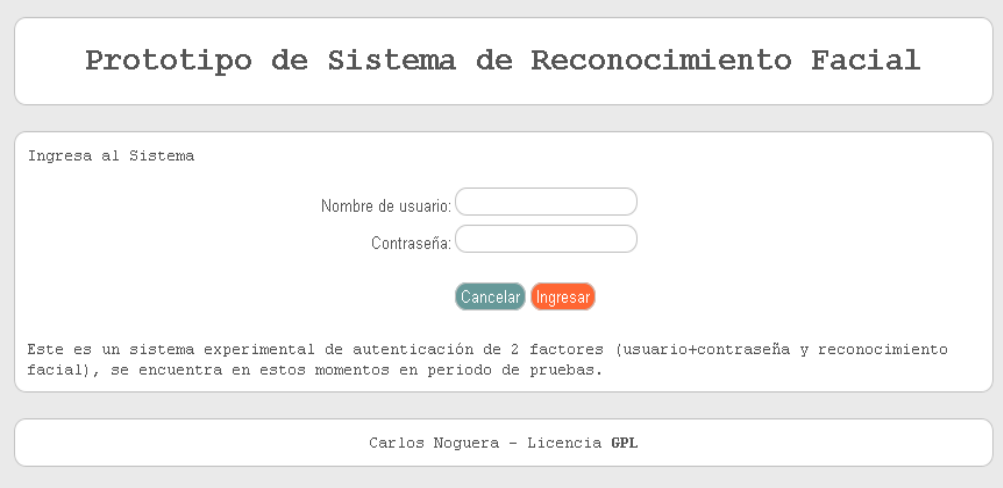
Tabla 12 Comandos para la creación del Certificado Digital del Servidor Web

	Comando	**Descripción**
1	cd /usr/local/ssl	Se cambia al directorio /usr/local/ssl
2	openssl req –config openssl.cnf –newkey rsa:2048 –keyout mva.key –out mva.csr –days 365 –nodes –outform PEM	Se indican los requerimientos de un servidor Web hacia la entidad certificadora. Genera la llave privada de 2048 bits y genera el requerimiento a la entidad certificadora por 365 días para el servidor Web
3	Country Name: VE State o Province Name: Tachira Locality Name: San Cristobal Organization Name: Servidor Prueba Reconocimiento Facial Organizational Unit Name: Reconocimiento Common Name: Carlos Email Address: carlos@localhost	Datos del certificado del servidor Web
4	openssl ca –config openssl.cnf –policy policy_anything –out mva.crt –infiles mva.csr	Se firma el requerimiento como la entidad certificadora
5	cp mva.crt /etc/apache2/conf/server.crt	Se copia el certificado digital del servidor mva.crt en la carpeta /etc/apache2/conf/ y se renombra a server.crt
6	cp mva.key /etc/apache2/conf/server.key	Se copia la clave privada del servidor mva.key en la carpeta /etc/apache2/conf/ y se renombra a server.key
7	cp cacert.pem /etc/apache2/conf/cacert.pem	Se copia el certificado digital de la entidad certificadora local cacert.pem en la carpeta /etc/apache2/conf/ y se renombra a cacert.crt

4.3.6 Implementación de la Interfaz Gráfica de Usuario

En la Fígura 4 se muestra la pantalla inicial de ingreso al sistema. Aquí el usuario debe ingresar su nombre de usuario y contraseña, y hacer clic en el botón **Ingresar**.

Fígura 4 Pantalla de Inicio del Sistema – Autenticación Básica



Una vez el usuario haya ingresado sus datos, el servidor comprueba si el usuario existe y si el hash de la contraseña coincide con el que se encuentra registrado en la base de datos. En caso de que exista error, se re-direcciona de nuevo a la página de inicio. En caso de éxito se re-direcciona a la pantalla de autenticación biométrica facial que se muestra en la Fígura 5. Allí, el componente flash, creado con flex3, debe capturar muestras de imágenes a través de la *Webcam*, para enviarlas vía petición HTTP POST al servidor Web, para que este valide la autenticación. Este componente sólo captura imágenes de fotos con rostros válidos.

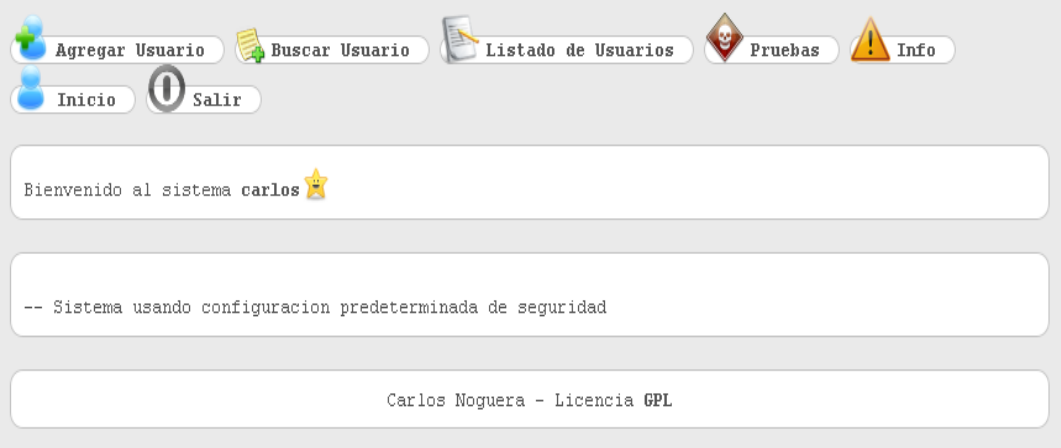
Fígura 5 Pantalla de Autenticación Biométrica Facial

La seguridad en la comunicación del componente Flash, según Adobe [43], se da con la seguridad que está implementada en la misma página HTML, y al usar el protocolo SSL, con certificados digitales, se puede decir que esta comunicación viaja a través de la red de forma segura.

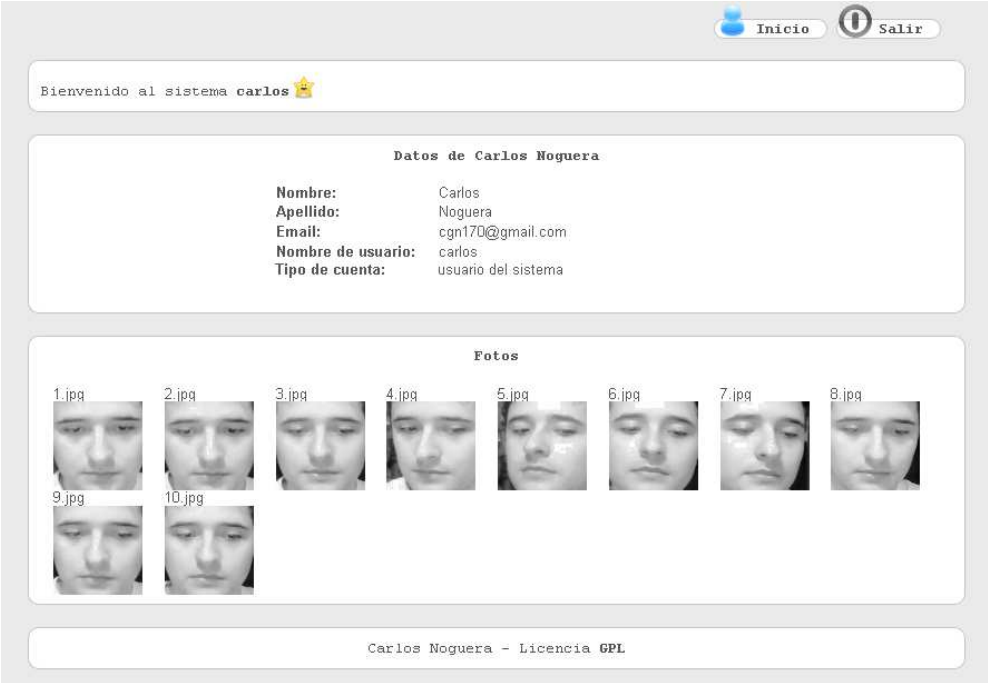
Fígura 6 Pantalla de Autenticación Biométrica Facial - Re direccionamiento

Al finalizar el proceso de captura de imágenes se recibe la respuesta del servidor en formato JSON, como se muestra en la Fígura 6. Si el sistema no logra reconocer al usuario que está ingresando, se registra un evento de seguridad, se elimina cualquier sesión que exista y se re-direcciona a la página de inicio. En caso de éxito, el sistema re-direcciona a la página correspondiente del usuario, es decir, si pertenece al tipo de usuario **administrador** se envía a la página de administración que se muestra en la Fígura 7, pero si es **usuario del sistema**, se envía a la página básica que se muestra en la Fígura 8, que presenta los atributos del usuario, así como las fotos que fueron capturadas y procesadas para el entrenamiento de la base de datos.

Fígura 7 Pantalla Principal Administrativa del Sistema

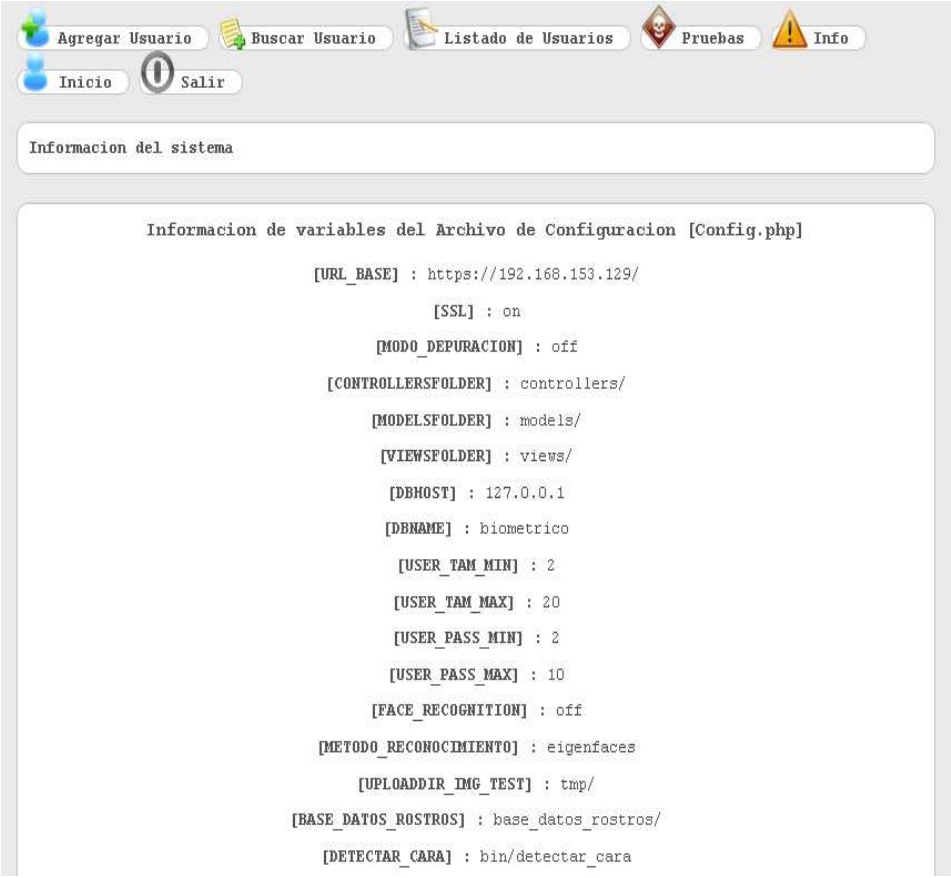


Fígura 8 Pantalla Principal de Inicio de Usuarios del Sistema



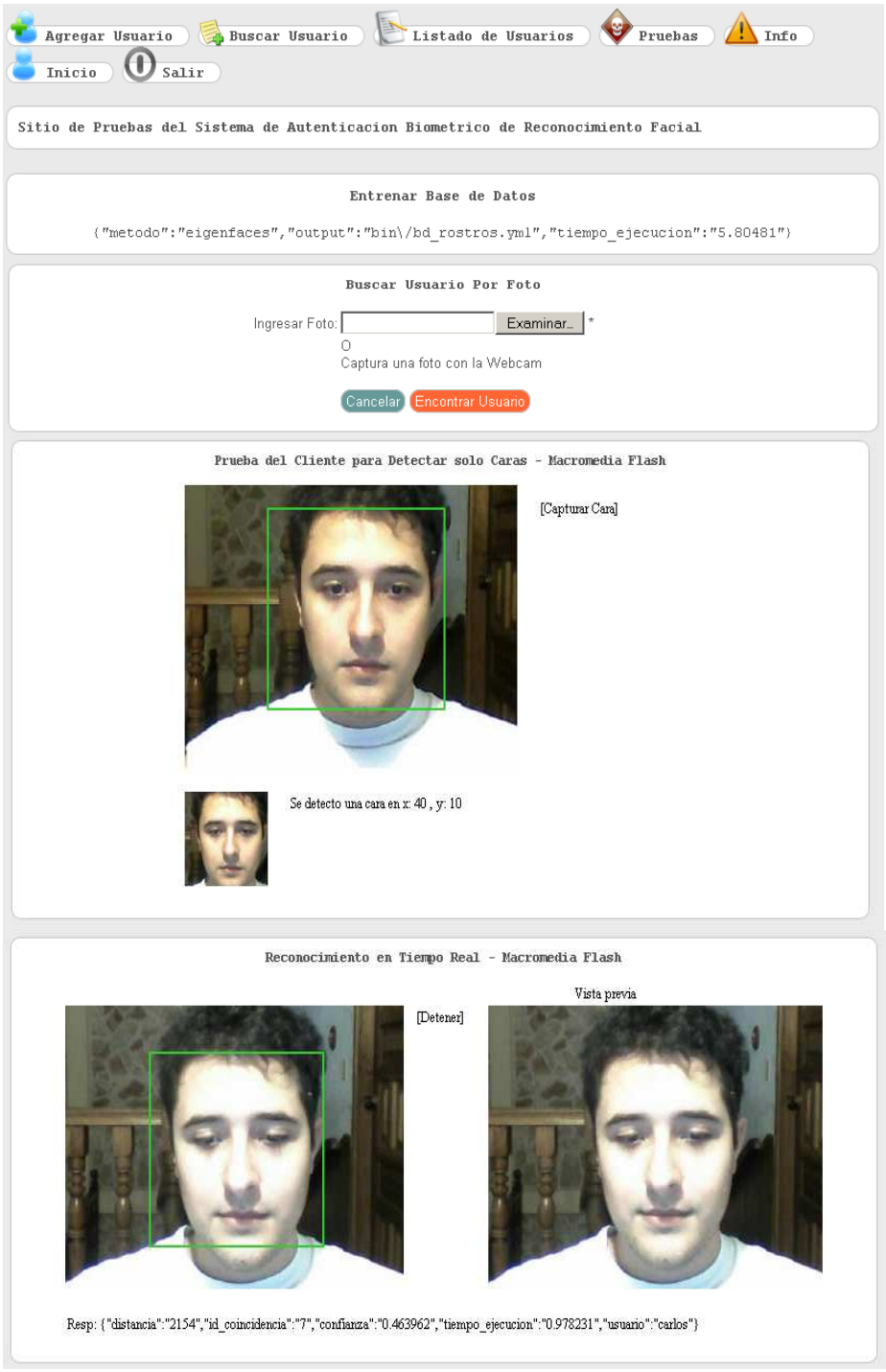
En la pantalla administrativa del sistema se observan diferentes opciones que permiten la gestión de los usuarios. En la Fígura 9 se muestra la pantalla de información, a la que se accede tras hacer clic en la opción **Info**, que se encuentra en la parte superior derecha de la Fígura 7. Aquí se detalla información relevante del archivo de configuración (explicado en la sección 5.3.2), además información del estado y configuración del servidor.

Fígura 9 Pantalla de Información del Sistema



Para fines prácticos del proyecto, se creó un módulo para pruebas del servicio Web, que permite entrenar la base de datos de usuarios conocidos según el método de reconocimiento facial especificado en el archivo de configuración y además realizar pruebas de los algoritmos de autenticación usados. A este módulo se accede haciendo clic en la opción **Pruebas**, del menú de administración del sistema. En la Fígura 10 se muestra la pantalla del módulo de pruebas. La primera opción permite entrenar la base de datos de usuarios, para esto se debe hacer clic en el hipervínculo **Entrenar Base de Datos**, y esta devuelve el resultado vía AJAX.

Fígura 10 Pantalla del Módulo de Pruebas



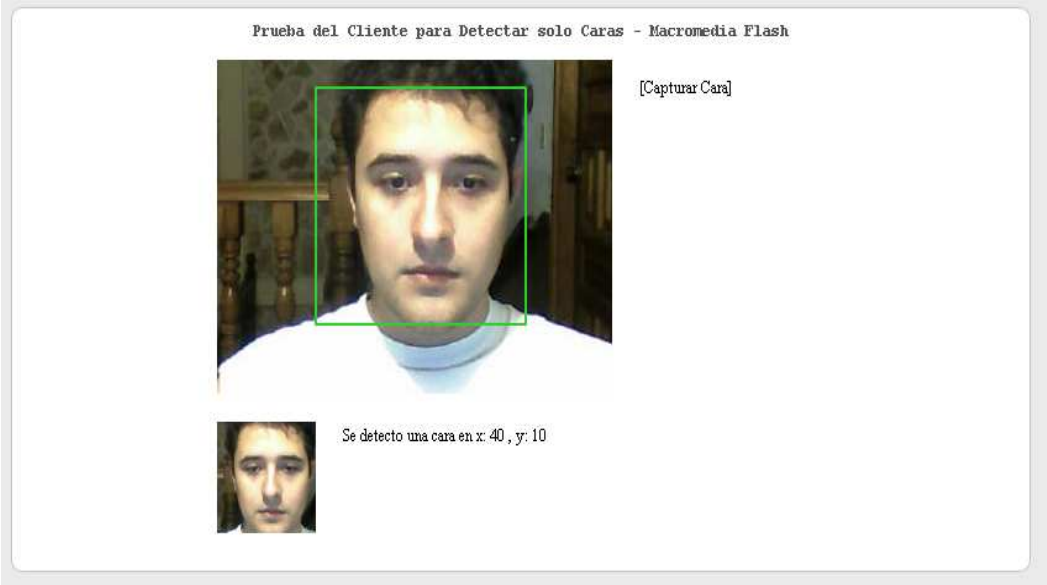
La Fígura 11 muestra la pantalla de formulario de pruebas que se encuentra dentro del Módulo de pruebas (Fígura 10). En esta se detalla un formulario para probar el algoritmo de

autenticación definido en el archivo de configuración del sistema. Permite validar si una imagen subida al servidor tiene una cara válida, para lo cual se hace clic en el botón **Examinar** y se selecciona la imagen. Luego, se hace clic en el botón **Encontrar Usuario.** El sistema extrae y procesa esa cara y realiza el proceso de comparación con la base de datos. Devuelve, finalmente, el resultado vía AJAX.

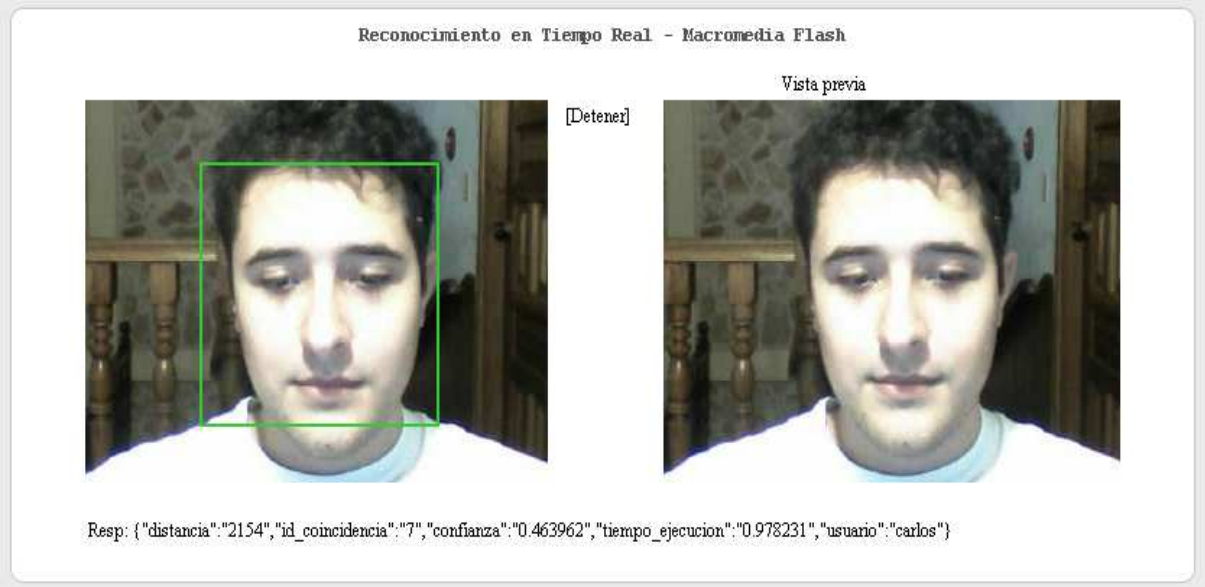
Fígura 11 Pantalla de Formulario de Pruebas de Imágenes

A fin de complementar la pantalla de pruebas se crearon dos componentes Flash para realizar pruebas de detección de caras (ver Fígura 12) y reconocimiento facial en tiempo real (ver Fígura 13), estos componentes se encuentran en la parte inferior del Módulo de pruebas (Fígura 10). Como requisito para su funcionamiento necesita una *Webcam* previamente instalada en el cliente. Para iniciar la ejecución de pruebas en cada uno de los componentes se debe hacer clic en **[Capturar Cara]** e **[Iniciar],** respectivamente en cada componente.

Fígura 12 Pantalla de programa Flash para detección de caras



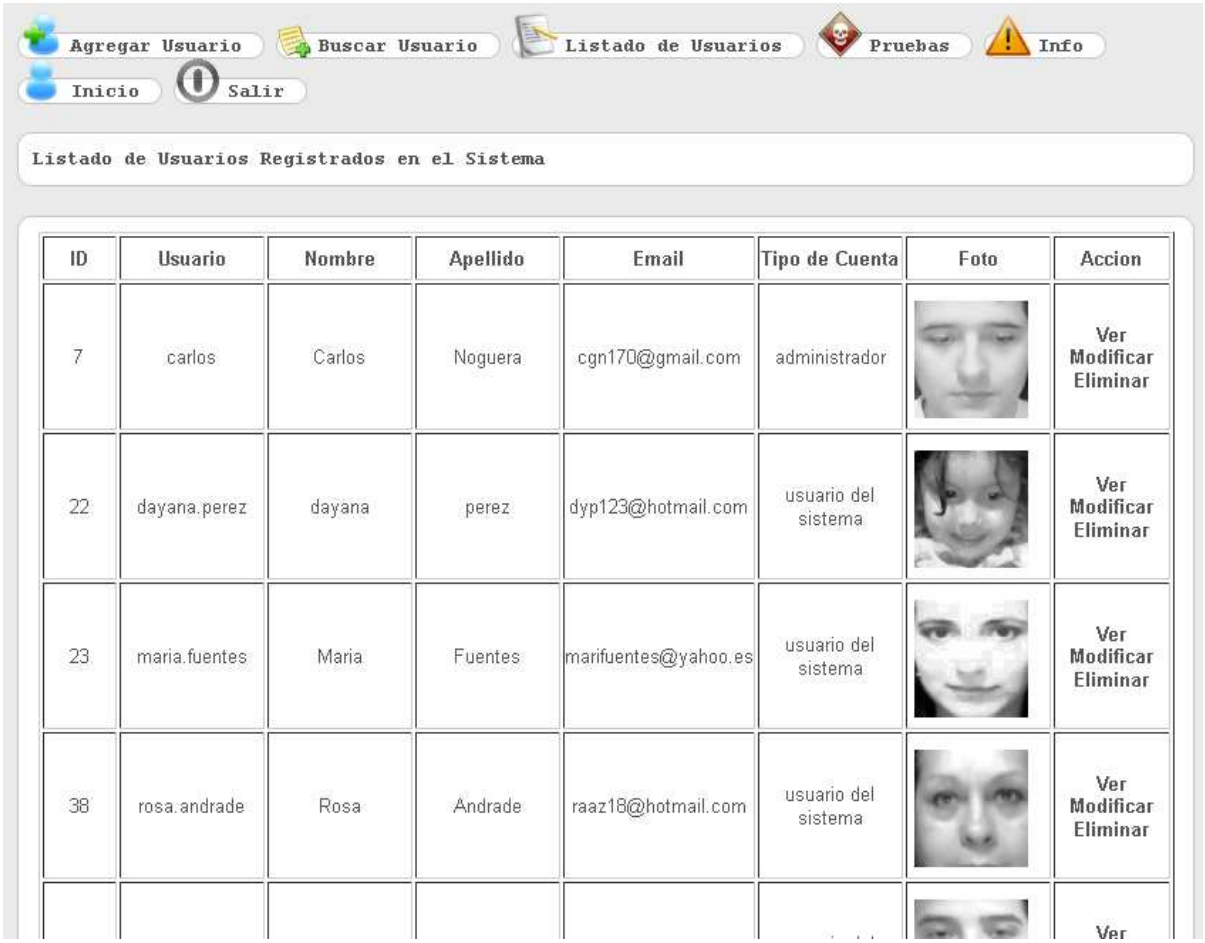
Fígura 13 Pantalla de programa flash para reconocimiento de caras



En la Fígura 14 se muestra la pantalla de listado de usuarios registrados en el sistema, a la que se accede haciendo clic en la opción **Listado de Usuarios**, ubicado en la parte superior media de la pantalla. Esta lista permite, además de mostrar algunos datos de usuarios,

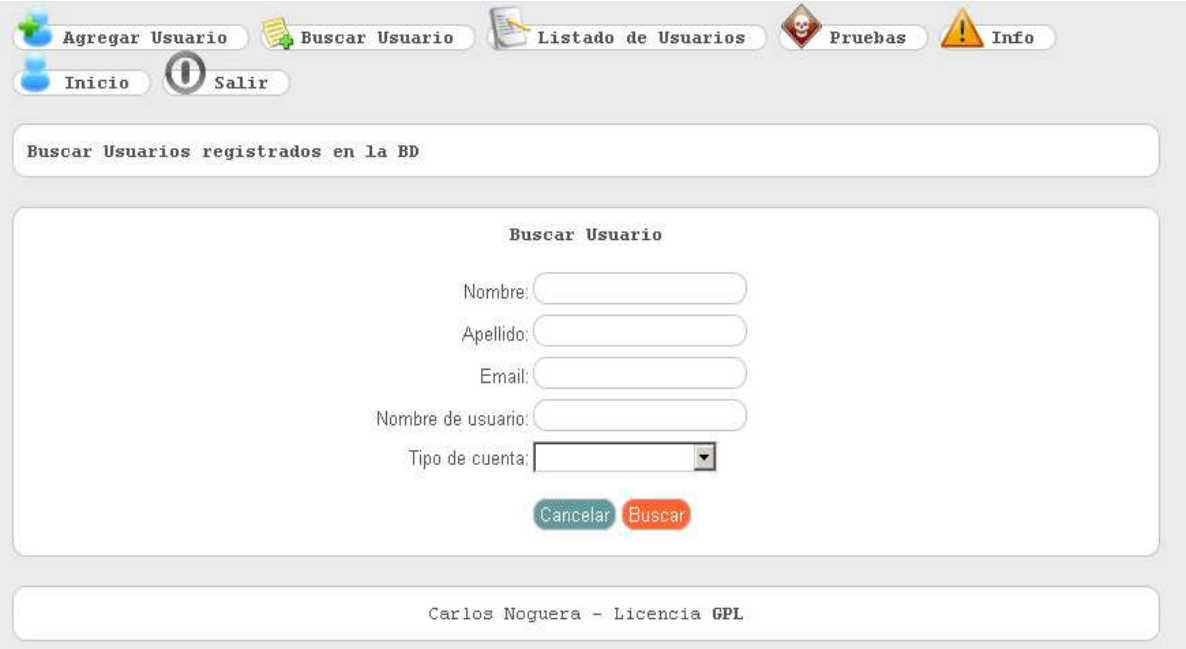
realizar acciones como: ver, modificar o eliminar usuarios.

Fígura 14 Pantalla de Listado de Usuarios del Sistema



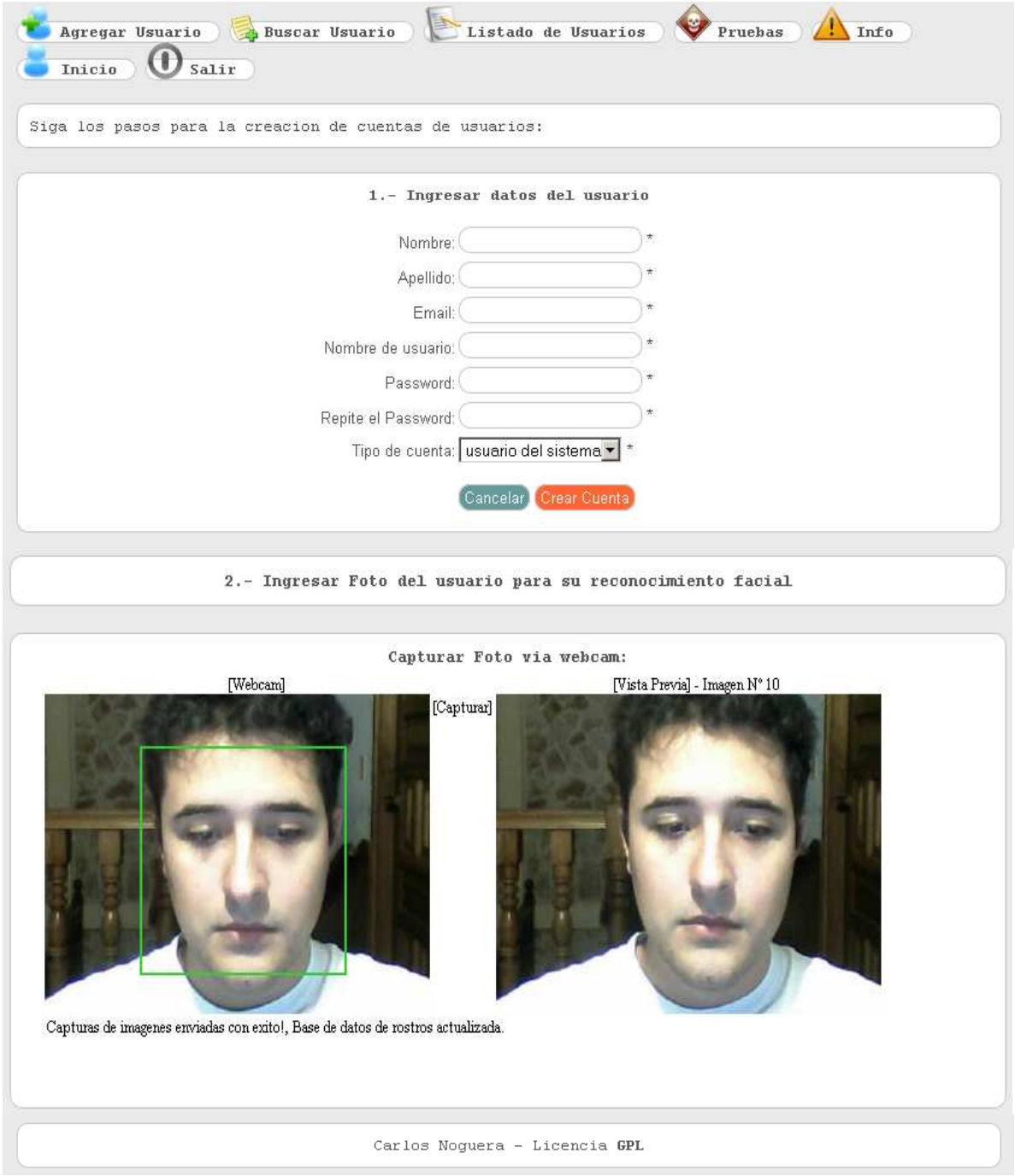
Para la búsqueda rápida de usuarios se creó la pantalla de búsqueda de usuario (ver Fígura 15), a la que se accede haciendo clic en la opción **Buscar Usuario**. En esta se pueden buscar usuarios según: su nombre, apellido, email, nombre de usuario y tipo de cuenta. Para ello se llenan los datos que se conozcan del usuario buscado y luego se hace clic en el botón **Buscar**.

Fígura 15 Pantalla para Buscar Usuarios en la Base de datos



En la Fígura 16 se muestra la pantalla para agregar un usuario al sistema, a la que se accede haciendo clic en la opción **Agregar Usuario**, y que presenta un formulario de atributos para la creación de cuentas de usuario. Esta pantalla hace un extenso uso de AJAX para validar los datos ingresados.

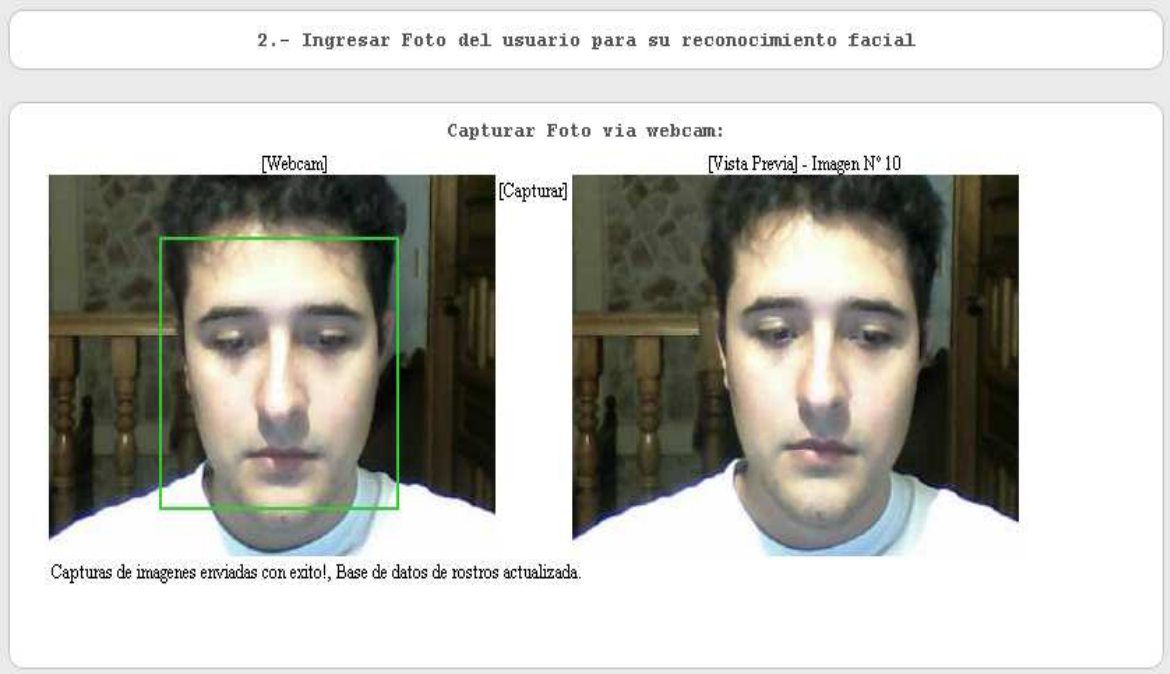
Fígura 16 Pantalla para Agregar Usuarios al Sistema



Una vez ingresados los datos del usuario, tras hacer clic en el botón **Crear Cuenta**, el sistema inicia el componente Flash, que se muestra en la Fígura 17. Para realizar la captura de imágenes a través de la *Webcam*, el usuario debe hacer clic en la etiqueta **[Capturar]** y

el componente Flash captura automáticamente la cantidad de imágenes especificadas en el archivo de configuración. Al finalizar la captura de imágenes, el componente flash invoca al módulo de entrenamiento del algoritmo de reconocimiento facial y, de esta manera, agrega las imágenes del usuario capturadas a la base de datos de rostros conocidos.

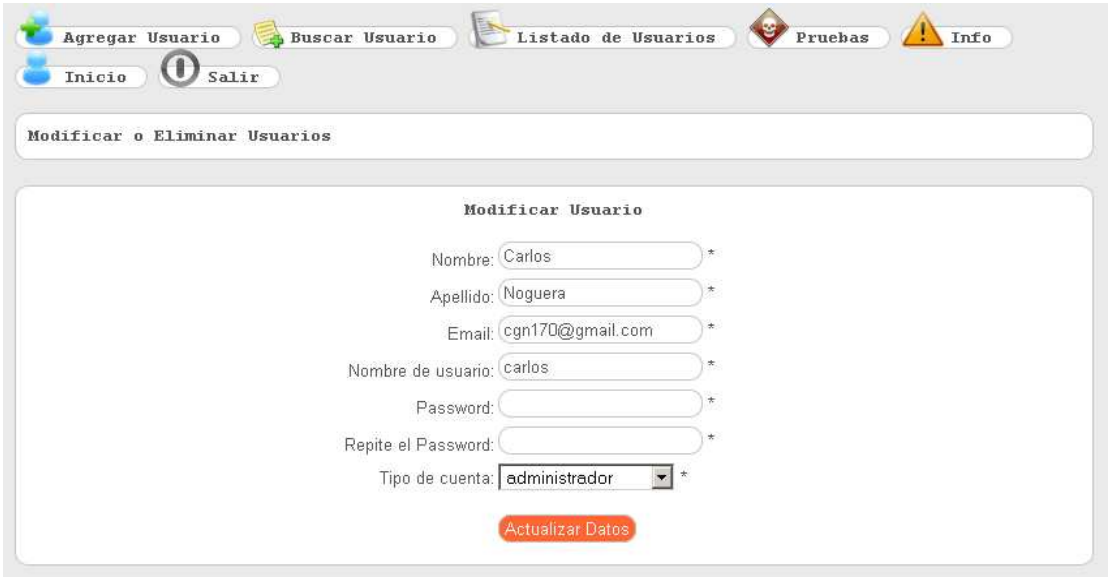
Fígura 17 Pantalla de Programa Flash para la captura de imagenes para entrenamiento



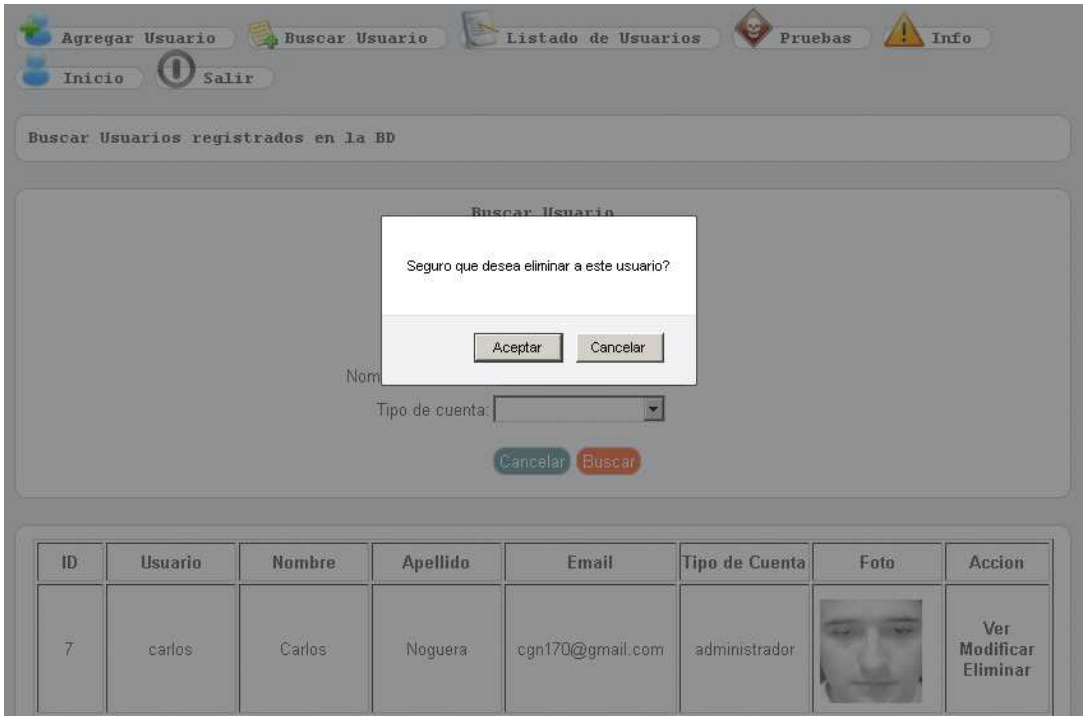
La Fígura 18 muestra la pantalla para modificar los datos de usuario. Esta pantalla puede ser accedida haciendo clic en la opción **Modificar**, mostrada junto al listado de resultados, luego de realizar una búsqueda de usuarios en la pantalla de búsqueda de usuarios (Fígura 15). Esta pantalla es similar a la pantalla para crear usuarios. También, hace uso del componente flash para captura de imágenes a través de la *Webcam* y el entrenamiento del algoritmo de reconocimiento facial (en el archivo de configuración del sistema se especifica que método de reconocimiento se usa en el módulo de autenticación biométrica). Así mismo, para eliminar un usuario es necesario hacer clic en la opción **Eliminar** en el listado de usuarios o en el resultado de búsqueda de usuarios, como se muestra

Fígura 19

Fígura 18 Pantalla para Modificar Usuarios



Fígura 19 Dialogo de Confirmación para Eliminar un Usuario



5. PRUEBAS Y RESULTADOS OBTENIDOS

5.1 EVALUACIÓN DEL SISTEMA PROPUESTO

Con el propósito de demostrar el rendimiento y utilidad del prototipo desarrollado se realizaron diferentes pruebas, a fin de analizar cada uno de los escenarios que se puedan presentar.

5.1.1 Rendimiento de los Métodos de Reconocimiento Facial

Al implementar los métodos de reconocimiento facial, se realizaron diferentes pruebas a fin de determinar su rendimiento como sistema de autenticación, estos algoritmos son ampliamente estudiados y se han realizado diferentes estudios sobre su rendimiento en diferentes condiciones y escenarios.

Marcelo [26] realizó un estudio comparativo de métodos basados en sub-espacios aplicados al reconocimiento de caras, en el cual analiza el rendimiento de los algoritmos *Eigenfaces* y *Fisherfaces* poniéndolos a prueba con una base de datos de caras conocidas como la AT&T [48], esta base de datos contiene imágenes de 40 individuos diferentes, a los cuales se les tomó 10 fotos en cada sesión con variaciones de luz, expresión facial, punto de vista y en algunos casos con presencia o no de anteojos. A continuación se muestra en la Tabla 13 un resumen de resultados de la taza de aciertos promediados de una prueba *Leaving one out*[10]:

Tabla 13 Resultados de prueba *Leaving one out* para distintos valores de C (números de individuos diferentes)

Fuente: Análisis comparativo de métodos basados en sub-espacios[26]

C=	5	10	15	20	25	30	35	40
Eigenfaces	0.9800	0.9600	0.9667	0.8950	0.9000	0.8833	0.8371	0.7675
Fisherfaces	0.9600	0.9800	0.9867	0.9650	0.9880	0.9900	0.9829	0.9825

[10] *Leaving one out* es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Esta técnica en particular implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar qué tan preciso es el modelo que se llevará a cabo en la práctica[26].

Partiendo de este análisis comparativo como punto de referencia del rendimiento de estos algoritmos, se realizó una prueba *Leaving one out* con 8 usuarios registrados y 10 muestras por usuario. El tamaño de las muestras por usuario fue de 80x80 pixeles. A continuación se muestra, en la Tabla 14, el resumen de los resultados promediados obtenidos de las pruebas realizadas al sistema implementado (para mayor información sobre las muestras parciales obtenidas en esta prueba ver Anexo A y Anexo B):

Tabla 14 Resultado de prueba Leaving one out para valores actuales del sistema

Método	**Número de usuarios en base de datos**	**Imágenes por usuario**	**Porcentaje de reconocimiento en promedio**	**Tiempo promedio de entrenamiento de base de datos**	**Tiempo promedio de reconocimiento por muestra**
Eigenfaces	8	10	71%	1,3130 segundos	0,4436 segundo
Fisherfaces	8	10	44%	0,43 segundos	0,0511 segundos

Debido a las limitantes en tiempo y recursos humanos se limitó el conjunto de pruebas de cambios de iluminación y resolución de *Webcam* a un solo usuario, se hizo una observación de 100 muestras de autenticación biométrica facial con una resolución de *Webcam* de 320x240, los resultados (porcentaje de aciertos) se presentan en la Tabla 15:

Tabla 15 Resultado de pruebas de reconocimiento del sistemas con cambios de iluminación

Método	**Habitación con Luz Natural**	**Habitación con Lámpara Iluminada**	**Habitación con Luz Escasa**	**Habitación Sin Iluminación**
				
Eigenfaces	93%	89%	91%	85%
Fisherfaces	56%	78%	24%	18%

Como se observa en la Tabla 14 el método *Fisherfaces* implementado en la librería Libfacerec no muestra un rendimiento adecuado cuando se realizan capturas de imágenes a través de una *Webcam*, en cambio *Eigenfaces* tiene un rendimiento aceptable para el reconocimiento de rostros conocidos en diferentes condiciones de luz.

Para verificar cual resolución era la más óptima para el reconocimiento de rostros se hizo una observación de 100 muestras de autenticación biométrica facial para cada una de las distintas resoluciones soportadas por una *Webcam* genérica, los resultados (porcentaje de aciertos) se muestran en la Tabla 16:

Tabla 16 Resultado de pruebas de reconocimiento del sistemas con cambios de resolución en Webcam

Método	**Resolución de Webcam**				
	160x120	176x144	320x240	352x288	640x480
Eigenfaces	25%	26%	91%	88%	34%
Fisherfaces	58%	15%	58%	59%	28%

Como se observa en la Tabla 16, el método *Eigenfaces* con la resolución de *Webcam* de 320x240 pixeles arrojó la mejor tasa de aciertos, y por eso se seleccionó como característica principal en la configuración predeterminada del sistema.

5.1.1.1 Prueba de Identificación de Gemelos

Adicionalmente se realizó una prueba para conocer cuál método podría distinguir mejor rostros conocidos en caso de presentarse el registro de gemelos (dos usuarios con características de rasgos faciales similares). Para esta prueba se realizó una observación de 100 muestras de autenticación biométrica facial y con una resolución de *Webcam* de 320x240 identificando al usuario Carlos, los resultados obtenidos se muestran en la Tabla 17:

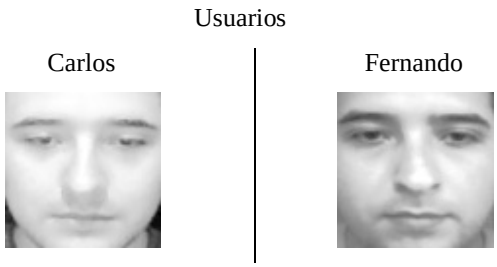


Tabla 17 Resultados prueba de identificación de gemelos

Método	**Aciertos**	**Rechazos**	**Confusión**
Eigenfaces	96%	3%	1%
Fisherfaces	63%	24%	13%

Como se observa en la Tabla 17 el método *Fisherfaces* fue el que obtuvo el mayor porcentaje de confusión en reconocer a los dos usuarios, por su parte *Eigenfaces* fue el método con menor porcentaje de error.

5.1.2 Rendimiento del Módulo de Autenticación Biométrico

Para calcular el rendimiento del módulo de reconocimiento facial, se realizó una observación de 100 muestras de autenticación biométrica facial y con una resolución de *Webcam* de 320x240, los resultados se presentan en la Tabla 18:

Tabla 18 Porcentaje de Falso Rechazo y Falso Aceptación

Método	**Porcentaje de Acierto**	**Porcentaje de Falso Rechazo**	**Porcentaje de Falsa Aceptación**
Eigenfaces	91%	7%	2%
Fisherfaces	63%	25%	12%

Como se observa en la Tabla 18, el método *Eigenfaces* es el que menor porcentaje de falso rechazo y falsa aceptación presenta, aunque es una prueba preliminar no es concluyente, puesto que harían falta más pruebas a fin de determinar la verdadera robustez de este sistema.

5.1.2.1 Tiempo de Ingreso

El tiempo de ingreso de un usuario al sistema usando autenticación biométrica de reconocimiento facial es estimado por el tiempo aproximado de descarga de los componentes de la página Web junto con el tiempo aproximado de subida de datos y respuesta del servidor. Se realizaron diferentes pruebas de tiempo durante el tráfico transmitido en la autenticación de un usuario arbitrario con diferentes anchos de banda, los resultados obtenidos se muestran en las Tablas 19,20,21,22,23:

Tabla 19 Estadísticas de Tráfico Transmitido durante la Autenticación de un Usuario con un ancho de banda de 28.8 Kbps

	Tráfico Entrante	**Tráfico Saliente**
Tasa de transferencia actual:	346 b/s	21 b/s
Tasa de transferencia promedio:	1,43 Kb/s	957 b/s

Tasa de transferencia máxima:	4,44 Kb/s	18,8 Kb/s
Total		
Recibidos	91,3 KB	
Enviados	59,9 KB	
Total de datos transferidos	151 KB	
Tiempo Transcurrido	64 segundos	

Tabla 20 Estadísticas de Tráfico Transmitido durante la Autenticación de un Usuario con un ancho de banda de 56 Kbps

	Tráfico Entrante	**Tráfico Saliente**
Tasa de transferencia actual:	0 b/s	0 b/s
Tasa de transferencia promedio:	2,23 Kb/s	1,46 Kb/s
Tasa de transferencia máxima:	7,88 Kb/s	18 Kb/s
Total		
Recibidos	98,4 KB	
Enviados	64,2 KB	
Total de datos transferidos	163 KB	
Tiempo Transcurrido	44 segundos	

Tabla 21 Estadísticas de Tráfico Transmitido durante la Autenticación de un Usuario con un ancho de banda de 64 Kbps

	Tráfico Entrante	**Tráfico Saliente**
Tasa de transferencia actual:	233 b/s	10 b/s
Tasa de transferencia promedio:	2,87 Kb/s	1,83 Kb/s
Tasa de transferencia máxima:	9,12 Kb/s	18,9 Kb/s
Total		
Recibidos	94,8 KB	
Enviados	60,4 KB	
Total de datos transferidos	155 KB	
Tiempo Transcurrido	33 segundos	

Tabla 22 Estadísticas de Tráfico Transmitido durante la Autenticación de un Usuario con un ancho de banda de 128 Kbps

	Tráfico Entrante	**Tráfico Saliente**
Tasa de transferencia actual:	603 b/s	3,64 Kb/s
Tasa de transferencia promedio:	4,24 Kb/s	2,54 Kb/s
Tasa de transferencia máxima:	17,7Kb/s	18,1 Kb/s
Total		
Recibidos	97,6 KB	
Enviados	58,5 KB	
Total de datos transferidos	156 KB	
Tiempo Transcurrido	23 segundos	

Tabla 23 Estadísticas de Tráfico Transmitido durante la Autenticación de un Usuario con un ancho de banda de 192 Kbps

	Tráfico Entrante	**Tráfico Saliente**

Tasa de transferencia actual:	351 b/s	179 Kb/s
Tasa de transferencia promedio:	5,18 Kb/s	2,54 Kb/s
Tasa de transferencia máxima:	26,4 Kb/s	15,8 Kb/s
Total		
Recibidos	104 KB	
Enviados	50,8 KB	
Total de datos transferidos	154 KB	
Tiempo Transcurrido	20 segundos	

Como se puede apreciar en los resultados de esta prueba se puede concluir que a medida que el ancho de banda incrementa disminuyen los tiempos de carga de datos y, por lo tanto, el tiempo necesario para la autenticación de usuarios. Para un mayor rendimiento durante la autenticación biométrica vía Web se necesita un ancho de banda mayor a 64 Kbps.

5.1.2.2 Requisitos Mínimos de Funcionamiento

Después de las pruebas de rendimiento del sistema se puede concluir que los requisitos mínimos para un correcto funcionamiento del módulo de autenticación biométrica, para los usuarios clientes, además de un navegador Web que tenga habilitados Javascript y Flash, son los siguientes (ver Tabla 24):

Tabla 24 Requisitos Mínimos del Equipo Cliente

Procesador:	Intel(R) Pentium(R) 4 CPU 2.66GHz
RAM:	512 MB
Sistema Operativo:	Microsoft Windows XP Professional Service Pack 3 (build 2600) (5.1.2600)
Webcam:	Marca Genérica
Resolución:	320x240 pixeles
Versión Adobe Flash	10 o superior
Ancho de Banda	28.8 Kbps o mayores

Además, las pruebas determinaron que el usuario debe ingresar al sistema haciendo uso de una habitación con la iluminación adecuada o en condiciones similares que cuando fue ingresado al sistema, con esto se logra una menor tasa de error.

5.1.3 Ataques al Sistema Propuesto

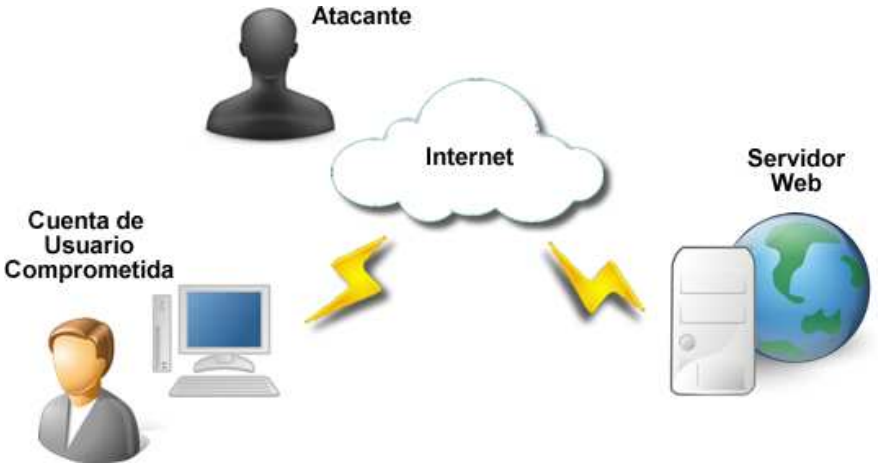
Se muestra a continuación la descripción de los diferentes tipos de ataques que podrían

afectar al sistema propuesto, así como su respectivo escenario.

5.1.3.1 Robo de Contraseña de Usuario

En esta prueba se evalúa el siguiente escenario de ataque: las credenciales de un usuario (nombre de usuario y contraseña) han caído en manos de un atacante e intenta ingresar al servidor Web. El sistema debe validar que el usuario que ingresa al sistema es realmente quien dice ser (ver Fígura 20).

Fígura 20 Escenario de una cuenta de usuario que tiene su contraseña comprometida



Información de la cuenta de usuario comprometida	
Nombre de Usuario:	**Contraseña:**
zaida	Azul5871

El atacante usa la información comprometida del usuario para ingresar al sistema Web como se observa en la Fígura 21:

Fígura 21 Atacante ingresando a través de credenciales comprometidas

Luego de ingresar las credenciales, el sistema necesita verificar la identidad del usuario. En ese momento se solicita acceso a la *Webcam* del usuario, como se muestra en la Fígura 22:

Fígura 22 Componente flash que verifica la identidad del usuario

Al finalizar el proceso de reconocimiento de caras, el sistema busca en la base de datos de usuarios conocidos la imagen del usuario y la compara con la que se encuentra en la base de datos como se muestra a continuación:

Foto asociada a la cuenta de usuario comprometida	Foto del atacante
	

De acuerdo al criterio mínimo de coincidencia, usado en el reconocimiento, el sistema no reconoce al usuario que intenta ingresar al sistema, y por lo tanto, se niega el ingreso.

5.1.3.2 Suplantación de Datos Biométricos – Imagen Impresa

Esta prueba evalúa el mismo escenario anterior, sólo que ahora el atacante intenta engañar al módulo de autenticación biométrico usando una fotografía impresa del usuario, en este escenario el atacante debe conocer al usuario de la cuenta comprometida y además tener acceso a una fotografía frontal de buena resolución de ella (320x240 pixeles como mínimo). (Ver Fígura 23)

Fígura 23 Escenario de ataque con fotografía impresa

Este tipo de ataque es grave para el sistema debido a que, el sistema usa métodos de reconocimiento facial de dos dimensiones por lo que usar una fotografía del usuario al momento de la autenticación podría engañar al sistema y permitir su ingreso al sistema.

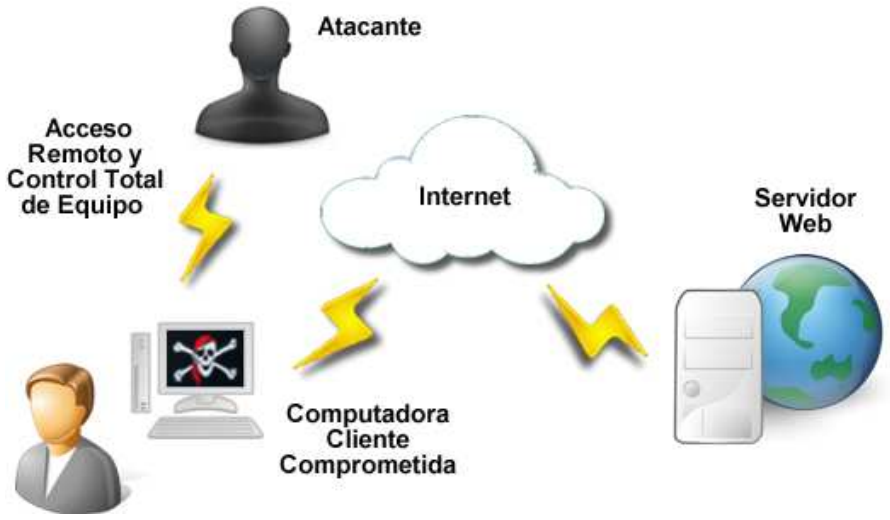
Como se observa en la Fígura 24 se hizo uso de una fotografía impresa del usuario víctima para engañar al sistema, durante los primeros intentos no se logró ingresar al sistema debido al ángulo e iluminación de la imagen, pero luego de varios intentos de fallidos se logró engañar al sistema de autenticación biométrica facial obteniendo el ingreso legítimo al sistema.

Fígura 24 Uso de imagen impresa para saltar la autenticación biométrica

5.1.3.3 Suplantación de Datos Biométricos – Captura Imagen Webcam

En este escenario se comprometen las credenciales del usuario por medio del equipo que usa para ingresar al sistema, por medio de la instalación de un software malicioso el atacante tiene acceso remoto y control total del equipo, y así, puede comprometer las credenciales de ingreso del usuario. (Ver Fígura 25)

Fígura 25 Escenario de ataque por medio de software malicioso en cliente

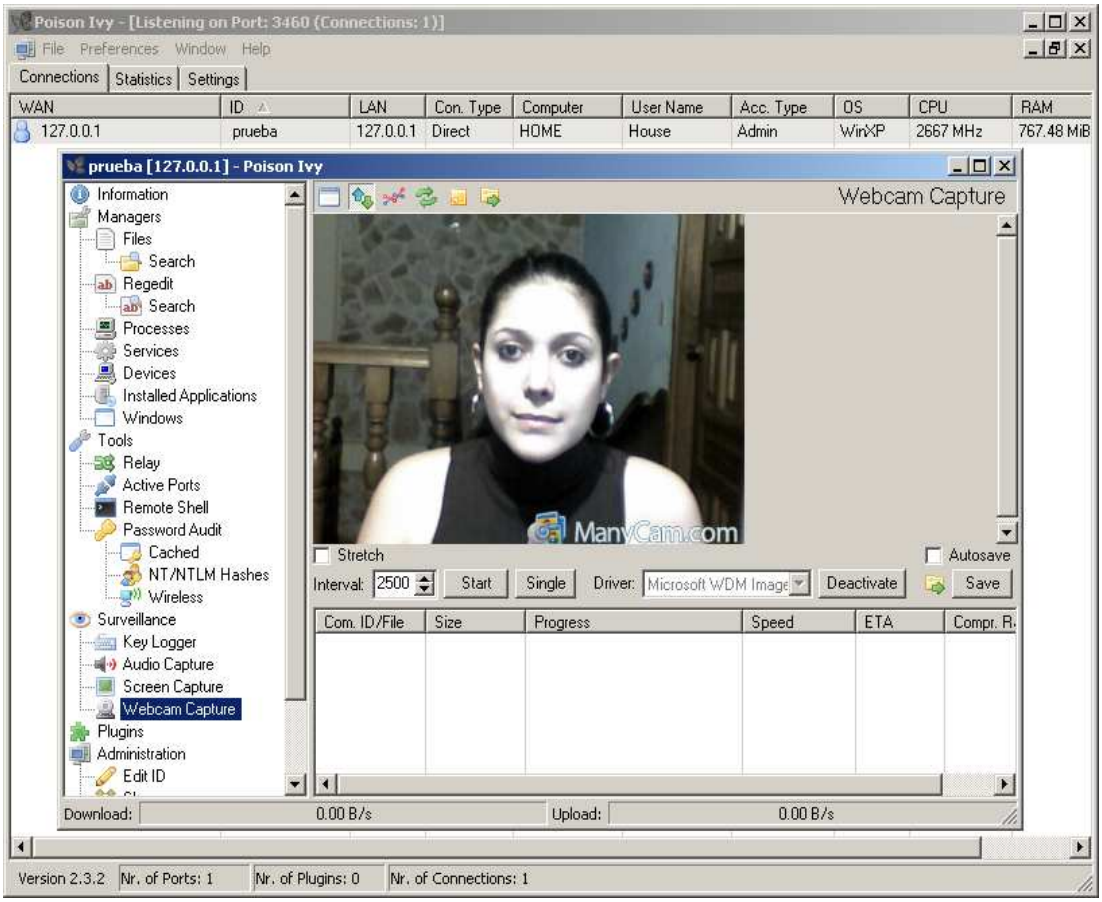


Para esta prueba se hizo uso de RAT Poison Ivy [49], es una herramienta de administración remota que puede instalarse de forma oculta dentro de procesos activos en el sistema y brindar acceso remoto y control total del equipo. Este escenario muestra como un atacante puede obtener las credenciales de ingreso de un usuario mediante el uso de un software malicioso.

Como punto de partida se considera que el equipo víctima ya ha sido comprometido con un software malicioso, en este caso en particular de la herramienta RAT Poison Ivy, una vez conectado con el equipo remoto la herramienta permite un control total sobre el sistema operativo, permitiendo acceso a procesos, consola del sistema, acceso al registro de Windows, etc. Entre otras funciones permite capturar teclas, impresiones de pantallas,

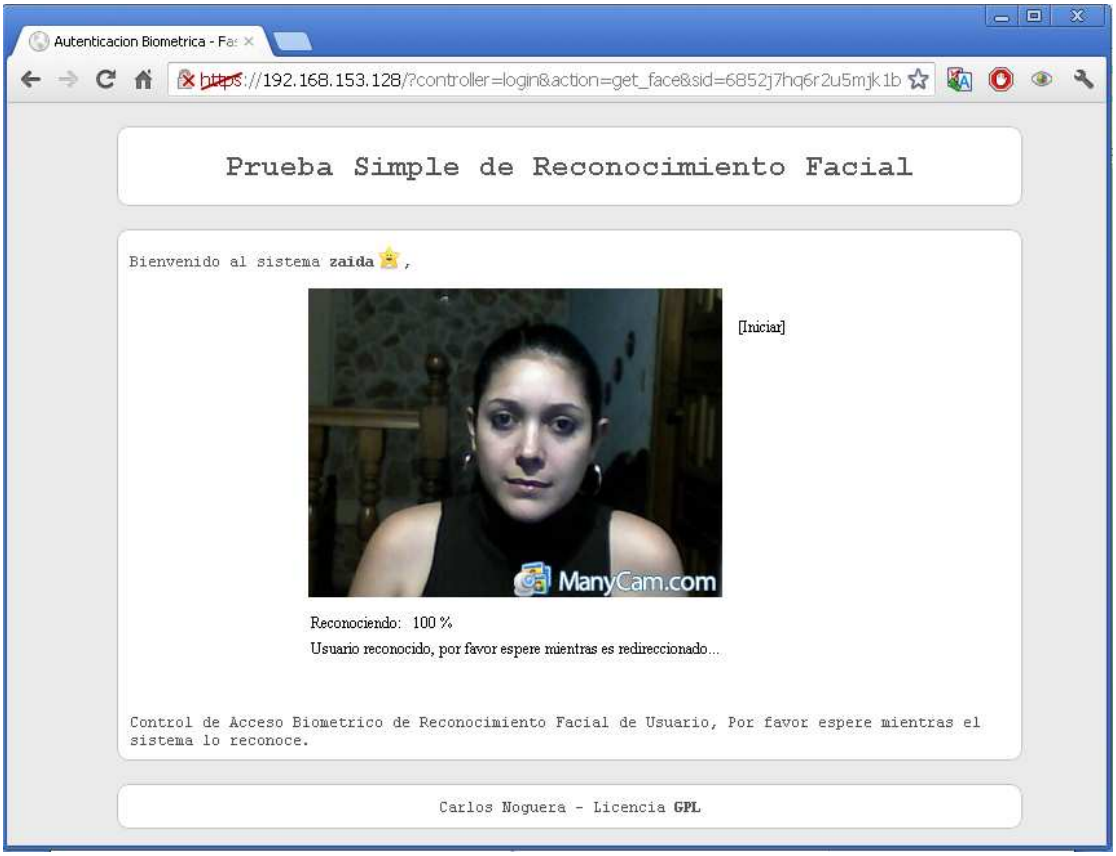
capturas de sonidos (si existe micrófono instalado) y realizar capturas de imágenes a través de la Webcam como se observa en la Fígura 26.

Fígura 26 Captura de imágenes de Webcam a través del equipo comprometido



Esta característica permite al atacante capturar imágenes que pueden ser usadas para engañar al sistema de reconocimiento facial. Se muestra a continuación como el atacante, por medio del uso de la herramienta ManyCam [50] puede simular una Webcam usando las imágenes capturadas para engañar al sistema de autenticación:

Fígura 27 Uso de imágenes capturadas para saltar la autenticación biométrica

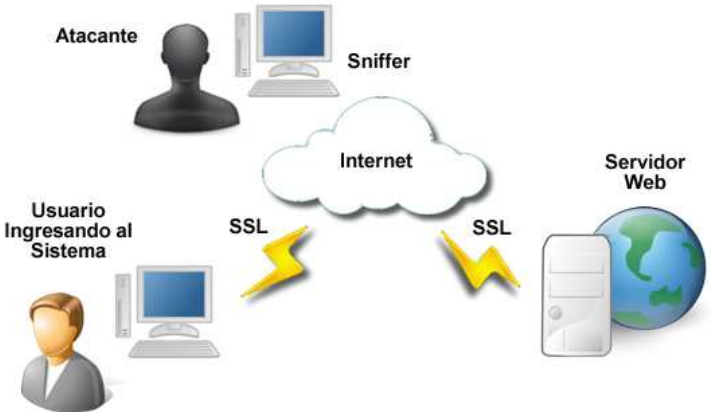


Como se observa en la Fígura 27 el atacante logró engañar al sistema de reconocimiento facial usando imágenes capturadas previamente y así obteniendo el ingreso legítimo al sistema.

5.1.3.4 Suplantación de Datos Biométricos – Sniffer

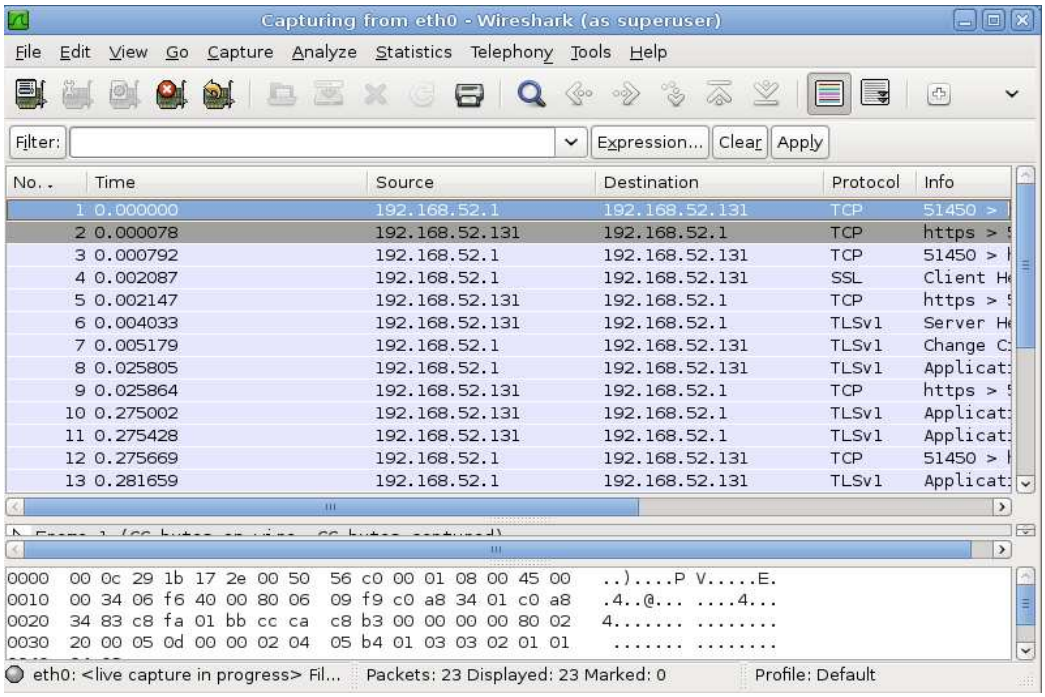
En este escenario se comprueba la seguridad en los datos que viajan a través de la red durante la comunicación entre el cliente y servidor (ver Fígura 28).

Fígura 28 Escenario de ataque por medio del uso de un sniffer en la red



En este escenario un atacante trata de comprometer los datos de un usuario que intenta autenticarse en el sistema por medio del uso de un *sniffer*, durante el proceso de autenticación biométrica. Para esta prueba se hizo uso de la herramienta Wireshark [51], este software permite realizar capturas de tráfico en una red local para posteriormente analizarlo (ver Fígura 29).

Fígura 29 Wireshark capturando trafico cifrado



Como se observa en la Fígura 29, todo el trafico capturado durante la autenticacion entre un usuario y el servidor viaja cifrado por medio del uso de certificados digitales, por lo que se hace muy difícil el análisis de tráfico y así como la captura de datos que pueda comprometer las credenciales del usuario.

6. CONCLUSIONES Y RECOMENDACIONES

Las principales conclusiones de este trabajo son las siguientes:

- Aunque existen en la actualidad algunas propuestas que hacen uso de la autenticación biométrica de reconocimiento facial en plataformas Web, todavía se presentan algunos retos que considerar para su implementación como: diversos modelos de *Webcam*, con diferentes resoluciones, que afectan la captura de imágenes; las condiciones de iluminación afectan el rendimiento de los algoritmos de reconocimiento junto con elementos adicionales (lentes, estilo de peinado y barba), así como la posición y pose de la cara.

- La velocidad de la comunicación, durante el proceso de autenticación, no es una gran limitante, ya que el ancho de banda usado no es muy grande (28.8 Kbps o superior), pudiendo ser implementado, incluso, como mecanismo de autenticación en dispositivos móviles.

- Dada la naturaleza de los datos biométricos transmitidos, es necesario garantizar que el equipo usado para ingresar al sistema cuente con las medidas de seguridad apropiadas como políticas de seguridad estrictas junto con el uso de aplicaciones como antivirus, firewall y parches de seguridad del sistema operativo actualizados a fin de evitar ser comprometido.

- Aunque este sistema implementa el reconocimiento facial como segundo factor de autenticación es relativamente inseguro, ya que el atacante podría usar una imagen asociada a la cuenta comprometida y así lograr engañar al mecanismo de autenticación (como se demostró en las pruebas), es por eso que es necesario implementar otro mecanismo complementario como reconocimiento de voz (biometría multimodal) a fin de verificar la identidad del usuario.

- En la actualidad las redes sociales son un común denominador en la vida cotidiana de

muchos de los usuarios de Internet, haciendo que compartir información personal como fotos, videos sea una tarea común, por lo que un atacante podría encontrar información relacionada con un usuario y usarla a su beneficio.

- La principal desventaja del uso de la biometría como método de autenticación remoto es que si los datos biométricos llegasen a ser comprometidos en alguna forma no pueden ser cambiados fácilmente, es por esto que debe garantizarse fuertes medidas de seguridad en los equipos clientes donde son capturados los datos biométricos.

- Como se demostró en los resultados de las pruebas, los requisitos de funcionamiento del sistema requeridos por lo aplicación para un funcionamiento correcto son: una resolución de cámara de 320x240, además de condiciones de luz similares a cuando fueron tomadas las fotos para el entrenamiento de la base de datos.

- Entre los algoritmos de reconocimiento facial que fueron implementados en el sistema *Eigenfaces y Fisherfaces*, el que obtuvo el mejor rendimiento referente a mayor tasa a aciertos y menores márgenes de error durante las pruebas realizadas fue *Eigenfaces*. Este método demostró, además, ser lo suficientemente robusto en diferentes condiciones de luz y resoluciones de cámara.

Como recomendaciones, se pueden señalar las siguientes:

- El servidor podría implementar un sistema de detección de intrusos que verifique la cantidad de ingresos fallidos en la autenticación biométrica (podría usar el sistema básico de *logs* que se implementó en el proyecto) y así lanzar una alarma.

- Debido a que este sistema usa métodos de reconocimiento facial en dos dimensiones el sistema no sabe distinguir una imagen de captura con una foto falsa, es por eso que este sistema por sí solo no es seguro, así que se podría adicionar otro elemento biométrico (como reconocimiento por voz) a fin de brindar seguridad al sistema.

- El producto resultante es un software funcional pero en un versión de pruebas, hace falta realizar una mayor cantidad de pruebas y desarrollo, a fin de obtener un producto más robusto que pueda ser usado con una mayor tasa de confianza.

- Es más fácil poder realizar mejoras a algo que está hecho y es tangible, que construir algo perfecto en el primer intento y desde cero basado en un diseño. Por lo tanto, el código fuente como el diseño son aquellos elementos que deben ser sujetos a críticas y mejoras. El software es una masa maleable, un organismo vivo que evoluciona constantemente. Por lo que debe aprovecharse esta característica para trabajarlo de forma iterativa e incremental ya que existen muchos aspectos difíciles de visualizar desde un modelo.

- Liberar el proyecto como software libre permite brindarle vida más allá de la realización de esta investigación, además de evitar que termine siendo un software sin utilidad. El que sea software libre permite ingresar esta herramienta a comunidades de usuarios interesados en seguridad de la información además de eliminar las trabas de uso impuestas por las licencias comerciales y permitir su mejoramiento continuo por medio del proceso de desarrollo del software libre.

REFERENCIAS

[1] Red Académica y de Investigación Española - RedIRIS, *"Autenticación de Usuarios"*, fecha de acceso: 05/03/2012, http://www.rediris.es/cert/doc/unixsec/node14.html

[2] F. Dinei, H. Cormac y C. Baris, *"Do Strong Passwords Accomplish Anything?"* en Actas del HOTSEC, Junio, 2007, fecha de acceso: 06/03/2012, http://static.usenix.org/event/hotsec07/tech/full_papers/florencio/florencio.pdf

[3] SANS Institute InfoSec Reading Room, *"Two-Factor Authentication: Can You Choose the Right One?"*, fecha de acceso: 20/01/2012, http://www.sans.org/reading_room/whitepapers/authentication/two-factor-authentication-choose-one_33093.

[4] W. Zhao, R. Chellappa, *"Image-based Face Recognition: Issues and Methods"*, Ed. B. Javidi, M. Dekker, 2002, pp. 375-402.

[5] V. Paul., J. Michael,*"Robust Real-Time Face Detection "*, International Journal of Computer Vision, Vol. 57, No. 2, Mayo 2004, pp.151–173.

[6] V. Iago, Alba, Castro. José, "*Detección de caras y localización de características faciales*", presentado en XXII Simposium Nacional de la Union Científica Internacional de Radio, Tenerife, España, Septiembre, 2007.

[7] O. Enrique, G. Elisardo, G. Carmen, C. José Luis, *"Biometrics for Web Authentication: an Open Source Java-Based Approach "*, Grupo de Procesamiento de Señales, Universidad de Vigo, Junio 2007, fecha de acceso: 20/01/2012, http://www.gts.tsc.uvigo.es/PRESA/papers/SWB07.pdf.

[8] BioAPI Consortium, *"BioAPI"*, fecha de acceso: 20/01/2012, http://www.bioapi.org/

[9] F. Carolina, Y. Leonardo, *"Reconocimiento facial en línea, una arquitectura Open Source para validación de identidad de estudiantes para el LMS MOODLE"*, IV Congreso de la CiberSociedad 2009, fecha de acceso: 25/01/2012, http://www.cibersociedad.net/congres2009/es/coms/trabajo-en-progreso-reconocimiento-facial-en-linea-una-arquitectura-open-source-para-validacion-de-

identidad-de-estudiantes-para-el-lms-moodle/990/

[10] G. Elisardo ,*"Desarrollo de Soluciones Cliente-Servidor para la Verificación Biométrica de Identidad y Monitorización en Plataformas Web: Aplicación a Tele-enseñanza "*. Tesis Doctoral, Universidad de Vigo, Campus Universitario, Orense y Pontevedra, España, 2010.

[11] Willow Garage, *"OpenCVWiki"*, fecha de acceso: 25/01/2012, http://opencv.willowgarage.com/wiki/.

[12] Departamento de Procesamiento de Señales y Comunicaciones – Universidad de Vigo, *"PRESA Project"*, fecha de acceso: 26/01/2012, http://www.gts.tsc.uvigo.es/PRESA/.

[13] C. Ruchir, *"Biometrics for Global Web Authentication: an Open Source Java/J2EE-Based Approach"*, International Journal of Computer Theory and Engineering, Vol. 3, No. 2, Abril 2011, pp. 324-327.

[14] BioID GmbH, *"BioID"*, fecha de acceso: 10/02/2012, http://www.bioid.com/

[15] Symantec Corporation, *"Symantec Internet Security Threat Report Trends for 2010"*, fecha de acceso: 02/03/2012, https://www4.symantec.com/mktginfo/downloads/21182883_GA_REPORT_ISTR_Main-Report_04-11_HI-RES.pdf.

[16] EMC Corporation, *"Online Fraud Report – January 2012"*, fecha de acceso: 03/03/2012, http://www.rsa.com/solutions/consumer_authentication/intelreport/11635_Online_Fraud_report_0112.pdf.

[17] PhishTank Community, *"PhishTank | Join the fight against phishing"*, fecha de acceso: 05/03/2012, https://www.phishtank.com.

[18] The Open Security Foundation's DataLossDB, *"OSF DataLossDB | Data Loss News, Statistics, and Research"*, fecha de acceso: 05/03/2012, http://www.datalossdb.org.

[19] Identidadrobada, *"Infografia de Estadisticas sobre robo de Identidad"*, fecha

de acceso: 20/03/2012, http://www.identidadrobada.com/estadisticas-sobre-robo-de-identidad-infografia/.

[20] Eset Latinoamerica, *"El robo de identidad y sus cifras en América Latina"*, fecha de acceso: 03/04/2012, http://www.onedigital.mx/ww3/2011/04/09/el-robo-de-identidad-y-sus-cifras-en-amrica-latina/.

[21] CNN Expansion, *"Mexico el octavo en robo de identidad"*, fecha de acceso: 05/04/2012, http://www.cnnexpansion.com/economia/2011/03/24/mexico-el-octavo-en-robo-de-identidad.

[22] Oasis-Open, *"Web Services Security"*, fecha de acceso: 15/03/2012, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss.

[23] The Open Web Application Security Project, *"OWASP"*, fecha de acceso: 18/03/2012, https://www.owasp.org/index.php/Main_Page.

[24] T. Anderson, *"The Theory and Practice of Online Learning"*. Athabasca University, 2nd ed. Ed. AU Press. 2009. Biometrics Glossary, fecha de acceso: 20/01/2012, http://www.biometrics.gov/Documents/Glossary.pdf.

[25] National Science Technology Council, *"Biometrics History"*, fecha de acceso: 02/06/2012, http://www.biometrics.gov/documents/biohistory.pdf

[26] A. Marcelo, *"Análisis comparativo de métodos basados en sub-espacios aplicados al reconocimiento de caras"*, Universidad de Valencia, España, Septiembre 2006, fecha de acceso: 20/01/2012, www.uv.es/marjoari/pdf/definitivo.pdf

[27] Y. Ming-Hsuan, K. David, y A. Narendra, *"Detecting Faces in Images: A Survey"*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 1, Enero 2002, fecha de acceso: 03/06/2012, http://vision.ai.uiuc.edu/mhyang/papers/pami02a.pdf

[28] G. Yang y T. S. Huang, *"Human face detection in a complex background"*, Pattern Recognition vol. 27, No. 1, Enero 1994, pp. 53-63.

[29] T. Matthew, P. Alex, *"Eigenfaces for Recognition"*, Journal of Cognitive

Neuroscience, Vol. 3, No. 1. 1991, fecha de acceso: 16/02/2012, http://www.face-rec.org/algorithms/PCA/jcn.pdf

[30] Bytefish, *"Libfacerec"*, fecha de acceso: 10/06/2012, http://www.bytefish.de/dev/libfacerec/

[31] Y. Cheng, K. Liu, J. Yang, Y. Zhuang, y N. Gu, "*Human Face Recognition Method Based on the Statistical Model of Small Sample Size"* SPIE Proc. Intelligent Robots and Computer Vision X: Algorithms and Technology, 1991, pp. 85-95

[32] B. Peter, H. João, K. David, *"Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection"*, IEEE transactions on pattern analysis and machie intelligence, Vol. 19, No. 7, Julio 1997, pp.711-720.

[33] Y. Ming-Hsuan, *"Face Recognition Using Kernel Methods"* Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 215.

[34] G. Shalini, M. Mia K. y B, Alan C. *"Anthropometric 3D Face Recognition*", International Journal of Computer Vision, Vol. 90 Issue 3, Diciembre 2010, pp. 331-349.

[35] B. Vinay y B. S Shreyas. *"Face Recognition Using Gabor Wavelets"* Proceedings of the IEEE Asilomar Conference On Signals, Systems and Computers, pp. 593–597, 2006.

[36] N. Ara V., *"Hidden Markov models for face recognition"*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing Vol. 5, pp. 2721- 2724, 1998

[37] G. Edwards, T. Cootes, y C.Taylor, *"Face Recognition Using Active Appearance Models"*, Proceedings European Conf. Computer Vision, pp. 582-595, 1998.

[38] Open Computer Vision Library, *"OpenCV"*, fecha de acceso: 10/06/2012, http://code.opencv.org/projects/opencv

[39] jQuery Foundation, *"JQuery"*, fecha de acceso: 09/06/2012, http://jquery.com

[40] Libspark, *“AS3 Marilena”*, fecha de acceso: 10/06/2012, http://www.libspark.org/wiki/mash/Marilena

[41] Quasimondo, *“Optimizing flash based face detection”*, fecha de acceso: 10/06/2012, http://www.quasimondo.com/archives/000687.php

[42] *“OpenSSL Project”*, fecha de acceso: 10/06/2012, http://www.openssl.org/

[43] Adobe Systems ,*“Applying Flex Security”*, fecha de acceso: 11/06/2012, http://livedocs.adobe.com/flex/3/html/help.html?content=security2_14.html

[44] Tony Marston, *“The Model-View-Controller (MVC) Design Pattern for PHP”*, fecha de acceso: 22/07/2012, http://www.tonymarston.net/php-mysql/model-view-controller.html

[45] Sourcemaking, *“Singleton Design Pattern”*, fecha de acceso: 22/07/2012, http://sourcemaking.com/design_patterns/singleton

[46] Json.org, *“Introducción a JSON”*, fecha de acceso: 25/07/2012, http://www.json.org/json-es.html

[47] Neurotechnology, *“VeriLook SDK”*, fecha de acceso: 23/07/2012, http://www.neurotechnology.com/verilook.html

[48] AT&T Laboratories Cambridge, *“The Database of Faces”*, fecha de acceso: 06/08/2012, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[49] Poisonivy-rat, *“Poison Ivy - Remote Administration Tool”*, fecha de acceso: 02/08/2012, http://www.poisonivy-rat.com/

[50] Manycam, *“ManyCam free virtual Webcam effects software”*, fecha de acceso: 03/08/2012, http://manycam.com/

[51] Wireshark Foundation, *"Wireshark"*, fecha de acceso: 27/08/2012, http://www.wireshark.org/

ANEXOS

Anexo A Resultados Parciales de Prueba Leave one out para el método Eigenfaces

Id de Prueba	Id Usuario	Ruta de Imagen de prueba	Taza de acierto	Id de usuario de coincidencia	Tiempo de entrenamiento de la base de datos en segundos	Tiempo en reconocimiento por muestra en segundos
1	1	./base_datos_rostros/1/1.jpg	0,596528	1	1,28892	0,472295
2	1	./base_datos_rostros/1/10.jpg	0,529547	1	1,28267	0,454574
3	1	./base_datos_rostros/1/2.jpg	0,644974	1	1,30917	0,455464
4	1	./base_datos_rostros/1/3.jpg	0,625058	1	1,31124	0,470716
5	1	./base_datos_rostros/1/4.jpg	0,72832	1	1,29532	0,44814
6	1	./base_datos_rostros/1/5.jpg	0,921611	1	1,29119	0,466713
7	1	./base_datos_rostros/1/6.jpg	0,970148	1	1,62405	0,449253
8	1	./base_datos_rostros/1/7.jpg	0,768588	1	1,29951	0,450006
9	1	./base_datos_rostros/1/8.jpg	0,669757	1	1,28552	0,451948
10	1	./base_datos_rostros/1/9.jpg	0,376086	1	1,56714	0,455065
11	2	./base_datos_rostros/2/1.jpg	0,671159	2	1,3044	0,451725
12	2	./base_datos_rostros/2/10.jpg	0,57593	2	1,30404	0,45184
13	2	./base_datos_rostros/2/2.jpg	0,653807	2	1,65942	0,458297
14	2	./base_datos_rostros/2/3.jpg	0,898112	2	1,38808	0,444964
15	2	./base_datos_rostros/2/4.jpg	0,484338	2	1,4188	0,456641
16	2	./base_datos_rostros/2/5.jpg	0,484338	5	1,32124	0,450658
17	2	./base_datos_rostros/2/6.jpg	0,703259	2	1,30214	0,465593
18	2	./base_datos_rostros/2/7.jpg	0,848872	2	1,29992	0,455933
19	2	./base_datos_rostros/2/8.jpg	0,660909	2	1,30057	0,461216
20	2	./base_datos_rostros/2/9.jpg	0,647303	2	1,30069	0,449696
21	3	./base_datos_rostros/3/1.jpg	1	3	1,30685	0,459392
22	3	./base_datos_rostros/3/10.jpg	0,973827	3	1,2911	0,444213
23	3	./base_datos_rostros/3/2.jpg	1	3	1,29556	0,455075
24	3	./base_datos_rostros/3/3.jpg	1	3	1,29788	0,442803
25	3	./base_datos_rostros/3/4.jpg	1	3	1,29324	0,454389
26	3	./base_datos_rostros/3/5.jpg	1	3	1,32735	0,446892
27	3	./base_datos_rostros/3/6.jpg	1	3	1,30277	0,466429
28	3	./base_datos_rostros/3/7.jpg	1	3	1,29429	0,442298
29	3	./base_datos_rostros/3/8.jpg	1	3	1,28487	0,451182
30	3	./base_datos_rostros/3/9.jpg	1	3	1,68801	0,482839

31	4	./base_datos_rostros/4/1.jpg	1	4	1,32652	0,451384
32	4	./base_datos_rostros/4/10.jpg	1	4	1,29338	0,467099
33	4	./base_datos_rostros/4/2.jpg	1	4	1,31095	0,442622
34	4	./base_datos_rostros/4/3.jpg	0,839007	4	1,29114	0,452877
35	4	./base_datos_rostros/4/4.jpg	1	4	1,29115	0,462041
36	4	./base_datos_rostros/4/5.jpg	1	4	1,29571	0,456137
37	4	./base_datos_rostros/4/6.jpg	1	4	1,29558	0,442284
38	4	./base_datos_rostros/4/7.jpg	1	4	1,29563	0,476513
39	4	./base_datos_rostros/4/8.jpg	1	4	1,29637	0,443842
40	4	./base_datos_rostros/4/9.jpg	1	4	1,2848	0,452542
41	5	./base_datos_rostros/5/1.jpg	0,658856	5	1,30551	0,447961
42	5	./base_datos_rostros/5/10.jpg	0,7204	5	1,3089	0,473451
43	5	./base_datos_rostros/5/2.jpg	0,7204	3	1,29128	0,442178
44	5	./base_datos_rostros/5/3.jpg	0,637244	5	1,2823	0,474438
45	5	./base_datos_rostros/5/4.jpg	0,558606	5	1,2966	0,446142
46	5	./base_datos_rostros/5/5.jpg	0,484216	5	1,30317	0,45439
47	5	./base_datos_rostros/5/6.jpg	0,589262	5	1,31023	0,446959
48	5	./base_datos_rostros/5/7.jpg	0,589262	2	1,28902	0,456516
49	5	./base_datos_rostros/5/8.jpg	0,651793	5	1,30046	0,447621
50	5	./base_datos_rostros/5/9.jpg	0,715004	5	1,62752	0,458768
51	6	./base_datos_rostros/6/1.jpg	1	6	1,30849	0,446553
52	6	./base_datos_rostros/6/10.jpg	0,759215	6	1,29172	0,453942
53	6	./base_datos_rostros/6/2.jpg	1	6	1,69802	0,455635
54	6	./base_datos_rostros/6/3.jpg	0,392178	6	1,29734	0,44583
55	6	./base_datos_rostros/6/4.jpg	0,554575	6	1,2898	0,47449
56	6	./base_datos_rostros/6/5.jpg	1	6	1,33207	0,446337
57	6	./base_datos_rostros/6/6.jpg	1	6	1,29045	0,456105
58	6	./base_datos_rostros/6/7.jpg	0,821663	6	1,2983	0,451743
59	6	./base_datos_rostros/6/8.jpg	0,825025	6	1,34282	0,45082
60	6	./base_datos_rostros/6/9.jpg	0,439098	6	1,28775	0,440669
61	7	./base_datos_rostros/7/1.jpg	0,663601	7	1,27756	0,453503
62	7	./base_datos_rostros/7/10.jpg	0,675856	7	1,30834	0,444207
63	7	./base_datos_rostros/7/2.jpg	0,562515	7	1,28359	0,455871
64	7	./base_datos_rostros/7/3.jpg	0,951767	7	1,29979	0,441381
65	7	./base_datos_rostros/7/4.jpg	0,914518	7	1,30159	0,463808
66	7	./base_datos_rostros/7/5.jpg	0,665716	7	1,29444	0,444384
67	7	./base_datos_rostros/7/6.jpg	0,735296	7	1,28622	0,481692
68	7	./base_datos_rostros/7/7.jpg	0,637129	7	1,29835	0,451087
69	7	./base_datos_rostros/7/8.jpg	0,612704	7	1,28515	0,460782

70	7	./base_datos_rostros/7/9.jpg	0,710313	7	1,29013	0,466325
71	8	./base_datos_rostros/8/1.jpg	0,438436	8	1,28799	0,450892
72	8	./base_datos_rostros/8/10.jpg	0,271387	8	1,29237	0,447274
73	8	./base_datos_rostros/8/2.jpg	0,529611	8	1,28421	0,453307
74	8	./base_datos_rostros/8/3.jpg	0,635832	8	1,31933	0,444575
75	8	./base_datos_rostros/8/4.jpg	0,91943	8	1,28172	0,455116
76	8	./base_datos_rostros/8/5.jpg	0,436009	8	1,29107	0,464501
77	8	./base_datos_rostros/8/6.jpg	0,417766	8	1,28196	0,458555
78	8	./base_datos_rostros/8/7.jpg	0,438103	8	1,29068	0,441789
79	8	./base_datos_rostros/8/8.jpg	0,512817	8	1,29746	0,470617
80	8	./base_datos_rostros/8/9.jpg	0,360827	8	1,28946	0,441618

Anexo B Resultados Parciales de Prueba Leave one out para el método Fisherfaces

Id de Prueba	Id Usuario	Ruta de Imagen de prueba	Taza de acierto	Id de usuario de coincidencia	Tiempo de entrenamiento de la base de datos en segundos	Tiempo en reconocimiento por muestra en segundos
1	1	./base_datos_rostros/1/1.jpg	0,365157	1	0,40501	0,046919
2	1	./base_datos_rostros/1/10.jpg	0,17171	1	0,411459	0,071254
3	1	./base_datos_rostros/1/2.jpg	0,946509	1	0,430197	0,046366
4	1	./base_datos_rostros/1/3.jpg	0,200638	1	0,410455	0,047504
5	1	./base_datos_rostros/1/4.jpg	0,581254	1	0,429169	0,046933
6	1	./base_datos_rostros/1/5.jpg	1	1	0,410835	0,065252
7	1	./base_datos_rostros/1/6.jpg	1	1	0,427566	0,046762
8	1	./base_datos_rostros/1/7.jpg	0,565455	1	0,410853	0,047362
9	1	./base_datos_rostros/1/8.jpg	0,251432	1	0,430119	0,046469
10	1	./base_datos_rostros/1/9.jpg	0,186458	1	0,419977	0,065007
11	2	./base_datos_rostros/2/1.jpg	0,177812	2	0,426905	0,047354
12	2	./base_datos_rostros/2/10.jpg	0,269079	2	0,411471	0,04712
13	2	./base_datos_rostros/2/2.jpg	0,629124	2	0,430217	0,046347
14	2	./base_datos_rostros/2/3.jpg	0,311956	2	0,409227	0,059936
15	2	./base_datos_rostros/2/4.jpg	0,106713	2	0,414395	0,047052
16	2	./base_datos_rostros/2/5.jpg	0,106713	5	0,409651	0,045978
17	2	./base_datos_rostros/2/6.jpg	0,248956	2	0,432161	0,046408
18	2	./base_datos_rostros/2/7.jpg	0,344968	2	0,410956	0,052587
19	2	./base_datos_rostros/2/8.jpg	0,515812	2	0,419267	0,047005
20	2	./base_datos_rostros/2/9.jpg	0,475919	2	0,414433	0,051434
21	3	./base_datos_rostros/3/1.jpg	1	3	0,421313	0,046545
22	3	./base_datos_rostros/3/10.jpg	0,460214	3	0,409827	0,046634
23	3	./base_datos_rostros/3/2.jpg	1	3	0,419295	0,047596
24	3	./base_datos_rostros/3/3.jpg	0,976127	3	0,415707	0,059784
25	3	./base_datos_rostros/3/4.jpg	0,423626	3	0,428937	0,046329
26	3	./base_datos_rostros/3/5.jpg	0,44465	3	0,410409	0,04703
27	3	./base_datos_rostros/3/6.jpg	0,604764	3	0,430159	0,046964
28	3	./base_datos_rostros/3/7.jpg	0,293098	3	0,412026	0,046204
29	3	./base_datos_rostros/3/8.jpg	1	3	0,441864	0,047822

30	3	./base_datos_rostros/3/9.jpg	0,309494	3	0,422276	0,065095
31	4	./base_datos_rostros/4/1.jpg	0,625219	4	0,428119	0,047082
32	4	./base_datos_rostros/4/10.jpg	1	4	0,411808	0,046857
33	4	./base_datos_rostros/4/2.jpg	0,870142	4	0,432428	0,047242
34	4	./base_datos_rostros/4/3.jpg	0,407059	4	0,412548	0,067738
35	4	./base_datos_rostros/4/4.jpg	0,772962	4	0,429903	0,047115
36	4	./base_datos_rostros/4/5.jpg	0,824995	4	0,411893	0,047548
37	4	./base_datos_rostros/4/6.jpg	1	4	0,429628	0,047283
38	4	./base_datos_rostros/4/7.jpg	0,344966	4	0,409828	0,060126
39	4	./base_datos_rostros/4/8.jpg	1	4	0,419292	0,046719
40	4	./base_datos_rostros/4/9.jpg	1	4	0,426137	0,060697
41	5	./base_datos_rostros/5/1.jpg	0,60499	5	0,428138	0,04706
42	5	./base_datos_rostros/5/10.jpg	0,182079	5	0,4117	0,047333
43	5	./base_datos_rostros/5/2.jpg	0,182079	3	0,42104	0,047139
44	5	./base_datos_rostros/5/3.jpg	0,832266	5	0,407034	0,047318
45	5	./base_datos_rostros/5/4.jpg	0,244792	5	0,416616	0,047661
46	5	./base_datos_rostros/5/5.jpg	0,153996	5	0,418793	0,062229
47	5	./base_datos_rostros/5/6.jpg	0,151211	5	0,427046	0,047492
48	5	./base_datos_rostros/5/7.jpg	0,151211	2	0,41298	0,048145
49	5	./base_datos_rostros/5/8.jpg	0,254547	5	0,41962	0,04786
50	5	./base_datos_rostros/5/9.jpg	0,271841	5	0,419389	0,065061
51	6	./base_datos_rostros/6/1.jpg	0,321595	6	0,42658	0,048153
52	6	./base_datos_rostros/6/10.jpg	0,255047	6	0,413847	0,046932
53	6	./base_datos_rostros/6/2.jpg	0,540567	6	0,415478	0,046492
54	6	./base_datos_rostros/6/3.jpg	0,134453	6	0,415061	0,060021
55	6	./base_datos_rostros/6/4.jpg	0,194613	6	0,423599	0,047754
56	6	./base_datos_rostros/6/5.jpg	0,355364	6	0,41217	0,046894
57	6	./base_datos_rostros/6/6.jpg	0,36896	6	0,419939	0,048601
58	6	./base_datos_rostros/6/7.jpg	0,121784	6	0,413321	0,055481
59	6	./base_datos_rostros/6/8.jpg	0,578172	6	0,42255	0,047027
60	6	./base_datos_rostros/6/9.jpg	0,166262	6	0,419059	0,045853
61	7	./base_datos_rostros/7/1.jpg	0,224788	7	0,41938	0,046639
62	7	./base_datos_rostros/7/10.jpg	0,353861	7	0,412869	0,046324
63	7	./base_datos_rostros/7/2.jpg	0,865786	7	0,429678	0,046878

64	7	./base_datos_rostros/7/3.jpg	1	7	0,412321	0,059421
65	7	./base_datos_rostros/7/4.jpg	0,615215	7	0,425476	0,046776
66	7	./base_datos_rostros/7/5.jpg	0,43517	7	0,412222	0,046614
67	7	./base_datos_rostros/7/6.jpg	1	7	0,440555	0,047571
68	7	./base_datos_rostros/7/7.jpg	0,404342	7	0,413347	0,051074
69	7	./base_datos_rostros/7/8.jpg	0,288835	7	0,417408	0,046985
70	7	./base_datos_rostros/7/9.jpg	0,344028	7	0,418904	0,054073
71	8	./base_datos_rostros/8/1.jpg	0,166203	8	0,421813	0,046892
72	8	./base_datos_rostros/8/10.jpg	0,166203	6	0,410907	0,046709
73	8	./base_datos_rostros/8/2.jpg	0,203348	8	0,423428	0,047693
74	8	./base_datos_rostros/8/3.jpg	0,47524	8	0,412564	0,047992
75	8	./base_datos_rostros/8/4.jpg	0,290102	8	0,449299	0,047044
76	8	./base_datos_rostros/8/5.jpg	0,195024	8	0,413859	0,047142
77	8	./base_datos_rostros/8/6.jpg	0,153618	8	0,436857	0,050918
78	8	./base_datos_rostros/8/7.jpg	0,262187	8	0,41419	0,04596
79	8	./base_datos_rostros/8/8.jpg	0,437722	8	0,447272	0,046212
80	8	./base_datos_rostros/8/9.jpg	0,135148	8	0,416442	0,046652