Universidad
Carlos III de Madrid
www.uc3m.es

Trabajo Fin de Grado:

SISTEMA DE ACCESO A SERVICIOS WEB MEDIANTE PLATAFORMA BIOAPI C#

GRADO EN INGENIERÍA DE SISTEMAS DE COMUNICACIONES

Autor: Sergio Delgado Menéndez

Tutor: Raúl Sánchez Reíllo

Director: Jaime de Uriarte

Leganés, 24 de Junio de 2013

Trabajo Fin de Grado:

GRADO EN INGENIERÍA DE SISTEMAS DE COMUNICACIONES

Sergio Delgado Menéndez

Agradecimientos

La finalización del Trabajo Fin de Grado supone un gran punto de inflexión en el que merece la pena echar la vista atrás y darse cuenta de todo lo que ha pasado. Supone la consecución de un objetivo marcado hace muchos años atrás, la conclusión del trabajo y sacrificio de todos estos años.

Llegado a este momento, puedo afirmar que estoy muy orgulloso de haber formado parte de la Universidad Carlos III, estando seguro de que acerté a la hora de decirme por el plan Bolonia que por aquel entonces era una novedad y solo ofertaba esta Universidad.

Para llegar hasta aquí, es necesario contar con gente que te apoye y te ayude constantemente, por esto y por muchas más cosas tengo que dar las gracias a mis padres, Clemente y Mercedes, y a mi hermana María, así como al resto de mi familia.

También gracias a mi tutor Raúl por sus consejos durante todo el trabajo, su disponibilidad y el buen trato recibido siempre han sido de gran valor, al director del trabajo, Jaime, con él aprendí los primeros pasos para desarrollar en C#, sus recomendaciones fueron de gran ayuda. Por supuesto no me olvidaré de la gente del laboratorio (José Francisco y Eugenio), gracias a sus consejos conseguí conocer el mundo de la biometría y sobre todo resolver aquellos pequeños problemas que me surgieron durante el desarrollo del proyecto.

Gracias a todos mis amigos, siempre me han animado cuando más lo necesitaba y mención también a los compañeros con los que me he cruzado a lo largo de todos estos años. Y, por supuesto, gracias a mi novia Jennifer por haber compartido estos años de tanto sufrimiento y a la vez de tantas alegrías, espero haberte ayudado tanto cómo tú lo has hecho.

Tabla de contenido

Índice de acrónimos

ALM Application Lifecycle Management (Dirección de ciclo de vida de aplicación).

ANSI American National Standards Institute (Instituto de Normas Americano Aacional).

ASP Active Server Pages (Servidor Activo Pagina).

BIAS Biometric Identity Assurance Services (Servicios de Aseguramiento de Identidad Biométricos).

CCD Un charge-coupled device o CCD (dispositivo de carga acoplada).

CMOS Complementary metal oxide semiconductor (Semiconductor complementario metálico de óxido).

EEUU United states (Estados Unidos).

ECMA European Computer Manufacturers Association (Asociación de Fabricantes de Ordenador Europea).

ECTS European Credit Transfer and Accumulation System (Sistema Europeo de Transferencia y Acumulación de Créditos).

GPL General public license (Licencia pública general).

IDE Integrated Development Environment (Entorno de Desarrollo Integrado).

IEC International Electrotechnical Commission (Comisión Internacional Electrotécnica).

INCITS International Committee for Information Technology Standards (Comité Internacional para tecnología de la información Normas).

iOS Operative system (Sistema operativo).

ISO International Organization for Standardization (Organización Internacional para Estandarización).

LGPL Lesser General Public License (Licencia pública general reducida).

MM	Mathematical Morphology (Morfología matemática).
MS-DOS	MicroSoft Disk Operating System (Sistema operativo de Disco de Microsoft).
OASIS	Advancing open standards for the information society (Avance de normas abiertas para la sociedad de la información).
OEM	Original Equipment Manufacturer (Fabricante de Equipo Original).
PC	Personal computer (Ordenador personal).
PCMCIA	Personal Computer Memory Card International Association (Asociación Internacional de Tarjetas de Memoria de Ordenador personal).
PDA	Personal digital assistant (Ayudante personal digital).
ROI	Region of intérest (Región de interés).
SDK	Software development kit (Kit de desarrollo software).
SOA	Service-Oriented Architecture (Arquitectura orientada a servicios).
TC	Technical committee (Comité Técnico).
USB	Universal Serial Bus (Bus Universal en Serie).
WPF	Windows Presentation Foundation (Fundación de presentación de Windows).

Índice de figuras

Índice de tablas

Resumen

A lo largo de esta memoria se va a presentar un Trabajo Fin de Grado en el cuál se desarrolla una aplicación cliente-servidor que tiene como objetivo controlar el acceso de individuos a diversas aplicaciones. En este caso se ha simulado una cuenta bancaria.

El acceso a la aplicación se controlará a partir de un reconocimiento de huella dactilar. El procedimiento para diferenciar una huella dactilar de otra se basa en un conjunto de líneas genéricas denominadas crestas. Estas son las partes de la piel donde esta se eleva sobre las zonas más bajas denominadas valles y cuya anchura oscila entre las 2 y 5 décimas de milímetro.

El funcionamiento de la aplicación se basa en el estándar de BIAS, concretamente en el nivel 4, a través del cual se ha conseguido establecer un intercambio de información seguro entre el cliente y el servidor, de manera que los datos tanto biométricos como biográficos recogidos puedan ser enviados y almacenados en el servidor para posteriores comprobaciones. La funcionalidad principal consistía en conseguir implementar las funciones de registro, verificación e identificación. No obstante se ha implementado el diseño de una cuenta bancaria, simulando los ingresos y extracciones de dinero que puede realizar el cliente, así como las peticiones o cambios de información en la cuenta que el cliente pueda solicitar.

Según el estándar de BIAS, este tipo de aplicaciones pueden desarrollarse para dos ámbitos distintos. En este caso, el que más interesa para una aplicación de simulación de una cuenta bancaria seria el modelo centrado en la persona.

En este modelo cuando se recibe un dato biométrico o bien se añade (si no existe) o reemplaza el anterior (si ya existe). Así, el sistema almacenaría unos datos biométricos iniciales que se recogen del sujeto tras el registro. Posteriormente cuando el usuario intente acceder al sistema o se reciban nuevos datos biográficos, los datos iniciales serán remplazados, de manera que el sistema siempre contenga la muestra más reciente del individuo. Este tipo de modelo es utilizado principalmente en sistemas de control de acceso, debido principalmente a que no necesita un gran espacio para almacenar los datos, ya que solo conserva una muestra por usuario.

No obstante como se comentará más adelante en el apartado que trata el estándar de BIAS en profundidad, la aplicación ha sido diseñada de modo que pueda funcionar tanto con ese modelo como con el centrado en el número de visita (se explica en el citado punto).

Por último se añadirá que un trabajo como el realizado en este documento es una fuente de posibles líneas de trabajo futuro, tanto en la mejora de la aplicación, como en el servidor, incluyendo mejoras de rendimiento conjunto.

Abstract

Along this document a Bachelor Thesis will be described, in which a client-server application is developed with the objective to control the access of individuals into different applications. In particular, the scenario emulated has been the access to a bank account.

The access to the application will be controlled by the verification of a fingerprint. The process to differentiate among fingerprints is based on the analysis of the fingerprint ridges. Fingerprint ridges are the parts of the skin that are elevated under the lower zones called valleys. The separation between ridges is about 0,2 – 0,5 mm.

The performance of the application is based in the BIAS standard (under final stages of development to generate ISO/IEC 30108), particularly in the 4^{th} level, through which a secure exchange of information between the client and the server can be established. Therefore, the biometric and biographic data collected can be sent and saved in the server for later comparisons. The principal functionality consists of obtaining implementations of the functions for the operations of registering, verification and identification. In addition to the implementation of BIAS, in order to illustrate the functionality, the design of a bank account, has been done, simulating the incoming and withdrawals of money that the client can make, as well as the petitions or changes of information in the account that the client may require.

According to BIAS standard, applications can be developed for two or more different scopes. In this case, the one used was the person centric model, that it has been considered as the most interest for the simulated application.

In this model, biometric data can be received, added, removed or replaced. In this way the system would save the initial biometric data that are taken from the subject after the enrollment. After this, when the user tries to access the system the new verified data will replaced the older one, so that the system always holds the last valid information. This kind of model is mainly used in access system control due to the fact that it is not necessary a large memory to save data, because it only keeps one sample per user.

However as this project will show,, the application has been designed in order to not only work in the person-centric model but also in the encounter –centric model.

The work developed in this Bachelor Thesis is part of a research line that provides future works lines, including improvement of the application and the server to optimize performance.

1. Introducción

En esta primera parte se va a dar a conocer una idea general de lo que el lector se va a encontrar a lo largo de este documento.

A lo largo de esta memoria se va a presentar un Trabajo Fin de Grado en el cuál se desarrolla una aplicación cliente-servidor que tiene como objetivo controlar el acceso de individuos a diversas aplicaciones. En este caso se ha simulado una cuenta bancaria.

El funcionamiento de la aplicación se basa en el estándar BIAS, concretamente en el nivel 4 de conformidad de este estándar, a través del cual se ha conseguido establecer un intercambio de información seguro entre el cliente y el servidor, de manera que los datos recogidos tanto biométricos como biográficos, puedan ser enviados y almacenados en el servidor para posteriores comprobaciones. La funcionalidad principal consistía en conseguir implementar las funciones de registro, verificación e identificación. No obstante se ha implementado el diseño de una cuenta bancaria, simulando los ingresos y extracciones de dinero que puede realizar el cliente.

Para la realización de un trabajo fin de grado, es muy importante conocer desde el principio los objetivos que se quieren alcanzar y la motivación con la que se cuenta.

1.1.Motivación

La motivación principal es tanto el aprendizaje de las técnicas biométricas usadas en la actualidad, como el aprendizaje de un nuevo lenguaje de programación (C#) que gana cada vez más peso en un mundo en constante evolución. La elección de desarrollar este tipo de aplicación viene determinada por el convencimiento de que la biometría será la clave del futuro en cuanto al control de acceso, ya sea para aplicaciones digitales o para barreras físicas como la simple puerta de casa.

Esto ha sido posible gracias al espectacular avance que se ha vivido en el siglo 21^{th} en el desarrollo de aplicaciones de seguridad basadas en biometría, ya que las tradicionales firmas o huellas dactilares impresas se han visto sustituidas por escaneos de iris y retina, procedimientos para reconocimiento de voz, etc.

Además es importante destacar que la creciente industria de la telefonía móvil, en la que se ha conseguido que los móviles actuales superen a los ordenadores de inicios del siglo XXI, es una importante apuesta de futuro en cuanto al uso de técnicas de biometría, ya que al igual que se ha conseguido integrar una cámara de fotos o un dispositivo GPS en el móvil, no sería nada raro ver en unos años un dispositivo de lectura de huella o de captura de iris en el móvil, con el que acceder a diversas cuentas, a las que hoy día se accede con una clave, en ocasiones fácilmente descifrable.

No obstante en la actualidad las aplicaciones biométricas van destinadas a los ordenadores, ya que estos poseen unas capacidades tanto de procesador como de autonomía mucho mayores que cualquier terminal móvil, por lo que las aplicaciones desarrolladas en estos, pueden contar con un uso de memoria, en ocasiones obligatorio debido a los potentes algoritmos a los que son sometidas estas operaciones, mucho mayor del que dispondrían en cualquier dispositivo móvil actual.

La motivación no es realizar una aplicación comercial, siendo una aplicación totalmente funcional, pero dejando las bases para que posteriormente otro alumno continúe el trabajo, convirtiéndola finalmente en una aplicación que pueda competir a nivel comercial con otras aplicaciones de su ámbito.

Con este trabajo se podrán ver las líneas futuras de un estándar que sentará las bases del intercambio de datos biométricos en los próximos años, un estándar que apenas ha dado sus primeros pasos en este mundo tan tecnológico.

A lo largo de este trabajo habrá que enfrentarse a problemas reales semejantes a los que se puede encontrar un ingeniero a lo largo de su vida laboral. Además permitirá aplicar lo aprendido a lo largo de los años de formación universitaria.

1.2.Objetivos

Los objetivos que se quieren alcanzar con este trabajo fin de grado se han planificado teniendo en cuenta:

- Limitaciones de tiempo. El Trabajo Fin de Grado es una asignatura de 12 créditos ECTS, lo que supone que la aplicación y la documentación de la misma, debe estar presupuestada en unas 300 horas de trabajo. Esto implica una duración de un cuatrimestre con una dedicación parcial de 20h/semana.
- Las capacidades del alumno. Se trata de un entorno de desarrollo totalmente nuevo, por lo que el trabajo cuenta con una complejidad adicional debida al aprendizaje de dicho entorno.
- Requisitos de la aplicación. Cuanto más generalista sea, será más accesible, consiguiendo por tanto mayor penetración de mercado.

El desarrollo partiendo de cero de este tipo de aplicaciones, aporta al alumno nuevos conocimientos y situaciones semejantes a las que podemos encontrar en el mundo empresarial o de investigación. Por lo tanto, el objetivo principal de un trabajo fin de grado debe ser el de aprender y preparar al alumno a la vida laboral.

Centrándose en la aplicación a desarrollar, el objetivo es el de conseguir registrar los datos biométricos (basados en la huella dactilar) de un individuo, para que posteriormente pueda acceder a dicha aplicación a través de dos procesos diferentes, como son el de verificación (comprobación única entre los datos del individuo y la plantilla indicada) e identificación (comprobación múltiple entre los datos del individuo y todas las plantillas del servidor).

Introduction

This first part is going to provide an overview of what the reader will find throughout this document.

This report describes the Bachelor Thesis developed, which is mainly a client-server application that aims to control the access of individuals to various operations. This has been illustrated by accessing a simulated bank account.

The operation of the application is based on the BIAS standard, particularly at level 4, through which has succeeded in establishing a secure exchange of information between client and server, so that the biometric data and biographic data collected, can be sent and stored on the server for further checks. The main functionality was to be able to implement the functions of registration, verification and identification. However it was implemented a bank account, simulating revenue and cash withdrawals.

Before describing the Bachelor Thesis developed, it is very important to know the initial objectives that were set and the motivation to perform this task.

Motivation

The main motivation is learning both biometric techniques used nowadays, as well as learning a new programming language (C #) that is gaining more weight in a changing world. The choice of developing this type of application is determined by the conviction that biometrics will be the future key in access control.

This has been made possible by the impressive progress that has occurred in the 21st century with the development of applications based on biometric security, as traditional printed signatures or fingerprints have been replaced with tablets, fingerprint sensor, iris and retina scans, procedures for recognition voice, etc.

It is also important to note that the growing mobile phone industry, is an important investment in the future then the use of biometric techniques because just as it has managed to integrate a camera or GPS device in the phone, it would not be unusual to see in a few years reading device capture fingerprint or iris sensor in the mobile, with which access various accounts, which today is accessed with a key, sometimes easily decipherable.

However currently biometric applications are mainly intended for computers, where the processing and power consumption capabilities are better than any mobile terminal, so that applications developed in these, can have a memory usage sometimes due to the large binding algorithms which are subjected to these operations, much greater than that would be available on any mobile device today.

The motivation is to provide a fully functional application, as to leave for the future if another student continue the work, making an application that can compete commercially with other applications its area.

Also, with this work the future lines for the BIAS standard could be seen, as to check if it will be positively used for exchanging biometric data in all kind of applications in the coming years.

Throughout this work the student had to face problems as the ones that an engineer could find throughout his/her working life. Finally, this Bachelor Thesis applies what the student has learned along the years of university education.

Objectives

The objectives to be achieved with this Bachelor Thesis are planned taking into account:

- Time constraints. The Final Project is a 12 ECTS course, which means that the application and the documentation of it, should be planned to be about 300 hours. This implies a duration of one semester with a partial dedication 20h/week.
- The student's abilities. This is a totally new development environment, so that the work has an added complexity due to the learning of that environment.
- Application requirements. When the application have less requirements, it would be more accessible, obtaining therefore more market penetration.

The development from scratch of this type of application, brings of the students new knowledge and situations similar to those found in business or research. Therefore, the main purpose of a final project should be to learn new abilities and be prepared for working life.

Focusing on the application to be developed, the aim is to acquire the biometric data record (based on a fingerprint) of an individual, who can then access the application via two different processes, such as the verification (unique testing among the individual data and template shown) or identification (multiple testing between individual data and all the templates of the server).

2. Introducción a la biometría

2.1.Historia de la biometría

He de reconocer que si me hubieran preguntado hace seis meses que entendía por biometría, mi definición no se habría parecido en nada a la correcta. La Biometría es el estudio de métodos automáticos para el reconocimiento de seres humanos basados en uno o más rangos físicos intrínsecos [1], es decir, desde mi opinión dentro de quizás diez o quince años, cuando se vaya a entrar a casa no se girará una simple llave metálica, fácil de robar o copiar, sino que se utilizará alguna de las características físicas, ya sea la huella dactilar o el iris entre otros (que posteriormente se comentarán) para conseguir así acceder al interior de la vivienda. O si se pensara a largo plazo, y viendo el increíble avance que están teniendo los terminales móviles, por qué conformarse con una validación de una clave a través de un simple sistema (como es el NFC) para poder pagar una cena en algún bar del mundo, cuando puede identificarse uno, en un par de segundos, únicamente colocando el dedo sobre algún sensor biométrico incorporado en el dispositivo para dar orden en el banco de realizar un pago.

Sin embargo la biometría no son solo identificaciones mediante diferentes dispositivos electrónicos, como podemos pensar hoy en día tras haber visto películas como Minority Report, ya que su historia se remonta en la cultura de occidente a finales del siglo XIX, y en China confiaban en ella desde el siglo XIV. La experiencia China se conoce debido a que un escritor llamado Joao de Barros, recogió en uno de sus libros biográficos, que los comerciantes chinos estampaban las impresiones y las huellas de la palma de la mano de los niños en papel con tinta, como método para identificarlos. Sin embargo en occidente no sería hasta 1883 cuando Alphonse Bertillon, jefe del departamento fotográfico de la Policía de París, desarrolló el sistema antropométrico (que más tarde sería conocido como Bertillonage). Gracias a la precisión de este sistema se consiguió en aquellos años identificar a numerosos criminales. Este método consistía en la medición de diferentes longitudes y tamaños de diferentes partes del cuerpo, además de registrar marcas individuales como tatuajes o cicatrices, lo que impulsó el desarrollo y estudio de este campo, el Reconocimiento Biométrico. No obstante, en 1903, el sistema colapsa al ser sentenciado un hombre inocente en la penitenciaría norteamericana de Leavenworth, Kansas, que tenía el mismo conjunto de medidas del hombre que había cometido el crimen. Esto llevó a que se comenzase a utilizar la huella dactilar, un sistema perfectamente integrado en la sociedad en el siglo XXI.

2.2.Procesos de autentificación de identificación biométrica

Tras esta breve introducción a la biometría, se procederá a tratar brevemente la gran variedad de modalidades biométricas que existen hoy en día [2], adjuntando posteriormente una tabla que recogerá los niveles de seguridad y complejidad (entre otros factores) de esta variedad de sistemas basándose en el conocimiento que he podido adquirir durante estos últimos seis meses.

Para empezar se hace necesario hacer una división de estos sistemas en función a su tipo:

1. **Biometría estática:** Consiste en la medición de las características físicas de un individuo. En este grupo nos encontramos con el estudio de la huella dactilar, el reconocimiento de iris o retina, el análisis de la forma de la mano, o el reconocimiento facial. Todas ellas serán posteriormente descritas, especialmente el estudio de la huella dactilar en el que se profundizará un poco más, dado que ha sido la tecnología que se ha implantado en el proyecto tratado en este documento.

2. **Biometría dinámica:** Consiste en la medición de los rasgos de comportamiento de un individuo y se basa en el reconocimiento de voz, y de firma manuscrita. Esta rama es menos conocida por todos, aunque en los últimos tiempos está logrando alcanzar puestos punteros en el mercado (como se verá posteriormente a través de un gráfico), debido principalmente a la técnica del reconocimiento de voz, algo cada vez más integrado en los dispositivos móviles de hoy en día.

2.2.1. Biometría estática

Reconocimiento de geometría de la mano: El uso de esta técnica se remonta al año 1870, cuando el antropólogo francés Alphonse Bertillon ideó un sistema de reconocimiento de individuos, basándose en las medidas físicas de la mano [3], [5].

Este sistema sería adoptado alrededor del año 1880 por la policía de Francia. Sin embargo, debido al problema anteriormente comentado, el sistema quedó bastante olvidado, dado que el nivel de seguridad que proporcionaba no era adecuado para situaciones que requiriesen una gran precisión. Sin embargo en la década de los 70 se lanzaría al mercado el primer sistema comercial de reconocimiento de geometría de la mano, siendo la universidad de Georgia una de las primeras instituciones en utilizarlo. Posteriormente en 1984 el ejercito probaría este tipo de sistemas para su uso en bancos, y un año después sería patentado por David Sidlauskas (quien mejoró notablemente el sistema de por aquel entonces, para dotarlo de una seguridad mucho mayor), quien fundó al mismo tiempo la empresa Recognition Systems Inc. La aceptación de este nuevo método de identificación a través de las características físicas de la mano fue tal que incluso en los juegos olímpicos de 1996 se utilizaría para controlar el acceso de los deportistas a la Villa olímpica.

A continuación se explicará los pasos que son necesarios seguir para llevar a cabo este método de reconocimiento biométrico.

1. **Método de captura:** Para realizar la captura de las medidas físicas de la mano se necesita una cámara digital, un espejo y una superficie plana de color sólido con cinco clavijas (que ayudará a alinear los dedos de la manera correcta para la realización de las medidas) como la que se muestra en la imagen.

Figura 1: Captura de la geometría de la mano [4]

Una vez hayamos colocado la mano con la palma hacía abajo y el espejo formando un ángulo de 60º de manera que refleje el perfil de esta hacía la cámara, procederemos a tomar la captura.

Posteriormente será necesario extraer los bordes de la imagen (preprocesado) para posteriormente extraer las características que la hagan única (extracción de características).

Para el preprocesado, se comienza pasando la imagen a escala de grises, de manera que obtengamos un alto contraste entre la mano y el fondo, y dado que esta cuenta con una componente de azul muy débil (prácticamente inexistente) la operación realizada para conseguir ese resultado será la siguiente: IByN = h(h(IR + IV)-IA).

Donde IR, IV e IA son las componentes roja, verde y azul de la imagen original, e IByN es la imagen resultante en escala de grises. Mientras que h representa la función de estiramiento del histograma.

Con esta operación lo que se intenta es eliminar aquellas zonas que cuenten con una componente azul mayor que roja y verde de la imagen original de la mano. Como ya hemos comentado antes, dado que la componente azul es muy débil, el resultado obtenido será negativo y gracias a la función h, convertiremos esos valores negativos en ceros (por lo que todo el fondo pasará a ser negro), mientras que la mano al tener una componente azul muy débil obtendrá valores muy cercanos a unos (tendrá un color muy cercano al blanco).

Tras esta operación, la imagen obtenida se pasa a valores binarios, utilizando un umbral seleccionado heurísticamente, de modo que se eliminen los brillos o ruidos de la imagen. Por último, a la imagen resultante se le aplicará un algoritmo de extracción de bordes basado en la operación Sobel, de manera que obtengamos una imagen binaria que representará los bordes de la imagen inicial y por lo tanto, el contorno del dorso de la mano y su perfil.

2. **Extracción de características:** tras la realización de los pasos anteriores, se tomarán una serie de medidas sobre la imagen obtenida, de modo que se conseguirá un vector de características que identificara a ese individuo. Las medidas tomadas se pueden dividir en cuatro grupos:
    - Alturas de la palma de la mano (h1), del dedo meñique (h2) y del dedo corazón (h3).
    - Anchura de la palma de la mano (w0) y de todos los dedos, excepto el pulgar (w11, w12, w13, w14 para el índice; w21, w22, w23, w24 y w25 para el corazón, w31, w32, w33, w34 para el anular y w41, w42, w43 y w44 para el meñique). También se mide la distancia inter-dedo (P1, P2 y P3) tanto en coordenadas horizontales (Pnx) como verticales(Pny) P1x-P2x, P1x-P3x, P1x-P2y, P1x-P3y.
    - Ángulos entre los puntos inter-dedo y la horizontal (a2 para el ángulo entre P1-P2 y la horizontal y a3 para el ángulo P1-P3 y la horizontal).
    - Desviación de los dedos, respecto a la línea recta ideal que deberían formar las falanges. Estas desviaciones serán la distancia entre el punto medio del contorno del dedo (P12 para el dedo índice) y el punto medio de la recta definida entre el punto inter-dedo correspondiente (P1 en nuestro caso) y el punto más alto del contorno de ese dedo (P14 para el índice). Es decir que en el caso tratado la fórmula matemática sería:
      $Desv = P_{12}^{X} - ((P_{14}^{X}-P_1^{X})/P_{14}^{Y}-P_1^{Y})*(P_{12}^{Y}-P_1^{Y})$.

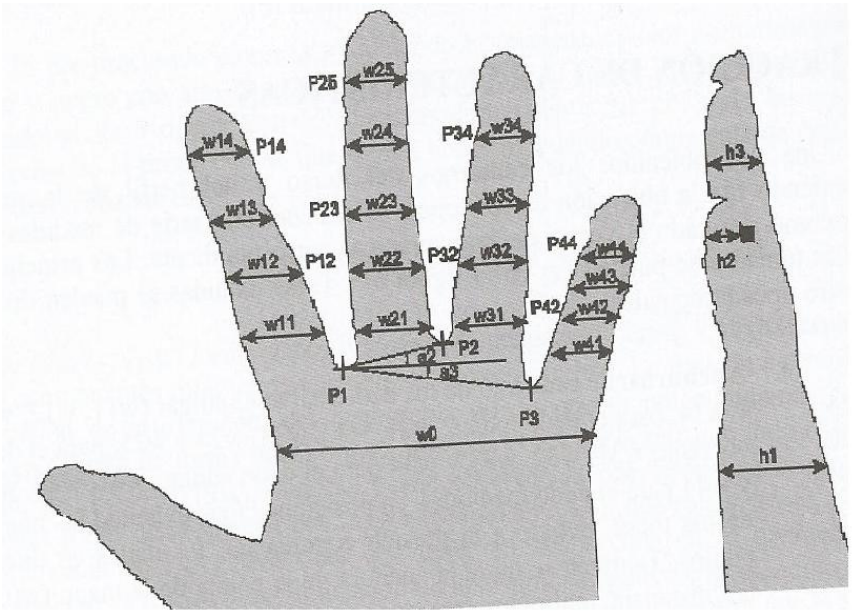

Figura 2: Extracción de características de geometría de la mano [5]

Además destacaremos que para minimizar la variación de las características con la edad y la pérdida o ganancia de peso, se han tomado medidas relativas en lugar de

absolutas, es decir todas las medidas han sido divididas por w21 (el ancho más cercano a los puntos inter-dedo del dedo corazón).

Por último es necesario añadir que para poder crear una base de datos aceptable, es necesario tomar 3 o 4 fotografías de la mano, extraer las medidas de cada una y almacenar un patrón medio, ya que con menos muestras no se obtendría un resultado óptimo, y con más el proceso resultaría demasiado pesado para el individuo. Además en el año 2007 se desarrolló un estándar en ISO de la familia de normas 19794, concretamente el 19794-10 que regula el intercambio y almacenamiento de la información obtenida a través del proceso explicado con anterioridad.

Reconocimiento de iris: Para empezar se hace necesario aclarar que no se debe confundir este procedimiento biométrico con el reconocimiento de retina, ya que como se verá posteriormente aunque ambos precisen del ojo humano para su funcionamiento, no tienen nada en común [6], [5].

Esta técnica de reconocimiento biométrico ha ido adquiriendo una gran relevancia en el mercado a lo largo de los últimos años, ya que es cierto que quizás cuenta con el procedimiento más complejo, ya que gracias a la protección que le confiere la córnea al iris, este nos proporciona una gran estabilidad frente a accidentes que con otros sistemas como el reconocimiento de huella dactilar, o el citado anteriormente (reconocimiento de la geometría de la mano) sería imposible conseguir.

Así, el auge de este tipo de tecnología se debe principalmente a dos factores:

- El primero sería que confiere una tasa de falsa aceptación mucho mayor que con el resto de procedimientos de reconocimientos basados en rasgos fisiológicos, ya que en el iris hay más información para identificar de forma unívoca al usuario, que incluso en la huella dactilar. Tanto es así que incluso se hace necesario distinguir entre ambos ojos de un mismo individuo, ya que la información que contienen sus irises es muy diferente.
- Por otro lado esta parte del ojo nos proporciona un mecanismo bastante fiable para saber si el individuo, en el momento de realizar el escaneo está vivo o no, ya que el iris presenta pequeños cambios en su apertura tanto con cambios de iluminación como de iluminación fija.

Esta tecnología comenzaría a ser estudiada en el año 1936 por parte del oftalmólogo Frank Burch, sin embargo no sería hasta 1987 cuando la idea fue patentada gracias a los esfuerzos de los oftalmólogos Leonard Flom y Aran Safir, que contando con la ayuda del profesor John G.Daugman de la Universidad de Harvard desarrollaron un algoritmo capaz de realizar todo el proceso, siendo éste patentado en 1994 y fijando las bases de cualquier mecanismo de reconocimiento de iris actual.

La primera compañía que se encargó de desarrollar este tipo de tecnología, sería la fundada por Daugman, Flom y Safir (IriScan Corp), pero debido a problemas económicos, tuvo que fusionarse en el año 2000 con Sensar Corp, formando así Iridian Technologíes, encargada de promover todo lo relacionado con este tipo de tecnología biométrica hasta la actualidad.

El procedimiento utilizado para el reconocimiento de individuos a través del reconocimiento de iris se basa en los siguientes pasos:

1. **Captura de la imagen del Iris:** Lo primero sería analizar el tipo de elemento con el que se quiere realizar la captura, ya que se puede utilizar tanto una cámara digital, como una cámara de video. Indiferentemente del dispositivo de captura utilizado se hace necesario tener presente tres cosas: Que la resolución de la cámara debe ser suficiente para capturar la imagen sin ningún problema, que el individuo debe acercarse en gran medida al dispositivo, sin que este le confiera ningún tipo de peligro, y que dado que el usuario no se encontrará pegado a la cámara, esta distancia no debe suponer un deformamiento en la imagen capturada. Además se recomienda trabajar en el rango infrarrojo (debido a que la córnea presenta un alto grado de reflexión de la luz que le llega) para todo el proceso de captura, ya que el usuario no sentirá ningún tipo de molestia, y las imágenes obtenidas son independientes del color.
   Como ya se ha comentado anteriormente, para evitar fraudes como el de presentar una fotografía que recoja el iris, se suelen realizar varias tomas consecutivas, en las que se debe apreciar una variación en la dilatación de la pupila, o también en ocasiones se fuerzan cambios de iluminación para analizar así la respuesta de la pupila a dichos cambios.

2. **Preprocesado del iris:** Lo primero sería realizar una conversión de la imagen a escala de grises, para posteriormente aumentar el contraste de esta a través de un estiramiento del histograma. Posteriormente será necesario detectar el borde externo del iris utilizando para ello un algoritmo iterativo de búsqueda del máximo gradiente de intensidad a lo largo de una circunferencia. El máximo se obtendrá debido a que el centro y radio de la circunferencia irán variando de tal forma que se recorra una gran superficie de la imagen. Tras la detección del borde externo, es preciso aplicar las transformaciones necesarias al centro y al valor del borde para obtener los correspondientes en la imagen original. Posteriormente se formará un cuadrado que englobe al iris detectado, para posteriormente eliminar aquellos puntos que queden fuera de la circunferencia que enmarca al iris. Más tarde se eliminan mediante umbral los puntos de sobreexposición y se estira el histograma. Por último en el paso del preprocesado es necesario detectar el borde interno del iris, para lo que es necesario realizar una nueva búsqueda del centro, ya que la pupila y el iris no son concéntricos por lo que no se puede utilizar el mismo máximo. Con esta nueva búsqueda, se conseguirá el centro de la pupila, el radio y su posición dentro del borde externo del iris. Para obtener este centro se realizará un proceso similar al utilizado para la detección del borde externo en el paso anterior. Posteriormente, los puntos

comprendidos dentro de la circunferencia definida se anularán en la imagen resultante de la detección del borde externo. Por último se realizará un estiramiento del histograma obteniendo así el iris aislado del resto de la imagen.

3. **Adaptación del iris detectado:** En este último paso lo que se deberá hacer es realizar un muestreo en ángulo y radio de la imagen obtenida al final del paso anterior (para así obtener la misma cantidad de datos independientemente del tamaño del iris y la pupila y suprimir los conos superior e inferior), y conseguir así una simplificación del algoritmo de extracción de características.
   En la imagen que tenemos a nuestra derecha se puede ver el proceso explicado anteriormente, en el que los conos laterales, a través de un muestreo de radio y ángulo se convierten en una imagen cuadrada. Conectando ambas imágenes, se obtendrá la matriz rectangular que se utilizará en el bloque de extracción de características. Se debe resaltar que esta matriz siempre contará con el mismo tamaño, ya que siempre se seleccionará el mismo punto a la hora de realizar el muestreo de separación entre el borde exterior e interior.
   Por último es necesario extraer las características de la matriz rectangular obtenida en el apartado anterior. Para ello se pueden utilizar diversas técnicas como la Transformada de Wavelet y la Extracción por Circunferencia, aunque la más utilizada son Los Filtros de Gabor.

Identificación por escaneo de retina: Esta técnica de identificación biométrica se basa en los vasos sanguíneos contenidos en la retina [7], [5]. La información obtenida a través del análisis de la distribución de los vasos sanguíneos que nacen en el nervio óptico es tan única, que incluso en el caso de dos hermanos gemelos idénticos, el patrón conseguido sería completamente distinto, por lo que este tipo de tecnología está ganando cada vez más peso en aplicaciones que requieran una alta seguridad y una tasa de falso aceptamiento muy baja. No obstante es cierto que esta técnica provoca un importante rechazo, ya que es necesario que el individuo se posicione muy cerca del dispositivo de captura, por lo que se dice que es una técnica intrusiva. Además algunos tipos de enfermedades e incluso la edad pueden alterar la imagen de la retina a lo largo del tiempo, por lo que aún no se ha explotado como se debería una técnica biométrica que confiere tal nivel de seguridad.

Los pasos seguidos para la identificación mediante el escaneo de retina serán similares a los del caso anterior, por eso solo se destacará el procesado de la imagen de la retina.

Captura y procesado de la imagen de la retina: tras realizar la captura de la imagen con dispositivos similares a los utilizados en el reconocimiento de Iris se procede a obtener un patrón de características único que identifique al individuo. Los pasos seguidos para la obtención de dicho patrón serían los siguientes:

- Extracción de los perfiles de intensidad de los vasos sanguíneos que cubren la retina.
- Determinación del área de estudio.

- Localización de los vasos sanguíneos.
- Generación del patrón de retina.

Tras estos pasos, se obtendrá un patrón que oscilará entre los 50 y los 96 bytes. Por último destacaremos que el mecanismo de comparación en este tipo de técnica biométrica suele realizarse a juicio del desarrollador y en general es bastante robusto.

Reconocimiento de huella dactilar: Como ya se ha comentado anteriormente, esta es la técnica de identificación biométrica más extendida en la sociedad, debido a que cumple con los dos requisitos principales que hacen de la biometría una ciencia exacta y única, como son la invariabilidad temporal y la variedad infinita del autentificador [8], [5].

Los primeros estudios sobre la huella dactilar se remontan a 1665 cuando un anatómico italiano (Marcelo Mapighi) con ayuda de su novedoso artilugio de laboratorio (microscopio), comenzase a realizar diferentes análisis sobre las crestas papilares, sentando así las bases de cualquier estudio consistente en analizar la huella dactilar, que desarrollarían el fisiólogo holandés Ruish y el médico botánico inglés Nehemiag Grew. No obstante, la idea de utilizar la huella dactilar como método de reconocimiento humano no surgiría hasta 1823 cuando gracias al fisiólogo checo Johannes Purkinje se conoció que los dibujos papilares permanecían inalterables a lo largo de toda la vida, y que estos surgían a los seis meses de vida.

Sin embargo no podría considerarse que la biometría surja como tal hasta el año 1888, ya que fue entonces cuando el científico inglés Sir Francis Galton comenzase a realizar diferentes estudios sobre el uso de las crestas dactilares como método de identificación biométrico. Consiguiendo posteriormente, en el año 1910 la medalla de The Royal England Society al confirmar el segundo requisito comentado anteriormente (la inexistencia de dos huellas digitales iguales).

En 1890, el inspector jefe de la Policía de Bengala, Sir Edward Henry asentó las bases de la dactiloscopia moderna, al idear un sistema de clasificación de huellas de un modo lógico, basándose en la forma del dibujo de las crestas.

Los avances producidos a partir de estos años han sido múltiples y numerosos, basándose principalmente en los sistemas automáticos de identificación de huellas dactilares, conocidos como AFIS (Automated Fingerprint Identification System). Estos permitían la obtención (relativamente rápida) de identificaciones biométricas de entre múltiples individuos.

El procedimiento para diferenciar una huella dactilar de otra se basa en un conjunto de líneas genéricas denominadas crestas. Estas son las partes de la piel donde esta se eleva sobre las zonas más bajas denominadas valles y cuya anchura oscila entre las 2 y 5 décimas de milímetro.

Como ya se ha comentado anteriormente la huella es única para cada individuo, incluso en el caso de gemelos idénticos, ya que esta se desarrolla como un proceso aleatorio y no genético a partir del sexto mes de vida. Además el dibujo papilar crece sin alterar el número, el grado de curvatura ni la situación de las crestas, ya que lo hace proporcionalmente según el desarrollo físico corporal, por lo que permanecen inalteradas con el paso de los años.

No obstante hay ciertos sectores de la población que presentan dificultades para poder ser identificados a través de sus huellas dactilares. Estos son principalmente:

- Colectivos profesionales: Las personas que utilizan sus manos para trabajar de una manera intensa (herreros, albañiles, carpinteros, etc.) pueden presentar diferentes callosidades en sus dedos, por lo que la adquisición de los datos de sus huellas dactilares, puede resultar bastante costosa, e incluso imposible.
- Colectivos étnico: Las personas de origen asiático, así como las personas de edad avanzada, presentan unas crestas pequeñas y finas, dificultando así la adquisición de las medidas que diferencian una huella de otra.

Las características particulares de las crestas, que diferencian una huella de otra, reciben el nombre de minucias, y se distinguen dos tipos:

- Bifurcación de la Cresta: Es el punto en el que la cresta se bifurca en dos o más crestas. Corresponden al 32% del total de minucias presentes en la huella (varía entre 40 y 100).
- Final de Cresta: Es el punto en el que la cresta se acaba de forma abrupta. Corresponden al 68% restante. Se debe destacar que para poder identificar a un individuo correctamente, la doctrina judicial nacional establece la obtención de al menos 12 puntos característicos.

A pesar del diferente estilo, ambas se definen a partir de su localización gracias a sus coordenadas espaciales (X e Y) y orientación (ángulo θ).

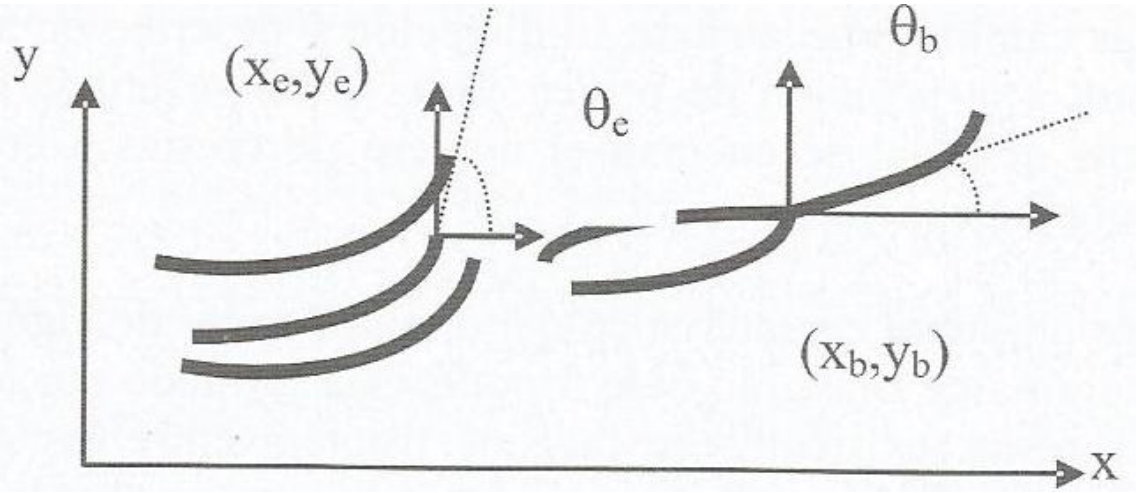

Figura 3: Coordenadas y orientación de huella dactilar [5]

Por otro lado, se distinguen dos singularidades presentes en las huellas:

- Core: Es el punto localizado den la zona nuclear de la huella, y se corresponde con un cambio brusco de la dirección de una cresta (180º). Este punto se utiliza como referencia para contar el número de crestas a considerar en un análisis dactiloscópico concreto.
- Delta: Es un punto que está formado por la aproximación o fusión de las crestas existentes en la zona de frontera entre las zonas nuclear, marginal y basilar de la huella (suele presentar forma de triángulo trípode). En la zona donde se encuentra ubicado, se encuentran muchos puntos característicos de la huella, además esta singularidad será utilizada para realizar una clasificación inicial de las huellas como se verá posteriormente.

Figura 4: Huella dactilar [5]

Para la clasificación de las huellas, se sigue un sistema piramidal (formado por dos peldaños), donde se realizaría una primera división en función del número de Deltas presentes: Adeltas (si no presentan ninguna), engloban el 20% del total, Monodeltas (si presentan una), engloban el 50% y Bideltas (si presentan dos) engloban el 30% restante. A continuación se dividen en otras

seis clases, las seis clases presentadas en la imagen. Siendo las dos primeras (leyendo de izquierda a derecha y de arriba abajo) un subtipo del grupo Adelta, la tercera y la cuarta un subtipo del grupo Monodelta y las dos últimas un subtipo del grupo Bidelta.

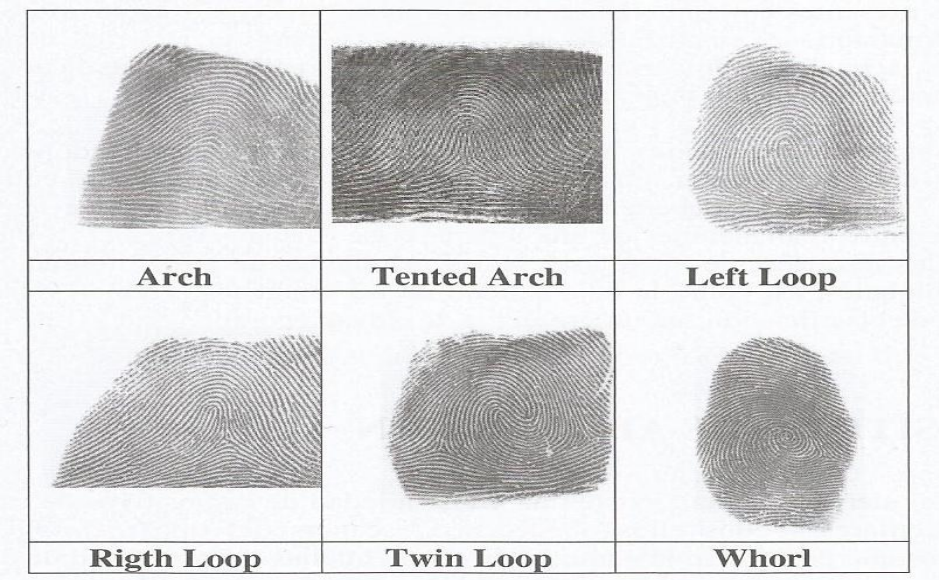

Figura 5: Tipos de huellas dactilares [5]

Sin embargo, debido a la multitud de muestras dactilares que manejan los sistemas actuales, se realiza una clasificación previa de la huella, para posteriormente (en la etapa de reconocimiento) manejar un subconjunto menor.

Una posible implementación de esta clasificación previa sería a través de los Filtros de Gabor, diferenciando así cuatro tipos de huellas, en función de la dirección de sus crestas (0º, 45º, 90º y 135º).

En cuanto a los dispositivos de captura existentes en el mercado actual, a pesar de que todos utilizan escáneres del tipo Inkless (permiten la adquisición de huellas sin tener que calcar los dibujos papilares previamente entintados) podemos distinguir diferentes clases en cuanto a su forma y utilización:

- Lectores integrados: Son dispositivos periféricos de captura de huellas que se conectan al ordenador a través de un puerto USB y almacenan la información obtenida en el disco duro. Pueden incorporar un lector de tarjetas inteligentes y una unidad de encriptación adicional (dotando al sistema de una mayor seguridad).
- Lectores OEM: Se encuentran perfectamente integrado en el sistema electrónico.
- Terminales completos de identificación: Realizan todo el proceso de captura y verificación de la huella dactilar.
- PCMCIA.

También se pueden clasificar los dispositivos de captura según el tipo de sensor utilizado:

- Óptico: Utiliza un sensor de imagen del tipo CCD (Dispositivo de Acoplamiento de Carga) para la captación de la imagen. Este dispositivo se basa en una matriz de fotosensores que convierten la radiación luminosa en una tensión proporcional a la misma. La resolución, el tamaño y el factor de ruido serán los parámetros que determinen la calidad de este tipo de sensores.

  Otro tipo de sensores ópticos serían los CMOS que a pesar de presentar una menor sensibilidad cuenta con ventajas como un menor consumo y una elevada capacidad de integración.

  La principal ventaja de este tipo de sensores sería la elevada resolución de la imagen digital obtenida, presentando como mayor contra la sensibilidad del dispositivo a la suciedad presente tanto en el sensor como en la huella a capturar. Además, este tipo de sensores suelen presentar problemas de aberración de perspectiva en la zona marginal de la huella.

- Capacitivo: Se basa en la utilización de un sensor electromagnético, que detecta la diferencia de capacidades entre la huella y el sensor, es decir, el sistema mide la capacidad existente entre él y las crestas de la huella, para posteriormente traducirla a niveles de gris. Se debe destacar que en caso de que la huella estuviese demasiado húmeda, la imagen obtenida sería demasiado negra, mientras que si estuviese excesivamente seca provocaría una captura demasiado blanca, debido a que el sudor humano presenta una elevada constante dieléctrica.

Sus principales ventajas son el reducido tamaño del dispositivo y su bajo consumo. Por el contrario su principal desventaja como ya se ha comentado anteriormente sería la elevada sensibilidad a variaciones en los parámetros de humedad de la huella.

- Ultrasónica: Es la más novedosas de las citadas anteriormente. Su funcionamiento se basa en medir la diferencia acústica (se envía un barrido de ondas ultrasónicas que rebotan sobre la huella) que existe entre las crestas y los valles de la huella.

  Su principal ventaja es que dado que realiza una lectura tridimensional de la huella en lugar de bidimensional será mucho más resistente que las anteriores tecnologías. Además evitará los errores de lectura producidos por la presencia de partículas ajenas a la piel o en la platina de escaneo. Por el contrario, se debe mencionar que la resolución obtenida con este tipo de tecnología, es sensiblemente inferior a la obtenida con sensores ópticos, además el precio es notablemente superior.

A continuación se describirá el proceso realizado para el reconocimiento de huellas dactilares.

Para empezar es necesario destacar que las técnicas de reconocimiento se dividen en dos categorías:

- Técnicas globales u holísticas, basadas en la correlación. Su principal desventaja es que este procedimiento necesita de la implementación de algoritmos de alineación muy precisos, por lo que se trata de una técnica muy sensible a traslaciones y rotaciones de la huella durante la captura de esta.
- Técnicas locales o analíticas, basadas en minucias. Su principal debilidad radica en la costosa extracción de las minucias en imágenes que no posean una buena calidad.

Para el desarrollo de este proyecto, se ha utilizado una técnica basada en minucias, que además es la utilizada en la mayoría de los casos, por eso centraremos el análisis en este tipo de técnicas.

El proceso para el reconocimiento de individuos a partir de la huella dactilar se divide en:

1. **Captura de la huella:** Permite almacenar la imagen de la huella dactilar para su posterior análisis. Es necesario destacar que existen dos técnicas de captura. Para la realización de este proyecto se ha utilizado el método on-line, que se realiza en tiempo real, y es el utilizado normalmente para aplicaciones civiles. No obstante para aplicaciones de ámbito criminalístico se suele utilizar un método off-line que obtiene la huella digitalizada con una resolución de 500 puntos por pulgada y a 256 niveles de profundidad de gris, a través del escaneo de un positivo impreso en papel obtenido anteriormente a partir de la operación tradicional de calcado del dedo tintado sobre papel satinado.

   Una vez realizada la captura, se procederá a realizar una valoración para decidir qué hacer con la huella capturada, teniendo como posibilidades declarar la huella apta para ser procesada, declararla inutilizable debido a su baja calidad, o declararla recuperable mediante técnicas de preprocesado digital de la imagen.

   Antes de decidir si la imagen es declarada apta o no apta, se realiza un preprocesado de la imagen (consiste en discriminar si los píxeles evaluados pertenecen a una cresta o no). Para ello es necesario:

   1) Mejorar la imagen para reducir así la información redundante y extraer las crestas correctamente. Para lo que utilizaremos filtros lineales direccionales que disminuirán el ruido y acentuarán las transiciones claro-ocuro y viceversa. Pero para ser posible la utilización de este tipo de filtros es necesario conocer la orientación local de las crestas próximas a cada pixel, que se habrá obtenido utilizando el mapa o campo de orientación (Se divide la imagen en bloques de 15x15 y se calcula el gradiente en X e Y de cada pixel). Posteriormente se obtendrá el ángulo de orientación aplicando al gradiente el algoritmo de ajuste por mínimos cuadrados.
   2) Binarizar la imagen para obtener la huella monocroma, es decir se convertirá la imagen de 256 niveles de gris en una imagen monocroma. Para ello se utiliza el método de Otsu o Criterio del Discriminante que dará un umbral a partir del cual

los píxeles que lo superen tomarán el valor uno (blanco) y los que no el valor cero (negro).

3) Valoración de la calidad de la huella para decidir si continuar con el proceso o descartar la imagen y reiniciar el procedimiento de captura de huella.
   Para decidir si la imagen tiene una calidad adecuada se utilizará la relación (S/N+S) siendo N+S la energía de la señal estimada del siguiente modo: $\sum_{u,v}(p(u,v)-p_{avg})^2$. Donde p(u,v) son los valores de los píxeles, p_{avg} el nivel de gris promedio de la región R y (u,v) pertenecen a la región R.
4) Extracción de la región de interés de la huella (ROI) para eliminar la información redundante perteneciente al fondo de la imagen, de manera que se reduzca considerablemente el tamaño final de la imagen. Para el cálculo de la ROI existen diferentes métodos:
   - Dividir la imagen en bloques de píxeles para extraer los que cuenten con una mayor varianza del nivel de gris en la dirección normal a las crestas existentes. El contorno de interés será el definido por dichos bloques.
   - Detectar el punto de referencia (core) y definir a partir de él una región de referencia circular siendo el centro de esta las coordenadas del punto core aislado. La región de interés será la región circular calculada.
5) Adelgazamiento (thinning) para reducir el grosor de las líneas, de modo que todas cuenten con un grosor de un píxel. Para ello se utilizará la técnica MM (Morfología Matemática), que se basa en un análisis no lineal de la imagen mediante estructuras geométricas.

   Esta técnica (MM) se basa en dos operaciones morfológicas:

   - Operador morfológico de adelgazamiento que elimina los píxeles que satisfacen el patrón que define el elemento resultante mediante la substracción entre la imagen original (I) y el resultado de la operación morfológica hit-miss (EE), cuyo objetivo es buscar la coincidencia de la imagen original con un patrón determinado
   - Función morfológica Skeleton para obtener la imagen final adelgazada. Consiste en una iteración de la operación thinning mediante la utilización de una secuencia de elementos estructurantes generados a partir de la rotación sucesiva del EE.

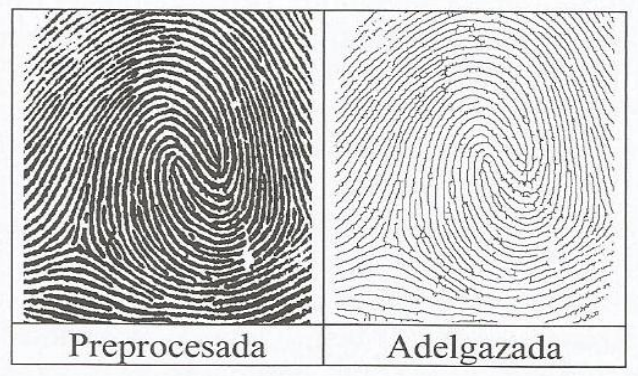

Figura 6: Huella dactilar adelgazada [5]

6) Depuración para eliminar los defectos producidos en las crestas de la huella durante el proceso de adelgazamiento. Para ello se siguen los dos pasos siguientes:
   - Utilizando el operador thinning que cuenta con un elemento estructurante que se adapta a las nuevas necesidades se eliminan las pequeñas líneas aisladas.
   - Conectar todas las crestas que se hayan roto.

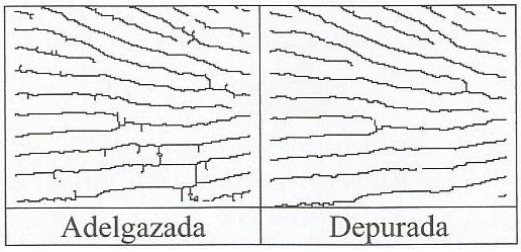

Figura 7: Huella dactilar depurada [5]

2. **Creación del modelo:** Se extraerán las minucias (características identificativas) de la imagen para almacenarlas formando un patrón único que identifique al individuo de entre un gran número de muestras. El fichero resultante tendrá un tamaño aproximado de 300 bytes, de manera que solo se almacenen las minucias y no la imagen al completo.
   Como no es posible obtener el mismo número de minucias siempre, se debe aplicar un algoritmo de reconocimiento que pueda comparar el vector de características candidato con la plantilla correspondiente. Dicho algoritmo proporcionará dos salidas:
   - La información local de las crestas.
   - Un conjunto de minucias caracterizadas por su posición y orientación espacial.

   Una vez localizadas las minucias, en función del número de vecinos (habiendo un total de 8) que presente cada pixel (de las líneas adelgazadas previamente) podrá ser un final de cresta (si solo cuenta con un vecino) o una bifurcación (si cuenta con 3 o más vecinos).

3. **Comparación del modelo:** Consiste en la comparación de la imagen que tenemos con una o varias de nuestra base de datos, a fin de comprobar si el usuario fue registrado en el sistema con anterioridad. En el caso de encontrarse en la operación de verificación, únicamente se comparará la imagen actual con la obtenida de la base de datos a través de los datos proporcionados por el usuario. En caso de que se trate de un mecanismo de identificación se deberá comparar la imagen recientemente obtenida con todas las de la base de datos, y mostrar una lista de los posibles candidatos.

   Para realizar esta comparación se deben seguir dos pasos:

   - Algoritmo de alineamiento: Se utiliza para calcular los parámetros de rotación y traslación que se ajusten al mayor nivel de correspondencia espacial, al ajustar la huella parametrizada con la plantilla.

     Lo primero será seleccionar una minucia de referencia de cada imagen, para posteriormente determinar el número de pares de minucias que se corresponden. A continuación se repetirá el proceso de selección para cada uno de los pares de combinaciones posibles de minucias que cuentan con características comunes. El par de minucias finalmente seleccionado será el que cuente con mayor número de pares de minucias que se corresponden (el que ha estimado mejor alineamiento).

     Posteriormente es necesario calcular los parámetros de rotación (media de los valores individuales de rotación de todos los pares de minucias que se corresponden) y traslación (coordenadas espaciales del par de minucias de referencia que ha supuesto un mejor alineamiento).

     Por último se aplicarán dichos parámetros a todas las minucias del modelo de test.
   - Algoritmo de comparación de modelos: Dará como resultado la identidad de la persona (si se almacenó previamente en la base de datos tratada) que presente una distancia euclídea menor con respecto al modelo de test evaluada. Para ello lo primero será representar las minucias en coordenadas polares para posteriormente ordenarlas formando cadenas en orden creciente de sus coordenadas angular y radial. Estas dos cadenas serán las utilizadas por el algoritmo para realizar la comparación anteriormente citada.

Reconocimiento facial (2D y 3D): Este procedimiento emplea técnicas similares a las empleadas en el reconocimiento de la geometría de la mano, por eso no se describirá con tanta exactitud como técnicas anteriores. Únicamente se comentará que la identificación de individuos mediante la estructura física de su cara existe, aunque no sea un campo muy extendido [9], [5].

También es necesario añadir que en las tecnologías que usan un reconocimiento en 2D no es posible distinguir si el rostro visualizado es real, o es tan solo una fotografía, por lo que su seguridad es muy baja en comparación con el resto de tecnologías analizadas.

Reconocimiento vascular: Al igual que el anterior es una técnica muy poco extendida en la actualidad, por lo que solo será mencionada. Esta técnica de identificación consiste en la extracción del patrón biométrico a partir de la geometría del árbol de venas del dedo, algo que confiere a esta técnica una gran seguridad, pues dado que este patrón es interno solo puede conseguirse en presencia de la persona física, evitando así una gran cantidad de posibles falsificaciones [5].

2.2.2. Biometría dinámica

Reconocimiento de escritura y firma: Se trata del método más antiguo de reconocimiento biométrico y a pesar de lo que se pueda pensar hoy en día, (frente a la cantidad de avances tecnológicos y nuevas formas de identificación con altos grados de seguridad como se ha visto anteriormente) este sigue siendo un sistema muy utilizado, y es que gracias a la aparición de las PDA´s , tablets y teléfonos móviles con pantallas táctiles, todo el proceso de captura y posterior comprobación de la firma escrita se ha digitalizado para poder así competir con tecnologías quizás más llamativas como el escaneo de retina [10], [5]. No obstante siguen existiendo campos donde se necesita una persona física que realice la comprobación de la firma como por ejemplo los cheques bancarios. Para el uso de esta tecnología, se debe realizar un proceso dividido en los siguientes pasos:

1. **Captura de la firma:** Esta se puede realizar de dos modos:
    - Off-line: Se realiza la captura de la firma sobre un papel físico, para posteriormente digitalizarla y extraer las características principales que la diferencien del resto de firmas.
    - On-line: Gracias al uso de dispositivos informáticos como tablets, PDA´s etc se realiza la captura y extracción de características de forma inmediata y simultánea.

    La principal diferencia entre ambas, es que en la captura Off-line, se pierde una gran cantidad de información dinámica y temporal de la firma, ya que es imposible saber a posteriori el tiempo empleado en la realización de la firma, así como la aceleración de la mano en diferentes puntos o la secuencia ordenada en la ejecución de los trazos.

2. **Acondicionamiento de la señal de la firma**: Consiste en la eliminación de la información que no sea relevante para el reconocimiento, en la corrección de la información degradada durante la adquisición y en la reducción de la variabilidad

entre las distintas realizaciones de una firma escrita. Esta parte del proceso será diferente en función de si trabajamos con una captura On-line o usamos la Off-line:

- Off-line: Comenzará con la binarización de la imagen previamente obtenida. Posteriormente se eliminará el ruido mediante la unión de trazos cortados o uniendo huecos (esta parte puede realizarse también antes de la binarización aunque en este caso será necesario usar filtros paso bajo). A continuación se aplicará la segmentación (aislamiento de los trazos que contienen información relevante). Por último se aplica una normalización en posición y tamaño.
- On-line: Lo primero será realizar un alineamiento respecto a la posición (tomando como referencias el punto inicial o el centro de masas normalmente) y a continuación se normalizará la imagen capturada en tiempo real en rotación y tamaño.

3. **Extracción de características y representación de la firma:** Las características extraídas de la firma y que diferencien a esta del resto deben ser discriminantes entre firmas falsas y verdaderas y permanecer estables ante las más que probables variaciones en las firmas verdaderas. Así las características extraídas se pueden clasificar en función a dos criterios:
    - Su naturaleza (estáticas o dinámicas): las estáticas miden la velocidad o duración de la realización de la firma (información temporal), mientras que las dinámicas se centran en la inclinación de los trazos verticales o la localización del inicio y fin de estos trazos (información geométrica).
    - El ámbito de representación (globales o locales): Las totales serían tales como la duración total, desviaciones típicas, medias y centro geométrico (toman información de la firma en su totalidad como unidad). Las locales serían la medición de máximos y mínimos o los valores instantáneos de diferentes parámetros (información de zonas específicas de la firma).

Una vez extraídas las características, es necesaria la representación de la firma. Al igual que en infinidad de ocasiones anteriores, (ya sea en esta u otras tecnologías), se cuenta con dos modos distintos de representar la firma tratada:

- Representación paramétrica: Las características serán un conjunto de parámetros (duraciones globales y locales, velocidad de escritura máxima, mínima y media, etc) que se agruparán en un vector que represente la firma.
- Representación mediante funciones (solo puede aplicarse a firmas adquiridas mediante el proceso On-line): La firma será representada como una función temporal o espacial que representará la evolución de determinados parámetros a lo largo de la realización de la firma. Las funciones más comunes son posición, presión y fuerza entre otros.

Reconocimiento de voz: Esta tecnología comenzó a desarrollarse en el año 1960 gracias a la compañía Texas Instruments y a pesar de que las investigaciones en este campo han sido numerosas y se confía en esta técnica como una de las más prometedoras actualmente, se debe destacar que la variabilidad presente en la señal de voz supone un gran punto negativo en la calificación global de esta tecnología [11], [5]. En este caso nos centraremos más en una visión global de este tipo de tecnología que en su descripción funcional.

Para empezar se tratará el funcionamiento de los principales sistemas de reconocimiento de voz. Estos pueden trabajar de tres formas distintas:

- Modo de entrenamiento: Consiste en la obtención de patrones y valores de referencia de los individuos.
- Modo de servicio: Es el proceso de comparación, es decir en el que el sistema decidirá sobre la identidad de un individuo.
- Modo de actualización: Consiste en la eliminación, sustitución o agregación de nuevos individuos al sistema durante su periodo de vida útil.

A continuación se intentará resumir el funcionamiento de un sistema de reconocimiento de voz:

Lo primero será convertir la secuencia de voz obtenida a una serie de vectores que identifiquen de manera inequívoca dicha señal de voz (Preprocesado acústico). A continuación se comparan dichos vectores, con unos parámetros previamente conocidos por el sistema (se obtuvieron en el modo de entrenamiento), en la fase de comparación de similitudes. Lo último será decidir si el individuo evaluado se corresponde con la identidad probada o no. Para visualizar este funcionamiento se presenta a continuación una imagen que lo recoge perfectamente.

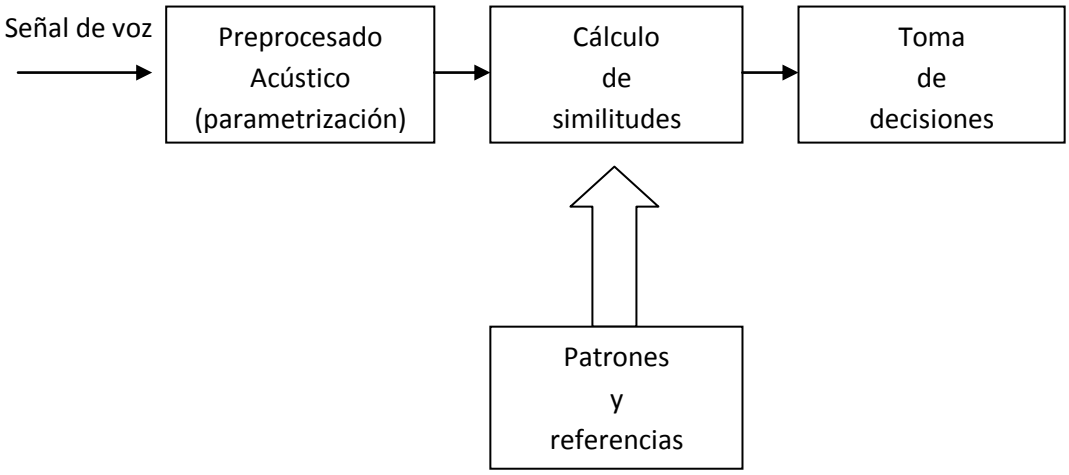

Figura 8: Reconocimiento de voz [5]

Tras esta breve descripción de su funcionamiento se pasará a distinguir entre los diferentes tipos de sistemas de reconocimiento de voz. Para ello se comenzará dividiéndolos en sistemas dependientes de texto o no dependientes de texto. Los primeros requerirán la vocalización de

una serie de palabras o frases exactas por parte del individuo, mientras que los segundos dejan el texto a pronunciar a libre elección del usuario.

Por otro lado, se debe distinguir entre los sistemas de identificación (clasifican una señal de voz, cuyo origen es desconocido como perteneciente a un individuo de entre un conjunto restringido. Ya sea identificación en conjunto cerrado, donde el resultado es una asignación de identidad a uno de los locutores modelados por el sistema y conocido como usuario, o en conjunto abierto donde se considera además de las posibilidades del caso anterior que el locutor no pertenezca al grupo de usuarios por lo que debería ser identificado como impostor en el sistema) y los de verificación cuya respuesta es únicamente la aceptación o el rechazo, (ya que reciben solo dos parámetros, la señal de voz a identificar y la solicitud de identidad que puede ser realizada mediante la lectura de una tarjeta magnética individual o mediante el tecleado de un código de locutor por ejemplo).

Por último se tratarán las aplicaciones más frecuentes de este tipo de tecnología y el estado actual del reconocimiento de voz como método de identificación. Actualmente los usos más comunes son el control por comandos para abrir y cerrar aplicaciones ya sea en dispositivos móviles o en PC´s y la introducción de información a un ordenador a través del habla, ya que el programa lo convierte a texto escrito (es una aplicación muy frecuente en personas con alguna deficiencia física que no les permita realizar esa tarea sin la ayuda de ese software). Pese a que estas aplicaciones son usadas cada vez con más frecuencia, el método de identificación mediante reconocimiento de voz, no termina de ganarse un puesto en el mercado actual, debido principalmente a los altos índices de error que presenta.

Para cerrar el apartado de biometría y como se comentó anteriormente, a continuación se presenta una tabla que recoge los niveles de fiabilidad o aceptación entre otros.

Tabla 1: Características de sistemas biométricos [12]

	Ojo (Iris)	**Ojo (Retina)**	**Huellas dactilares**	**Vascular**	**Geometría de la mano**	**Escritura y firma**	**Voz**	**Cara 2D**	**Cara 3D**
Fiabilidad	Muy alta	Muy Alta	Muy Alta	Muy Alta	Alta	Media	Alta	Media	Alta
Facilidad de uso	Media	Baja	Alta	Muy Alta	Alta	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Muy Alta	Alta	Muy Alta	Alta	Media	Media	Media	Alta
Aceptación	Media	Baja	Alta	Alta	Alta	Muy Alta	Alta	Muy alta	Muy alta
Estabilidad	Alta	Alta	Alta	Alta	Media	Baja	Media	Media	Alta

3. Biometric Identity Assurance Services (BIAS)

3.1. Introducción

Para empezar a tratar esta sección del proyecto, se hace indispensable la necesidad de contestar a una pregunta: ¿Qué es BIAS? Bien, BIAS es un estándar definido por OASIS (Organization for the Advancement of Structured Information Standards) que establece y define los métodos necesarios a implementar para realizar una correcta comunicación entre las transacciones de servicios web basadas en XML y arquitecturas SOA (Orientadas a Servicios), con el fin de proporcionar a las industrias de seguridad biométrica, un marco adecuado de comunicación [13], [15].

La historia de BIAS comenzó en el año 2008, cuando en EEUU la organización OASIS, decidiese abrir un comité técnico (TC) para desarrollar servicios de identificación para aplicaciones cliente/servidor. Una vez avanzada esta especificación y enviada a consulta pública (incluso por voluntarios fuera de la organización OASIS), decidió donar dicha especificación a ISO/IEC JTC1/SC37 para que sirviera de base para un estándar internacional. Dicha donación fue aceptada en 2010 y resultó en la apertura de un proyecto de normalización que diera como resultado la futura norma ISO/IEC 30108, que actualmente se encuentra en fase de borrador de norma (DIS), la penúltima etapa antes de su publicación.

Antes de continuar tratando el tema de BIAS, se hace necesario comprender qué es ISO (International Organization for Standardization) y qué es OASIS.

ISO es una organización no gubernamental sin ánimo de lucro donde el desarrollo de las normas se hace por votación nacional. Es el mayor desarrollador de estándares internacionales, que definen las especificaciones de productos, servicios y buenas prácticas para ayudar a hacer que la industria sea más eficiente y eficaz, consiguiendo así eliminar las barreras del comercio internacional. ISO fue fundado en 1947 y desde entonces ha publicado más de 19500 normas internacionales, que abarcan casi todos los aspectos de la tecnología y los negocios, desde la seguridad de los alimentos, a los ordenadores, y de la agricultura, a la salud. Por último se añadirá que hoy en día cuentan con 163 países miembros y 3368 organismos técnicos para cuidar la elaboración de las normas.

OASIS por el contrario es un consorcio privado donde las votaciones se hacen por miembros, los cuales pagan una tarifa para formar parte. De esta manera, OASIS no puede garantizar el acuerdo internacional de sus trabajos. OASIS impulsa el desarrollo, convergencia y adopción de estándares abiertos para la sociedad de la información mundial. Tratan de promover el consenso de la industria a través de la producción de estándares internaciones para la seguridad, las redes inteligentes, los servicios web y otras áreas. Los estándares abiertos de OASIS ofrecen la posibilidad de reducir notablemente los costes, impulsar la innovación y el crecimiento de los mercados mundiales, así como proteger el derecho a la libre elección tecnológica. El consorcio cuenta con más de 5000 participantes en representación de más de 600 organizaciones y miembros individuales en 100 países diferentes.

Tras este breve paréntesis para aclarar la magnitud de las organizaciones que estamos tratando, se procederá a tratar el contenido del estándar en sí.

Para empezar se presentará una imagen de la especificación original de OASIS, que recoge perfectamente el contexto en el que se desarrollarán este tipo de aplicaciones.

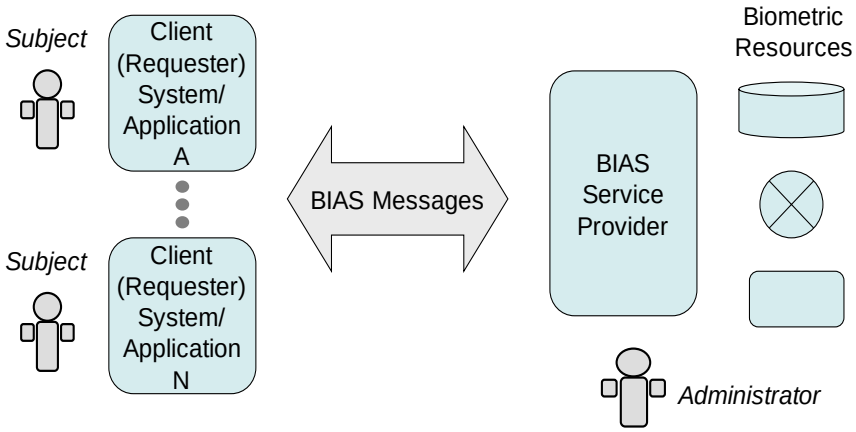

Figura 9: BIAS según OASIS [14]

Como se ha comentado anteriormente, BIAS está destinado a proporcionar una interfaz común, pero flexible, de servicios web que se pueda utilizar dentro de los sistemas SOA tanto abiertos como cerrados. Esto es así dado que en la actualidad se están desarrollando sistemas biométricos que recogen, procesan y almacenan todo tipo de datos personales con una gran variedad de propósitos. Sin embargo y a pesar de que en numerosas ocasiones se hace necesario realizar un intercambio de información entre diferentes sistemas, la falta de normalización en este campo, ha llevado a los desarrolladores a proporcionar servicios personalizados para cada servicio o aplicación. Por ello con BIAS se pretende unificar el mercado y conseguir un estándar único que de un servicio adecuado a la gran diversidad de aplicaciones tanto existentes como emergentes.

3.2.Tipos de servicios y modelos

Nada más comenzar a leer el estándar, se puede ver que éste hace una distinción entre dos categorías diferentes de servicios (primitivos y agregados). Los primitivos son servicios más sencillos de implementar, que proporcionan las funcionalidades básicas, mientras que los servicios agregados tienden a ser un nivel superior, es decir tanto su complejidad, como el nivel de flexibilidad que proporcionan en la parte del servidor BIAS son mayores. La diferenciación entre estos tipos de servicios, es crucial para determinar qué nivel de conformidad queremos que se aplique al proyecto, dotándole así de una mayor o menor gama de servicios.

La otra gran distinción que realiza el estándar es en cuanto al modelo de identificación utilizado, ya que este puede estar centrado en la persona (person-centric) o en un número de

visita (encountar-based). A continuación se procederá a describir detalladamente los dos tipos de modelos de identificación citados anteriormente.

- **Person-Centric model:** En este modelo centrado en la persona, cuando se recibe un dato biométrico o bien se añade (si no existe) o reemplaza el anterior (si ya existe). Así en la figura que se puede ver debajo de este texto, el sistema almacenaría unos datos biométricos iniciales que se recogen del sujeto tras el registro. Posteriormente cuando el usuario intente acceder al sistema o se reciban nuevos datos biográficos, los datos iniciales serán remplazados, de manera que el sistema siempre contenga la muestra más reciente del individuo. Este tipo de modelo es utilizado principalmente en sistemas de control de acceso, debido principalmente a que no necesita un gran espacio para almacenar los datos, ya que solo conserva una muestra por usuario. Por contra, este modelo no será capaz de almacenar un historial completo del usuario, un inconveniente que en las aplicaciones en las que suele ser utilizado no es muy influyente.

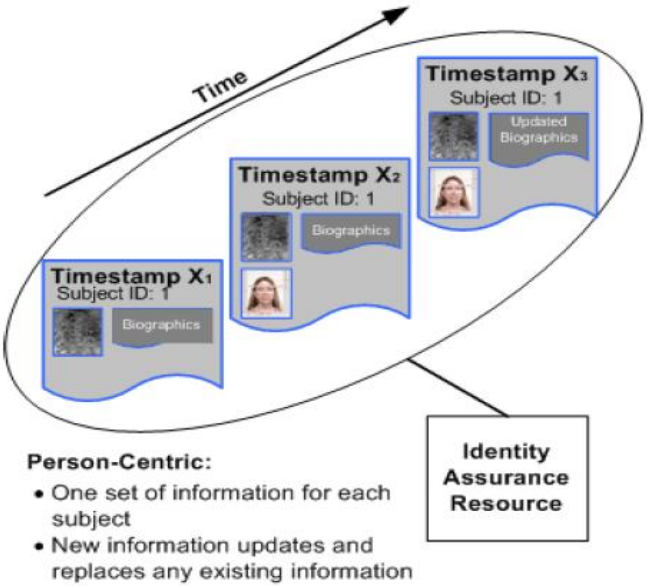

Figura 10: Modelo Person-Centric [15]

- **Encounter-Centric model:** En este caso se almacenan una gran cantidad de registros pertenecientes al usuario, de manera que se cree un historial y se pueda ver la evolución de la muestra biométrica. Es decir, como se puede ver en la imagen siguiente, en el registro inicial se almacenarán los datos biométricos y biográficos pertenecientes al individuo, pero posteriormente, cuando el usuario intente acceder al sistema, los nuevos datos recogidos se almacenarán aparte asociándoles un número que normalmente representa el número de visita. De esta forma se consigue un historial (más o menos completo, en función del número de visitas que el sistema haya sido definido para almacenar) de los datos del usuario. A diferencia del sistema anterior, este resulta mucho más completo, aunque la cantidad de memoria que

requiere es mucho mayor que la del sistema anterior. Este tipo de modelo se utiliza por ejemplo en un sistema de gestión de fronteras.

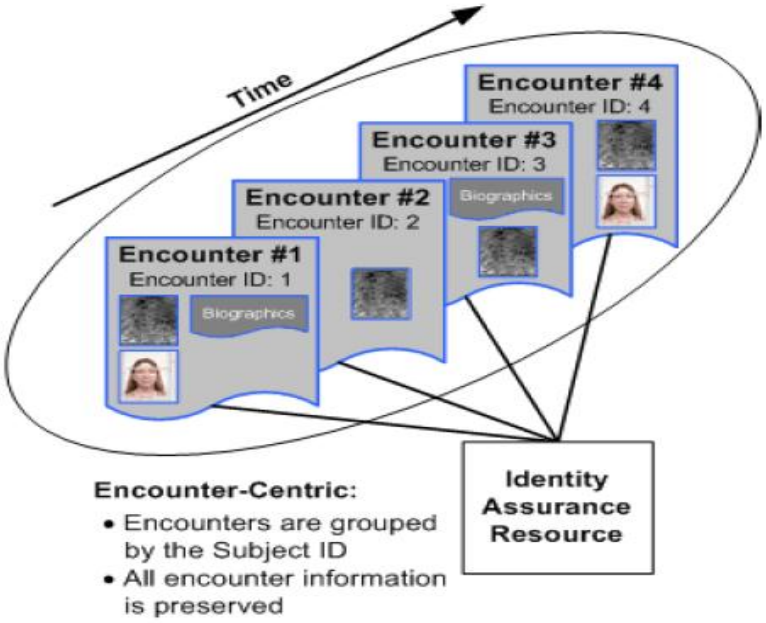

Figura 11: Modelo Encounter-Centric [15]

En la imagen que se presenta a continuación queda perfectamente definido el funcionamiento de este tipo de modelos.

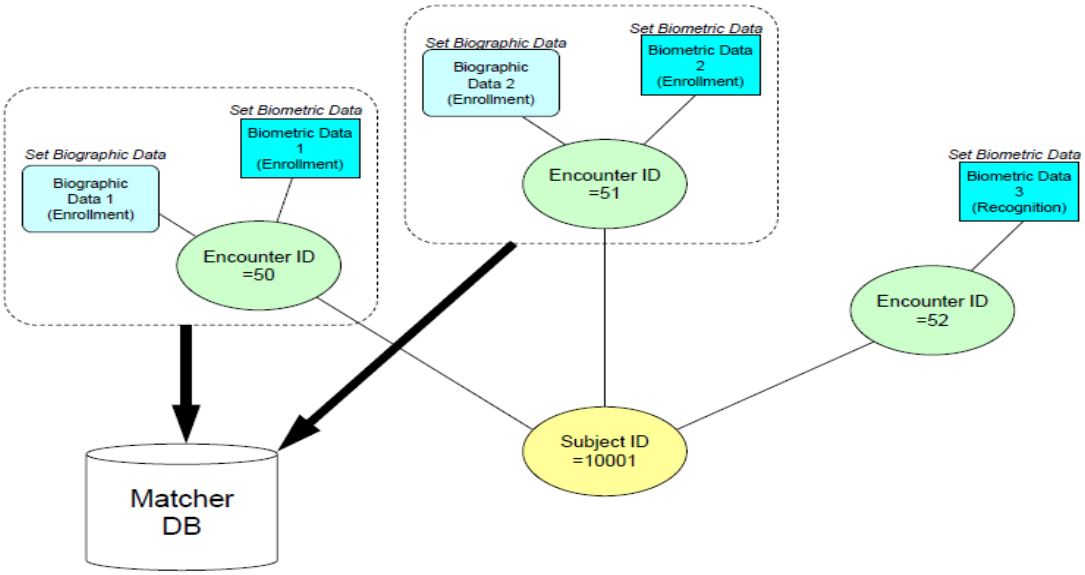

Figura 12: Modelos BIAS [15]

Tras esta breve descripción de los tipos de modelos y de servicios que nos proporciona BIAS, se pasará a tratar el tema del almacenamiento de datos, es decir porque en ocasiones se utiliza el modelo centrado en la persona (que consume menos memoria) y no el centrado en el número de visita que es mucho más completo.

3.3.Bases de datos

Para el almacenamiento de datos, BIAS utiliza bases de datos, con un gran núcleo de almacenamiento denominado maestro, en el que se guardan todos los datos biográficos del usuario que nuestra aplicación requiera, así como el número de visita en caso de que nos encontremos ante el modelo correspondiente. Sin embargo, normalmente las aplicaciones que utilizan BIAS, también cuentan con otro núcleo (más pequeño) llamado comparador en el que se almacenan generalmente los datos biométricos de los diferentes usuarios, y en caso de que la aplicación utilizada, requiera demasiada memoria y el núcleo principal no pueda prestar un servicio adecuado, algunos datos biográficos serán desviados a esta segunda base de datos. No obstante debemos destacar que en el estándar se define que todos los servicios deben operar sobre la base de datos central, a menos que puntualmente se indique lo contrario, como por ejemplo en las actualizaciones de las bases de datos secundarias.

A continuación se presenta una figura con la que se pretende aclarar perfectamente el proceso de utilización de bases de datos según el estándar.

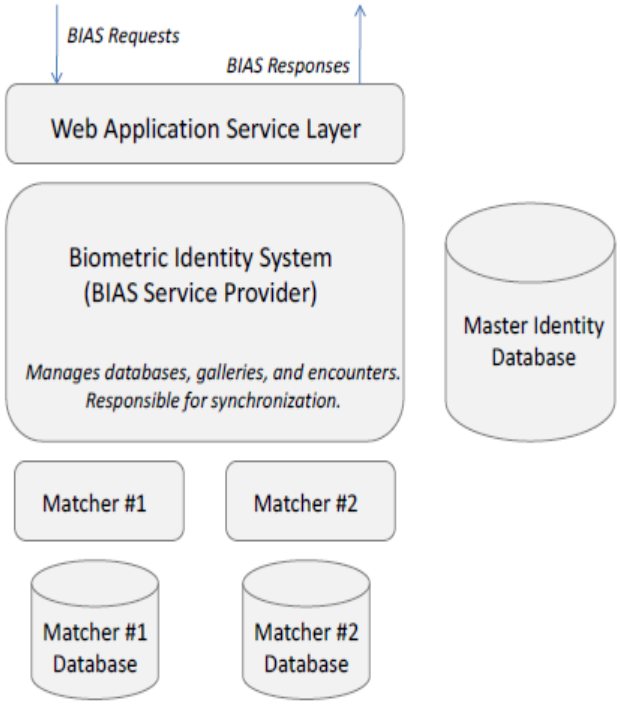

Figura 13: Base de datos BIAS [15]

Este sería uno de entre muchos posibles casos. Aquí tras el registro del usuario en el sistema, los datos biométricos de las huellas dactilares, el iris, la retina etc, serían almacenados en la base de datos maestra. Adicionalmente las huellas, el iris, etc, serían procesados y sus plantillas se almacenarían en la base de datos comparador. Posteriormente, durante la segunda visita, solo se recogerían los datos necesarios para la verificación o identificación, y estos serían almacenados en la base de datos maestra, donde serán comparadas con las primeras muestras obtenidas, y en caso de que estas últimas sean de mejor calidad, sustituirán a las anteriores, para posteriormente crear nuevas plantillas y enviarlas de nuevo a la base de datos comparador.

Tras esta descripción de las bases de datos utilizadas se pasará a tratar la arquitectura orientada a servicios, que como bien se comentó anteriormente utiliza este estándar.

3.4.Arquitectura orientada a servicios

Una arquitectura orientada a servicios, es un tipo de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Esta arquitectura permite la creación de sistemas de información altamente escalables, que reflejan el negocio de la organización y a su vez ofrecen una forma bien definida de exposición e invocación de servicios, lo que facilita enormemente la interacción entre diferentes sistemas propios o de terceros.

Los servicios biométricos se pueden proveer a través de una interfaz remota existente en un sistema de información distribuido a través de varias redes. Un buen ejemplo de esto se muestra en la siguiente figura. En ella se recoge el diagrama de capas utilizado por cualquier tipo de aplicación de reconocimiento biométrico que requiera de los servicios de BIAS para realizar una correcta comunicación entre el cliente donde se capturan los datos del usuario, y el servidor en el que estos datos serán almacenados, para posteriormente puedan ser recuperados y realizadas las comprobaciones pertinentes.

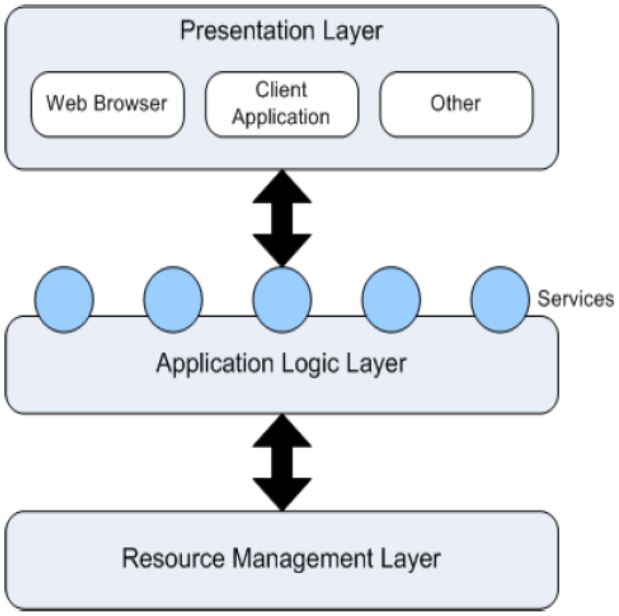

Figura 14: Capas BIAS [15]

En este simple diagrama, los servicios BIAS serían implementados entre la Application Logic Layer (capa lógica de aplicación) y el Resource Management Layer (capa de gestión de recursos). Alguno de los recursos biométricos que se podrían incluir serían por ejemplo: Una base de datos de identidad biográfica basada en el nombre y/o apellidos, un sistema de identificación de huellas digitales automatizado penal o civil, un archivo de identificadores biométricos, una población de sujetos o una simple lista de vigilancia biométrica facial. Este estándar define un conjunto genérico de servicios que permiten a los clientes acceder y administrar remotamente estas capacidades. Además es necesario destacar que en la medida de lo posible las implementaciones de dominio específicas deben ser evitadas.

También se define que los servicios deben estar bien definidos y que los módulos que proporcionan una funcionalidad estándar de negocio, son independientes de la situación o el contexto de otros servicios y que pueden ser fácilmente ensamblados para formar un conjunto de procesos de negocio autónomos y de acoplamiento flexible.

El estándar pretende evitar que la lógica empresarial se encuentre instanciada dentro de las definiciones de servicio, ya sea en el sistema de nivel superior que inicia la serie de solicitudes, o en el nivel intermedio. En caso de hacerlo sería necesario que la interfaz fuese menos genérica, modular y flexible y exigir que la interfaz sea actualizada cada vez que se cambie la lógica.

Por último se añadirá que los servicios que se definen no están dirigidos a una implementación de SOA en particular. En su lugar se definen de una manera tal como para ser capaz de ser utilizados dentro de dicha arquitectura. Esto se consigue mediante la definición por separado

(en otro estándar) de los enlaces para la arquitectura/aplicación. Por ejemplo los enlaces de servicios Web están definidos en el protocolo de mensajería OASIS BIAS.

3.5.Arquitectura BIAS

La arquitectura BIAS consta de los siguientes componentes:

- Servicios BIAS (definición de la interfaz).
- Los datos de BIAS (definición de esquema).
- Enlaces BIAS (definido fuera de este estándar).

Los servicios de BIAS exponen un conjunto común de operaciones a los solicitantes externos de estas operaciones. Los solicitantes pueden ser un sistema externo, una aplicación web o un intermediario. Los propios servicios de BIAS se encuentran en plataforma y lenguaje independiente, además pueden implementarse con diferentes tecnologías en múltiples plataformas. Por ejemplo, OASIS es la definición de enlaces de servicios Web para los servicios de enlace.

La figura que se presenta a continuación representa los servicios BIAS dentro de un entorno de aplicación. Estos proporcionan una funcionalidad básica biométrica como operaciones modulares e independientes que se pueden montar de diversas maneras. Además pueden estar expuestos públicamente directa o indirectamente en apoyo de los propios servicios públicos.

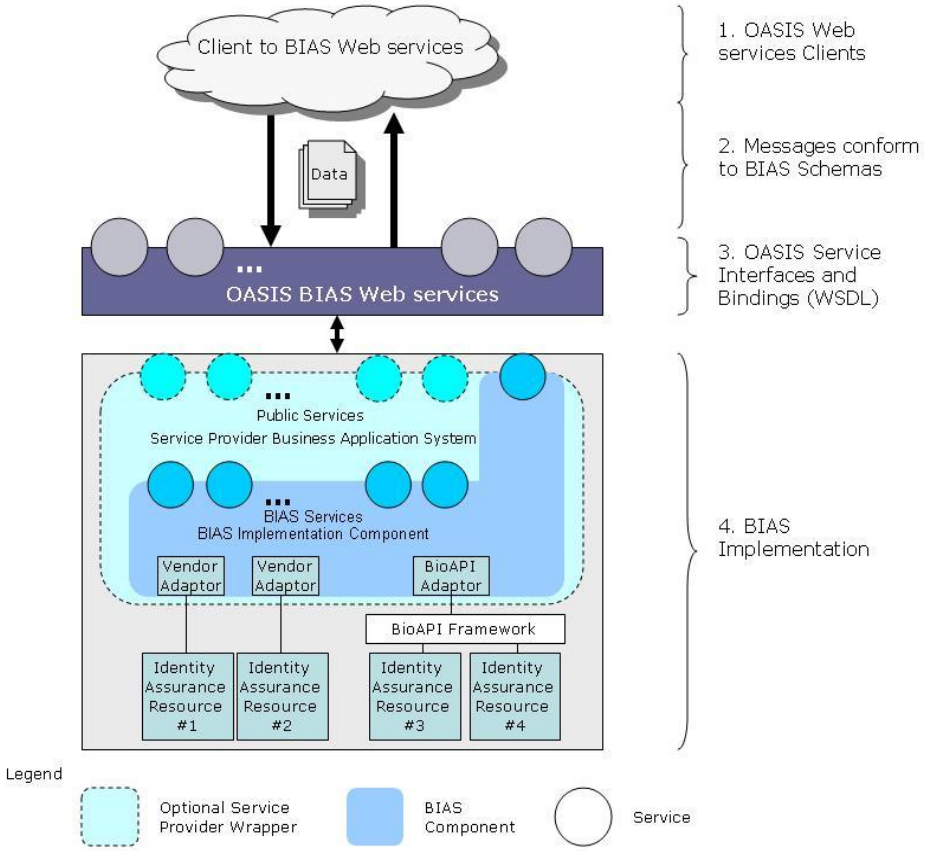

Figura 15: Servicios BIAS [14]

3.6.Requisitos de conformidad de clase

Este estándar establece una serie de servicios que se deben implementar obligatoriamente y que variaran en función de la robustez y capacidad de implementación que se le quiera otorgar a la aplicación implementada.

A continuación se presenta una tabla que recoge dichos servicios.

Tabla 2: Servicios BIAS [15]

Service/capability	Class 1	Class 2	Class 3	Class 4	Class 5
Servicios primitivos					
Add Subject to Gallery	X		X	X	
Chek Quality	X				
Classify Biometric Data	X				
Create subject	X		X	X	
Delete Biographic Data	X		X	X	
Delete Biometric Data	X		X	X	
Delete Subject	X		X	X	
Delete Subject From Gallery	X		X		

Get identify Subject Results	X		X		
Identify Subject	X		X	X	
List Biographic Data	X		X	X	
List Biometric Data	X		X	X	
Perform Fusion	X				
Query Capabilities	X	X	X	X	X
Retrieve Biographic Data	X		X	X	
Retrieve Biometric Data	X		X	X	
Set Biographic Data	X		X	X	
Set Biometric Data	X		X	X	
Transform Biometric Data	X				
Update Biographic Data	X		X	X	
Update Biometric Data	X		X	X	
Verify Subject	X		X	X	
Servicios agregados					
Delete		X			
Enroll		X			X
Get Delete Results		X			
Get Enrol Results		X			*
Get Identify Results		X			
Get Verify Results		X			*
Identify		X			
Retrieve Data		X			X
Verify		X			X
Elementos de información de capacidad					
AggregateInputDataOptional		X			X
AggregateInputDataRequired		X			X
AggregateProcessingOption		X			X
AggregateReturnData		X			X
AggregateServiceDescription		X			X
BiographicDataSet	X	X	X	X	X
CBEFFPatronFormat	X	X	X	X	X
ClassificationAlgorithm Type	X				
ConformanceClass	X	X	X	X	X
Gallery	X	X	X		
IdentityModel	X	X	X	X	X
MatchScore	X	X	X	X	X
QualityAlgorithm	X				
SupportedBiometric	X	X	X	X	X
Transform Operation	X				

*Requerido si el servicio implementado es asíncrono.

Por último se añadirá que la clase implementada en este proyecto ha sido de nivel 4.

3.7.Implementación de BIAS

Se hace necesario comentar que los servicios biométricos y las aplicaciones que se utilizan requieren de un contexto de seguridad adecuado. Algunas de las medidas adoptadas son las citadas a continuación:

- Existen dos tipos de servicios en cuanto a la duración temporal. Por un lado se presentan aquellos que requieren de un gran tiempo para realizar una verificación o identificación correcta, como una identificación 1:N en una gran población. Por otro lado se encuentran aquellos servicios que requieren un reconocimiento rápido, como por ejemplo cualquier sistema de control de acceso utilizado en la actualidad. Por ello la interfaz implementada debe ser capaz de dar soporte a ambos tipos de servicios, es decir a operaciones tanto síncronas como asíncronas.
- Otra medida fundamental a tener en cuenta para la implementación de BIAS, sería que los servicios para los que vaya a utilizarse, podrían ser multi-biométricos o singulares, es decir a pesar de que este proyecto se centre en el reconocimiento biométrico a través de huella dactilar, no se debe olvidar que algunas aplicaciones requieren de varios métodos de reconocimiento biométrico simultaneo, por lo que BIAS debe conseguir dar soporte adecuadamente a este tipo de aplicaciones también, y no solo a las que utilicen un único sistema de reconocimiento biométrico.

Por otro lado se debe destacar que la aplicación realizada mediante la utilización de BIAS debe cumplir rigurosamente todos los protocolos de protección de datos del usuario, para lo que será necesario fijar los términos y condiciones de la utilización de los datos del usuario entre emisor y receptor antes de iniciar cualquier tipo de transacción. Para asegurarse del debido cumplimiento del citado protocolo de protección de datos, es necesario plantearse las siguientes cuestiones:

- ¿Quién será el destinatario de los datos que se comparten?
- ¿Con que propósito usará el receptor dicha información?
- ¿Cuánto tiempo puede conservar los datos el receptor?
- ¿Quién o qué autoriza que esta información sea compartida con el receptor?
- ¿Una vez finalizado el periodo de retención de datos por parte del receptor, de qué modo se procederá a su destrucción?
- ¿Comparte el receptor esta información con otras identidades? Si es así ¿Con qué propósito?

Como se puede observar, la seguridad es muy importante en este tipo de aplicaciones, de manera que antes de cualquier intercambio de datos, las características de seguridad proporcionadas por BIAS deben ser conocidas tanto por emisor como por receptor. Pero no solo eso, sino que una cadena de protección de datos debe ser creada para el seguimiento de estos mientras dure el intercambio, y una vez finalizado dicho intercambio y acabado el periodo de retención de datos por parte del emisor, la supresión de dichos datos debe ser rastreada y registrada.

4. Diseño

4.1.Entorno de desarrollo (Visual Studio 2010)

4.1.1. Introducción

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, visual C# [16], Visual J# y Visual Basic. Net, al igual que entornos de desarrollo web como ASP.NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros [17], [18].

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

4.1.2. Creación de Visual Studio

Microsoft Visual Basic es una plataforma desarrollada por Microsoft en 1990 con el objetivo de facilitar la programación de aplicaciones a todos los usuarios.

Para ello, Microsoft pensó en un producto con un lenguaje de programación sencillo, como el BASIC, pero a su vez potente que permita crear cualquier tipo de programa para Microsoft Windows.

En 1990, el hecho de realizar programas de ordenador requería un alto conocimiento de un lenguaje de programación como C/C++, lo que requería numerosos esfuerzos para crear cualquier programa de MS-DOS o de Windows, o bien utilizar Microsoft QuickBasic, para la elaboración de programas en MS-dos aunque resultaban ser de baja calidad.

Por ello el equipo de Microsoft decidió inventar un concepto de programación sencilla para todos. El resultado fue Microsoft Visual Basic, que permite generar programas con interfaz gráfica programando de forma sencilla, aunque haciendo programas potentes y estables.

En 1991 fue presentado Microsoft Visual Basic 1.0. Esta versión era algo escasa, pero era posible usar su interfaz gráfica para hacer ventanas, botones, cuadros de texto etc, y vincularlos entre sí.

A continuación se presenta una captura de pantalla de la primera versión del programa (Microsoft Visual Basic 1.0) que posteriormente se contrastará con la versión más actual (Visual Studio 2012).

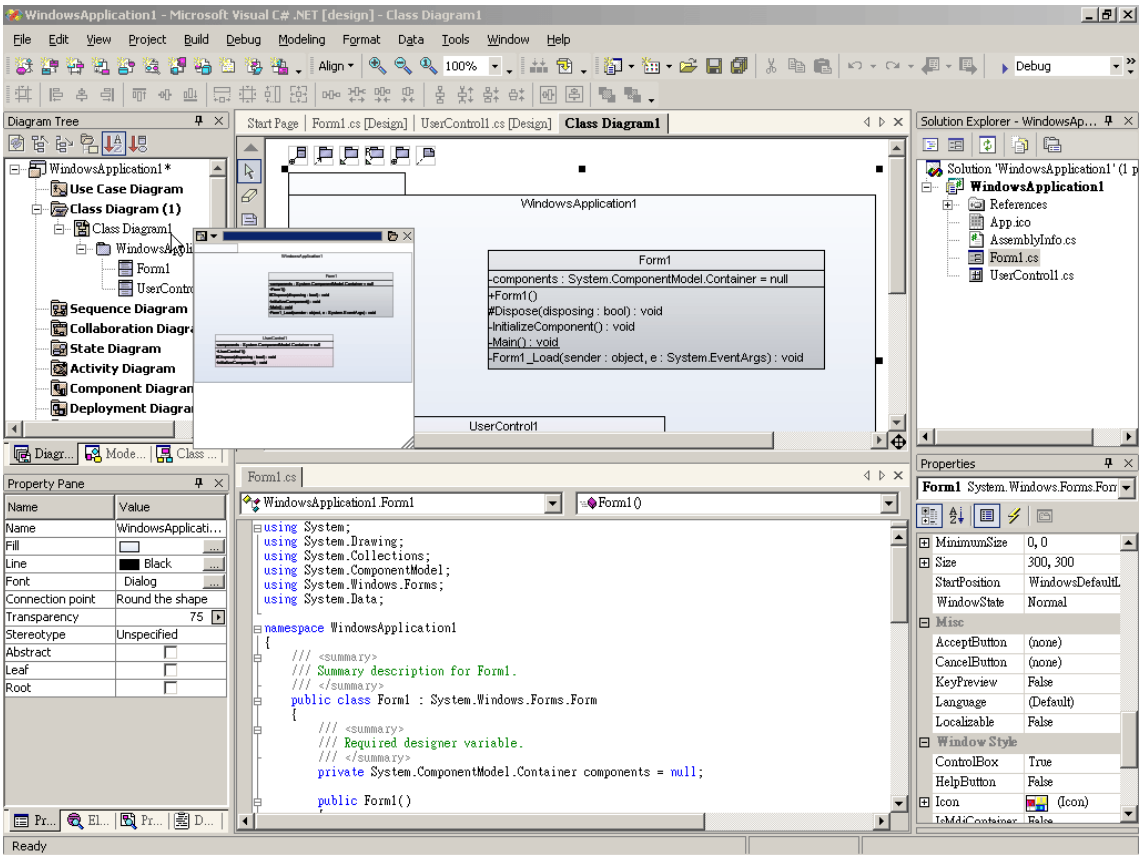

Figura 16: Visual Studio 1.0 [18]

A los años siguientes se lanzaron nuevas versiones de Microsoft Visual Basic, y en 1992 se lanzó una versión para MS-DOS, que hacía interfaces gráficas de texto y que usaba el mismo estilo que la versión para Windows.

4.1.3. Historia Reciente de Visual Studio

La historia reciente de Visual Studio comienza en 1998 cuando se lanzó la última versión que se ejecutó en la plataforma Win9x. Los números de versión de todas las partes constituyentes pasaron a 6.0 incluyendo Visual J++ y Visual InterDev, que se encontraban en las versiones 1.1 y 1.0 respectivamente. Esta versión fue la base para el sistema de desarrollo de Microsoft para los siguientes 4 años, en los que Microsoft migró su estrategia de desarrollo al .NET Framework.

Visual Studio 6.0 fue la última versión en la que Visual Basic se incluía de la forma en que se conocía hasta entonces. Las versiones posteriores incorporarían ya una versión muy diferente del lenguaje con muchas mejoras, fruto de la plataforma .NET. También supuso la última versión en incluir Visual J++, que proporcionaba extensiones de la plataforma Java, lo que lo hacía incompatible con la versión de Sun Microsystems. Esto acarreó problemas legales a

Microsoft y se llegó a un acuerdo en el que Microsoft dejaba de comercializar herramientas de programación que utilizaban la máquina virtual de Java.

Aunque el objetivo a largo plazo de Microsoft era unificar todas las herramientas en un único entorno, esta versión en realidad añadía un entorno más Visual Studio 5.0, ya que Visual J++ y Visual InterDev se separaban del entorno de Visual C++, al tiempo que Visual FoxPro y Visual Basic seguían manteniendo su entorno especifico.

Posteriormente se lanzarían versiones como Visual Studio .NET 2002, Visual Studio .NET 2003, Visual Studio 2005, Visual Studio 2008 antes de llegar a la versión utilizada en este proyecto: Visual Studio 2010.

4.1.4. Visual Studio 2010

Visual Studio 2010 es la versión de esta herramienta que se ha utilizado en el TFG, acompañada por .NET Framework 4.0. La fecha de lanzamiento de la versión final fue el 12 de Abril de 2010.

Hasta ahora, uno de los mayores logros de esta versión ha sido el de incluir las herramientas para desarrollo de aplicaciones para Windows 7, tales como herramientas para el desarrollo de características de Windows 7 (System Windows Shell) y la Ribbon Preview para WPF.

Entre sus principales características se encuentra la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo.

Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft, como Widows, Azure, Windows Phone 7 o Sharepoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con este Visual Studio 2010 también es posible desarrollar aplicativos para pantallas multitáctiles.

Entre las versiones disponibles de Visual Studio 2010 se deben distinguir:

- Visual Studio 2010 Ultimate (utilizado para el desarrollo del proyecto tratado en este documento): Conjunto completo de herramientas de gestión de ciclo de vida de una aplicación para los equipos que garantizan unos resultados de calidad, desde el diseño hasta la implementación. Ya sea creando nuevas soluciones o mejorando las aplicaciones existentes, Visual Studio 2010 Ultimate permite llevar las ideas a la vida en un número creciente de plataformas y tecnologías, incluyendo la nube y la computación paralela.
- Visual Studio 2010 Premium: Un conjunto de herramientas completo que simplifica el desarrollo de aplicaciones para personas o equipos que entregan aplicaciones escalables de alta calidad
- Visual Studio 2010 Professional: La herramienta esencial para las personas que realizan tareas de desarrollo básico. Visual Studio 2010 Professional simplifica la compilación, la depuración y el despliegue de las aplicaciones en una variedad de plataformas

incluyendo SharePoint y la Nube. También viene con el soporte integrado para el desarrollo con pruebas y con las herramientas de depuración que ayudan a garantizar unas soluciones de alta calidad.

- Visual Studio Team Foundation Server 2010: Una plataforma de colaboración en el centro de la solución de gestión del ciclo de vida de una aplicación (ALM) de Microsoft. Team Foundation Server 2010 automatiza el proceso de entrega del software y contiene las herramientas necesarias para gestionar eficazmente los proyectos de desarrollo de software a través del ciclo de vida de IT.
- Visual Studio Test Professional 2010: Visual Studio Test Professional 2010 es un conjunto de herramientas integrado que entrega un flujo de trabajo completo planificar-probar-seguir para una colaboración en contexto entre los probadores y los desarrolladores, aumentando considerablemente la visibilidad de los probadores en la globalidad del proyecto.
- Visual Studio Team Explorer Everywhere 2010: Permite a los equipos de desarrollo colaborar fácilmente entre las plataformas. Team Explorer Everywhere 2010 contiene las herramientas y los plug-ins necesarios para acceder a Visual Studio Team Foundation Server 2010 desde dentro de los entornos basados en Eclipse, de manera que todas las personas puedan trabajar juntas y lograr los objetivos del negocio.

4.1.5. Última versión de Visual Studio (2012)

La última versión de esta herramienta ha sido Visual Studio 2012, concretamente la segunda actualización, ya que la primera apareció en agosto de 2012, mientras que la segunda lo hizo ya en el año 2013 a finales del primer trimestre.

De esta última versión se debe destacar su compatibilidad con Windows 8, ya que esta última versión ofrece plantillas, diseñadores y herramientas de evaluación y depuración, con las que generar nuevas aplicaciones con esta última versión de Windows, es decir, esta nueva versión ofrece un kit de herramientas visuales que consiguen aprovechar al máximo la nueva interfaz de Windows 8.

Como otra característica fundamental se debe destacar su compatibilidad con la nube, es decir hace años el escalamiento requería importantes inversiones en infraestructura. Ahora, se tiene acceso rápido a servidores virtuales ilimitados en la nube con la posibilidad de agregar más almacenamiento y capacidad informática. Esta última versión de Visual Studio contiene excelentes herramientas para llevar las aplicaciones a Windows Azure, con nuevas plantillas y opciones de publicación, soporte para almacenamiento en caché distribuido y menor superficie de instalación.

Por último se hace necesario añadir una imagen que permita realizar una comparación entre la primera y la última versión de esta herramienta. Aunque parece evidente que la verdadera remodelación ha sido por dentro, pues la estructura en si no se ha visto demasiado alterada.

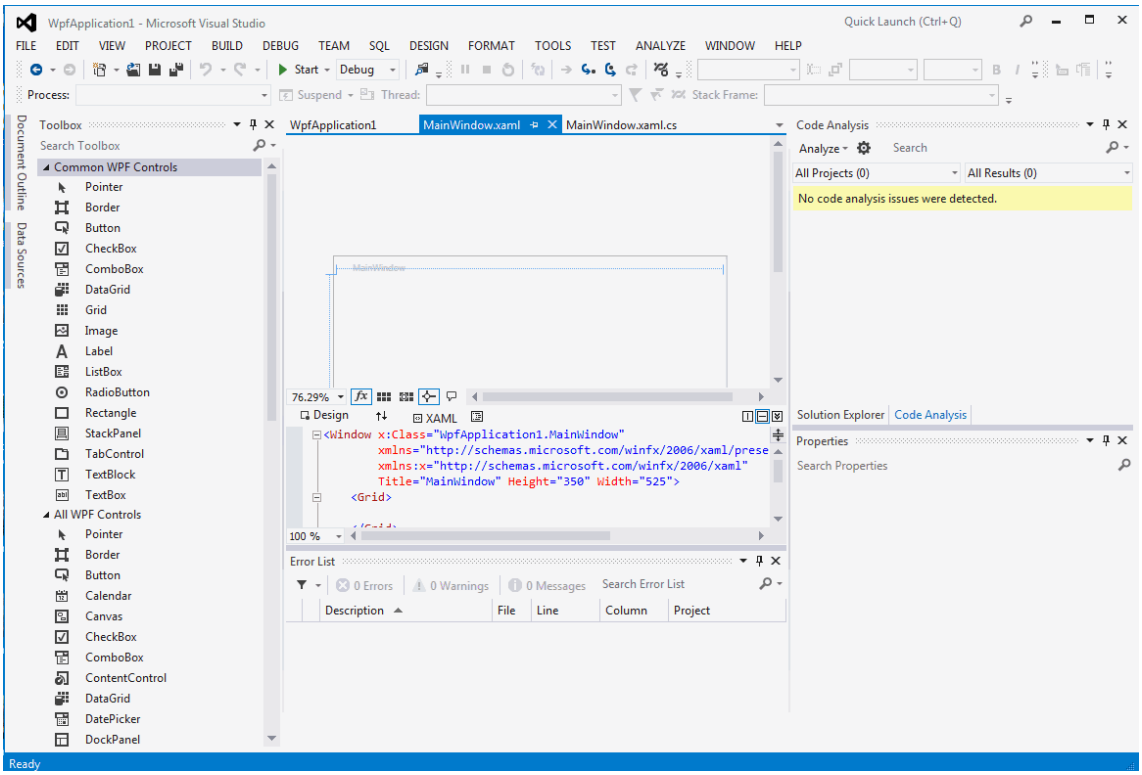

Figura 17: Visual Studio 2012 [18]

4.2.Lenguaje utilizado en el desarrollo (C#)

4.2.1. Introducción

El lenguaje utilizado para el desarrollo del proyecto tratado en este documento ha sido C#, por lo que a continuación se procederá a tratar las líneas fundamentales de dicho lenguaje.

C# [19] es un lenguaje de programación orientado a objetos, que fue desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Posteriormente sería aprobado como un estándar por la ECMA y la ISO. Además C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Utiliza un modelo de objetos de la plataforma .NET similar al de java, aunque incluye mejoras de otros lenguajes y su sintaxis básica deriva de C/C++.

El nombre de C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés Sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# [20] es superior a C/C++. Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Se debe destacar que ya existen compiladores implementados que proveen el marco Mono.DotGNU, que genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

4.2.2. Historia

Durante el desarrollo de la plataforma .NET, las bibliotecas de clases fueron escritas originalmente usando un sistema de código gestionado llamado Simple Managed C (SMC). En enero de 1999, Anders Hejlsberg formó un equipo con la misión de desarrollar un nuevo lenguaje de programación llamado Cool (C orientado a objetos). Este nombre tuvo que ser cambiado debido a problemas de marca, por lo que pasaría a llamarse C#. La biblioteca de clases de la plataforma .NET fue migrada entonces al nuevo lenguaje.

Hejlsberg lideró el proyecto de desarrollo de C#. Anteriormente, ya había participado en el desarrollo de otros lenguajes como turbo Pascal, J++ y Embarcadero Delphi.

4.2.3. Metas del diseño del lenguaje

El estándar ECMA-334 fija las siguientes metas en el diseño para el lenguaje C#:

- Aplicaciones económicas en cuanto a memoria y procesado.
- Soporte para internacionalización.
- Lenguaje de programación orientado a objetos simple, moderno y de propósito general.
- Portabilidad del código fuente.
- Adecuación para escribir aplicaciones de cualquier tamaño: desde las más grandes y sofisticadas como sistemas operativos hasta las más pequeñas funciones.
- Capacidad para desarrollar componentes de software que se puedan usar en ambientes distribuidos.
- Inclusión de principios de ingeniería de software tales como revisión estricta de los tipos de datos, revisión de límites de vectores, detección de intentos de usar variables no inicializadas y recolección de basura automática.
- Fácil migración del programador al nuevo lenguaje, especialmente para programadores familiarizados con C, C++ y Java.

4.2.4. Compiladores

En la actualidad existen diversos tipos de compiladores para el lenguajes C#. Son los siguientes:

- Delphi 2006, de Borland Software Corporation.
- Microsoft .NET Framework 2.0 (SDK) incluye un compilador de C#, pero no un IDE.
- Mono, es una implementación con licencia GNU GPL de todo el entorno .NET desarrollado por Novell. Como parte de esta implementación se incluye un compilador de C#.
- DotGNU Portable.NET, de la Free Software foundation.
- Microsoft Visual Studio, IDE por excelencia de este lenguaje.

4.3.La aplicación

El proyecto tratado en este documento consiste en el desarrollo de una aplicación que permita el acceso de un usuario a su cuenta a través de una validación mediante datos biométricos basados en la huella dactilar.

Para empezar se hace necesario aclarar que la aplicación implementada pretende simular una aplicación propia de un banco, de manera que una vez se haya accedido a la sesión se puedan realizar operaciones tales como consultar el saldo de la cuenta, ingresar dinero, sacar dinero etc. que posteriormente serán explicadas.

Para comprender perfectamente la aplicación tratada se hace necesario presentar un diagrama de flujo que recoja el funcionamiento del proyecto.

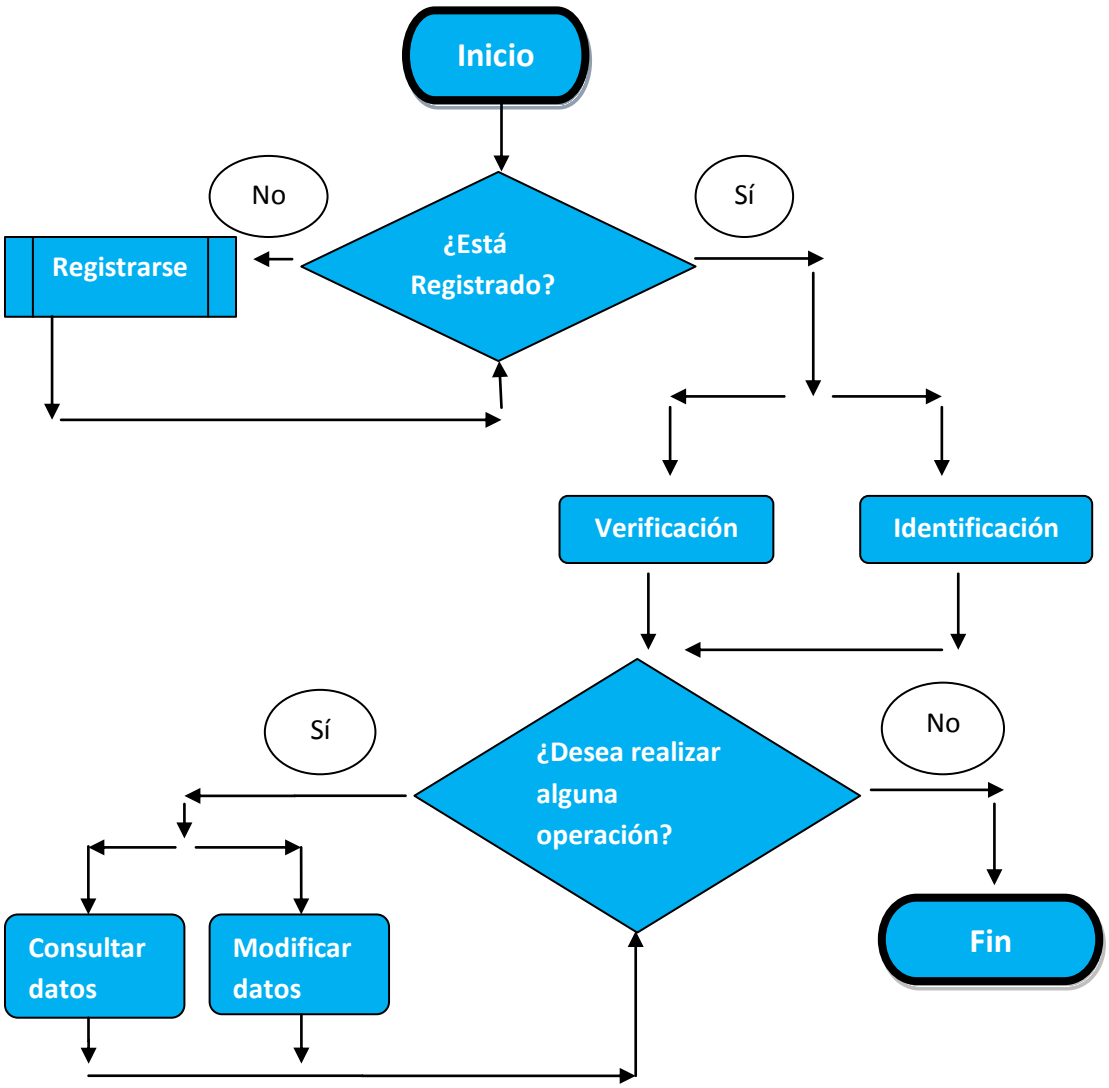

Figura 18: Diagrama de bloques

A su vez, esta aplicación está compuesta por dos grandes bloques, la parte de cliente y la parte de servidor. Las siguientes secciones explican cada uno de esos dos bloques.

4.3.1. Cliente

En este proyecto, en la parte del cliente se ha pretendido conseguir una aplicación muy visual, que permita al usuario adquirir un rápido control sobre ella, sin necesidad de poseer conocimientos sobre biometría o sobre lenguajes de programación.

Por si acaso el usuario presentase dudas sobre su funcionamiento, se ha incluido una ventana de ayuda, que pretende guiar a este a lo largo de todo el proceso.

La parte del cliente, únicamente se encargará de la recogida de datos, es decir tras capturar la huella, enviara los datos biométricos y biográficos (en caso de que los haya) al servidor. Únicamente en el proceso de registro, debido a que se le pide al usuario que realice una doble captura de su huella, el cliente realizará una comprobación entre estas dos muestras, a fin de comprobar que se hayan obtenido correctamente, y que efectivamente ambas corresponden al mismo usuario. Una vez hecho esto, enviará los datos de una de ellas al servidor en caso de que la comprobación sea correcta, o solicitará una repetición del proceso en caso de que esta sea incorrecta.

La otra ocasión en la que el cliente deberá procesar datos, será en caso de que se inicie la operación de identificación, donde el servidor devolverá una lista con hasta un máximo de 4 posibles candidatos. En este momento, el cliente iniciará un proceso a través del cual descartará a todos los posibles candidatos a excepción de uno (el que mayor similitud presente con la muestra enviada).

De este modo, en la parte del cliente se ha pretendido que la aplicación diseñada, facilite al usuario cualquier operación que desee realizar para controlar el crédito de su cuenta bancaria. Por ello se ha decidido crear una pantalla única para dichas operaciones, algo que inicialmente no se contemplaba en el desarrollo de este proyecto, ya que el objetivo final de esta aplicación, era conseguir un perfecto registro, validación e identificación del usuario a través de patrones biométricos y mediante un enlace cliente-servidor.

4.3.2. Servidor

En la parte del servidor, se ha pretendido conseguir una eficiencia, que permita desarrollar cualquier tipo de las funciones planificadas para este proyecto en un periodo relativamente corto de tiempo, de modo que el usuario no se sienta incómodo en ningún momento esperando a que acabe las diversas comprobaciones.

La funcionalidad principal del servidor será la de recibir datos del cliente (tanto biográficos como biométricos) y dependiendo del tipo de acción que se le haya indicado que realice, realizará alguna de las siguientes operaciones:

- Almacenará estos datos en un fichero que creará y al que le asignará el nombre del usuario introducido por la persona que pretende registrarse en la aplicación, colocándolo en una carpeta cuyo nombre debe ser “usuarios”.

- Comparará dichos datos con los almacenados en los ficheros creados anteriormente, de manera que el individuo sea aceptado o rechazado en la aplicación.
- Eliminará o modificará los datos (tanto biográficos como biométricos) de un determinado fichero, creado anteriormente.
- Devolverá una serie de datos solicitados por el cliente, de algún fichero específico, con el fin de que este pueda mostrarlos por pantalla.

En este caso, y al contrario que en la parte del cliente, se ha pretendido conseguir un servidor robusto, resistente frente a posibles amenazas, a pesar de que ello conlleve una mayor complejidad del código. Esto es así, ya que el cliente no deberá comprender ni manipular en ningún momento el servidor, por lo que su complejidad le será totalmente transparente.

5. Desarrollo

Siguiendo la división realizada en la fase de diseño, el desarrollo de este proyecto se divide en dos grandes bloques. Por un lado se encuentra el bloque del cliente, donde se realizan algunas funciones a nivel local, como en el caso del registro, o el caso de la identificación como se verá posteriormente. Por otro lado estaría el bloque del servidor, que implementa los métodos necesarios para que el cliente a través de llamadas a estos métodos, pueda realizar la totalidad de funciones que demanda.

A grandes rasgos, la idea del proyecto es ver si se puede sustituir la tarjeta de crédito, por un simple paso del dedo por el sensor de captura de la huella.

5.1.Pantalla inicial

Nada más arrancar la aplicación aparece la siguiente ventana:

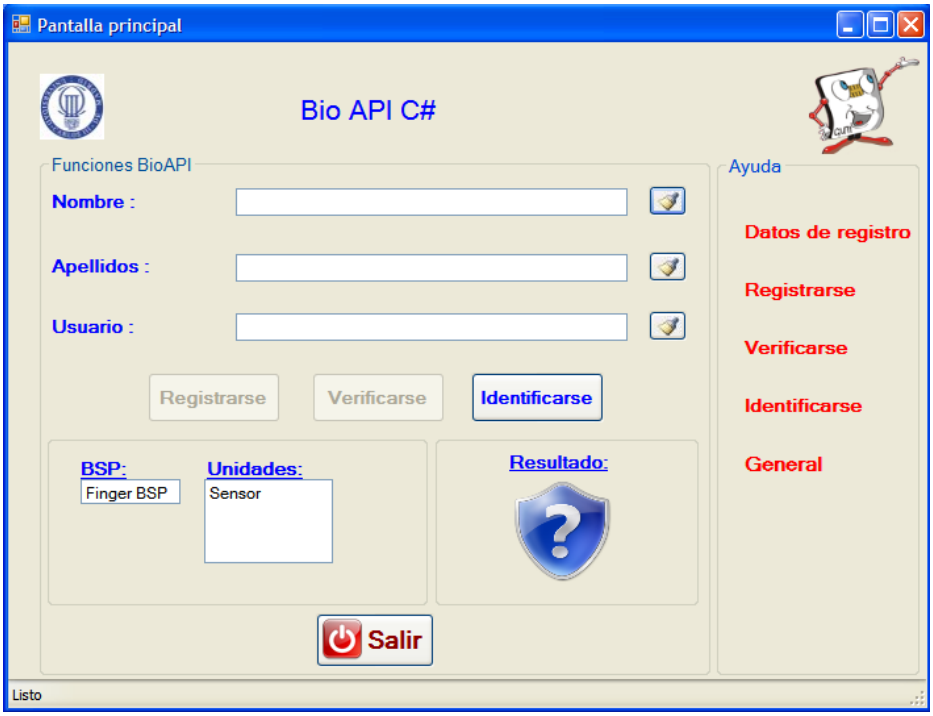

Figura 19: Pantalla inicial

Lo primero que se puede ver en ella son unos campos para introducir los datos con los que se será conocido en la aplicación (todos ellos van acompañados de unos botones con los que se puede limpiar el campo correspondiente).

A la derecha de estos, se encuentra una ventana de ayuda que cuenta con los siguientes campos:

- Datos de registro: Al pasar el ratón por encima mostrará la siguiente imagen:

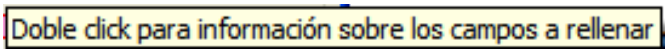

Figura 20: Campos a rellenar

Si se hiciese doble click sobre esta etiqueta, se accedería a la siguiente ventana:

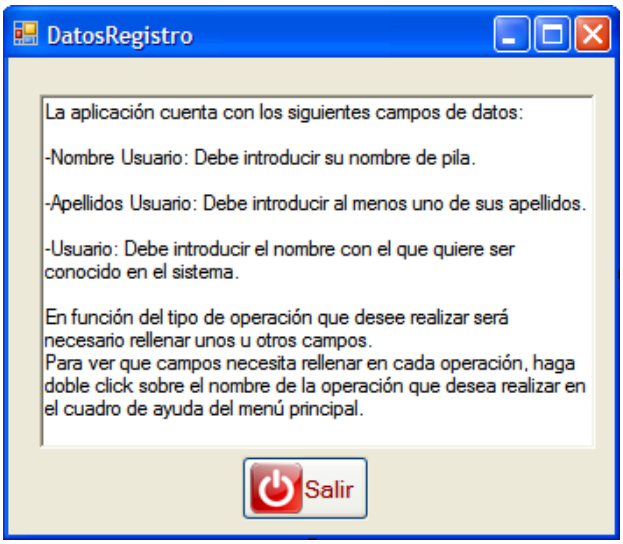

Figura 21: Información de datos de registro

En esta ventana se mostrará una descripción de los campos a rellenar quedando perfectamente identificados, así como una referencia al resto de ventanas de ayuda en función de la operación que se quiera realizar.

Por último se incluye un botón (Salir), que permitirá abandonar dicha ventana y volver a la pantalla principal.

- Registrarse: Al pasar el ratón por encima se mostrará el siguiente mensaje:

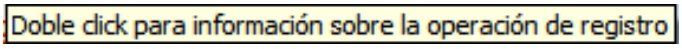

Figura 22: Información de registro

Si se hiciese doble click sobre esta etiqueta, se accedería a la siguiente ventana:

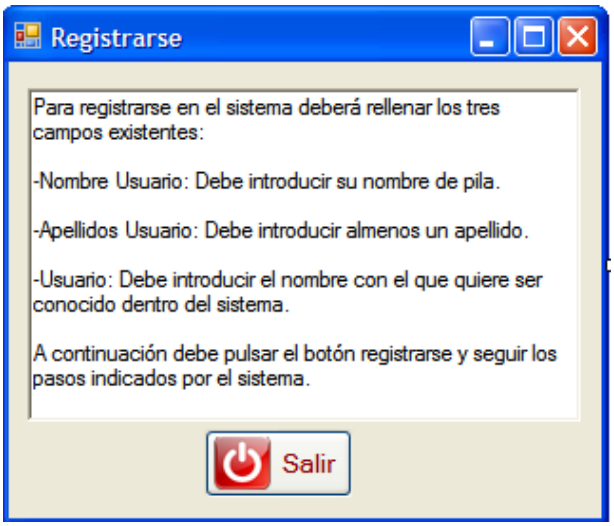

Figura 23: Información para registrarse

En esta ventana se puede ver que campos es necesario rellenar, así como el procedimiento a seguir para comenzar con la operación de registro.

Por último incorpora un botón (Salir), que permite abandonar dicha ventana y volver a la pantalla inicial.

- Verificarse: Siguiendo la misma estructura de las ventanas anteriores, al pasar el ratón por encima se mostrará un mensaje similar a los anteriores.

  Si se hiciese doble click sobre esta etiqueta, se accedería a la siguiente ventana:

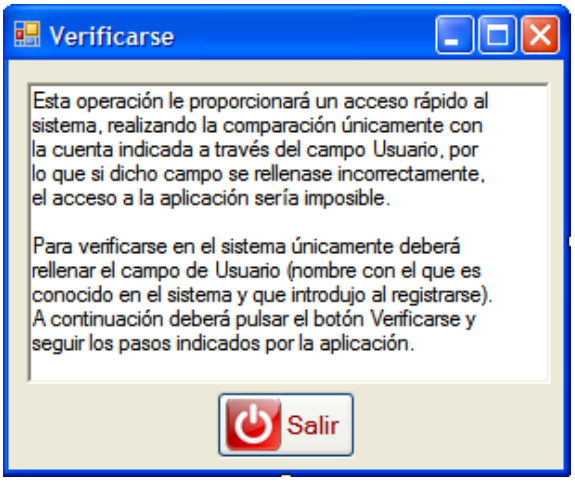

Figura 24: Información para verificarse

Al igual que el caso de registrarse, esta ventana indica los campos que son necesarios rellenar para verificarse en el sistema, así como el procedimiento a seguir para comenzar con la operación de verificación.

Por último incorpora un botón (Salir), que permite abandonar dicha ventana y volver a la pantalla inicial.

- Identificarse: Al igual que en los casos anteriores, al pasar el ratón por encima se mostrará un mensaje similar a los anteriores.

  En caso de hacer doble click sobre ella, se accederá a la siguiente ventana:

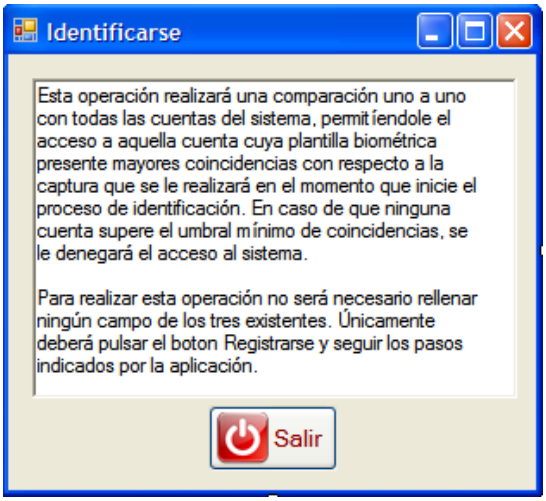

Figura 25: Información para identificarse

Al igual que en los casos anteriores, esta ventana indica los campos que son necesarios rellenar para identificarse en el sistema, así como el procedimiento a seguir para comenzar con la operación de identificación.

Por último incorpora un botón (Salir), que permite abandonar dicha ventana y volver a la pantalla inicial.

- General: Al pasar el ratón por encima de esta etiqueta se mostrará un mensaje similar a los anteriores.

  En caso de hacer doble click sobre ella aparecerá la siguiente ventana:

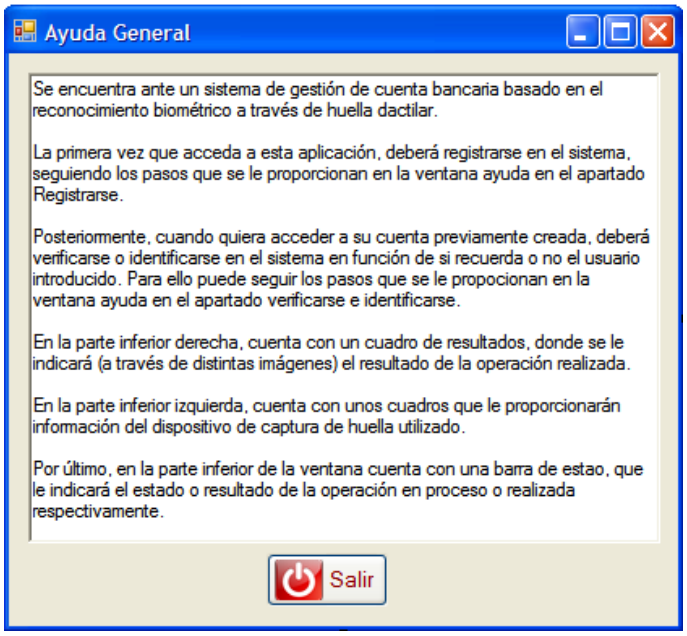

Figura 26: Información de la aplicación

En ella se puede ver una breve descripción tanto de la aplicación en sí, como de las diferentes opciones que nos proporciona la ventana inicial.

Por último se incluye un botón (Salir) que permite abandonar la ventana de ayuda y volver a la pantalla inicial.

En la parte inferior izquierda de la pantalla, se encuentran unos cuadros de información sobre el BSP y el dispositivo de captura de huella utilizado.

En caso de pasar el ratón por encima de la palabra FingerBSP, se mostrará un mensaje similar a los anteriores.

En caso de hacer doble click se abrirá una ventana como esta:

Figura 27: Información BSP

En caso de que el programa estuviese en ejecución, en ella se podría observar una descripción del BSP utilizado. Además cuenta con un botón (salir) que permite abandonar esta ventana y volver a la pantalla principal.

La otra opción que existe, sería pasar el ratón por encima de la palabra Sensor, apareciendo entonces un mensaje como los anteriores.

En caso de hacer doble click se abriría una ventana similar a esta:

Figura 28: Información de la unidad

En ella se puede observar una descripción del dispositivo de captura de huella utilizado. Además cuenta con un botón (salir) que permite abandonar esta ventana y volver a la pantalla inicial.

En la parte inferior derecha aparece un recuadro que indicará si el proceso de validación de los datos biométricos a comparar ha sido realizado con éxito o no.

- Si la operación ha concluido exitosamente, la imagen mostrada será la siguiente:

Figura 29: Icono indicativo de que la operación ha sido exitosa

  Esta imagen se acompañará de un mensaje en la barra de estado indicado la aceptación de la operación.

- Si la operación no ha conseguido concluir correctamente, la imagen mostrada será la siguiente:

Figura 30: Icono indicativo de que la operación ha sido errónea

Acompañada además de un mensaje que se mostrará en la barra de estado, y que informará del error producido.

Por último se ha introducido un botón (salir) con el que se podrá abandonar la aplicación cuando se desee.

Tras esta descripción de la ventana inicial se procederá a tratar cada una de las operaciones que permite realizar esta aplicación.

5.1.1. Registro

Será la operación que habrá que realizar la primera vez que se acceda a la aplicación.

Para poder Registrarse en el sistema, será necesario rellenar todos los campos que aparecían en la ventana anterior (nombre, apellidos y usuario). Una vez que se pulse el botón Registrar, el sistema lanzará una ventana emergente como esta:

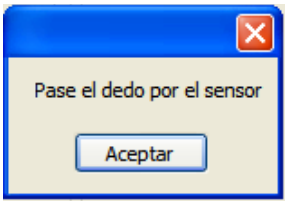

Figura 31: Petición de captura de huella para registro

Este proceso será realizado dos veces, a fin de que en el propio cliente (en el ordenador propio del usuario por ejemplo) se realice una validación interna para comprobar que el usuario ha pasado el dedo correctamente por el lector, y la plantilla de minucias generada es correcta, ya que en caso de solo realizar la captura una vez si esta no fuese correcta y se generase una plantilla errónea, la cuenta de dicho usuario quedaría inutilizada. Esto es así ya que una vez se cerrase sesión en ella, nunca más se podría volver a iniciar sesión, debido a que el algoritmo de comparación, nunca devolvería un valor superior al umbral fijado (en este caso 40), es decir nunca se reconocería a ese usuario como el titular de la cuenta.

Para toda la parte de captura de la huella y su posterior creación de la plantilla de referencia (será almacenada para realizar las comprobaciones posteriores en caso de querer acceder al sistema de nuevo) se han utilizado las librerías tanto del BSP_UPEK (tipo de sensor utilizado), como el BSP_Huella (contiene los algoritmos necesarios para realizar esta plantilla y sus posteriores comparaciones). Técnicamente sería más correcto decir que únicamente se utilizan

las librerías de BSP_Huella, pero estás a su vez realizan llamadas a las funciones del BSP_UPEK, para informarle por ejemplo de que es necesario realizar una captura.

Una vez se haya comprobado en el cliente, que el proceso de captura ha sido correcto, se procederá a mandar todos los datos del usuario (tanto biográficos como biométricos) al servidor (simula el ordenador de un banco), a través de las funciones previamente implementadas SetBiographicData y SetBiometricData. Una vez el servidor haya recibido la información, procederá a crear un fichero donde escribirá todos los datos del individuo (utilizando la función CreateSubject), asignado como nombre a dicho fichero el usuario del sujeto. Además le asignará un identificador único de manera que quede perfectamente definido de forma inequívoca dentro del sistema. Para ello se ha utilizado el siguiente fragmento de código:

```
string identificador = Guid.NewGuid().ToString();
```

Este código genera un string de 16 bytes (128 bits) aleatorios, por lo que será considerado como un identificador único.

Por último, se ha implementado otra función (crearCuentaBancaria) en el servidor que añada un apartado dentro del fichero, denominado cuenta bancaria (inicialmente tendrá el valor 0), de manera que se puedan realizar operaciones simulando una cuenta bancaria como las que se explicarán posteriormente en el apartado 5.2.

Además se ha querido dotar a la aplicación de una funcionalidad doble, ya que se ha implementado de manera que pueda ser utilizada para aplicaciones centradas en la persona o en el número de visita, por lo que inicialmente se le asignará un número de visita a dicha información, que posteriormente podrá ser sobrescrita (modelo centrado en la persona) o añadir otro número de visita y completarlo con nueva información (modelo centrado en el número de visita). Inicialmente, la aplicación será utilizada como modelo centrado en la persona, ya que un modelo centrado en el número de visita no tiene demasiada utilidad en este tipo de aplicaciones (cuenta bancaria).

Así, el fichero creado tendría la siguiente estructura:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <usuario>
  - <encounter>
      <apellidos>delgado</apellidos>
      <value>1</value>
      <nombre>sergio</nombre>
      <datos_biométricos>AAEAAAD/////AQAAAAAAAAAMAgAAAD5ib3pvcnRoLCBWZXJzaW9uPTE
      <imagen_huella>6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6Ojo6
    </encounter>
    <identificadorUnico>eb1e7aa7-83e4-4430-9dd7-13216719934c</identificadorUnico>
    <cuenta_bancaria>0</cuenta_bancaria>
</usuario>
```

Figura 32: Fichero

Donde se puede apreciar perfectamente el nombre y apellidos del usuario que ha creado la cuenta.

Por otro lado aparece el campo "value" que indica el número de visita en el que se centraría el sistema (como ya se comentó anteriormente, debido a que la funcionalidad utilizada es la centrada en la persona, el número de visita siempre será 1) en caso de utilizar una aplicación centrada en el número de visita.

Además se ven los campos "cuenta_bancaria" que indica el saldo de la cuenta, e "identificadorUnico" que es el número que identifica inequívocamente al usuario dentro del sistema, como ya se explicó anteriormente.

También aparecen los campos "datos_biométricos" que es donde se almacena la plantilla biométrica creada en base 64, y que en esta ocasión se ha decidido recortar, debido a su larga extensión (2600 caracteres) y "imagen_huella", que será donde se almacene la imagen de la huella correspondiente, a fin de poderla mostrar en ocasiones futuras y que al igual que el campo anterior también ha sido recortado, debido a su excesiva longitud.

Por último, el servidor añadirá dicho fichero a una galería denominada con el nombre "usuarios" a través de la función AddSubjectToGallery, de manera que la información de este quede almacenada para visitas posteriores.

Por último es necesario añadir que tras muchos esfuerzos se ha conseguido que esta aplicación pueda trabajar con diferentes tipos de sensores de captura de huellas, ya que se modificaron las funciones del estándar de BIOAPI (sin romper ninguna norma) para que el método captura devolviese el tamaño en alto y ancho de la imagen al final del campo "BiometricData". Así gracias al siguiente fragmento de código implementado justo antes de cada comparación de la huella adquirida con la plantilla de referencia (ya sea en el cliente para la función registrarse o en el servidor para las funciones de verificación e identificación) se consiguió poder crear un objeto tipo IBIR, con el alto y ancho adecuado de la imagen capturada en función del tipo de sensor, que posteriormente se pasaría a las funciones de verificación e identificación para realizar la comprobación.

A continuación se muestra el código con el que se ha conseguido lo anteriormente explicado

```
Finger_CaptureResult huella =
(Finger_CaptureResult)cliente.myBSPSesion.Sensor.Capture(null, null, null, -1,
null);//Se almacena en huella la captura realizada

byte[] w1 = { 0, 0 };
w1[0] = huella.CapturedBIR.BiometricData[0];//Se obtiene el primer
byte del ancho de la imagen
w1[1] = huella.CapturedBIR.BiometricData[1];//Se obtiene el segundo byte del
ancho de la imagen

byte[] h1 = { 0, 0 };
h1[0] = huella.CapturedBIR.BiometricData[2];//Se obtiene el primer byte del alto
de la imagen
```

```
h1[1] = huella.CapturedBIR.BiometricData[3];//Se obtiene el segundo byte el alto
de la imagen

int resultW11 = Convert.ToInt32(w1[0]);//Se convierte a int el primer byte del
ancho
int resultW12 = Convert.ToInt32(w1[1]);//Se convierte a int el segundo byte del
ancho
int resultW1 = resultW11 + resultW12;//Su suman ambos números para obtener el
ancho correcto de la imagen

int resultH11 = Convert.ToInt32(h1[0]);//Se convierte a int el primer byte del
alto
int resultH12 = Convert.ToInt32(h1[1]);//Se convierte a int el segundo byte del
alto
int resultH1 = resultH11 + resultH12;//Se suman ambos números para obtener el
alto correcto de la imagen

int longitudImagen = huella.CapturedBIR.BiometricData.Length - 4;//creamos un
número con la longitud de la huella capturada (se le resta 4 para quitar los
datos de altura y anchura añadidos anteriormente)

byte[] biometrico1 = new byte[longitudImagen];//Se crea un array de byte de la
longitud anterior
Array.Copy(huella.CapturedBIR.BiometricData, 4, biometrico1, 0,
longitudImagen);//Se copia en ese array la captura realizada eliminando las
últimas 4 posiciones que pertenecian al alto y ancho de la imagen

IBIR imag1 = cliente.registro(biometrico1, resultW1, resultH1);//se construye el
objeto IBIR
```

Tras finalizar el proceso de registro se pasa a una nueva pantalla que simulará la cuenta bancaria, y que será tratada al finalizar la descripción de las dos operaciones restantes.

5.1.2. Verificación

Es una de las dos opciones que se tienen una vez el usuario ha sido registrado y se ha salido de la aplicación si posteriormente quiere volver a entrar. A diferencia de la operación registro, esta consta solo de un método (VerifySubject) y únicamente es necesario rellenar el campo de Usuario. La operación realizada sería similar a preguntar al sistema: ¿Soy el usuario X?

Una vez pulsado el botón de verificación se pedirá que se pase el dedo por el sensor de captura de huella. En este caso, esta operación solo se realizará una vez, debido a que en caso de que la nueva huella no se capture adecuadamente, siempre se puede repetir el proceso rápidamente, mientras que en el método de registro, se produciría un gran problema en la aplicación, si la plantilla creada fuese errónea y no identificara al sujeto correspondiente.

Tras la captura de la muestra biométrica, esta es enviada al servidor, donde se realiza una comprobación entre la muestra biométrica que se acaba de obtener y la plantilla de referencia, a la que se accederá gracias al conocimiento del usuario, por lo que se realizaría

una búsqueda por todos los ficheros existentes, hasta dar con aquel cuyo nombre coincida con el usuario, y se recuperaría su plantilla biométrica.

Si la comprobación es exitosa se iniciará sesión en la aplicación y se pasará a la pantalla de la cuenta bancaria correspondiente al igual que tras el proceso de registro. En este caso se hace necesario recuperar el identificador único asignado a dicho usuario, para poder realizar diversas operaciones que se explicarán a continuación. Para ello se ha desarrollado un método en el servidor que proporcione dicho identificador. Este método consiste en una comparación de todos los nombres de los ficheros existentes (como se dijo antes dichos nombres son el usuario introducido por el sujeto en el proceso de registro) con el usuario proporcionado al pulsar el botón de verificación (único campo obligatorio a rellenar en este caso). Una vez se encuentre el fichero correcto, se accederá a su interior, donde está almacenado el identificador único.

En caso de que la comparación no sea correcta, se contemplan dos casos principales:

- El primero se produciría, si la persona que está intentando acceder a la cuenta pasase mal el dedo por el sensor, o intentase suplantar la identidad de otra persona, en cuyo caso la aplicación detectaría que la huella proporcionada no corresponde a la plantilla almacenada y asociada a la cuenta que se quiere acceder. En este caso se mostrará una imagen como esta:

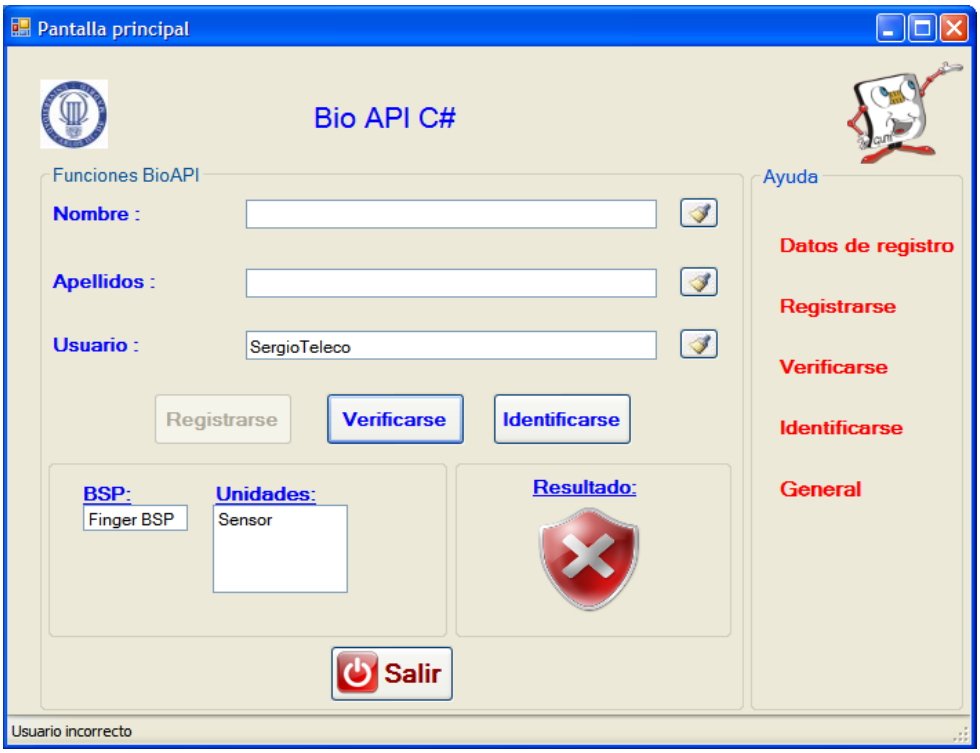

Figura 33: Usuario incorrecto tras verificación

- El segundo, se produciría si el usuario introducido no existe, es decir el individuo no recuerda su usuario inicial e introduce otro incorrecto, o lo recuerda pero lo escribe mal, en cuyo caso el sistema no detectaría ningún fichero con dicho nombre, por lo que se mostraría una imagen como la siguiente:

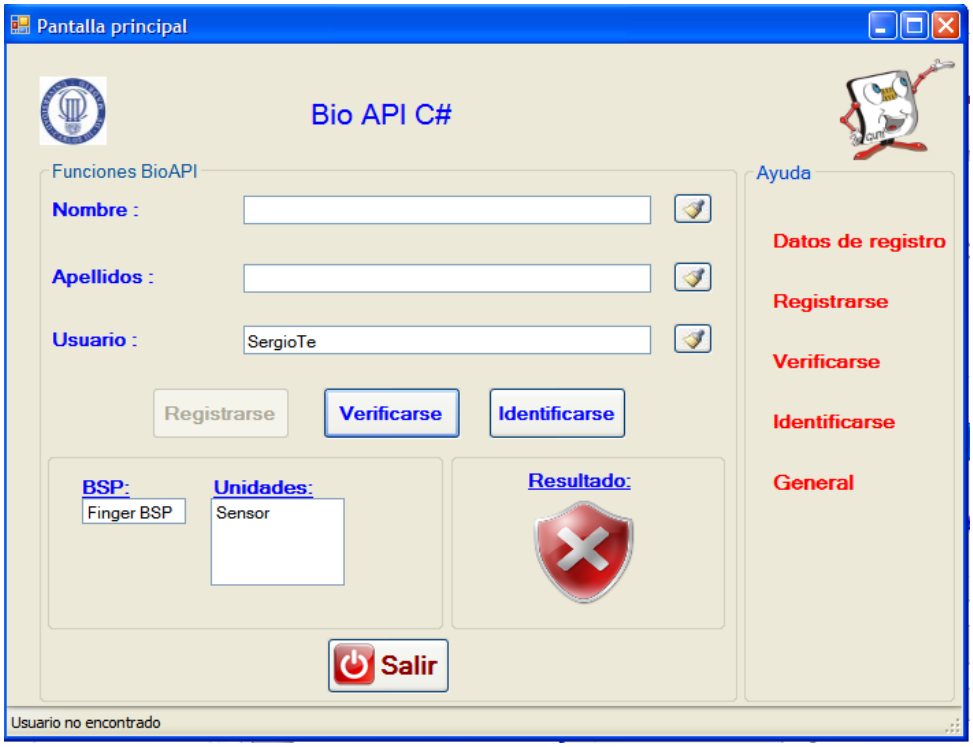

Figura 34: Usuario no encontrado tras verificación

Como se puede ver la única diferencia sería el mensaje mostrado en la parte inferior de la pantalla (barra de estado), donde se describe el error producido.

También se contemplan otros casos de error como que no se encuentre la carpeta donde se almacenan los ficheros, o que el sistema cree mal la plantilla de la huella, en cuyo caso el mensaje de error mostrado sería distinto. No obstante no se plasmarán imágenes de todos los posibles errores, ya que son muchos, y de muy rara aparición, ya que para que no se encontrara la carpeta, alguien debería haber manipulado el ordenador del servidor manualmente y haberla borrado o cambiado de sitio, o para que el sistema crease mal la plantilla debería producirse un error interno en el código, algo que no se contempla siempre y cuando el código no sea manipulado por personas ajenas a su creador.

5.1.3. Identificación

Es la segunda opción que se tiene, una vez el usuario ha sido registrado y se ha salido de la aplicación si posteriormente quiere volver a entrar.

Al igual que el proceso de verificación, en este solo es necesario implementar un método (IdentifySubject) y no es obligatorio rellenar ningún campo. La operación realizada en este caso sería como preguntar al sistema: ¿Quién soy?

Al igual que en el proceso de verificación, una vez se pulse el botón de identificación se pedirá que se pase el dedo una única vez por el sensor, siendo los motivos los mismos que en el caso anterior.

Tras la captura de la nueva muestra biométrica, esta es enviada al servidor, donde se realizará una comparación muestra a muestra con las plantillas de referencia de todos los ficheros existentes. Una vez hecho esto, se devolverá al servidor una lista de posibles candidatos que tendrá un tamaño determinado por la aplicación utilizada (4 en este caso). Esta lista contendrá tanto los datos biográficos de dichos usuarios, los datos biométricos, y la puntuación obtenida en el proceso de comparación de plantillas (cuanto más alta sea más parecidos se habrán encontrado).

En el cliente tras recibir dicha lista y dado que en nuestra aplicación únicamente nos interesa un candidato, se ha realizado un método, en el que se recorre la lista recibida para quedarnos con los datos del usuario con una mayor puntuación, para iniciar sesión en la cuenta de dicho usuario. El código es el siguiente:

```
int i = 0;
double aux = 0;
string nombre = "";
string apellidos = "";
int longitudLista =
responseidenti.IdentifySubjectResponsePackage.IdentifySubjectResult.CandidateList
.Count();//Almacenamos la longitud de la lista devuelta por la función
IdentifySubjectResult

for (i = 0; i < longitudLista; i++)//recorremos todas las posiciones de la lista
(máximo 4 candidatos)
{
    double valor =
    responseidenti.IdentifySubjectResponsePackage.IdentifySubjectResult.Candidate
    List[i].Score;//Almacenamos el valor de la comparación realizada entre la
    muestra biométrica y la plantilla perteneciente al candidato i de la lista

    if (valor > aux)//Si el valor almacenado anteriormente es mayor que el que
teniamos (inicialmente es 0) se entra en la operación
    {
         nombre =
         responseidenti.IdentifySubjectResponsePackage.IdentifySubjectResult.Cand
         idateList[i].BiographicData.FirstName;//Recuperamos el nombre
         apellidos =
         responseidenti.IdentifySubjectResponsePackage.IdentifySubjectResult.Cand
         idateList[i].BiographicData.LastName;//Recuperamos los apellidos

         aux = valor;//asignamos al comparador auxiliar dicho valor
    }
}
```

No obstante, es posible que el sistema devolviese una lista vacía, ya que no encontrase similitud entre la huella y ninguna de las plantillas del sistema, en cuyo caso se mostraría una imagen similar a la siguiente:

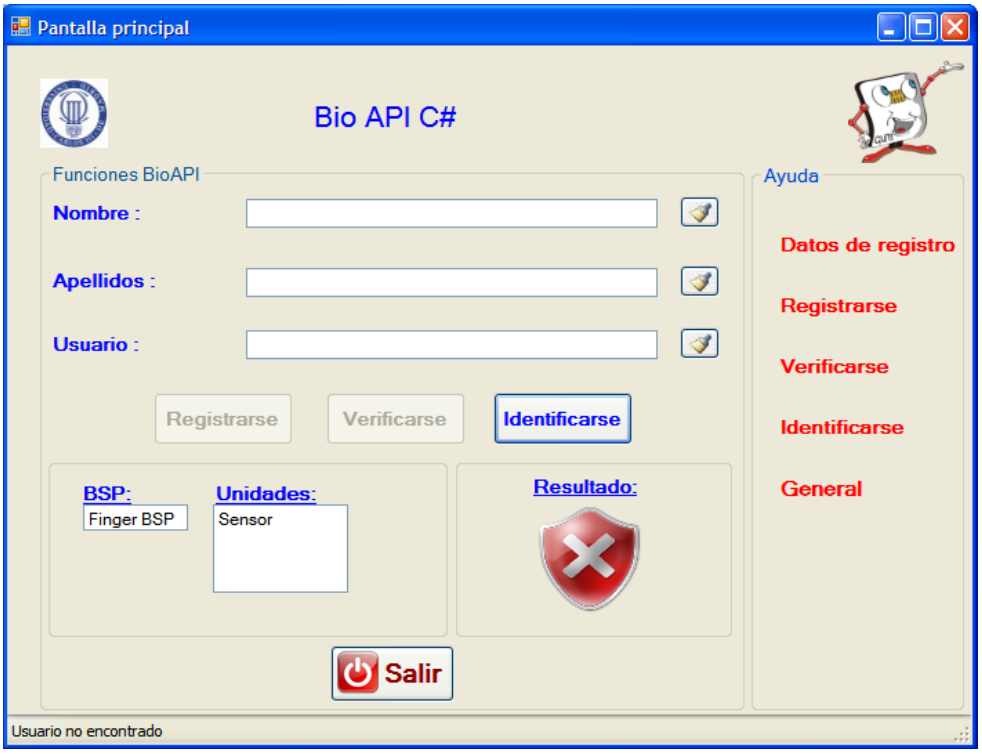

Figura 35: Usuario no encontrado tras identificación

Donde como se puede ver en la barra de estado se especifica que el usuario no ha sido encontrado.

Al igual que en el caso de verificación, es necesario recuperar el identificador único de dicho sujeto. Sin embargo en este caso no se cuenta con el usuario de este individuo, por lo que se ha creado una nueva función algo más compleja que devuelva dicho identificador conociendo únicamente los datos biográficos del sujeto. Para ello se recorrerán todos los ficheros leyendo de su interior los datos biográficos de cada uno y comparándolos con los que se han obtenido del sujeto con mayor puntuación de la lista.

Por último se accederá a la pantalla de la cuenta bancaria, estando en la sesión del usuario con una mayor puntuación, es decir de aquel usuario cuya plantilla se asemejaba más a la muestra biométrica proporcionada.

5.2.Pantalla de la cuenta bancaria

Una vez se haya iniciado sesión por cualquiera de los métodos explicados anteriormente, el proceso a seguir será el mismo para los tres. De este modo aparecerá la siguiente pantalla:

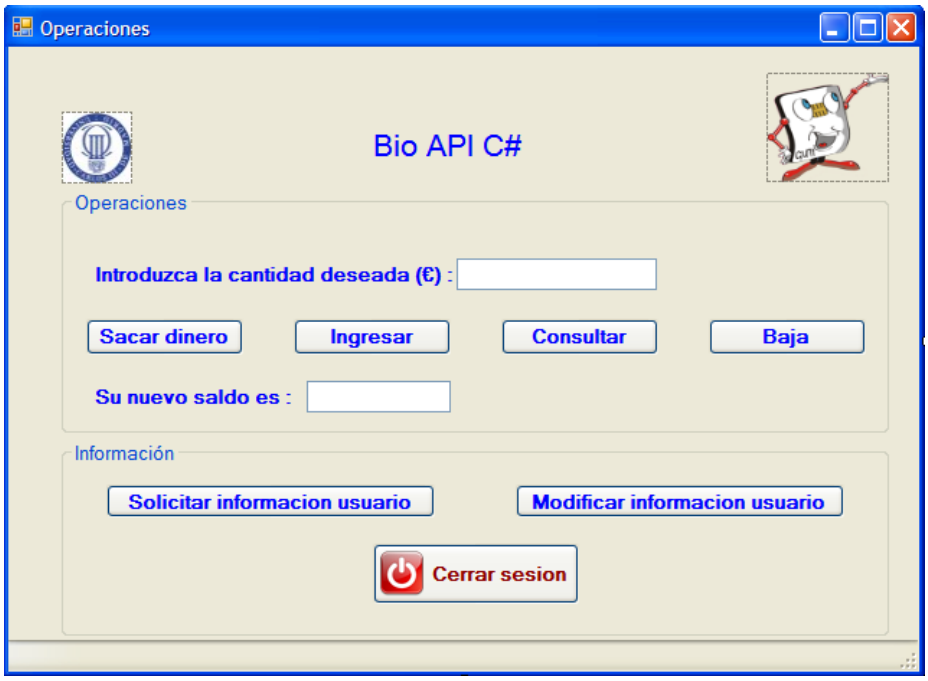

Figura 36: Operaciones

Dentro de esta pantalla se deben dividir en dos las operaciones que se pueden realizar:

- Operaciones bancarias: Dentro de este apartado, debemos incluir operaciones tales como:
  - Consultar el estado de cuenta: En la que el cliente realizará una llamada al servidor para acceder al campo cuenta bancaria del fichero adecuado y ver así el saldo con el que cuenta.
  - Ingresar dinero: En la que el cliente realizará una llamada al servidor para acceder al campo cuenta bancaria del fichero correspondiente y sumarle al saldo existente la cantidad deseada a ingresar.
  - Sacar dinero: En la que el cliente realizará una llamada al servidor para acceder al campo cuenta bancaria del fichero correspondiente y restarle al saldo existente la cantidad deseada. En caso de que la cuenta de dicho usuario no disponga de crédito suficiente se le avisará mostrando un mensaje como el siguiente:

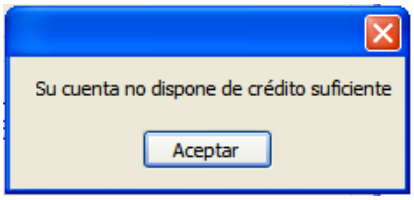

Figura 37: Crédito insuficiente

En caso de que el usuario pulse los botones ingresar o sacar sin haber especificado ninguna cantidad, se le avisará con un mensaje como el siguiente:

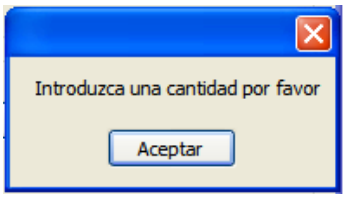

Figura 38: Error

Todas estas operaciones, son posibles debido a que el usuario que introdujo el sujeto cuando se registró en el sistema por primera vez es conocido independientemente de la forma en la que se haya iniciado sesión (registro, verificación o identificación). Para localizar el fichero adecuado se realizará una búsqueda por todos los existentes en el sistema, de manera que se localice aquel cuyo identificador único coincida con el de dicho usuario, para posteriormente acceder a su campo "cuenta_bancaria" y modificar el dato y mostrarlo por pantalla (ingresar o sacar dinero), o simplemente leerlo (consultar el estado de cuenta).

- Operaciones de datos de usuario: Dentro de este apartado se incluirán operaciones como:
  - Cerrar sesión: tras pinchar este botón, se abandonará inmediatamente esta pantalla, y volveremos a la pantalla inicial, donde se podrá cerrar la aplicación, o realizar de nuevo alguna de las operaciones citadas anteriormente (registro, verificación o identificación).
  - Baja: en la que el fichero de datos del usuario será borrado del sistema, de manera que no quede ningún rastro de información de este, y automáticamente se cerrará la sesión, debido a que el usuario en este momento ya no se encontraría registrado en el sistema.
    Esta operación se realizará llamando a la función DeleteSubject, implementada en el servidor. Esta función realiza una búsqueda para localizar el fichero adecuado (búsqueda similar a las anteriores conociendo el usuario) y ejecutar la siguiente línea de código:

    ```
    File.Delete('nombreFichero');
    ```

    De manera que elimina cualquier rastro de dicho fichero, y por lo tanto de la cuenta del usuario.

- Adicionalmente la aplicación cuenta con dos botones más. Estos serían:
  - Información de usuario: Tras pinchar en este botón se abrirá una nueva ventana:

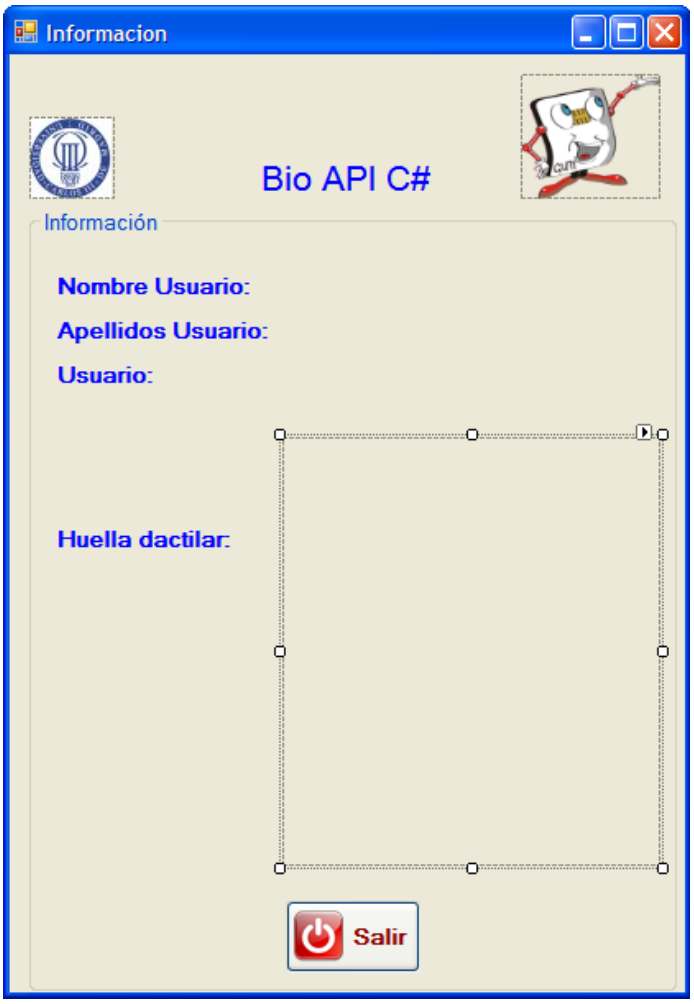

Figura 39: Información de usuario

En esta imagen se puede ver cómo el sistema muestra toda la información que posee de la cuenta:

1. El nombre y apellidos del titular de la cuenta: Para ello se hará uso de la función RetrieveBiographicData, que tras localizar el fichero adecuado en el sistema devolverá un objeto con varios campos, entre ellos el nombre y el apellido del titular de dicha cuenta, de manera que a la derecha de los campos “Nombre Usuario” y “Apellidos” Usuario aparecerán los del titular de la cuenta nada más cargar dicha ventana.
2. El usuario: Para ello se ha creado una nueva función (ya que el estándar no contemplaba ninguna apropiada) en el servidor, que accederá a los ficheros del sistema, y tras localizar el adecuado, devolverá un string con el nombre del fichero, al que se le abrá eliminado la extensión .xml gracias al siguiente fragmento de código:

```
string result = Path.GetFileName('nombreFichero');//Se
recupera el nombre del fichero
string resultado = result.Replace(".xml", "");//Se elimina
la extensión .xml
```

De este modo a la derecha del campo "Usuario" aparecería el usuario introducido por el individuo cuando creó la cuenta nada más cargar la ventana anteriormente mostrada.

3. La huella de dicho usuario, que se ubicará en el recuadro de la parte inferior de la pantalla, gracias a la función RetrieveBiometricData, cuyo funcionamiento será similar a RetrieveBiographicData (explicada anteriormente), pero devolviendo los datos biométricos del usuario (tanto la imagen como la plantilla de comparación).

Además cuenta con un botón (salir) que al igual que en el resto de pantallas que lo posean permitirá abandonar dicha pantalla y volver a la anterior en cualquier momento.

- Modificar información de usuario: Tras pinchar en este botón se abrirá una nueva pantalla:

Figura 40: Modificar información de usuario

En esta imagen se puede ver cómo es posible modificar o eliminar del sistema cualquier tipo de información tanto biométrica como biográfica perteneciente al usuario. En este caso las opciones que se tienen son las siguientes:

1. Borrar información biométrica: Para desarrollar esta tarea se ha utilizado la función definida en el estándar con el nombre de DeleteBiometricData, que realiza una búsqueda por todos los ficheros del sistema, y una vez encuentre el adecuado, eliminará cualquier dato

introducido en los campos “datos_biométricos” e “imagen_huella”. Se debe destacar que hay que tener especial cuidado con la eliminación de este tipo de datos, ya que en caso de no sustituirlos por unos nuevos, la cuenta quedaría inutilizada, debido a que la plantilla de referencia ha sido eliminada, y por lo tanto no se podrán realizar las comparaciones pertinentes para poder acceder a dicha cuenta en intentos futuros. Por lo tanto en caso de que el usuario decida realizar este tipo de operación, se le informará con un mensaje como el siguiente:

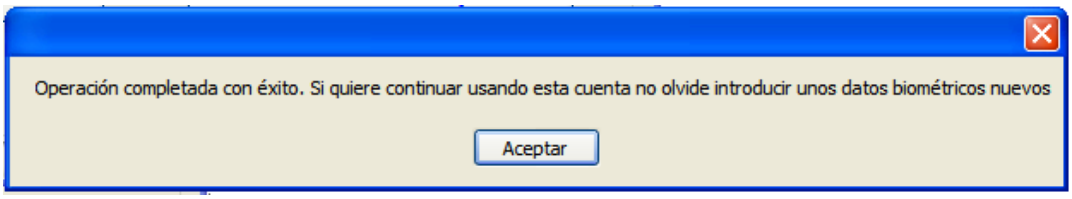

Figura 41: Petición de inserción de datos biométricos

2. Borrar información biográfica: Para ello se utilizará la función definida en el estándar con el nombre de DeleteBiographicData, que realizará una búsqueda por todos los ficheros del sistema, y una vez encuentre el adecuado, eliminará cualquier dato introducido tanto en el campo nombre como apellidos. Debemos destacar que hay que tener cuidado con este tipo de operaciones, ya que en caso de borrar dichos datos y no introducir unos nuevos, la información de esa cuenta quedará incompleta, por lo que se podrían producir errores en futuros accesos y quedar por tanto dicha sesión inutilizada. Por ello, en caso de que el usuario decidiese borrar sus datos biográficos, se le informaría de que debe introducir unos nuevos con un mensaje como este:

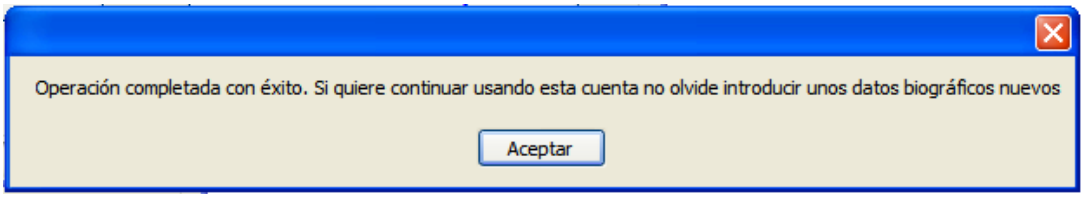

Figura 42: Petición de inserción de datos biográficos

3. Actualizar datos biométricos: Para realizar esta tarea se ha utilizado la función definida en el estándar como UpdateBiometricData. Esta es una operación compleja, ya que en primer lugar se debe realizar un proceso similar al de la primera vez que entramos en la aplicación, es decir se capturarán los datos dos veces, a fin de que la plantilla elaborada sea correcta, y posteriormente, a través de la función antes

indicada, se recorrerán los ficheros del sistema en busca del correcto. Una vez encontrado se sustituirán los datos biométricos anteriores, por los obtenidos recientemente.

4. Actualizar datos biográficos: En este caso la función utilizada ha sido la definida como UpdateBiographicData, con la que se recorrerán de nuevo los ficheros del sistema en busca del correcto. Una vez localizado se sustituirán los datos de los campos nombre y apellidos, por los nuevos que el usuario ha introducido en los campos de la parte superior derecha de la imagen anterior. Dado que el sistema no puede funcionar únicamente con el nombre o los apellidos del usuario, en caso de que éste rellene solo uno o ningún campo y pulse sobre el botón actualizar datos biográficos, se mostrará un mensaje por pantalla como este:

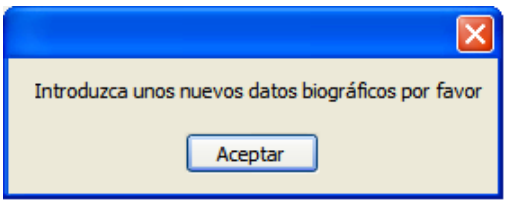

Figura 43: Actualizar datos biográficos

Por último se hace necesario añadir que las funciones de ListBiographicData y ListBiometricData, que confieren a la aplicación un nivel de conformidad 4 según el estándar seguido han sido implementadas y probadas, pudiendo asegurar que su funcionamiento es correcto. No obstante no se utilizan en el desarrollo de la aplicación, ya que estas están más orientadas a una aplicación centrada en el número de visita, en la que al necesitar mostrar los datos (tanto biográficos como biométricos) por pantalla, se puedan mostrar todos los datos ordenados por su número de visita, mientras que las operaciones de RetrieveBiometricData y RetrieveBiographicData únicamente mostrarían los datos de la última captura, o en caso de una aplicación centrada en la persona como es esta, mostrarían los datos existentes.

6. Pruebas

Para comenzar se probarán las tres funciones básicas del programa, para posteriormente continuar con las pruebas sobre las operaciones de simulación de una cuenta bancaria.

6.1.Registro

6.1.1. Prueba de usuario genuino

Para comenzar este apartado, se simulará que la persona que accede a la aplicación es la primera vez que lo hace, y por tanto no se encuentra familiarizada con la aplicación, por lo que requerirá de la utilización de la ventana de ayuda.

Una vez arrancada la aplicación el usuario se encontrará ante una pantalla como la siguiente:

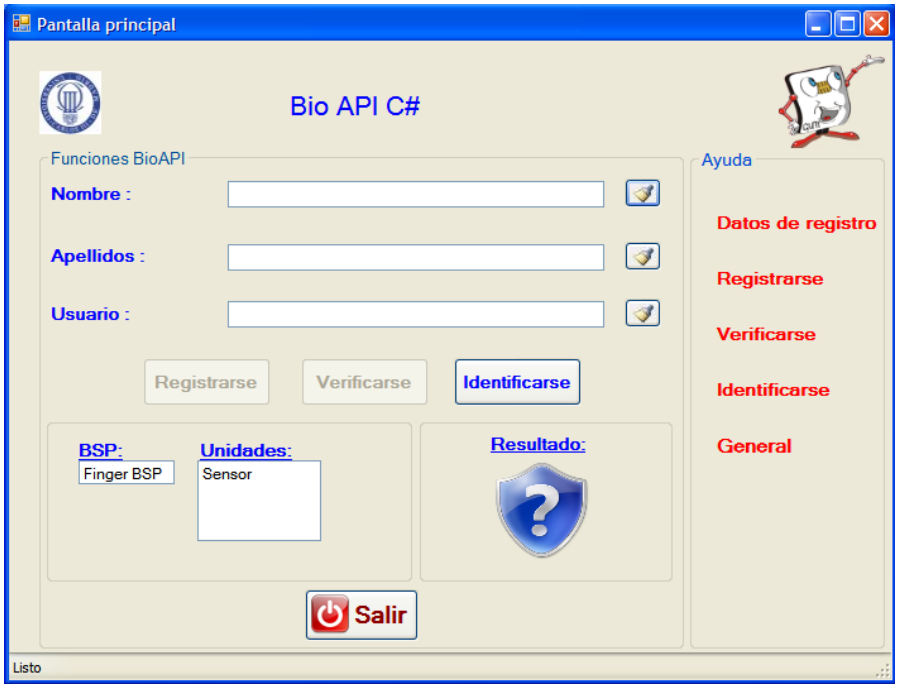

Figura 44: Usuario genuino (pantalla inicial)

Para comenzar el registro, y dado que el usuario no posee conocimientos previos de este tipo de aplicación, su primer impulso será consultar la ventana de ayuda, por lo que al pasar el ratón por encima de la opción “Registrarse” de dicha ventana, se le indicará que haga doble click sobre él para poder acceder a su contenido. Si el usuario sigue dichos pasos se encontrará con una ventana como la siguiente:

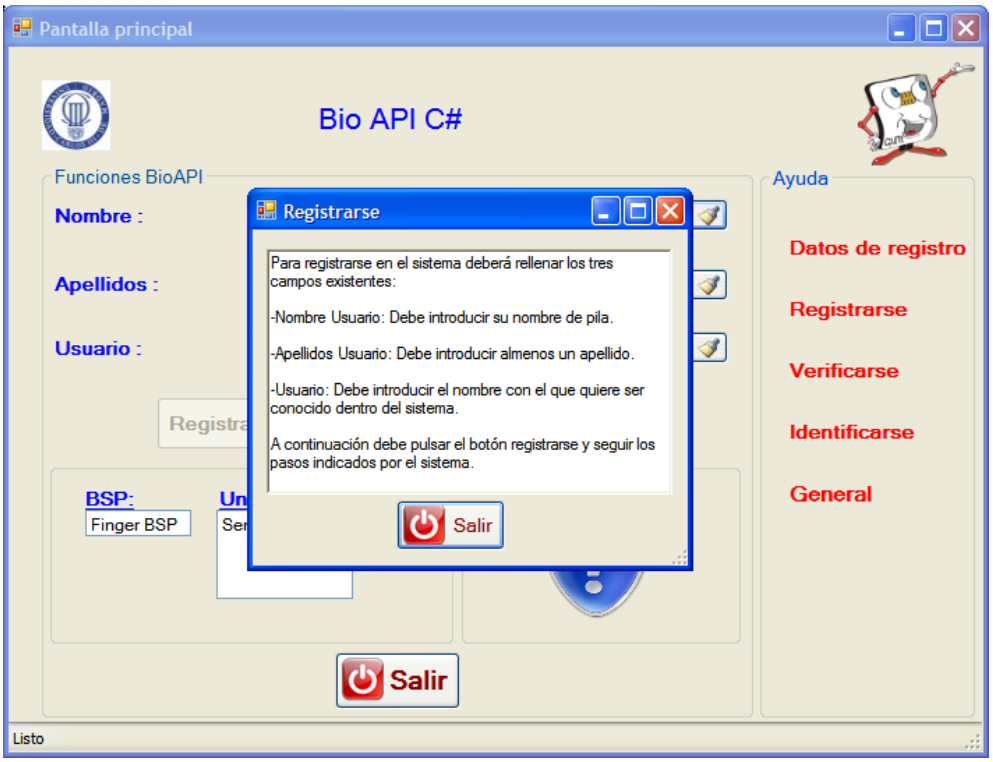

Figura 45: Usuario genuino (información de registro)

Donde se puede leer perfectamente los pasos que se deben seguir para comenzar con el registro.

Una vez el usuario ha entendido perfectamente el funcionamiento de la aplicación, procederá a rellenar los campos “Nombre”, “Apellidos” y “Usuario”, para a continuación presionar el botón Registrarse.

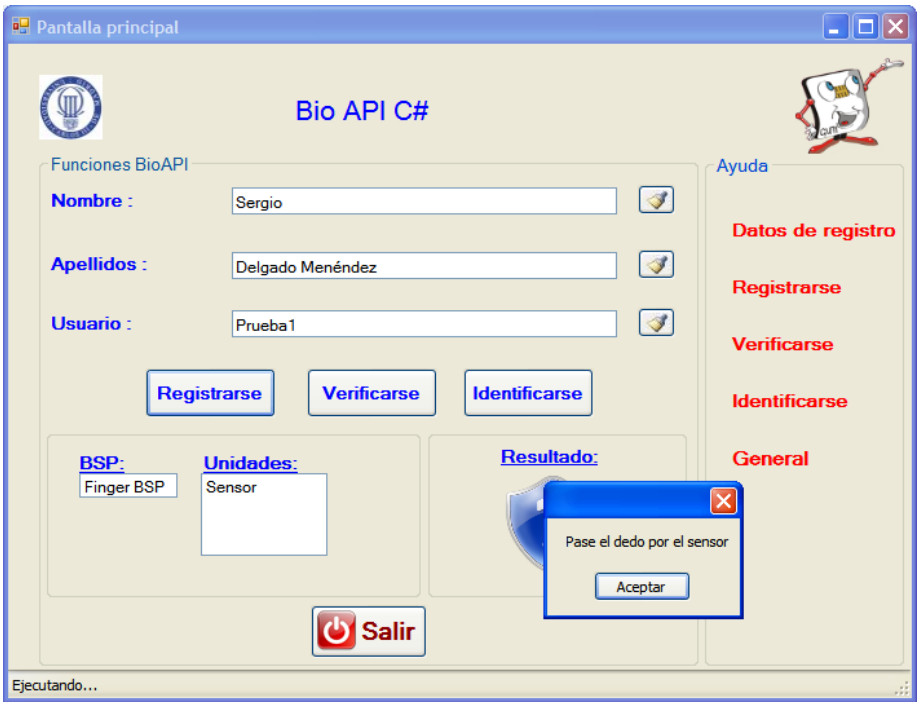

Figura 46: Usuario genuino (petición de captura)

De este modo, se puede ver como se le pide al usuario que pase el dedo por el sensor de captura de huella, a fin de realizar la primera de las dos capturas necesarias para el proceso de registro.

La siguiente imagen muestra el momento de la captura de la huella:

Figura 47: Usuario genuino (captura)

Así, cuando el usuario complete el proceso de captura de huella, se le mostrará una pantalla como la siguiente:

Figura 48: Usuario genuino (operaciones)

Lo que querrá decir que el proceso de registro del usuario ha terminado, y ya se le ha creado una cuenta, a la que podrá acceder en futuras ocasiones.

Como se puede ver en la siguiente imagen, el sistema crea un fichero en el que almacena toda la información introducida por el usuario en el registro:

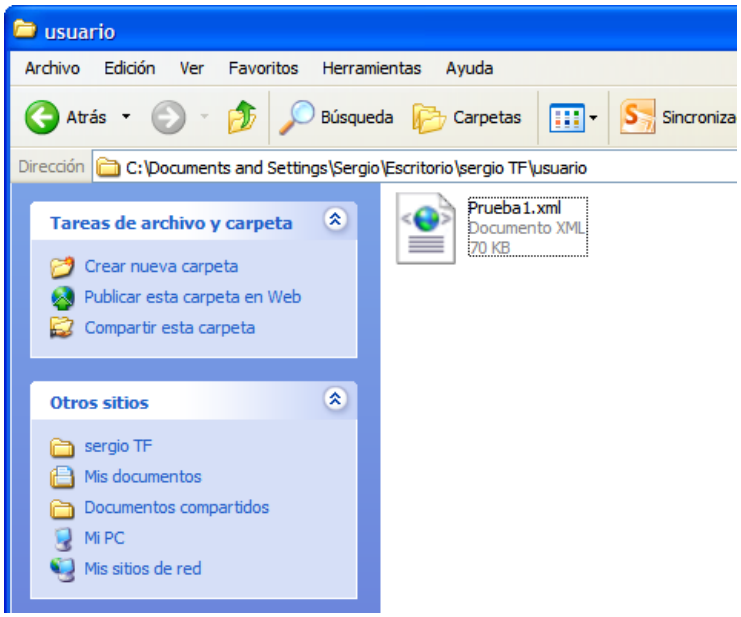

Figura 49: Usuario genuino (ubicación de fichero)

Si se accediese al contenido de dicho fichero, la imagen obtenida sería esta:

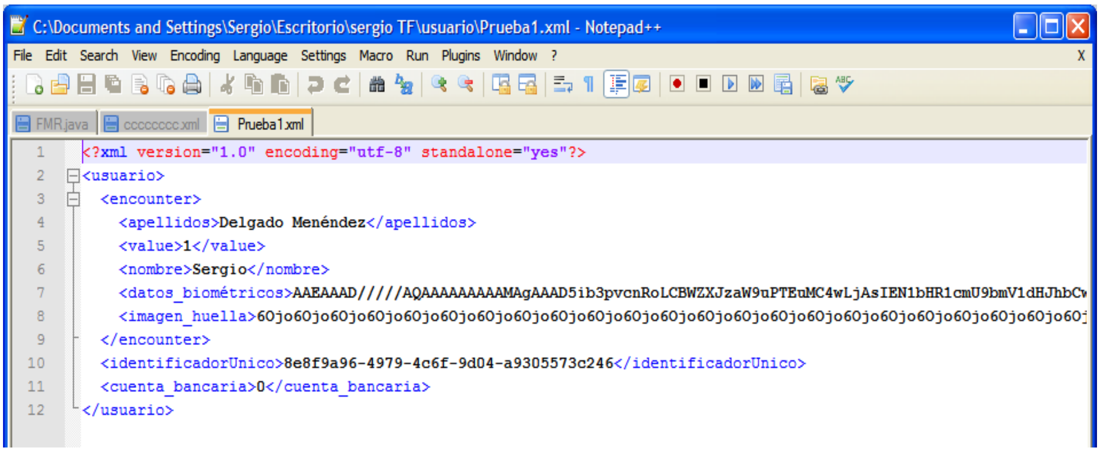

Figura 50: Usuario genuino (fichero)

Donde se puede ver como la información ha sido escrita correctamente, y al campo destinado a la cuenta bancaria se le ha asignado por defecto el valor 0.

De este modo finalizaría el registro del usuario en el sistema.

6.1.2. Prueba de captura de huella fallida

En caso de que el usuario decidiese registrarse y tuviese algún problema físico en la huella dactilar que impidiese su correcta captura, o simplemente se equivocase y en cada proceso de captura proporcionase un dedo diferente, el resultado obtenido sería el siguiente:

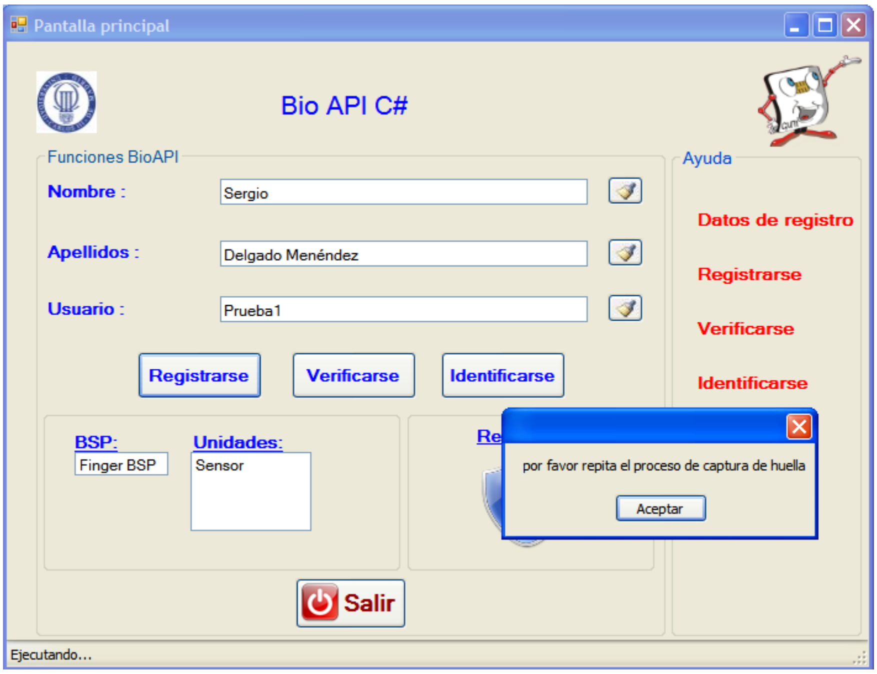

Figura 51: Captura de huella fallida

Como se puede ver, esta es una aplicación muy guiada, que no cuenta con graves defectos, a excepción del presentado en la prueba siguiente.

6.1.3. Prueba de aborto de proceso

En caso de que el usuario decidiese abortar el proceso de registro y pulsar el botón de cerrar de la ventana que indica que se debe pasar el dedo por el sensor, el sistema se queda colgado como se puede ver en la imagen siguiente:

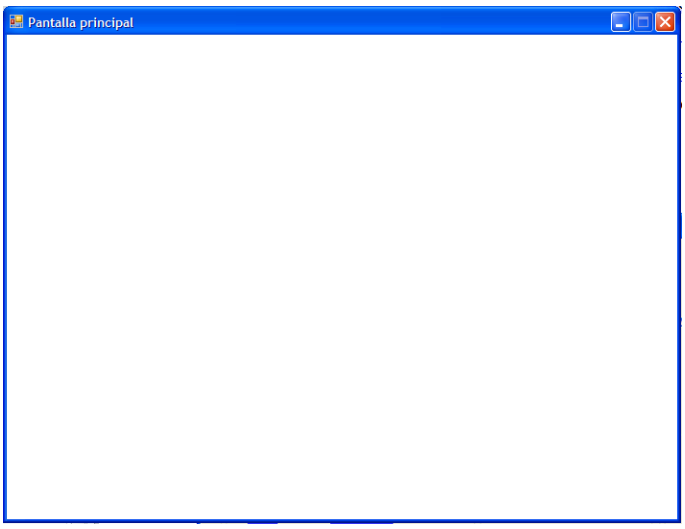

Figura 52: Proceso abortado

No obstante es necesario aclarar que este es un error totalmente ajeno al diseñador de esta aplicación, ya que es un error interno del algoritmo utilizado en el proceso de captura de huella, por lo que esta será una de las medidas propuestas en mejoras futuras.

6.2.Verificación

6.2.1. Prueba de usuario genuino

Una vez el usuario se ha registrado, si posteriormente decidiese volver a acceder a su cuenta, debería verificarse (siempre y cuando recuerde su usuario).

Para ello rellenaría el campo “Usuario” tal y como se indica en la ventana de ayuda para el proceso de verificación:

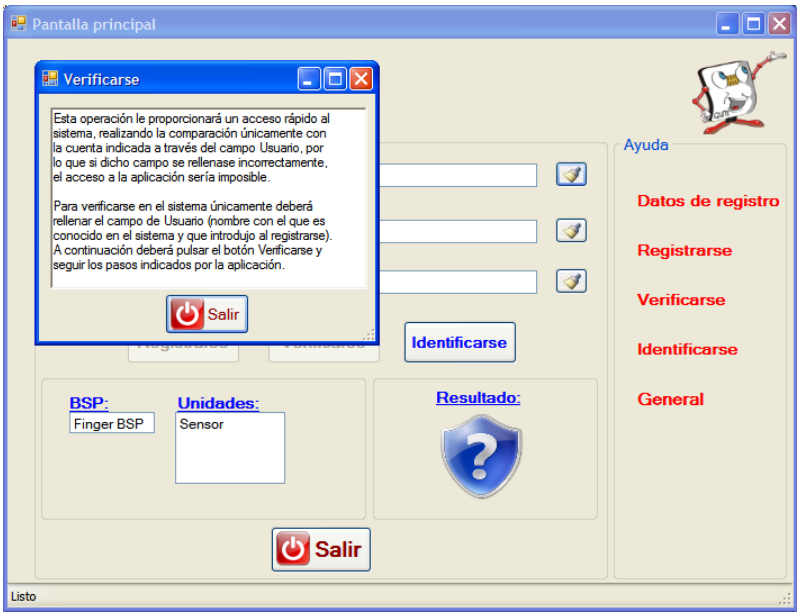

Figura 53: Usuario genuino (verificarse)

Tras rellenar el campo correspondiente y pulsar el botón Verificarse, el sistema pedirá que se realice una captura de huella, a fin de comprobar esta nueva muestra con la almacenada en el servidor:

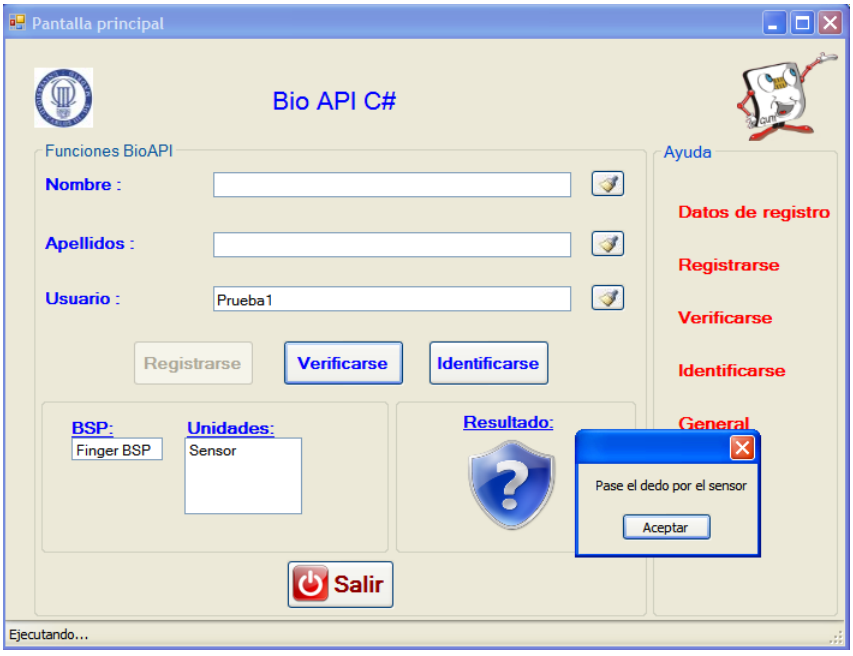

Figura 54: Usuario genuino (petición de captura para verificación)

En caso de que la verificación de la huella sea correcta, se mostrará la misma imagen que se mostró una vez el usuario había completado el registro, y en la que se pueden realizar diversas operaciones bancarias que posteriormente serán probadas.

6.2.2. Verificación de huella incorrecta

En caso de que el usuario pasase el dedo de forma incorrecta por el sensor, pasase un dedo diferente al que pasó en el registro, o simplemente el sensor se encontrase muy sucio, el resultado obtenido sería el siguiente:

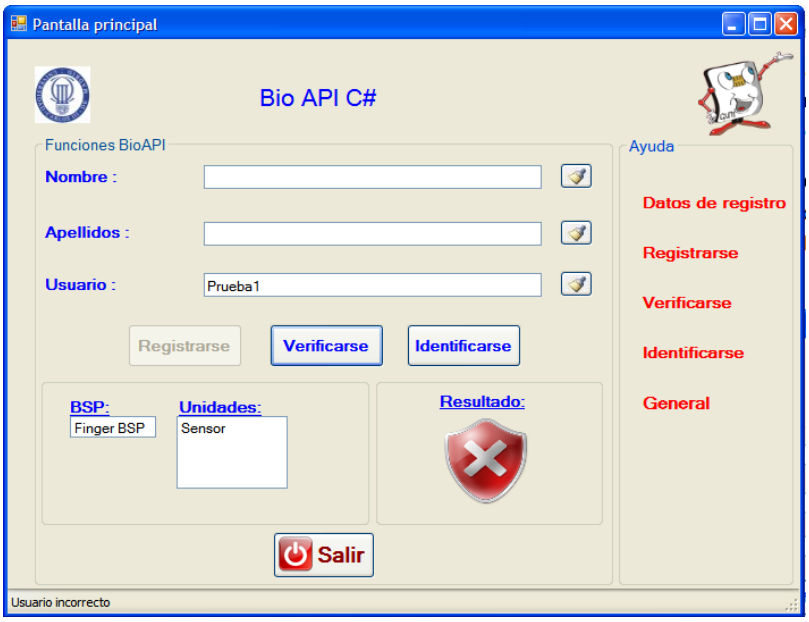

Figura 55: Usuario incorrecto tras verificación

Donde se puede apreciar perfectamente en el cuadro de resultados, que la comprobación ha sido incorrecta, proporcionándose además una pequeña descripción del error producido en la parte inferior izquierda de la pantalla (“Usuario incorrecto”).

En caso de que la captura de la huella fuese realizada correctamente, pero el individuo introdujera mal el Usuario, el resultado obtenido sería el mismo que el anterior como se puede ver en la siguiente imagen:

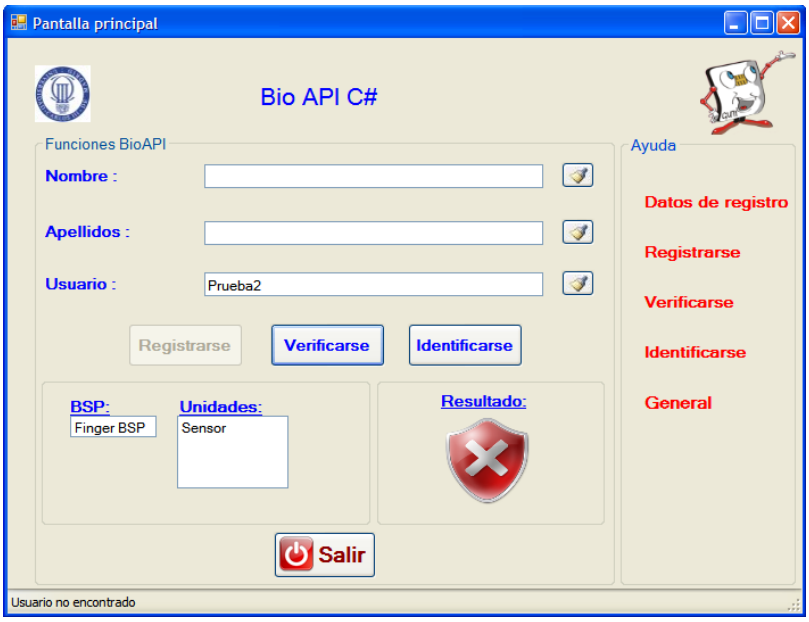

Figura 56: Usuario no encontrado tras verificación

A diferencia del mensaje mostrado en la parte inferior izquierda de la pantalla, donde en este caso se puede apreciar que el error producido ha sido no encontrar el usuario indicado.

6.2.3. Manipulación del servidor incorrecta

En caso de que alguien manipulase el servidor de manera incorrecta, ya sea borrando archivos, desplazándolos, o simplemente modificándolos, cuando el usuario intentase acceder a la aplicación mediante una verificación de su huella dactilar, el resultado obtenido sería el siguiente:

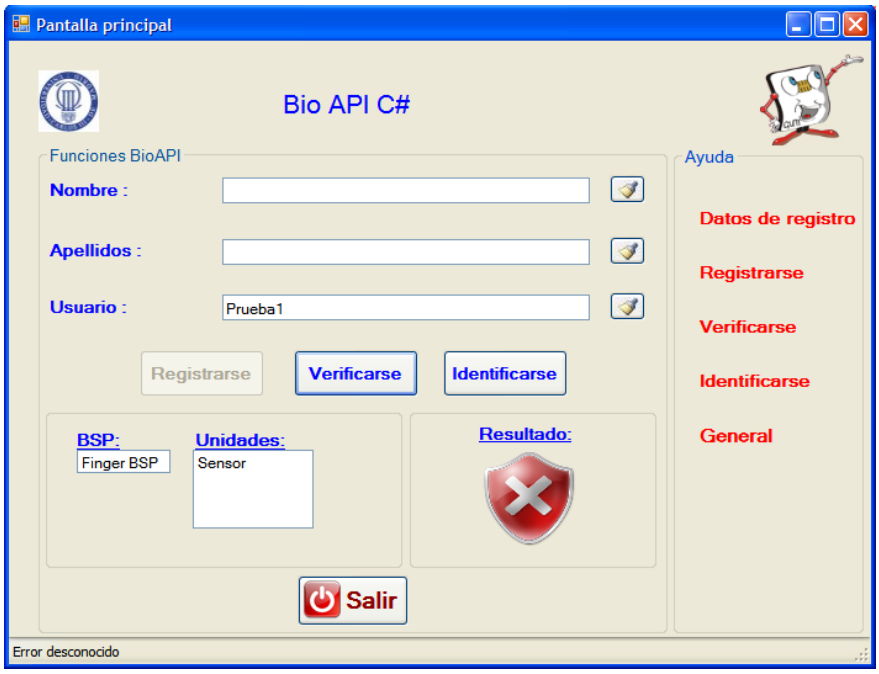

Figura 57: Error desconocido tras verificación

Donde aparece una ventana similar a la obtenida en el caso anterior, pero con la diferencia de que el mensaje proporcionado es el de "Error desconocido".

6.3.Identificación

6.3.1. Prueba de usuario genuino

En caso de que el individuo no recordase el usuario introducido en el proceso de registro, podría continuar accediendo a la aplicación, a través de la opción Identificarse.

En caso de que el usuario no conociese el procedimiento de esta funcionalidad, abriría de nuevo la ventana de ayuda de dicho proceso, obteniéndose la siguiente imagen:

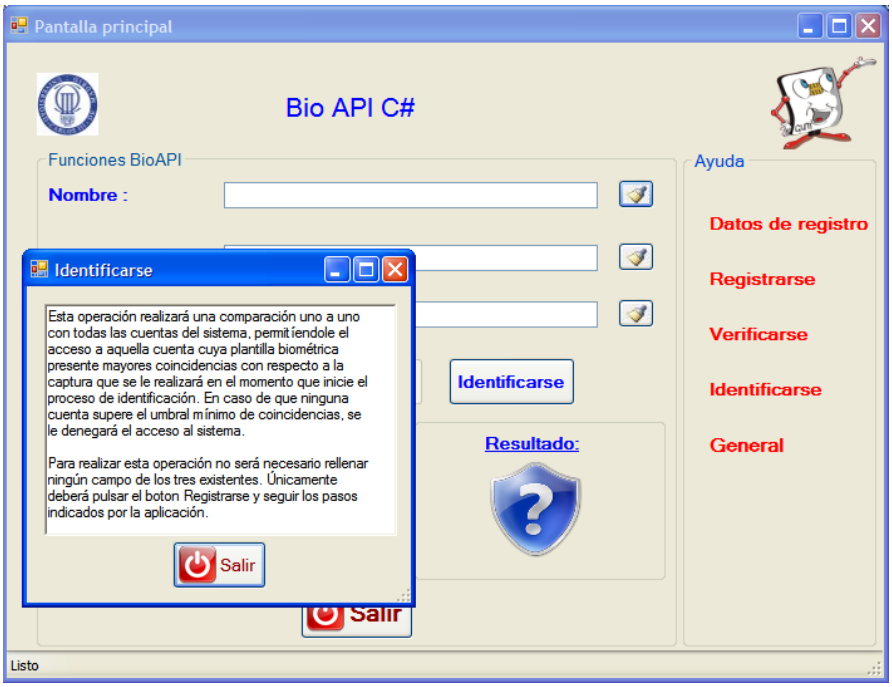

Figura 58: Usuario genuino (identificarse)

Una vez el usuario ha entendido perfectamente el funcionamiento de esta parte de la aplicación, accionaria el botón Identificarse, de manera que se le volvería a pedir de nuevo una captura de huella:

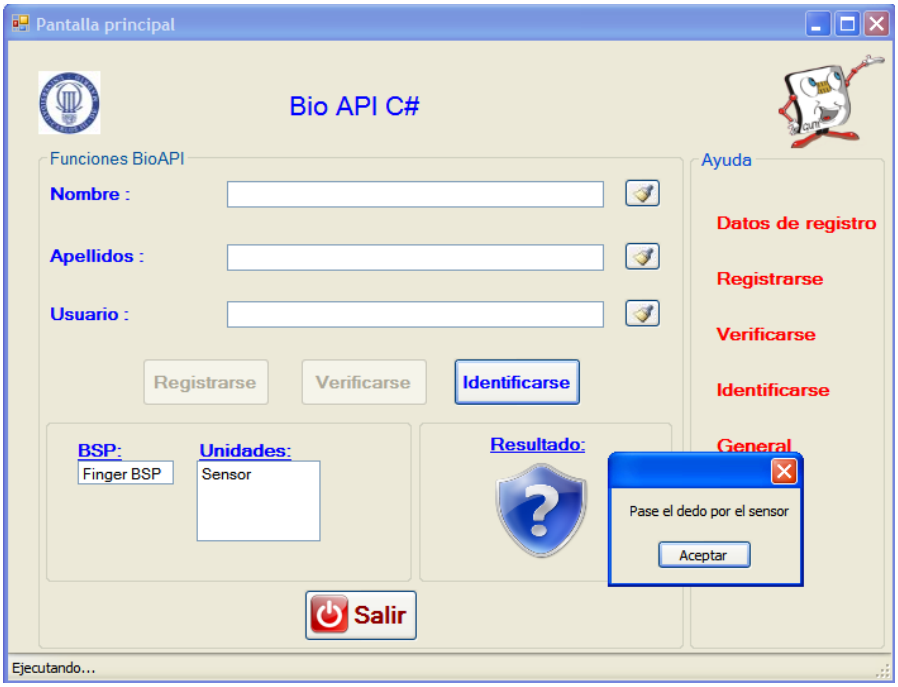

Figura 59: Usuario genuino (petición de captura para identificación)

Y tras comparar dicha muestra con todas las del sistema, se le permitirá el acceso a su cuenta, apareciendo de nuevo la pantalla anterior sobre operaciones bancarias.

6.3.2. La huella proporcionada no coincide con ninguna del sistema

En caso de que el usuario intentase acceder al sistema, y la captura realizada no fuese correcta, o simplemente dicho usuario no tenga una cuenta en el sistema, el resultado obtenido sería el siguiente:

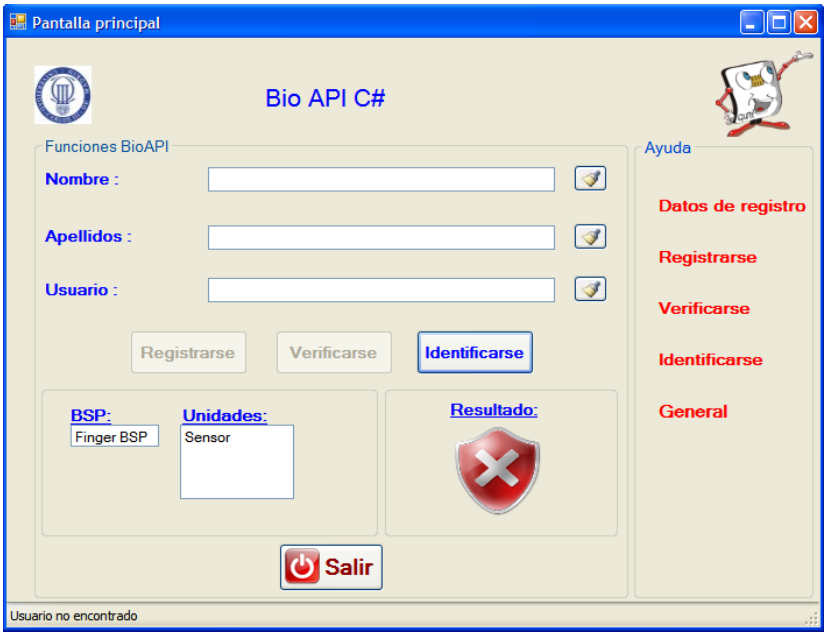

Figura 60: Usuario no encontrado tras identificación

Donde se puede ver que el proceso de autentificación ha sido incorrecto, y el mensaje de error devuelto es el de “Usuario no encontrado”.

6.4. Pruebas de operaciones bancarias

6.4.1. Consulta de saldo

Una vez el usuario ha accedido a la aplicación, ya sea tras el registro, o en futuras visitas mediante las operaciones de verificación e identificación, en caso de querer consultar su saldo lo único que deberá hacer será hacer click sobre el botón “Consultar” de manera que en el recuadro que se encuentra debajo de este aparecerá el saldo del que se dispone. En este caso el saldo que se espera obtener es de 0 €, ya que la cuenta acaba de ser creada, y aún no se han realizado operaciones monetarias en ella:

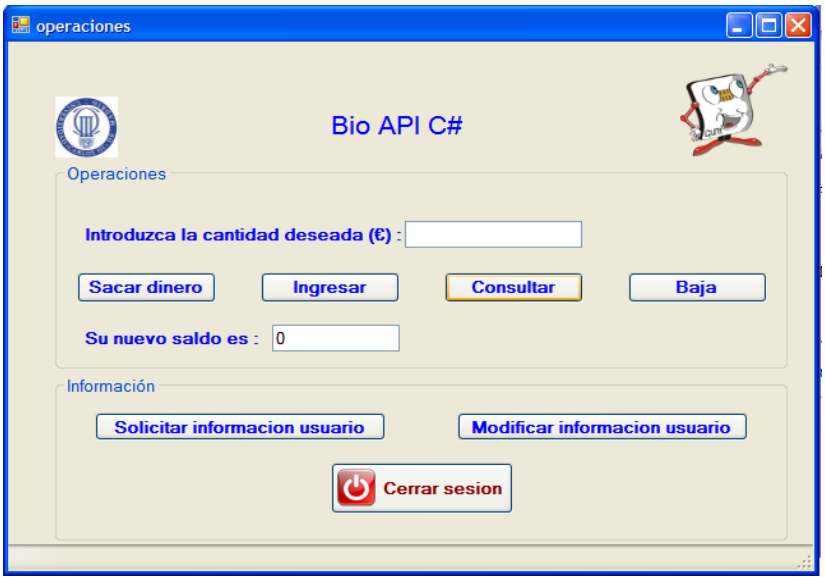

Figura 61: Operaciones (consulta de saldo)

Como se puede ver esta operación no admite confusión y el resultado obtenido es el esperado.

6.4.2. Ingreso de saldo

Para ingresar saldo, se introduce la cantidad deseada en el recuadro superior, y se pulsa el botón “Ingresar”. En esta prueba, se intentan ingresar 10€:

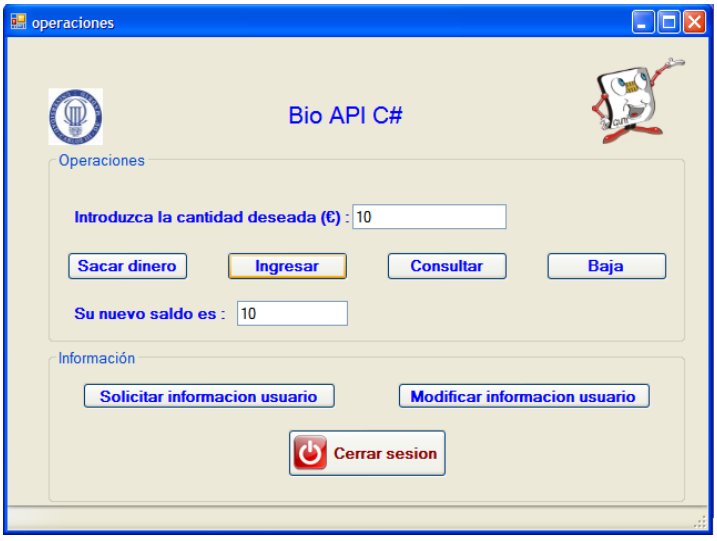

Figura 62: Ingreso de saldo

Como se puede ver el resultado obtenido es el correcto, ya que la aplicación actualiza automáticamente (tras un ingreso o una extracción) el saldo de la cuenta y lo muestra en pantalla.

6.4.3. Extracción de saldo

Para extraer saldo, se introduce la cantidad deseada en el recuadro superior, y se pulsa el botón "Sacar dinero". En esta prueba, se intentan sacar 9€:

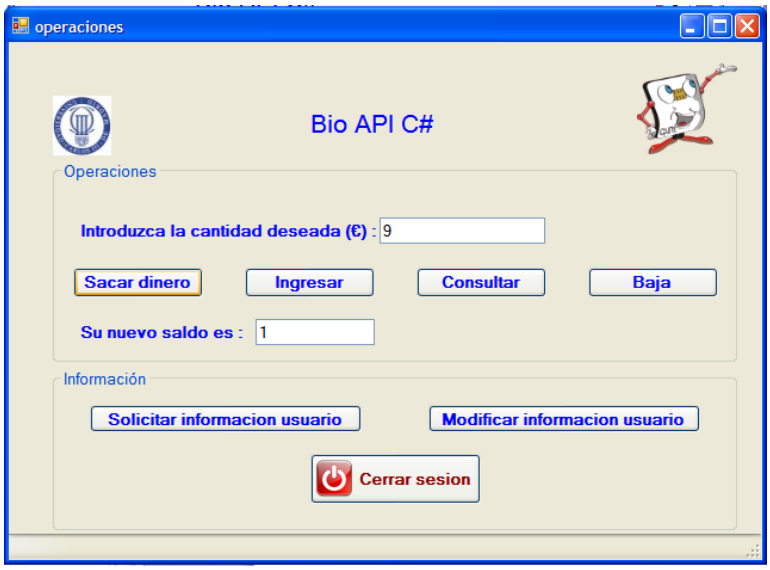

Figura 63: Extracción de saldo

Como se puede ver el resultado obtenido es correcto, ya que el nuevo saldo de la cuenta es de 1€.

Si en lugar de sacar 9€, se intentase extraer una cantidad mayor de la que dispone la cuenta (11€), el resultado que se obtendría sería el siguiente:

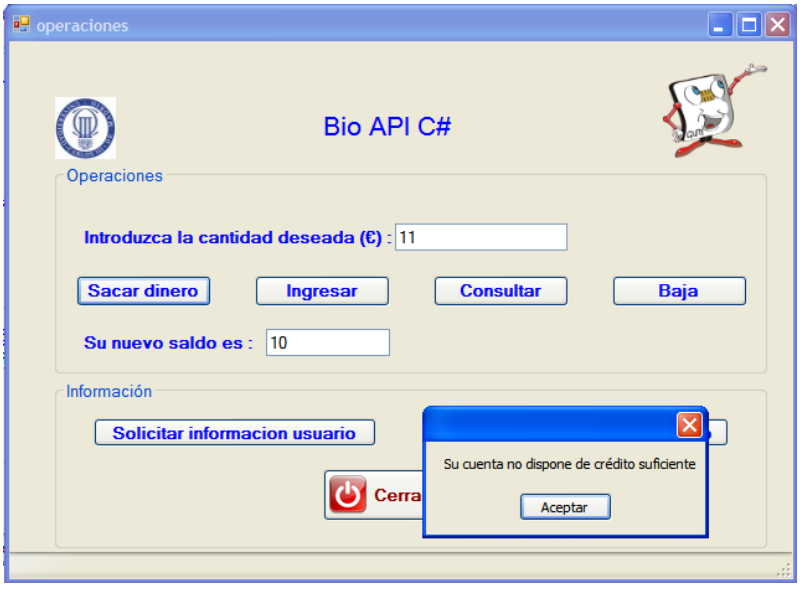

Figura 64: Extracción de saldo (crédito insuficiente)

Como se puede ver el resultado coincide con lo descrito en el punto 5 de la memoria (Desarrollo del proyecto), por lo que este es el esperado.

6.5. Pruebas tratamiento de la información de usuario

6.5.1. Mostrar información de la cuenta

En caso de que el usuario quisiese comprobar los datos actuales de su cuenta, lo único que debería hacer sería pulsar sobre el botón “Solicitar información de usuario”, obteniendo en el caso de esta prueba el resultado siguiente:

Figura 65: Información de la cuenta

Donde como se puede ver el nombre, apellidos y usuario son los que se introdujeron en la prueba de registro, y la huella la facilitada por el usuario. Por ello se concluye diciendo que el resultado obtenido es el correcto.

6.5.2. Modificación de información de usuario

En caso de que el usuario quisiese modificar los datos de su cuenta, podría hacerlo pinchando en el botón “Modificar información de usuario”, en cuyo caso accedería a la siguiente pantalla:

Figura 66: Modificar información de usuario

En caso de pulsase sobre los botones de "Borrar datos biométricos" o "Borrar datos biográficos", los resultados obtenidos sería los siguientes:

Figura 67: Modificar información de usuario (operación exitosa datos biométricos)

Figura 68: Modificar información de usuario (operación exitosa datos biográficos)

Donde se indica que dichos datos han sido borrados. Para comprobarlo se mostrará la nueva configuración del fichero almacenado en el servidor:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<usuario>
  <encounter>
    <apellidos>
    </apellidos>
    <value>1</value>
    <nombre>
    </nombre>
    <datos_biométricos>
    </datos_biométricos>
    <imagen_huella>
    </imagen_huella>
  </encounter>
  <identificadorUnico>daacb444-f901-4d18-b929-4491e53fc147</identificadorUnico>
  <cuenta_bancaria>10</cuenta_bancaria>
</usuario>
```

Figura 69: Fichero

Donde como se puede apreciar tanto el nombre, apellidos o datos biométricos han sido borrados de la cuenta, por lo que el resultado obtenido es una vez más el esperado.

Si el usuario quisiese introducir unos nuevos datos biográficos, (ya sea porque los ha borrado, o simplemente porque quiere actualizarlos) deberá rellenar las casillas de nombre y apellidos y pulsar el botón actualizar datos biográficos:

Figura 70: Nuevos datos biográficos

Obteniéndose el siguiente resultado:

Figura 71: Nuevos datos biográficos (operación exitosa)

En caso de que se quisiesen modificar los datos biométricos se pulsaría el botón Actualizar datos biométricos, y tras realizar de nuevo dos capturas (para crear una plantilla de referencia sin errores) el resultado obtenido sería el siguiente:

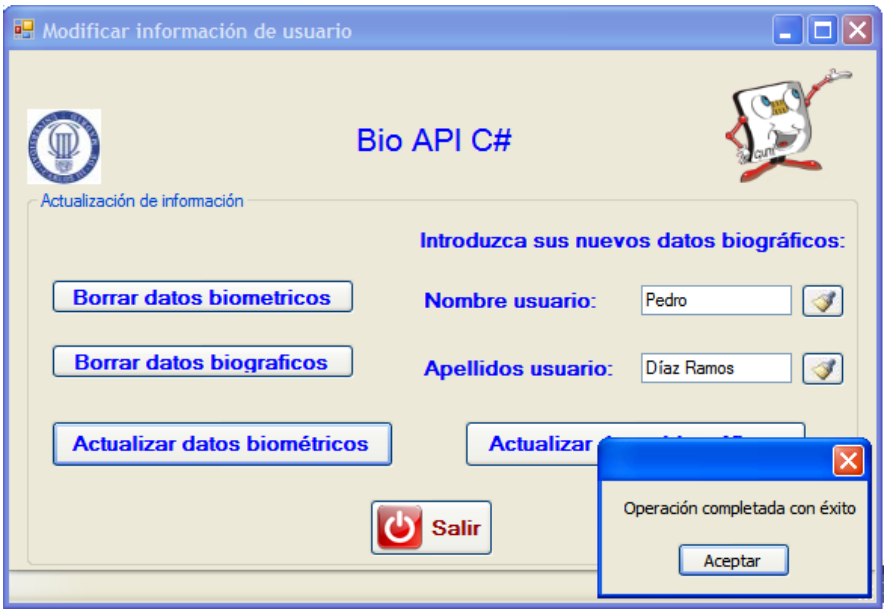

Figura 72: Nuevos datos biométricos (operación exitosa)

Por lo que si de nuevo se pidiese al sistema que mostrase la información de la cuenta, el resultado obtenido en este caso sería el siguiente:

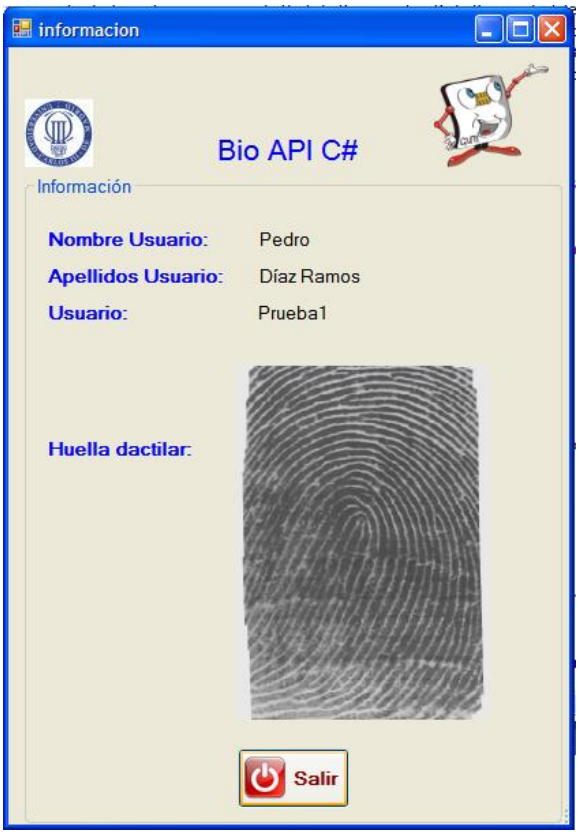

Figura 73: Información de la cuenta (operación exitosa)

Donde se puede apreciar que tanto el nombre y apellidos como la huella son diferentes de los del registro inicial, por lo que el resultado obtenido sería correcto.

6.6.Eliminación de la cuenta

Para concluir el apartado de pruebas se intentará borrar la cuenta creada anteriormente. Para ello el usuario debería pulsar el botón "Baja" de la pantalla "operaciones". El resultado obtenido sería el siguiente:

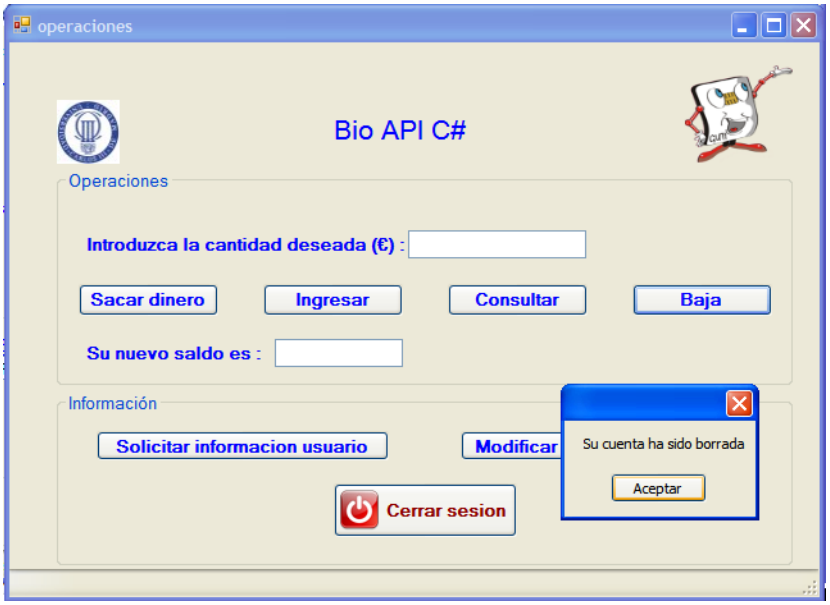

Figura 74: Eliminación de la cuenta (operación exitosa)

Una vez que el usuario pulse el botón aceptar se le redireccionará a la pantalla inicial, ya que al no poseer cuenta en el sistema, no puede navegar por este.

En este caso si se intentase acceder a la carpeta del servidor donde se almacenaba el fichero de dicho usuario, el resultado sería el siguiente:

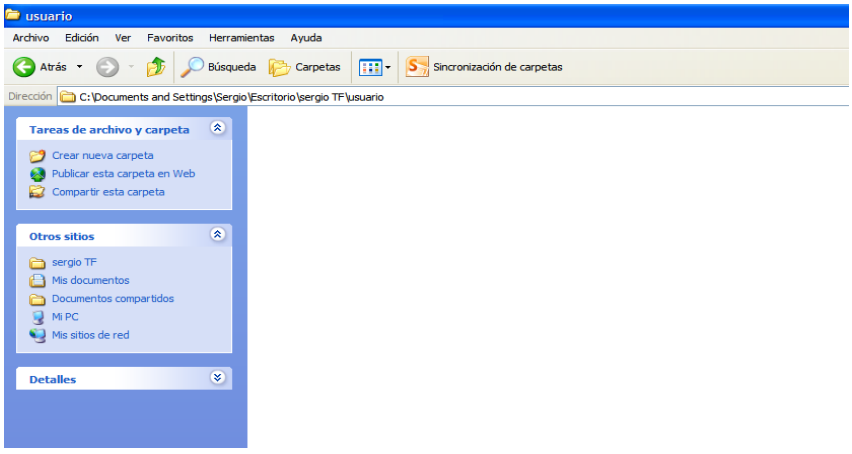

Figura 75: Eliminación de la cuenta (fichero)

Donde como se puede ver no aparece ningún fichero sobre dicho usuario, por lo que el resultado obtenido será calificado como correcto.

7. Conclusiones y líneas futuras

7.1.Conclusiones

En este trabajo se ha creado un prototipo de un sistema de registro y control de acceso de individuos a una cuenta bancaria en lenguaje C#. Este prototipo permite controlar y organizar los datos tanto bancarios como personales ya sean biográficos o biométricos. Se ha buscado una aplicación sencilla pero que a la vez cumpla todas las funciones especificadas.

Los principales problemas que han surgido a lo largo de la realización del trabajo vienen derivados de la falta de experiencia con el lenguaje de desarrollo (C#) y la escasez de conocimientos sobre las técnicas de reconocimiento biométrico. Otro de los puntos de mayor dificultad fue el servidor, ya que con anterioridad a este trabajo el alumno no había tenido contacto alguno ni teórico ni práctico con enlaces cliente-servidor. Fueron de gran ayuda la gran variedad de tutoriales existentes en la red.

No obstante gracias a la similitud del lenguaje C# con otros estudiados en Grado, tal como Java, y a la gran facilidad con la que el entorno de desarrollo usado crea interfaces gráficas fácilmente manejables, se ha conseguido crear una aplicación muy intuitiva y de fácil seguimiento, capaz de cumplir con nota los requisitos iniciales marcados al comienzo del proyecto.

El conjunto de esta aplicación cliente servidor difiere del tipo de aplicaciones que se pueden encontrar en el mercado, ya que actualmente no existe ningún banco que nos permita acceder a nuestra cuenta a través de procedimientos biométricos. Lo que actualmente se utilizan son accesos a través de claves numéricas.

Para resumir se puede concluir que el autor partía de unos conocimientos muy básicos tanto sobre biometría como sobre enlaces cliente-servidor y que tras la realización de este proyecto conoce y dispone de las herramientas necesarias para poner en práctica cualquier idea que surja, cumpliéndose el objetivo principal de este trabajo, que no es otro que el de aprender.

7.2.Líneas futuras

Un trabajo como el realizado en este documento es una fuente de posibles líneas de trabajo futuro, tanto en la mejora de la aplicación, como en el servidor, incluyendo mejoras de rendimiento conjunto.

Respecto al servidor, algunas líneas futuras son:

- El extender el nivel de conformidad del estándar que se ha implementado en la aplicación (nivel 4), ya que la subida de un nivel incorpora notables mejoras a nivel global en la aplicación, por lo que la mejora de la aplicación hasta otorgarla un nivel 1, constituiría el desarrollo de una aplicación muy completa, capaz de estar operativa en cualquier lugar del mundo.

- Utilizar un sistema de bases de datos para almacenar la información de los clientes, en lugar de un sistema basado en ficheros. Además se podría aumentar la información requerida de los clientes, de manera que estuvieran más identificados como por ejemplo su edad o su dirección.

Respecto al cliente, posibles líneas futuras son:

- El cliente desarrollado en este proyecto, centra sus esfuerzos en la captura de datos biométricos basados en la huella dactilar. No obstante, un futuro desarrollo de la aplicación, podría consistir en otorgar al cliente capacidad para capturar y enviar al servidor datos biométricos basados en otras técnicas de reconocimiento como las comentadas en el apartado sobre biometría (escáner de retina, de iris, etc).

Conclusions and future works

Conclusions

In this work a prototype of a system was created to register and monitor individual access to a bank account in C#. This prototype allows controlling and organizing both banking and personal data such as biographical and biometrics. It has fulfilled a simple application but at the same time an application that complies with all specified function.

The main problems that have arisen over the performance of work come from the lack of experience with the development language (C#) and the lack of knowledge about biometric techniques. Another point of greatest difficulty was the development of a server, because prior to this work the student had not prior theoretical or practical contact with such a development. The wide variety of tutorials available on the network.

However due to the similarity of the C# language with others studied during the Bachelor such as Java, and the ease with which the development environment used creates graphical interfaces easily manageable, it has been possible to create a very intuitive and easy to follow application, able to meet the initial requirements set at the start of the project.

This entire client/server application differs from the type of applications that can be found on the market, since currently there is no bank that allows to access to the account through biometric procedures, rather than through numeric keys.

To summarize it can be concluded that the author started from very basic knowledge on biometrics and client-server applications, and that after the completion of this Bachelor Thesis, he knows and has the tools necessary to implement any new ideas that emerge, fulfilling the main objective of this work, which is no other than to learn.

Future works

The work done in this Bachelor Thesis is a source of potential future work, improving the application and server, including overall performance improvements.

On the server, some future lines are:

- To improve the conformance level of the BIAS standard addressed in this work (level 4). A higher level incorporates globally significant improvements in the application, so that the improvement of the application to the Level 1, would be a comprehensive development, able to be operational in any required scenario and application.
- Using a database system to store customer information, rather than a file system. It also may increase the required information from customers, so that they were identified with more of their personal data, such as age or address.

On the client, possible future lines are:

- The client developed in this project focuses its efforts in capturing biometrics fingerprint based. However, further development of the application, would be to give the customer ability to capture and send to server biometric data base in other techniques as discussed in the section on biometrics (voice, iris, etc.).

Bibliografía

[1] Historia de la biometría. **Wikipedia** - http://es.wikipedia.org/wiki/Biometr%C3%ADa – Fecha de última visita: 15/4/2013.

[2] Tipos de sistemas biométricos. **UNAM (Facultad de Ingeniería Biométrica informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/clasificaciontipo.html - Fecha de última visita: 3/5/2013.

[3] Reconocimiento de geometría de la mano. **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recomano.html - Fecha de última visita: 7/5/2013.

[4]Captura geometría mano. **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recomano.html - Fecha de última visita: 8/5/2013.

[5]Extracción características geometría mano. **Tapiador Mateos, M y Siguenza Pizarro, J.A**. Tecnologías Biométricas aplicadas a la seguridad. Editorial: MIC. Año 2005.

[6] Reconocimiento de iris. **Wikipedia** - http://es.wikipedia.org/wiki/Reconocimiento_de_iris - Fecha de última visita: 15/5/2013.

[7]Escaneo de retina. **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/escaneoretina.html - Fecha de última visita: 15/5/2013.

[8] **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recohuella.html - Fecha de última visita: 20/5/2013.

[9] Reconocimiento facial. **Wikipedia** http://es.wikipedia.org/wiki/Sistema_de_reconocimiento_facial - Fecha de última visita: 22/5/2013.

[10] Reconocimiento de escritura y firma. **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recofirma.html - Fecha de última visita 23/5/2013.

[11] Reconocimiento de voz. **UNAM (Facultad de Ingeniería Biometría informática) -** http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recovoz.html - Fecha de última visita: 23/5/2013

[12] Características de sistemas biométricos. **Wikipedia** - http://es.wikipedia.org/wiki/Biometr%C3%ADa – Fecha de última visita 23/5/2013.

[13] Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0 – Draft 2012.

[14] BIAS según OASIS. Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0 – Draft 2012.

[15] ISO/IEC JTC 1/SC 37 N 4883-Draft Febrero 2012

[16] Tutorial de C#. **msdn** - http://msdn.microsoft.com/es-es/library/aa288436(v=vs.71).aspx – Fecha de última visita: 12/9/2012.

[17] Wikipedia. Visual Studio. **Wikipedia** - http://es.wikipedia.org/wiki/Microsoft_Visual_Studio - Fecha de última visita: 1/6/2013.

[18] Visual Studio 1.0. **Microsoft -** http://www.microsoft.com/visualstudio/esn/whats-new - Fecha de última visita: 1/6/2013.

[19] Tutorial de C#. **Programación Fácil** - http://www.programacionfacil.com/csharp_net/start - Fecha de última visita: 15/9/2012.

[20] Tutorial de C#. **C# Ya** http://www.csharpya.com.ar/index.php?inicio=60 – Fecha de última visita 20/9/2012.

Anexo 1. Presupuesto y planificación del trabajo

A continuación se va a llevar a cabo un desglose de las tareas que se han realizado a lo largo de este trabajo fin de grado, lo que facilitará posteriormente un cálculo aproximado sobre su coste.

Debido a la complejidad de un trabajo de estas características se ha optado por dividirlo en distintas fases, las cuales se van a comentar a continuación:

Fase 1: Documentación inicial

I. Estudio del lenguaje C# y del entorno de desarrollo (20 horas)
II. Preparación de las herramientas de trabajo (4 horas)
III. Búsqueda y realización de tutoriales y aplicaciones sencillas. (30 horas)
IV. Asistencia a charlas y presentaciones sobre C# (16 horas)

Fase 2: Desarrollo de la aplicación

I. Actividad principal (60 horas)
II. Pantalla de configuración (20 horas)
III. Actividades secundarias (10 horas)
IV. Interconexión de todas las actividades (30 horas)

Fase 3: Pruebas en un dispositivo real

I. Pruebas de campo con un sensor de huella (10 horas)
II. Corrección y depuración (20 horas)

Fase 4: Elaboración de la memoria

I. Redacción de la memoria (65 horas)
II. Corrección y maquetación (15 horas)

Tabla 3: Fases del proyecto

FASES	HORAS EMPLEADAS
Documentación inicial	70
Desarrollo de la aplicación	120
Pruebas en un dispositivo real	30
Elaboración de la memoria	80
TOTAL	**300**

PRESUPUESTO DEL TRABAJO FIN DE GRADO

COSTES MATERIALES

Los materiales necesarios han sido un ordenador, se recomienda de altas prestaciones para un correcto funcionamiento del emulador, y un dispositivo capaz de realizar capturas de huellas dactilares. Considerando un periodo de amortización de cada uno de los dos dispositivos de 3 años y teniendo en cuenta el tiempo del proyecto, los costes materiales quedan como se expone en la Tabla 2.

Tabla 4: Costes materiales

CONCEPTO	PRECIO (€)
Ordenador de altas prestaciones	100
Dispositivo de captura de huella	6.66
TOTAL	**106.66**

COSTES DE PERSONAL

Para la realización de este trabajo, ha sido necesaria la presencia de un jefe de proyecto y un ingeniero.

Tabla 5: Costes de personal

OCUPACIÓN	HORAS	PRECIO/HORA	IMPORTE (€)
Jefe de proyecto	25	90	2250
Ingeniero	275	60	16500
TOTAL	**300**		**18750**

COSTES TOTALES

Tabla 6: Costes totales

CONCEPTO	PRECIO (€)
Costes de materiales	106.66
Costes de personal	18750
Costes indirectos (20%)	3771.33
Subtotal	22627.99
IVA (21%)	4751.88
TOTAL	**27379.87**

El coste total del proyecto es de *veintisiete mil trescientos setenta y nueve euros con ochenta y siete céntimos.*

Leganés, 24 de Junio de 2013

El ingeniero