Universidad Carlos III de Madrid

PROYECTO FIN DE CARRERA

Autenticación biométrica en redes sociales: Diseño e implementación de reconocimiento facial mediante EmguCV para autenticación en redes sociales.

Ingeniería Técnica en Informática de Gestión

Autor: Javier Vázquez Míguez

Tutor: Lorena González Manzano

Leganés, 3 noviembre de 2014

Título: Diseño e implementación de reconocimiento facial mediante EmguCV para autenticación en redes sociales.

Autor: Javier Vázquez Míguez

Director: Lorena González Manzano

EL TRIBUNAL

Presidente: José María de Fuentes

Vocal: Irene Pérez

Secretario: Sergio Pastrana

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 3 de noviembre de 2014 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Este trabajo no habría sido posible sin la ayuda de mi mujer Noemí, que me ha apoyado constantemente, así como toda mi familia.

También me gustaría dar las gracias a Lorena, por toda su ayuda y su paciencia.

Resumen

En este proyecto se implementa y se analiza un sistema de autenticación en redes sociales mediante reconocimiento facial con la librería Emgu CV.

En la implementación del proyecto se han seguido dos líneas principales. Primero, debido a que los sistemas de reconocimiento facial necesitan de un entrenamiento para reconocer a los usuarios, ha comenzado con la realización de una página web en la cual el usuario pueda entrenar el sistema con imágenes de su rostro tomadas con la cámara web y administrar sus imágenes guardadas en la base de datos.

Posteriormente se ha realizado una extensión para el navegador Google Chrome la cual hace posible la autenticación en una de las redes sociales dentro del ámbito de este proyecto (Facebook y Twitter).

Para la realización del análisis de funcionamiento se han realizado pruebas con los distintos algoritmos implementados en EmguCV sobre 4 personas variando ciertos valores de entrada, evaluando los aciertos y los errores así como el tiempo medio de procesamiento.

Palabras clave:

Redes sociales, autenticación, reconocimiento facial, Emgu CV

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

1.1 Introducción

El auge de las redes sociales, en las cuales para poder hacer uso de ellas se necesita tener una cuenta de usuario y en las que normalmente se almacena información personal, ha suscitado la necesidad de utilizar multitud de usuarios y contraseñas.

Este problema se puede solucionar si se utiliza el gestor de contraseñas implementado en nuestros navegadores; pero éstos pueden ser accesibles para cualquier persona, por lo que la información almacenada en ellos no es totalmente segura.

En este proyecto se plantea una solución al problema de "usuario y contraseña". Se desarrolla un sistema de reconocimiento facial que permite iniciar sesión en las redes sociales de Facebook y Twitter. Sin embargo, aunque el ámbito de este proyecto está pensado para su funcionamiento en dichas redes, el sistema podría implementarse en cualquier otro tipo de aplicaciones con unas pequeñas modificaciones.

Las personas utilizamos diariamente el reconocimiento facial en nuestras vidas, siendo algo rutinario y casi inmediato: cuando vemos a alguien por la calle, cuando estamos viendo la televisión, etc. Siempre que vemos a alguien que conocemos lo reconocemos inmediatamente. Sin embargo, el reconocimiento facial, en este proyecto, ha de ser realizado por una máquina.

Los avances en las capacidades de computación han hecho posible la implementación de algoritmos capaces de realizar reconocimiento facial con un alto grado de acierto, siendo estos los que se tomarán como base para el desarrollo de este proyecto.

Por tanto, dada la necesidad de crear un sistema que simplifique el acceso a las distintas aplicaciones (a las redes sociales en este caso) y dada la posibilidad de realizar reconocimiento facial haciendo uso de la cámara web de un ordenador, este proyecto plantea el desarrollo de un sistema de autenticación basado en el reconocimiento facial. Además, todo está desarrollado considerando la seguridad de los credenciales de los usuarios, es decir, de sus usuarios y contraseñas.

1.2 Motivación

Actualmente existen una gran cantidad de páginas web y aplicaciones online que requieren el registro de sus usuarios para el acceso o uso de todos sus servicios. Debido a esto, un único usuario puede tener una gran cantidad de cuentas con un usuario y una contraseña distintos.

Por lo anterior y debido al gran auge en el uso de las redes sociales se ha decidido diseñar e implementar un sistema que permita el acceso a las mismas mediante un sistema de reconocimiento facial tratando de evitar que los credenciales estén almacenados en el navegador.

Aunque el ámbito de este proyecto consistirá en las redes sociales de Facebook y Twitter, se podrá implementar en cualquier página web.

1.3 Objetivos

El principal objetivo que se persigue en el desarrollo de este proyecto es el diseño de un sistema de reconocimiento facial que permita la autenticación en las redes sociales de Facebook y de Twitter, utilizando para ello la librería de código abierto Emgu CV. En concreto:

- El sistema deberá ser capaz de reconocer al usuario entre varias personas tras ser entrenado. Dicho entrenamiento lo debe realizar el usuario sacándose más de diez fotografías en las que el sistema lo detecte.

- El usuario podrá crear un fichero de credenciales encriptado.

- El usuario podrá iniciar sesión con el fichero creado en el punto anterior en una de las redes sociales disponibles en el ámbito de este proyecto (Facebook y Twitter).

Se ha fijado como objetivo adicional un estudio comparativo de los resultados obtenidos en los reconocimientos faciales de varios usuarios utilizando los distintos algoritmos empleados en Emgu CV para, de este modo, analizar la viabilidad del sistema desarrollado.

El proyecto constará de dos zonas diferenciadas:

- Aplicación web: el usuario podrá sacarse varias fotografías para que lo detecte el sistema (lo que consiste en entrenar la aplicación), editar las que tenga guardadas y crear un archivo de credenciales codificado.

- Extensión de Google Chrome: se instalará en el navegador y activará la funcionalidad de inicio de sesión facial en la página web (en este caso Facebook y Twitter).

1.4 Organización del presente documento

Con el objetivo de facilitar la lectura del documento en esta sección presenta una breve descripción del contenido de cada una de las partes que forman este documento, compuesto por seis capítulos y dos anexos

Capítulo 1. Introducción y objetivos:

En este capítulo se ofrece una introducción al proyecto, junto con las motivaciones y objetivos que se persiguen.

Capítulo 2. Estudio del arte:

En este capítulo se realiza un análisis de las herramientas disponibles para el desarrollo de aplicaciones de reconocimiento facial.

Capítulo 3. Análisis:

En este capítulo se expone una perspectiva general del sistema junto con el estudio de la arquitectura que tiene, definiéndose los principales componentes que la integran. El capítulo también incluye un estudio de las tecnologías a utilizar en el desarrollo del proyecto junto con la especificación de los casos de uso, los requisitos de software y las pruebas de aceptación definidas para comprobar el cumplimiento de estos requisitos.

Capítulo 4. Diseño:

En este capítulo se detallan los componentes definidos en el capítulo de análisis mostrando sus diagramas de clases. Adicionalmente se mostrarán un conjunto de diagramas de secuencia que permitan comprender las distintas interacciones que tendrán lugar entre los distintos componentes y clases del sistema.

Capítulo 5. Implementación:

En este capítulo se presentan los detalles más destacables de la implementación del sistema

Capítulo 6. Pruebas:

En este anexo se analizan las pruebas realizadas a cuatro usuarios, mostrando el porcentaje de acierto de la aplicación en los reconocimientos faciales así como una comparación entre los algoritmos que se incluyen en EmguCV.

Capítulo 7. Conclusiones y trabajo futuro:

Este último capítulo expone las conclusiones obtenidas con el desarrollo del proyecto así como alguna de las líneas de desarrollo futuras a seguir para extender la funcionalidad del sistema o mejorar su funcionamiento.

Anexo A. Gestión del proyecto

En este anexo se detalla la planificación del proyecto junto con el seguimiento del mismo. Adicionalmente se muestra el presupuesto del proyecto.

Anexo B. Manual de usuario

En este anexo se presenta un manual de usuario para mostrar el funcionamiento de la aplicación.

Anexo C. Plantillas

En este anexo se muestran las plantillas utilizadas para recoger distintos datos a lo largo de la memoria.

Capítulo 2

Estudio del arte

2.1 Programas de reconocimiento facial de escritorio

A continuación se muestra una visión general de los programas disponibles para Windows para realizar reconocimiento facial, así como sus funcionalidades y características. La funcionalidad principal de estos programas reside en la posibilidad de sustituir la contraseña de inicio de sesión en Windows por un sistema de reconocimiento facial.

KeyLemon Control Center

KeyLemon, una empresa especializada en reconocimiento facial, es la desarrolladora de KeyLemon Control Center, un software de reconocimiento facial que permite sustituir a la contraseña en el inicio y en el desbloqueo de la sesión en un PC Windows y en Mac OSX.

Sus principales características son las siguientes:

- Permite el acceso al ordenador desbloqueando la sesión o iniciándola mediante reconocimiento facial. Incluso permite iniciar sesión en un ordenador conectado a un dominio.
- Guarda información sobre las personas que han intentado acceder al ordenador mediante el reconocimiento facial. Se puede saber quién ha estado delante del ordenador y quien ha escrito una contraseña incorrecta de inicio de sesión mientras el ordenador estaba bloqueado.
- Puede bloquear el ordenador si no nota presencia mediante la cámara web o si quien está frente al ordenador no es una persona autorizada.
- Implementa reconocimiento de imagen en movimiento mediante parpadeo, para evitar que una persona no autorizada pueda tener acceso utilizando una fotografía de una persona autorizada.
- Permite la creación de diferentes perfiles para cada usuario dependiendo de las condiciones de iluminación.
- También permite mantener un seguimiento de los cambios faciales del usuario a lo largo del tiempo.

Se presenta en tres versiones: BASIC, BRONZE y GOLD. Esta última implementa todas las funcionalidades listadas anteriormente y es compatible con Microsoft Windows y Mac OSX, además de estar disponible en 6 idiomas, entre ellos el español.

Más información sobre este programa en su página web:

https://www.keylemon.com

Fotobounce

Fotobounce es una aplicación que nos permite organizar nuestras imágenes, centrada en aquellas en las que aparezcan personas. Para ello, busca todas las imágenes en las carpetas que le indiquemos y, mediante detección facial, determina si una imagen es de una persona o no (se debe poder encontrar por lo menos un rostro en la imagen). Tras esto, permite etiquetar a personas en las imágenes encontradas y tener una organización más personalizada. Si se etiqueta a una persona en una foto, mediante reconocimiento facial trata de encontrar imágenes de esa misma persona para etiquetarla automáticamente.

También permite conectar la aplicación a nuestras cuentas de Facebook, Twitter, Google y Flickr. Esto nos mostrará una lista de nuestras amistades y podremos etiquetarlos en las imágenes de nuestro ordenador. Posteriormente, podremos compartirlas en estas redes, manteniendo todas las etiquetas que hayamos establecido.

Permite además el acceso mediante el teléfono móvil para poder visualizar las imágenes, conectando a una cuenta que se haya creado en fotobounce.

Se puede encontrar más información sobre este programa en la siguiente dirección web:

http://fotobounce.com

Luxand Blink!

Luxnad Blink! es una aplicación desarrollada por Luxand, Inc. usada para el inicio de sesión mediante reconocimiento facial en Windows Vista/7. Para ello, el programa guiará al usuario para un entrenamiento en el que se pueda captar su rostro con unas leves variaciones de posición, y guardar la contraseña asociada a su usuario para iniciar sesión automáticamente.

El programa se inicia automáticamente con Windows, activa la cámara y capta al usuario que esté delante del ordenador. Si el usuario es reconocido, se inicia sesión automáticamente. Además de esto, puede guardar un historial con todos los inicios de sesión realizados por el usuario, o con los intentos de inicio de sesión por usuarios que no pertenecen al sistema.

Para más información sobre este programa, se puede acceder a su página web en la siguiente dirección:

http://luxand.com/blink/

Rohos Face Logon

Rohos Face Logon es un programa desarrollado para poder iniciar sesión en Windows, o bien desbloquear la sesión, mediante reconocimiento facial evitando tener que introducir el usuario y la contraseña de manera manual. Para ello se encarga de realizar patrones

Tiene soporte multiusuario y registra los patrones útiles del rostro de los usuarios, eliminando aquellos con poca o nula información sobre el usuario (poca iluminación, mala posición del rostro…). En el proceso de inicio de sesión permite ocultar la ventana con el proceso de reconocimiento facial y guarda un registro con el último inicio de sesión que se haya realizado.

Es compatible con Windows XP/Vista y con cuentas de usuario locales o en red. Para obtener más información sobre este programa, se puede acceder a la siguiente dirección web:

http://www.rohos.com/products/rohos-face-logon/

Capítulo 3

Análisis

3.1 Perspectiva general del sistema

En esta sección se aborda el sistema que se desarrollará e implementará para cubrir el objetivo definido en el Capítulo 1 de este documento: una aplicación para iniciar sesión mediante reconocimiento facial en redes sociales.

Para conseguir este objetivo, se ha propuesto el siguiente sistema dividido en dos partes:

- Una aplicación web en la cual se incluye un sistema de entrenamiento para el reconocedor facial y con un sistema para la creación de un fichero cifrado con los credenciales del usuario.
  Se dispone de una zona de entrenamiento del sistema debido a que es totalmente necesario que el usuario se saque fotografías en las que el sistema detecte su cara, asociándola al usuario, lo cual permitirá posteriormente el reconocimiento.
  Para la autenticación en la red social se usará un fichero de credenciales cifrado, el cual es creado por el sistema cuando el usuario introduce los credenciales en la aplicación web.

- Una extensión del navegador Google Chrome que permita acceder a la cámara web del usuario y mediante la cual se realiza la autenticación en la red social. En esta extensión se debe adjuntar el fichero de credenciales creado en la aplicación web para poder rellenar con los datos de autenticación (usuario y contraseña) de la red social los campos establecidos para ello.

Considerando el análisis de las soluciones de reconocimiento facial realizado en el Capítulo 2 se ha decidido utilizar la librería Emgu CV para desarrollo en .Net, ya que además de tratarse de una herramienta de código abierto y libre distribución, cuenta con una amplia comunidad desarrolladora y está en constante actualización.

3.2 Análisis de herramientas existentes

A continuación se realiza una descripción de las herramientas disponibles para la implementación de un sistema de reconocimiento facial.

VeriLook SDK

Verilook es una tecnología de identificación facial diseñada por la empresa Neurotechnology. La tecnología asegura el rendimiento del sistema y la fiabilidad con la detección de rostros en vivo y el reconocimiento simultaneo de múltiples rostros.

Está disponible como un kit de desarrollo de software que permite el desarrollo de soluciones de escritorio y móviles (Microsoft Windows, Linux, Mac OS X y plataformas Android) o para Web.

No permite el acceso al código y tiene un coste de entre 339€ (la versión Standard) y 859€ (la versión Extended).

Toda la información disponible sobre este SDK se encuentra en la siguiente dirección web:

http://www.neurotechnology.com/verilook.html

Luxand FaceSDK

El FaceSDK de Luxand proporciona entornos de desarrollo para Microsoft Visual C++, C#, Objective C, VB.NET, VB6, Java, Delphi y C++Builder, en sistemas Windows/Linux 32/64 bit, Mac OS X 64-bit y para las plataformas móviles iOS y Android. Ofrece también compatibilidad con arquitecturas multi-core para aumentar el rendimiento, utilizando varios procesos simultáneos.

Además de reconocimiento facial, permite la detección de cara y de ojos en imágenes fijas y en video, así como de otros 66 rasgos faciales (contorno de ojos, cejas, la punta de la nariz,…). Afirman un buen funcionamiento con condiciones de luz variables, bajo la luz del día, luz fluorescente o luz incandescente, afirmando tener una tasa de reconocimiento del 93% (sobre la base de datos Face Recognition Grand Challenge - FRGC -) cuando la tasa de falsa aceptación se establece en 0.1%.

Se puede consultar más información en la página web del desarrollador:

http://www.luxand.com/facesdk/index.php

FaceVACS-SDK

FaceVACS-SDK proporciona una API que permite desarrollar aplicaciones de reconocimiento facial, cubriendo todos los casos de uso necesarios para realizarlo y es compatible con múltiples algoritmos.

Proporciona además funcionalidad adicional para la búsqueda de rostros y ojos en las imágenes, seguimiento de rostros en secuencias de vídeo y análisis y control de características faciales.

El API es compatible con los entornos de desarrollo de C++, C, Java y .Net, además de proporcionar compatibilidad con entornos de desarrollo para móviles Android e iOS.

Se puede consultar más información en la siguiente dirección web:

http://www.cognitec.com/facevacs-sdk.html

3.3 Arquitectura del sistema

Como se verá posteriormente, se ha diseñado el sistema de manera que en la parte de entrenamiento únicamente se utiliza la detección facial, en la cual se extrae la cara del usuario y se asocia al mismo.

Por otra parte tenemos el reconocimiento facial, el cual es empleado tanto en la extensión de Google Chrome para el acceso a las redes sociales como en el acceso al sistema. Debido a que, previo al reconocimiento facial, se debe detectar la cara del usuario dentro de la imagen, se han separado los componentes que realizan estas acciones.

Además tenemos el componente Fichero, que genera el fichero con los credenciales cifrados mediante una clave única para cada usuario (que se asigna automáticamente al crear un nuevo usuario en la aplicación web). Esta clave se almacena en la base de datos en la tabla que contiene la información del usuario. Cuando un usuario intente autenticarse en una red social, el sistema detectará de qué usuario se trata para, posteriormente, obtener su clave, con la que descifrará el fichero. Si se produce un error en el reconocimiento del usuario, indicando que es otra persona, el fichero no se podrá descifrar debido a que la clave será distinta.

La Ilustración 1 muestra el diagrama de componentes de la arquitectura del sistema. En este diseño se observa que la detección y el reconocimiento se encuentran separados en dos componentes independientes para atender las funcionalidades del sistema.

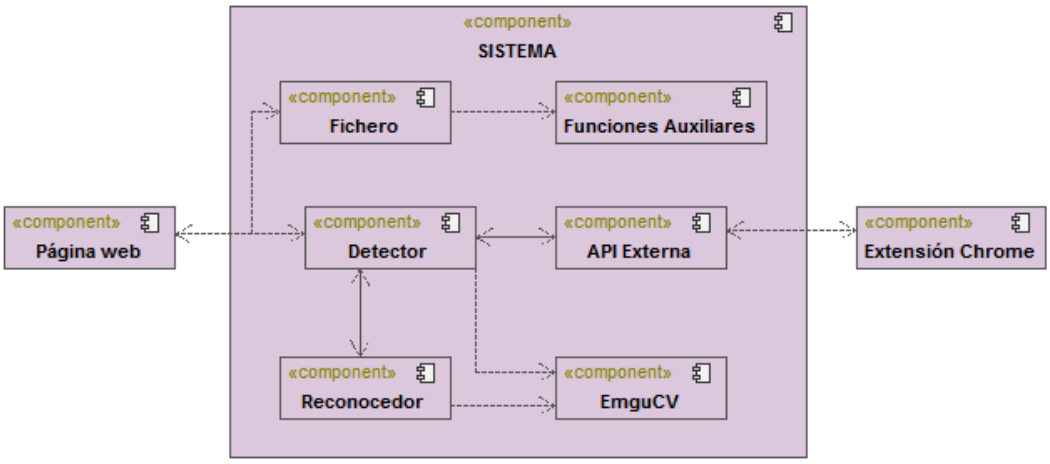

Figura 1. Arquitectura del sistema

La funcionalidad de los distintos componentes es la siguiente:

- *Detector*: Realizará la detección facial en las imágenes obtenidas del usuario mediante la cámara web.

- *Reconocedor*: Será el encargado de realizar el reconocimiento facial de la cara obtenida del Detector.

- *API externa*: Procesará la solicitud de reconocimiento facial de la extensión de Google Chrome.

- *Fichero*: Será el componente encargado de generar el fichero cifrado con los credenciales del usuario.

- *Funciones auxiliares*: Implementará las funciones auxiliares empleadas por los componentes anteriores.

3.4 Selección de tecnologías

El sistema que se ha detallado en los apartados anteriores será desarrollado en base a los siguientes componentes:

- .NET: El empleo de esta tecnología de Microsoft se debe a la mayor experiencia con estas herramientas. Se ha seleccionado por ello ASP.NET con C# siguiendo el patrón MVC para la realización de la página web de administración del usuario, donde podrá manejar sus imágenes y sus

credenciales. Además, se ha escogido Visual Studio 2012 como entorno de desarrollo por motivos de completitud del mismo.

- Emgu CV. Es un *wrapper* en .Net necesario para realizar llamadas a las funciones de Open CV utilizado en el procesamiento de las imágenes. Se ha seleccionado esta opción por su gran capacidad de integración con .Net, además de ser una herramienta de código abierto y tener licencia GPL *.

- SQL Server 2014. Última versión de la base de datos de Microsoft para maximizar la compatibilidad con .Net.

- Google Chrome. Éste navegador es uno de los más extendidos y con mayor crecimiento en los últimos años y por tanto, ha sido el escogido para la realización del sistema propuesto. Además permite el desarrollo de extensiones de una manera muy sencilla.

3.5 Casos de uso

En esta sección se muestra el diagrama de casos de uso de la aplicación junto con la definición textual de cada uno de los casos presentes en el diagrama. El estudio de los casos de uso descritos facilitará la extracción de requisitos para fases posteriores del proyecto.

3.5.1 Diagramas de casos de uso

La Ilustración 2 muestra el diagrama que representa los tres casos de uso identificados para el usuario. En dicho diagrama se pueden apreciar las tres funcionalidades principales ofrecidas por la aplicación: entrenamiento del sistema, creación de fichero de credenciales encriptado e inicio de sesión en la red social.

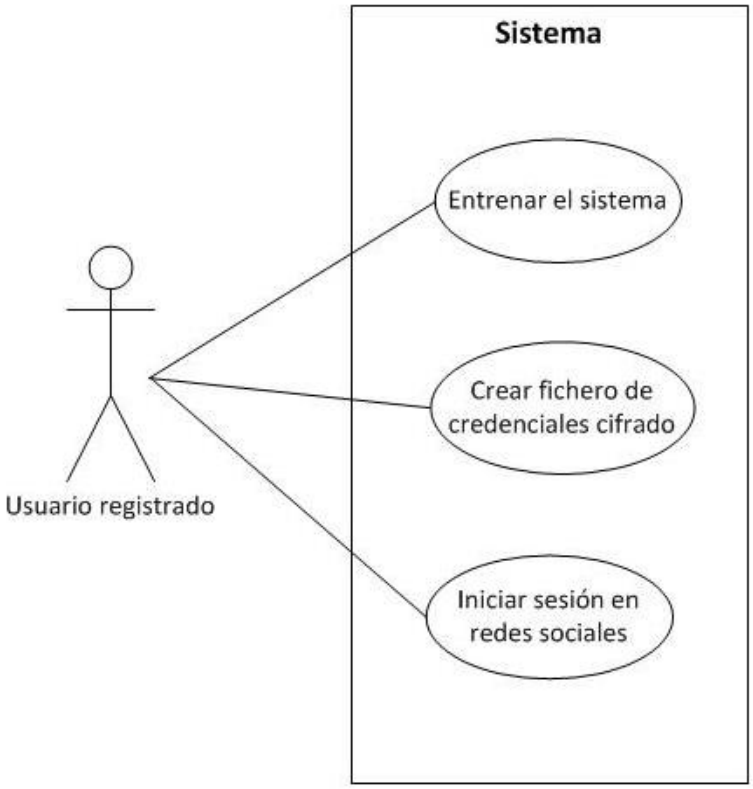

Figura 2. Diagrama de casos de uso

3.5.2 Definición textual de los casos de uso

En este apartado se muestra información detallada de los distintos casos de uso identificados en el diagrama de casos de uso del apartado anterior. Para mostrar la información se seguirá el formato definido en la plantilla 1 del Anexo C.

La Tabla 1 muestra el segundo caso de uso definido consistente en el entrenamiento del sistema para que sea capaz de reconocer al usuario. Para poder asegurar un buen funcionamiento del reconocimiento facial, como mínimo el usuario debería entrenar al sistema con diez fotografías suyas.

Identificador: CU-1	
Nombre	Entrenar el sistema
Autor	Javier Vázquez Míguez
Descripción El usuario puede entrenar al sistema sacándose fotografías en las que el sistema le reconozca la cara. Una vez reconocida la cara, se asociará con el usuario. A mayor cantidad de fotografías reconocidas por el sistema, mayor rango de aciertos podrá proporcionar el sistema, recomendándose un mínimo de 10.	
Actores Usuario de la aplicación.	

Precondiciones • Estar registrado en la aplicación. • Tener la sesión iniciada.
Post-condiciones • Almacenamiento de todas las imágenes en las que el usuario ha sido reconocido.
Flujo normal 1. Iniciar sesión en la aplicación. 2. Acceder a la página web de entrenamiento. 3. Sacarse fotografías en las que el sistema reconozca al usuario. 4. Guardar las imágenes. 5. Finalizar
Flujo alternativo 1.A. Si los datos introducidos no son correctos, se notifica al usuario

Tabla 1. Caso de uso Entrenar al sistema

La siguiente figura muestra el funcionamiento, de manera visual, de este caso de uso:

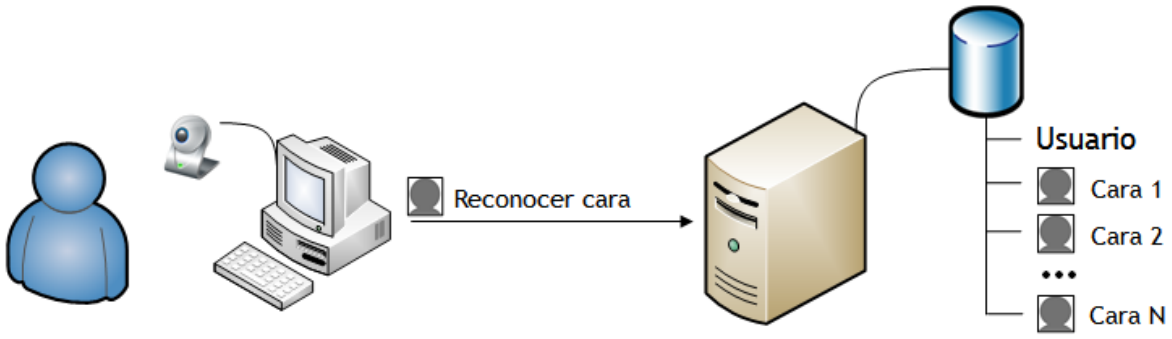

Figura 3. Entrenar el sistema

La Tabla 2 muestra el tercer caso de uso definido consistente en la generación del fichero de credenciales necesarios para acceder a la red social. El contenido de este fichero estará encriptado con una clave única para cada usuario que se genera cuando se registra en la aplicación.

Identificador: CU-2	
Nombre	Crear fichero de credenciales cifrado
Autor	Javier Vázquez Míguez
Descripción El usuario puede crear un fichero con los credenciales necesarios para iniciar sesión en una red social. Este fichero tendrá la información cifrada.	
Actores Usuario de la aplicación.	
Precondiciones • Tener la sesión iniciada.	

Post-condiciones • Fichero descargable con los credenciales encriptados.
Flujo normal 1. Acceso a la página web de creación del fichero. 2. Introducir los datos necesarios del usuario. 3. Descargar fichero encriptado. 4. Finalizar
Flujo alternativo 2.A. Si las contraseñas no son iguales, se notifica al usuario para que lo solucione.

Tabla 2. Caso de uso Crear fichero de credenciales encriptado

La siguiente figura muestra el funcionamiento, de manera visual, de este caso de uso:

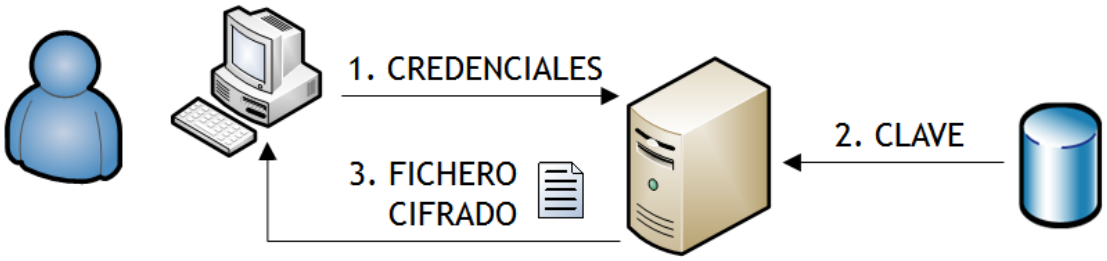

Figura 4. Crear fichero cifrado

Por último, la Tabla 3 muestra el cuarto caso de uso identificado consistente en el inicio de sesión en las redes sociales mediante la extensión de Google Chrome. Para ello, el usuario debe activar la extensión.

Identificador: CU-3	
Nombre	Iniciar sesión en Facebook/Twitter
Autor	Javier Vázquez Míguez
Descripción El usuario, a través de la extensión de Google Chrome, podrá iniciar sesión mediante reconocimiento facial directamente en las páginas de Facebook o Twitter.	
Actores Usuario de la aplicación.	
Precondiciones • Acceder a la página web de la red social deseada. • Tener instalada la extensión en Google Chrome. • Haber creado un fichero de credenciales cifrado en la aplicación web.	
Post-condiciones • Sesión iniciada en la red social elegida	
Flujo normal 1. Acceder a la página web de la red social. 2. Activar la extensión de Google Chrome.	

3. Activar la cámara web. 4. Sacarse una fotografía. 5. Seleccionar el fichero encriptado con los credenciales de la red social 6. Pulsar el botón iniciar sesión. a. Si el usuario no es reconocido, volver al punto 4. b. Si el fichero no es válido, volver al punto 5. 7. Iniciar sesión en la red social. 8. Finalizar
Flujo alternativo 1. Acceder a la página web de la red social. 2. Activar la extensión de Google Chrome. 3. Activar la cámara web. 4. Sacarse una fotografía. 5. Pulsar el botón “Iniciar sesión” de la extensión. a. Si el usuario no es reconocido, volver al punto 4. 6. Seleccionar el fichero encriptado con los credenciales de la red social 7. Pulsar el botón “Iniciar sesión” de la extensión. a. Si el fichero no es válido, volver al punto 6. 8. Iniciar sesión en la red social. 9. Finalizar

Tabla 3. Caso de uso Iniciar sesión en Facebook/Twitter

La siguiente figura muestra el funcionamiento, de manera visual, de este caso de uso:

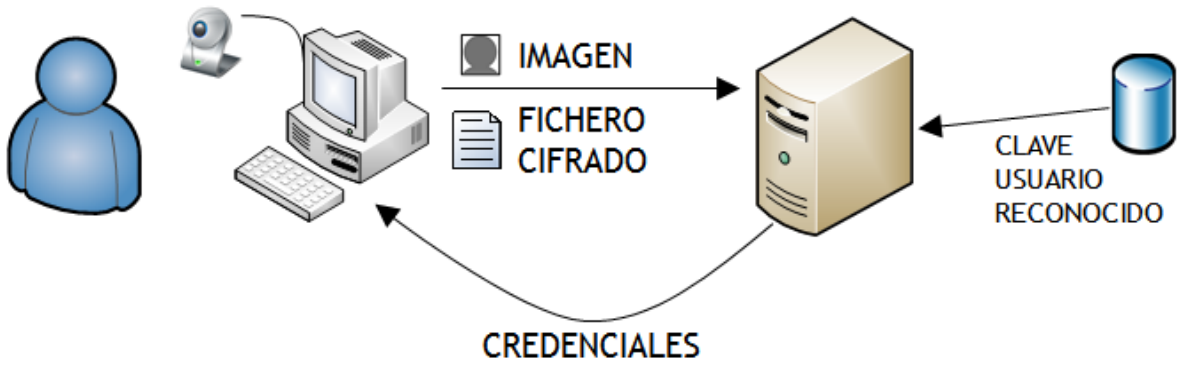

Figura 5. Iniciar sesión en la red social

3.6 Requisitos software

En esta sección se presentan los requisitos de software identificados que deberán ser cubiertos para cumplir con los objetivos especificados para este proyecto.

Las distintas subsecciones que componen esta sección detallan los requisitos de los distintos tipos definidos por la metodología de la ESA. Se muestran únicamente los tipos de requisitos utilizados en esta definición.

3.6.1 Requisitos funcionales

La Tabla 4 agrupa la información de los distintos requisitos funcionales de acuerdo al formato definido en la plantilla 2 del Anexo C.

Requisitos de software				
Tipo: Funcional				
Id	Nombre	Descripción	Estabilidad	Prioridad
RF-01	Registrar Usuario	Registrar al usuario en la base de datos	Media	Media
RF-02	Iniciar sesión	Iniciar sesión en la aplicación con usuario y contraseña	Media	Media
RF-03	Iniciar sesión por reconocimiento facial	Iniciar sesión mediante reconocimiento facial, tanto en la aplicación como en la red social	Alta	Alta
RF-04	Utilizar cámara web para entrenar el sistema	Entrenar al sistema sacándose fotografías y permitiendo que el sistema reconozca al usuario.	Alta	Alta
RF-05	Visualizar imágenes entrenadas	Visualizar todas la imágenes entrenadas en el sistema	Media	Media
RF-06	Eliminar imágenes	Eliminar una imagen entrenada del sistema	Media	Media
RF-07	Crear fichero cifrado	Crear fichero cifrado con los credenciales que introduzca el usuario	Alta	Alta
RF-08	Subir fichero cifrado	Leer fichero cifrado que adjunte el usuario al iniciar sesión	Alta	Alta
RF-09	Guardar fichero de credenciales en las opciones de la extensión	Para comodidad del usuario, se creará una página de opciones de la extensión que permita asignar a las redes sociales un archivo de credenciales determinado permanentemente.	Alta	Alta

Tabla 4. Requisitos funcionales

3.6.2 Requisitos no funcionales

La Tabla 5 agrupa la información de los distintos requisitos no funcionales de acuerdo al formato definido en la plantilla 2 del Anexo C. Como se ha indicado anteriormente la tabla no menciona los tipos de requisitos no utilizados.

Requisitos de software				
Tipo: Operacional				
Id	Nombre	Descripción	Estabilidad	Prioridad
RNF-01	Idioma de la aplicación	La aplicación muestra toda la información en español	Alta	Alta
RNF-02	Validar datos introducidos	La aplicación validará todos los datos introducidos por el usuario	Media	Media
RNF-03	Mensajes de error	Se mostrarán mensajes de error en caso de producirse un problema	Alta	Alta
RNF-04	Eliminar sesión tras 1 minuto	Si al iniciar sesión mediante reconocimiento facial no se adjunta el fichero de credenciales, el usuario tendrá un minuto para adjuntarlo	Media	Media
RNF-05	Formato del archivo cifrado	La aplicación solo aceptará archivos con extensión .txt	Alta	Alta

Tabla 5. Requisitos no funcionales

3.6.3 Requisitos inversos

La Tabla 6 agrupa la información de los distintos requisitos inversos de acuerdo al formato definido en la plantilla 2 del Anexo C.

Requisitos de software				
Tipo: Inverso				
Id	Nombre	Descripción	Estabilidad	Prioridad
RI-01	Únicamente se aceptarán imágenes de una cámara web	Todas las imágenes que se envíen al sistema serán captados por el mismo mediante una cámara web.	Alta	-

Tabla 6. Requisitos inversos

3.7 Plan de pruebas de aceptación

Una vez definidas las características de la aplicación que se desarrollará a lo largo de este proyecto se puede proceder a realizar el plan de pruebas de aceptación de la misma.

En esta sección se detalla el plan de pruebas de aceptación de la aplicación en donde se muestran las distintas pruebas que se han de superar y los resultados esperados de las mismas para que se superen las pruebas de aceptación.

Pruebas de aceptación			
Id	Elementos probados	Entrada	Salida
PA-01	RF-01	Registrar un nuevo usuario en la aplicación. Se introduce para ello un nombre y una contraseña.	Se crea correctamente un nuevo usuario en la base de datos con los datos introducidos
PA-02	RF-02	Iniciar sesión con los datos del usuario. Debe introducir el nombre de usuario y la contraseña.	El usuario accede a la aplicación.
PA-03	RF-03, RF-08	Iniciar sesión mediante reconocimiento facial, tanto en el sistema como en las distintas redes sociales disponibles. La cara del usuario debe ser reconocida y adjunta el fichero cifrado con los datos de inicio de sesión.	Si la cara es reconocida y el usuario adjunta el fichero de credenciales, accede al sistema, a Facebook y a Twitter.
PA-04	RF-04, RF-05	El sistema debe reconocer al usuario en la sección “Entrenar” de la página web y mostrar las imágenes reconocidas. Para ello se envía al sistema la imagen obtenida del usuario.	El usuario se saca una fotografía, el sistema lo reconoce y muestra la imagen.
PA-05	RF-06	El usuario puede eliminar las imágenes entrenadas. Se debe seleccionar la imagen a borrar.	El usuario selecciona la imagen a borrar, y esta es borrada de la base de datos.
PA-06	RF-07	Crear un fichero con los credenciales cifrados. El usuario introduce los datos de nombre y contraseña necesarios para acceder a una red social o al sistema.	El usuario introduce los datos y se descarga en su ordenador el fichero de credenciales cifrado.

PA-07	RF-09	Guardar permanentemente en la extensión los archivos que se usarán en la autenticación en una red social	El usuario selecciona los ficheros y se guardan correctamente.

Tabla 7. Pruebas de aceptación

3.8 Análisis EmguCV

En este apartado se detallan los tipos de reconocedores disponibles en EmguCV para realizar el reconocimiento facial.

Antes de realizar cualquier comparación para reconocer a la persona, se realiza una reducción de la dimensionalidad de la imagen para extraer la información más relevante y además para reducir el coste computacional del reconocimiento facial. Dicha dimensionalidad depende de la imagen en cuestión, es decir, si la imagen tiene *m* pixeles de ancho y *n* pixeles de alto, la dimensionalidad de la imagen será de *m* x *n*.

La reducción de la dimensionalidad es posible debido a que cuando se trata de caracterizar rostros, todos comparten ciertas características globales, como por ejemplo la posición de los ojos, de la nariz y de la boca, pudiendo extraer la características más importantes y descartando aquella información que sea innecesaria.

Para conseguir esta reducción, se utilizan diversas técnicas en EmguCV, que con las siguientes:

- Análisis de Componentes Principales (PCA de sus siglas en inglés)
- Análisis de Discriminantes Lineales (LDA de sus siglas en inglés)
- Patrones Binarios Locales (LBP de sus siglas en inglés)

3.8.1 Algoritmos implementados en Emgu CV

A continuación se hace una breve descripción de los diferentes algoritmos empleados por EmguCV.

Eigenfaces

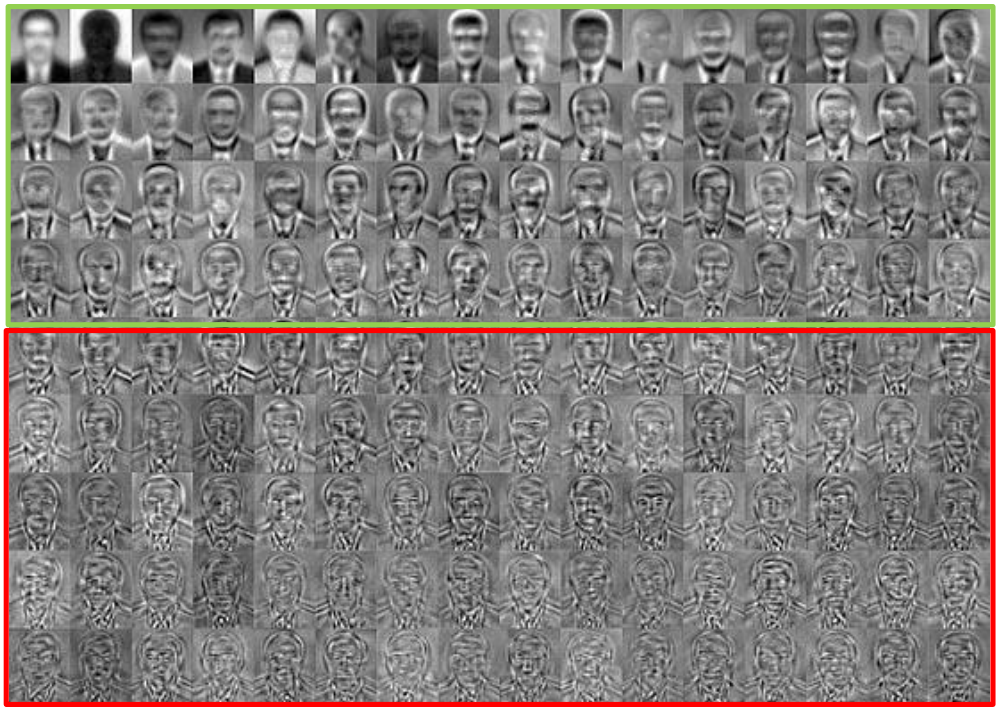

Figura 6. Ejemplo de eigenfaces.

Los eigenfaces son vectores de datos, también denominados eigenvectors, en el que cada uno posee valores escalares conocidos como eigenvalues. Cada eigenface representa un rasgo característico de las imágenes faciales siguiendo un patrón obtenido del estudio de muchos rostros. El funcionamiento de este método es el siguiente:

Por cada imagen del set de entrenamiento (M imágenes) se concatenan sus filas de pixeles, transformándola en un vector en el que cada valor se corresponde con el valor del pixel correspondiente.

Tras esto, obtenemos una matriz de vectores, siendo cada vector una imagen. A este vector se le aplica el método PCA para obtener los componentes principales y reducir la dimensionalidad. Se pasa a tener una matriz de k vectores (siendo $k < M$), donde cada vector se corresponde con un eigenface y se calcula el eigenface promedio.

A partir de estos eigenfaces, cada imagen del set de entrenamiento se puede reconstruir de una forma muy aproximada combinándolos en mayor o menor medida. Por ejemplo, uno de los rostros del set de entrenamiento se podría obtener mediante la combinación del eigenface promedio más el 75% del eigenface 1, menos el 37% del eigenface 2, más el 17% del eigenface 3,…. Con ello, se obtiene un vector con estos valores por cada imagen, haciendo lo mismo con la imagen de entrada. Se calcula la distancia entre el vector de valores de la imagen de entrada con el resto de las imágenes

del set de entrenamiento, y se selecciona la que produce la distancia mínima. Si dicha distancia se encuentra dentro de un umbral establecido, se selecciona como valor aceptado. Si no está dentro del umbral, se considerará un rostro desconocido.

Este método es muy aceptado debido a su fácil implementación y a que bajo unas buenas condiciones, tiene un buen porcentaje de aciertos. Por el contrario, un inconveniente de este enfoque es que no solo se maximiza la dispersión entre clases (personas), siendo información útil para la clasificación, sino que también se maximiza la dispersión intra-clases, producido en mayor parte por las condiciones de iluminación y por la expresión facial, lo cual dificulta la clasificación.

Fisherfaces

El funcionamiento de este método es análogo al de Eigenfaces, diferenciándose en que primero se aplica PCA para la reducción de la dimensionalidad y posteriormente se aplica LDA.

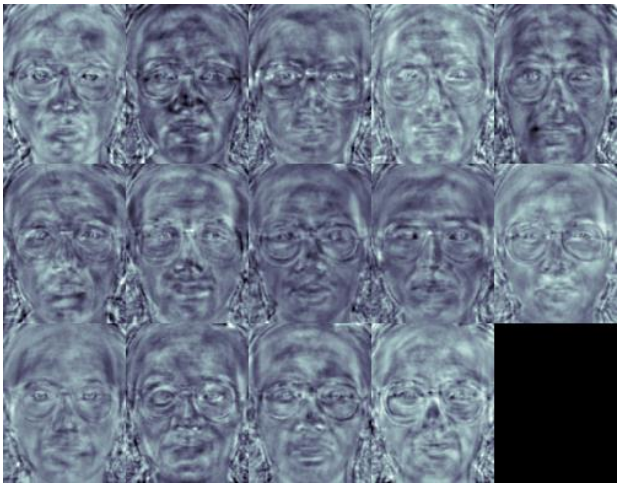

Figura 7. Ejemplo de Fisherfaces

Para ello, y ayudándose de que las imágenes están etiquetadas con la persona a la que pertenece, permite reducir la problemática de la dispersión intra-clases. La idea es simple: las mismas clases deben agruparse muy juntas, mientras que diferentes clases deben estar lo más alejadas posibles la una de la otra para mejorar la clasificación.

LBPH

Local Binary Pattern Histogram tiene un enfoque distinto a los dos anteriores algoritmos que hemos visto, Eigenfaces y Fisherfaces. En estos dos últimos se trata la

imagen como un vector de datos en un espacio de muy alta dimensionalidad. Pero en LBPH se pretenden describir las características locales de cada rostro. De esta manera, las características extraídas tendrán una baja dimensionalidad.

El funcionamiento es el siguiente. Se basa en ir tomando pixeles "vecinos" respecto de un pixel central, el cual establece un valor de umbral. Si el valor del pixel central es menor, se etiqueta con un uno, y si es mayor se etiqueta con un cero. Estos valores obtenidos se concatenan para formar un número binario, que posteriormente se convierte en decimal, el cual será el nuevo valor del pixel. Por cada región se obtiene un histograma, que posteriormente son concatenados para obtener una representación del rostro (Figura 8).

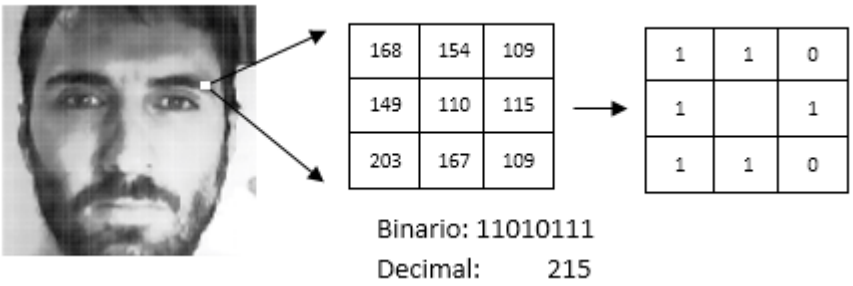

Figura 8. Funcionamiento LBPH

Se obtiene un histograma por cada una de las partes extraídas, y posteriormente se combinan:

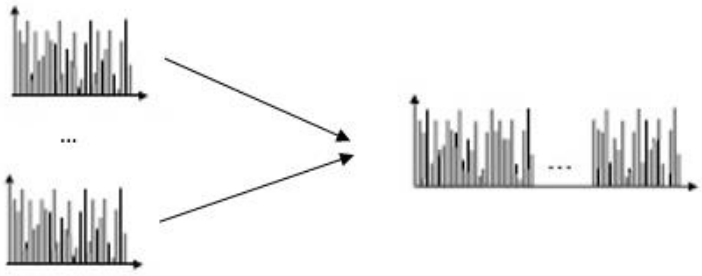

Figura 9. Histogramas por región se combinan en un histograma

Tras esto, se realiza la comparación con la imagen de entrada realizando las mismas operaciones.

Figura 10. Imágenes obtenidas al aplicar LBPH

Capítulo 4

Diseño

4.1 Diseño de software

En esta sección se mostrará una descripción detallada de los distintos componentes identificados en el capítulo de análisis. En el caso de que durante esta fase se detecte la necesidad de descomponer un componente en varios subcomponentes se detallarán, por claridad, en la misma subsección. Asimismo, se han obviado los métodos y variables que carecen de interés para la implementación, como por ejemplo los métodos de obtención y asignación de variables.

4.2 Diagrama de clases

En este apartado se muestra el modelo conceptual de la aplicación, las divisiones de los distintos módulos que se han decidido diseñar para que la implementación sea más sencilla. Para ello se ha realizado un Diagrama de Clases en UML, del cual se muestra el diagrama y un breve comentario sobre cada una de las clases junto con las relaciones entre ellas.

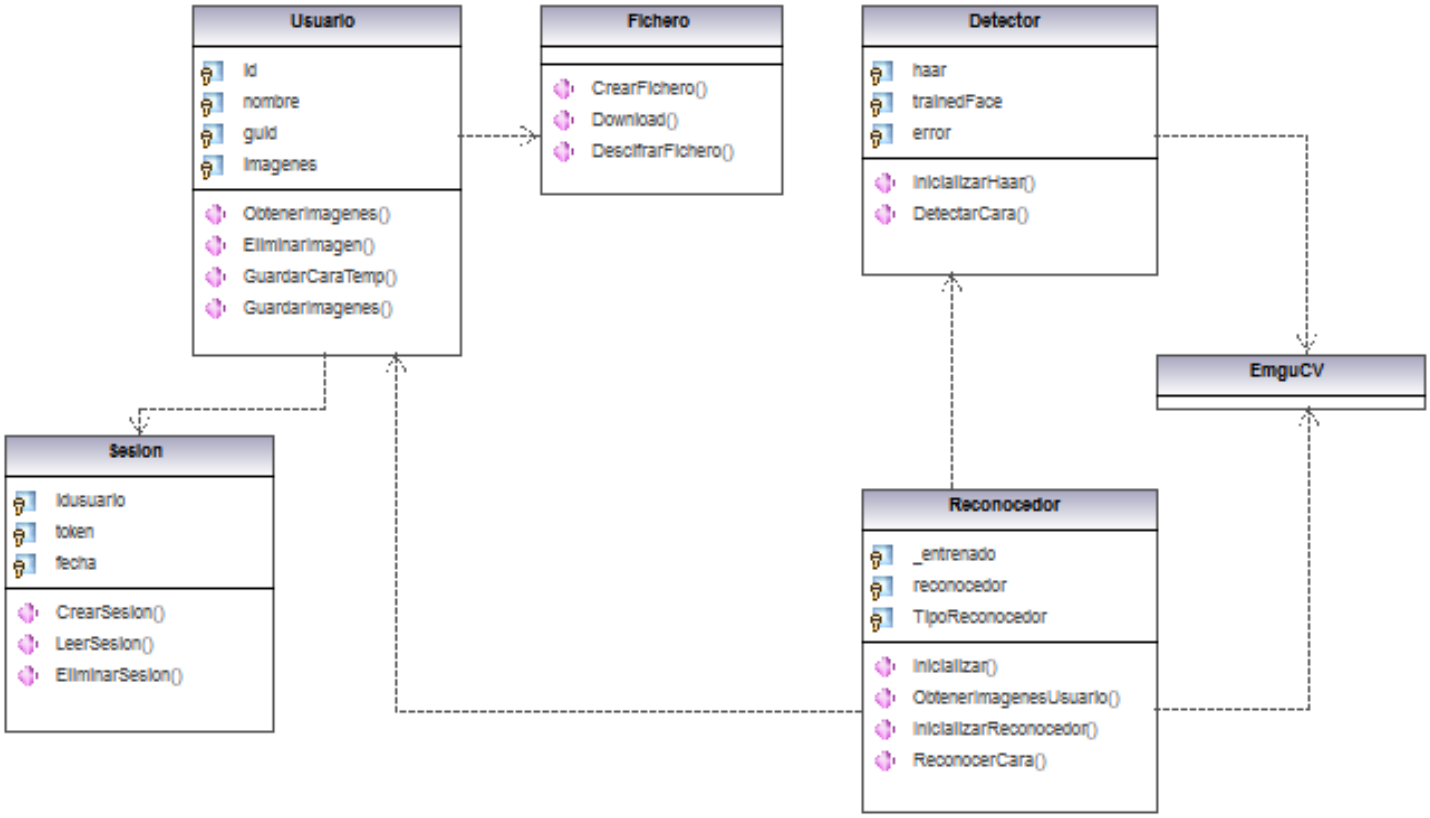

Figura 11. Diagrama de clases

4.2.1 Descripción de las clases

En este apartado se realizará una descripción de cada una de las clases en las que se ha decidido dividir el sistema.

Usuario

Dispone de los métodos necesarios para el manejo de la información del usuario, a excepción de las acciones de registro y de manejo de contraseña, lo cual es controlado mediante la SimpleMembership implementada en .Net.

- ObtenerImagenes: Obtiene todas las imágenes entrenadas por el usuario, las cuales se mostrarán en la pantalla de *Entrenar*.
- EliminarImagen: Elimina una imagen seleccionada por el usuario de la pantalla de entrenamiento.
- GuardarCaraTemp: Una vez detectada la cara del usuario en la imagen entrenada, se guardará en la carpeta de imágenes temporales asociadas al usuario.
- GuardarImagenes: Guarda las imágenes entrenadas por el usuario (almacenadas en la carpeta de imágenes temporales), en la carpeta definitiva asociada al mismo y además inserta la ruta asociada en la base de datos.

Detector

Dispone de los métodos necesarios para la detección de la cara del usuario en las imágenes, implementando las funciones disponibles en la librería EmguCV.

- InicializarHaar: Inicializa el componente Haar de EmguCV, cargando un archivo xml con el contenido.
- DetectarCara: Implementa la lógica necesaria para detectar el patrón de una cara dentro de una imagen que se le pase como parámetro.

Reconocedor

Dispone de los métodos necesarios para poder reconocer la cara obtenida mediante una imagen. Anteriormente al proceso de reconocimiento, se obtiene una imagen recortada que contiene únicamente la cara del usuario llamando al Detector.

- Inicializar: Este método inicializa el Reconocedor llamando a los métodos ObtenerImagenesUsuarios() y a InicializarReconocedor(). El primer método se encarga de obtener todas las imágenes de los usuarios disponibles en la base de datos, sobre las cuales se realizará el reconocimiento facial, comparándolas con la imagen de entrada. El segundo método inicializa el tipo de reconocedor seleccionado entre los disponibles en EmguCV (EMGU.CV.LBPHFaceRecognizer, EMGU.CV.FisherFaceRecognizer y EMGU.CV.EigenFaceRecognizer).

- ReconocerCara: Este método es el encargado de realizar la comparación para el reconocimiento facial. Se hace una llamada al método FaceRecognizer.Predict y pasando como parámetro la imagen que se quiere comparar. Dicho método devuelve el nombre del usuario

Sesion

Es la clase encargada del manejo del *token* de sesión asignado al usuario en la extensión de Google Chrome.

El flujo de funcionamiento de la extensión permite dos formas de actuar:

1. Activar la extensión, adjuntar el fichero y pulsar “Iniciar sesión” para que el sistema reconozca al usuario y autenticarlo con el fichero adjuntado. Si el usuario no es reconocido, no se descifra el fichero de credenciales.
2. Activar la extensión, pulsar “Iniciar Sesión” y, una vez reconocido al usuario, se podrá adjuntar el fichero de credenciales.

Con motivo de este último punto, cuando un usuario accede a la página de Facebook o de Twitter, la extensión le asigna un valor aleatorio, el cual será enviado al servidor en el momento de que se pulse iniciar sesión. Si no se ha adjuntado fichero y se reconoce al usuario, se crea una sesión para ese usuario con el token enviado, siendo válida durante un minuto. Si dentro de un minuto no se adjunta el fichero y se autentica, la sesión será eliminada del servidor, teniendo que realizar el usuario el proceso desde cero.

Fichero

Es la clase encargada de la creación del fichero con los credenciales del usuario cifrados. En el momento en el que el usuario se registra en la aplicación, se le asigna un Guid (*globally unique identifier*). Dicho Guid se almacena en la base de datos, en la tabla de usuarios, y se usará como clave para el cifrado/descifrado del fichero empleando para

ello el método 3DES implementado en la clase TripleDESCryptoServiceProvider de .Net. Implementa los siguientes métodos:

- CrearFichero(): Se encarga de crear y cifrar el contenido del fichero.

- Download(): Descarga en el ordenador del usuario el fichero creado en el punto anterior.

- DescifrarFichero(): Descifra el contenido del fichero.

4.3 Diagrama de Base de datos

A continuación se muestra la estructura de la base de datos del proyecto:

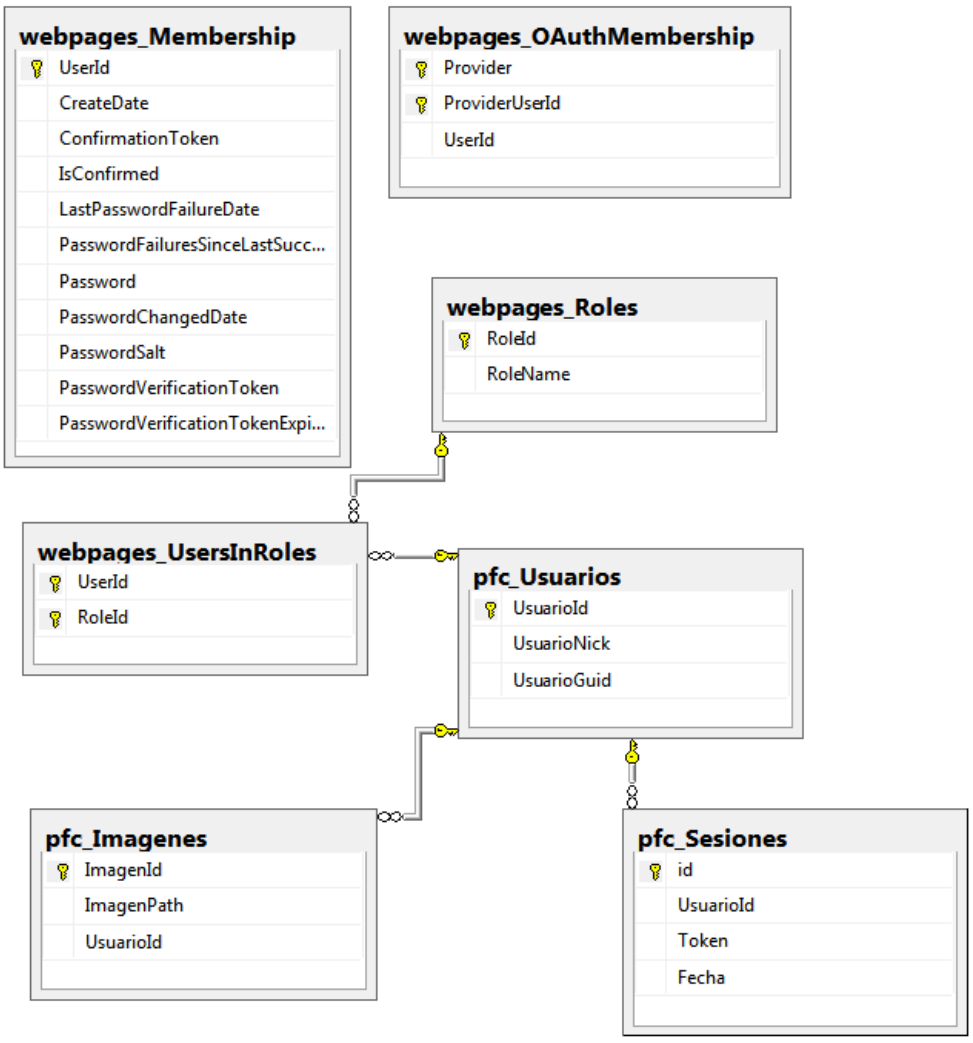

Figura 12. Diagrama de base de datos

Como se muestra en el diagrama, las tablas que comienzan por “webpages_” son las empleadas por el *Simplemembership Provider* de .NET para el manejo de las cuentas de usuario, facilitando la labor del manejo de las sesiones de los distintos usuarios de la aplicación, así como el registro de los mismos. Este sistema emplea la tabla pfc_Usuarios para el manejo de la información personalizada que necesite la aplicación. En la aplicación, cuando un usuario se registra, se le asigna un *ID* único y un *GUID* (un identificador único global) en el campo UsuarioGuid, el cual será el empleado para el cifrado del archivo.

Por otra parte, la tabla pfc_Imagenes es empleada para el manejo de las imágenes entrenadas de los usuarios, las cuales se guardan como una ruta relativa del archivo y enlazadas con el usuario al que pertenecen mediante el *UsuarioId*. La tabla pfc_Sesiones se utiliza para el manejo de las sesiones creadas en la extensión de Google Chrome.

4.4 Diagramas de secuencia

En este apartado se mostrarán los diagramas de secuencia que representan los casos de uso. En los diagramas se muestra el flujo normal de funcionamiento y no se tendrá en cuenta ningún flujo alternativo (tales como fallo al detectar la cara, fallo al introducir datos, etc).

Se han simplificado todos los diagramas eliminando interacciones con clases del lenguaje así como las invocaciones a métodos que no aportan información ni son de interés, como los métodos get y set. De esta forma se ha reducido considerablemente el tamaño de los diagramas y con esto la claridad de los mismos pudiéndose visualizar las interacciones más importantes.

4.4.1 Entrenar el sistema (CU-1)

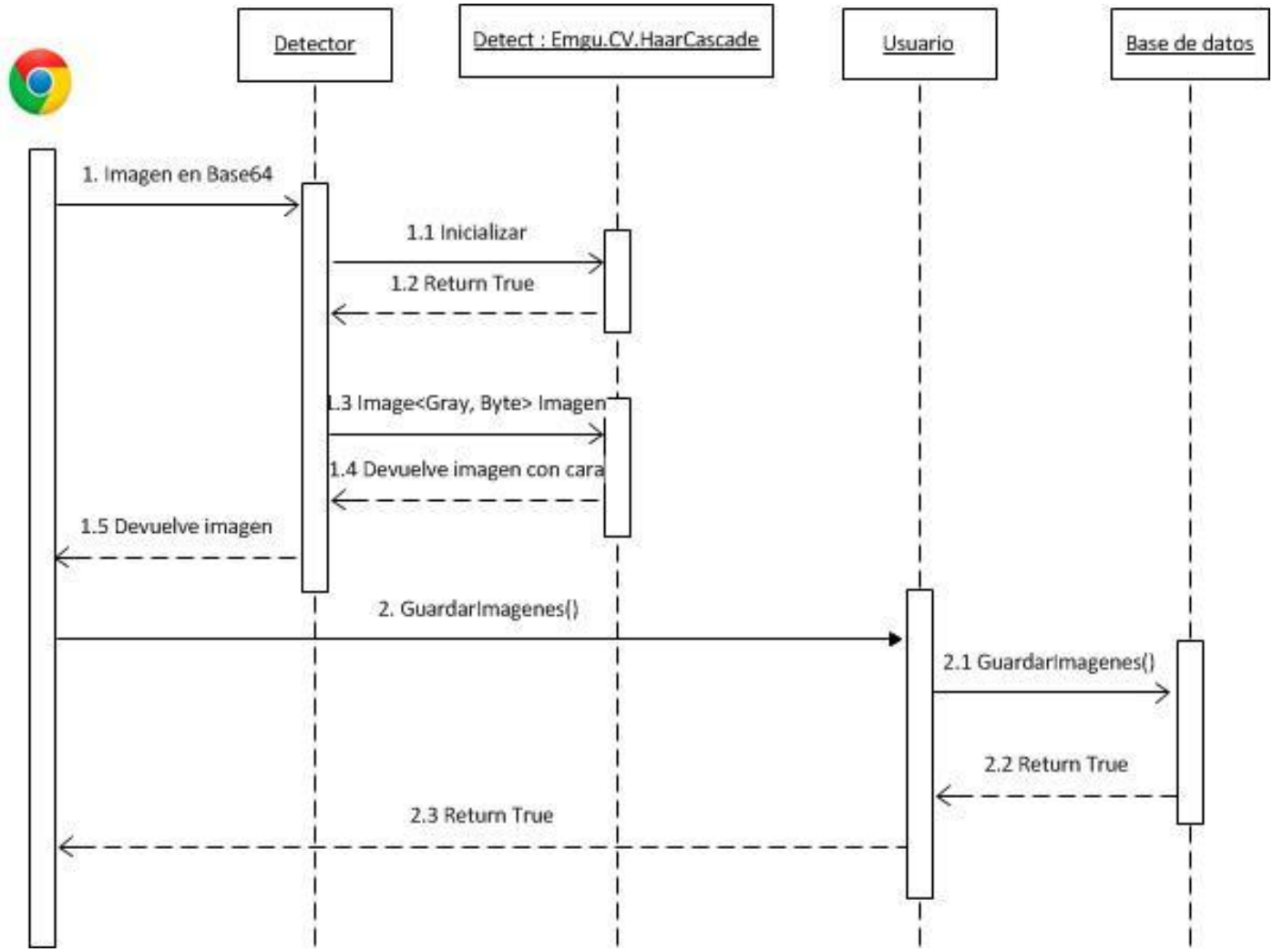

Figura 13. Diagrama de secuencia: Entrenamiento del sistema

En el anterior diagrama de secuencia se muestran las actividades que se realizan a la hora de entrenar el sistema.

Al pulsar el botón "Captura", se envía la imagen al detector, el cual es inicializado en ese momento, cargando el archivo Haar correspondiente. Tras esto se realiza una llamada al método Detect disponible en Emgu.CV.HaarCascade. Este método devuelve una imagen con la cara del usuario y la muestra en el navegador web.

4.4.2 Crear fichero de credenciales cifrado (CU-2)

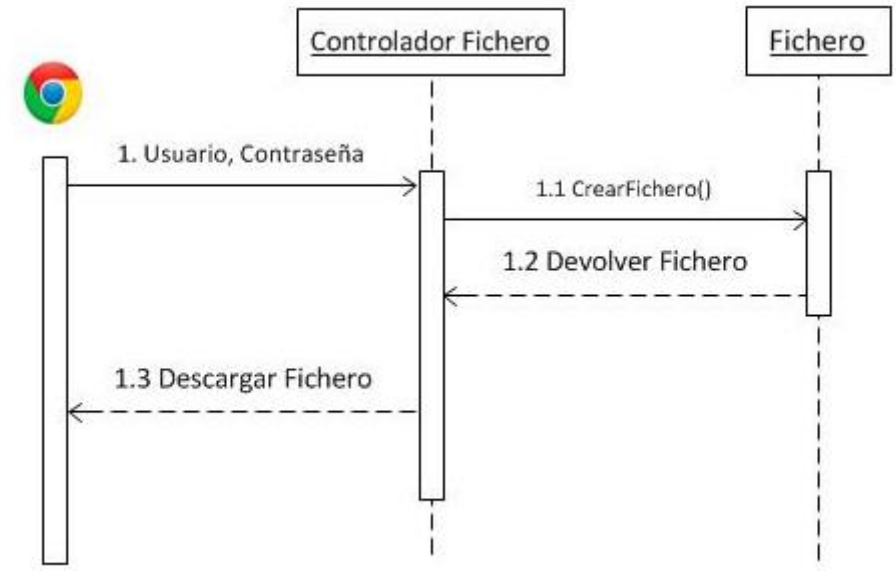

Figura 14. Diagrama de secuencia: Crear fichero de credenciales

En el anterior diagrama de secuencia se muestran las actividades necesarias para la creación delos ficheros de credenciales cifrados.

4.4.3 Iniciar sesión en Facebook/Twitter (CU-3)

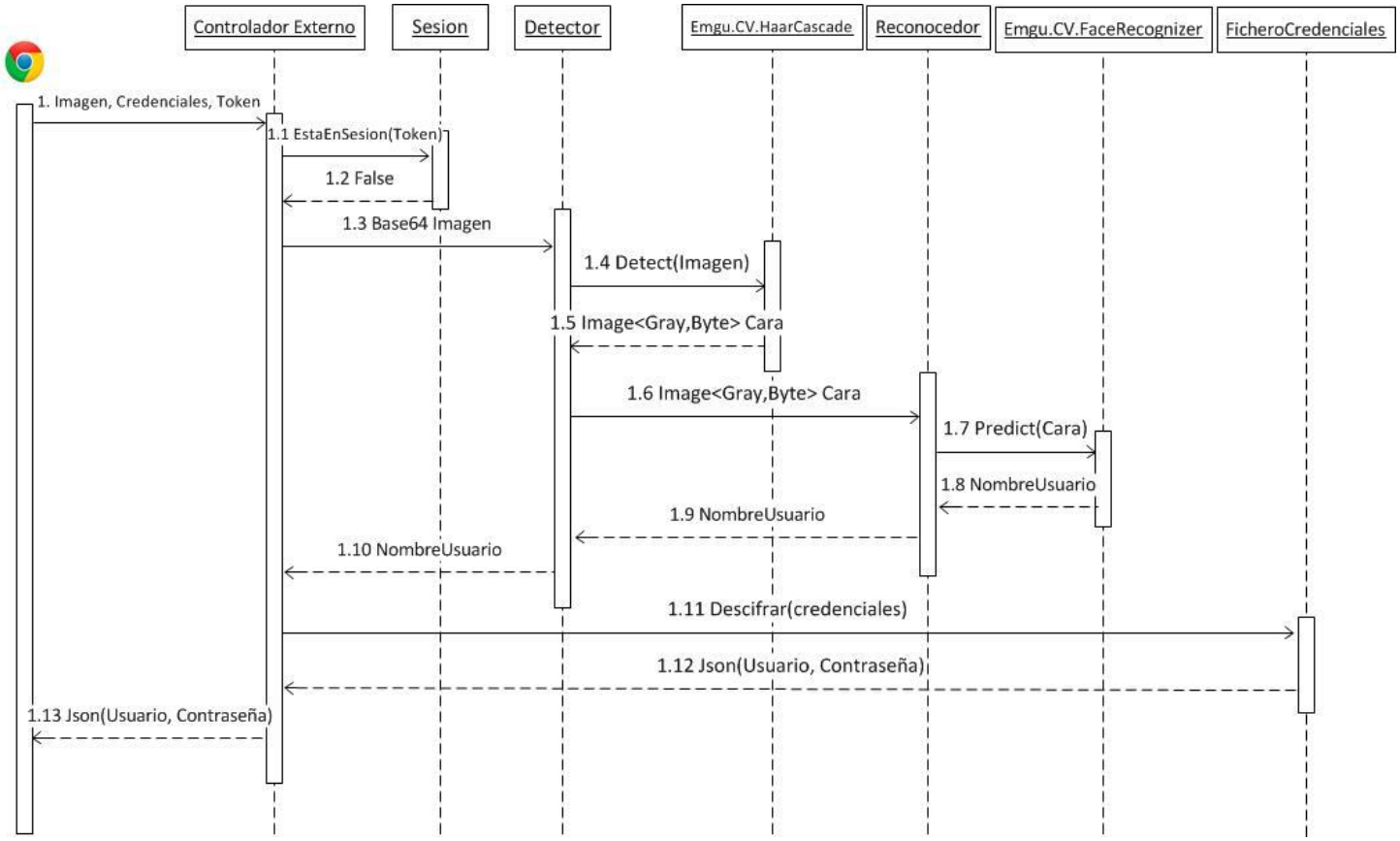

Figura 15. Diagrama de secuencia: Iniciar sesión en red social

En este diagrama se muestra la acción de autenticar en la red social seleccionada adjuntando a la vez el archivo de credenciales cifrado. Para ello, y teniendo la extensión de Google Chrome activada y habiendo seleccionado el fichero de credenciales, se pulsa el botón iniciar sesión.

La extensión envía al servidor la imagen, el contenido del fichero de credenciales y el token. Primero se comprueba que el usuario no está en sesión, lo cual devuelve el valor False, y posteriormente se detecta la cara en la imagen enviada. Tras esto, se envía la imagen facial al Reconocedor, el cual llama al método Predict del FaceRecognizer de EmguCV obteniendo el nombre del usuario que ha reconocido, devolviéndolo a la extensión.

Una vez obtenido el nombre del usuario, se analiza la cadena de credenciales, descifrándolo y obteniendo los valores del usuario y contraseña de la red social. Con esto, se introduce este valor en las cajas de texto destinadas a ello en la página web y se envía el formulario.

4.5 Descripción del sistema

El sistema se compone de dos partes. Una parte consistente en un cliente web, en el cual el usuario puede administrar sus datos y crear los ficheros de credenciales cifrados con los datos de la red social. Además, también se desarrolló una extensión en Google Chrome para poder iniciar sesión en la red social.

Para el desarrollo del cliente web se ha seguido el patrón de desarrollo MVC de Framework .Net 4, empleando como lenguajes de programación ASP.NET sobre C#. En dicho cliente web, el usuario puede sacarse fotografías y guardarlas, enlazándolas así con su usuario para que pueda ser reconocido posteriormente (entrenamiento), además de eliminar aquellas que no considere útiles. La otra funcionalidad de la página web será la creación de un fichero con los credenciales necesarios para acceder a la red social cifrado (en Facebook es necesario la dirección de correo electrónico y la contraseña y en Twitter el nombre de usuario y la contraseña).

La extensión de Google Chrome carga los controles HTML necesarios en las páginas web de Facebook y de Twitter para proporcionar al sistema el acceso y gestión de la cámara web del usuario y para que el usuario pueda adjuntar el fichero de credenciales. Asimismo, es necesaria para rellenar con los valores extraídos del fichero cifrado, tras el reconocimiento facial, los campos de usuario y de contraseña de ambas redes sociales.

Capítulo 5

Implementación

5.1 Fases en el reconocimiento facial

Para realizar la implementación del reconocimiento facial se realizan los siguientes pasos:

- **Detección**: El sistema detecta una cara en la imagen.
- **Alineación**: Localiza y normaliza respecto a un patrón geométrico establecido y un formato de obtención de datos los rasgos de interés de la cara.
- **Características**: Una vez procesada la imagen en el paso anterior, carga el sistema con la información necesaria para realizar las comparaciones precisas contra la base de datos. Es en esta parte donde se emplea alguno de los algoritmos que se explican en el siguiente apartado.
- **Identificación**: Compara los valores obtenidos de analizar las imágenes en la base de datos con la imagen de entrada.

5.2 Cliente Web

Como se ha comentado en el Capítulo 4, para el control de usuarios se ha implementado el sistema SimpleMembership provisto por el Framework de desarrollo. Este sistema automatiza todo el sistema de control de usuarios (creación, actualización y eliminación) así como el manejo de sesiones de los mismos (iniciar sesión y cerrar sesión). En este caso se ha elegido la tabla pfc_Usuarios, en el cual se recoge la información del usuario, a excepción de la contraseña, que se encuentra en webpages_Membership.

El uso del cliente web será únicamente para entrenar al sistema con imágenes del usuario. Podrá añadir nuevas fotografías y eliminarlas.

Para la parte del detector facial, se implementa la clase Detector, en la cual se ha especificado la función “DetectarCara” que es la encargada de detectar el patrón perteneciente a una cara en una imagen que se le pase como parámetro. Para realizar esto, se debe inicializar el componente HaarCascade, del cual hablaremos en este capítulo más adelante, asignándole un fichero XML.

Se espera recibir esta imagen de un control video HTML5 en una cadena en base64, por lo cual es necesaria una función para convertir esta información en una variable de tipo Image de EmguCV para poder utilizarlo en el detector. Usaremos una imagen en escala de grises para eliminar cualquier diferenciación de colores que pueda producir fallos en el reconocimiento.

Tras esto se llama a la función Detect pasándole como parámetro la imagen en escala de grises obtenida anteriormente. Después de la detección, el flujo de funcionamiento puede requerir:

- devolver la cara reconocida al cliente web, con lo que se mostrará en la interfaz la última imagen obtenida. En la imagen se muestra la zona de entrenamiento de la interfaz del usuario, donde se pueden ver las imágenes entrenadas asociadas al usuario.

- llamar a la clase Reconocedor con la imagen de la cara reconocida. Esta clase es la encargada de reconocer la imagen con la cara que se le envía como parámetro. Se pueden modificar los parámetros a la hora de inicializar el reconocedor, para comprobar cuando presenta una mayor ratio de acierto en la detección. Para cada algoritmo varían los valores a inicializar:

  1. El **EigenFaceRecognizer** presenta dos valores de inicio:
     a. numComponents: Es el número de componentes guardados para el PCA. No hay ninguna regla sobre qué valor debe tener. La documentación de Open CV sugiere mantener 80 componentes.
     b. Threshold: Representa el umbral de predicción. En este algoritmo, este umbral funciona de manera inversa a FisherFaceRecognizer y LBPHFaceRecognizer.

     Como resultado de reconocer la cara presente en la imagen, se obtiene un valor. Cuanto mayor sea este valor, más "seguro" está el sistema de que la persona que indica es la que está en la imagen.

     ```
     new EigenFaceRecognizer(80, 3500);
     ```

  2. El **FisherFaceRecognizer** presenta dos valores de inicio:
     - numComponents: Número de componentes que se utilizarán en el LDA.
     - Threshold: Representa el umbral de predicción. Como se comentó anteriormente, el reconocedor da un resultado por cada cara comparada. La imagen que obtenga menor valor será la seleccionada. Si el valor está por encima de este umbral, dará como resultado un valor "Unknown" (Desconocido).

     ```
     new FisherFaceRecognizer(80, 3500);
     ```

  3. El **LBPHFaceRecognizer** presenta cinco valores de inicio:
     - Radius: El radio utilizado para construir los Patrones Binarios Locales Circulares (Circular Local Binary Patterns, CLBP).

- Neighbors: El número de puntos de muestreo para construir un CLBP. Un valor sugerido por la documentación de OpenCV es de '8'. Hay que tener en cuenta que a mayor número de puntos de muestreo, mayor será el coste computacional.
- gridX: El número de celdas en la dirección horizontal, 8 es un valor recomendado.
- gridY: El número de celdas en la dirección vertical, 8 es un valor recomendado.
- threshold: El umbral aplicado en el algoritmo, que funciona de la misma manera que en FisherFaceRecognizer.

```
new LBPHFaceRecognizer(1, 8, 8, 8, 100);
```

Después de inicializar el reconocedor y con las imágenes de la base de datos cargadas, se llama a la función Predict, la cual es la que realiza todo el proceso del reconocimiento facial:

```
FaceRecognizer.PredictionResult ER = reconocedor.Predict(imagen);
```

Predict compara la imagen pasada como parámetro con todas las obtenidas de la base de datos correspondientes al set de entrenamiento. La imagen que obtenga un mayor valor (o menor, dependiendo del algoritmo empleado) en esta comparación, y si entra dentro de un límite definido, es la seleccionada. Se extrae el nombre asociado a esta imagen y éste es el valor devuelto.

Otra parte importante es la que encripta el fichero con los credenciales del usuario. Al registrar al usuario en la aplicación se le asigna automáticamente un Guid aleatorio que es el utilizado a la hora de encriptar los credenciales. Primero se obtiene una codificación md5 de este Guid, y posteriormente se emplea la cadena resultante a la hora de encriptar el fichero.

5.2.1 Archivo de cascada Haar

Detectar si dentro de una imagen se encuentra un rostro humano es el paso previo a reconocerlo o identificarlo, esta tarea se basa en comparaciones con patrones específicos de un rostro humano tales como su morfología, ubicación relativa de ojos, nariz y boca entre sí, color característico de la piel y combinaciones de tales características que hacen algoritmos más sofisticados.

Para realizar la implementación de esta dinámica, es necesario emplear clasificadores en cascada con filtros de base Haar, también conocidos tan solo como clasificadores Haar. Estos clasificadores permiten realizar la detección facial.

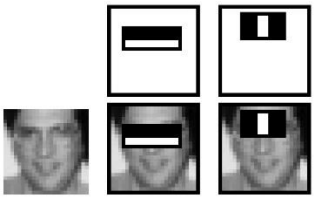

Figura 16. Ejemplo de componentes Haar

La librería EmguCV posee una implementación de clasificadores Haar, el cual se debe cargar en el momento de inicializar el detector.

5.3 Extensión de Google Chrome

La extensión de Google Chome desarrollada inyecta en las páginas web establecidas (configuradas en el archivo manifest.json) los controles necesarios para realizar el reconocimiento facial. En este caso inserta el control video, un control para poder adjuntar el fichero de credenciales y un botón. Además se mostrará en la barra de direcciones un icono indicando que el inicio de sesión mediante reconocimiento facial está disponible en esa página.

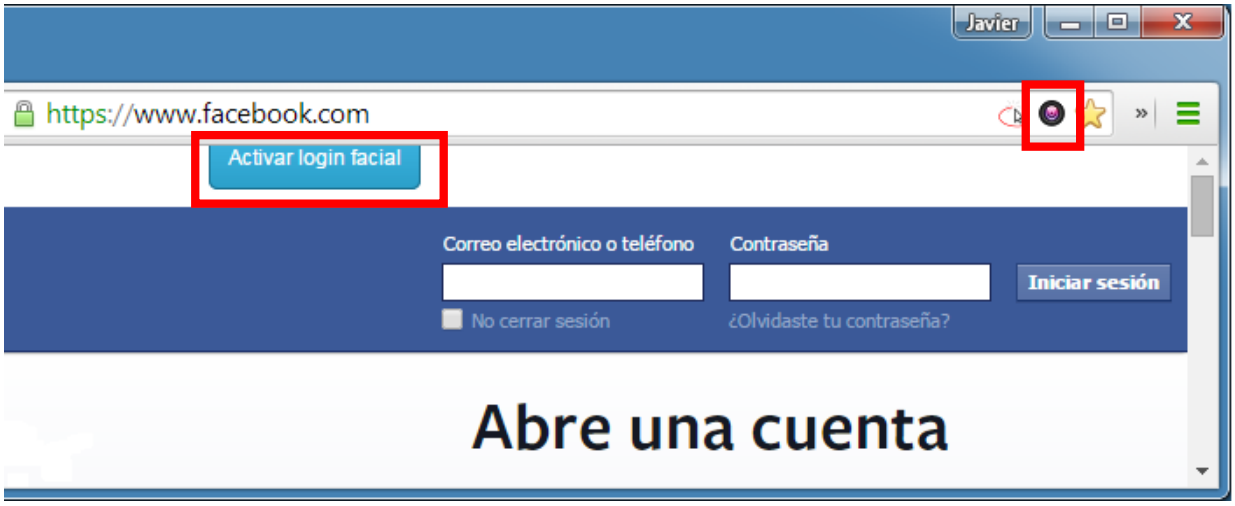

Figura 17. Extensión de Google Chrome instalada y activada

Cuando el usuario quiera realizar el reconocimiento facial, únicamente debe pulsar en “Activar login facial” y se desplegarán los controles. En ese momento deberá

sacarse la fotografía y adjuntar el fichero de credenciales. Si todo funciona correctamente, el inicio de sesión se producirá automáticamente.

Para realizar la extensión de Google Chrome han sido necesarios los siguientes ficheros:

1. manifest.json: Es el archivo principal de la extensión y contiene toda la información básica de la extensión. En él se establecen los permisos y las acciones que debe realizar la extensión, así como los archivos que se deben cargar para que la extensión funcione correctamente.

2. background.js: Este archivo es el encargado de definir qué páginas utilizarán el código. Esto se define de la siguiente forma:

   Además hará de intermediario entre el servidor y la extensión, recibiendo de la extensión la imagen, el fichero y un *Token* con la sesión actual y enviando dicha información al servidor.

   Tras esto, también es el encargado de recibir la información del servidor con el resultado y enviarle esta información a la extensión

```
chrome.declarativeContent.onPageChanged.removeRules(undefined, function ()
{
      // Agregamos una nueva regla
      chrome.declarativeContent.onPageChanged.addRules([
      {
            // que se ejecuta cuando se cumplen las siguientes condiciones
            conditions: [
                  new chrome.declarativeContent.PageStateMatcher({
                        pageUrl: { urlContains: 'facebook' }
                  }),
                  new chrome.declarativeContent.PageStateMatcher({
                              pageUrl: { urlContains: 'twitter' }
                              }),
                  new chrome.declarativeContent.PageStateMatcher({
                        pageUrl: { urlContains: 'LoginPFC' }
                     })
               ],
               // Y se muestra la extensión si se cumple
               actions: [new chrome.declarativeContent.ShowPageAction()]
      }]);
});
```

3. contenido.js: Este archivo contiene el código necesario para la creación dinámica de los controles de la extensión y de las funciones necesarias para la utilización de la cámara web, sacar fotografías y enviar la información necesaria al servidor.

```
chrome.extension.onMessage.addListener(
   function (request, sender, sendResponseParam) {
      if (request.method == 'LoginFacial') {
        $.post("http://localhost/LoginPFC/External/Login", { _ "imagen":
        request.img, "cred": request.cred, "sesion": _ request.sesion },
        function (respuesta)
        {
           var cred = JSON.parse(respuesta);
            if (cred[0]["Mensaje"] == null) {
               if (respuesta != "") {
                   sendResponseParam({ usuario: cred[0]["usuario"], pass:
                         cred[0]["pass"] });
                   }
                   else {
                       sendResponseParam({ usuario: null, pass: null });
                   }
                }
                else {
                   sendResponseParam({ Mensaje: cred[0]["Mensaje"] });
                }
           });
           }
      return true;
      }
);
```

Una vez tomada la fotografía, la extensión la envía mediante un mensaje al fichero background, junto con un token de control. A su vez este fichero recibe el mensaje y lo reenvía a la función LoginExterno en el API externo de la aplicación, la cual será la encargada de realizar toda la funcionalidad de autenticación facial en la red social tal y como se especifica más adelante en este capítulo.

El flujo de la función es el mostrado en la Figura 18:

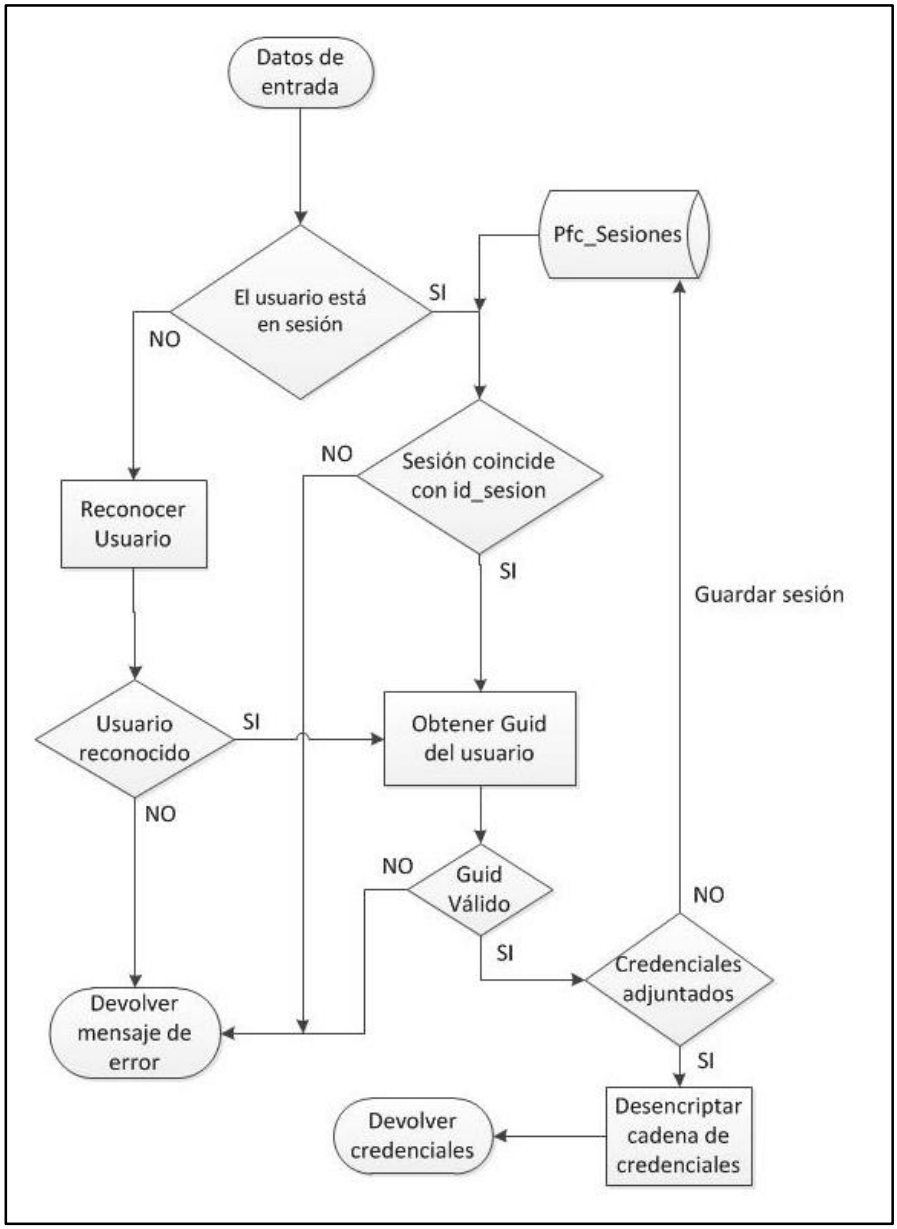

Figura 18. Flujo de la función LoginExterno

El *Token* y el almacenamiento de la sesión son necesarios debido a que el usuario puede primero reconocer su cara y posteriormente adjuntar el fichero (es decir, puede no realizar las dos cosas en el mismo momento).

En el caso de que el usuario siga este flujo es necesario recordar su sesión para que, en el momento en el que adjunte el fichero de credenciales encriptado, el sistema no lo tenga que reconocer otra vez, siempre y cuando el fichero sea adjuntado en el minuto posterior al reconocimiento. Esto evitará una sobrecarga innecesaria en el sistema, reconociendo al mismo usuario dos veces en un periodo de tiempo muy corto. Si no se adjunta en este límite de tiempo, la sesión se borra y se debe iniciar el proceso de reconocimiento.

5.4 Resultados de las pruebas de aceptación

En esta sección se muestran los resultados de las pruebas realizadas, detalladas en la sección 3.7. Como se puede ver en la tabla, se han superado todas las pruebas.

Pruebas de aceptación		
Id.	**Elementos probados**	**Resultado**
PA-01	RF-01	Superada
PA-02	RF-02	Superada
PA-03	RF-03, RF-08	Superada
PA-04	RF-04, RF-05	Superada
PA-05	RF-06	Superada
PA-06	RF-07	Superada
PA-07	RF-09	Superada

Tabla 8. Resultados de las pruebas de aceptación

Capítulo 6

Pruebas

Como se ha comentado en apartados anteriores, las pruebas estarán limitadas a unas condiciones concretas, ya que las debilidades de estos métodos quedan fuera del alcance de esta implementación.

Las pruebas se realizaron con una buena iluminación y con una mínima variación del ángulo (dentro de los 20° ya que, como se comentó en el Capítulo 6.2 sobre Trabajo futuro, a partir de ese ángulo se pierde eficacia en el reconocimiento) . Además, no hay cambios físicos en el usuario entre las imágenes entrenadas y las imágenes de prueba.

La base de datos con las imágenes para poder realizar las pruebas la componen los rostros de los distintos usuarios a los que se les realizan las mismas (la cantidad de imágenes de cada usuario se indica al comienzo de la prueba), mientras que el resto de imágenes están constituidas por rostros de personajes públicos, extraídos de imágenes accesibles por internet a las que se les ha extraído el rostro.

6.1 Algoritmo Eigenfaces

Usuario 1. Mujer

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	8 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 5.117 s 80 componentes: 5.422 s 120 componentes: 5.8 s

Tabla 9. Datos de la prueba del Usuario 1

Se han realizado tres bloques de diez pruebas cada uno, modificando el número de componentes y obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	7/10
80	8/10
120	8/10

Tabla 10. Resultados de Eigenfaces en Usuario 1

Como podemos observar, no hay gran diferencia entre los cambios de la variable numComponentes con estas condiciones. El error producido con *numComponentes = 40* pudo deberse a la pose del usuario.

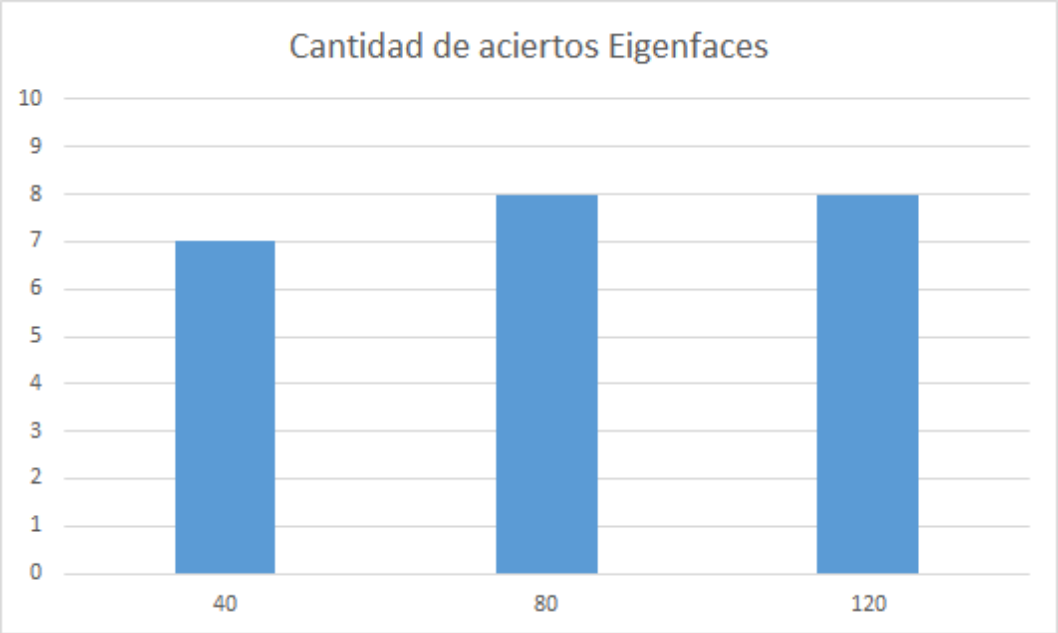

Figura 19. Resultados de Eigenfaces en Usuario 1

Usuario 2. Hombre 1

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	10 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 5.531 s 80 componentes: 5.713 s 120 componentes: 6.121 s

Tabla 11. Datos de la prueba del Usuario 2

Al igual que al usuario anterior, se han realizado tres bloques de diez pruebas cada uno, modificando el número de componentes y la cantidad de imágenes del usuario, obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	8/10
80	8/10
120	8/10

Tabla 12. Resultados de Eigenfaces en Usuario 2

Como se puede ver, en este caso las tres pruebas han dado los mismos resultados, aunque con un tiempo medio ligeramente superior al Usuario 1 en cada una de las pruebas.

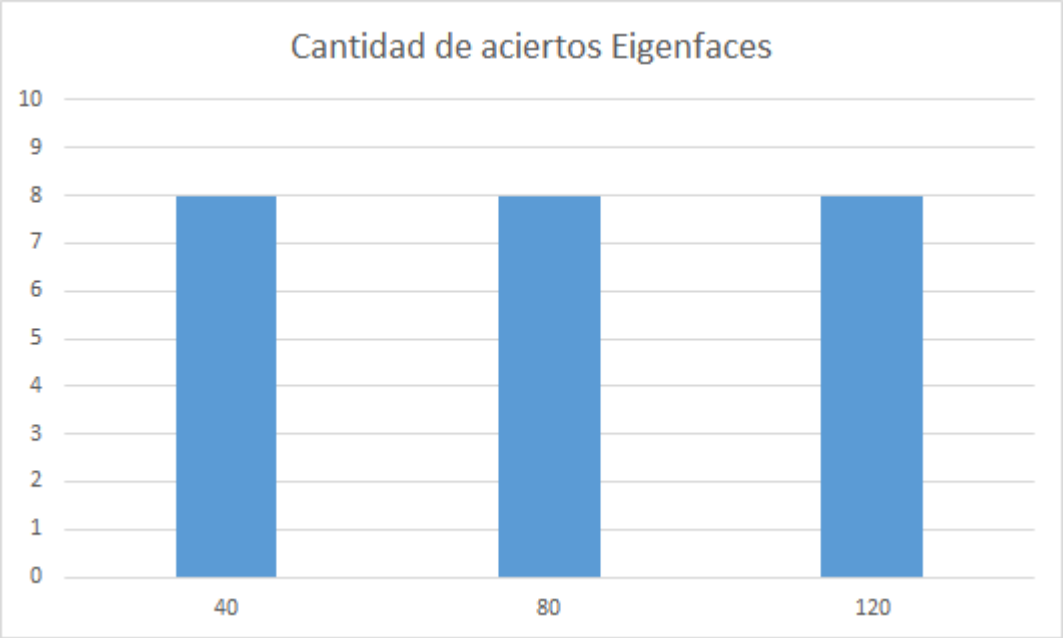

Figura 20. Resultados de Eigenfaces en Usuario 2

Usuario 3. Hombre 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	12 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 5.470 s 80 componentes: 5.759 s 120 componentes: 6.151 s

Tabla 13. Datos de la prueba del Usuario 3

Al igual que con los usuarios anteriores, se han realizado tres bloques de diez pruebas cada uno, modificando el número de componentes y una vez más la cantidad de imágenes del usuario, obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	8/10
80	9/10
120	9/10

Tabla 14. Resultados de Eigenfaces en Usuario 3

En esta ocasión podemos comprobar como al aplicar Eigenfaces con 40 componentes obtenemos otra vez peores resultados que con los otros dos, como sucedió en el Usuario 1.

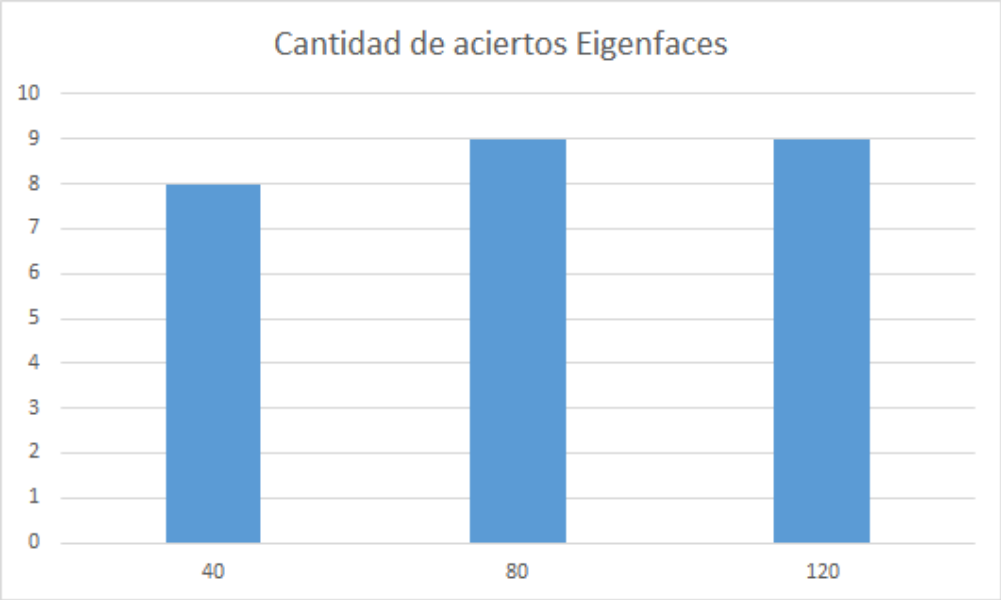

Figura 21. Resultados de Eigenfaces en Usuario 3

Usuario 4. Mujer 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	14 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 5,346 s 80 componentes: 5.618 s 120 componentes: 6.045 s

Tabla 15. Datos de la prueba del Usuario 4

Para el cuarto usuario también se han realizado tres bloques de diez pruebas cada uno, en los cuales se ha aumentado una vez más el número de imágenes entrenadas del usuario, obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	8/10
80	10/10
120	8/10

Tabla 16. Resultados de Eigenfaces en Usuario 4

En esta ocasión podemos comprobar que el algoritmo Eigenfaces ha tenido mejores resultados en su versión con 80 componentes, obteniendo con 40 y 120 componentes peores resultados, aunque en estos dos últimos se ha alcanzado un 80% de aciertos.

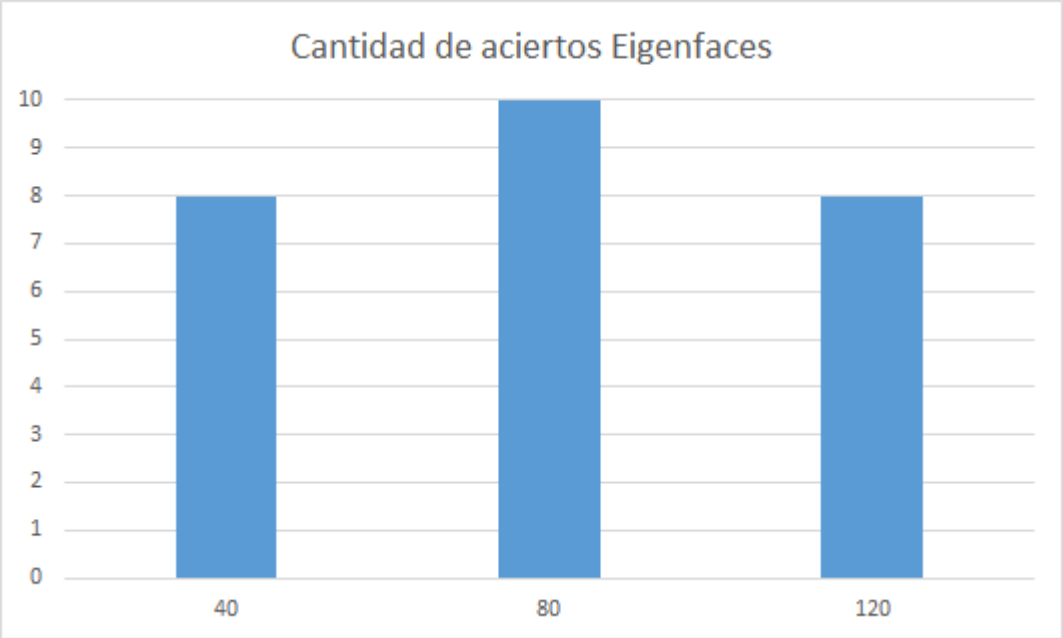

Tabla 17. Resultados de Eigenfaces en Usuario 4

Resumen de los resultados en Eigenfaces

Como se ha podido observar en los diferentes resultados de las pruebas, la versión de 80 componentes ha obtenido de media los mejores resultados (87.5%), aumentado la tasa de aciertos a medida que las imágenes entrenadas de cada usuario han ido aumentando, alcanzando el 100% de aciertos en la última prueba.

La versión de 40 componentes ha obtenido de media peores resultados que las otras dos versiones, 77.5%.

Por último, la versión de 120 componentes ha obtenido unos buenos resultados, obteniendo una media de 82.5%, siendo inferior en todo caso a Eigenfaces 80, además de requerir un tiempo de respuesta mayor.

En este caso se cumplen las recomendaciones de la documentación de EmguCV, en la cual especifican que 80 componentes deberían ser más que suficientes para obtener unos buenos resultados, siendo en el peor de los casos del 80% de aciertos y en el mejor de los casos ha sido del 100%.

6.2 Algoritmo Fisherfaces

A continuación planteamos las mismas pruebas que con Eigenfaces, modificando el número de componentes y la cantidad de imágenes entrenadas por cada usuario. Se muestran los resultados obtenidos de las pruebas realizadas.

Usuario 1. Mujer

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	8 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 7.151 s 80 componentes: 7.181 s 120 componentes: 7.200 s

Tabla 18. Datos de la prueba del Usuario 1

Planteamos ahora las mismas pruebas, y en las mismas condiciones, que para el algoritmo anterior, siendo los resultados los siguientes:

Número de componentes	Número de aciertos
40	7/10
80	8/10
120	8/10

Tabla 19. Resultados de Fisherfaces en Usuario 1

Como podemos comprobar, el funcionamiento en este primer usuario del algoritmo Fisherface es igual al de Eigenface, siendo la versión de 40 componentes la que peor resultado obtiene.

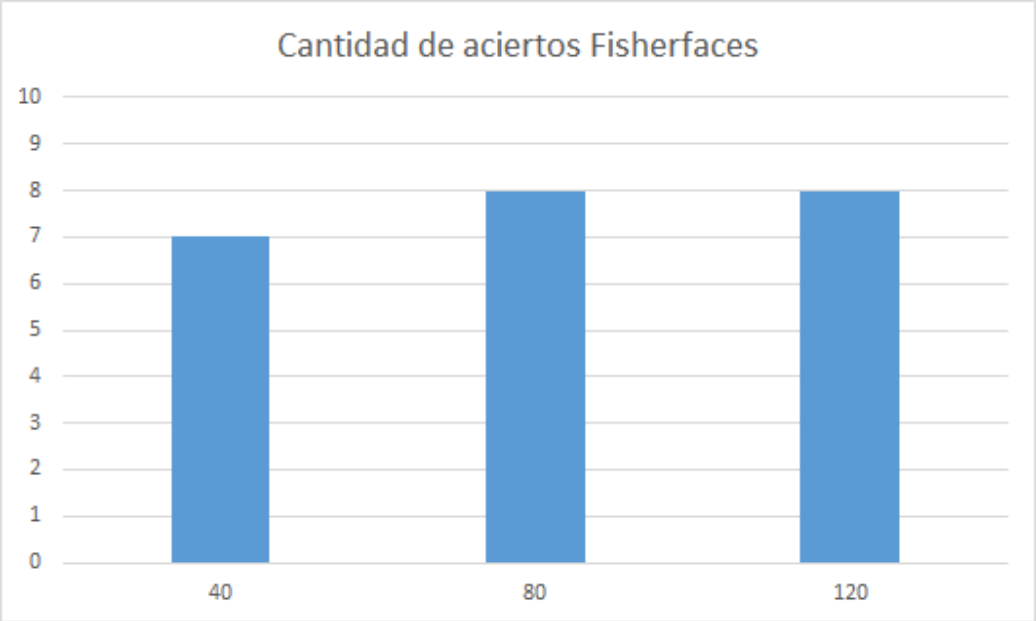

Figura 22. Resultados de Fisherfaces en Usuario 1

Usuario 2. Hombre 1

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	10 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 7.293 s 80 componentes: 7.620 s 120 componentes: 7.763 s

Tabla 20. Datos de la prueba del Usuario 2

A continuación se muestran los resultados de las pruebas realizadas al Usuario 2, aumentado el número de imágenes entrenadas:

Número de componentes	Número de aciertos
40	8/10
80	10/10
120	9/10

Tabla 21. Resultados de Fisherfaces en Usuario 2

En esta ocasión, utilizando 80 componentes obtenemos los mejores resultados.

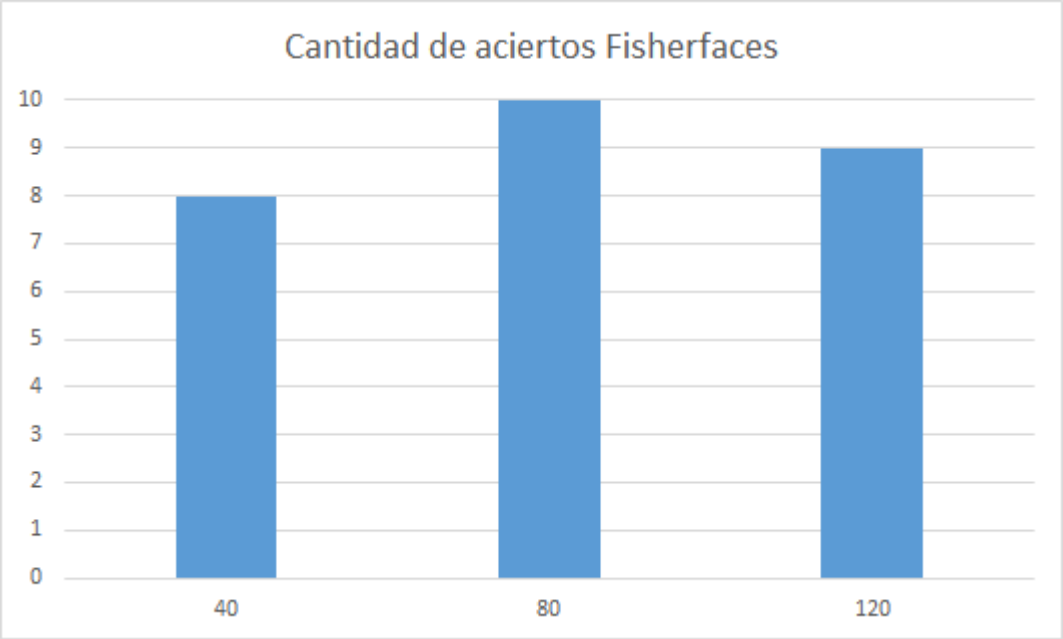

Figura 23. Resultados de Fisherfaces en Usuario 2

Usuario 3. Hombre 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	12 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 7.479 s 80 componentes: 7.703 s 120 componentes: 7.913 s

Tabla 22. Datos de la prueba del Usuario 3

Para el tercer usuario aumentamos una vez más el número de, obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	8/10
80	9/10
120	9/10

Tabla 23. Resultados de Fisherfaces en Usuario 3

En esta ocasión, los algoritmos de 80 y de 120 componentes obtienen los mismos resultados, quedando una vez más el algoritmo con 40 componentes en última posición.

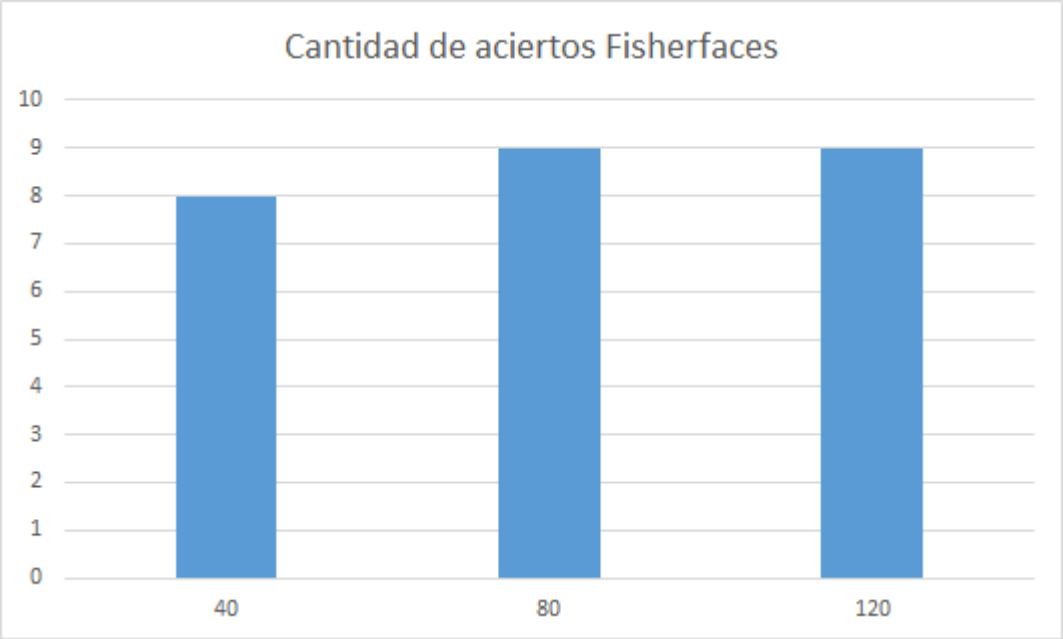

Figura 24. Resultados de Fisherfaces en Usuario 3

Usuario 4. Mujer 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	14 imágenes del usuario
Número de pruebas	10 por bloque. 3 bloques de pruebas
Tiempo medio del reconocimiento	40 componentes: 7.660 s 80 componentes: 7.558 s 120 componentes: 7.601 s

Tabla 24. Datos de la prueba del Usuario 4

Para la última prueba de este algoritmo, en el último usuario, volvemos a aumentar una vez más el número de imágenes, obteniendo los siguientes resultados:

Número de componentes	Número de aciertos
40	9/10
80	10/10
120	9/10

Tabla 25. Resultados de Fisherfaces en Usuario 4

En esta última prueba, el algoritmo Fisherfaces ha obtenido unos muy buenos resultados en cada una de las tres pruebas, con una tasa de acierto del 90% con 40 y 120 componentes, y con un 100% de aciertos con 80 componentes.

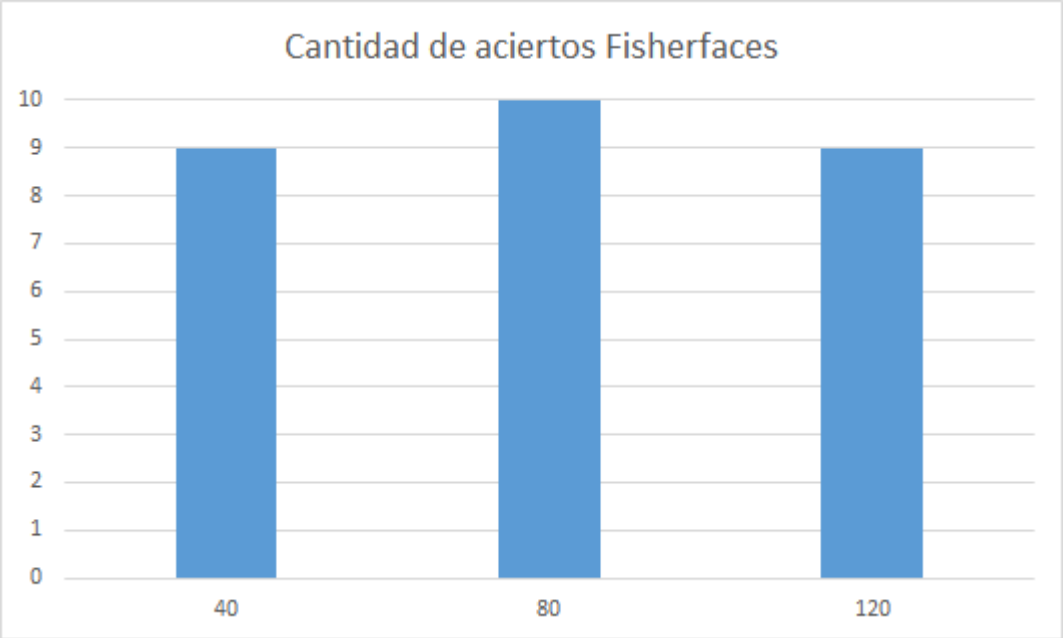

Figura 25. Resultados de Fisherfaces en Usuario 4

Resumen de los resultados en Fisherfaces

Al igual que con el algoritmo Eigenfaces, en el Fisherfaces con 40 componentes se obtienen los peores resultados de media en las cuatro pruebas realizadas. Y una vez más, con 80 componentes se ha comportado mejor que con 120 componentes, además de necesitar menos tiempo para realizar el reconocimiento.

Sin embargo, en comparación con Eigenfaces el tiempo medio necesario para el reconocimiento aumenta en un 33.55%, es decir, casi 2 segundos. (5.628 de media en Eigenfaces 80 frente a los 7,516 de Fisherfaces 80).

6.3 Algoritmo LBPH

Para este algoritmo se ha ejecutado una prueba con los siguientes valores:

Radius	1
Neighbors	8
gridX	8
gridY	8
threshold	100

Tabla 26. Datos de la prueba con LBPH

Se muestran a continuación los resultados obtenidos mediante este algoritmo para los cuatro usuarios:

Usuario 1. Mujer

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	8 imágenes del usuario
Número de pruebas	10 pruebas
Número de aciertos	9 aciertos
Tiempo medio del reconocimiento	1.287 segundos

Tabla 27. Resultados de LBPH en Usuario 1

Usuario 2. Mujer 1

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	10 imágenes del usuario
Número de pruebas	10 pruebas
Número de aciertos	10 aciertos
Tiempo medio del reconocimiento	1.332 segundos

Tabla 28. Resultados de LBPH en Usuario 2

Usuario 3. Hombre 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	12 imágenes del usuario
Número de pruebas	10 pruebas
Número de aciertos	10 aciertos
Tiempo medio del reconocimiento	1.379 segundos

Tabla 29. Resultados de LBPH en Usuario 3

Usuario 4. Mujer 2

Número de imágenes entrenadas en total	468 imágenes
Número de imágenes entrenadas del usuario	14 imágenes del usuario
Número de pruebas	10 pruebas
Número de aciertos	10 aciertos
Tiempo medio del reconocimiento	1.279 segundos

Tabla 30. Resultados de LBPH en Usuario 4

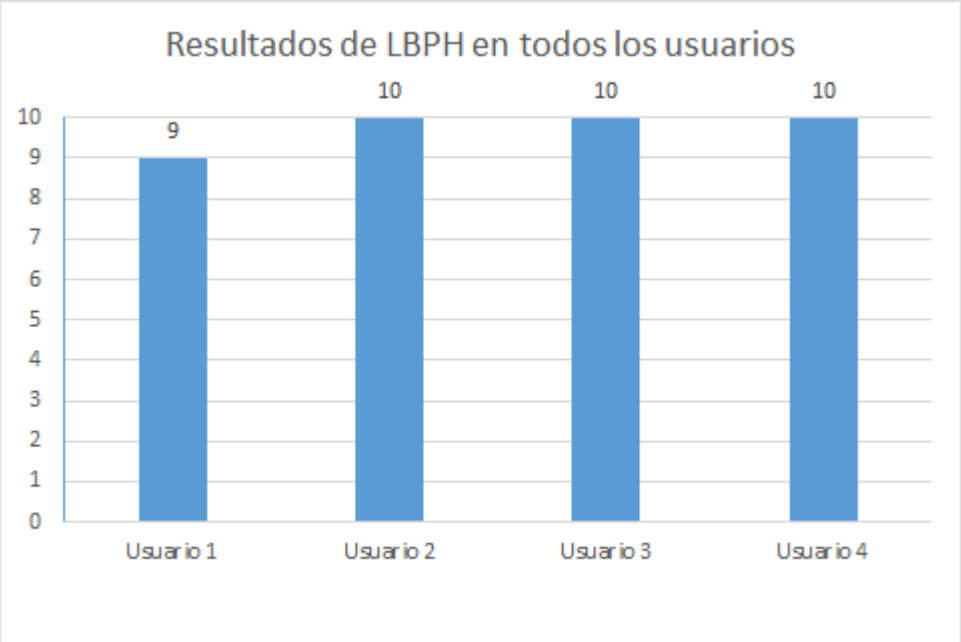

Figura 26. Comparativa de resultados de LBPH en todos los usuarios

Como podemos ver, las pruebas con LBPH arrojan los mejores resultados obtenidos de todos los algoritmos, siendo la media de aciertos muy superior al 90 %, concretamente se sitúa en el 97.5%, habiendo obtenido un 100% de aciertos en 3 de las 4 pruebas.

Su tiempo de reconocimiento ha sido muy bueno en todas las pruebas, siendo el tiempo máximo de 1.379 segundos, muy inferior al obtenido en los otros algoritmos.

6.4 Comparación de los algoritmos

En las Figuras 27 y 28 se comparan los mejores resultados de cada uno de los algoritmos. Como podemos ver, los resultados obtenidos mediante LPBH son mucho mejores que con los Eigenfaces y Fisherfaces, siendo además el tiempo medio de menos de segundo y medio, lo que lo hace unos 4,3 segundos más rápido que Eigenfaces y unos 6,2 segundos más rápido que Fisherfaces (un 76,56 % más rápido y un 82,45 % respectivamente).

Algoritmo	Media de aciertos	Tiempo necesario
Eigenfaces	87.5 %	5.628 s
Fisherfaces	92.5 %	7.515 s
LBPH	97.5 %	1.319 s

Tabla 31. Comparación media de aciertos y de tiempo entre algoritmos

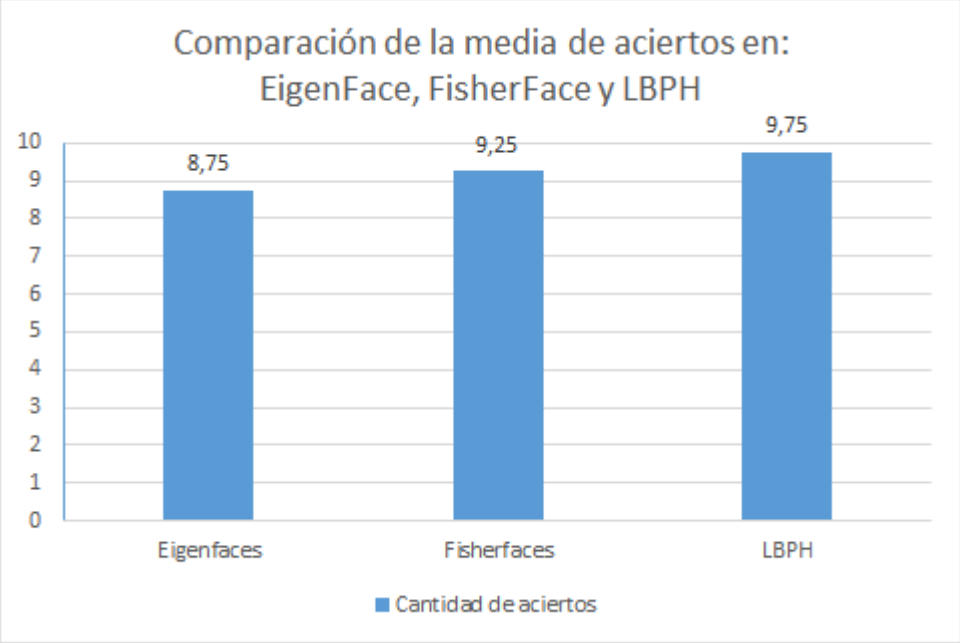

Figura 27. Comparación de los resultados medios de los tres algoritmos

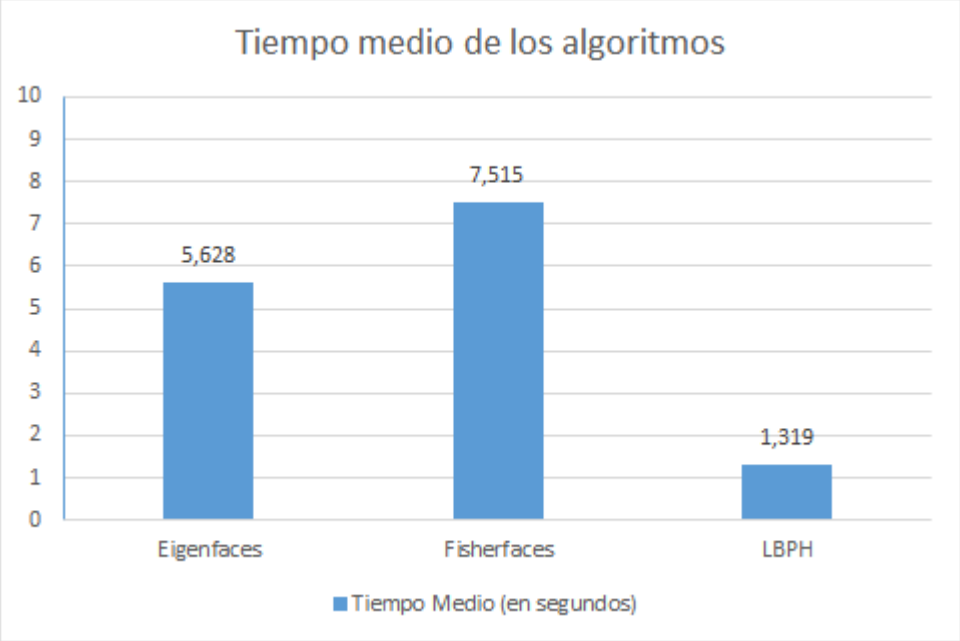

Figura 28. Comparación del tiempo medio en cada algoritmo

Tal y como se muestra en la gráfica anterior, el algoritmo LBPH ha obtenido mejores resultados en nuestro sistema, con unos tiempos de respuesta muy inferiores en comparación con Eigenfaces y Fisherfaces.

Se ha seleccionado el algoritmo LBPH para ser el utilizado en el sistema, debido a su bajo tiempo de resolución del reconocimiento facial, dando un tiempo de respuesta al usuario muy corto además de haber dado unos muy buenos resultados, permitiendo una autenticación mucho más rápida en la red social que se haya seleccionado.

Capítulo 7

Conclusiones y trabajo futuro

7.1 Conclusiones sobre el proyecto

En esta sección se muestran las conclusiones extraídas una vez finalizado el proyecto, así como algunas dificultades experimentadas durante su desarrollo.

7.1.1 Resultado obtenido

En este proyecto se ha desarrollado e implementado una aplicación totalmente funcional, la cual permite la autenticación en redes sociales mediante reconocimiento facial, utilizando para ello la librería Emgu CV.

Esta aplicación permite el entrenamiento por parte del usuario de un sistema de reconocimiento facial para que, posteriormente, dicho sistema pueda reconocerlo mediante imágenes extraídas de su cámara web. Por otra parte, se ha optado por la creación de un fichero cifrado con los credenciales del usuario (dichos credenciales representan el nombre de usuario y la contraseña de la red social en la que se pretenda usar el reconocimiento facial).

La autenticación mediante el reconocimiento facial se realiza mediante una extensión en el navegador Google Chrome, la cual permite agregar la funcionalidad necesaria a nuestro navegador para poder realizar la autenticación. Por comodidad para el usuario, se ha facilitado una página de opciones para guardar dicho fichero y que no sea necesario adjuntarlo cada vez que se realice el reconocimiento.

Paralelamente se han realizado pruebas de funcionamiento de la aplicación bajo unas condiciones externas controladas, arrojando unos buenos resultados.

7.1.2 Dificultades del proyecto

Durante el desarrollo del proyecto se han presentado varias situaciones en las cuales se han concentrado las mayores dificultades.

En el inicio del desarrollo se optó por la versión de 64 bits de la librería EmguCV, la cual provocó diversos problemas de incompatibilidad y errores, que provocaron una ligera pérdida de tiempo en el momento de la implementación y que no se pudieron solventar, cambiando posteriormente a la versión de 32 bits, con la cual se pudieron solucionar esos problemas.

Otra dificultad que se ha encontrado es la forma en la cual se pretendía realizar la autenticación en las distintas redes sociales. En un primer momento se pretendía tener una sección en la aplicación web que permitiera realizar la autenticación en la red social sin necesidad de acceder a ella. Pero debido a las dificultades que se presentaban por políticas de seguridad, era necesario buscar otra forma para realizarlo. Por ello, y como se ha comentado anteriormente, se desarrolló la extensión de Google Chrome.

7.2 Trabajo futuro

En este proyecto se ha realizado una implementación funcional de un sistema de reconocimiento facial, en el cual se han tenido que delimitar ciertas condiciones externas debido a los principales puntos débiles de este sistema biométrico.

Uno de los principales problemas que nos encontramos, como se ha visto en los diferentes algoritmos implementados, es el funcionamiento con condiciones de luz cambiantes debido a que provocan cambios más significativos en la imagen que las diferencias que pueda haber entre las personas. Se han propuesto algunas soluciones basadas en el conocimiento, teniendo en cuenta que todas las caras pertenecen a una misma clase:

- **Métodos heurísticos**: por ejemplo, cuando utilizamos los subespacios de eigenfaces, descartando las componentes principales.
- **Métodos de comparación de imágenes**: se utilizan representaciones apropiadas de la imagen y medidas de distancia.
- **Métodos basados en la clase**: utilizan múltiples imágenes de la misma cara en una pose fija pero bajo diferentes condiciones lumínicas.
- **Métodos basados en el modelo**: utilizan modelos 3D.

Otro problema que nos encontramos es con el ángulo en el que se encuentra el rostro de la persona a la hora de realizar el reconocimiento. Según ciertos estudios, el reconocimiento actúa correctamente hasta los 20°. Una vez superado este ángulo, se empiezan a tener problemas para llevar a cabo un reconocimiento facial. Se han propuesto diferentes soluciones para esta problemática:

- Métodos donde la base de datos incluye imágenes de una persona en diferentes poses.
- Métodos híbridos, donde hay disponibles diferentes imágenes por persona durante el entrenamiento, pero sólo una por persona en el reconocimiento. Es la más utilizada.
- Métodos basados en una única imagen, donde no hay entrenamiento. No es popular.

Por último, el problema de suplantación mediante una fotografía. Es posible que un usuario pueda suplantar la identidad de otro si en el momento del reconocimiento facial muestra a la cámara una fotografía de la persona a suplantar. Actualmente se están estudiando diversos métodos para poder evitar esta problemática:

- Hacer parpadear al usuario para constatar que realmente se trata de una persona la que está delante de la cámara y no de una imagen fija.
- Indicar al usuario que realice algún movimiento con la cabeza.

Además de los posibles avances en materia de reconocimiento facial indicados anteriormente, se identifica como una posible mejora en la seguridad en la aplicación cifrar el fichero con los datos obtenidos del patrón facial (distancia entre los ojos, la nariz, la boca, posición dentro del rostro,…) de cada usuario en lugar de cifrarlo mediante una clave.

Bibliografía

EmguCV [En línea]
Disponible: http://www.emgu.com/wiki/index.php/Main_Page

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In IEEE Conference on Computer Vision and Pattern Recognition, 2001

Detección de Rostros en Imágenes Digitales usando Clasificadores en Cascada; Guevara, M., Echeverry, J., Ardila, W

M.TurkandA.Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 1991.

P.Belhumeur, J.Hespanha, D.Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEEPAMI, 1997.

Timo Ahonen, Abdenour Hadid, and Matti Pietikainen, "Face Recognition with Local Binary Patterns," M. Lecture Notes in Computer Science, Vol. 3021, May. 2004.

Kamran Etemad and Rama Chellappa Discriminant analysis for recognition of human face images, Copyright ©1997 Optical Society of America

Y. Rodriguez and S. Marcel, "Face authentication using adapted local binary pattern histograms," Lecture Notes in Computer Science, vol. 3954, p. 321, 2006.

Microsoft Windows 7 [En línea]
Disponible: http://support.microsoft.com/Product/windows/windows-7

Visual Studio 2013 [En línea]
Disponible: http://www.visualstudio.com/

SQL Server 2014 Express Edition [En línea]
Disponible: http://www.microsoft.com/en-us/server-cloud/products/sql-server/

Microsoft .NET [En línea]
Disponible: http://www.microsoft.com/net/

C# [En línea]
Disponible: http://msdn.microsoft.com/es-es/library/kx37x362.aspx

Google Chrome [En línea]
Disponible: http://www.google.es/chrome/

Microsoft Project 2007 [En línea]
Disponible: http://office.microsoft.com/es-es/project

Microsoft Office 2010 [En línea]
Disponible: http://office.microsoft.com/es-es/

Notepad++ [En línea]
Disponible: http://notepad-plus-plus.org/

EMGU Multiple Face Recognition using PCA and Parallel Optimisation [En línea]
Disponible: http://www.codeproject.com/Articles/261550/EMGU-Multiple-Face-Recognition-using-PCA-and-Paral

Plantilla para realizar la memoria de PFC ofrecida por la Universidad [En línea]
Disponible:
https://www.uc3m.es/portal/page/portal/administracion_campus_leganes_est_cg/proyecto_fin_carrera/Plantilla_PFC.doc

Anexo A

Gestión del proyecto

En este capítulo se detallan aspectos referentes a la gestión del proyecto que incluyen la planificación del trabajo, los medios técnicos utilizados para el desarrollo de todo el proyecto así como el análisis económico del sistema.

1 Planificación del trabajo

En esta sección se detallan la planificación inicial y el desarrollo real del proyecto, junto con un estudio de las desviaciones observadas entre ambos.

2 Planificación inicial

Se muestra a continuación la planificación inicial elaborada para el desarrollo del proyecto. Se pueden observar las distintas fases en las que se ha descompuesto el proyecto junto con las estimaciones de esfuerzo realizadas para cada una de ellas.

En la elaboración de esta planificación se han tenido en cuenta jornadas de 3 horas diarias en días laborables. El motivo de esta planificación se debe a la necesidad de compaginar el desarrollo del proyecto con el trabajo.

Descripción	Duración	Fecha inicio	Fecha fin
Proyecto fin de carrera	91 días	23/04/2014	01/09/2014
Planificación inicial	5 días	23/04/2014	29/04/2014
Estudio del estado del arte	4 días	30/04/2014	06/05/2014
Búsqueda y análisis de herramientas existentes	4 días	30/04/2014	06/05/2014
Análisis	20 días	07/05/2014	03/06/2014
Arquitectura del sistema	10 días	07/05/2014	20/05/2014
Estudio tecnológico	5 días	21/05/2014	27/05/2014
Definición de casos de uso	1 día	28/05/2014	28/05/2014
Definición de requisitos software	4 días	29/05/2014	03/06/2014
Diseño	5 días	04/06/2014	10/06/2014
Diagrama de clases	3 días	04/06/2014	06/06/2014
Diagrama de arquitectura	2 días	09/06/2014	10/06/2014
Implementación	30 días	11/06/2014	22/07/2014
Pruebas	7 días	23/07/2014	01/08/2014
Documentación	20 días	04/08/2014	01/09/2014

Tabla 32. Planificación inicial del proyecto

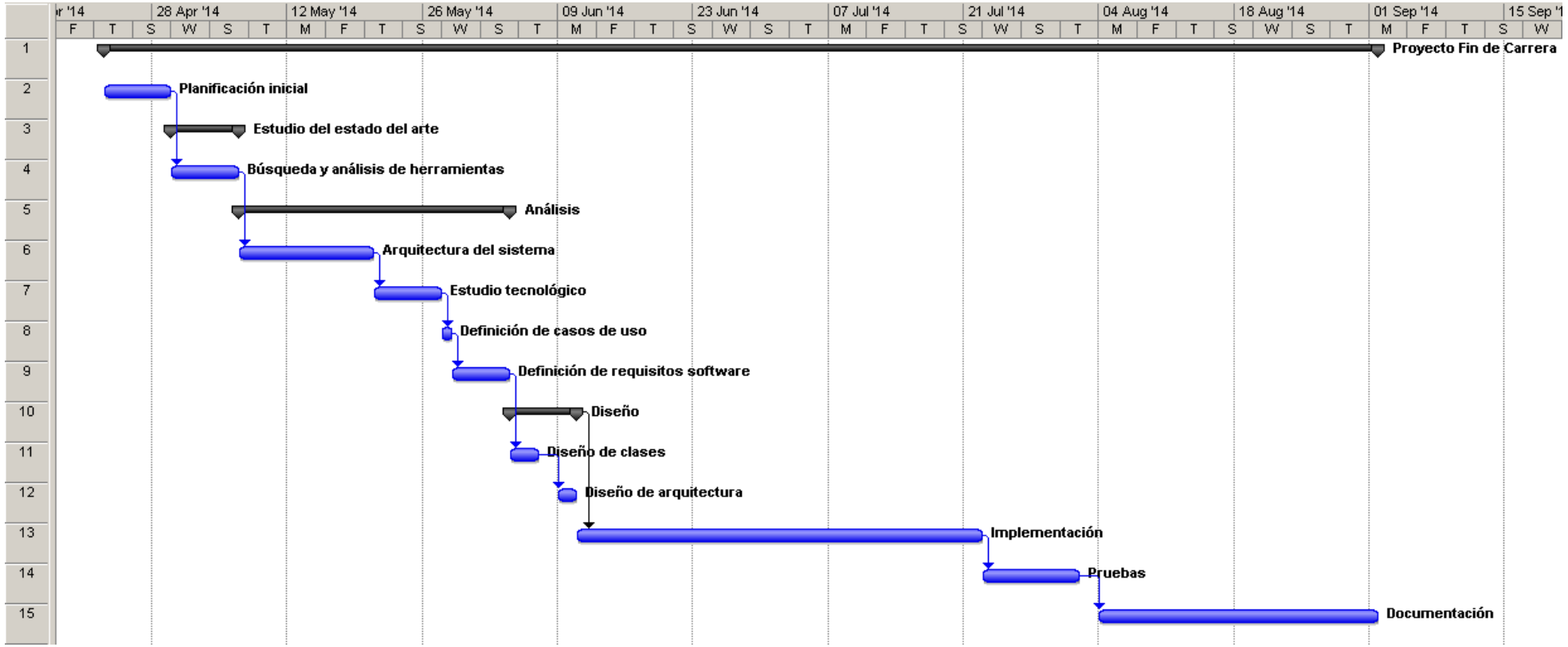

Figura 29. Diagrama de Gantt de la planificación inicial

3 Desarrollo real del proyecto

En esta sección se muestra el desarrollo real del proyecto y se compara con la planificación inicial para estudiar las desviaciones surgidas a lo largo del proyecto.

La Figura 30 muestra el diagrama de Gantt del desarrollo final del proyecto. En ella se puede apreciar que el reparto de tiempos es ligeramente superior en el momento de la implantación.

Descripción	Duración	Fecha inicio	Fecha fin
Proyecto fin de carrera	115 días	23/04/2014	03/10/2014
Planificación inicial	5 días	23/04/2014	29/04/2014
Estudio del estado del arte	4 días	30/04/2014	06/05/2014
Búsqueda y análisis de herramientas existentes	4 días	30/04/2014	21/11/2014
Análisis	23 días	07/05/2014	06/06/2014
Arquitectura del sistema	13 días	07/05/2014	23/05/2014
Estudio tecnológico	5 días	21/05/2014	30/05/2014
Definición de casos de uso	1 día	28/05/2014	02/06/2014
Definición de requisitos software	4 día	29/05/2014	06/06/2014
Diseño	6 días	04/06/2014	16/06/2014
Diagrama de clases	3 días	04/06/2014	11/06/2014
Diagrama de arquitectura	3 días	09/06/2014	16/06/2014
Implementación	40 días	12/06/2014	12/08/2014
Pruebas	7 días	08/08/2014	22/08/2014
Documentación	30 días	20/08/2014	03/10/2014

Tabla 33. Desarrollo real del proyecto

En la Tabla 34 se muestra en detalle la diferencia en la duración entre la planificación inicial y el desarrollo real del proyecto.

La desviación que se puede observar en la etapa de análisis y en la de implementación es debido a que en un principio se pretendía desarrollar la autenticación en la red social directamente en la página web, y posteriormente redirigirlo a la red social con la sesión iniciada.

Debido a políticas de seguridad, ha sido imposible realizar tal acción, por lo que se han tenido que buscar alternativas para permitir la autenticación. Tras varias pruebas de implementación no válidas, se optó por el desarrollo de la extensión de Google

Chrome, la cual permite simular las acciones del usuario directamente en la red social accedida.

Por otro lado, también se puede observar una desviación en la etapa de documentación debido a un error en el cálculo inicial.

Descripción	Planificado	Real	Diferencia	Variación
Proyecto fin de carrera	91 días	115 días	24 días	26,37 %
Planificación inicial	5 días	5 días	0 días	0 %
Estudio del estado del arte	4 días	4 días	0 días	0 %
Búsqueda y análisis de herramientas existentes	4 días	4 días	0 días	0 %
Análisis	20 días	23 días	3 días	15,00 %
Arquitectura del sistema	10 días	13 días	3 días	30,00 %
Estudio tecnológico	5 días	5 días	0 días	0 %
Definición de casos de uso	1 día	1 día	0 días	0 %
Definición de requisitos software	4 días	4 día	0 días	0 %
Diseño	5 días	6 días	1 día	20,00 %
Diagrama de clases	3 días	3 días	0 días	0 %
Diagrama de arquitectura	2 días	3 días	1 día	50,00 %
Implementación	30 días	40 días	10 días	33,33 %
Pruebas	7 días	7 días	0 días	0 %
Documentación	20 días	30 días	10 días	50,00 %

Tabla 34. Análisis de las desviaciones en la planificación

Figura 30. Diagrama de Gantt del desarrollo real

4 Medios técnicos empleados para el proyecto

En esta sección se detallan las herramientas que se han utilizado a lo largo del desarrollo del proyecto. La Tabla muestra las distintas herramientas utilizadas junto con una breve descripción.

Herramienta	Descripción
Microsoft Windows 7	Sistema operativo sobre el que se ha desarrollado este proyecto.
Visual Studio 2013	Entorno de desarrollo de Microsoft utilizado para la implementación del sistema.
SQL Server 2014 Express Edition	Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.
Google Chrome	Navegador de Google necesario para la realización de la extensión de inicio de sesión mediante reconocimiento facial.
Microsoft Office 2010	Herramienta ofimática utilizada para la documentación de este proyecto
Microsoft Project 2007	Aplicación para la administración de proyectos empleada para realizar la planificación y el seguimiento del proyecto.
Notepad++	Editor de texto

Junto con las herramientas de software utilizadas se detalla a continuación el equipo utilizado para el desarrollo del proyecto:

- PC Sobremesa Intel Core i7, 8GB de RAM, Disco duro principal SSD
- Cámara web.

5 Análisis económico del proyecto

En esta sección se incluye un análisis económico del proyecto en la que se muestra una estimación de costes, un presupuesto inicial, un presupuesto para el cliente y el coste real.

5.1 Metodología de estimación de costes

Se ha realizado la estimación de costes teniendo en cuenta los costes directos e indirectos. Los costes directos corresponden a los conceptos que están relacionados directamente con el desarrollo del proyecto, entre los que se encuentran los gastos de mano de obra, los equipos empleados y las herramientas de software. Los costes indirectos son aquellos en los que se incurre en el desarrollo del proyecto pero que no tienen que ver directamente con el desarrollo del mismo (la conexión a internet por ejemplo). Para este proyecto, y de acuerdo con la plantilla proporcionada en la web de la Universidad, los costes indirectos se han calculado como un 20% de los costes directos.

Los gastos de mano de obra se estimarán contando que no se es autónomo y se está empleado para una empresa, los precios de los equipos y del software utilizado son los precios de venta al público.

5.2 Presupuesto inicial

En esta sección se muestra el presupuesto inicial del proyecto, donde se detallan los costes presupuestados y el presupuesto total estimado.

En ninguno de los gastos está reflejado el 21% de IVA a excepción del coste total del proyecto, en el que se especifica claramente que se ha añadido el IVA para su cálculo.

5.2.1 Gastos de personal

La siguiente tabla detalla los gastos de personal. En el cálculo se ha tenido en cuenta un único ingeniero a lo largo de todo el proyecto y, como se ha detallado en este anexo en la sección 2, una dedicación de 3 horas diarias en días laborables durante toda la duración del proyecto.

El coste hombre mes del recurso se ha tomado del modelo proporcionado por la Universidad, si bien se ha considerado que es un importe bruto al que se ha añadido el importe a pagar a la Seguridad Social, 29,9% para el régimen general de la Seguridad Social.

Recurso	Dedicación (hombre/mes)	Coste hombre/mes	Coste bruto	Seg. Social	Coste total
Ingeniero	2,08 h/m	2.694,39 €	5.604,33 €	1.675,70 €	7.280,03 €

Tabla 35. Gastos de personal

Como se puede observar en la tabla anterior, los cálculos muestran una dedicación de poco más de 2 hombres/mes, suponiendo un coste total bruto de 5.604,33 €, a los cuales sumándoles la cuota de la Seguridad Social, 1.675,70 €, asciende a un total de 7.280,03 €.

5.2.2 Gastos de equipos

En esta sección se detallan los gastos relacionados con el equipo utilizado a lo largo del proyecto. Entre estos equipos se encuentra un ordenador de sobremesa y la cámara web. La Tabla muestra los costes imputables para cada equipo así como de los datos utilizados para calcularlos: porcentaje de uso dedicado, coste y periodo de depreciación. El periodo de depreciación es el utilizado en el ámbito de la Administración General.

Descripción	P.V.P	Dedicación	Periodo depreciación	Coste final
PC Sobremesa	957 €	3,03 meses	36 meses	80,55 €
Cámara web	15,72 €	3,03 meses	36 meses	1,32 €
			Coste total	81,87 €

Tabla 36. Gastos de equipos

5.2.3 Gastos de software

En esta sección se muestran los gastos relacionados con el software utilizado en el proyecto, siempre que no sea gratuito. Esta tabla solo muestra el software mencionado en la sección 4 de este anexo que no sea gratuito.

Descripción	P.V.P	Dedicación	Periodo depreciación	Coste final
Windows 7	106,61 €	3,03 meses	36 meses	8,97 €
Visual Studio 2013	510,34 €	3,03 meses	36 meses	42,95 €
Microsoft Office 2010	78,21 €	3,03 meses	36 meses	6,58 €
Microsoft Project 2007	607,51 €	3,03 meses	36 meses	51,13 €
			Coste total	109,63€

Tabla 37. Gastos de software

5.2.4 Costes directos

En esta sección se muestra la suma de los costes directos asociados a este proyecto calculados en los apartados anteriores: gastos de personal, gastos de equipos y gastos de software. La Tabla muestra la suma de estos conceptos dando como resultado el total de costes directos del proyecto.

Concepto	Coste
Gastos de personal	7.280,03 €
Gastos de equipos	81,87 €
Gastos de software	109,63 €
Costes directos	**7.471,53 €**

Tabla 38. Costes directos

5.2.5 Costes indirectos

Los costes indirectos se han calculado de acuerdo a lo visto en la sección 3.1 de este anexo como el 20% de los costes directos.

De esta forma, y acorde a los resultados obtenidos en el apartado anterior, los costes indirectos del proyecto son 1.494,31 €.

5.2.6 Estimación de costes

A continuación se muestra la Tabla 39, en la cual se detalla la estimación de costes del proyecto con los cálculos realizados en los apartados anteriores. En ella se detalla la suma de los costes directos e indirectos obtenidos en los apartados anteriores sin IVA, a los que se les sumará el IVA en el total.

Concepto	Coste
Gastos de personal	7.280,03 €
Gastos de equipos	81,87 €
Gastos de software	109,63 €
Gastos directos	**7.471,53 €**
Gastos indirectos	**1.494,31 €**
Total gastos sin IVA	**8.965,84 €**
IVA (21%)	**1.882,83 €**
Total gastos con IVA	**10.848,67 €**

Tabla 39. Estimación de costes

5.3 Presupuesto para el cliente

En esta sección se muestra el presupuesto que se presenta al cliente, agrupando los gastos vistos en las secciones anteriores y añadiendo un porcentaje de riesgo, usado para hacer frente a imprevistos. Además, se añade un porcentaje que representará el beneficio esperado por el desarrollo del proyecto.

El porcentaje de riesgos ha sido definido en un 10%. Se ha tomado dicho valor de acuerdo a los valores que se han utilizado en otros proyectos.

El porcentaje de beneficio se ha definido en un 15%. Se ha decidido utilizar este valor porque, al igual que pasaba con el porcentaje de riesgo, entraba dentro del rango de valores vistos en otros proyectos. Este beneficio será calculado una vez incluido el porcentaje de riesgo.

La siguiente tabla muestra el presupuesto a entregar al cliente:

Concepto	Coste
Gastos directos	7.471,53 €
Gastos de personal	7.280,03 €
Gastos de equipos	81,87 €
Gastos de software	109,63 €
Gastos indirectos	1.494,31 €
Total gastos (Sin riesgo)	**8.965,84 €**
Riesgo (10%)	896,58 €
Total gastos (Sin beneficio)	**9.862,42 €**
Beneficios (15%)	1.479,36 €
Total (Sin IVA)	**11.341,78 €**
IVA (21%)	2.381,77 €
TOTAL	**13.723,55 €**

Tabla 40. Presupuesto para el cliente

5.4 Coste final y análisis de la desviación

En este apartado se muestra el coste final que ha tenido el proyecto, comparándose con el coste estimado inicialmente. El número de días requerido para el desarrollo del proyecto ha sufrido una desviación en comparación con la estimación, por lo que cabe esperar una desviación en el proyecto en los costes de personal, de equipos y de software.

La Tabla 41 muestra la comparación entre el presupuesto inicial y el coste final del proyecto, indicando las variaciones en cada concepto y en el total. Se ha utilizado el

mismo proceso para el cálculo del coste real, modificando los meses de desarrollo del proyecto, así como el número de horas hombre.

Concepto	Coste presupuestado	Coste real	Variación
Gastos Directos	7.471,53 €	9.443,60 €	1.972,07 €
Gastos personal	7.280,03 €	9.201,53 €	1.921,50 €
Gastos equipos	81,87 €	103,48 €	21,61 €
Gastos software	109,63 €	138,59 €	28,96 €
Gastos indirectos	1.494,31 €	1.888,72 €	394,41 €
Total	**8.965,84 €**	**11.337,31 €**	**2.366,48 €**

Tabla 41. Coste real del proyecto en comparación con el presupuestado

Como se puede apreciar existe un aumento de 2.366,48 €, que representa algo más de un 26 % en el coste total en relación con el coste presupuestado. Esta diferencia no puede repercutir en el cliente, por lo que primero procederemos a utilizar el porcentaje riesgo, y todavía obtenemos unas pérdidas de 1.469,90 €. A continuación podemos descontar estas pérdidas de nuestro beneficio presupuestado, 1.479,36 €, quedando un beneficio final de 9,46 €.

Debido al aumento en el tiempo necesario para el desarrollo del proyecto, se podría esperar un aumento en los gastos, como así ha pasado.

El resultado tan ajustado de este proyecto se debe principalmente al aumento del tiempo necesario para el desarrollo del proyecto, debido a los problemas detallados en este mismo anexo en el apartado 3, así como la variación del tiempo disponible que se ha podido dedicar al mismo en diversos momentos.

Anexo B

Manual de usuario

1 Manual de usuario

A continuación se muestra un manual de usuario sobre cómo se usa la aplicación, mostrando desde el registro hasta la autenticación en la red social de Facebook (la autenticación en Twitter es análoga a la de Facebook, por lo que no se mostrará).

Antes de comenzar, instalaremos la extensión de Google Chrome para que esté disponible la autenticación en la red social de Facebook.

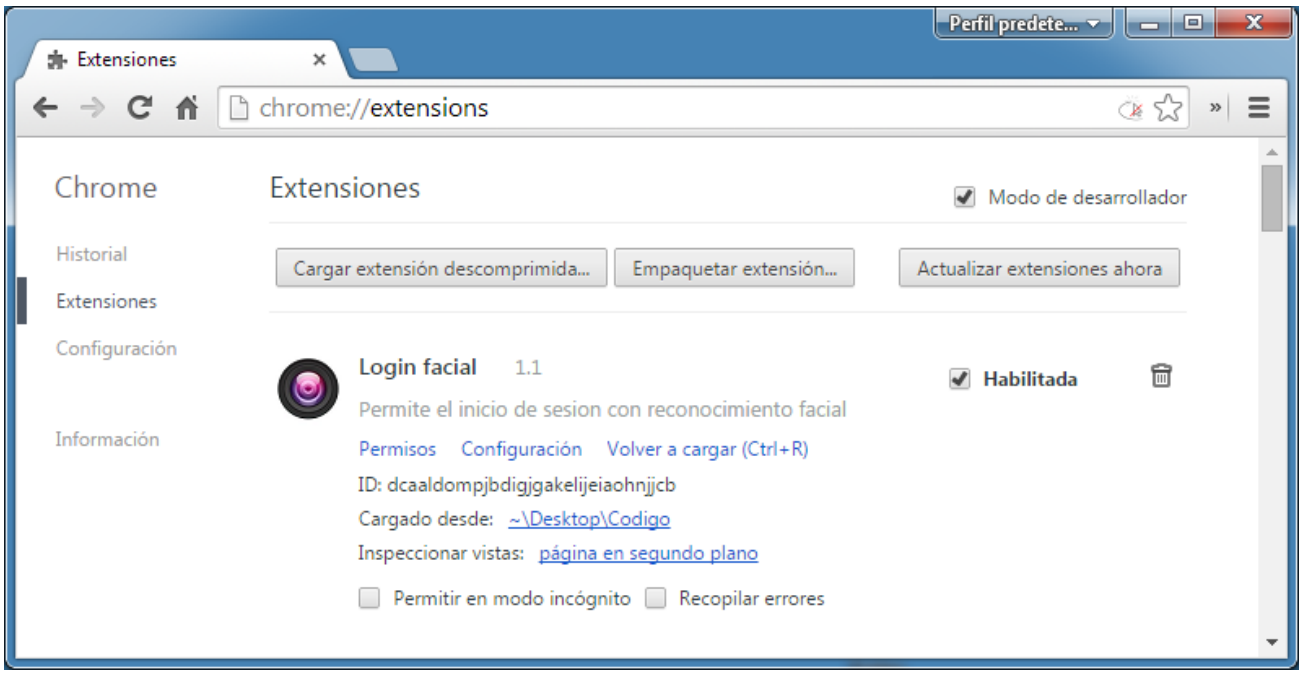

Figura 31. Instalación de la extensión en Google Chrome

Para empezar, el usuario deberá registrarse en la aplicación, para que los rostros entrenados sean enlazados con su usuario y para poder administrar las imágenes.

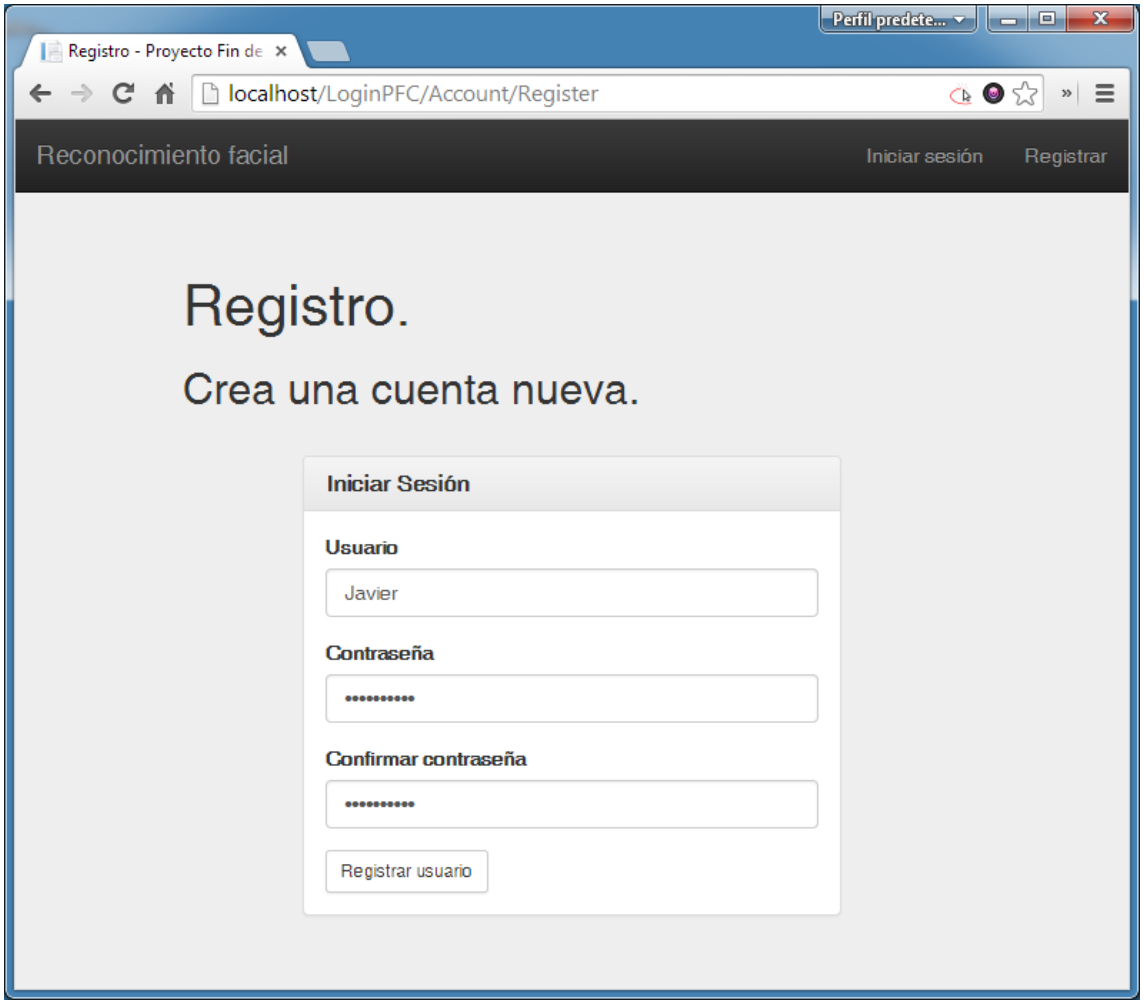

Figura 32. Registrando una nueva cuenta de usuario

A continuación, se nos dirigirá a la pantalla de entrenamiento de la aplicación, en la cual nos deberemos sacar más o menos diez fotografías en las cuales nos reconozca el sistema. Nos irá mostrando en la parte inferior los rostros reconocidos en las fotografías que nos estamos sacando pulsando el botón *Capturar*. Una vez que tenemos el número de fotografías deseado, pulsamos *Guardar imágenes* para añadirlas al sistema asociadas a nuestro usuario (ver Figura 34).

Tras el entrenamiento (o antes, según se prefiera), procedemos a crear el fichero de credenciales accediendo en el menú superior en *Crear archivo*. Únicamente se debe introducir en las cajas de texto el usuario y la contraseña que se usa en la red social para la que se quiere utilizar dichos credenciales. Una vez hecho esto, y pulsando en *Guardar credencial*, se descarga automáticamente el fichero cifrado que usaremos posteriormente (ver Figura 33).

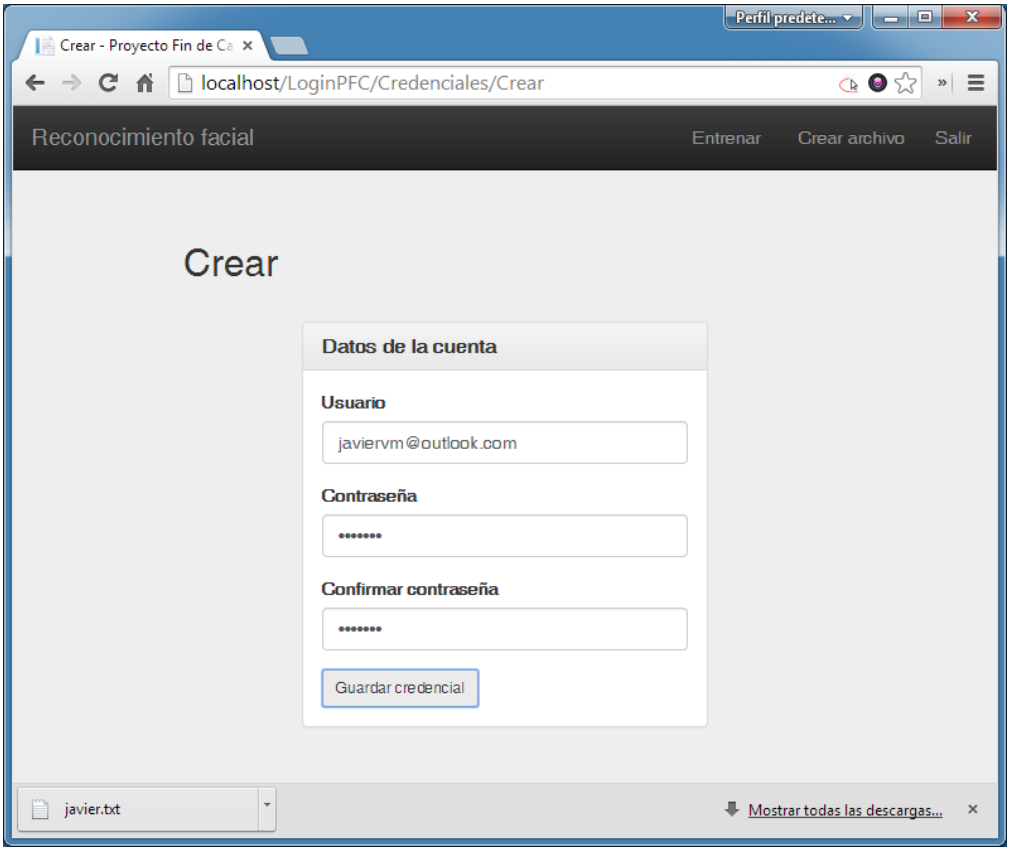

Figura 33. Crear fichero de credenciales

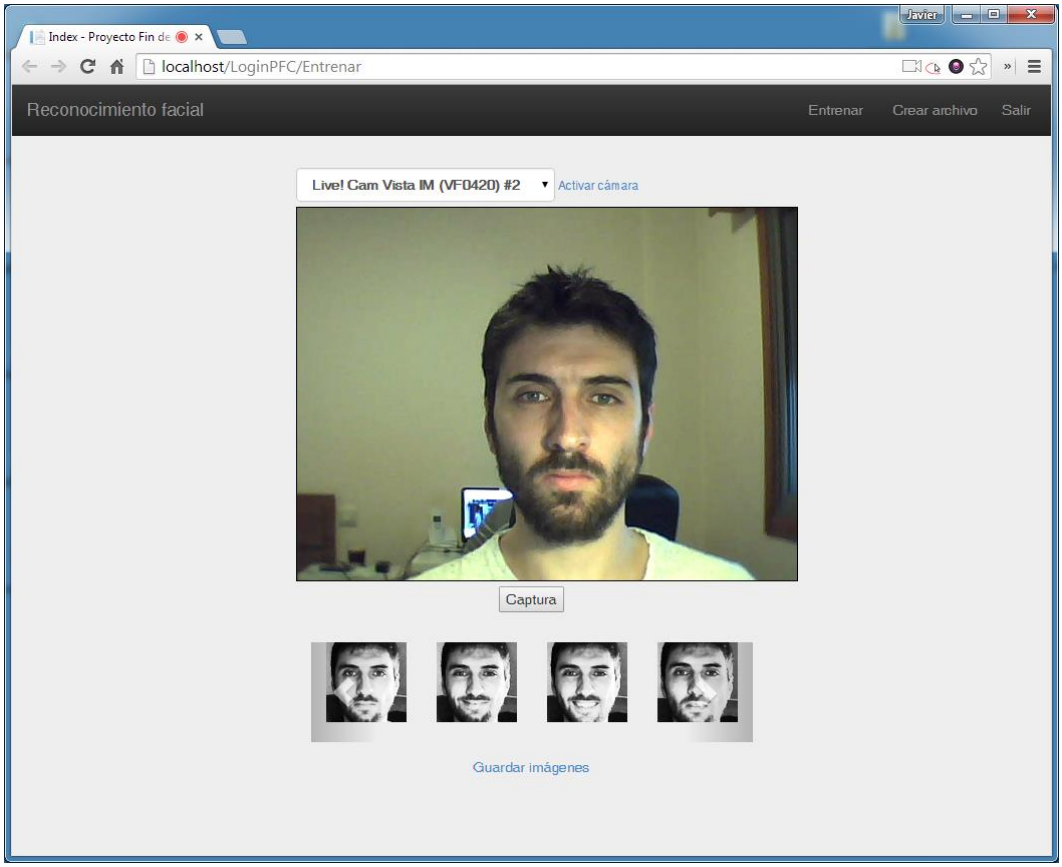

Figura 34. Página de entrenamiento del sistema

Tras esto, podemos acceder a la página de configuración de la extensión para guardar en las opciones el fichero descargado, ya que esto nos ahorrará tener que adjuntarlo en todas las ocasiones en las que nos autentiquemos (ver Figura 35).

Tras la configuración, accedemos a la página de Facebook para realizar la autenticación. Si la extensión está instalada correctamente, en la barra de direcciones se mostrará el icono de la extensión, indicando que la autenticación mediante reconocimiento facial está disponible en esa página, y en la parte superior de la página web se mostrará un enlace para activar la extensión (tal y como se muestra en la Figura 14 del Capítulo 5.3).

Figura 35. Página de configuración de la extensión

Activamos la extensión, permitimos el acceso a la cámara web y, una vez situados de forma correcta, pulsamos en el botón de *Iniciar sesión*. Si somos reconocidos, en los campos de texto habilitados para ello, se inserta el usuario y la contraseña obtenidos del fichero cifrado y posteriormente se inicia sesión mostrando la página principal de nuestra cuenta de Facebook (ver Figura 36 y 37).

Anexo B

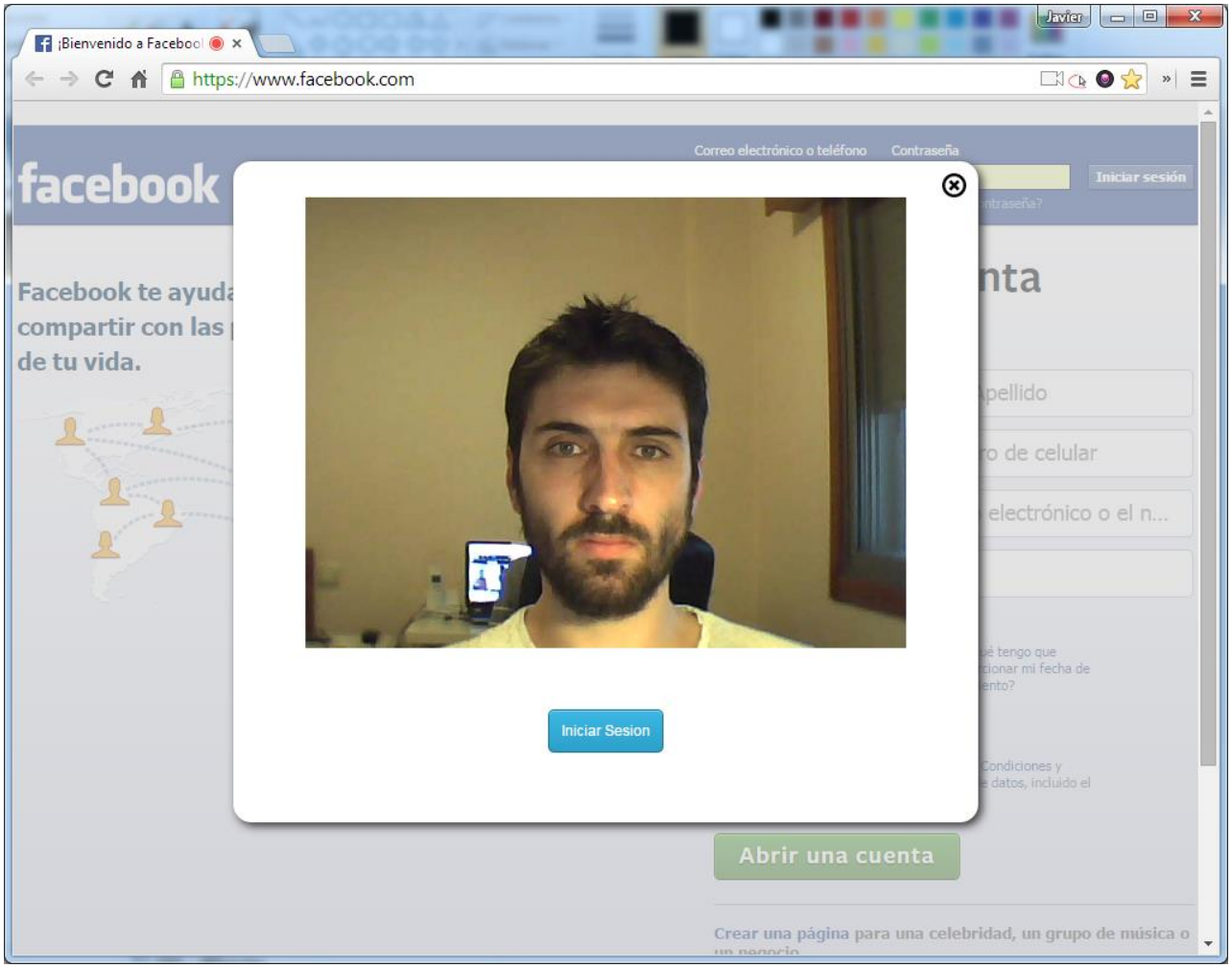

Figura 36. Extensión de Google Chrome en funcionamiento

Anexo B

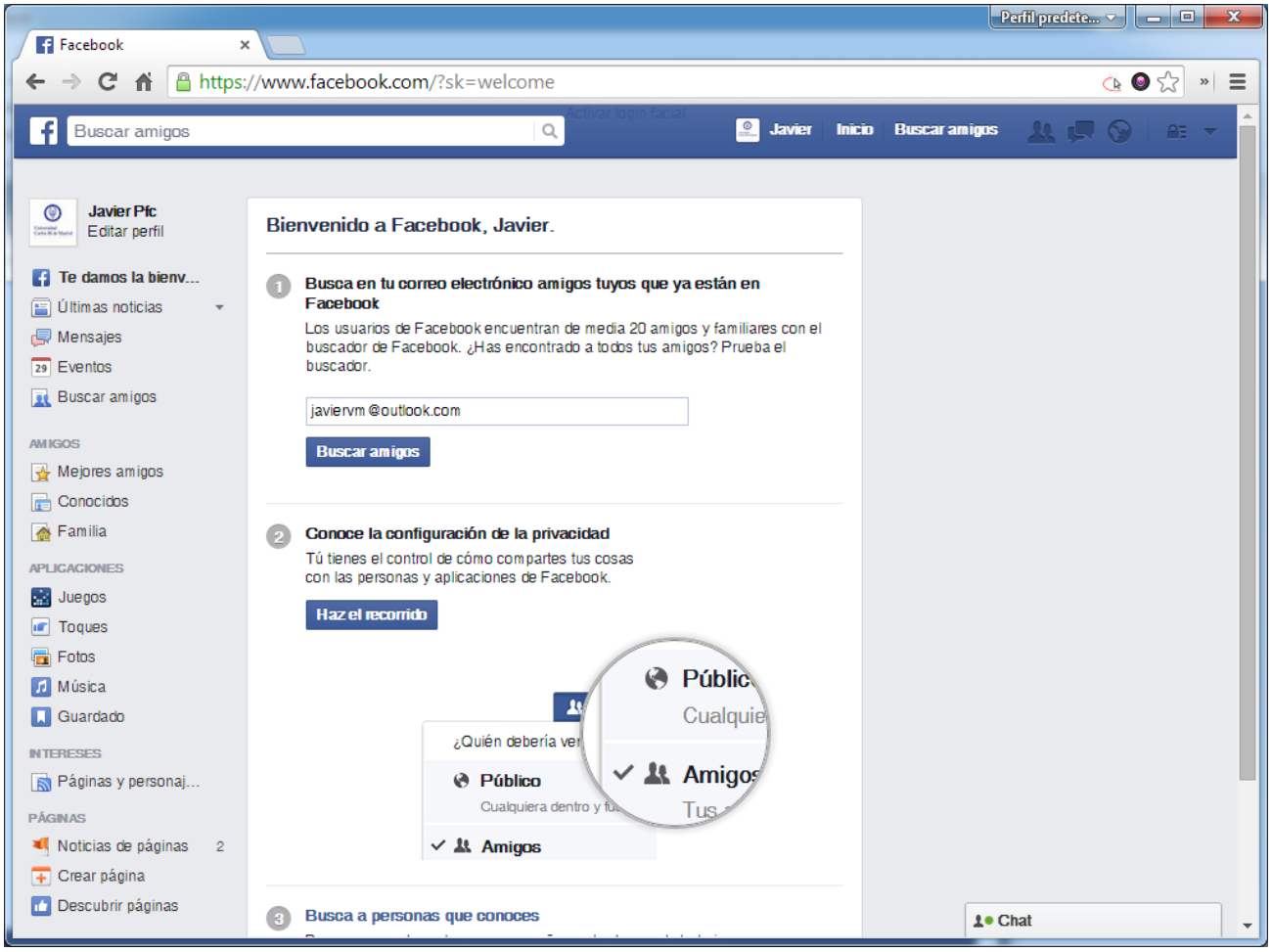

Figura 37. Página principal de Facebook

Anexo C

Plantillas

1. Plantilla para la definición de casos de uso

A continuación se muestra la plantilla que se usará para la definición de los casos de uso. En la parte superior aparece el identificador del caso de uso, posteriormente se muestra el nombre del caso de uso, su autor, la descripción, que autores participan, las precondiciones (condiciones que se deben cumplir al iniciar el caso de uso), las post-condiciones (el estado del sistema una vez terminado el caso de uso), el flujo normal que sigue el caso de uso y un flujo alternativo si se presenta algún problema.

Identificador:	
Nombre	
Autor	
Descripción	
Actores	
Precondiciones	
Post-condiciones	
Flujo normal	
Flujo alternativo	

Tabla 42. Plantilla para la definición de casos de uso

2. Plantilla para la especificación de requisitos de software

A continuación se muestra la plantilla que se usará para la definición de los requisitos de software. En ella se especifica un identificador único para el requisito, su nombre, la descripción del requisito, su estabilidad (indica la probabilidad de que un requisito no cambie a lo largo del desarrollo) y su prioridad (la importancia del requisito dentro del proyecto).

Requisitos de software				
Tipo:				
Id	Nombre	Descripción	Estabilidad	Prioridad

Tabla 43. Plantilla para la especificación de requisitos software

3. Plantilla para la especificación de pruebas de aceptación

A continuación se muestra la plantilla que se usará para la especificación de pruebas de aceptación. En ella se especifica el identificador de la prueba, los elementos probados, los cuales se corresponden con los identificadores de los requisitos a probar, los datos que recibe la prueba y la salida esperada en la misma.

Pruebas de aceptación			
Id	Elementos probados	Entrada	Salida

Tabla 44. Plantilla para la especificación de pruebas de aceptación