Universidad Carlos III de Madrid
Escuela Politécnica Superior
Departamento de Tecnología Electrónica
Ingeniería Superior de Telecomunicación

Proyecto Fin de Carrera

SISTEMA INTEGRADO DE CONTROL DE ACCESOS MEDIANTE TARJETA Y BIOMETRÍA

Autor: Francisco José Díez Jimeno
Tutor: Raúl Sánchez Reíllo
Septiembre 2009

A mis padres, por haberme dado todo

lo que ellos nunca pudieron tener

ÍNDICE

LISTADO DE ACRÓNIMOS

- **PIN**: Número de Identificación Personal (*Personal Identification Number*)
- **FAR**: Tasa de falsa aceptación (*False Accept Rate*)
- **FMR**: Tasa de falso reconocimiento (*False Match Rate*)
- **FRR**: Tasa de falso rechazo (*False Reject Rate*)
- **FNMR**: Tasa de falso no reconocimiento (*False Non-Match Rate*)
- **FTA**: Fallo al adquirir (*Failure To Acquire*)
- **FTR** o **FER**: Tasa de fallo al inscribir (*Failure To Enroll Rate*)
- **FIR**: Tasa de falsa identificación (*False Identification Rate*)
- **CPU**: Unidad central de procesamiento (*Central Processing Unit*)
- **RISC**: Computadora con Conjunto de Instrucciones Reducidas (*Reduced Instruction Set Code*)
- **RAM**: Memoria de acceso aleatorio (*Random-access memory*)
- **ROM**: Memoria de solo lectura (*Read-only memory*)
- **EEPROM**: Memoria de solo lectura re-escribible eléctricamente (*Electrically Erasable Programmable Read-Only Memory*)
- **SOTI**: Sistema operativo de la tarjeta inteligente (*SOTI*)
- **MF**: Fichero maestro (*Master File*)
- **DF**: Fichero dedicado (*Dedicated File*)
- **EF**: Fichero elemental (*Elementary file*)
- **RFID**: Identificación por radiofrecuencia (*Radio-frequency Identification*)
- **SDK**: Kit de desarrollo de software (*Software Development Kit*)
- **LCD**: pantalla de cristal líquido (*Liquid Crystal Display*)
- **TCP/IP**: Protocolo de Control de Transmisión / Protocolo de Internet (*Transmission Control Protocol / Internet Protocol*)
- **UDP**: Protocolo de datagrama de usuario (*User Datagram Protocol*)
- **CRT**: Bibliotecas de ejecución de C (*C Run-Time Libraries*)

1. INTRODUCCIÓN

1.1. Biometría

1.1.1. Introducción a la Biometría

Etimológicamente la palabra biometría procede del griego '*bios*' (vida) y '*metron*' (medida) y hace referencia a la medida de las características biológicas de un individuo. Según el diccionario de la lengua española el término biometría es el "estudio mensurativo o estadístico de los fenómenos o procesos biológicos", es decir, la aplicación de técnicas matemáticas y estadísticas a las ciencias biológicas.

Una definición más precisa del significado que realmente tiene este término puede darse dentro del campo de la identificación de personas, donde la biometría se define como la ciencia que estudia las características anatómicas o de comportamiento de una persona, con el objeto de que pueda ser reconocida.

Es de sobra conocido por todos que el ser humano posee características particulares (rasgos biológicos), que le diferencian enormemente del resto de individuos de su especie, haciendo único e irrepetible a cada individuo. Características biofísicas como pueden ser el rostro, la huella de sus dedos, los ojos o las manos, y características de comportamiento como pueden ser su manera de caminar, la forma en que habla, o su manera de escribir.

No fue sin embargo hasta finales del siglo XIX cuando realmente se desarrolla la biometría como ciencia que utiliza parámetros biológicos para la identificación de un individuo. Es entonces cuando aparece el sistema de identificación antropométrica de Bertillón, mediante el cual una persona era identificada según una serie de medidas de su cuerpo (como el diámetro de su cabeza, la longitud de sus extremidades o el tamaño del tronco). En esta época también comienza a estudiarse la identificación de personas a través de sus huellas dactilares. Se llevan a cabo multitud de estudios sobre la huella dactilar, estudiándose cuáles son los diferentes elementos característicos de una huella, su estabilidad con el tiempo en una persona, es decir, si varía o no con los años, y también profundos estudios sobre la unicidad de la huella. Todos estos estudios provocaron una gran aceptación de la huella dactilar como método de identificación biométrica en el ámbito policial. Y con su más de un siglo de existencia se ha convertido en la modalidad biométrica más desarrollada.

1.1.2. Identificación biométrica

Identificar es el acto de reconocer si una persona es aquella misma que se supone o se busca, o también facilitar determinados datos personales necesarios para ser reconocido.

La identidad de una persona puede determinarse de muy diferentes maneras. Una de ellas es solicitándole una determinada información que solamente él y la entidad donde quiere identificarse conocen, como un número de identificación personal (PIN) o una contraseña. El inconveniente de esta técnica es doble, por un lado el usuario puede perder u olvidar el patrón

que le identifica, sobre todo si los patrones a recordar difieren según donde el usuario quiera autenticarse y tiene que recordar un número elevado de ellos. El otro inconveniente es el robo, cualquiera que consiga hacerse con la contraseña o número de identificación de otro usuario podrá suplantar su identidad. No obstante cuando un patrón de identificación es comprometido éste es fácilmente reemplazable por otro nuevo.

Otra manera de identificar a un usuario es mediante la posesión de un elemento que le identifique, como puede ser una llave, una tarjeta o un certificado digital. Este tipo de técnicas basadas en la posesión a menudo se complementan con otras técnicas de identificación para hacerlas más seguras. Por ejemplo, un usuario de un cajero electrónico debe autenticarse antes de efectuar cualquier operación económica. El usuario dispone de una tarjeta de banda magnética que contiene información acerca de su identidad, y además debe introducir un número secreto que solamente él y la entidad bancaria conocen para probar su identidad. Los inconvenientes de estas técnicas son los mismos que en la técnica anterior, ya que un usuario malintencionado podría hacerse con el número secreto y la tarjeta de identificación de otra persona.

La utilización de claves secretas y/o tarjetas de identificación no es suficiente en entornos que requieran un mayor nivel de seguridad, y en los cuales es imprescindible verificar sin lugar a dudas la identidad que un usuario dice ser. Es en este tipo de aplicaciones donde entran en juego las modalidades de identificación biométricas.

La identificación biométrica se basa en el reconocimiento de características biológicas o de comportamiento, como pueden ser la huella dactilar, el iris, la voz o la firma.

Las modalidades de identificación biométrica, frente a otras formas de autenticación personal, presentan la ventaja de que los patrones son distintivos del individuo, no pueden ser olvidados, ya que siempre los llevamos con nosotros, no pueden ser sustraídos y no son fácilmente reproducibles. Debido a que la persona a ser identificada necesita estar físicamente frente al terminal de identificación la biometría es una técnica más confiable.

Sin embargo no todo son ventajas, para evitar el fraude en los sistemas de autenticación biométricos es necesaria la realización de pruebas colaterales, (como la detección de elemento vivo), de manera que se tenga constancia de que lo que realmente se encuentra delante del punto de identificación es una persona viva y no una imagen o una copia. Otro inconveniente importante es el siguiente; si una información biométrica es comprometida no es posible reemplazarla. Por ejemplo en el caso de identificación por número secreto o por contraseña, si la clave de la cuenta de un usuario es comprometida puede sustituirse fácilmente por otra nueva, estando el resto de cuentas con distinta clave a salvo. Si una información biométrica es comprometida el atacante podría tener acceso a parte o a todo el sistema, dependiendo del nivel de privilegio asociado a la biometría. Otros inconvenientes de la identificación biométrica son que la identificación se da en términos de probabilidad y también que determinados equipamientos presentan un coste de adquisición y mantenimiento elevados.

1.1.3. Fases de un sistema de identificación biométrica

Un sistema de identificación biométrica se basa en la medición, ya sea directa o indirecta, de las características biológicas o de comportamiento de un usuario para identificarle de forma automática. Para ello se pueden utilizar técnicas estadísticas como reconocimiento de patrones de señal o comparación, y también técnicas de inteligencia artificial como lógica borrosa o redes neuronales. Independientemente de la técnica biométrica utilizada se sigue un esquema que consta de dos fases o etapas diferentes: reclutamiento y utilización.

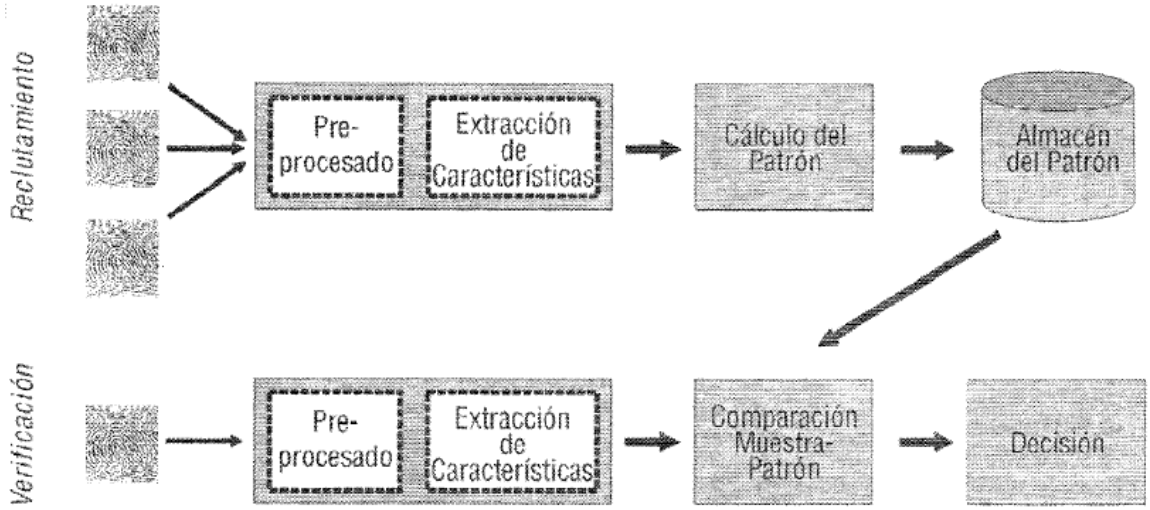

Fig. 1.1: Etapas en un sistema de identificación biométrica (Fuente:[1]).

- **Reclutamiento**: en esta primera fase los datos biométricos del usuario son capturados y procesados para posteriormente extraer su información característica. Esta información, conocida como patrón biométrico, caracteriza al usuario, y puede ser almacenada bien en una base de datos o bien en un dispositivo portátil como una tarjeta inteligente.
- **Utilización**: una vez obtenido y almacenado el patrón biométrico del usuario éste está preparado para utilizar el sistema. Para ello cada vez que el usuario desee autenticarse debe presentar su rasgo biométrico al sistema, que calculará el patrón biométrico de la muestra obtenida del usuario, comparándolo con el patrón almacenado del usuario. Como resultado de la comparación se obtendrá una determinada probabilidad de éxito o fracaso en la identificación.

Como puede observarse en la figura anterior, las fases de reclutamiento y de utilización comparten una serie de pasos que son comunes:

- **Captura**: en esta primera fase se obtienen los datos biométricos o de comportamiento del usuario, dependiendo este proceso de la modalidad biométrica empleada. Durante la fase de reclutamiento el proceso de captura se realiza de manera supervisada, es decir, una persona comprueba cómo se toman los datos, cerciorándose además de la identidad de la persona.
- **Pre-procesado y extracción de características**: la fase de pre-procesado modifica los datos capturados de manera que resulte más fácil obtener las características más

significativas de los mismos. El proceso de extracción de características es la parte más importante del sistema de identificación biométrica, ya que determina la capacidad del sistema para distinguir entre dos sujetos distintos con características biométricas parecidas. En la fase de reclutamiento una vez obtenidas las características de la muestra capturada se calcula el patrón identificativo del usuario y se almacena. Este patrón caracterizará al usuario.

- **Comparación y decisión**: en la fase de utilización, una vez obtenidas las características de la muestra en cuestión, se comparan con el conjunto de patrones almacenados en el sistema y se toma una decisión. Es importante destacar que el proceso de comparación no se trata de una comparación de igualdad entre muestras, ya que estas mismas pueden verse modificadas según se realice el proceso de captura, e incluso las características biológicas del sujeto pueden variar levemente.

Por este motivo el proceso de comparación consiste básicamente en una medida de semejanza entre el patrón de características de la muestra tomada y los patrones almacenados en el sistema. Esta comparación puede realizarse de muy diversas maneras: mediante métricas como la distancia Euclídea o la distancia de Hamming, técnicas estadísticas como el empleo de funciones de distribución, o técnicas basadas en modelado de problemas, como redes neuronales.

Como resultado de la comparación se obtiene una probabilidad de semejanza entre la muestra presentada y una de las muestras anteriormente registradas en el sistema. Según un umbral previamente establecido, que establece el nivel de seguridad del sistema, el éxito o fracaso de la comparación vendrá determinado.

Existen una serie de parámetros que reflejan la calidad del sistema de identificación biométrica:

- **FAR** (False Accept Rate): la tasa de falsa aceptación indica el porcentaje de número de veces que el sistema produce una falsa aceptación. Es decir cuando un individuo es identificado como usuario de manera incorrecta.
- **FMR** (False Match Rate): es la probabilidad de que las personas no autorizadas sean incorrectamente reconocidas durante el proceso de comparación de características. A diferencia de la FAR la FMR no contabiliza los intentos previamente rechazados.
- **FRR** (False Reject Rate): la tasa de falso rechazo indica la probabilidad de que un usuario autorizado sea rechazado por el sistema.
- **FNMR** (False Non-Match Rate): es la tasa a la que las personas autorizadas sean falsamente no reconocidas durante el proceso de comparación de características.
- **FTA** (Failure To Acquire): es el número de intentos de verificación o identificación sin éxito por error de personas inscritas correctamente. Puede producirse cuando la imagen tomada por el sensor no sea de calidad suficiente.

- **FTE** o **FER** (Failure To Enroll Rate): es la proporción de personas que no consiguen realizar con éxito el procedimiento de inscripción.
- **FIR** (False Identification Rate): es la probabilidad de que una persona autorizada sea identificada pero se le asigne un identificador falso.

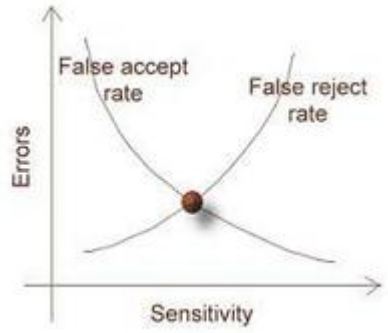

Fig. 1.2: Selección del umbral en un sistema de identificación biométrica (Fuente:[7]).

La elección de un umbral muy alto provoca que el sistema sea muy restrictivo, la tasa de aceptación de usuarios no autorizados se reduce, pero sin embargo la probabilidad de rechazar a usuarios autorizados aumenta. Sin los usuarios son rechazados erróneamente con frecuencia parecerá que el sistema no funciona correctamente, incrementando el grado de insatisfacción de los usuarios. Un umbral más bajo provocará el efecto contrario, el sistema se relajará, permitiendo un mayor porcentaje de accesos de usuarios no autorizados. Por tanto la decisión de un umbral adecuado es imprescindible según el nivel de seguridad y el grado de confianza y amigabilidad hacia el usuario que se le quiera dar al sistema.

1.1.4. Esquemas de funcionamiento en un sistema de identificación biométrica

La metodología de funcionamiento de un sistema de identificación biométrica puede presentar dos esquemas de operación bien diferenciados.

El primero de ellos recibe el nombre de **reconocimiento o identificación** y se trata de un proceso de comparación de 1 a N. Consiste en identificar a un usuario dentro de todos los registrados previamente en el sistema, para ello se extraen las características biométricas del usuario a identificar y se comparan con las características existentes en la base de datos del sistema. Como resultado de la comparación puede identificarse al usuario con aquel con el que se haya obtenido una mayor probabilidad de parecido, es decir, siempre se identifica al usuario con uno de los existentes en la base de datos. O sin embargo puede existir un determinado umbral en el sistema, de manera que solo se identifique al usuario con otro siempre y cuando se supere una probabilidad determinada de semejanza. Este esquema de funcionamiento presenta algunos inconvenientes, el primero de ellos es la necesidad de una base de datos que almacene los patrones característicos de todos los usuarios reclutados en el sistema, con los consecuentes requisitos de capacidad, seguridad y conectividad a la base de datos, ya que es imprescindible que siempre sea posible acceder a la misma y se garantice la privacidad de los datos intercambiados. Otro inconveniente es el tiempo necesario para efectuar la

identificación, que puede ser elevado si el número de usuarios en el sistema es muy grande, ya que el número de comparaciones que hay que realizar es alto.

El otro esquema de funcionamiento recibe el nombre de **autenticación o verificación** y se trata de un proceso 1 a 1. En este esquema de funcionamiento el usuario debe dar a conocer su identidad en el sistema biométrico. El sistema, a partir de las características biométricas del usuario calcula su patrón característico y lo compara con el patrón facilitado. Si como resultado de la comparación se obtiene una probabilidad de similitud que supera un determinado umbral el sistema autentica al usuario, ya que da por hecho que el usuario es aquel quien dice ser. El patrón característico del usuario puede estar almacenado en una base de datos tal y como sucede en el esquema de identificación, o por el contrario estar contenido en un dispositivo portátil como una tarjeta inteligente. En este caso se suprime la necesidad de una base de datos que almacene los patrones característicos, y el usuario a identificar debe presentar dicho elemento identificativo en el momento en el que quiera probar tal identidad mediante sus características biológicas o de comportamiento. Otra ventaja que presenta este esquema es la velocidad, en general el proceso de verificación es mucho más rápido que el de identificación, ya que solamente es necesario efectuar una comparación de patrones biométricos.

1.1.5. Modalidades de identificación biométricas

Las modalidades de identificación biométricas se fundamentan en el análisis de una característica fisiológica o de comportamiento. Aunque en principio cualquier parte del cuerpo humano, o característica de comportamiento de una persona, serían susceptibles de ser usadas para la identificación, se atiende a una serie de criterios prácticos. Lo ideal es que la característica utilizada para la identificación se demuestre única y propia de la persona a identificar. Para ello se selecciona una característica robusta, no sujeta a grandes cambios, que sea lo más distintiva posible respecto al resto de la población, que sea una característica disponible en la mayor cantidad de individuos posibles y que cuente con la aceptación del usuario, ya que algunas veces éste puede percibir al sistema biométrico como excesivamente intrusivo.

En general si el proceso identificativo se hace atendiendo a la anatomía del usuario se habla de Biometría Estática, y si se realiza según la forma en que el sujeto se comporta se llama Biometría Dinámica. Habitualmente los dispositivos que miden el comportamiento requieren de la cooperación del usuario, por ejemplo, que el usuario diga su nombre o una determinada frase frente al sistema de reconocimiento. Los que miden alguna característica fisiológica también pueden requerir de la colaboración del usuario, aunque existen casos en los cuales el proceso de identificación puede pasar totalmente inadvertido para el usuario, que no tiene que llevar a cabo ninguna acción. Por ejemplo, en la identificación del rostro una videocámara capta una imagen del usuario que se aproxima al sistema, y la procesa automáticamente para su identificación.

Existen gran cantidad de modalidades de identificación biométrica, que para llevar a cabo el proceso de identificación registran un aspecto exclusivo del individuo. Las más utilizadas habitualmente son las siguientes:

- Huella dactilar

La identificación de individuos mediante huellas dactilares ha sido siempre reconocida como una de las mejores modalidades de identificación biométrica, ya que se trata de una modalidad profundamente estudiada y cuenta a sus espaldas con más de un siglo de existencia. Multitud de estudios científicos avalan la unicidad de la huella de un individuo, ya que ha sido demostrado que no existen dos dedos con huellas idénticas, ni siquiera entre dedos de una misma persona, ni tampoco entre gemelos. La estabilidad de la huella de una persona con el transcurso del tiempo también ha sido verificada.

Los sistemas biométricos de huella dactilar suelen estar basados en un sensor óptico que captura la huella digital del usuario. Una vez tomada la imagen se lleva a cabo el proceso de extracción de características que puede realizarse de diferentes maneras. Una forma consiste en estudiar la correlación entre la imagen tomada y otra previamente almacenada. Habitualmente el proceso de extracción de características se realiza analizando los elementos de la huella, bien analizando los poros del dedo, o bien obteniendo las minucias de la huellas. Una huella dactilar está formada por una sucesión de crestas, cuyo flujo sufre una serie de puntos singulares denominados **minucias**. Aunque la clasificación de los diferentes tipos de minucias puede ser extensa las minucias más frecuentes son las siguientes:

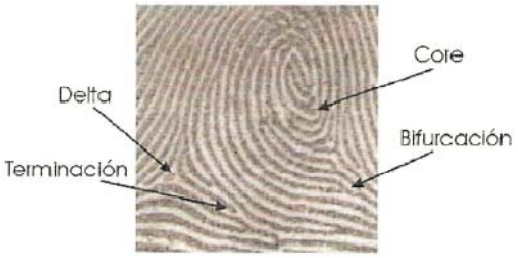

Fig. 1.3: Elementos característicos de una huella dactilar (Fuente: [8]).

- **Núcleo**: es el punto de la huella donde la orientación de las crestas tiende a converger.
- **Terminación**: punto de fin de una cresta.
- **Bifurcación**: punto donde una cresta se divide en dos.
- **Delta**: punto donde el flujo de las crestas presenta una divergencia.

Como resultado del análisis de los puntos más significativos de la huella se obtiene un patrón dactilar denominado vector de características, que caracteriza unívocamente al usuario. Este patrón garantiza también que la huella original del que procede no puede ser reconstruida a partir de él. A pesar de ser una modalidad muy desarrollada y de medio coste, la identificación por huella presenta una serie de inconvenientes. El primero de ellos es la connotación policial que siempre ha venido asociada a esta tecnología. Otro inconveniente deriva del propio proceso de captura de la huella, ya que la presión y la posición del dedo

sobre el lector pueden variar la imagen capturada y producir falso rechazo. Además para evitar posibles riesgos de suplantación de identidad es necesaria la realización de pruebas complementarias durante el proceso de captura, como la detección de elemento vivo.

- Geometría de la mano

Este tipo de modalidad puede centrarse en el estudio de determinados parámetros morfológicos de la mano o incluso de un determinado dedo del usuario, como pueden ser la anchura, la altura o las dimensiones. Otras modalidades más avanzadas se basan en la utilización de sensores de infrarrojos capaces de detectar el patrón venoso de la mano.

Estos sistemas habitualmente son más rápidos que otros sistemas biométricos, ya que son bastante sencillos y requieren una carga computacional pequeña. Tienen la capacidad de aprender, ya que a medida que el usuario se identifica a lo largo del tiempo el sistema detecta la evolución sufrida por la geometría de la mano. El principal atractivo de estos sistemas recae en la elevada aceptación y facilidad de uso por parte de los usuarios, a diferencia de otras modalidades como las de huella, la geometría de la mano no tiene connotación policial. Sin embargo, la unicidad y estabilidad de la mano no han sido probadas en grandes poblaciones. Las tasas de falsa aceptación y falso rechazo son peores que en otras tecnologías, lo que ha provocado que este tipo de modalidades no se utilicen en entornos de alta seguridad, y la detección de mano viva depende de la realización de pruebas colaterales en la detección.

- Iris, retina

Los sistemas de identificación mediante patrones oculares pueden estar basados en dos esquemas diferentes, topografía del iris y topografía de la retina. Ambos esquemas aprovechan las ventajas que las características oculares en las que se basan ofrecen. La textura del iris presenta un elevado grado de unicidad, muy superior al resto de tecnologías biométricas comúnmente empleadas. Además la protección que le ofrece la córnea hace que dicho patrón permanezca inalterable durante la vida del sujeto. Por otra parte, el patrón de los vasos sanguíneos de la retina presenta una mayor unicidad que el patrón de iris, esto, unido a la casi imposible modificación de esta característica convierten a la exploración de la retina en una modalidad biométrica aún más segura que la de reconocimiento de iris.

Los métodos de autenticación basados en patrones oculares presentan una tasa de falsa aceptación cercana a cero, debido a la elevadísima unicidad de las características biométricas que usan. El tejido ocular se degrada muy rápidamente tras la muerte del individuo, de manera que el sistema biométrico puede detectar con relativa facilidad si el ojo pertenece a un organismo vivo o no. Para evitar la utilización de imágenes de alta resolución de un ojo humano frente al sistema se llevan a cabo pruebas colaterales en la detección, como la comprobación de la circulación sanguínea (pulsaciones) en el tejido ocular.

Sin embargo, a pesar de todo ello, este tipo de sistemas no cuentan con una elevada aceptación por parte de los usuarios. En primer lugar el hecho de que el usuario tenga que acercar su rostro al sistema y que un haz de luz láser pase por su ojo provoca incomodidad en el usuario, y una cierta desconfianza inicial al uso del sistema (aunque en realidad el sistema

solamente capte una simple fotografía). Es por eso por lo que esta tecnología solo se ha implantado en entornos de extrema seguridad, donde el grupo de usuarios es muy reducido, y se es consciente del nivel de seguridad requerido. Otro inconveniente importante es que el estudio del ojo puede ser altamente intrusivo, ya que puede desvelar información privada que los usuarios no tienen por qué querer dar, como ciertas enfermedades o el consumo de alcohol o drogas. Es importante destacar el elevado coste de este tipo de sistemas ópticos.

- Rostro

El método de reconocimiento facial es el método de identificación que de manera más natural y con mayor frecuencia realiza nuestro cerebro, ya que a diario necesitamos reconocer a las personas que nos rodean. Los sistemas de identificación por rostro disponen de una cámara que graba al usuario y analizan sus características faciales para realizar la identificación. Este método puede resultar enormemente cómodo para el usuario, ya que puede pasar incluso inadvertido para él. Sin embargo presenta un gran inconveniente, que es la variabilidad de las características del rostro del sujeto a lo largo del tiempo. Factores como la edad, la expresión o simples cambios en el rostro (peinado, barba, gafas) dificultan el proceso de identificación.

- Oreja

Estudios forenses han demostrado que la oreja de un ser humano posee multitud de características que son propias del mismo, lo que significa que pueden ser utilizadas para su identificación. Esta modalidad, de estudio bastante reciente, requiere que el usuario descubra su oreja frente a una cámara, lo que puede resultar bastante intrusivo para determinadas culturas, e incómodo para personas de pelo largo.

- Voz

El reconocimiento de voz es una modalidad de identificación biométrica bastante estudiada que utiliza la voz del sujeto para realizar su identificación. Existen multitud de algoritmos, tanto para obtener los rasgos característicos de la voz, como para realizar el proceso de comparación con los patrones almacenados. Determinadas aplicaciones requieren que el sujeto pronuncie de la manera más fiel posible un código de acceso previamente grabado, ya sea un número secreto o simplemente su nombre y apellidos. Este tipo de sistemas son muy sensibles a ataques de repetición mediante grabadoras de sonidos. Otras técnicas más avanzadas son independientes del texto pronunciado, y tratan de identificar ciertas características de la voz del usuario y no lo que realmente éste dice. En estas aplicaciones se invita al usuario a pronunciar una frase diferente cada vez que desee acceder al sistema.

La identificación mediante la voz es una modalidad de identificación de muy bajo coste, ya que no son necesarios equipos muy avanzados para ser llevada a cabo, y está bastante aceptada. En algunas aplicaciones, como servicios de atención telefónica, puede resultar inapreciable para el usuario. Sin embargo los sistemas de reconocimiento por voz presentan una serie de inconvenientes importantes: el tono de voz del usuario puede sufrir grandes cambios como consecuencia de su estado de ánimo, la edad o simplemente enfermedades tan

comunes como el resfriado o la afonía. Este tipo de factores repercuten directamente en la calidad del sistema, ya que pueden producir falso rechazo. Para un óptimo funcionamiento del sistema se requiere un entorno con una buena acústica y carente de ecos y ruidos externos, de manera que las interferencias se reduzcan al máximo. Esto no siempre es posible, por lo que se buscan algoritmos que minimicen el efecto de estos factores.

- Andadura, dinámica de teclado y firma

Todas ellas son modalidades de identificación biométricas basadas en características de comportamiento, y por tanto susceptibles de ser imitadas.

Las técnicas de reconocimiento de andadura analizan la manera particular en la que un individuo camina, son técnicas de desarrollo muy reciente y presentan como principal inconveniente que no son aplicables a personas que no pueden caminar, ni tampoco en aquellas otras que presenten una lesión que les impida caminar con normalidad.

Las técnicas de dinámica de teclado utilizan para la identificación el ritmo característico con el que una persona es capaz de escribir con un teclado. Los estudios realizados han determinado un alto grado de unicidad en este comportamiento, sin embargo, además de los problemas de imitación que toda modalidad de identificación basada en comportamiento presenta, tiene la limitación de no ser aplicable a usuarios con dificultades a la hora de teclear.

El reconocimiento de firma se trata de una modalidad profundamente estudiada, y utilizada desde hace mucho tiempo como método de identificación de personas. La evolución de la tecnología ha intentado solventar los problemas de falsificación siempre ligados a ella, y hoy en día se habla de técnicas de verificación de escritura, en las cuales mediante un lápiz especial se analizan características dinámicas de la escritura del individuo, como el tiempo necesario para firmar, la presión del lápiz o el ángulo de colocación del mismo.

- Olor

Es una modalidad de desarrollo muy reciente que se basa en el análisis del olor corporal del individuo para su reconocimiento. Los sensores de olor, aún en desarrollo, utilizan un proceso químico similar al que se produce entre la nariz y el cerebro, de manera que agentes externos como otros olores o perfumes no enmascaren el olor particular de cada uno.

- ADN

El ADN es sin lugar a dudas la única característica biométrica que permite identificar unívocamente y sin ambigüedades a un sujeto. Ya que en dicho código está contenida toda la información genética del individuo, que es única e irrepetible. La principal limitación en la actualidad es la tecnología, ya que se requieren sistemas de identificación automática en tiempo real capaces de extraer las características genéticas del individuo, y que además resulten cómodos de usar para el usuario. Como toda nueva tecnología la identificación biométrica mediante ADN plantea ventajas evidentes, pero también riesgos, en este caso sobre todo intrusivos, como la invasión de la privacidad y el control exhaustivo que produciría el disponer del código genético completo de un usuario.

1.2. Tarjetas inteligentes

1.2.1. Introducción a las tarjetas inteligentes

Una tarjeta inteligente es una tarjeta de plástico cuyas dimensiones se encuentran normalizadas, y que en su interior contiene un microcontrolador (microprocesador y memoria). Habitualmente las tarjetas inteligentes se clasifican según el método de comunicación que utilicen, así pueden distinguirse los siguientes tipos de tarjetas:

- **Con contactos**: la tarjeta dispone de unos contactos metálicos desde los cuales se realiza una conexión física con los contactos de la unidad lectora.
- **Sin contactos**: este tipo de tarjetas, denominadas *tags*, no requieren de un contacto físico entre la tarjeta y la unidad lectora, y la comunicación se realiza de forma aérea por radiofrecuencia. Se utilizan para la identificación de objetos y personas, sobre todo en aplicaciones de control de acceso a edificios.
- **Híbridas**: disponen tanto de contactos como de una antena de RF, y permiten aprovechar las ventajas de las dos tecnologías anteriores.

1.2.2. Bloques de una tarjeta inteligente

Las tarjetas inteligentes se caracterizan por contener un circuito integrado en su interior encargado de almacenar y procesar información. Este circuito integrado es un microcontrolador, es decir, un microprocesador, cierta memoria asociada y determinados periféricos. El esquema de los diferentes bloques que integra una tarjeta inteligente es el siguiente:

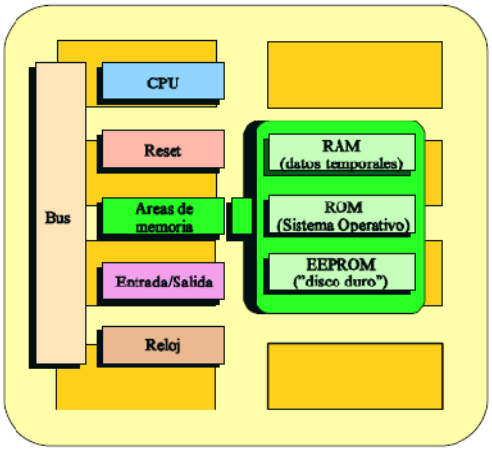

Fig. 1.4: Bloques de una tarjeta inteligente (Fuente: [10]).

- Unidad central de proceso (CPU)

La CPU es el componente más importante de la tarjeta inteligente, se encarga de ejecutar las operaciones a bajo nivel, interconectando entre sí los distintos bloques funcionales de la tarjeta y tomando las decisiones necesarias. El sistema operativo de la tarjeta permite

controlar el conjunto de operaciones que se pueden realizar dentro de la tarjeta, de manera que el bloque de la CPU pasa prácticamente desapercibido para el usuario final.

Tradicionalmente la CPU de una tarjeta inteligente se ha basado en microprocesadores de 8 bits, como la familia 8051 de Intel, o la familia 68HC05 de Motorola. En otros ámbitos como en los ordenadores personales los microprocesadores han sufrido grandes evoluciones, apareciendo procesadores de 64 bits. En el campo de las tarjetas inteligentes la evolución no ha sido tan notoria, aunque han aparecido procesadores de 16 bits, e incluso algunos de 32 bits con tecnología RISC (Reduced Instruction Set Code), sobre todo en tarjetas por radiofrecuencia, la mayoría de las tarjetas siguen utilizando procesadores de 8 bits. La explicación es simple, las aplicaciones habituales de las tarjetas inteligentes no necesitan mayor potencia de cálculo para realizar satisfactoriamente su tarea, y el precio de cambio de adoptar procesadores más potentes resulta demasiado elevado.

Por otra parte la evolución de la tecnología ha permitido la aparición de coprocesadores matemáticos en ciertas tarjetas, que permiten realizar los cálculos de forma mucho más rápida. También se han introducido bloques adicionales con funcionalidades criptográficas, así los algoritmos de cifrado requeridos para operaciones seguras pueden ser realizados en muy poco tiempo, una cantidad mucho menor que la necesaria si todo el cálculo lo tuviera que realizar directamente el microprocesador.

- Memoria

La memoria de una tarjeta inteligente se compone de varios bloques de memoria, cada uno de los cuales puede ser de un tipo diferente de memoria, según la función que desempeñe. Habitualmente se distinguen los siguientes tipos de memoria:

- **Memoria de acceso aleatorio** (RAM): este tipo de memoria es volátil, es decir su contenido se pierde cuando cesa la alimentación. Es utilizada por el microprocesador para ejecutar las instrucciones, depositando en ella resultados intermedios y también datos de entrada/salida. Esta memoria suele ser transparente tanto para el programador como para el usuario final, ya que no es directamente accesible, y es el propio sistema operativo de la tarjeta inteligente el encargado de gestionarla. Habitualmente el tamaño de la memoria RAM es de muy pocos kilobytes, lo que significa que su uso tiene que estar muy optimizado.
- **Memoria de solo lectura** (ROM): esta memoria contiene el código del sistema operativo de la tarjeta inteligente. Su tamaño suele ser de unos pocos kilobytes, de manera que el sistema operativo se desarrolla en un lenguaje de bajo nivel para aprovechar mejor el espacio disponible. La memoria ROM también es transparente, lo que impide cualquier tipo de modificación o lectura por parte de un usuario malintencionado, y aporta seguridad a las operaciones realizadas.
- **Memoria de solo lectura re-escribible eléctricamente** (EEPROM): se trata de la única memoria accesible directamente tanto para el programador como para el usuario final, en ella se almacenan datos necesarios para realizar las operaciones. Su tamaño, aunque mayor que las anteriores, suele ser también del orden de kilobytes, y puesto

que es la única memoria aprovechable por el usuario su tamaño es el que se asocia al tamaño de la memoria de la tarjeta. Gracias al sistema operativo de la tarjeta esta memoria no volátil se organiza en ficheros y directorios, resultando más sencilla su utilización.

Al contrario que sucede con la CPU, en las memorias utilizadas en las tarjetas inteligentes sí que se ha producido una notable evolución en las prestaciones. En sus inicios estas tarjetas no contaban con memorias re-escribibles, ya que o eran muy caras o la tecnología no estaba lo suficientemente desarrollada, de manera que las tarjetas eran de usar y tirar. La evolución tecnológica y el abaratamiento de los costes permitieron introducir las memorias EEPROM, y las tarjetas pudieron ser borradas y escritas un elevado número de veces. Posteriores mejoras tecnológicas redujeron los tiempos de borrado y escritura, y consiguieron aumentar la escala de integración incrementando la capacidad de almacenamiento.

- Bloque de entrada y salida

El bloque de entrada y salida permite la comunicación de la tarjeta inteligente con el exterior. Esta comunicación se realiza de forma serie, para ello el bloque obtiene los datos uno tras otro, los trata y los entrega a la CPU para que los procese. Para realizar este procedimiento el sistema operativo de la tarjeta debe disponer de unas rutinas de control muy robustas, de modo que la comunicación se realice según los protocolos de comunicación específicos de las tarjetas. Este bloque difiere de unas tarjetas a otras según la tarjeta sea con contactos o sin contactos. Si la tarjeta es sin contactos la comunicación a nivel físico se realiza por radiofrecuencia, de manera que la tarjeta tendrá que modular y demodular la señal de información, además de compensar en la medida de lo posible las posibles interferencias.

- Sistemas de control de la alimentación

Este bloque supervisa los niveles de alimentación de la tarjeta, de manera que se encuentren en todo momento dentro de unos umbrales preestablecidos, lo que garantiza la seguridad tanto física como lógica de la información almacenada en la tarjeta. Si la alimentación supera en algún momento un límite determinado este bloque debe cortar la alimentación de la tarjeta, para así evitar posibles daños a la CPU o a las memorias. Si por el contrario se produce una caída de tensión este bloque debe ser capaz de mantener la tensión de alimentación el tiempo suficiente para interrumpir a la CPU, y que ésta guarde el estado de ejecución.

- Circuito de arranque

Este bloque permite enviar a la tarjeta una señal de reset, que inicialice la CPU con unos valores por defecto que garanticen el correcto funcionamiento de la tarjeta. Durante el funcionamiento normal de la tarjeta puede enviarse una señal de reset para que ésta vuelva a sus condiciones de inicio.

- Sistema de supervisión del reloj

Se encarga de vigilar el reloj de funcionamiento de la CPU entregado desde el exterior, de modo que para un correcto funcionamiento éste debe tener unos niveles y una estabilidad

determinados. Si este bloque detecta alguna anomalía en el reloj suministrado puede interrumpir a la CPU para que cese su funcionamiento.

1.2.3. Sistema operativo de la tarjeta inteligente (SOTI)

El SOTI proporciona una interfaz de comandos de alto nivel que facilita la utilización de la tarjeta inteligente. Una vez cargado la tarjeta se comporta como un sistema portátil de información extremadamente seguro, para ello realiza una serie de funciones adiciones destinadas a garantizar la seguridad en todas las operaciones que se realicen:

- Bloquea la ejecución de todas las instrucciones hasta que se reciba una señal externa de reset. Tras el reset inicializa los parámetros de la tarjeta, envía una respuesta y espera la recepción de una instrucción.
- Al recibir una instrucción realiza una serie de comprobaciones, comprueba que la instrucción es válida y sus parámetros sean correctos, verifica que las condiciones de seguridad necesarias para la ejecución de una determinada instrucción son satisfechas, así como el estado de los datos referidos dentro de la tarjeta. Si todas las condiciones se cumplen la tarjeta ejecuta la instrucción y devuelve el resultado, en caso contrario se produce un error y no se lleva a cabo la operación.
- Gestiona las memorias y los datos almacenados, de manera que el usuario no utiliza direcciones de memoria que hagan referencia a una posición determinada, sino únicamente identificadores de ficheros y de directorios.
- Incorpora a los ficheros unas reglas de acceso determinadas conforme a unas claves. Estas claves se implementan de forma segura, verificándose siempre en el interior de la tarjeta y no pudiendo ser nunca leídas desde el exterior.

Gracias al SOTI la práctica totalidad de los bloques funcionales que componen una tarjeta inteligente son enmascarados, así por tanto una tarjeta inteligente puede caracterizarse por:

- Una estructura de datos: que define el sistema de archivos, directorios y archivos, que almacenan los datos en la tarjeta.
- La capacidad de almacenamiento de datos en la tarjeta, es decir, la cantidad de memoria EEPROM disponible para el usuario final.
- Una arquitectura de seguridad que determine los mecanismos de seguridad disponibles para proteger los datos.
- Un conjunto de instrucciones, imprescindibles para operar con la tarjeta.

El SOTI se encuentra ubicado en la ROM de la tarjeta inteligente, pero en algunas ocasiones como durante el proceso de desarrollo y depurado del SO puede residir en la memoria EEPROM. En este caso se prueban los algoritmos y se habilita un mecanismo para que el fabricante pueda modificarlos. Una vez verificado el correcto funcionamiento del SOTI se pasa a memoria ROM, liberando el espacio que ocupaba en la memoria EEPROM.

1.2.4. Sistema de ficheros de una tarjeta inteligente

Tal y como se ha mencionado anteriormente el SOTI se encarga de todo el proceso de gestión de memoria, así el usuario lo que realmente ve es un sistema de ficheros. El SOTI puede permitir una estructura de ficheros lineal, de manera que solo puedan existir ficheros distribuidos en un mismo nivel, o una estructura jerárquica, permitiendo directorios que contengan ficheros u otros directorios. Los ficheros pueden referenciarse habitualmente mediante un código hexadecimal, un nombre largo o un nombre corto, la posibilidad de utilizar una u otra manera depende exclusivamente de la implementación del SOTI.

Se distinguen los siguientes tipos de ficheros en el sistema de ficheros de una tarjeta inteligente:

- **Fichero maestro** (MF): representa el directorio raíz de la tarjeta.
- **Fichero dedicado** (DF): representa un directorio, es decir un fichero especial que puede contener ficheros e incluso otros ficheros dedicados si el SOTI lo soporta.
- **Fichero elemental** (EF): se trata de un fichero que contiene datos. Pueden distinguirse varios tipos de ficheros elementales según su uso dentro de la tarjeta. Así pues existen ficheros internos, como los ficheros de claves, que sólo pueden ser escritos desde el exterior y nunca leídos. Ficheros sin estructura, llamados transparentes, donde la información no se encuentra estructurada. Ficheros organizados en registros, donde la información está organizada en bloques, ya sean de longitud fija o variable. Y ficheros de trabajo, que son los que utiliza el usuario libremente según las condiciones de acceso establecidas en la aplicación.

1.2.5. Tarjetas inteligentes sin contactos

Las tarjetas inteligentes sin contactos, comúnmente denominadas *tags* RFID o simplemente *tags*, se caracterizan porque la comunicación con ellas se realiza a través de señales de radiofrecuencia, sin que exista contacto físico directo entre las mismas y la unidad lectora.

Existen tarjetas RFID denominadas pasivas, capaces de extraer de la señal que reciben la energía necesaria para efectuar la comunicación. Este tipo de tarjetas están dotadas de un componente que recibe el nombre de *transponder*, formado por un circuito integrado y una antena, que absorbe energía electromagnética procedente de la unidad lectora siempre y cuando la tarjeta se encuentre dentro de su campo de acción.

Otro tipo de tarjetas, denominadas tarjetas activas, son capaces de emitir por sí mismas una señal de RF, ya que contienen en su interior una batería. Estas tarjetas permiten una recepción y transmisión a distancias mayores que las tarjetas pasivas, ya que aprovechan la energía almacenada para transmitir señales de mayor potencia.

Según la distancia requerida entre la tarjeta inteligente y la unidad lectora para un funcionamiento adecuado las tarjetas sin contactos pueden clasificarse en tarjetas de acoplo, tarjetas de proximidad y tarjetas de vecindad. Las tarjetas de acoplo son aquellas cuya

distancia de funcionamiento es próximo a cero, sus características se encuentran definidas en la norma IS 10536. Las tarjetas de proximidad tienen una distancia de operación de aproximadamente 10cm, y su regulación se recoge en la norma IS 14443. Las tarjetas de vecindad son las que permiten una mayor distancia de funcionamiento, con distancias muy superiores a los 10cm, sus características y funcionamiento están regulados en la norma IS 15693.

En los sistemas de control de acceso biométricos el patrón característico del usuario puede estar contenido en una tarjeta inteligente, siendo las tarjetas inteligentes con contactos o sin contactos de proximidad las más frecuentemente utilizadas. La segunda parte de la norma IS 14443 establece las características de funcionamiento de las tarjetas de proximidad. Este tipo de tarjetas utilizan una frecuencia de funcionamiento de 13.56MHz, con una tolerancia de ±7kHz, además establece que el nivel de intensidad de campo magnético necesario para el correcto funcionamiento de la tarjeta debe estar comprendido entre 1.5 y 7.5A/m eficaces.

Existen dos familias diferentes de tarjetas inteligentes de proximidad, las tarjetas de tipo A (Mifare® de Philips) y las tarjetas de tipo B (Gemplus y Motorola). Aunque las especificaciones y modos de funcionamiento de ambas tarjetas se recogen en la norma, la tecnología Mifare® es la tecnología de tarjetas inteligentes sin contactos que más popularidad ha alcanzado, convirtiéndose de hecho en un estándar de facto. El modo de funcionamiento habitual de una tarjeta de proximidad es el siguiente:

- La tarjeta se activa al recibir una señal de radiofrecuencia procedente de la unidad lectora, y espera en silencio la recepción de un comando.
- La unidad lectora envía un comando a la tarjeta, que tras procesarlo envía una respuesta.
- Se produce un flujo de mensajes comando-respuesta entre la unidad lectora y la tarjeta inteligente, teniendo en todo momento la unidad lectora la iniciativa y el control.

La segunda parte de la norma IS 14443 establece también las características de la señal transmitida por el interfaz aéreo. Las modulaciones y codificaciones utilizadas en las tarjetas de proximidad Mifare® son las siguientes:

Unidad lectora -> tarjeta	ASK al 100%. Miller modificado
Tarjeta -> unidad lectora	ASK. Manchester

Fig. 1.6: Modulaciones y codificaciones en tarjetas de proximidad

El tipo de modulación y codificación empleado difiere según el sentido de la comunicación, siendo la velocidad de transmisión de 106kbps en cada uno de los sentidos de la comunicación.

El hecho de que las tarjetas por proximidad operen a una distancia máxima de 10cm tiene como inconveniente la posibilidad de que existan más de una tarjeta en el radio de acción de la unida lectora. Esto hace necesario la existencia de un protocolo de inicialización que evite las colisiones en la comunicación y permita a la unidad lectora controlar varias tarjetas de manera simultánea.

1.3. Biometría y tarjetas inteligentes en sistemas de control de acceso.

Un sistema de control de acceso es un sistema en el cual se controla la entrada de usuarios, de manera que el acceso de usuarios está restringido única y exclusivamente a personas autorizadas. En este tipo de sistemas pueden definirse áreas con diferentes niveles de seguridad, a las cuales solamente podrán acceder aquellos usuarios que dispongan de los permisos de acceso necesarios.

El primer problema que hay que hacer frente en un sistema de control de acceso es como identificar a un usuario cometiendo el menor error posible. De manera que se tenga la certeza de la identidad del individuo frente al sistema, y de si debe permitírsele o no el acceso. Esta identificación puede realizarse de muy diversas maneras, algunas de ellas son las siguientes:

- Solicitando al usuario una información secreta, que solamente sea conocida por él y por el sistema, como puede ser una contraseña, o algún elemento material que posea, como una tarjeta identificativa. El principal inconveniente en la utilización de este tipo de técnicas radica en la relativa facilidad de una persona malintencionada para hacerse con esta información y suplantar la identidad del usuario, poniendo en riesgo la seguridad del sistema.

- A partir de una característica biológica del sujeto en cuestión. Aquí es donde entra en juego la biometría. Las características biológicas de una persona van siempre con el usuario, no pueden olvidarse, perderse o robarse, y son difícilmente falsificables. Para ello debe elegirse una característica biométrica que presente un elevado índice de unicidad (mayor cuanto más alto sea el nivel de seguridad requerido en la aplicación de control de acceso), y cuya medida resulte cómoda de utilizar para el usuario. Éste es el caso de la huella dactilar. Es totalmente viable el estudio de un sistema de control de acceso biométrico basado en huella dactilar, en el cual el usuario posicione su dedo en un lector que verifique su identidad y decida si debe o no autorizar su acceso.

Para que el usuario pueda ser identificado es necesario que su identidad esté almacenada en algún sitio. En el caso del reconocimiento por huella se extrae un patrón biométrico denominado vector de características, que representa a la huella dactilar. Si el número de usuarios es muy elevado resulta necesario dotar a los terminales biométricos de control de acceso de una gran memoria de almacenamiento, o en su defecto disponer de una base de datos siempre on-line que almacene los patrones, que pueda ser consultada en el momento de realizar la identificación.

Como alternativa pueden utilizarse tarjetas inteligentes en el sistema de control de acceso, que contengan el patrón biométrico del usuario. Las tarjetas inteligentes son dispositivos extremadamente seguros, que el usuario puede llevar consigo y utilizar para presentar su identidad, y demostrarla con ayuda de alguna de sus características biológicas.

2. DESCRIPCIÓN DEL SISTEMA DE CONTROL DE ACCESO DE BIOMETRIKA

2.1. Componentes de un sistema FxGate

La solución desarrollada por Biometrika para aplicaciones de control de acceso y aplicaciones de control de tiempo y asistencia se trata de una arquitectura cliente-servidor, conocida por el nombre de Sistema FxGate. Un sistema FxGate está formado por tres bloques funcionales diferentes, tal y como puede observarse en la siguiente figura:

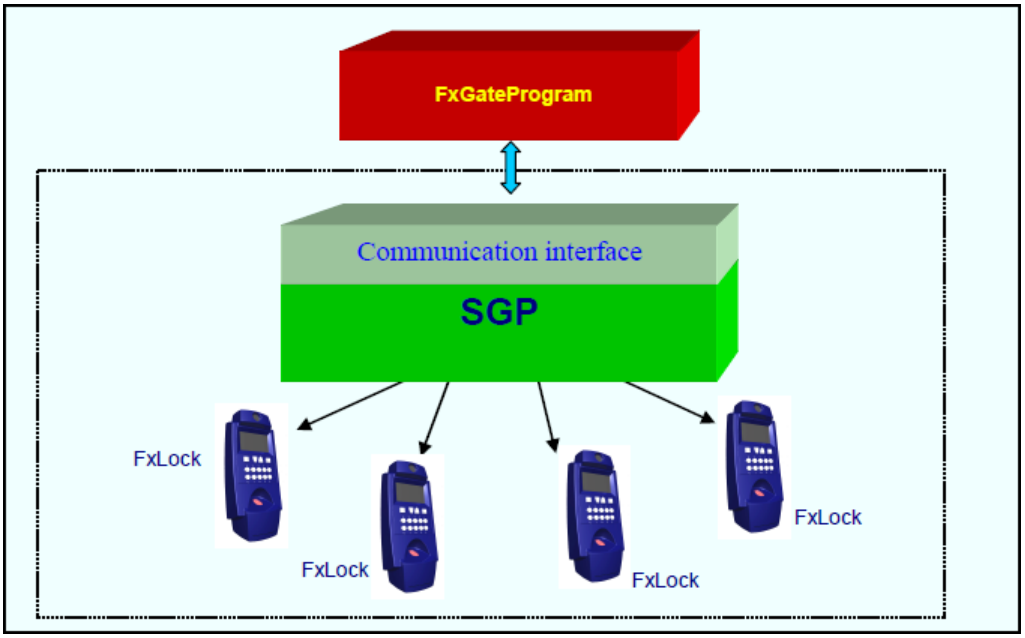

Fig. 2.1: Componentes de un Sistema FxGate (Fuente: [11]).

- Terminales FxLock

Un terminal FxLock es un dispositivo biométrico situado en aquellos puntos de acceso al sistema en los cuales se desea restringir el acceso de usuarios (aplicaciones de control de acceso) o bien monitorizar los instantes de entrada y salida de usuarios (aplicaciones de control de tiempo y asistencia). Está dotado de un lector de huella dactilar y un procesador, lo suficientemente potente como para procesar en el mismo dispositivo la huella dactilar del usuario.

- El Servidor SGP

El Servidor SGP es quien lleva a cabo todo el peso de la gestión del sistema completo. Proporciona una interfaz centralizada para gestionar todos los terminales FxLock existentes en el sistema, ya que es el único componente que puede comunicarse directamente con los terminales FxLock. Es una aplicación software programada por Biometrika que debe instalarse en un equipo conectado en red local con los diferentes terminales FxLock.

- El Programa FxGate

Es una aplicación software desarrollada con ayuda del FxGate SDK de Biometrika. Proporciona una interfaz de gestión al administrador del sistema, ya que a través de esta aplicación es posible controlar todo el sistema de control de acceso (o sistema de control de tiempo y asistencia). El programa FxGate es una aplicación cliente del Servidor SGP, que es quien realiza la verdadera gestión del sistema completo, y su funcionalidad se limita a enviar órdenes al servidor de gestión y recibir las respuestas correspondientes. Para la comunicación entre el Programa FxGate y el Servidor SGP ambas aplicaciones pueden estar localizadas en el mismo ordenador, o bien estar instaladas en equipos diferentes, caso en el cual debe existir una conexión de red entre los dos ordenadores. También es posible utilizar el Programa FxGate para gestionar los datos contenidos en tarjetas inteligentes con contactos asociadas a usuarios del sistema, siempre y cuando se use en la aplicación el BioCard SDK de Biometrika.

2.2. Terminales FxLock

El FxLock de Biometrika se trata de un terminal biométrico destinado a aplicaciones de control de acceso o aplicaciones de tiempo y asistencia, basadas en huella dactilar. Está dotado de un sensor de huella con gran área de detección, y un procesador que permite llevar a cabo dentro del propio dispositivo el proceso de identificación de un usuario en un tiempo de unos 0.8 segundos. Dispone de una pantalla LCD y un teclado numérico y de navegación, que permiten visualizar información sobre el acceso al sistema y configurar el terminal.

Cada usuario tiene asociados un PIN numérico que debe introducir para poder acceder al sistema. Este PIN es utilizado por el FxLock para clasificar los patrones de huella de los usuarios que tiene almacenados en su interior. Cuando un usuario pretenda acceder al sistema en primer lugar debe teclear su PIN y se le invitará a posicionar su dedo en el lector. El terminal FxLock compara el patrón biométrico obtenido con todos los patrones biométricos almacenados de usuarios del mismo PIN (modo identificación, 1:N). Si todos los usuarios tienen asociados un PIN diferente el FxLock trabaja en modo verificación, 1:1 y efectúa una única comparación.

Un terminal FxLock pueden funcionar de dos maneras diferentes, de forma independiente (modo autónomo) o conectado en red (controlado por un servidor centralizado).

- En el modo autónomo el propio FxLock controla el acceso a una puerta y la gestión se realiza desde el mismo dispositivo. Este modo de funcionamiento resulta útil en aplicaciones de control de acceso donde el número de usuarios no es muy elevado, y el número de puntos de acceso es bajo. La gestión de usuarios y configuración del terminal es efectuada por un administrador desde los menús de configuración del FxLock.

- En el modo centralizado la gestión del sistema es efectuada de manera remota por el Servidor SGP. Este modo de funcionamiento está orientado a aplicaciones de control de acceso en las que el número de terminales es grande, o a aplicaciones de control de

tiempo y asistencia en las cuales se desea tener un control más detallado sobre el sistema completo. Para la conectividad de los terminales FxLock con el Servidor SGP existen dos interfaces posibles, a través de una conexión de red Ethernet TCP/IP, o a través de puerto serie RS232. Periódicamente los terminales FxLock envían información sobre su estado al Servidor SGP, así como registros de los accesos al sistema tan pronto como suceden. Si la conexión entre el Servidor SGP y un terminal FxLock se cae el terminal FxLock pasa a operar en modo autónomo. Los usuarios previamente almacenados en el terminal, e insertados desde el Servidor SGP, pueden seguir accediendo al sistema. También pueden activarse temporalmente nuevos usuarios desde el menú de configuración del FxLock, pero tan pronto se restablezca la conexión con el Servidor SGP, el FxLock pasará a funcionar en modo centralizado.

Los terminales FxLock pueden equiparse opcionalmente con una tarjeta de relés que permite controlar hasta cuatro dispositivos diferentes, como por ejemplo cerraduras electrónicas o alarmas. La comunicación con esta tarjeta se realiza a través de un puerto serie RS485 de manera segura, mediante un protocolo de reto-respuesta. Los usuarios pueden tener asociados hasta 4 permisos de acceso diferentes, correspondientes a cada uno de los diferentes relés, de manera que cuando se autoriza el acceso a un usuario la pantalla del FxLock muestra un menú de selección. Al seleccionar una de las opciones se activa el contacto correspondiente y por ejemplo la cerradura de una puerta controlada por el FxLock se abre.

Es posible equipar a un terminal FxLock con una batería de emergencia, utilizable en el caso de cortes de electricidad. Y también dotar al terminal de un lector de tarjetas inteligentes, bien de contactos o bien de radiofrecuencia (RFID). En el caso de utilizar tarjetas inteligentes que contengan el patrón biométrico del usuario se elimina la necesidad de una base de datos centralizada que contenga todos los patrones biométricos. Y tampoco resulta necesario que el terminal FxLock almacene patrones. Un usuario con tarjeta inteligente presenta su identidad con la tarjeta inteligente, y el FxLock la verifica comparando el patrón biométrico extraído de la huella del usuario con el patrón contenido en la tarjeta.

Por último se presentan con mayor detalle algunas de las características técnicas de los terminales FxLock:

- Sensor óptico de huella dactilar con 569 puntos por pulgada de resolución y área de adquisición de 13.2x25 mm^2.

- Microprocesador RISC de 32bits, core ARM9 y frecuencia de reloj a 200MHz. 32MB de memoria RAM.

- Memoria Flash interna de 32MB. Permite almacenar los datos y el patrón biométrico de hasta 10.000 usuarios.

- Display LCD retro-iluminado con 4 líneas de 20 caracteres cada una.

- Teclado ergonómico de 14 botones. Numérico más teclas de navegación por los menús, selección y cancelación.

2.3. Servidor SGP

La aplicación SGP Server se distribuye conjuntamente con el FxGate SDK, y debe instalarse en un equipo con sistema operativo Windows, conectado en red con los diferentes terminales FxLock del sistema. El SGP Server no incluye ningún tipo de documentación que explique el proceso de instalación y configuración, ni tampoco un manual de uso que indique al administrador del sistema cual es el funcionamiento del servidor. Por lo tanto a continuación se detalla toda la información que se ha podido averiguar.

2.3.1. Instalación y configuración del Servidor SGP

Para la instalación del Servidor SGP hay que ejecutar el archivo setup.exe contenido en el directorio SGP Server del FxGate SDK. Una vez iniciada la instalación se nos preguntará por el directorio de instalación del SGP Server (por defecto C:\Archivos de programa\Biometrika\SGP_300), y por el tipo de instalación, tal y como puede observarse a continuación:

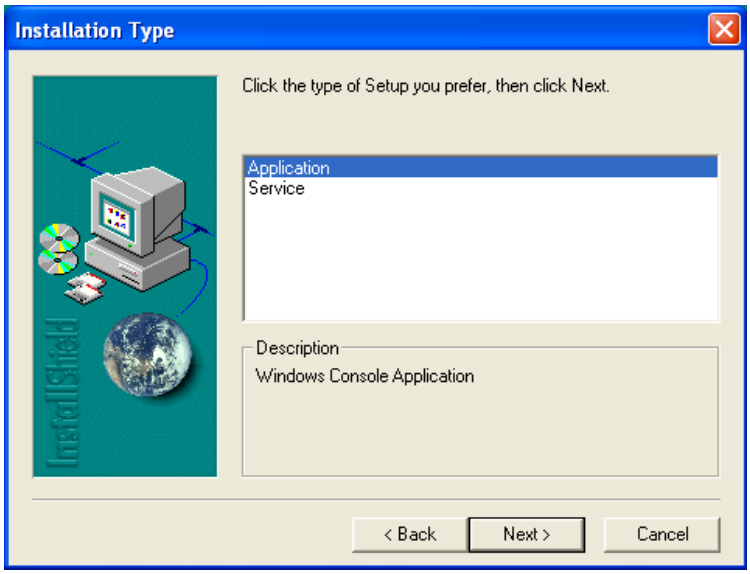

Fig. 2.2: Asistente de configuración del Servidor SGP. Tipo de instalación.

El servidor SGP puede instalarse de dos maneras diferentes, como aplicación de consola o como servicio de Windows NT/2000.

- Aplicación de consola: en el caso de instalar el servidor como aplicación de consola hay que ejecutar el programa server.exe cada vez que se quiera lanzar el Servidor SGP. Se abrirá una aplicación de consola que muestra en tiempo real una breve descripción de los eventos producidos por la comunicación del Servidor SGP con los terminales FxLock y el Programa FxGate. Al cerrar esta ventana se detiene la ejecución del Servidor SGP.

- Servicio de Windows NT/2000: si instalamos el servidor como servicio de Windows el Servidor SGP se iniciará automáticamente al arrancarse Windows. De esta manera siempre que el equipo esté encendido el Servidor SGP estará listo para atender peticiones de los terminales FxLock y del Programa FxGate. El principal inconveniente de utilizar el Servidor SGP como servicio de Windows es que se renuncia a la información adicional que ofrece la aplicación de consola.

Una vez seleccionado el modo de instalación para el Servidor SGP y copiados los archivos necesarios en el sistema se inicia el asistente de configuración del servidor. Este asistente puede abrirse de nuevo ejecutando la aplicación ConfigWizard.exe, contenida en el directorio de instalación del servidor.

La primera ventana del asistente de configuración se muestra en la Figura 2.3.

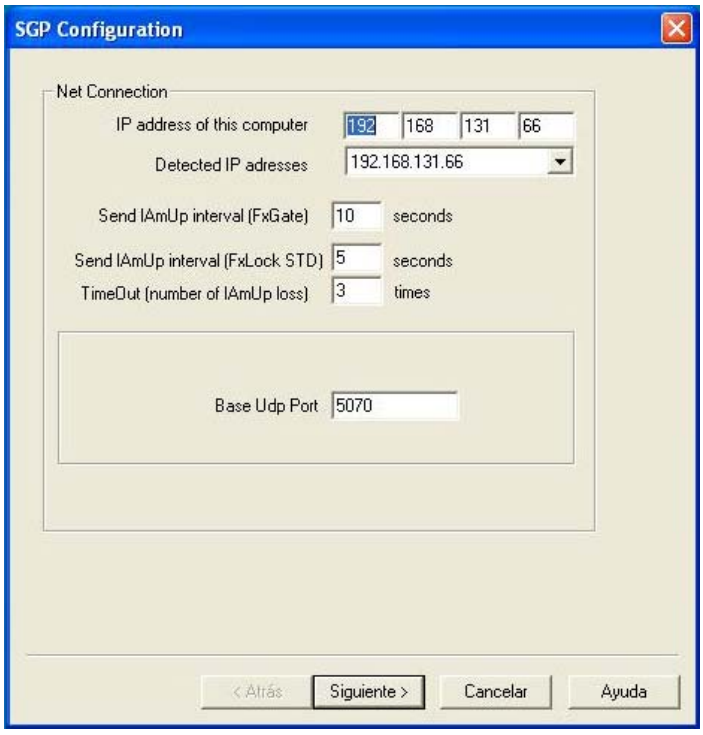

Fig. 2.3: Asistente de configuración del Servidor SGP. Configuración de red.

En ella pueden configurarse los siguientes parámetros:

- La dirección IP del Servidor SGP, que coincide con la dirección IP del adaptador de red utilizado para conectar el Servidor SGP con el resto de elementos del Sistema FxGate.

- El intervalo de tiempo entre dos envíos consecutivos por parte del Servidor SGP de una señal 'IAmUp' a través de la interfaz de red. Esta señal es utilizada por los terminales FxGate y el Programa FxLock para conectarse con el servidor SGP.

- El puerto UDP base (se usan 4 puertos UDP consecutivos) en el cual se espera que escuche el Programa FxGate la recepción de la señal 'IAmUp'.

- El intervalo de tiempo entre dos envíos consecutivos por parte del servidor SGP de la señal 'IAmUp' por el puerto serie.

- El número de envíos consecutivos sin respuesta de la señal anterior, utilizados para considerar que el terminal FxLock conectado por puerto serie no está presente.

A continuación se muestra una ventana para la configuración criptográfica del Servidor SGP, que puede observarse en la Figura 2.4. El sistema FxGate se trata de un sistema criptográfico asimétrico basado en la utilización de claves públicas y privadas. Todos los componentes del sistema poseen sus propias claves, y las utilizan para autenticarse y cifrar los mensajes intercambiados.

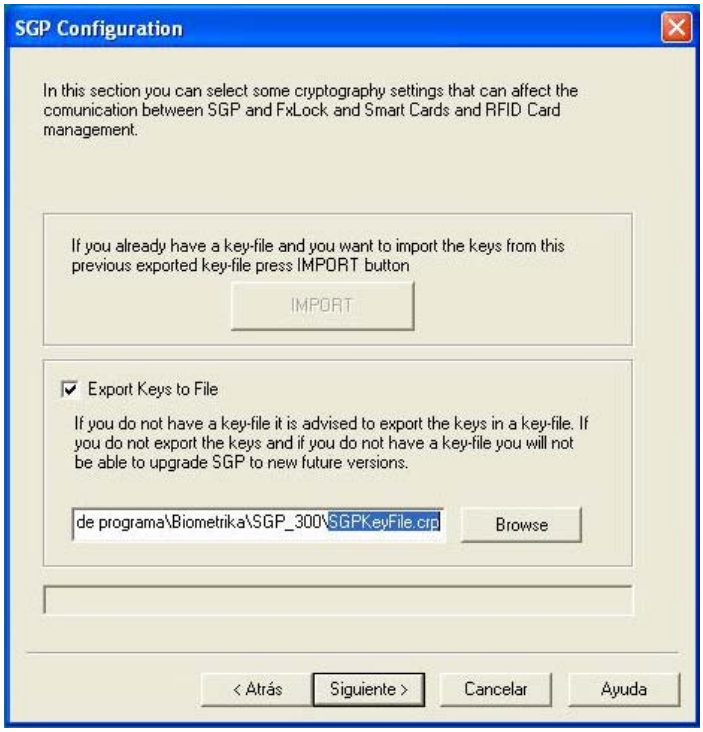

Fig. 2.4: Asistente de configuración del Servidor SGP. Configuración criptográfica.

Desde esta ventana se puede generar el fichero de claves del Servidor SGP (SkData.bin), y también exportarlo o importar un fichero de claves de una instalación anterior. Si se reinstala el Servidor SGP y se genera un nuevo fichero de claves hay que tener en cuenta las siguientes consideraciones criptográficas:

- Es imprescindible borrar la clave pública del Servidor SGP almacenada por el Programa FxGate para que el Programa FxGate y el Servidor SGP intercambien de nuevo sus claves públicas.

- Igualmente hay que realizar un reset criptográfico en todos los terminales FxLock desde su menú de configuración.

Por último debe configurarse la lista de terminales FxLock conocidos por el Servidor SGP:

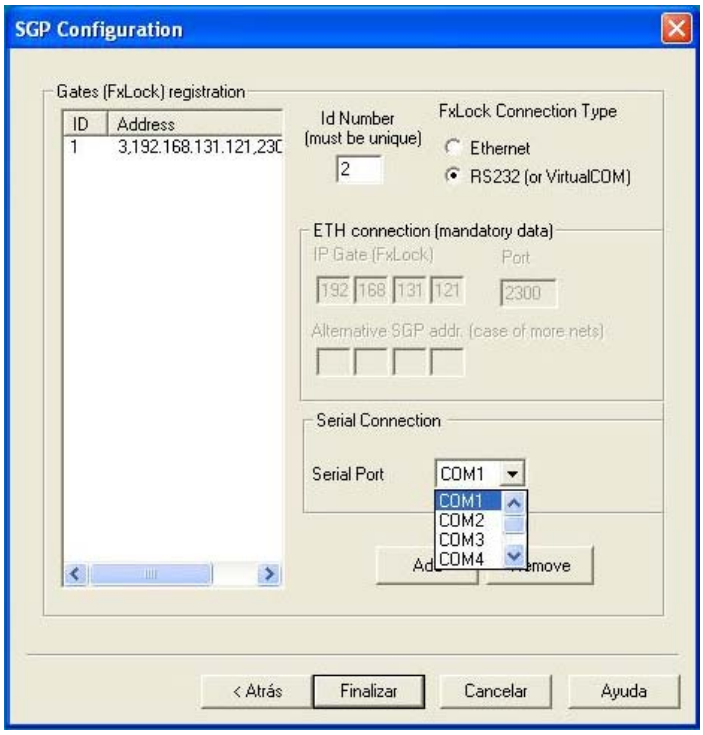

Fig. 2.5: Asistente de configuración del Servidor SGP. Configuración de terminales FxLock.

Para ello hay que introducir un identificador numérico único asociado a cada FxLock y su tipo de conexión con el Servidor SGP. Si la conexión se realiza a través de una conexión de red hay que indicar la dirección IP asociada a cada terminal FxLock. Si la conexión es mediante un puerto serie (RS232) hay que indicar cuál es el puerto COM utilizado para conectar el FxLock.

2.3.2. Funcionamiento del Servidor SGP en modo consola

A continuación se presentan dos ejemplos que ilustran brevemente el comportamiento del Servidor SGP, y en los cuales puede observarse que tipo de información se obtiene de la aplicación de consola.

El primer ejemplo se ilustra en la Figura 2.6, y la información mostrada por la aplicación de consola es la siguiente:

1. Al iniciar el Servidor SGP éste realiza una comprobación del fichero de claves y permanece a la espera de recibir conexiones procedentes del Programa FxGate o de un terminal FxLock. Para ello envía periódicamente por el puerto serie y/o la interfaz de red una señal 'IAmUp'.

2. El terminal FxLock con identificador 1 recibe la señal del servidor y se inicia la conexión. Como es la primera vez que servidor y FxLock dialogan ambos intercambian sus claves públicas. El servidor envía un reto al terminal FxLock y se negocia una clave de sesión que se utiliza para cifrar las comunicaciones. Una vez ha concluido satisfactoriamente la negociación el servidor solicita al terminal información sobre su capacidad de almacenamiento.

3. El Servidor SGP envía al terminal FxLock la lista de motivos, los perfiles de tiempo, los días festivos y la lista de usuarios activos en el FxLock. Esta información previamente ha sido configurada en el Servidor SGP a través del Programa FxGate.

4. El terminal FxLock ha recibido una solicitud de activación de un nuevo usuario, de manera que solicita al servidor SGP los datos correspondientes al usuario.

5. Periódicamente el terminal FxLock envía un mensaje al servidor indicando que se encuentra operativo.

6. Si se produce un acceso en el terminal FxLock éste envía de inmediato el suceso al Servidor SGP.

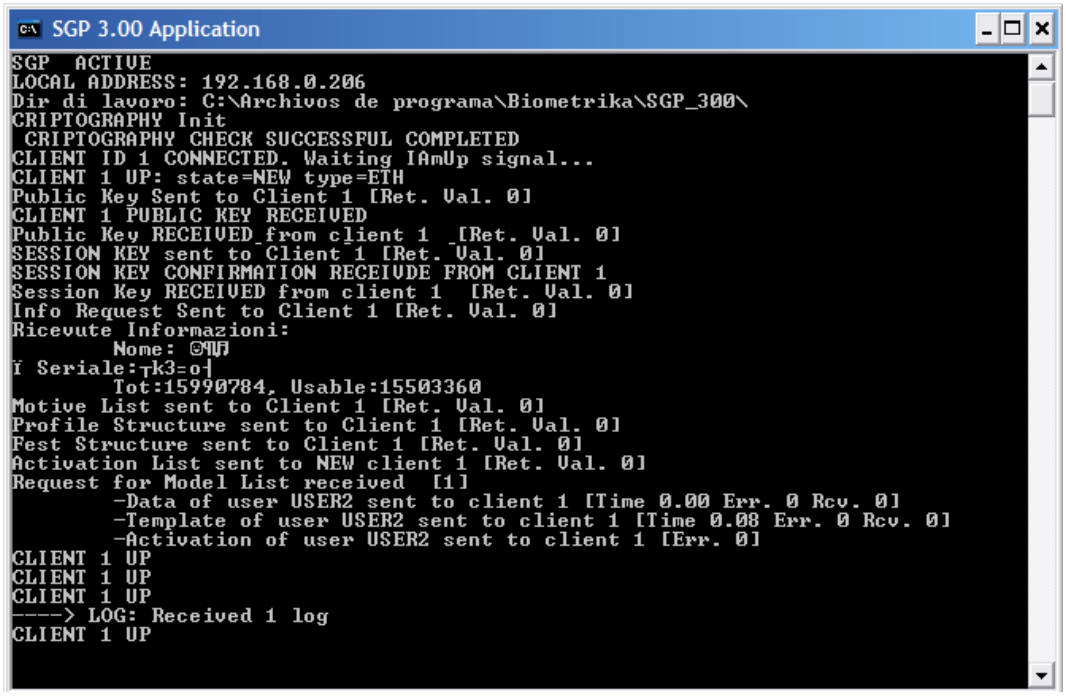

Fig. 2.6: Servidor SGP. Ejemplo de comunicación con FxLock.

El segundo ejemplo prueba la conexión del Programa FxGate con el Servidor SGP:

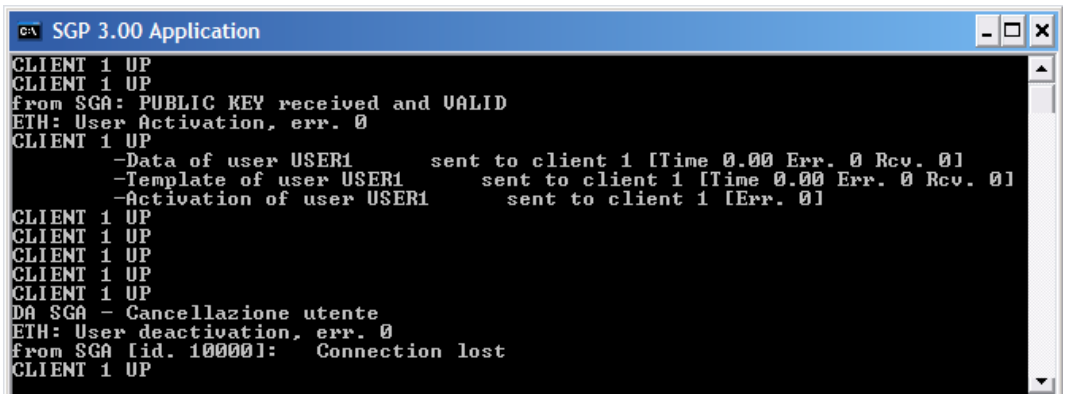

Fig. 2.7: Servidor SGP. Ejemplo de comunicación con Programa FxGate.

1. En primer lugar el programa FxGate se autentica frente al Servidor SGP y se negocia una clave de sesión.

2. Desde el Programa FxGate se inserta un usuario en el Servidor SGP (el Servidor SGP no notifica de esto) y posteriormente ese mismo usuario se activa en el terminal FxLock 1. El Servidor SGP envía al FxLock toda la información conocida del usuario activado.

3. El terminal FxLock periódicamente notifica al Servidor SGP de que se encuentra on-line.

4. Desde el Programa FxGate se desactiva del FxLock 1 al usuario anteriormente insertado.

5. El Programa FxGate se desconecta del Servidor SGP.

2.3.3. Archivos del Servidor SGP

El Servidor SGP se proporciona sin ningún tipo de documentación sobre su funcionamiento interno, su código fuente no es público, y el protocolo de comunicación entre el Servidor SGP y los terminales FxLock no es conocido. Es imprescindible la utilización del Servidor SGP para llevar a cabo una gestión remota y centralizada del Sistema FxGate, de manera que nos vemos obligados a utilizar el Servidor SGP programado por Biometrika. Por ello se ha analizado el comportamiento del Servidor SGP y a continuación se describe la utilidad de algunos de los ficheros utilizados por el servidor.

En el directorio de instalación del Servidor SGP se encuentran los siguientes archivos:

- config.txt. Es el fichero de configuración del Servidor SGP, y es generado por el asistente de configuración. A través de este asistente (explicado en el apartado 2.3.1) se especifican los parámetros de configuración del servidor y los terminales FxLock del Sistema FxGate.

- error.log. Se ha podido comprobar cómo en este fichero de error se recogen registros relacionados con errores en la inicialización del Servidor SGP, y errores provocados en el envío de la señal 'IAmUp'.

- skData.bin. Es el fichero de claves del Servidor SGP y contiene las claves públicas y privada utilizadas por el servidor. La única manera de generar este fichero es a través del asistente de configuración. Durante la configuración del servidor puede exportase una copia del fichero de claves generado, que por defecto se almacena como SGPKeyFile.crp.

- Server.exe y ConfigWizard.exe. Son los ejecutables del Servidor SGP en modo consola y del asistente de configuración.

El directorio Models del Servidor SGP contiene los patrones biométricos almacenados por el servidor. Los patrones de cada usuario se guardan en un fichero diferente, con extensión mdl y con un nombre de 20 caracteres alfanuméricos. Cada grupo de dos caracteres corresponde a la representación hexadecimal de cada carácter que compone el UID del usuario.

En el directorio Profiles se almacenan los ficheros Festivity.dat y Profiles.dat, que contienen la lista de días festivos y perfiles de tiempo insertados desde el Programa FxGate.

El directorio Srv contiene una serie de ficheros utilizados internamente por el Servidor SGP para la activación de usuarios, las listas de motivos de acceso y los siguientes archivos:

- Access.log. En este archivo se almacenan todos los registros de acceso de usuarios en el Sistema FxGate. Es el fichero que se envía al Programa FxGate cuando éste solicita al Servidor SGP la lista de eventos de acceso. Una vez recibido con éxito por el Programa FxGate se borra del Servidor SGP.

- CryptoSga.ini, CryptoEth.ini y CriptoComx.ini (con x=1,2,...). Estos ficheros contienen las claves públicas conocidas por el Servidor SGP. El primero de ellos corresponde a la clave pública del Programa FxGate. El segundo a todas las claves públicas de los terminales FxLock conectados por Ethernet. Y el último (pueden existir varios) a un FxLock conectado por puerto serie al Servidor SGP.

2.4. FxGate SDK Versión 3.10

2.4.1. Descripción del FxGate SDK

En primer lugar se presentan los archivos que componen el FxGate SDK, junto con una breve descripción de su utilidad. El SDK consiste básicamente en una serie de bibliotecas dinámicas que contienen todas las funciones que se pueden invocar desde el Programa FxGate. Estas bibliotecas deben estar situadas en el mismo directorio que el Programa FxGate o en su defecto en C:\Windows\System32.

Las bibliotecas que componen el FxGate SDK son las siguientes:

- SGADll.dll: contiene todas las funciones que pueden ser utilizadas dentro del Programa FxGate.

- CryptoApi.dll: contiene un conjunto de funciones criptográficas que implementan cifrado AES y RSA, y utilizadas internamente por las funciones de la biblioteca SGADll.dll.

- SCCard.dll, scardsyn.dll y RFCard.dll: incluyen todas las funciones necesarias para que las funciones contenidas en SGADll.dll puedan manejar las tarjetas inteligentes, (con contactos o RF), los lectores de huella, y los lectores de tarjetas, todos ellos de Biometrika.

Además de las bibliotecas anteriores se incluyen una serie de archivos que sirven de utilidad al programador de la aplicación:

- SGADll.h. Este fichero de encabezado contiene las declaraciones de las funciones incluidas en la biblioteca que son accesibles para el programador. Así como también una serie de constantes y estructuras de datos definidas para su utilización con las funciones. Proporciona al programador una interfaz de las funciones que puede utilizar para desarrollar el programa FxGate, indicando los parámetros y tipos de retorno.

- SGADll.lib. Se trata de una biblioteca de importación que referencia a SGADll.dll, y que permite enlazar estáticamente la biblioteca dinámica. Para su utilización es imprescindible configurar el compilador para añadir el archivo a la línea de vinculación y que dicha librería sea enlazada. (Para una información más detallada acerca de las bibliotecas de importación puede consultarse [17]).

- Error.h. En este archivo de encabezado se definen una serie de constantes numéricas que corresponden a los diferentes códigos de error que pueden producirse en las llamadas a funciones de la biblioteca SGADll.dll.

Por último se incluye el siguiente ejecutable:

- Keygen.exe. Es una aplicación de consola que debe estar localizada en el mismo directorio que las bibliotecas del SDK. Permite generar los ficheros de claves del Programa FxGate, formados por los ficheros skData.bin, skData1.bin y skData2.bin. En estos archivos se almacenan las claves pública y privada de la aplicación, utilizadas para el cifrado de todas las comunicaciones entre el Programa FxGate y el Servidor SGP, y para autenticar al Programa FxGate. Si los ficheros de claves ya existen el programa Keygen.exe genera nuevos ficheros de claves que sobre-escriben a los anteriores. En este caso se genera una nueva clave pública para el Programa FxGate, y puesto que el Servidor SGP almacena en un fichero la clave pública del Programa FxGate, es necesario borrar tal fichero para que Servidor SGP y Programa FxGate intercambien nuevas claves públicas y puedan autenticarse correctamente.

2.4.2. Descripción de las funciones contenidas en el FxGate SDK v3.10

A continuación se describen de manera funcional las diferentes funciones incluidas en el manual del programador del FxGate SDK. Para una información detallada de los parámetros de cada función se recomienda consultar [12]. Conjuntamente con las funciones se describen funcionalidades del Sistema FxGate estrechamente relacionadas con ellas, y necesarias para explicar su funcionamiento. También se incluyen comentarios relacionados con el comportamiento de algunas funciones, bien porque se ha observado un funcionamiento incorrecto, o bien porque la descripción recogida en el manual del programador es incompleta.

Las funciones documentadas en el manual pueden clasificarse en seis grupos diferentes:

- Funciones para establecer la conexión entre el Programa FxGate y el Servidor SGP.

- Funciones para capturar el patrón de huella dactilar.
- Funciones para la gestión de usuarios del sistema.
- Funciones para la gestión de tarjetas inteligentes.
- Funciones auxiliares.
- Funciones de administración y configuración del sistema.

Funciones para establecer la conexión entre el Programa FxGate y el Servidor SGP.

- FXGate_Init.

Se trata de la primera función que debe ser llamada por el Programa FxGate antes de poder usar cualquier otra función de las bibliotecas, ya que inicializa los recursos de las bibliotecas y establece una conexión con el Servidor SGP. El Servidor SGP envía periódicamente una señal 'IAmUp' a través de un determinado puerto UDP, (según como haya sido configurado en su instalación). De modo que en la llamada a esta función hay que indicar tal puerto para que el Programa FxGate y el Servidor SGP establezcan la conexión.

Previo al establecimiento de la conexión, Servidor SGP y Programa FxGate se autentican mutuamente, por lo que como argumento de esta función también es necesario indicar el directorio que contiene los ficheros de claves del Programa FxGate. Si es la primera vez que Servidor SGP y Programa FxGate se comunican ambos intercambian sus claves públicas. A continuación, o en el caso de que hayan mantenido alguna comunicación en el pasado y dispongan de las claves públicas del otro, las aplicaciones proceden a autenticarse. El Servidor SGP envía un reto al Programa FxGate, consistente en una cadena aleatoria cifrada con la clave pública del Programa FxGate. Si el Programa FxGate es quien dice ser conocerá la clave privada correspondiente y podrá descifrar el reto, cifrarlo con la clave pública del Servidor SGP, y enviarle la respuesta al servidor. Por su parte el servidor podrá descifrar con su clave privada la respuesta recibida y compararla con la cadena aleatoria enviada. En el caso de que sean iguales el Programa FxGate habrá probado su identidad ante el Servidor SGP y ambos establecerán la conexión. La cadena aleatoria enviada por el Servidor SGP no solamente es utilizada para probar la identidad del Programa FxGate, sino que se utilizará como clave privada para cifrar todos los mensaje intercambiados entre ambas aplicaciones.

Otro argumento de esta función es el identificador del Programa FxGate. Según el manual del FxGate SDK ([12]) sería posible comunicarse con el Servidor SGP desde un total de hasta tres Programas FxGate diferentes. Sin embargo se ha podido comprobar cómo el Servidor SGP utilizado no soporta varias conexiones simultáneamente, solamente acepta la conexión de un único Programa FxGate, y siempre el mismo. Si el Servidor SGP dispone de la clave pública de un Programa FxGate y recibe una solicitud de conexión de un programa FxGate diferente (con clave pública distinta) se produce un error de autenticación. En este caso, independientemente del identificador de Programa FxGate, y de que exista o no otro Programa FxGate conectado al Servidor SGP, el servidor rechaza la conexión.

- FXGate_FastInit .

Al igual que la función anterior esta función inicializa las bibliotecas y permite que el Programa FxGate se conecte al Servidor SGP. (Solamente una de estas dos funciones debe ser llamada al comienzo del Programa FxGate). La diferencia de esta función se encuentra en que debe indicarse cuál es la dirección IP del Servidor SGP. El Programa FxGate no espera a recibir la señal 'IAmUp' del SGP sino que se pone en contacto directamente con el servidor, estableciéndose la conexión de forma mucho más rápida que con la anterior función, con la cual el establecimiento de la conexión puede durar unos segundos.

Aunque en la documentación del FxGate SDK no se hace ninguna referencia se ha comprobado cómo esta función solamente puede utilizarse en el caso de que el Servidor SGP y el Programa FxGate dispongan de las correspondientes claves públicas del otro, que únicamente son intercambiadas cuando se utiliza la función FXGate_Init. Si se llama a esta función desde un Programa FxGate que nunca se ha conectado con el Servidor SGP, o bien la clave pública del programa FxGate es distinta a la almacenada por el Servidor SGP, se producirá un error en el Servidor SGP y la conexión no se establecerá.

- FXGate_End.

Cancela la conexión entre el Programa FxGate y el Servidor SGP y libera los recursos asociados a las bibliotecas del FxGate SDK.

Funciones para capturar el patrón de huella dactilar.

- FXGate_Enroll

Esta función permite capturar un nuevo patrón de huella o editar uno ya existente. Para ello es imprescindible que en el equipo donde se ejecute el Programa FxGate que llama a esta función esté conectado y correctamente configurado un lector de huella de Biometrika, bien un lector FX2000 o bien un lector FX3000.

Funciones para la gestión de usuarios del sistema.

- FXGate_InsertUserStruct.

Permite insertar un nuevo usuario en la base de datos de usuarios del Servidor SGP. Si el usuario a insertar ya existe en el Servidor SGP al llamar a esta función sus datos serán actualizados.

A ojos del programador del Programa FxGate un usuario del sistema se representa como una estructura de datos que básicamente contiene la siguiente información sobre el usuario:

- Identificación del usuario: Todo usuario del sistema se identifica mediante un identificador único consistente en una secuencia de como máximo 10 caracteres. Además de este identificador todo usuario tiene asociado un PIN numérico de hasta

cuatro dígitos. Este PIN debe ser introducido por el usuario cuando pretenda acceder al sistema. No se trata de un identificador único, ya que varios usuarios pueden tener el mismo PIN. El hecho de que existan muchos usuarios insertados en el sistema con el mismo PIN repercute en el tiempo necesario por el FxLock para realizar la identificación. Cuando se produce un acceso el FxLock compara el patrón biométrico obtenido de la huella del usuario con todos los patrones biométricos conocidos asociados a usuarios del mismo PIN. Si como resultado de la comparación se obtiene un nivel de semejanza que supera el umbral de parecido asociado al usuario, el usuario es identificado. Si todos los usuarios del Sistema FxGate son insertados con diferente PIN la identificación será más rápida, ya que el FxLock solamente efectuará una comparación.

- Modos de acceso del usuario al sistema: si el usuario puede acceder o no a través de su huella dactilar, contraseña o tarjeta.

- Permisos de acceso: Los permisos de acceso al sistema hacen referencia a las diferentes acciones que un usuario, tras ser identificado con éxito, puede ejecutar. Cada terminal FxLock puede ser configurado con un menú personalizado que permite la selección de uno de hasta 4 ítems diferentes. Cada uno de estos ítems puede asociarse a una determinada acción, por ejemplo, activar el contacto de una puerta para abrirla o cerrarla, o disparar una alarma.

También pueden asociarse una serie de permisos de acceso avanzados, como los siguientes:

- Usuario anti-asalto: Los terminales FxLock pueden configurarse para activar un relé denominado anti-asalto, conectado a una alarma silenciosa o a un sistema de seguridad. La utilidad de esta función es la siguiente: imaginemos un usuario insertado dos veces en el sistema, una como usuario normal del sistema, (con su patrón biométrico asociado el correspondiente al dedo índice de su mano derecha) y la otra insertado como usuario anti-asalto (en este caso con el dedo corazón de la mano derecha). Un usuario malintencionado podría forzar, en contra de su voluntad, al usuario anterior para que se identificara y así conseguir acceder al sistema. En este caso el usuario podría utilizar el dedo corazón de su mano derecha para identificarse. El proceso de identificación se realizaría con normalidad, sin que el maleante sospeche nada, pero se activaría una alarma silenciosa para notificar de la situación de pánico.

- Acceso en días no laborables: En un Sistema FxGate pueden definirse una serie de días en los cuales ningún usuario normal, ni siquiera un administrador, puede acceder al sistema. Solamente aquellos usuarios dotados de este permiso especial pueden ser autorizados por el sistema.

- Adquisición de huella remota. Con este permiso un usuario puede ser insertado en el sistema de manera provisional sin un patrón biométrico asociado. El usuario no podrá acceder al sistema hasta que se capture su huella dactilar. Para ello un administrador del sistema debe seleccionar la opción correspondiente en un terminal FxLock y

supervisa el proceso de captura. El proceso de adquisición remota es especialmente útil en aplicaciones en las que resulte complicado que el usuario a insertar en el sistema se acerque hasta el lugar en el que se ejecuta el Programa FxGate para capturar su huella. En este caso es posible insertar al usuario en el sistema desde el Programa FxGate y posteriormente realizar la adquisición del patrón biométrico directamente desde un terminal FxLock. Una vez adquirido el patrón de huella éste es enviado inmediatamente al Servidor SGP, que se encarga de reenviarlo a aquellos otros terminales FxLock en los cuales el usuario también haya sido activado.

- Perfil temporal de acceso. Los permisos de acceso de un usuario pueden restringirse a una franja horaria. Un perfil temporal de acceso especifica en que períodos del día a lo largo de la semana un usuario puede acceder al sistema.

- FXGate_ActivateLocalUser

Como su propio nombre indica sirve para activar un usuario en un determinado FxLock. Para utilizar esta función es necesario que el usuario a activar haya sido insertado previamente en la base de datos de usuarios del Servidor SGP. La activación es válida para un único Terminal FxLock, si se quiere activar a un mismo usuario en varios terminales FxLock debe llamarse a la función tantas veces como número de terminales. El proceso de activación permite definir si un usuario tiene permisos de administrador. Este permiso no se trata de una característica propia del usuario, sino que está relacionado con el tipo de activación. También es posible modificar los permisos de acceso concedidos a un usuario con esta función.

- FXGate_RemoveUser y FXGate_DeactivateLocalUser

A partir del identificador de un usuario permiten eliminar al usuario del sistema o desactivarlo de un determinado terminal FxLock, caso en el que es necesario especificar el identificador de FxLock.

- FXGate_RetrieveRemoteEnrolledUsers y FXGate_RetrieveUserModel

La primera de estas dos funciones permite consultar al Servidor SGP la lista de usuarios insertados con adquisición de huella remota, y cuyo patrón de huella ha sido capturado pero todavía no ha sido recibido por el Programa FxGate. La segunda función permite recibir desde el Servidor SGP el patrón de huella de un usuario.

- FXGate_RetrieveLocalList

Permite consultar al Servidor SGP los usuarios que han sido activados en un determinado terminal FxLock. Se obtiene una lista que contiene la siguiente información sobre cada usuario: su identificador de usuario, si es o no administrador del terminal, y cuáles son sus permisos de acceso.

Funciones para la gestión de tarjetas inteligentes.

- FXGate_C_CreateBioCard

Permite la inicialización de una tarjeta biométrica proporcionada por Biometrika (tarjeta inteligente con contactos o sin contactos) para su uso en el sistema. La función permite escribir en la tarjeta el identificador y el patrón biométrico del usuario.

- FXGate_C_LoadModelToMemory

Función auxiliar que permite cargar en memoria un patrón biométrico previamente capturado.

- FXGate_C_GetSerialNumber

Permite obtener el número de serie de una tarjeta RF. Para la lectura del número de serie es imprescindible que exista un lector de RF de Biometrika (FX2000RF) correctamente configurado en el equipo, y que la tarjeta inteligente esté dentro del radio de acción de la unidad lectora. No se ha podido probar el funcionamiento de esta función ya que no se ha contado con el equipamiento hardware necesario.

Funciones auxiliares.

Existen una serie de funciones auxiliares que permiten generar y comprobar los bytes utilizados para representar el tipo y permisos de acceso de un usuario en el sistema.

- FXGate_S_CreateAccessModeByte y FXGate_S_ParseAccessModeByte

Estas funciones permiten generar y comprobar un byte que representa el tipo de acceso de un usuario al sistema. Los diferentes modos de acceso posibles en un Sistema FxGate son los siguientes:

- Huella dactilar: si el usuario puede acceder a través de su huella dactilar. Debería ser siempre posible para que el sistema sea biométrico.

- Contraseña: solamente aquellos usuarios activados como administradores de un FxLock pueden acceder al sistema y al menú de configuración del FxLock mediante una contraseña.

- Tarjeta: un Sistema FxGate de Biometrika soporta cuatro tipos diferentes de tarjetas.

    - Tarjeta de acceso: en cuanto el terminal FxLock detecta este tipo de tarjeta el usuario es identificado. Se trata de un tipo de tarjeta poco segura, ya que puede ser sustraída y utilizada por cualquier otro usuario.

- Tarjeta Password: tras ser detectada por el sistema requiere que el usuario introduzca una contraseña para ser autorizado. Una tarjeta de este tipo podría ser robada y utilizada si se conoce la contraseña asociada.

- Tarjeta PIN: se trata de un tipo de tarjeta que tan pronto como es detectada por el FxLock se activa el lector de huella para que el usuario coloque su dedo en el sensor. El patrón biométrico obtenido a partir de la huella del usuario es comparado con el patrón almacenado en la memoria del FxLock para decidir si se permite el acceso al sistema.

- BioCard: se trata de una tarjeta inteligente que almacena el patrón biométrico del usuario. Cuando la tarjeta es detectada por el terminal FxLock, bien porque el usuario introduce la tarjeta con contactos en el lector del terminal, o bien porque el usuario acerca la tarjeta de RFID lo suficiente al lector RF del terminal, el FxLock invita al usuario a que posicione su dedo en el lector de huella. Se extrae el vector de características de la huella y se compara con el patrón contenido en la tarjeta. Si el resultado de la comparación supera un determinado umbral de semejanza y los permisos de acceso lo permiten se autoriza al usuario.

– FXGate_S_CreateGrant y FXGate_S_ParseGrant

Estas dos funciones permiten generar y comprobar respectivamente un byte que representa los permisos de acceso del usuario. Es decir, cuáles de los cuatro elementos que como máximo el display de un FxLock puede mostrar tras una identificación correcta pueden ser activados por el usuario.

Funciones de administración y configuración del sistema.

– FXGate_UpdateProfile

Permite añadir o actualizar un perfil de tiempo en el Servidor SGP. Un perfil de tiempo consiste en un identificador numérico y un conjunto de hasta diez reglas temporales. En cada regla temporal se definen una hora de inicio y otra de fin, que indican un intervalo de tiempo. También se especifican los días de la semana en los que el usuario puede acceder al sistema en la franja horaria definida por cada regla.

El Servidor SGP permite almacenar hasta 100 perfiles de tiempo diferentes. Dos de ellos, los correspondientes a los identificadores 0 y 1, son perfiles predefinidos, que respectivamente bloquean o permiten el acceso a los usuarios independiente de la hora y del día de la semana. El resto de perfiles temporales, con identificadores comprendidos entre 2 y 98, pueden ser personalizados desde el Programa FxGate, para posteriormente ser asignados a usuarios del sistema.

- FXGate_GetActiveClients

Permite consultar al Servidor SGP la lista de terminales FxLock que están operativos en el sistema y conectados al servidor.

- FXGate_RetrieveLogFile

Esta función permite obtener el fichero de registros de acceso almacenado en el Servidor SGP. El Servidor SGP recibe periódicamente información procedente de los terminales FxLock acerca de los accesos al sistema. Tan pronto como se produce un intento de acceso al sistema, el terminal FxLock envía el suceso al Servidor SGP. Si la conexión de red entre el FxLock y el Servidor no se encuentra operativa el terminal FxLock espera a que se restablezca la conexión, para enviar al Servidor SGP los eventos producidos, que son eliminados del FxLock.

- FXGate_UpdateMotive

El Sistema FxGate definido por Biometrika permite la implementación de aplicaciones de control de tiempo y asistencia. En cada terminal FxLock se puede establecer una lista de motivos de entrada/salida al sistema, consistentes en un par de instantes de tiempo que determinan el rango de validez del motivo, y una descripción, que se muestra en la pantalla del terminal FxLock.

Se ha podido comprobar cómo el período de validez de un motivo no está correctamente implementado en el Sistema FxGate de Biometrika. Independientemente de la hora del sistema si un terminal FxLock está configurado en modo de tiempo y asistencia cuando se autoriza el acceso a un usuario muestra en su pantalla todos los motivos almacenados, y no sólo aquellos cuyos períodos de validez incluyen a la hora actual, tal y como cabría esperarse.

- FXGate_UpdateFestivity

En un Sistema FxGate pueden definirse una serie de días denominados festivos o no laborables, en los cuales por defecto ningún usuario puede acceder al sistema. Únicamente aquellos usuarios dotados del permiso especial de acceso en días no laborables tendrán la posibilidad de acceder al sistema. Esta función permite establecer en el sistema los días considerados no laborables en el año actual y en el siguiente.

- FXGate_Syncronize

Permite establecer la hora de todo el Sistema SGP. Una vez establecida la hora del Servidor SGP éste se encarga de sincronizar la hora de todos los terminales FxLock.

- FXGate_GetKnownGates

Permite consultar al Servidor SGP la lista de terminales FxLock conocidos por el servidor. Esta lista incluye todos aquellos terminales que previamente han sido configurados en el Servidor SGP. Incluye tanto a los FxLock on-line como aquellos que no se encuentren on-line.

- FXGate_VoidCmd

La utilidad de esta función es verificar la conectividad y el estado del Servidor SGP, de manera que el Programa FxGate se asegure de que el Servidor SGP se encuentra operativo para recibir comandos. Sin embargo se ha comprobado cómo la implementación de esta función no es correcta, ya que independientemente de que el servidor se encuentre o no on-line siempre devuelve un valor 0.

2.4.3. Otras funciones contenidas en las bibliotecas del FxGate SDK

Las funciones explicadas en el apartado 2.4.2 constituyen todas las funciones documentadas en el manual del programador del FxGate SDK v3.10 ([12]). Además de estas funciones existen una serie de funciones cuyo nombre y parámetros se definen en el fichero de encabezado SGADll.h, pero no se contemplan en la documentación.

- FXGate_C_GetCardInfo y FXGate_C_Format. Se supone que estas funciones son para la gestión de tarjetas inteligentes. Estas dos funciones son utilizadas en uno de los ejemplos en Visual Basic proporcionados con el FxGate SDK v3.00, para su uso con tarjetas RFID. Se desconoce con qué tipo de tarjetas podrían utilizarse estas funciones.

- FXGate_SendFastCommand. Esta función también se utiliza en uno de los ejemplos en Visual Basic proporcionados con el FxGate SDK v3.00. Y su utilidad sería la de resetear un FxLock, cambiar el contraste de la pantalla o seleccionar una opción determinada del menú. Se ha podido comprobar cómo esta función no efectúa ninguna operación con los terminales FxLock utilizados.

- FXGate_UpdateList, FXGate_GetUnknownUser y FXGate_RetrieveList. Funciones para la gestión de usuarios y listas de usuarios.

- FXGate_RetrieveLogCopy, FXGate_Version, FXGate_IsGatePresent y FXGate_IsConnected. Funciones para consultar información de un terminal FxLock.

- FXGate_SendFile. Función para enviar un archivo a un FxLock. Se desconoce qué tipos de archivos pueden ser enviados a un FxLock, y cuál es su utilidad.

Por último se ha inspeccionado el archivo SGADll.dll con la utilidad link.exe de Visual Studio, con el objeto de obtener el nombre de todas las funciones contenidas en esta biblioteca. Puede afirmarse que el número de funciones contenidas en esta dll es muy superior al número de funciones recogidas en los puntos 2.4.2 y 2.4.3, existiendo un total de 70 funciones en esta biblioteca. Por el nombre de estas funciones adicionales no documentadas puede deducirse cual es su utilidad, que va desde la gestión de listas y grupos de usuarios, hasta el envío de comandos al Servidor SGP. Se desconoce si estas funciones son funciones internas utilizadas por otras funciones de las bibliotecas o incluso si estas funciones están correctamente implementadas, ya que Biometrika no ha querido dar ningún tipo de documentación adicional

de uso. Lo que sí parece probable es que su uso permitiría tener un mayor control sobre el Sistema FxGate.

2.5. BioCard SDK Versión 1.02

2.5.1. Descripción del BioCard SDK

Las tarjetas inteligentes son dispositivos extremadamente seguros, el intercambio de información se realiza de forma cifrada, y se llevan a cabo complejos procesos de autenticación. El BioCard SDK de Biometrika permite simplificar la realización de estas tareas rutinarias, necesarias para la manipulación de datos en las tarjetas inteligentes. Este SDK ofrece un conjunto de funciones básicas que permiten la gestión de las tarjetas inteligentes con contactos proporcionadas por Biometrika.

El funcionamiento y utilización de este SDK es totalmente análogo al del FxGate SDK utilizado para el desarrollo del Programa FxGate. El BioCard SDK se compone de las siguientes bibliotecas dinámicas:

- BioCard.dll: contiene todas las funciones necesarias para gestionar las tarjetas inteligentes de Biometrika.

- scardsyn.dll: se trata de la misma biblioteca incluida en el FxGate SDK. Proporciona las funciones necesarias para la utilización de los lectores y tarjetas inteligentes suministrados por Biometrika.

Además de las bibliotecas anteriores se incluyen los siguientes archivos para la utilización de las bibliotecas:

- BioCard.h: es un fichero de encabezado que contiene las declaraciones de las funciones para manipular tarjetas. Así como la definición de una serie de constantes y estructuras de datos para su uso con las funciones. Proporciona una interfaz de las funciones y parámetros para gestionar tarjetas.

- BioCard.lib: es una biblioteca de importación que referencia a BioCard.dll, y permite enlazarla estáticamente. Para el correcto uso del BioCard SDK debe configurarse el compilador para añadir este fichero a la línea de vinculación.

2.5.2. Descripción de las funciones contenidas en el BioCard SDK v1.02

Las funciones documentadas en el manual del BioCard SDK pueden clasificarse en tres bloques diferentes:

- Funciones de inicialización del lector de tarjetas / tarjetas.
- Funciones para la gestión de tarjetas.
- Funciones para la lectura / escritura de datos en las tarjetas.

Funciones de inicialización del lector de tarjetas / tarjetas.

- BioCard_OpenReader

Realiza la apertura del lector de tarjetas para la lectura o escritura de datos. Una llamada satisfactoria a esta función significa que se ha detectado un lector de tarjetas de Biometrika en el equipo donde se llama a esta función, está correctamente configurado y listo para ser usado.

- BioCard_DetectCard

Detecta si existe una tarjeta inteligente insertada en el lector de tarjetas y si se trata de una tarjeta proporcionada por Biometrika.

- BioCard_CloseReader

Finaliza la comunicación con la tarjeta y cierra el lector de tarjetas.

Funciones para la gestión de tarjetas.

- BioCard_SetPassword

El acceso a los datos en las tarjetas suministradas por Biometrika está protegido por tres contraseñas diferentes de 16 bytes cada una. Una contraseña de lectura, otra de escritura y otra de borrado de la tarjeta, que se establecen al inicializar la tarjeta.

Las tarjetas de Biometrika soportan dos modos diferentes de autenticación. Un modo personalizado, en el cual el usuario establece las contraseñas a utilizar durante la sesión por las funciones que acceden al sistema de archivos de la tarjeta, y un modo por defecto, en el que no se especifican explícitamente unas contraseñas, y se usan unas contraseñas por defecto. Es imprescindible que estas contraseñas coincidan con las contraseñas con las que se inicializó la tarjeta para que no se produzca un error de autenticación que inutilice la tarjeta.

Hay que tener en cuenta que tanto el FxGate SDK como los terminales FxLock solamente son compatibles con tarjetas que utilizan el modo de autenticación por defecto. Si utilizamos el BioCard SDK para leer o escribir información en una tarjeta de un usuario del Sistema FxGate estamos obligados a utilizar autenticación por defecto.

- BioCard_OpenFileSystem

Permite la apertura del sistema de archivos de la tarjeta. Antes de llamar a esta función debe establecerse el modo de autenticación de la tarjeta con la función SetPassword. Como resultado de una llamada exitosa a esta función se obtiene una serie de datos sobre el sistema de archivos de la tarjeta, entre los que se encuentran:

- El número de archivos contenidos en la tarjeta.
- Si la tarjeta contiene el template de un usuario.
- El espacio ocupado, disponible y total en la memoria de usuario de la tarjeta.

- BioCard_LightFormat y BioCard_Format

Permiten inicializar una tarjeta con unas contraseñas determinadas, lo que provoca el borrado de toda la información contenida en la memoria de usuario de la tarjeta. Para formatear una tarjeta en primer lugar se debe llamar a la función Format, indicando el modo de autenticación y contraseñas de acceso. Después establecer las contraseñas con la función SetPassword. Y por último se debe llamar a la función LightFormat, que borra todos los datos de la tarjeta.

- BioCard_GetLastAuthenticationStatus

Devuelve el número de intentos de autenticación fallidos que restan para el bloqueo de la tarjeta. Cada vez que se intenta acceder al sistema de archivos de la tarjeta con una contraseña incorrecta se disminuye en una unidad este valor.

Funciones para la lectura / escritura de datos en las tarjetas.

- BioCard_GetCardId y BioCard_SetCardID

Permiten obtener y establecer el identificador de la tarjeta (Card ID), consistente en una cadena de como máximo 10 caracteres de longitud. Como resultado de la llamada a esta función se obtiene también cual es el tipo de autenticación utilizado por la tarjeta insertada (contraseñas por defecto o personalizadas). Así antes de escribir un nuevo identificador en la tarjeta con la función SetCardID se debe establecer el modo de autenticación y las contraseñas correspondientes con la función SetPassword en el caso de que proceda.

- BioCard_GetFileInfo

Una vez correctamente abierto el sistema de archivos de la tarjeta permite consultar la existencia de un fichero en la memoria de usuario y su tamaño. Los ficheros en las tarjetas suministradas por Biometrika se referencian por un identificador numérico único. Además son almacenados en el mismo nivel jerárquico, ya que el sistema de archivos de las tarjetas de Biometrika no soporta la creación de directorios.

- BioCard_ReadFile y BioCard_WriteFile

Estas dos funciones permiten la lectura, escritura y borrado de archivos en la tarjeta. Antes de leer un fichero con un identificador de fichero determinado debe utilizarse la función GetFileInfo para comprobar si el fichero está presente en la tarjeta y obtener su tamaño, ya que para la lectura de un fichero es necesario especificar cuál es su tamaño y su identificador de fichero.

- BioCard_WriteModelFromMemory y BioCard_WriteModelFromFile

La función WriteModelFromMemory permite tras la apertura del sistema de archivos de la tarjeta escribir un patrón biométrico previamente cargado en memoria. La función BioCard_WriteModelFromFile permitiría realizar lo mismo pero especificando un archivo que contenga un template, sin embargo se ha comprobado cómo en las versiones del BioCard SDK utilizadas esta última función no puede utilizarse.

- BioCard_ReadModelToMemory y BioCard_ReadModelToFile

Permiten la lectura del patrón biométrico almacenado en la tarjeta inteligente, bien a un buffer de memoria previamente reservado por el programa que invoca la función ReadModelToMemory, o bien a un fichero, con la función ReadModelToFile. Al igual que sucede con el resto de funciones que acceden al sistema de archivos de la tarjeta hay que establecer en primer lugar el modo de autenticación y contraseñas de la tarjeta y abrir el sistema de archivos.

3. IMPLEMENTACIÓN DE UN SISTEMA FXGATE

3.1. Equipamiento disponible

Para el desarrollo de este Proyecto Fin de Carrera se ha contado con los siguientes componentes software y hardware de Biometrika:

- FxGate SDK, en sus versiones 3.00 y 3.10.
- BioCard SDK, en sus versiones 1.01 y 1.02.
- Un terminal FxLock con 32MB de memoria Flash y lector de tarjetas inteligentes con contactos.
- Un terminal FxLock con 32MB de memoria Flash y lector de tarjetas inteligentes RFID.
- Un lector de huella / tarjetas inteligentes con contactos Fx3000SC
- Tarjetas inteligentes con contactos suministradas por Biometrika, de tipo Watchdata TimeCOS, y 32kbits de memoria de usuario. ([15]).
- Tarjetas inteligentes sin contactos suministradas por Biometrika, de tipo Atmel AT88SC6416CRF y 64kbis de memoria de usuario. ([16]).

3.2. Análisis de las plataformas de programación

Previamente al desarrollo del Programa FxGate se analizaron varias alternativas posibles para el entorno y lenguaje de programación de la aplicación. De partida nos encontramos con una serie de inconvenientes derivados de la naturaleza de los SDKs utilizados:

- Ambos SDKs, tanto FxGate SDK como BioCard SDK, están formados por bibliotecas compiladas con Microsoft Visual Studio. Además incluyen un fichero de importación .lib para el vinculador del compilador, y ficheros de encabezado en lenguaje C.
- Las propias funciones de las bibliotecas están programadas en lenguaje C, y por tanto utilizan como parámetros tipos de datos en este lenguaje de programación.

Teniendo en cuenta las consideraciones anteriores el primer problema que nos encontramos es elegir un entorno de programación para desarrollar la aplicación.

Como primera opción se analizó la posibilidad de desarrollar el Programa FxGate bajo Borland C. Aunque Borland por defecto no es compatible con bibliotecas compiladas con Visual Studio existen una serie de utilidades de línea de comando (implib.exe e impdef.exe) que permiten convertir este tipo de bibliotecas para ser utilizadas con Borland. La aplicación

de estas utilidades sobre las bibliotecas del FxGate SDK no permitió obtener bibliotecas compatibles con Borland. De modo que se contactó con Biometrika para preguntar acerca de la compatibilidad de sus bibliotecas con Borland. Pero la única respuesta obtenida fue algo parecido a "Existe una utilidad en Borland para utilizar bibliotecas hechas con Microsoft Visual Studio", y eso ya se había intentado sin éxito.

Una vez comprobada la imposibilidad de usar Borland, y puesto que los SDKs de Biometrika incorporaban proyectos de ejemplo de Microsoft Visual Studio, se decidió utilizar Visual Studio como entorno de desarrollo para el Programa FxGate.

El siguiente paso fue decidir que lenguaje de programación utilizar en Visual Studio para implementar el Programa FxGate, para lo que en primer lugar se analizaron las aplicaciones demo incluidas en los SDKs. El FxGate SDK v3.00 incluye tres proyectos de ejemplo en Visual Basic 6, que posteriormente, ya que no funcionaban bien, fueron reemplazados en el FxGate SDK v3.10 por un único proyecto Visual C++ de Visual Studio 2008. Conjuntamente con las versiones BioCard SDK utilizadas se incluye un proyecto de ejemplo Visual C++, también de Visual Studio 2008.

Después de analizar los ejemplos proporcionados, y tras investigar otros posibles lenguajes de programación para el desarrollo de aplicaciones con estos SDKs, se han encontrado las siguientes posibilidades:

- Utilizar Visual Basic como lenguaje de programación. Puesto que las funciones contenidas en las bibliotecas son de C es necesario tener en cuenta una serie de consideraciones para su utilización con este lenguaje:

  - Declarar una referencia a las funciones contenidas en las bibliotecas con instrucciones del tipo Declare Function.

  - Declarar en lenguaje Visual Basic todas las constantes y estructuras de datos definidas en los ficheros de encabezado de los SDKs, para lo cual pueden utilizarse instrucciones de tipo Const y Type.

  - Utilizar con las funciones tipos de datos de Visual Basic equivalentes a los de C. En este caso existe una relación directa entre tipos. Por ejemplo int de C con Long de Visual Basic o char* de C con String de Visual Basic.

- Importar las bibliotecas de C en el código de una aplicación de Visual C#. Para lo cual hay que seguir un procedimiento equivalente al caso de Visual Basic, pero con la complejidad añadida de que los tipos de datos usados en C no tienen una correspondencia directa con los de C#. El procedimiento a seguir sería el siguiente:

  - Importar las funciones de las bibliotecas utilizando atributos DllImport, y utilizar tipos de datos compatibles con C en las declaraciones de las funciones.

- Definir las constantes y estructuras de datos contenidas en los ficheros de encabezado de los SDKs. En el caso de las estructuras de C hay que tener en cuenta que las estructuras en C# por defecto no permiten establecer un orden fijo en las variables que contienen, y para forzarlo deben utilizarse atributos del tipo MarshalAsAttribute y SetLayout(LayoutKind.Sequential) en su declaración. ([23]).

- Usar Visual C++ como lenguaje de programación, que se trata de la opción más natural. Puesto que con cada SDK se incluye un fichero de encabezado .h y una biblioteca de importación .lib basta con incluir el fichero de encabezado en nuestro programa, y añadir a la línea de vinculación el fichero .lib. De esta manera, y debido a que en C++ pueden utilizarse directamente los tipos de datos de C, no hay que realizar ninguna tarea adicional para utilizar las funciones contenidas en las bibliotecas.

- Por último también se ha analizado una solución intermedia a las dos anteriores, que utiliza los dos lenguajes de programación. Esta alternativa consiste en la creación de una biblioteca de clases en C++, en la que se definen un conjunto de métodos que utilizan las funciones de los SDKs, y que pueden ser llamados desde un proyecto C# aprovechando la equivalencia de tipos entre C++ y C#. Una vez creada la biblioteca de clases y añadida como referencia a un proyecto Visual C# todos los métodos y variables definidos en la biblioteca quedan encapsulados en una clase, y pueden ser utilizados en C# instanciando un objeto de la clase importada.

3.3. Arquitectura / Escenario de aplicación

Para la prueba de las funcionalidades del Sistema FxGate de Biometrika, y puesto que se ha dispuesto solamente de dos terminales FxLock, se ha optado por implementar un sencillo Sistema FxGate que corresponde al siguiente esquema:

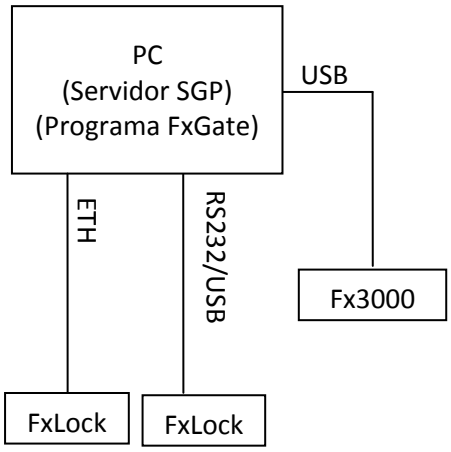

Fig. 3.1: Arquitectura del Sistema FxGate implementado.

En este sistema un único equipo actúa a la vez como Servidor SGP y como terminal desde el cual gestionar el sistema completo. En este PC se encuentran instaladas las aplicaciones Programa FxGate y Servidor SGP, y también dispone de un lector de huella / tarjetas inteligentes conectado a través de un puerto USB. Los dos terminales FxGate se encuentran

conectados al Servidor SGP a través de interfaces diferentes, uno a través del puerto Ethernet y otro a través del puerto serie. En el caso del terminal FxLock conectado por puerto serie se ha utilizado un conversor RS232/USB.

A continuación se propone un esquema de Sistema FxGate correspondiente a un escenario de aplicación más general que el presentado antes. La arquitectura de este sistema es la siguiente:

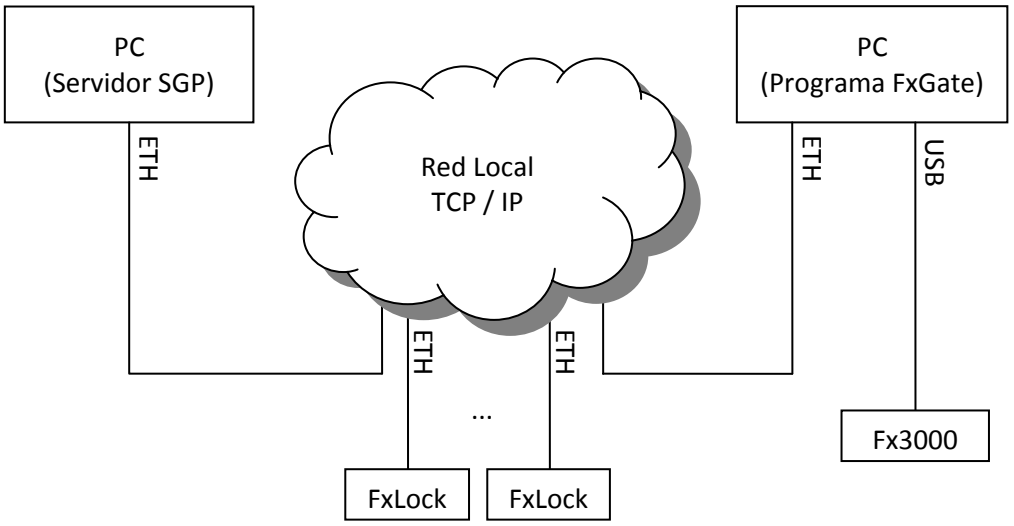

Fig. 3.2: Arquitectura general de un Sistema FxGate.

El Sistema FxGate se basa en una red local TCP/IP que interconecta a todos los elementos que componen el Sistema FxGate. Como se observa en la figura 3.2 el número de terminales FxLock puede ser cualquiera, siempre y cuando estos terminales sean visibles por el Servidor SGP.

En este caso el Programa FxGate se encuentra instalado en un equipo diferente al del Servidor SGP, de manera que podría situarse en un área restringida y altamente segura del entorno en el cual se instale el sistema de control de acceso.

3.4. Programa FxGate desarrollado

Se han programado dos aplicaciones software diferentes para el sistema de control de acceso implementado. Una de ellas corresponde al Programa FxGate, que gestiona todo el Sistema FxGate e integra los dos SDKs de Biometrika. Y la otra aplicación se trata de una interfaz gráfica para la aplicación de consola del Servidor SGP.

El Programa FxGate creado se ha desarrollado sobre la plataforma .NET de Microsoft. Se trata de una aplicación de Windows Form programada en Visual C++ con Microsoft Visual Studio 2008, y con marco de trabajo .NET Framework 3.5. Para la ejecución de esta aplicación es necesario un entorno Windows en el que se haya instalado previamente .NET Framework 3.5 o una versión superior.

Se ha escogido Visual C++ como lenguaje de programación porque como se comentó en el punto 3.2 este lenguaje permite manipular directamente las estructuras y tipos de datos de C empleadas por las funciones de los SDKs, y además no hay que hacer ningún tipo de definición adicional en el código para poder utilizar las funciones de las bibliotecas.

A continuación se presenta el funcionamiento del Programa FxGate, describiendo las diferentes tareas que permite realizar, y haciendo hincapié en las funcionalidades adicionales que se han implementado. Se indican además entre paréntesis las funciones de los SDKs utilizadas en cada caso.

La ventana principal del Programa FxGate se muestra en la Figura 3.3.

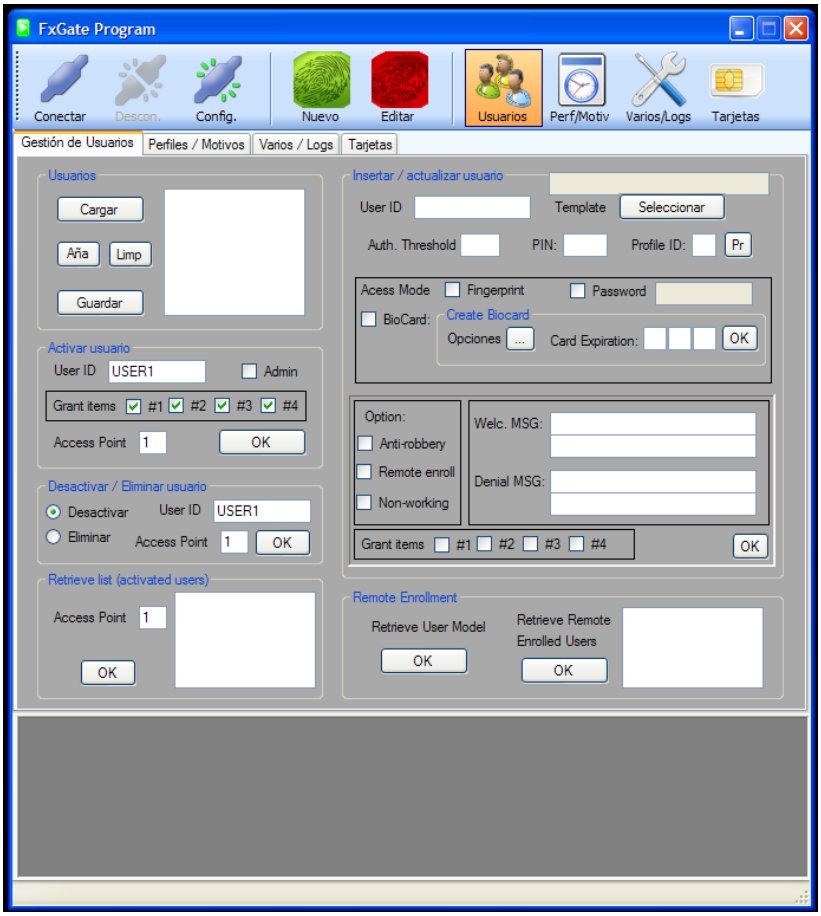

Fig. 3.3: Programa FxGate. Ventana principal.

La interfaz gráfica del programa se compone de una barra superior con botones, una consola situada en la parte inferior de la ventana en la que se muestra información sobre la ejecución del programa (las funciones llamadas y los valores devueltos), y un área principal organizada en cuatro vistas diferentes que agrupan todas las tareas que se pueden realizar.

Los tres primeros botones de la barra superior permiten respectivamente:

- Conectar el Programa FxGate con el Servidor SGP (FXGate_Init, FXGate_FastInit). Se trata de la primera tarea a realizar, ya que la gestión del Sistema FxGate se realiza on-

line a través del Servidor SGP, y la misma función que conecta ambas aplicaciones inicializa los recursos de las bibliotecas del FxGate SDK.

- Desconectar el Programa FxGate del Servidor SGP (FXGate_End).

- Configurar los parámetros de la conexión. Al hacer click sobre el botón se muestra la ventana de la figura 3.4. Esta ventana permite establecer los puertos UDP en los que escucha el Programa FxGate (parámetros que deben coincidir con los configurados en el Servidor SGP), el identificador del Programa FxGate y el directorio que contiene los ficheros de claves de la aplicación. También permite activar la conexión rápida con el Servidor SGP

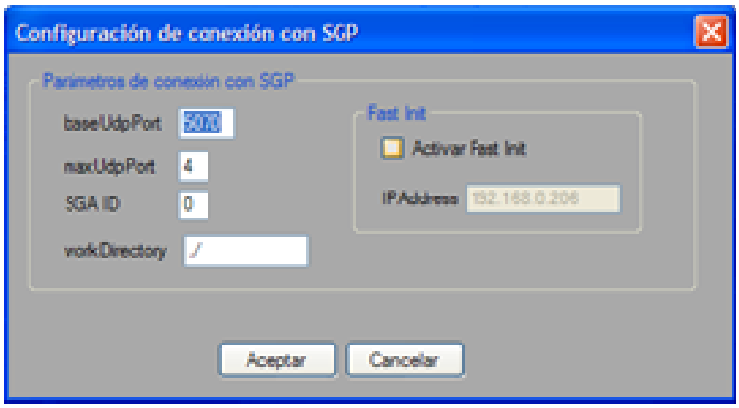

Fig. 3.4: Programa FxGate. Configuración de conexión con Servidor SGP.

A continuación existen 2 botones que permiten capturar un nuevo patrón de huella o editar uno ya existente. Para ello debe estar conectado al equipo un lector de huella dactilar proporcionado por Biometrika.

Por último en la parte derecha de la barra superior se tienen cuatro botones que permiten cambiar la vista principal de la aplicación. Se han implementado cuatro vistas diferentes en el Programa FxGate:

- Gestión de usuarios.

- Perfiles de tiempo y listas de motivos.

- Administración del sistema.

- Gestión de tarjetas inteligentes con contactos.

En cada una de las cuatro vistas del Programa FxGate se han utilizado controles de tipo groupBox, que permiten agrupar en un cuadro con título diferentes controles que tienen una funcionalidad común. De esta manera para explicar el funcionamiento del Programa FxGate se hace referencia a los títulos de cada grupo, que además ofrecen una breve descripción de las tareas que pueden realizarse desde cada grupo.

3.4.1. **Gestión de usuarios**

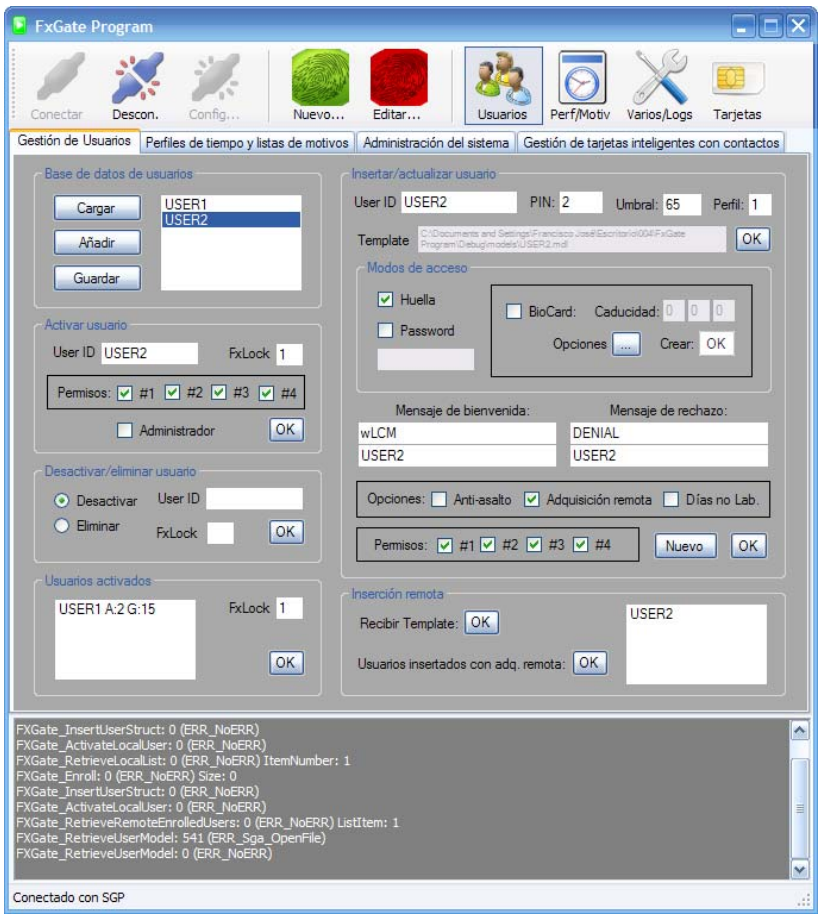

Fig. 3.5: Programa FxGate. Vista de gestión de usuarios.

En la vista de gestión de usuarios se han agrupado todas las funcionalidades ofrecidas por el FxGate SDK relacionadas con los usuarios del Sistema FxGate.

- Base de datos de usuarios. Las bibliotecas que componen el FxGate SDK no incluyen ninguna función que permita llevar la cuenta de los usuarios insertados en el sistema, ni tampoco recuperar la información completa de los usuarios insertados en el Servidor SGP. De este modo es responsabilidad del programador almacenar de alguna manera un registro de los usuarios del sistema y toda la información que tengan asociada. Para la implementación de esta tarea se ha dotado al Programa FxGate de una sencilla funcionalidad que permite guardar en un fichero llamado usuarios.txt los datos de los usuarios del sistema, añadir un nuevo usuario en esta base de datos local, o cargar en el grupo Insertar/actualizar usuario la información de un determinado usuario. Al hacer click sobre el botón cargar se muestra una lista con los identificadores de todos los usuarios almacenados en el fichero, y haciendo click sobre un identificador determinado se rellenan los campos del grupo Insertar/actualizar usuario.

- Insertar/actualizar usuario (FXGate_InsertUserStruct). Como su propio nombre indica este grupo permite insertar un nuevo usuario en el Servidor SGP, o actualizar sus datos en el caso de que ya exista. Toda la información introducida en este grupo se utiliza para generar una estructura de datos que representa al usuario y utilizada para insertar sus datos en el Servidor SGP.

- Activar usuario (FXGate_ActivateLocalUser). Permite activar en un terminal FxLock al usuario especificado, indicar sus permisos de acceso y si tiene o no permisos de administrador del terminal.

- Desactivar / eliminar usuario: (FXGate_DeactivateLocalUser, FXGate_RemoveUser). Para eliminar del Sistema FxGate un usuario, o desactivarlo de un terminal FxLock determinado.

- Usuarios activados (FXGate_RetrieveLocalList). Consulta al Servidor SGP los usuarios que han sido activados en un FxLock. Se muestra una lista donde cada entrada indica el identificador de usuario, si es o no administrador y sus permisos de acceso.

- Inserción remota (FXGate_RetrieveRemoteEnrolledUsers, FXGate_RetrieveUserModel). Este grupo permite obtener una lista de identificadores de usuarios correspondientes a usuarios activados con permisos de inserción remota, y cuyo patrón biométrico ha sido capturado pero todavía no ha sido recibido por el Programa FxGate. También es posible recibir el patrón biométrico de un usuario, para lo cual hay que indicar en el grupo Insertar/actualizar usuario el identificador de usuario y el archivo en el que será almacenado.

3.4.2. **Perfiles de tiempo y listas de motivos**

La vista de perfiles de tiempo y lista de motivos se muestra en la Figura 3.6 y permite gestionar los perfiles de tiempo del Sistema FxGate y la lista de motivos de cada uno de los terminales FxLock.

- Perfiles de Tiempo (FXGate_UpdateProfile). El FxGate SDK no dispone de una función para consultar al Servidor SGP la lista de perfiles de tiempo, por lo que se ha implementado la funcionalidad de guardar los perfiles en un fichero profiles.txt. Al pulsar el botón cargar se muestra una lista con los nombres de todos los perfiles almacenados. Existen dos perfiles predefinidos y no editables, correspondientes a NEVER (identificador de perfil 0) y ALWAYS (identificador 1). El resto de perfiles, así como nuevos perfiles creados, pueden ser seleccionados en la lista de perfiles. Al seleccionar un perfil existente se muestran su identificador y su nombre, y se actualiza la tabla de slots de tiempo. En esta tabla se indican las franjas horarias y días de la semana de los slots que componen un perfil de tiempo, y pueden añadirse o eliminarse slots temporales con los botones + y -. El perfil seleccionado puede enviarse al Servidor SGP haciendo click sobre el botón Insertar en SGP.

- Lista de motivos (FXGate_UpdateMotive). Para la gestión de las listas de motivos de acceso de los Terminales FxLock se ha desarrollado una funcionalidad parecida a la implementada para los datos de usuarios y los perfiles de tiempo. La diferencia es que al pulsar el botón cargar o guardar se debe especificar el nombre del fichero de motivos en el cuadro de diálogo mostrado, ya que cada terminal FxLock puede tener asociado una lista de motivos de acceso diferente a la de los demás terminales. La edición de la lista de motivos se realiza de manera similar a la de los perfiles de tiempo. El botón enviar permite enviar la lista de motivos al Servidor SGP para que la reenvíe al Terminal FxLock que corresponda.

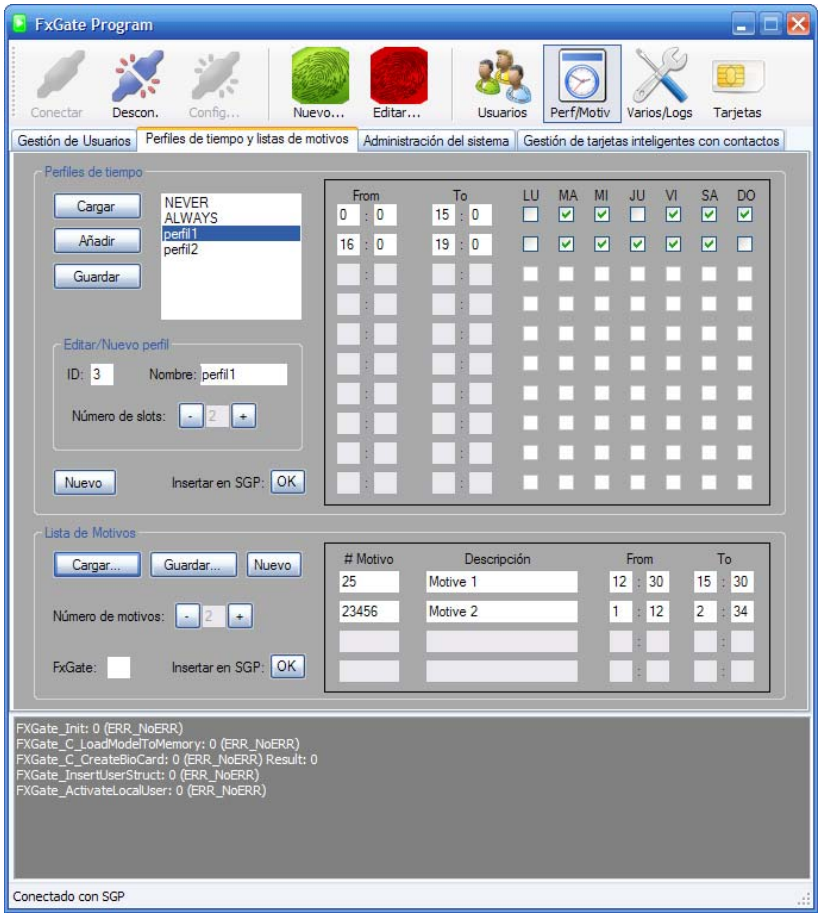

Fig. 3.6: Programa FxGate. Vista de perfiles de tiempo y lista de motivos.

3.4.3. **Administración del sistema**

La vista de administración permite realizar varias tareas de gestión del Sistema FxGate, así como recibir y visualizar los registros de acceso. Se muestra en la Figura 3.7.

- Terminales FxLock (FXGate_GetActiveClients, FXGate_GetKnownGates). Permite consultar al Servidor SGP la lista de terminales FxLock on-line y los terminales configurados en el servidor. La información recibida se muestra en la consola del Programa FxGate.

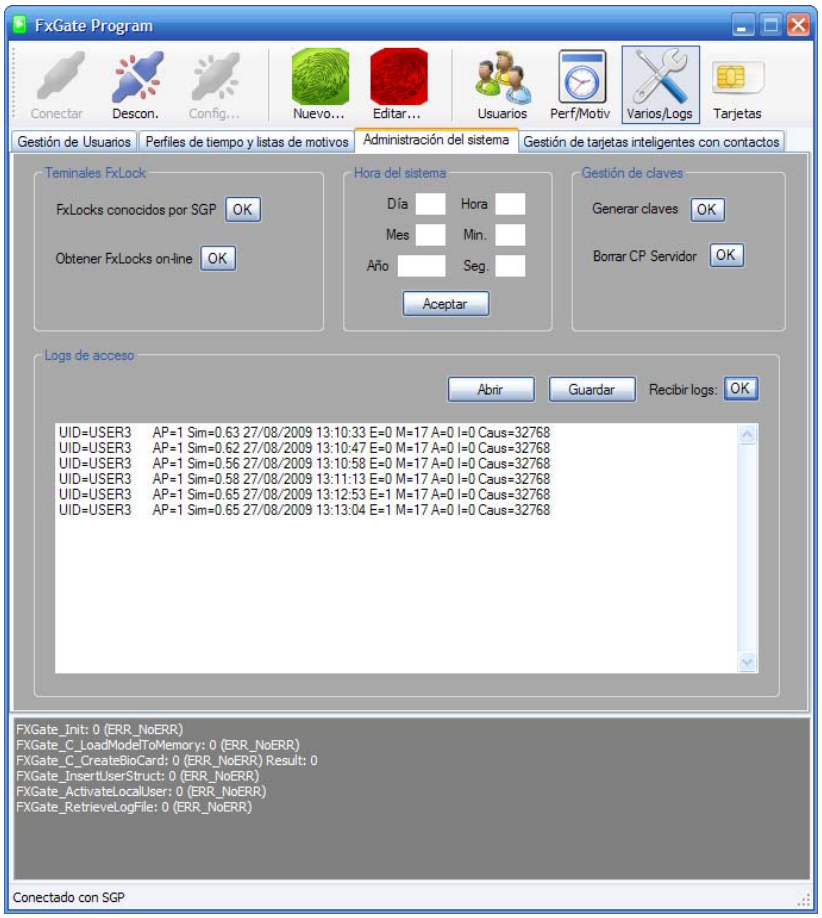

Fig. 3.7: Programa FxGate. Vista de administración del sistema.

- Hora del sistema (FXGate_Syncronize). Permite establecer la fecha y hora del Sistema FxGate.

- Gestión de claves. En este grupo se encuentran dos funcionalidades relacionadas con las claves del Programa FxGate, que solamente pueden ejecutarse cuando la aplicación no se encuentre conectada al Servidor SGP. Consisten en la creación de un nuevo fichero de claves para el Programa FxGate, y en el borrado de la clave pública del Servidor SGP almacenada por la aplicación. En el caso de cambiar las claves de alguno de los componentes del Sistema FxGate hay que tener en cuenta las consideraciones criptográficas explicadas en el capítulo 2.

- Logs de acceso (FXGate_RetrieveLogFile). Este grupo permite solicitar al Servidor SGP la lista de logs de acceso del Sistema FxGate, y también visualizar y exportar los eventos de acceso. Una vez cargado o recibido un fichero de accesos se muestra una lista con todos los eventos, y al hacer click sobre uno cualquiera aparece una ventana como la de la figura 3.8. En ella se muestra información detallada sobre el acceso, como el identificador de usuario, la fecha y hora, el tipo de acceso, el modo de acceso o el motivo de acceso.

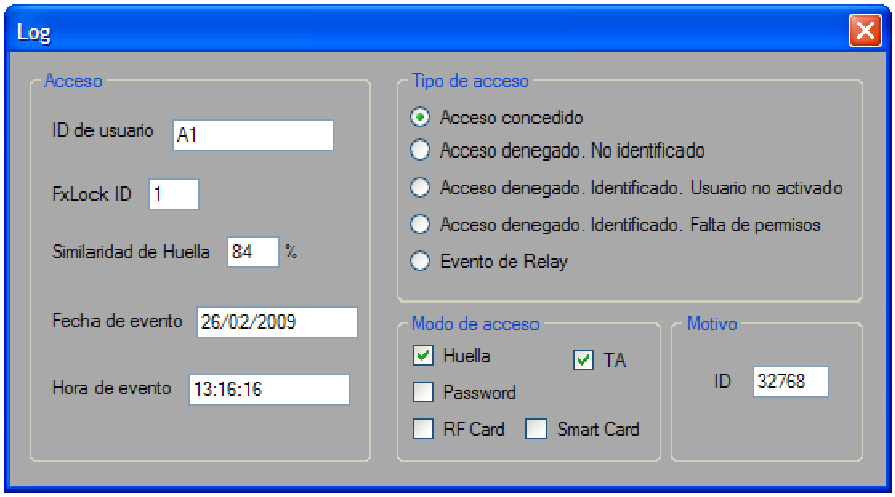

Fig. 3.8: Programa FxGate. Detalle de un evento de acceso.

3.4.4. **Gestión de tarjetas inteligentes con contactos**

Fig. 3.9: Programa FxGate. Vista de gestión de tarjetas inteligentes.

En esta última vista del Programa FxGate se han incluido todas las funcionalidades relacionadas con la gestión de tarjetas inteligentes con contactos, y que prueban el funcionamiento del BioCard SDK.

A pesar de que el acceso a los datos en las tarjetas suministradas por Biometrika puede protegerse con claves personalizadas no se ha implementado tal funcionalidad en el programa, ya que tanto el FxGate SDK como los terminales FxLock usan claves por defecto en su comunicación con las tarjetas inteligentes.

Se distinguen 3 grupos en la vista de gestión de tarjetas:

- Conexión. Permite inicializar el lector de tarjetas y establecer la comunicación con una tarjeta insertada en el lector, desconectar el lector, y abrir el sistema de archivos de la tarjeta. En este último caso se visualizan los parámetros del sistema de archivos de la tarjeta insertada.

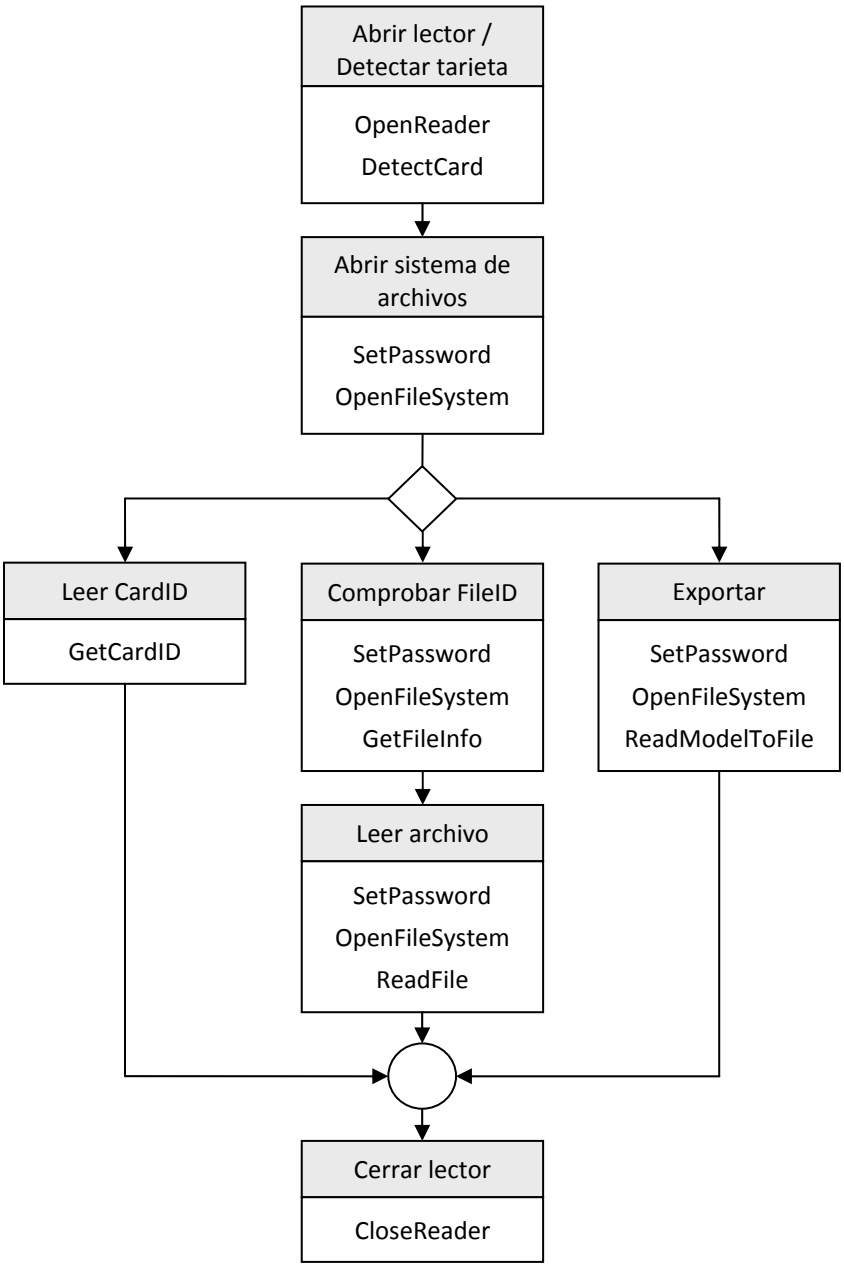

Fig. 3.10: Operaciones y funciones para la lectura de datos en las tarjetas.

- Lectura de datos. Desde este grupo pueden realizarse todas las tareas relacionadas con la lectura de datos de la tarjeta, como leer el identificador de la tarjeta, comprobar si existe un fichero con un determinado identificador en la tarjeta, y en tal caso leerlo, o exportar a un archivo el patrón biométrico almacenado en la tarjeta. En la Figura 3.10 se detalla un diagrama que indica cuales son las operaciones a seguir para efectuar una operación de lectura de datos en la tarjeta, así como las funciones del BioCard SDK utilizadas por el programa en cada acción. Una vez abierto correctamente el sistema de archivos de la tarjeta puede efectuarse una operación de lectura de datos, y al completar la lectura puede hacerse una nueva operación de lectura o escritura, o finalizar la comunicación con la tarjeta.

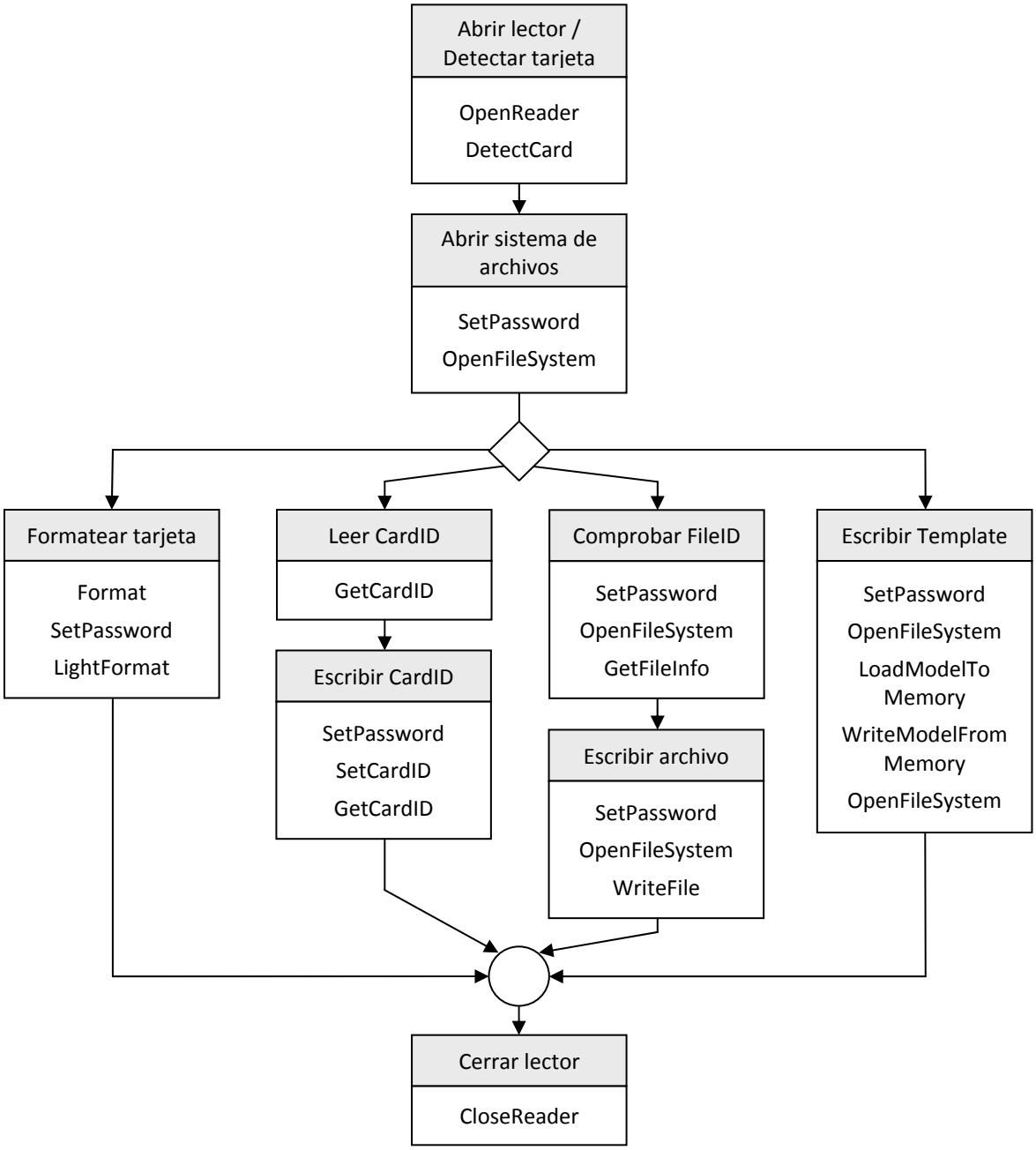

Fig. 3.11: Operaciones y funciones para la escritura de datos en las tarjetas.

- Escritura de datos. Permite formatear la tarjeta con claves por defecto, y la escritura del identificador de la tarjeta, ficheros (cadenas de caracteres) y el patrón biométrico del usuario. De manera similar a la lectura de datos se presenta en la Figura 3.11 un diagrama con las operaciones que hay que seguir para escribir datos en las tarjetas, así como las funciones utilizadas por el programa. Tras efectuar una operación de escritura puede abrirse de nuevo el sistema de archivos, para visualizar el espacio libre en la tarjeta tras la escritura, realizar una nueva escritura o lectura de datos, o finalizar la comunicación con la tarjeta. Para borrar un fichero existente hay que escribir un archivo vacío, seleccionando el identificador de fichero del fichero a borrar.

3.5. Servidor SGP desarrollado

Como se explicó en el capítulo 2.3 el Servidor SGP proporcionado por Biometrika puede ejecutarse como aplicación de consola, mostrando en tiempo real registros relacionados con los sucesos ocurridos en el Sistema FxGate. La información mostrada por esta consola se trata de la única información extraíble del Servidor SGP, a parte de los registros de error y accesos almacenados. De esta manera puede resultar de utilidad la posibilidad de procesar esta información a medida que se va produciendo.

También se ha podido comprobar cómo el funcionamiento del Sistema FxGate no es del todo satisfactorio, y además no se tiene un control absoluto sobre el sistema (se detallan todos los problemas encontrados en el capítulo 4). A pesar de todo esto se ha trabajado para buscar una solución que permita incorporar una interfaz gráfica al Servidor SGP. Para ello se ha desarrollado un programa que permite gestionar el Servidor SGP y procesar los datos de la aplicación de consola.

La tarea de redirigir la salida estándar de una aplicación de consola, como el Servidor SGP, puede parecer a priori sencilla, pero basta con tener en cuenta las siguientes consideraciones para darse cuenta de su complejidad:

- Se pretende procesar en tiempo real los mensajes mostrados en la consola del SGP. Luego por tanto no es una solución válida redirigir la salida estándar a una tubería y leer datos de ésta cada cierto tiempo, cuando se llene un buffer o cuando la aplicación termine su ejecución.

- La solución a encontrar tiene que ser una aplicación multi-hilo. Si se pretende contar con una interfaz gráfica de usuario, ejecutar la aplicación SGP Server y procesar al mismo tiempo su salida estándar se necesitan al menos tres hilos funcionando simultáneamente.

- Se ha escogido como entorno de desarrollo para la aplicación gráfica Microsoft Visual Studio 2008, y como marco de trabajo .NET Framework 3.5. Hay que tener en cuenta que desde la versión 3.0 de .NET Framework por defecto los atributos de un objeto solo pueden ser modificados desde el mismo proceso que lo creó. Si se pretende desarrollar una aplicación multi-hilo en la que un hilo lea la salida estándar del

Servidor SGP, y después modifique algún control del proceso principal (por ejemplo un cuadro de texto con la información leída) la cosa se complica. Y conviene utilizar métodos que garanticen una ejecución segura y sin cruces de hilos. ([19]).

- El comportamiento de una aplicación de consola al redirigir su salida estándar a una tubería puede no ser el deseado. Si en el código fuente de la aplicación de consola no se fuerza la impresión justo después de cada orden que imprima valores por pantalla (por ejemplo en C fflush tras printf), puede suceder que los mensajes se muestren bien por pantalla, pero al redirigir la salida a una tubería no se envíen los datos hasta que se llene un buffer o finalice la ejecución de la aplicación de consola. (Este comportamiento concreto es el caso del Servidor SGP. Más adelante se explica cómo se ha solventado).

Para resolver los tres primeros puntos se ha utilizado una solución implementada en Visual C# que simplifica enormemente la realización de estas tareas. Esta solución puede encontrarse en [21], y se encuentra basada en el artículo de Microsoft [19]. Consiste en un conjunto de clases entre las que se encuentra una clase llamada ProcessCaller, que hereda de otra clase llamada AsyncOperation, y que para su utilización basta con instanciar y establecer los parámetros que determinan la aplicación de consola a ejecutar con salida estándar redirigida.

La clase AsyncOperation facilita la creación de hilos trabajadores capaces de desencadenar un evento en el hilo principal de ejecución (el correspondiente a la interfaz gráfica de usuario). Existen dos hilos que de forma continuada, y mientras la aplicación de consola se ejecute, leen de la salida estándar y de la consola de error. Cada vez que leen una línea completa producen un evento en el hilo principal que provoca la invocación de un método delegado. Dentro de este método se tiene acceso a la línea leída, que puede procesarse para ser añadida a un cuadro de texto de la aplicación, o para realizar alguna acción en el programa.

La particularidad de los delegados en .NET es que representan funciones que pueden ser invocadas de manera asíncrona desde cualquier proceso. La utilidad en este caso es la siguiente, cada vez que el hilo que lee de la salida estándar obtiene una línea, encola una solicitud para que el hilo principal la procese dentro de una función.

El último escollo para redirigir correctamente la salida estándar del Servidor SGP se ha solventado con la ayuda de una aplicación de consola en tiempo real (RTConsole.exe [22]). Esta utilidad de consola se puede emplear para que aplicaciones de consola desarrolladas con las bibliotecas de ejecución de C de Microsoft (CRT), fuercen la escritura inmediata de datos en la salida estándar. Para ello realiza una serie de operaciones, y monitoriza la salida de la aplicación de consola que se le pasa como parámetro. Su utilización en el programa desarrollado es simple, la aplicación de consola cuya salida estándar se redirige es RTConsole.exe, y se indica como argumente el ejecutable server.exe correspondiente al Servidor SGP.

El programa SGP Server implementado se trata de una aplicación Windows Form desarrollada al igual que el programa FxGate sobre la plataforma .NET. El lenguaje de programación empleado ha sido Visual C#, ya que las clases utilizadas para redirigir la salida estándar están escritas en este lenguaje.

La estructura y funcionamiento del programa son similares a las del Programa FxGate:

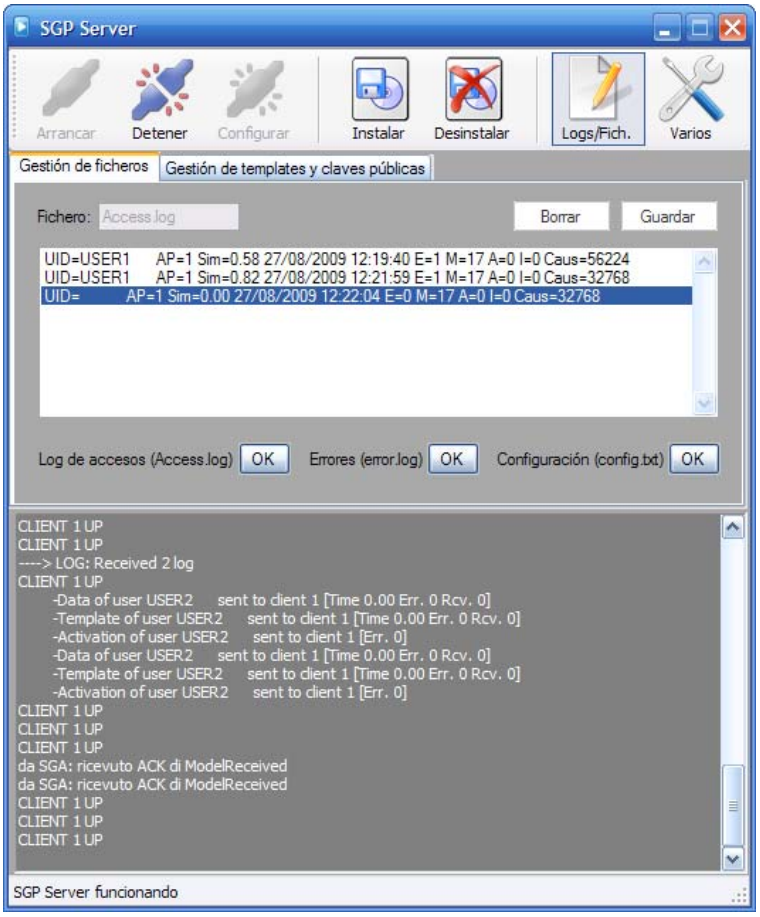

Fig. 3.12: Interfaz gráfica del Servidor SGP. Vista de gestión de ficheros.

En la parte superior del programa existe una banda de botones que permiten respectivamente arrancar y detener el Servidor SGP, lanzar el asistente de configuración del servidor (solamente cuando el servidor se encuentra detenido), instalar y desinstalar el Servidor SGP suministrado por Biometrika, y cambiar entre las dos vista del programa.

En la parte central se han implementado dos vistas diferentes:

- Una vista de gestión de ficheros. Que permite visualizar y editar el fichero de configuración del servidor (config.txt), el fichero de errores en el servidor (error.log) y también permite visualizar el registro de accesos almacenado por el Servidor SGP (Access.log). Al hacer click sobre un evento de la lista de accesos se muestra una ventana idéntica a la mostrada por el Programa FxGate (Figura 3.8).

– Una vista de gestión de templates y claves públicas. En la cual se han implementado dos funcionalidades:

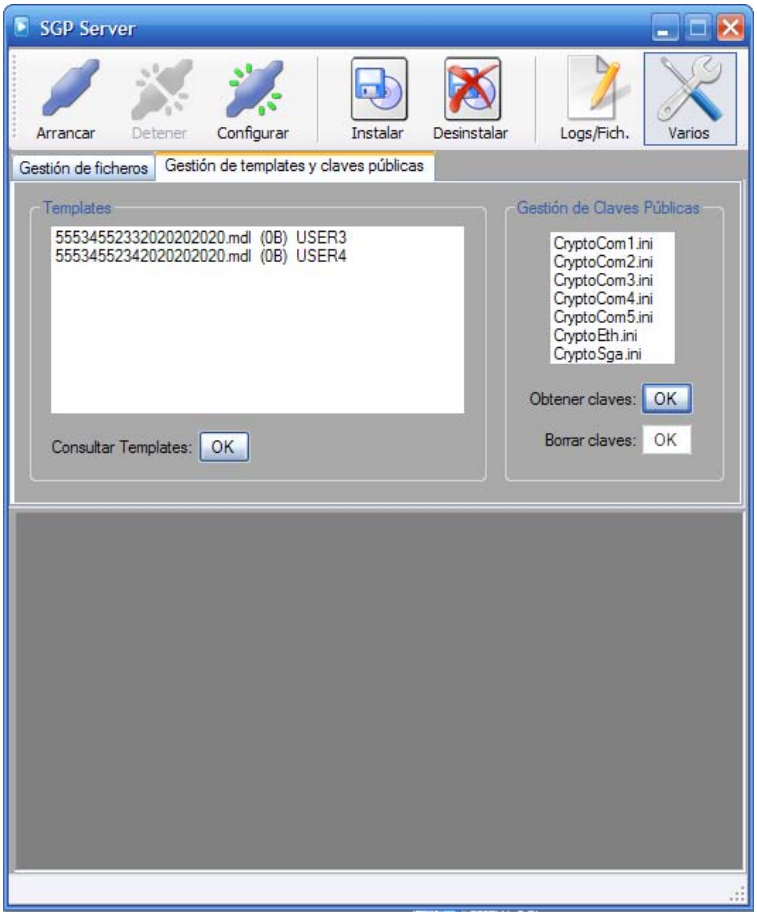

Fig. 3.13: Interfaz gráfica del Servidor SGP. Vista de gestión de templates y claves públicas.

- Consulta de templates: permite obtener una lista con los patrones de huella almacenados en el servidor, así como el identificador de usuario al que corresponden.

- Gestión de claves públicas: muestra una lista con las diferentes claves públicas almacenadas en el servidor, y permite su borrado siempre y cuando el servidor no se encuentre corriendo.

Por último en la parte inferior del programa existe un cuadro de texto en cual se muestran los mensajes de la consola del SGP Server. Debido a que no todos los mensajes mostrados por la consola están inglés (algunos se muestran en italiano) se ha aprovechado la redirección de la salida estándar de la consola del Servidor SGP para modificar algunos de los mensajes originalmente mostrados. También, aunque no se ha implementado, sería sencillo utilizar los mensajes del Servidor SGP para desencadenar eventos en la aplicación desarrollada. Como por ejemplo mostrar una ventana de aviso cuando un terminal FxLock se desconecte, o guardar en un fichero de log determinados eventos y su hora de ocurrencia.

3.6. Problemas encontrados durante el desarrollo del Proyecto

La realización de este Proyecto Fin de Carrera ha sido bastante problemática, ya que como se recoge en este apartado se ha hecho frente a multitud de problemas relacionados con el software y hardware de Biometrika. Muchos problemas se han solventado con éxito y se han resumido las tareas llevadas a cabo, sin embargo otros tantos no han podido resolverse, y se ha tratado de explicar en qué consisten.

Uno de los principales problemas que se ha tenido durante el estudio del Sistema FxGate está relacionado con la documentación del sistema, insuficiente, incompleta, a veces errónea y otras muchas inexistente. Pero los problemas de mayor importancia que han sido detectados derivan de un comportamiento incorrecto de los SDKs, o de un mal funcionamiento de los terminales FxLock utilizados.

3.6.1. **Problemas relacionados con el FxGate SDK Versión 3.00**

El primer problema nos lo encontramos nada más recibir los SDKs y los terminales de Biometrika e inspeccionar la documentación incluida. Conjuntamente con los dos FxLock se adjunta un CD-ROM que contiene el FxGate SDK 3.00, el BioCard SDK 1.01, y documentación electrónica de uso. Esta documentación consiste en los siguientes manuales:

- FxGate SDK Developers Manual Version 3.00. Este documento, salvo por un breve apartado sobre consideraciones criptográficas del Sistema FxGate, se limita a explicar las funciones del FxGate SDK y sus parámetros. Muchas de las definiciones de parámetros son incorrectas, e incluso el comportamiento de algunas funciones difiere del descrito. En ningún momento este manual menciona que entornos de desarrollo pueden utilizarse con las bibliotecas, ni tampoco la utilidad de los ficheros .lib y .h incluidos en el SDK.

- FxGate System Introduction Manual Version 3.01. Es una descripción de 6 hojas de lo que es un Sistema FxGate.

- BioCard SDK Developer's Manual Version 1.0.1. Describe las diferentes funciones del BioCard SDK y sus parámetros de uso. A diferencia del manual del FxGate SDK por lo menos menciona que para compilar un proyecto cuyo código utilice el BioCard SDK debe incluirse el fichero BioCard.h y añadir el fichero BioCard.lib a la línea de vinculación del compilador.

- FxLock User Manual. Version 1.03. Firmware Version 1.10. El manual de FxLock incluido es para terminales con firmware 1.10, y los terminales suministrados tienen firmware 2.00. En este manual no se cubren todas las opciones de los menús del FxLock, por ejemplo, las opciones relacionadas con tarjetas inteligentes no están incluidas.

El Servidor SGP se incluye sin ningún manual que explique cómo configurarlo o cual es su funcionamiento interno. Las tarjetas inteligentes suministradas se proporcionan 'a granel', sin ningún tipo de manual que describa el funcionamiento interno de las tarjetas, o sus comandos de comunicación. Ni si quiera se especifica el tipo de las tarjetas suministradas (que ha tenido que ser averiguado).

Como puede observarse la documentación ofrecida es incompleta, e incluso en algunos casos incorrecta, de manera que se ha trabajado para desarrollar una documentación más adecuada en el apartado 2.

El FxGate SDK 3.00 incluye tres proyectos de ejemplo en Visual Basic 6 y los correspondientes ejecutables resultado de su compilación. En un primer lugar y puesto que se contaba con las versiones 2005 y 2008 de Visual Studio (versiones 8 y 9, incompatibles con estos proyectos), se procedió a probar la funcionalidad de los ejecutables proporcionados. Para ello se deben situar en el directorio de cada programa, o en su defecto en C:\Windows\System32, las bibliotecas del FxGate SDK. Con este SDK se incluyen dos juegos de bibliotecas, contenidas en los directorios ENG e ITA del SDK. Se desconoce el por qué, pero se supone que está relacionado con el idioma utilizado en los nombres de las funciones. Se explica esto porque independientemente de las bibliotecas utilizadas y del directorio en el que éstas se coloquen las aplicaciones de prueba muestran el siguiente mensaje al invocar cualquier método del SDK:

Fig. 3.14: Mensaje de error del FxGate SDK 3.00 en Visual Studio 6.

Debido a esto nos vimos obligados a instalar la versión 6 de Visual Studio para inspeccionar y depurar el código de los proyectos demo proporcionados. Aún así no se encuentra ningún problema en el código y se comienza a sospechar que puede existir algún inconveniente con las bibliotecas proporcionadas.

Después de probar sin éxito los ejemplos proporcionados se procede a averiguar cómo utilizar las bibliotecas del FxGate SDK bajo una versión moderna de Microsoft Visual Studio. Una vez correctamente configurado el compilador y añadidos al proyecto los ficheros de encabezado proporcionados nos encontramos con que el fichero de encabezado SGADll.h proporcionado está incompleto. Una serie de constantes numéricas que definen longitudes de datos utilizados por el FxGate SDK, como la longitud de un identificador de usuario, del PIN, del Password y alguna más, no están definidas. Por lo que se tiene que inspeccionar los ejemplos en Visual Basic para averiguar sus valores.

Una vez completadas las definiciones correctamente el código de ejemplo escrito compila, pero en tiempo de ejecución se muestra el siguiente mensaje:

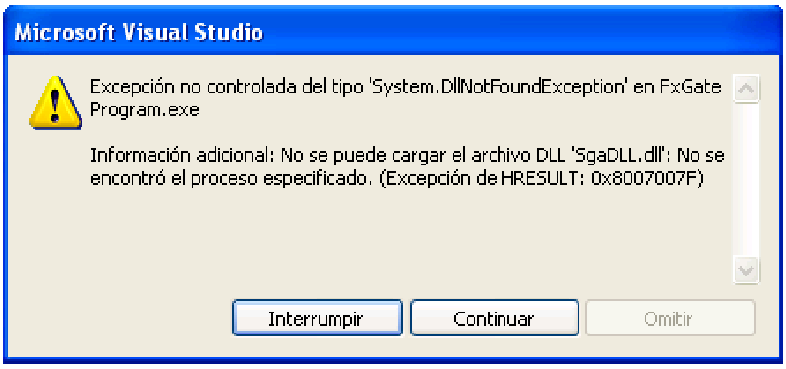

Fig. 3.15: Mensaje de error del FxGate SDK 3.00 en Visual Studio 2008.

Como se observa se produce un error equivalente al producido en los ejemplos de Visual Basic, lo que confirma que las bibliotecas proporcionadas no funcionan. Debido a esto nos ponemos en contacto con Biometrika para informarles de la situación. No es posible utilizar las bibliotecas, ni tampoco probar los ejemplos proporcionados, además los ficheros de encabezado del SDK están incompletos. La única solución por parte de Biometrika es enviar por correo electrónico la última versión del FxGate SDK, a la que nos referiremos como FxGate SDK Versión 3.00 Actualizada.

3.6.2. **Problemas relacionados con el FxGate SDK Versión 3.00 Actualizada**

El nuevo SDK recibido es aparentemente igual al proporcionado originalmente, sin embargo se observa como el directorio ITA ha sido eliminado, y las dlls del directorio ENG han sido modificadas por Biometrika, así como también el fichero .lib. Con estas nuevas bibliotecas puede por fin probarse el funcionamiento del SDK y los programas de ejemplo. De los tres programas de ejemplo proporcionados solamente uno, y eliminando la parte en la que interactúa con tarjetas RFID, funciona correctamente. De los otros dos uno utiliza funciones no documentadas en el FxGate SDK para enviar comandos a un terminal FxLock a través del Servidor SGP, y su ejecución no produce efecto alguno en los terminales. El otro programa aparentemente no funciona correctamente.

De esta manera, y visto que los ejemplos proporcionados no son de mucha ayuda y que no se tiene la intención de programar en Visual Basic, se pasa a probar a fondo el funcionamiento del FxGate SDK. Y se crea un proyecto de ejemplo de Visual Studio 2008 para analizar las diferentes funcionalidades ofrecidas por el Sistema FxGate.

El principal problema aparece cuando se quiere utilizar este SDK para insertar en el Servidor SGP un usuario con tarjeta. Ya que al utilizar la función FxGate_C_CreateBioCard se producen varios errores al intentar compilar el código:

```
[...]
Microsoft (R) Windows (R) Resource Compiler Version 6.0.5724.0
Copyright (C) Microsoft Corporation.  All rights reserved.
Vinculando...
FxGate Program.obj : error LNK2028: se hace referencia al símbolo (token) sin resolver (0A000046) "int __stdcall
FXGate_C_CreateBioCard(int,int,struct UserData *,int,unsigned char *,int,char *,int,int,unsigned char *)"
(?FXGate_C_CreateBioCard@@$$FYGHHHPAUUserData@@HPAEHPADHH1@Z) en la función "private: void __clrcall
FxGateProgram::Form1::button4_Click(class System::Object ^,class System::EventArgs ^)"
(?button4_Click@Form1@FxGateProgram@@$$FA$AAMXP$AAVObject@System@@P$AAVEventArgs@4@@Z)
FxGate Program.obj : error LNK2019: símbolo externo "int __stdcall FXGate_C_CreateBioCard(int,int,struct UserData *,int,unsigned
char *,int,char *,int,int,unsigned char *)" (?FXGate_C_CreateBioCard@@$$FYGHHHPAUUserData@@HPAEHPADHH1@Z) sin resolver
al que se hace referencia en la función "private: void __clrcall FxGateProgram::Form1::button4_Click(class System::Object ^,class
System::EventArgs ^)"
(?button4_Click@Form1@FxGateProgram@@$$FA$AAMXP$AAVObject@System@@P$AAVEventArgs@4@@Z)
C:\Documents and Settings\Owner1\Escritorio\FxGate Program\Debug\FxGate Program.exe : fatal error LNK1120: 2 externos sin
resolver
El registro de compilación se guardó en el "file://c:\Documents and Settings\Owner1\Escritorio\FxGate Program\Debug\BuildLog.htm"
FxGate Program - 3 errores, 0 advertencias
========== Generar: 0 correctos, 1 incorrectos, 0 actualizados, 0 omitidos ==========
```

Fig. 3.16: Registro de compilación con el FxGate SDK 3.00 modificado en Visual Studio 2008

Estos errores indican que la función FxGate_C_CreateBioCard no está correctamente implementada en las bibliotecas, y por tanto no se puede utilizar. Así pues nos vemos obligados a contactar de nuevo con Biometrika, y su solución tras un par de semanas de espera vuelve a ser nuevamente otro SDK, esta vez la versión 3.10.

3.6.3. **Problemas relacionados con el FxGate SDK Versión 3.10 y el BioCard SDK**

Se ha trabajado con dos versiones del BioCard SDK, la 1.01 incluida en el FxGate SDK 3.00 y la 1.02, incluida en el FxGate SDK 3.10. Ambas versiones funcionan exactamente igual y no se ha apreciado diferencia alguna en cuanto a comportamiento. En el FxGate SDK 3.10 la función FxGate_C_CreateBioCard está implementada, y se ha comprobado cómo es posible inicializar tarjetas inteligentes con contactos y escribir el patrón biométrico del usuario en ellas. La inicialización de tarjetas RFID no se ha podido probar, ya que no se ha contado con un lector RFID de Biometrika.

Ha sido detectado un serio inconveniente relacionado con las tarjetas y el FxGate SDK, el BioCard SDK y los terminales FxLock. Para asociar correctamente una tarjeta inteligente con contactos a un usuario es imprescindible haber utilizado previamente la función FxGate_C_CreateBioCard del FxGate SDK para inicializar la tarjeta. Sería lógico utilizar el BioCard SDK para formatear la tarjeta inteligente con claves por defecto, y para escribir en ella el identificador del usuario, su patrón biométrico y el umbral de reconocimiento. Sin embargo si posteriormente se asocia a un usuario esta tarjeta y se inserta en el Servidor SGP con la función FxGate_InsertStruct, a la hora de introducir la tarjeta inteligente en el FxLock se muestra el mensaje 'Inconsistent Data' en la pantalla. Sucede lo mismo al modificar con el BioCard SDK el patrón biométrico de una tarjeta escrito con la función FxGate_C_CreateBioCard del FxGate SDK.

Se ha notificado a Biometrika en repetidas ocasiones acerca de esta incompatibilidad detectada pero no han ofrecido ninguna solución. Así pues se ha podido comprobar como la

única manera de poder utilizar correctamente tarjetas en el sistema FxGate es utilizar la función FxGate_C_CreateBioCard para inicializarlas, (lo que provoca el borrado de todos los datos de la memoria de usuario), y utilizar el BioCard SDK solamente para leer o escribir ficheros en la tarjeta y leer el patrón biométrico del usuario.

3.6.4. **Problemas relacionados con los terminales FxLock**

Existen graves problemas con los terminales FxLock utilizados. El FxLock dotado con lector de tarjetas inteligentes con contactos solamente es capaz de leer las tarjetas durante un breve período de tiempo, inferior a dos minutos, e inmediatamente después de ser encendido. Si el terminal lleva un rato conectado y se introduce una tarjeta en el lector el terminal no la detecta, ni tampoco puede efectuarse con éxito un test del lector de tarjetas desde los menús de configuración del FxLock. Todo hace indicar que existe un problema con el módulo controlador de la unidad lectora, y que al poco tiempo de ser encendido el FxLock el lector se bloquea, dejando de funcionar. Este problema no ha podido solucionarse, pese a habérselo hecho saber a Biometrika en reiteradas ocasiones.

El terminal dotado de lector RFID también ha sido bastante problemático. Desde un principio no era posible conectarlo al Servidor SGP a través del puerto Ethernet, ya que al seleccionar esta interfaz en la pantalla del FxLock se mostraba el mensaje "error". Solamente era posible conectar este terminal a través de su puerto serie.

Otro problema detectado está relacionado con el funcionamiento de las tarjetas RFID. Si se efectúa un test del lector de RFID al aproximar una tarjeta la pantalla del FxLock muestra un mensaje 'Found card: AT88SC6416C Size=8192', pero si trabajando en modo local se intenta insertar un usuario con tarjeta en el FxLock directamente desde el propio terminal, al escribir los datos en la tarjeta se muestra un mensaje 'Writing card error Retry. N.:1048582'.

Tras ponernos en contacto con Biometrika y comunicarles los problemas con esta unidad se nos suministró una nueva versión de firmware, válida solamente para el FxLock con lector RFID (2.05 beta). Una vez actualizado por puerto serie el firmware del terminal el puerto Ethernet comenzó a funcionar. Sin embargo el comportamiento con las tarjetas se mantuvo invariable, con el nuevo firmware se muestra un número de mensaje de error diferente, pero sigue sin ser posible que el FxLock inicialice tarjetas RFID.

De nuevo se volvió a contactar con Biometrika, y como no se obtuvo una solución finalmente se acabó desistiendo de seguir intentándolo. Se desconoce si existe un fallo en las rutinas que manejas las tarjetas RFID en el terminal, o si existe algún tipo de incompatibilidad con las tarjetas proporcionadas. Al no disponer tampoco de un lector Fx2000RF de Biometrika no se ha podido comprobar si el FxGate SDK puede inicializar correctamente este tipo de tarjetas.

3.7. Conclusiones sobre el FxGate

El Sistema de control de acceso FxGate de Biometrika se trata de una solución cerrada, en el sentido de que estamos obligados a utilizar el servidor de gestión programado por Biometrika, los terminales de control de acceso de Biometrika y las tarjetas inteligentes proporcionadas por Biometrika.

El Servidor SGP es quien lleva a cabo toda la complejidad del proceso de gestión de los terminales de control de acceso. Está programado por Biometrika y su código fuente no se suministra. Es el único elemento del Sistema FxGate capaz de dialogar con los terminales FxLock y el protocolo de comunicación que utiliza es también desconocido. En el Sistema FxGate no se sabe realmente como funciona por dentro el Servidor SGP, ni tampoco como son los mensajes que se intercambian con los terminales FxLock. Su funcionamiento es totalmente privado y Biometrika no ha tenido en ningún momento la intención de facilitar tal información.

El funcionamiento interno de los terminales FxLock es también privado. Su firmware solamente es modificable por Biometrika, que es la única que conoce el modo de operación del dispositivo. Se depende de Biometrika para que ésta desarrolle futuros firmwares que solucionen problemas existentes, o añadan nuevas funcionalidades a los terminales.

Las funciones de bajo nivel utilizadas para la comunicación con las tarjetas no han sido facilitadas por Biometrika, y tampoco se conoce como los terminales FxLock se comunican con las tarjetas. De esta manera si se pretende en el Sistema FxGate adoptar una solución basada en tarjetas inteligentes no hay más remedio que utilizar las tarjetas inteligentes de Biometrika.

El Sistema FxGate es también una solución no flexible, ya que toda la funcionalidad del sistema recae en el Servidor SGP y éste viene ya programado, desconociéndose todos los detalles internos de su funcionamiento. En el momento en el que toda la complejidad de la gestión recae en un componente cuyo funcionamiento interno es desconocido se pierde el control de lo que realmente sucede en el sistema.

El hecho de que la gestión de todo el sistema sea centralizada no debería suponer un problema, ya que simplifica las tareas de gestión, pero en el sistema estudiado es un grave inconveniente. Las funcionalidades ofrecidas por el servidor son las que son, las que han sido implementadas por Biometrika. Resulta imposible añadir nuevas funciones al servidor o personalizar características ya existentes, ya que como se ha explicado el Sistema FxGate es una solución cerrada.

El sistema de control de acceso estudiado tampoco es flexible respecto al resto de elementos hardware que componen el sistema. La arquitectura y funcionamiento interno de los terminales FxLock solamente es conocida por Biometrika. No existe la posibilidad de modificar el firmware de los FxLock por uno personalizado, que permita programar el terminal según se desee. Esto impide dotar a los terminales de todo tipo de funcionalidad adicional. Y al igual que sucede con el Servidor SGP, se depende de Biometrika para que ésta lance nuevos firmwares que doten a los terminales FxLock de funcionalidades adicionales.

Otro aspecto no flexible del sistema son las tarjetas inteligentes, ya que solamente pueden utilizarse tarjetas y lectores suministrados por Biometrika. La forma en la cual un FxLock se comunica con las tarjetas se desconoce, al igual que los comandos de comunicación de bajo nivel. Esto provoca que no sea posible el empleo de cualquier otro tipo de solución basada en tarjetas inteligentes. Podría pensarse el diseñar un sistema biométrico de control de acceso que utilice un determinado tipo de tarjeta inteligente cuyo protocolo de comunicación a bajo nivel sea totalmente conocido y se tenga ya implementado. Pero con un Sistema FxGate no puede llevarse a cabo.

El Sistema FxGate es una solución inestable. Como se ha explicado en el apartado 3.5 ha resultado imposible que el FxLock con lector de tarjetas inteligentes con contactos funcione correctamente, y el FxLock con lector RFID ni siquiera ha podido utilizarse para inicializar tarjetas RFID. Además Biometrika no ha ofrecido una solución válida que resuelva estos problemas. Con las versiones de firmware y software que se ha trabajo no es viable el desarrollar un sistema de control de acceso biométrico que utilice tarjetas inteligentes para almacenar el patrón de huella de los usuarios.

El único punto personalizable en el Sistema FxGate es el Programa FxGate desarrollado con el FxGate SDK. Sin embargo las funciones que pueden implementarse dependen de las funcionalidades ofrecidas por el Servidor SGP. Si lo que se desea hacer en el sistema no está incluido en la interfaz ofrecida por el Servidor SGP de Biometrika sencillamente no se puede hacer. Además se ha detectado como algunas funciones del FxGate SDK no funcionan bien, o su comportamiento no es del todo correcto.

El BioCard SDK no resulta demasiado útil. Solamente sirve para escribir datos en tarjetas inteligentes con contactos de Biometrika. Al igual que sucede con el FxGate SDK el funcionamiento interno del BioCard SDK es privado, y por tanto no es posible utilizarlo para gestionar otros tipos de tarjetas, como tarjetas RFID. El uso de este SDK sería prescindible, ya que el FxGate SDK tiene la capacidad de inicializar las tarjetas inteligentes de Biometrika, y escribir y leer el patrón biométrico almacenado por éstas. Además se han encontrado serias incompatibilidades con el FxGate SDK, puesto que para que las tarjetas con contactos proporcionadas sean correctamente leídas por el FxLock deben haber sido creadas con el FxGate SDK y no con el BioCard SDK.

Por todo lo expuesto anteriormente se concluye que el Sistema FxGate de Biometrika es una solución de control de acceso cerrada, no flexible e inestable. En la cual no se tiene un control absoluto sobre el sistema. Puede resultar válida para aplicaciones biométricas de control de acceso o control de asistencia sencillas, en las cuales los requisitos sean cubiertos por las características ofrecidas por el Sistema FxGate, y siempre y cuando no se pretenda utilizar tarjetas inteligentes. Pero el sistema estudiado no es válido para aplicaciones de control de acceso avanzadas o personalizadas.

4. ESPECIFICACIONES Y DISEÑO DE UN NUEVO SISTEMA DE ACCESO

En el Sistema FxGate de Biometrika se han encontrado multitud de problemas. Se trata de una solución cerrada, no personalizable, el funcionamiento interno de los componentes no es conocido, y se han detectado graves inconvenientes en el uso de tarjetas inteligentes. Por todo esto a continuación se propone un nuevo sistema de control de acceso, indicándose las especificaciones de sus componentes, así como su funcionamiento.

4.1. Requisitos del sistema de acceso

En primer lugar el sistema de control de acceso propuesto debe cumplir una serie de características generales:

- Extremadamente seguro. Puesto que empleará la biometría para la identificación de sujetos deben utilizarse técnicas biométricas que presenten un elevado índice de unicidad, resulten cómodas de utilizar para el usuario y minimicen la probabilidad de accesos no autorizados.

- Seguro en el sentido de que se garantice la privacidad de los datos intercambiados en las comunicaciones entre los diferentes componentes del sistema. Para ello se utilizarán técnicas criptográficas como cifrado asimétrico, claves de sesión o firma digital.

- Robusto, estable, capaz de mantenerse siempre en un estado controlado, y que ofrezca mecanismos adicionales de recuperación ante fallos.

- Flexible en cuanto a la posibilidad de utilizar diferentes componentes hardware, como terminales de acceso, tarjetas inteligentes o lectores biométricos. Y flexible en cuanto a que resulte válido para implementar diferentes aplicaciones, como control de acceso, de presencia, asistencia, visitas, etc.

- Ampliable. Que sea posible introducir en el sistema nuevos componentes, y también incorporar nuevas técnicas biométricas de identificación.

- Personalizable. El comportamiento de los diferentes componentes del sistema debe poder ser modificado para adaptarse a aplicaciones específicas o personalizadas.

Para verificar los requisitos impuestos al sistema de control de acceso propuesto los terminales de control de acceso necesarios deben contar con las siguientes especificaciones técnicas:

- Escáner de huella dactilar con amplia área de detección y alta resolución. De manera que se obtengan en el sistema unos buenos parámetros biométricos de calidad.

- Lector de tarjetas inteligentes con contactos, que permita trabajar con tarjetas inteligentes de distintos fabricantes que cumplan el estándar ISO/IEC 7816.

- Lector de tarjetas inteligentes sin contactos (RFID), capaz de manejar tarjetas cuyo protocolo de comunicación siga el estándar ISO/IEC 14443, incluyendo tarjetas de tipo A y B.

- Conexión de red de banda ancha. Utilizable para la gestión remota del terminal de acceso, soportando la recepción de comandos de configuración procedentes de un servidor de gestión. Esta conexión puede utilizarse también para el envío de registros de acceso a un servidor de monitorización del sistema de control de acceso, y también para consultar a una base de datos centralizada información sobre usuarios registrados en el sistema, o para registrar nuevos usuarios.

- Microcontrolador (microprocesador + memoria RAM) lo suficientemente potente como para controlar los diferentes componentes hardware que integra el terminal de acceso, y llevar a cabo todas las tareas posibles por el terminal. Como por ejemplo, procesar con rapidez la huella dactilar de un usuario para extraer el vector de características y realizar la identificación, almacenar en una memoria local información relacionada con los permisos de acceso de determinados usuarios, o consultar a través de la interfaz de red información de una base de datos centralizada.

- Controladora de relés integrada para su uso con cerraduras electrónicas o dispositivos de alarma, de modo que los contactos correspondientes se activen de manera automática tras la identificación y autorización del usuario.

- Batería de emergencia que permita la correcta operación de los terminales de acceso aún cuando se produzcan cortes de electricidad, dotando a los terminales de una cierta autonomía de funcionamiento.

- Interfaz de uso amigable e intuitiva. El terminal por un lado debe resultar sencillo de utilizar para los usuarios, y por otro debe permitir a los administradores del sistema configurar y visualizar los parámetros de configuración del sistema desde la propia pantalla del terminal.

Además de las características técnicas anteriormente mencionadas los terminales de control de acceso deben ofrecer ciertas facilidades para ser personalizados:

- La arquitectura y funcionamiento interno del terminal deben ser perfectamente conocidas. De manera que sea posible personalizar el firmware del dispositivo para

programar el funcionamiento del terminal, añadiendo nuevas funciones o modificando su comportamiento.

- Exista la posibilidad de añadir periféricos como sensores, micrófonos, o cámaras que permitan la adopción de otros tipos de técnicas biométricas para la identificación. Es decir, que los terminales de control de acceso no sirvan única y exclusivamente para aplicaciones basadas en huella dactilar.

- Los terminales estén preparados para ser utilizados en aplicaciones de control de acceso, control de tiempo y asistencia, entrada y salida de usuarios, etc.

4.2. Descripción del sistema

En rasgos generales el sistema de control de acceso biométrico propuesto se trata de un sistema basado en huella dactilar, al igual que el Sistema FxGate. Sin embargo se contempla la posibilidad de que el sistema sea ampliable, en el sentido de que resulte posible la utilización de otros tipos de técnicas biométricas además del reconocimiento de huella.

El sistema de control de acceso se basa en el uso de tarjetas inteligentes, pero a diferencia del Sistema FxGate se concede una mayor importancia al uso de tarjetas. Las tarjetas inteligentes no solamente almacenan el patrón biométrico del usuario, sino que también pueden contener la lista de permisos de acceso de un usuario. Es decir, cuáles y en qué circunstancias son las zonas del sistema a las que un usuario puede acceder. Esto conlleva un ahorro de memoria en los terminales de acceso, ya que se prescinde de la necesidad de que cada terminal biométrico almacene la lista de usuarios del sistema y sus permisos de acceso.

La arquitectura del sistema de control de acceso se compone de una serie de terminales de control de acceso, un servidor de gestión/monitorización y una base de datos de usuarios.

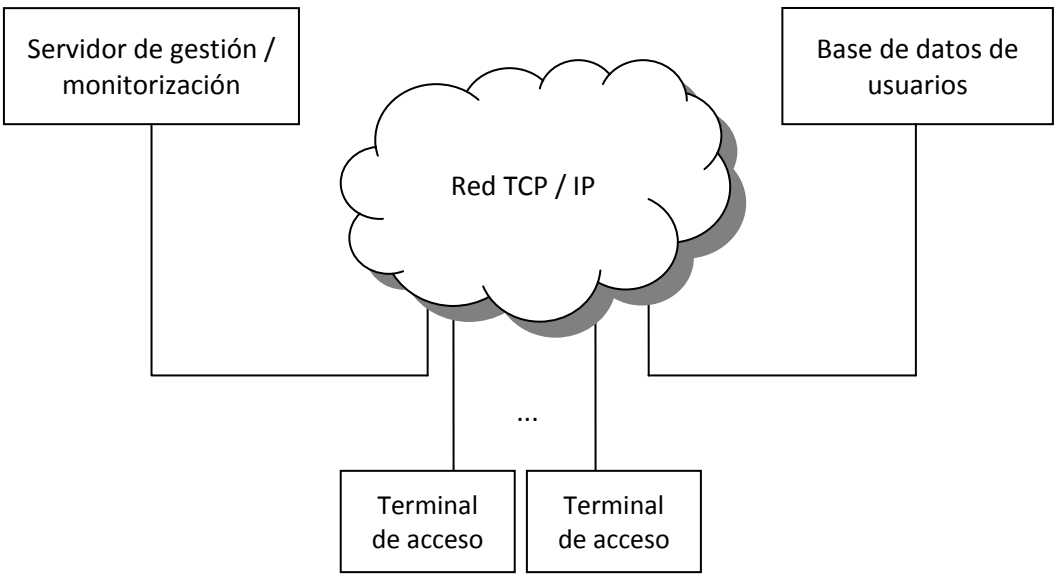

Fig. 4.1: Arquitectura del sistema de control de acceso propuesto.

- Terminales de control de acceso

Los terminales de control de acceso están conectados en red con el servidor de gestión/monitorización y con la base de datos, y se encuentran localizados en aquellos puntos del sistema en los cuales se desea controlar o restringir el acceso de usuarios.

Los terminales de acceso pueden comunicarse con el servidor de gestión/monitorización para enviar información relacionada con eventos de acceso y recibir comandos de configuración remota.

Los terminales también pueden comunicarse con la base de datos, ya sea para solicitar información sobre los permisos de acceso de un usuario identificado, o para insertar un nuevo usuario.

- Servidor de gestión/monitorización.

Permite configurar de manera remota los terminales que forman parte del sistema de acceso, gestionar los usuarios almacenados en la base de datos, y monitorizar los sucesos ocurridos en el sistema. Para llevar a cabo la última tarea recibe periódicamente información de los terminales, que es procesada y analizada. También puede solicitar a los propios terminales información sobre su estado.

- Base de datos de usuarios.

Contiene la lista de usuarios registrados en el sistema de control de acceso y sus permisos de acceso. Se proponen varios modos diferentes para llevar a cabo el proceso de autorización de acceso de un usuario, que afectan a la comunicación entre la base de datos de usuarios y el resto de componentes del sistema.

Los permisos de acceso de un usuario pueden estar almacenados en la tarjeta inteligente asociada, que contiene también su patrón biométrico. En este caso no es necesario que el terminal biométrico consulte a la base de datos, ya que la tarjeta contiene toda la información necesaria para autorizar o denegar el acceso al usuario a partir de su huella dactilar. Pero sí podría considerarse la posibilidad de que el terminal consultara en la base de datos los permisos de acceso del usuario para contrastar su validez.

Otra alternativa es que los permisos de acceso del usuario estén almacenados en la memoria interna de cada terminal de acceso, y no en la tarjeta inteligente. En este caso cada terminal ha tenido que solicitar con anterioridad a la base de datos la lista de usuarios y sus permisos de acceso. Supone que cada terminal almacena en su memoria interna los diferentes usuarios que pueden acceder al sistema y sus permisos de acceso.

Por último también se contempla la posibilidad de que los permisos de acceso solamente estén almacenados en la base de datos de usuarios. De esta manera cada vez que un usuario pretenda acceder al sistema el terminal de acceso debe consultar en la base de datos los correspondientes permisos de acceso del usuario. Esto requiere que la base de datos siempre se encuentre operativa para poder acceder al sistema.

4.3. Funcionamiento del sistema

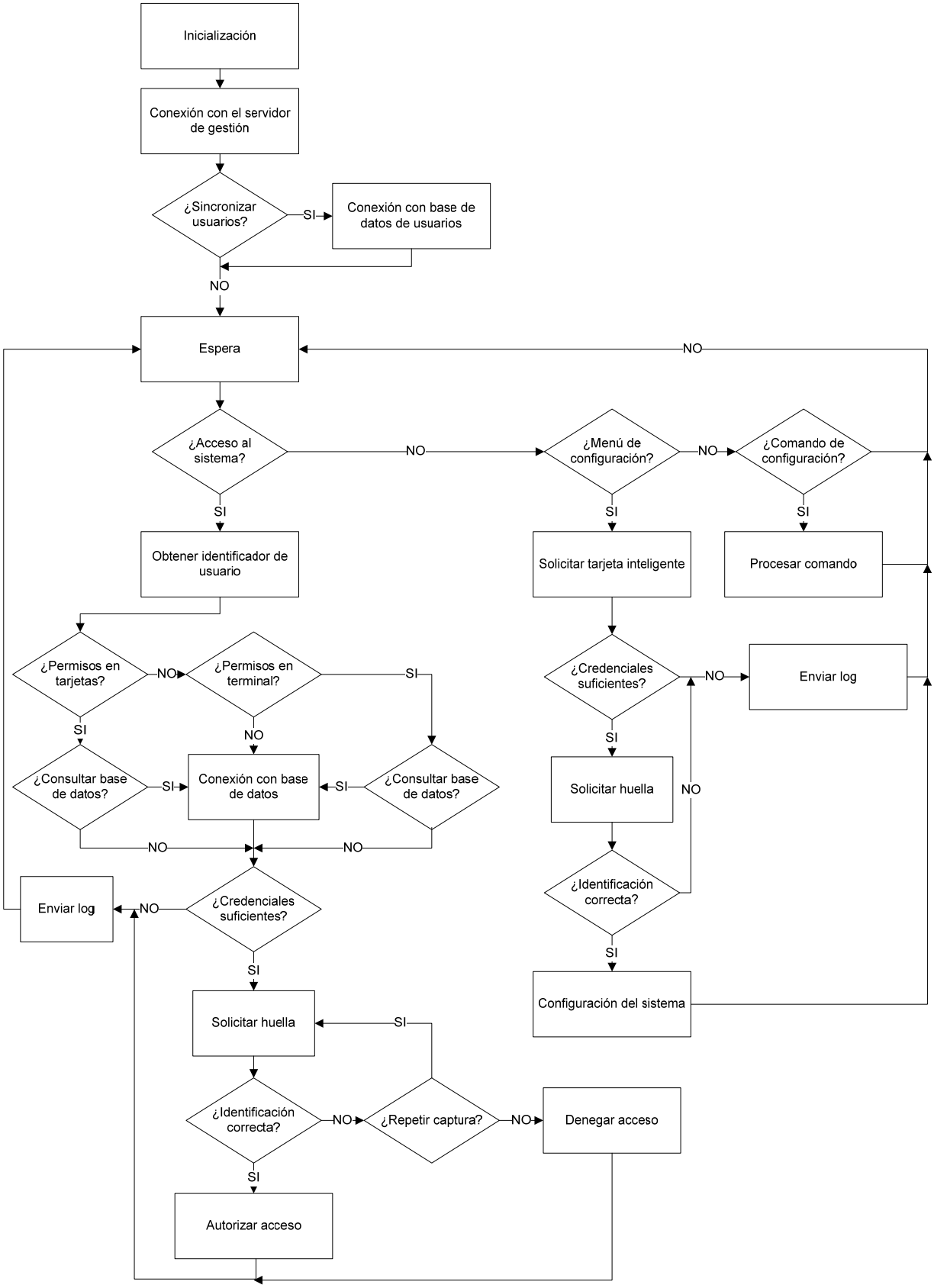

Fig. 4.2: Diagrama de flujo del funcionamiento del terminal de control de acceso.

El funcionamiento básico de los terminales de control de acceso se recoge en el diagrama de flujo de la figura 4.2.

Inmediatamente después del encendido del terminal tiene lugar el proceso de inicialización del aparato. Se realizan una serie de comprobaciones para verificar el estado de los diferentes componentes hardware del dispositivo, así como la integridad del firmware. Una vez inicializado el terminal éste establece una conexión con el servidor de gestión/monitorización, que en el caso de que proceda puede enviar al terminal comandos de configuración. La conexión permanece abierta, y periódicamente el terminal notifica de su estado al servidor.

Si el terminal está configurado para almacenar la lista de usuarios y permisos de acceso establece una conexión con la base de datos y obtiene la lista actualizada. En caso contrario o una vez completada la sincronización de datos el terminal pasa a modo de espera.

Durante el modo de espera el terminal de acceso permanece a la espera de que se produzca un acceso en el sistema, se reciba un comando de configuración procedente del servidor de configuración o un administrador acceda al menú de configuración del terminal.

En el caso de que se detecte una solicitud de acceso, bien porque el usuario introduce su tarjeta inteligente con contactos en el lector, o bien porque el lector de RFID detecta una tarjeta en su radio de acción, lo primero que realiza el terminal es obtener de la tarjeta el identificador único asociado al usuario.

Una vez obtenido el identificador de usuario el terminal lleva a cabo un proceso para obtener los permisos de acceso del usuario, que varía según esté configurado el terminal. Si los permisos de acceso están contenidos en la tarjeta inteligente el terminal puede consultar o no a la base de datos para contrastarlos, según la configuración del terminal o del usuario. Si la tarjeta inteligente no contiene los permisos de acceso el terminal puede consultar su memoria interna y decidir si debe consultar o no en la base de datos los permisos de acceso del usuario.

Después de obtener los permisos de acceso se verifica si el usuario dispone de credenciales suficientes para acceder al sistema. En caso afirmativo se invita al usuario a que posicione su dedo en el lector de huella para digitalizarla y obtener su patrón biométrico, que es comparado en el interior de la tarjeta con el patrón biométrico del usuario. Si como resultado de la comparación se supera un cierto nivel de semejanza se autoriza el acceso al usuario. En el caso de que la comparación no sea satisfactoria puede repetirse el proceso un par de veces, solicitando de nuevo al usuario que posicione su dedo sobre el lector.

Tras la autorización del acceso el terminal puede mostrar en su pantalla una lista de tareas que el usuario puede seleccionar, o directamente activar un contacto que abra por ejemplo una puerta, según el tipo de autorización concedida al usuario y de la aplicación concreta de control de acceso. Una vez completado el proceso el terminal notifica al servidor de monitorización y vuelve de nuevo a estado de espera.

Si un administrador ha solicitado al terminal mostrar el menú de configuración (se sugiere que esto se haga tecleando un código secreto en el terminal), éste pedirá al administrador que introduzca su tarjeta inteligente, comprobando si dispone de permisos de administrador del

terminal. En caso afirmativo el administrador deberá posicionar su dedo sobre el lector y probar su identidad. Si la identificación es satisfactoria se mostrará el menú de configuración del terminal de acceso.

Se propone que desde el menú de configuración del terminal sea posible configurar los periféricos del terminal, el modo de funcionamiento del aparato y también sea posible llevar a cabo algunas tareas de gestión de usuarios. Como la consulta de permisos de acceso o la inserción en el sistema de nuevos usuarios previamente pre-configurados en el servidor de gestión, dejando para el servidor de gestión/monitorización el grueso de la gestión de usuarios.

En el caso de que el terminal de acceso reciba un comando de configuración procedente del servidor de gestión primero comprueba si está llevando a cabo una operación de identificación, caso en el que encola la solicitud recibida hasta que la operación en curso finalice. O si bien se encuentra activo el menú de configuración, caso en el que puede decidir si cerrar la interfaz de configuración y atender la solicitud o esperar.

Se propone que desde el servidor de gestión sea posible configurar todos los parámetros seleccionables desde el menú de configuración. Y además características adicionales solamente configurables desde remoto, como listas de usuarios a almacenar en los terminales, o parámetros de configuración avanzados que determinen el comportamiento del terminal.

El funcionamiento simplificado de la base de datos de usuarios es el siguiente:

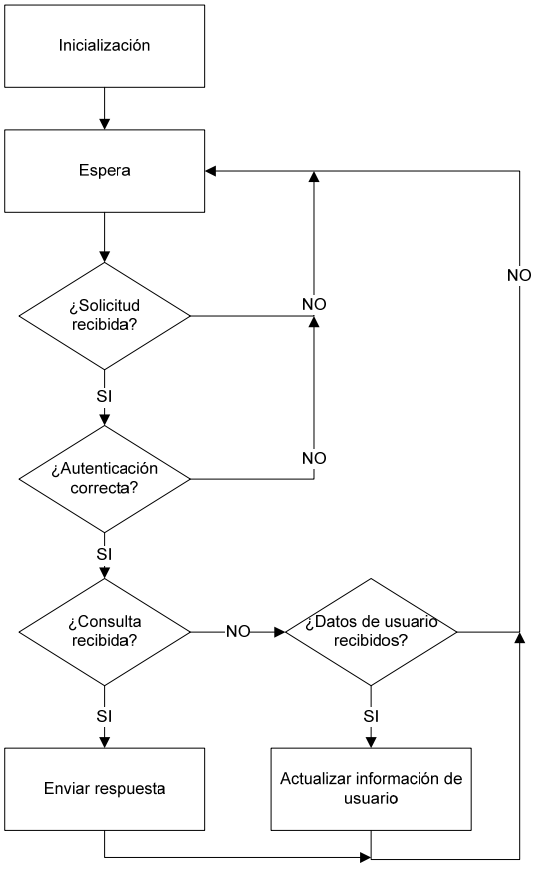

Fig. 4.3: Diagrama de flujo del funcionamiento de la base de datos de usuarios.

En primer lugar se realiza la inicialización de la base de datos, comprobándose la integridad de los datos y la conectividad de red. Una vez completadas estas tareas se pasa a estado de espera.

Tras recibir una consulta lo primero que se realiza es un proceso de autenticación, ya que únicamente los terminales de control de acceso y el servidor de gestión pueden comunicarse con la base de datos. Además la información intercambiada va cifrada con una clave de sesión.

Si la solicitud recibida corresponde a una petición sobre los permisos de un determinado usuario se consulta la información, se elabora la respuesta y se pasa de nuevo al estado de espera. Si se trata de una actualización de datos o de la inserción de un nuevo usuario se procesa la información recibida para añadir los nuevos datos a la base de datos.

El funcionamiento general del servidor de gestión/monitorización se representa en el diagrama de flujo de la Figura 4.4.

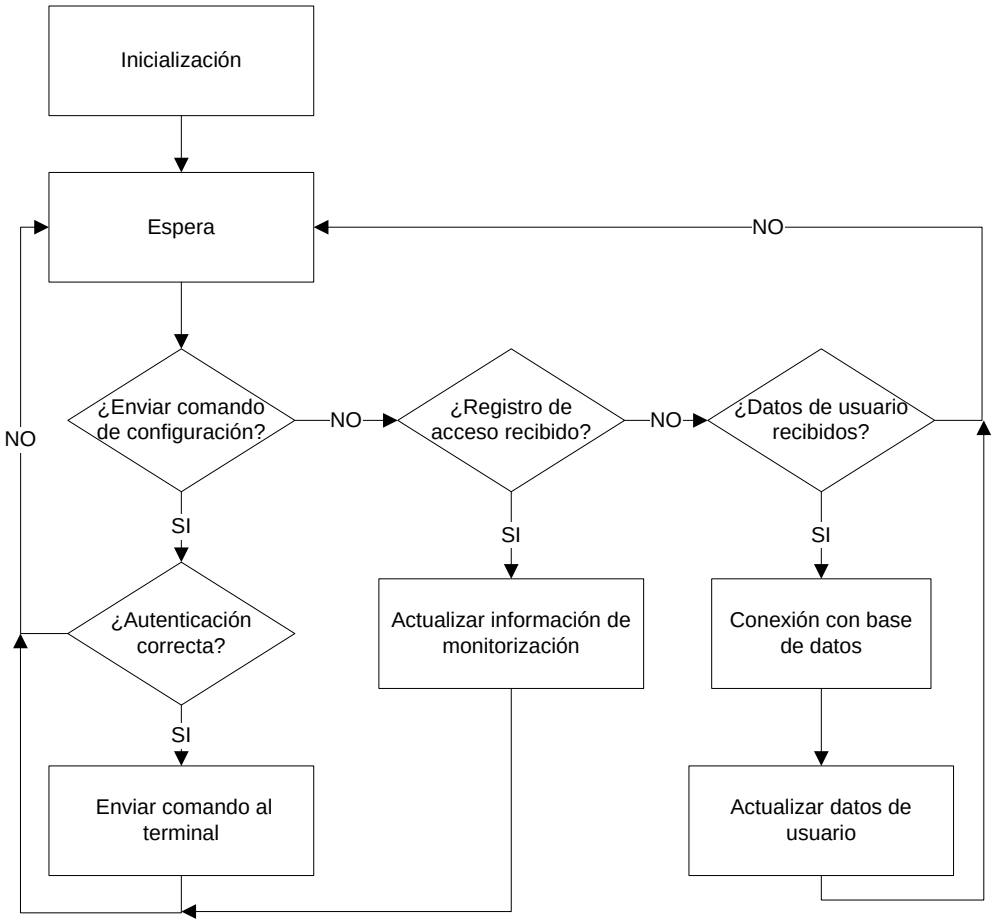

Fig. 4.4: Diagrama de flujo del funcionamiento del servidor de gestión/monitorización.

El servidor de gestión dispone de una interfaz gráfica desde la cual monitorizar los terminales de acceso, configurar remotamente los terminales y administrar los usuarios almacenados en la base de datos.

Durante la fase de inicialización se arranca el servidor de gestión/monitorización, se inicia la interfaz gráfica que permite controlar el sistema y el servidor permanece a la espera de recibir conexiones desde los terminales de acceso, o se envíen comandos de configuración a los terminales desde la aplicación de gestión.

En la aplicación de gestión se presentan los registros de acceso a medida que se producen. Para ello mantiene una conexión activa con cada uno de los terminales y periódicamente recibe información relacionada con eventos de acceso.

Permite la gestión avanza de los usuarios del sistema de control de acceso, así como la modificación de sus permisos de acceso. Cada vez que se produce una consulta o actualización de los datos de los usuarios almacenados en la base de datos se establece una conexión con la base de datos.

Desde el servidor de gestión también es posible modificar la configuración de los terminales de acceso. Para ello se establece una conexión entre el servidor de gestión y el terminal de acceso, ambos elementos se autentican, negocian una clave de sesión y se envían los comandos de configuración desde el servidor de gestión.

5. CONCLUSIONES Y LÍNEAS DE FUTURO

En el presente Proyecto Fin de Carrera se ha estudiado el funcionamiento y las posibilidades ofrecidas por un sistema de control de acceso biométrico comercial, concretamente el Sistema FxGate de la compañía italiana Biometrika.

Durante el desarrollo de este proyecto se ha tenido que hacer frente a multitud de problemas. Desde un primer momento el funcionamiento del sistema no ha sido del todo correcto, la documentación disponible ha sido escasa y en algunos casos errónea, y la comunicación con el fabricante ha dejado mucho que desear. Aún así se ha trabajado para documentar el Sistema FxGate, e intentar resolver los problemas encontrados.

Después de analizar a fondo el Sistema FxGate puede afirmarse que el sistema estudiado no funciona bien y además no se tiene un control total sobre el mismo, ya que el funcionamiento interno del Servidor SGP y de los terminales FxLock se desconoce. Esto provoca que no puedan añadirse nuevas funcionalidades al sistema. Las únicas funcionalidades posibles son las ofrecidas por el Servidor SGP programado por Biometrika. Tampoco es posible utilizar lectores de huella, de tarjetas inteligentes o tarjetas que no sean los suministrados por Biometrika.

La realización de este proyecto ha permitido profundizar los conocimientos personales sobre biometría y tarjetas inteligentes. Ya que anteriormente a su desarrollo se realizó un estudio de las diferentes técnicas de identificación biométrica, y también de tarjetas, analizándose los bloques funcionales de una tarjeta inteligente con contactos.

El proyecto también ha servido para adquirir experiencia en el uso de un entorno de programación visual no utilizado nunca antes, Microsoft Visual Studio 2008. Puesto que se han diseñado dos aplicaciones diferentes para la gestión del Sistema FxGate se han adquirido los conocimientos necesarios para programar aplicaciones gráficas sencillas en un entorno Windows, utilizando dos lenguajes de programación diferentes, Visual C++ y Visual C#.

En lo referente a los sistemas de control de acceso se ha trabajado con terminales de control de acceso biométricos profesionales, disponibles en el mercado. Se han analizado las características y funcionalidades ofrecidas por un sistema de control de acceso, y puesto que se han encontrado numerosas deficiencias se ha propuesto un nuevo sistema de control de acceso. Especificándose las características que el sistema y sus terminales deberían tener, así como cuál debería ser su funcionamiento.

Se proponen varias líneas futuras de trabajo relacionas con este proyecto. La primera de ellas consiste en aprovechar la documentación escrita y los programas desarrollados para seguir trabajando con el Sistema FxGate de Biometrika. De manera que una vez resueltos todos los problemas que no han podido solucionarse, especialmente los problemas relacionados con la utilización de tarjetas inteligentes en el sistema, se implemente una aplicación de control de acceso mucho más avanzada. Esta aplicación podría por ejemplo contemplar la definición de diferentes perfiles de seguridad, un control más detallado sobre la

entrada y salida de usuarios, o una gestión más avanzada de la información contenida en las tarjetas inteligentes.

Para la futura realización de esta parte es imprescindible que en primer lugar Biometrika solucione los problemas encontrados con la utilización de tarjetas inteligentes en su sistema. También resultaría conveniente que Biometrika facilitara información, hasta el momento privada, relacionada con la manipulación interna de sus tarjetas por los terminales FxLock. De esta manera otra línea posible de trabajo sería la implementación de una interfaz común que permitiera utilizar cualquier tipo de tarjetas inteligentes en el mercado, sin más que añadir unas funciones apropiadas de comunicación de bajo nivel.

Otra línea de trabajo posible consistiría en la modificación del firmware de los terminales FxLock, de manera que fuera posible programar a voluntad el funcionamiento de los terminales. Puesto que Biometrika no ha querido facilitar en ningún momento ningún tipo de información relacionada con la arquitectura y funcionamiento interno de sus terminales esta tarea se presenta a prior bastante complicada.

También se propone como futura línea de trabajo el estudio detallado del Servidor SGP con el objeto de reemplazar el servidor SGP programado por Biometrika por un servidor personalizado. Ya que Biometrika tampoco ha querido facilitar información detallada de su funcionamiento interno resultaría necesario analizar las diferentes tramas intercambiadas por Servidor SGP y terminales FxLock, averiguando en primer lugar el tipo de cifrado y autenticación utilizado en las comunicaciones.

Por último se propone como línea de trabajo la implementación de un nuevo sistema de control de acceso, basado en el sistema propuesto en el apartado 4. En este caso habría que buscar en primer lugar terminales biométricos de control de acceso capaces de verificar los requisitos propuestos en el apartado 4.1.

6. REFERENCIAS BIBLIOGRÁFICAS

[1] ***Identificación Biométrica y su unión con las Tarjetas Inteligentes***. Raúl Sánchez Reíllo. Ágora SIC Divulgación. Abril 2000.

[2] ***Identificación biométrica***. Joaquín González Rodríguez. 14/07/2000. http://www.instisec.com/publico/verarticulo.asp?id=20

[3] ***Biometría: ¿Realidad o ficción?*** Orestes Sánchez. 22/12/2005. http://sociedaddelainformacion.telefonica.es/jsp/articulos/detalle.jsp?elem=1744

[4] ***Técnicas de seguridad biométricas***. Network World. 01/09/2004. http://www.idg.es/comunicaciones/articulo.asp?id=161095

[5] ***Identificación biométrica, la llave del futuro***. Elvira Fernández. 2000. http://www.cienciadigital.es/hemeroteca/reportaje.php?id=83

[6] ***Biometría y la identificación automática de personas***. Herbert Saal Gutierrez. 08/01/07. http://capacitacionencostos.blogia.com/2007/010814-biometria-y-la-identificacion-automatica-de-personas.php

[7] ***Biometría e identificación de personas***. Biometría. Control de Acceso y Presencia. RFID. http://www.kimaldi.com/kimaldi/area_de_conocimiento

[8] ***Verificación Automática de Personas mediante Huella Dactilar***. Raúl Sánchez Reíllo.

[9] ***Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación***. María Teresa y Javier Areitio Bertolín. Revista española de electrónica, ISSN 0482-6396, Nº 630, 2007, págs. 52-67

[10] ***La Tecnología de las Tarjetas Inteligentes***. José Carlos Acedo Jiménez. David Cerezo Quesada. Xoan R. Rodríguez Lorenzo. Raúl Sánchez Reíllo.

[11] ***FxGate System. Introduction Manual***. Version 3.01

[12] ***FxGate SDK. Developers Manual***. Version 3.02

[13] ***FxLock user Manual***. Version 1.04. Firmware version 2.00 (September 2008)

[14] ***BioCard SDK Developer's manual***. Version 1.0.1

[15] ***Watchdata - Products - TimeCOS®/Calypso*** http://www.watchdata.com/products_detail.php?id=29&subid=17#features

[16] ***Atmel Products - Product Card AT88SC6416CRF*** http://www.atmel.com/dyn/products/product_card.asp?part_id=3107

[17] ***Tipos de Librerías en C y C++.*** http://www.zator.com/Cpp/E1_4_4b.htm

[18] ***Cómo: Realizar llamadas seguras para subprocesos en controles de formularios Windows Forms***. http://msdn.microsoft.com/es-es/library/ms171728.aspx

[19] ***Give Your .NET-based Application a Fast and Responsive UI with Multiple Threads***. Ian Griffiths http://msdn.microsoft.com/es-es/magazine/cc300429(en-us).aspx

[20] ***El lenguaje de programación C#. Tema 12: Delegados y eventos***. José Antonio González Seco. http://www.programacion.com/tutorial/csharp/13/

[21] ***CodeProject: Launching a process and displaying its standard output***. Mike Mayer http://www.codeproject.com/KB/threads/launchprocess.aspx

[22] ***CodeProject: Real-Time Console Output Redirection***. Olivier Marcoux http://www.codeproject.com/KB/threads/RTconsole.aspx

[23] ***How to: Create a C/C++ Union Using Attributes (C# Programming Guide)***. http://msdn.microsoft.com/en-us/library/acxa5b99(VS.80).aspx