# Desarrollo de una Interfaz Biométrica basada en la Lectura de Huellas Dactilares para Autenticación de Usuarios en un Cajero Automático.

Juan Francisco Salcedo Polanco, Escuela Politécnica Nacional (EPN), Quito – Ecuador

Paola Cecilia Sempértegui Jácome, Escuela Politécnica Nacional (EPN), Quito – Ecuador

Ing. Pablo Hidalgo Lascano, Escuela Politécnica Nacional (EPN), Quito – Ecuador

***Resumen* – Debido al incremento en los últimos años en el uso de los cajeros automáticos por la masificación de los servicios que estos prestan, y frente al grupo de población que no puede acceder a estos servicios debido a su índice de analfabetismo se presenta la implementación de una interfaz biométrica como alternativa a la autenticación tradicional (creando un sistema más seguro ya que la huella dactilar permite garantizar que la persona que desea realizar una transacción es quien dice ser); y como puente para el acceso a la tecnología a las personas analfabetas.**

***Abstract* - Due to the massive increment in the last few years over the use of ATMs (automatic teller machine) because of the amount of services which they provide and with the amount of unlettered people who can't access those services, we present a biometric interface as an alternative solution to traditional authentication (with a more secure and a guaranteed system based in fingerprint authentication) and as a bridge between technology and unlettered people.**

***Índices* – ATM, Autenticación, Biométrica, Cajero Automático, Huellas Dactilares.**

## I. Nomenclatura

- *ATM:* Automatic Teller Machine, Cajero Automático.
- *PIN:* Personal Identificaction Number, Número de Identificación Personal
- *UML:* Lenguaje de Modelado Unificado.
- *FMR:* Falso Positivo
- *FNMR:* Falso Negativo
- *EER:* Tasa de Igual Error

## II. Introducción

El proyecto estudia, diseña e implementa una interfaz que permite la autenticación de usuarios en cajeros automáticos a través de la lectura de sus huellas dactilares. Esta interfaz permite a un Usuario reemplazar el número de identificación personal *(Personal Identificaction Number - PIN)* por su huella dactilar, que junto al identificador de su tarjeta le permite acceder a los servicios de un cajero automático. De esta manera se garantiza que la persona que realiza una transacción es quien dice ser ya que los rasgos de las huellas dactilares de cada persona son únicos.

Dentro del proyecto se desarrolla toda la metodología del Lenguaje de Modelado Unificado - *Unified Model Language* (UML) para el diseño de la aplicación y se diseña la base de datos basada en un modelo relacional.

Para el manejo de huellas dactilares se utiliza el *Fingerprint SDK de Griaule* que es un software de gran aceptación en el mercado internacional y está en la lista del *top five* de software para manejo de huellas dactilares por el gran número de características que ofrece.

El lenguaje utilizado para la implementación de la aplicación es Visual Basic .NET en su edición 2005, y para la base de datos se emplea SQL Server 2005.

Para verificar la eficiencia operativa del prototipo implementado se desarrollan una serie de pruebas que permiten determinar la fiabilidad de la aplicación al autenticar usuarios mediante la huella dactilar y el nivel de aceptación por parte de los mismos.

## III. Diseño

Para la implementación del proyecto se desarrolló todo el procedimiento de diseño de software sugerido por la metodología de UML en el que se determinaron los siguientes requisitos para la aplicación que permitirá la autenticación de usuarios en un Cajero Automático mediante su huella dactilar:

R.1. Verificar la autenticidad de un usuario mediante su número de identificación y su huella dactilar en reemplazo del PIN.

R.2. La comunicación entre el cajero y el servidor debe emplear los protocolos IP y TCP. El control de los *sockets* lo debe realizar la aplicación, entendiéndose por *socket* a la definición de una dirección IP y el puerto TCP asociado a un servicio.

Este trabajo fue realizado con el apoyo de Multiservice y Multisoft.

R.3. Los mensajes que se transmiten entre el cajero y el servidor deben estar encriptados con una llave conocida por los dos módulos.
R.4. El servidor y el cajero deben manejar un registro de monitorización de uso de la aplicación.

R.4.1. El servidor registra el número de identificación del usuario, el puntaje de verificación, si el usuario fue o no aceptado y la dirección IP del cajero que envió el mensaje.

R.4.2. El cajero registra el número de identificación del usuario, el puntaje de verificación y si el usuario fue o no aceptado.

R.5. Solo personas debidamente autorizadas pueden enrolar clientes nuevos o actualizar datos de un cliente en el sistema a través de una interfaz gráfica.
R.6. La información que se debe ingresar por cada cliente es: número de identificación, nombre, lenguaje de preferencia, cooperativa a la que pertenece, dedo que ha sido registrado e identificación de la persona que enroló al cliente.
R.7. La aplicación manejará temporizadores de espera para cada una de las transacciones y estado del cajero. Estos temporizadores son:

R.7.1. Temporizador de espera de la huella dactilar después de ingresada la tarjeta.

R.7.2. Temporizador de espera para recibir la confirmación de autenticidad.

R.8. El software permitirá generar reportes del estado de la aplicación y de históricos de uso. Estos reportes son:

R.8.1. Número de clientes aceptados o no por la aplicación, filtrado por fecha.

R.8.2. Número de identificación de los clientes que han sido rechazados por el sistema filtrado por fecha.

R.8.3. Identificación de los cajeros que se han conectado al servidor y que han realizado alguna transacción

R.8.4. Número de clientes nuevos que han sido enrolados, filtrados por fecha.

Adicionalmente se enumeran y describen los distintos seis casos de uso que presenta la aplicación, se identifican los objetos que participan, y se ilustran y detallan los diagramas de secuencia, colaboración, estados, actividad, estáticos y de implementación, que permiten tener un esquema completo de la estructura de la aplicación y determinar que la misma requiere de tres sistemas o módulos: Host Cajero, Host Administrador y Host Servidor. Los módulos están representados en la Fig. 1.

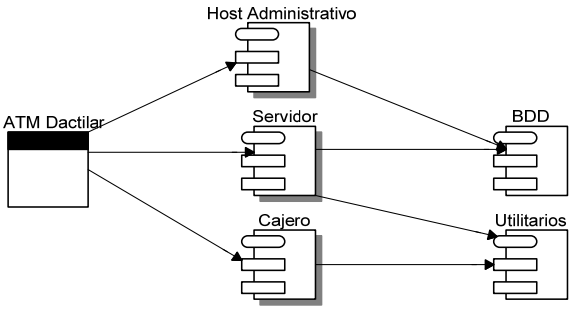


Fig. 1. Módulos de la aplicación.

A partir de los diagramas indicados anteriormente, principalmente de los diagramas estáticos, se diseña la Base de Datos; determinándose los siguientes requerimientos de datos:

- Usuario Administrativo:
  - Número de Identificación: Alfabético
  - Nombre: Alfabético
  - Identificación de la Cooperativa: Alfabético
  - Huella: Huella
  - Contraseña: Numérico (Entero)
  - Identificación de Rol: Numérico (Entero)

- Cliente:
  - Número de Identificación: Alfabético
  - Nombre: Alfabético
  - Identificación de la Cooperativa: Alfabético
  - Huella: Huella
  - Idioma: Alfabético
  - Usuario Administrativo que realizó registro del cliente: Alfabético

- Rol:
  - Nombre: Alfabético
  - Descripción: Alfabético

- Huella:
  - Plantilla: Datos binarios
  - Tamaño: Numérico (Entero)
  - Dedo: Numérico (Entero)
  - Usuario: Alfabético

- Cooperativa:
  - Número de Identificación: Alfabético
  - Nombre: Alfabético

- Cajero:
  - Número de Identificación: Alfabético
  - Ubicación: Alfabético
  - Sistema: Alfabético
  - Cooperativa: Alfabético

- Log:
  - Usuario: Alfabético
  - Cajero: Alfabético
  - Estado: Alfabético
  - Fecha: Tiempo

En función de los requerimientos se diseña la base de datos modelada en la Fig. 2.

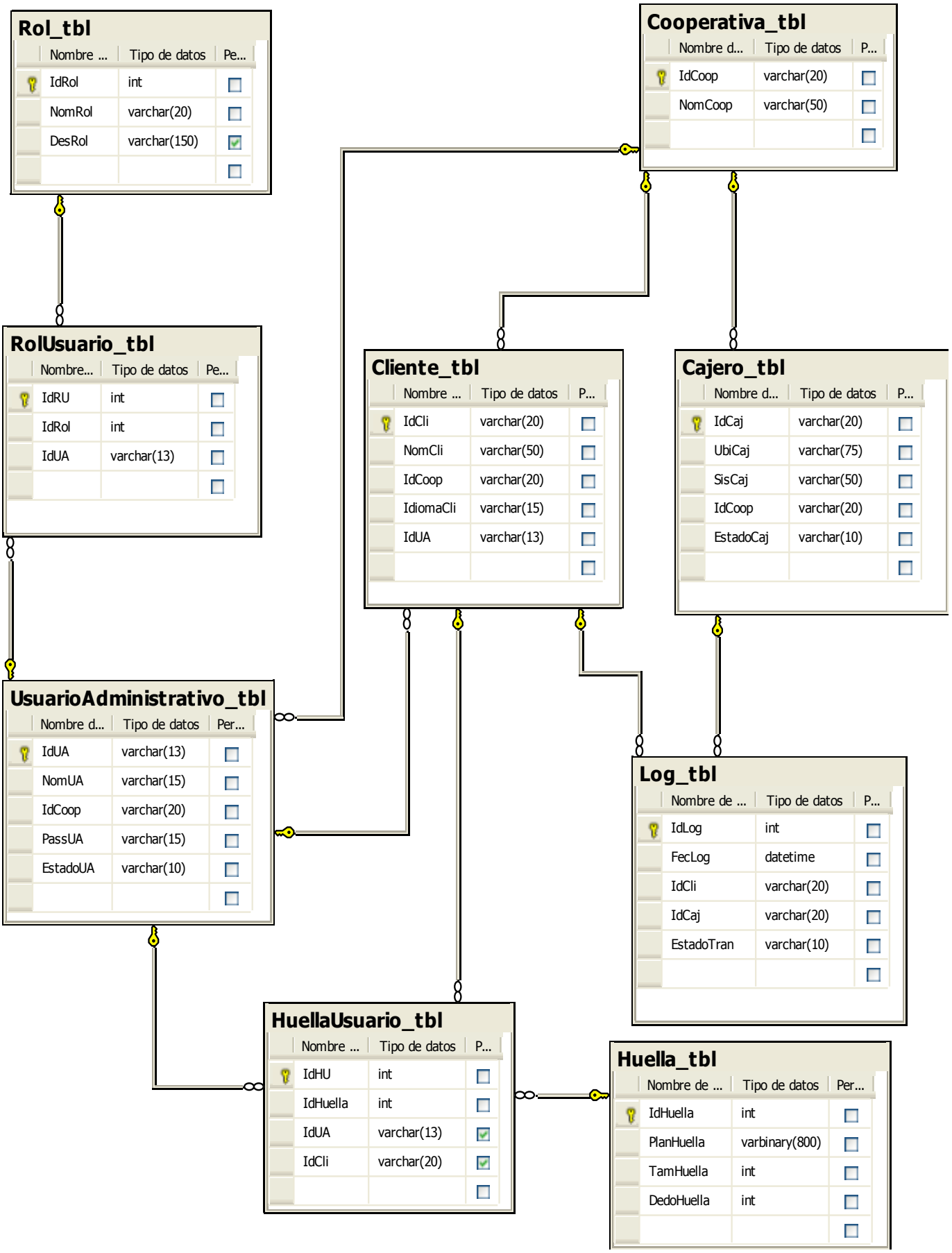


Fig. 2. Modelo de Base de datos

## IV. IMPLEMENTACIÓN

Un cajero automático representa un terminal de datos conectado a un Host Central, al que pueden estar conectados varios cajeros. El Host Central se encarga del procesamiento de los mensajes provenientes del cajero y retorna mensajes con respuestas a las solicitudes o con información de administración. Las transacciones no son más que el intercambio de mensajes entre el Host Central y el cajero; y son ejemplificadas en el diagrama de la Fig. 3. El Host Central se encuentra a su vez conectado a un servidor de aplicación y de servicios; este servidor es el encargado de ofrecer y procesar todos los servicios que se tienen en el cajero automático.

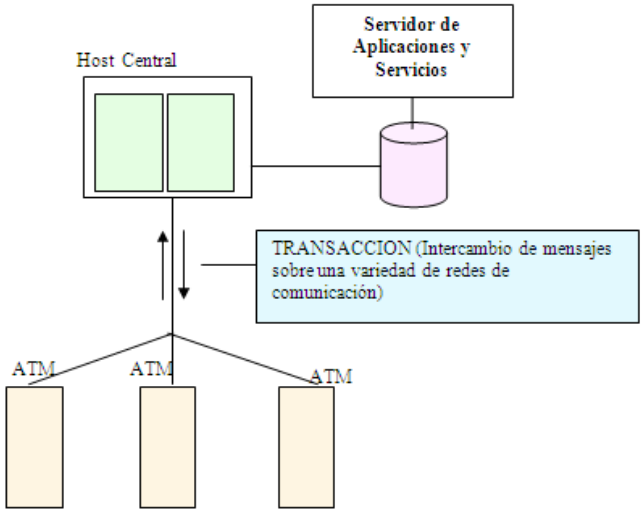


Fig. 3. Esquema de Funcionamiento de los Cajeros Automáticos

Entendiendo la estructura de funcionamiento de los cajeros automáticos, el prototipo se diseña en una arquitectura de tres capas en donde se identifican tres componentes: la interfaz del usuario, el procesamiento de la lógica de la aplicación y la gestión de la información. Estos tres componentes se distribuyen en tres capas, la primera de ellas tiene al usuario en sí y se encarga de manejar la lógica de la presentación; la segunda capa incluye un servidor intermedio que maneja la lógica de la aplicación, y la tercera capa maneja el procesamiento de la información como se puede apreciar en la Fig. 4.

Mediante el uso del servidor intermedio se consigue tener un *middleware*[1] capaz de permitir múltiples conexiones con clientes; ofrece flexibilidad, escalabilidad y seguridad al sistema, incrementa el rendimiento, y esconde la complejidad del procesamiento al usuario.

Adicionalmente, el uso del *middleware* permite aligerar la carga al servidor de base de datos y sirve como un punto de protección de la información.

[1] *Middleware*: Módulo (software generalmente) intermedio que ofrece un punto de acceso transparente para la comunicación de dos procesos, adaptando la información y brindando seguridad.

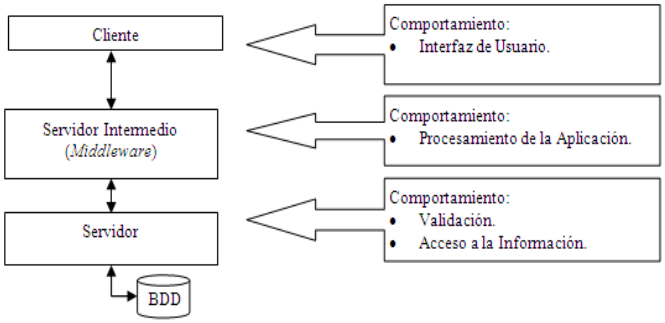


Fig. 4. Arquitectura de 3 Capas

Una implementación de la arquitectura de tres capas implica una complejidad mayor puesto que se tienen que controlar un mayor número de situaciones y elementos, pero brinda un mayor número de beneficios que una arquitectura de dos capas (la misma que define en una capa al cliente y la gestión del procesamiento, y en la otra capa se encuentra la gestión de la base de datos).

El lenguaje que se utiliza para el desarrollo de la aplicación es Visual Basic .NET en su versión Visual Basic 2005 por las características y facilidades que ofrece. Se diseña una aplicación para cada sistema, y dentro de cada aplicación existe uno o más espacios de nombres que permiten una agrupación lógica de las clases. Para el manejo de las huellas dactilares se utiliza el SDK de Griaule Biometrics conocido como Fingerprint SDK en su versión trial de 90 días. Entre los lectores soportados por este SDK se selecciona el Nitgen Hamster por las características de lectura rápida de la imagen de la huella.

Para la comunicación TCP/IP entre el cajero y el servidor, se utilizan *threads* para que se encarguen del control de la comunicación. Los mensajes que se envían incluye la plantilla de la huella dactilar que es la representación numérica de los vectores que la forman y la distancia entre los mismos. Para cada uno de estos mensajes se aplica un proceso de compresión mediante el cambio de base decimal a base36 logrando comprimir hasta un cuarenta por ciento el mensaje original (ya que al cambio de base se agregan caracteres de control y relleno para su correcta descompresión). Después de este proceso se aplica un mecanismo de encripción basado en el algoritmo de 3DES para garantizar la confidencialidad de la información que se transmite.

Para el manejo de los datos, .NET Framework ofrece los *datasets* que corresponden a espacios de memoria donde se almacenan temporalmente los datos consultados a la base de datos a través de llamadas a los procedimientos almacenados.

## V. APLICACIONES

A continuación se describen las aplicaciones para cada uno de los sistemas.

### A. *HATMDactilar*

Es instalada en el Cajero Automático el mismo que debe mantener una comunicación con el Servidor. Despliega una ventana que contiene el listado de todos los eventos generados desde el arranque de la aplicación los mismos que son almacenados en un archivo de log.

HATMDactilar lee la información un archivo denominado *FingerprintRequest* (ver Fig. 5) cada 200 milisegundos la misma que le permite determinar si un nuevo usuario ha ingresado su tarjeta y requiere autenticación mediante su huella dactilar. Este archivo contiene el número de identificación del Usuario y una vez que ha ingresado su tarjeta, HATMDactilar espera por la lectura de la huella del Usuario.

Después de capturada la huella del usuario se construye un mensaje que contiene el número de identificación, la información de la plantilla de la huella del usuario y la plantilla (Fig. 6), y es enviado al servidor para ser autenticado. La aplicación espera por la respuesta del servidor un tiempo determinado y en otro archivo denominado *FingerprintResponse* (ver Fig. 7) escribe la respuesta y pasa el control a la aplicación propia del cajero para que continúe su operación normal y pase al estado correspondiente, en función de la respuesta leída del archivo *FingerprintResponse*.

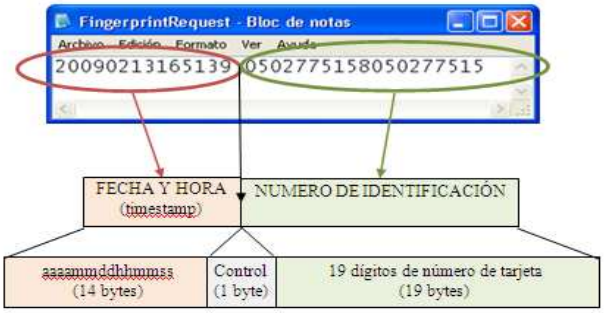


Fig. 5: Archivo FingerprintRequest


Fig. 6: Formato mensaje cliente

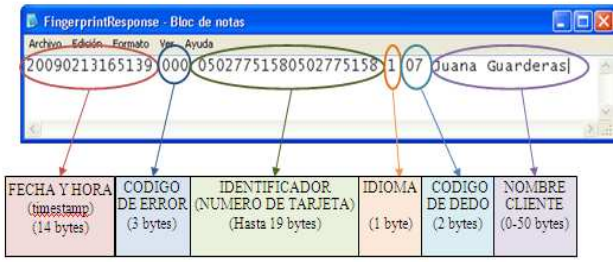


Fig. 7: Archivo FingerprintResponse

### B. *SATMDactilar*

Corresponde a la aplicación que corre en el servidor de autenticación.

Al iniciarla, se abre la ventana correspondiente al formulario principal de la aplicación, y el *thread* que administra el *socket* que espera las conexiones de clientes arranca y solo se cierra una vez que la aplicación es cerrada. El formulario principal contiene un subformulario en el que se pueden observar todos los eventos que se han generado mientras se ejecuta la aplicación (Fig. 8) y además se guardan en un archivo log.

Una vez que llega un mensaje proveniente de un cajero, se crea un *thread* que corresponde a un nuevo *socket* para la comunicación entre el nuevo cliente y el servidor, el mismo que se cierra cuando finaliza la comunicación entre ambos o una vez vencido el temporizador de espera por un mensaje de respuesta.

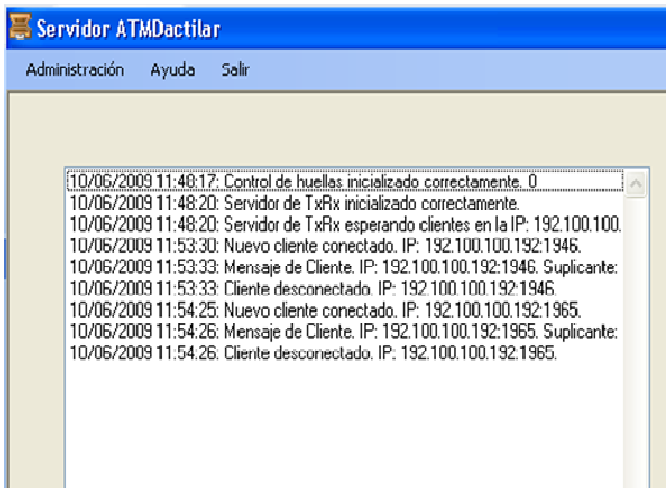


Fig. 8: Interfaz gráfica SATMDactilar

SATMDactilar procesa el mensaje y mediante el número de identificación del usuario consulta a la base de datos su información para verificar la identidad del mismo.

La verificación se realiza mediante una comparación entre la huella obtenida del usuario y la extraída de la base de datos en base a un puntaje mínimo de aceptación. Si el resultado de la comparación supera el puntaje mínimo, un usuario es autenticado caso contrario es rechazado y se envía el mensaje correspondiente al cajero.

El puntaje representa el grado de coincidencia hallado entre la plantilla a verificar y la plantilla original. El puntaje de referencia se selecciona en base a la tasa de falsa coincidencia (FMR o falso positivo) y a la tasa de falsa no coincidencia (FNMR o falso negativo). Estos dos valores forman una curva y el punto en donde son iguales se le conoce como Tasa de Igual Error (EER por sus siglas en inglés), siendo éste el valor referencial en el cual la probabilidad de tener un falso positivo es igual a la de tener un falso negativo. Estas curvas se pueden apreciar en la Fig. 9.

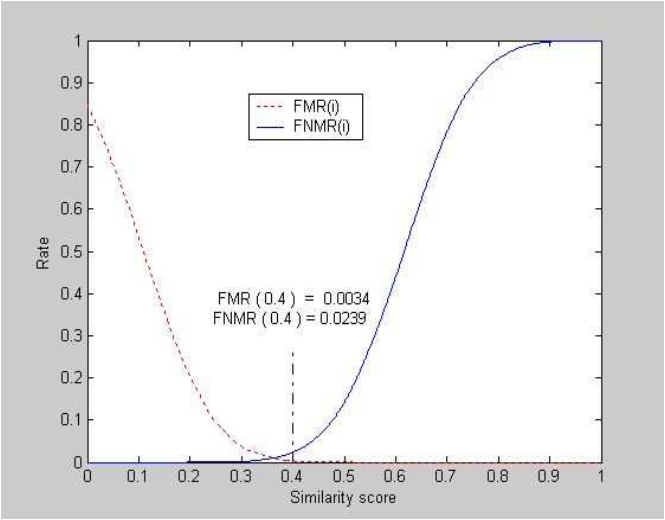

Fig. 9: Curvas de FMR y FNMR

La comunicación entre SATMDactilar y HATMDactilar es a través de *sockets* TCP, uno para cada cliente conectado (Fig. 10). Los puertos en SATMDactilar asignados para cada nueva conexión con un cliente son seleccionados aleatoriamente. Además puede soportar la comunicación de varios clientes simultáneamente.

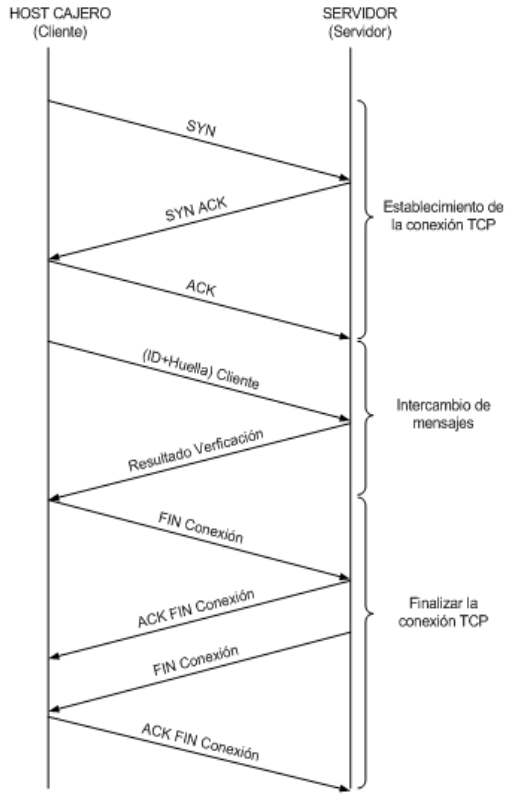

Fig. 10: Comunicación Host Cajero – Servidor

### *C. AATMDactilar*

Corresponde a la aplicación en el Host Administrador, y permite a personas autorizadas el ingreso de nuevos usuarios, la actualización de su información, el ingreso de usuarios administrativos, su actualización, el ingreso de cajeros automáticos, entre otras funciones que pueden ser ejecutadas por un usuarios dependiendo de los roles asignados a estos.
Dentro del proceso de implementación se dimensiona la capacidad de los servidores de base de datos y aplicación en base al comportamiento de cada sistema.

## VI. DIMENSIONAMIENTO DEL SERVIDOR DE APLICACIÓN

Se analiza el tamaño de los mensajes intercambiados entre las aplicaciones SATMDactilar y HATMDactilar y los tiempos de respuesta del servidor ante las peticiones de autenticación de clientes, tanto de manera secuencial como simultánea. Las características de los Host Servidor y Cajero se ilustran en la tabla 1 y las características del servidor recomendado se indican en la tabla 2.

TABLA 1: CARACTERÍSTICAS HOST CAJERO Y HOST SERVIDOR

| DETALLE | | SERVIDOR | HOST CAJERO |
|---|---|---|---|
| Sistema Operativo | | Windows XP Professional | Windows XP Professional |
| Versión | | 2002 | 2002 |
| Procesador | Fabricante | Intel Core 2 | Intel Pentium 4 |
| | Frecuencia | 1.86 GHz | 1.8 GHz |
| | Número de Núcleos: | 2 | 1 |
| RAM | Tamaño | 1 GB | 512 MB |
| Red | Adaptador | Intel PRO/100 VE | VIA PCI 10/100 Mbps |
| | Tipo de Conexión | 100BaseT | 100BaseT |

TABLA 2: CARACTERÍSTICAS DEL SERVIDOR RECOMENDADO PARA LA APLICACIÓN

| | |
|---|---|
| Tipo de Procesador | Pentium IV en adelante |
| Velocidad de Procesador | 600 MHz en adelante |
| Memoria RAM | 512 MB |
| Disco Duro | Arquitectura: DAS<br>Protocolo: SCSI o SATA |
| Sistema Operativo | Windows Server 2003 |
| Tarjeta de Red | Interfaz RJ45 para conexión 100BaseT |

## VII. DIMENSIONAMIENTO DEL SERVIDOR BASE DE DATOS

Para el servidor de base de datos se toman en cuenta dos parámetros, los requerimientos mínimos establecidos por SQL Server 2005 para su instalación y operación, y los determinados en base a la cantidad de información a almacenar y procesar.
Para determinar la cantidad de información procesada se siguen las recomendaciones de la ayuda de Microsoft SQL Server, que permiten determinar de manera aproximada el tamaño que podría alcanzar la base de datos tomando en cuenta que las tablas en la base tienen índices agrupados (índices agrupados con los datos). Los resultados obtenidos para cada tabla se pueden observar en la Tabla 3.

TABLA 3: ESPACIO REQUERIDO APROXIMADO PARA CADA TABLA DE LA BASE DE DATOS

| Nombre Tabla | Tamaño Tabla (bytes) |
|---|---|
| Cajero_tbl | 24576 |
| Cliente_tbl | 3497984 |
| Cooperativa_tbl | 16384 |
| Huella_tbl | 33202176 |
| HuellaUsuario_tbl | 1392640 |
| Log_tbl | 445087744 |
| Rol_tbl | 16384 |
| RolUsuario_tbl | 16384 |
| UsuarioAdministrativo_tbl | 24576 |
| **Tamaño Base de Datos (bytes) [26]** | **483278848** |
| **Tamaño Base de Datos (MB)** | **461** |

Tomando en cuenta el resultado anterior y los requerimientos de SQL, en la Tabla 4 se indican las características del servidor de base de datos recomendado.

TABLA 4: CARACTERÍSTICAS SERVIDOR RECOMENDADO PARA BASE DE DATOS

| | |
|---|---|
| **Tipo de Procesador** | Pentium IV en adelante |
| **Velocidad de Procesador** | 1 GHz en adelante |
| **Memoria RAM** | 512 MB |
| **Disco Duro** | Arquitectura: DAS<br>Protocolo: SCSI<br>Tamaño: 10 GB en adelante |
| **Sistema Operativo** | Windows XP Home Edition<br>Windows XP Professional Edition<br>Windows Server 2003 Web Edition |

## VIII. PRUEBAS DE CAMPO

Se realizan dos tipos de pruebas, las de confiabilidad de la aplicación y las del tiempo de respuesta de la aplicación frente a una solicitud, ambas pruebas se las realiza a nivel de laboratorio bajo dos escenarios; en el primer escenario se prueba la aplicación desarrollada en un computador en donde se emula el cajero automático mientras que en el segundo escenario se prueba la aplicación ya instalada en un cajero automático.

Dentro de los dos escenarios anteriores se miden cuatro factores:

- *Tiempo de respuesta (Prueba 1):* Desde que la aplicación Host Cajero captura la huella dactilar, hasta que recibe una respuesta por parte del servidor.
- *Tiempo de respuesta con carga en el servidor (Prueba 2):* Se utiliza el software Load Runner de HP para visualizar el desempeño de una aplicación simulando tener varios clientes conectados simultáneamente y poder analizar así los posibles "cuellos de botella" en los servidores, en la red o en la aplicación.
- *Confiabilidad de la aplicación (Prueba 3):* En base al número de falsos positivos y negativos que determina la aplicación.
- *Nivel de aceptación del usuario (Prueba 4):* Este factor es medido frente a la aceptación que tiene el sistema por parte de las personas que probaron el mismo.

Las pruebas fueron realizadas en un cajero marca NCR que mantenía una conexión al servidor mediante una LAN. Este cajero automático tiene todos los dispositivos en el cajero y el dinero dispensado era dinero de prueba así como las opciones del cajero conFig.das sólo permiten el retiro del bono. En la Fig. 11 se tiene el cajero de pruebas.

Fig. 11: Cajero empleado para pruebas

## IX. ANÁLSIS DE RESULTADOS

- *Prueba1 y Prueba2:* De los análisis realizados se aprecia que la diferencia de tiempo entre el uso del cajero automático y un computador es de 3.7 ms en condiciones normales y cuando se realizan múltiples conexiones hacia el servidor la diferencia de tiempo entre tener una sola transacción y tener varias es de 0.15 ms. Por ser valores de tiempo muy pequeños las diferencias son imperceptibles para el usuario y bastante difíciles de medir.
- *Prueba 3:* De los usuarios entrevistados, se puede apreciar que las respuestas demuestran que la aplicación tiene una gran aceptación y se evidencia también que su uso es bastante intuitivo para los nuevos usuarios. De lo conversado con las personas que usaron la aplicación, reconocen que la misma es más segura que el uso de una clave (96.5 %). Pero manifiestan su miedo frente al hecho que les puedan cortar el dedo o la mano para poder retirar dinero, por eso es que el 3.5% de las personas prefieren usar un cajero con clave (el 3.5% es la diferencia entre las personas que lo consideran más seguro al cajero con huellas dactilares pero que no estarían dispuestos a usarlo).

- *Prueba 4:* El 2.3% de los intentos de autenticación de una huella válida en el sistema falló, este valor representa los falsos negativos que presenta el sistema en un escenario en el cual ya se tiene instalado el cajero y conectado al servidor. En la práctica, el porcentaje de errores cometidos por el sistema es menor al porcentaje de transacciones rechazadas en los cajeros automáticos por error en el ingreso de la clave. El 100% de las huellas que no debían ser autenticadas fueron rechazadas por el sistema con lo que se comprueba que la aplicación es robusta frente a los falsos positivos dando la garantía de ser un sistema confiable frente a ataques de violación de identidad.

Además de las pruebas indicadas anteriormente, se realizaron pruebas para comprobar el grado de seguridad de la aplicación determinando a través de un *sniffer* si la trama que viaja del cajero al servidor realmente tenía encriptada la huella dactilar; además se revisó si la trama que viaja con la respuesta del servidor es susceptible de algún ataque. El *sniffer* que se empleó para esta prueba es el *Wireshark* y los resultados se indican en Fig. 12.

De la trama que revela el *sniffer* en la Fig. 12 se puede apreciar que la huella se encuentra completamente encriptada; esto se evidencia ya que si la huella estuviera en texto plano en la parte derecha de la captura se vería el texto plano con información de la trama, es decir con el número de identificación y la huella. La Fig. 13 muestra la trama capturada. Del mismo modo la trama de respuesta del servidor viaja encriptada hacia el host ya que se tiene la información del proceso de autenticación del cliente.

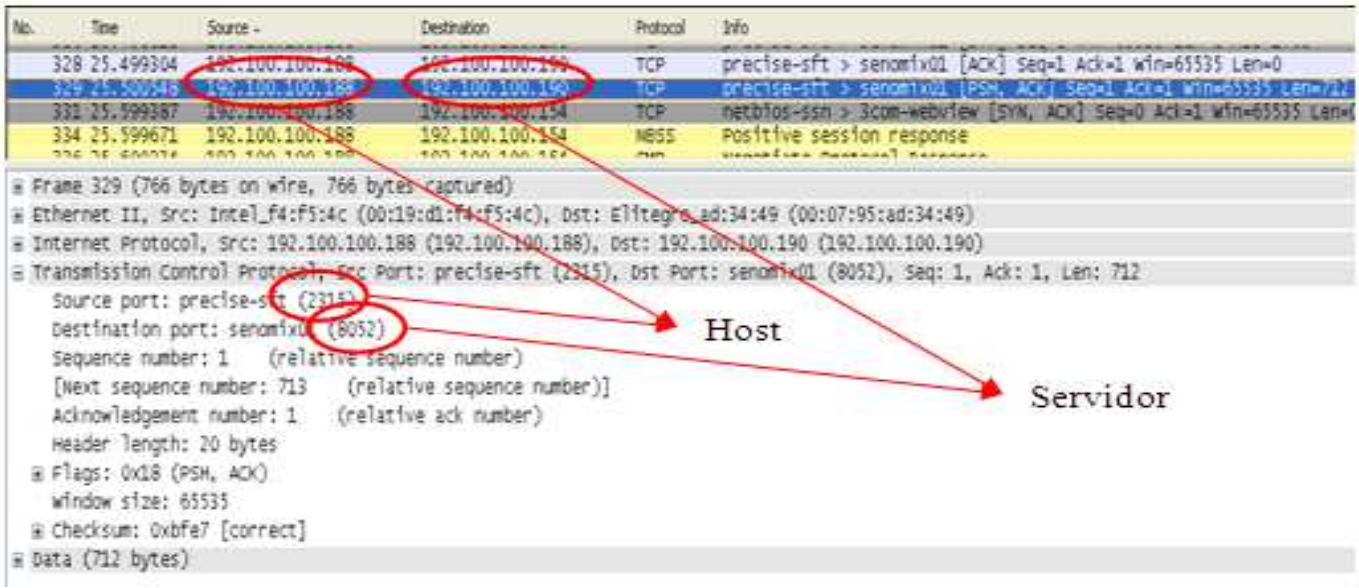


Fig. 12: Trama del host hacia el servidor capturada por Wireshark.

## X. CONCLUSIONES

- La confiabilidad del sistema frente a ataques de identidad es alto; es decir que el 100% de las huellas no válidas son rechazadas. Se aprecia que el porcentaje de huellas válidas no aceptadas por el sistema es comparable al porcentaje de rechazo de un cajero automático por un error en el PIN.
- El sistema desarrollado permite tener mayor seguridad a nivel de cajeros automáticos, ya que la persona tiene que estar físicamente en el sitio para validar su identidad. En caso de robo de una tarjeta ésta es inútil (en donde el ladrón de alguna forma consigue la clave) ya que no se tiene la huella de la persona con lo que no se puede hacer ninguna transacción.
- La metodología de diseño de aplicaciones mediante UML permiten modelar un sistema y tener un mapa de la aplicación útil en todas las etapas de la vida del software. En la etapa de diseño permite tener en cuenta todos los aspectos que se deben diseñar y prever, en la etapa de implementación ya que permite tener una guía para ir creando los diferentes módulos y en la etapa de operación ya que bajo cualquier eventualidad se tiene la guía y el mapa de la aplicación para determinar dónde puede estar el posible punto de falla o la guía para modificar la aplicación en caso de tener un nuevo requerimiento.
- Por los rasgos fisiológicos de nuestra gente y los patrones de huellas que son bastante marcados, sistemas de autenticación biométricas basadas en huellas dactilares funcionan bastante bien, siendo el lector más recomendado el basado en lectura óptica.
- Aunque la comunicación entre el host y el servidor (o servidores) se considera confiable ya sea por la seguridad de la red LAN o por el uso de una VPN, para aumentar la seguridad y garantizarla en el sistema desarrollado se emplea una encripción 3DES, elevando aún así la seguridad total del sistema.

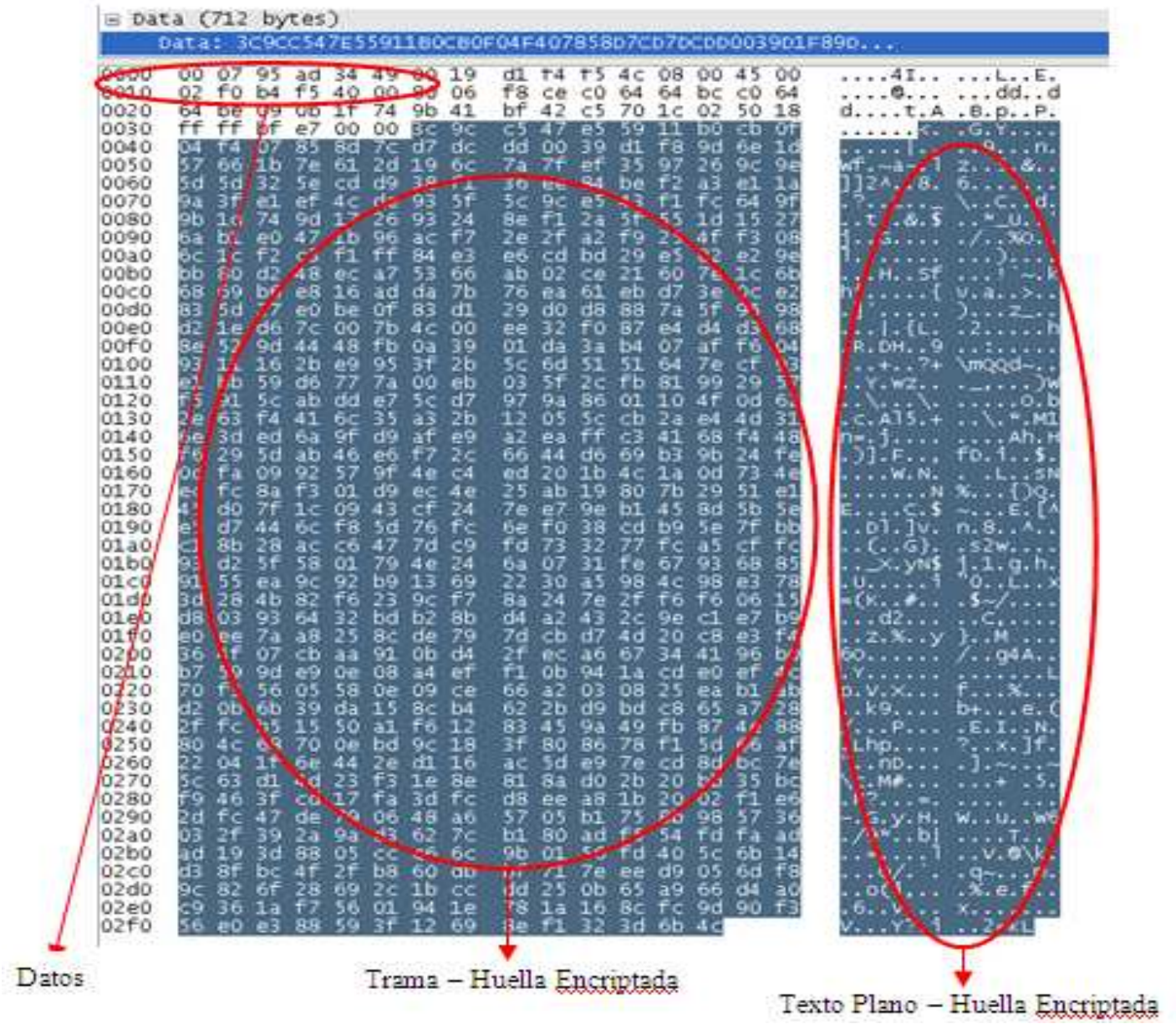


Fig. 13: Huella dactilar encriptada.

## XI. RECOMENDACIONES

- Aunque las pruebas realizadas en laboratorio dieron resultados positivos, antes de comercializar el producto o ponerlo en producción, sería recomendable realizar pruebas en un ambiente más agresivo como tener un cajero en un lugar remoto para poder analizar los retardos de tiempo que se tienen por motivos de propagación.
- La capacitación del sistema, tanto para los usuarios administrativos así como para los clientes es muy importante previa la puesta en marcha, ya que sólo así se puede garantizar un correcto uso del mismo y en posterior el éxito del sistema.
- Hasta conseguir la difusión del sistema, es recomendable no forzar a los clientes a usarlo sino más bien manejarlo como un proyecto piloto y pedir voluntarios para su uso. Esto básicamente para eliminar los miedos de las personas frente su uso y también eliminar la asociación de las huellas dactilares a hechos legales o forenses.
- En caso que se implemente el proyecto para el cobro del bono de desarrollo, se puede implementar ayudas adicionales para tener una mayor penetración. Por ejemplo, se puede implementar ayudas auditivas en el idioma de preferencia de la persona (el idioma es uno de los valores que devuelve el servidor en su trama de respuesta) o se puede modificar los teclados para que tengas colores y sean de fácil distinción para las personas analfabetas.
- La base de datos y el servidor de la aplicación se deben mantener en lugares altamente seguros, ya que poseen información susceptible y privada. Esta información es

valiosa y podría comprometer la seguridad de la persona.

- Por el diseño con el cual se implementó la aplicación, su integración con otras aplicaciones es bastante sencillo, por lo que se recomienda que se continúe su investigación para el desarrollo de aplicaciones afines como autenticación de usuarios en ventanillas.

## XII. AGRADECIMIENTOS

Nuestro sincero agradecimiento al Ing. Pablo Hidalgo que fue nuestro guía durante el desarrollo de este proyecto. A nuestros tribunal para la defensa del proyecto de titulación Ing. Rodrigo Chancusig y MSc. Xavier Calderón por sus aportes y al MSc. Tarquino Sánchez por su confianza para la realización del proyecto.
Gracias a nuestros amigos Nicolás Martínez, Alberto Andrade, María Eugenia Andrade y Frank Koch por su apoyo constante e incondicional.

## XIII. REFERENCIAS

[1] Salcedo J.; Sempértegui P., *"Desarrollo de una interfaz biométrica basada en la lectura de huellas dactilares para autenticación de usuarios en un cajero automático"*, Proyecto de Titulación, Escuela Politécnica Nacional, Quito, Ecuador. Agosto, 2009.
[2] Griaule Biometrics (2008), *"Understanding Biometrics"*. [Online]. Disponible: http://www.griaulebiometrics.com/page/en-us/book
[3] IBM (año). *"Nombre del Archivo que se utilizó"*. [Online]. Disponible: http://www.redbooks.ibm.com.
[4] Microsoft Tech Net Libary. "Cómo estimar el tamaño de una base de datos". [Online]. Disponible: http://technet.microsoft.com/es-es/library/ms187445.aspx
[5] AMERICAN NATIONAL STANDARD FOR INFORMATION SYSTEMS - Data Format for the Interchange of Fingerprint, Facial, & Scar Mark & Tatto. (SMT) Information. NIST Special Publication 500 – 245. ANSI/NIST-ITL 1-2000. Revision of ANSI/NIST-CSL 1-1993 & ANSI/NIST-ITL 1a-1997. [Online]. Disponible: www.nist.gov

## XIV. BIOGRAFÍAS

**Juan Francisco Salcedo Polanco**, Autor. Nació en Quito Ecuador el 31 de Julio de 1985. Sus estudios primarios los realizó en la escuela Antonio Neumane y la secundaria en el colegio Marista Pío XII de la ciudad de Santo Domingo. A los 17 años ingresó en la EPN para realizar sus estudios de pregrado en la carrera de Electrónica y Redes de Información; los mismos que fueron culminados en febrero del 2008. Miembro IEEE. Obtuvo el título de Ingeniero en Electrónica y Redes de Información en Octubre del año 2009 siendo el mejor egresado de su promoción. Actualmente, trabaja en Telefónica Ecuador en el cargo de Ingeniero de Gestión de Red.
[email: pancho_salcedo85@hotmail.com]

**Paola Cecilia Sempértegui Jácome**, Autor. Nació el 20 de noviembre de 1984 en Quito – Ecuador. Su instrucción primaria y secundaria la realizó en el Colegio de América, lugar donde obtuvo el título de Bachiller en Ciencias en la especialización de Físico Matemáticas. Sus estudios de pregrado fueron culminados en la Escuela Politécnica Nacional obteniendo el título de Ingeniero en Electrónica y Redes de Información en Octubre del 2009. Actualmente, trabaja como Soporte IT en la empresa Businesswise.
[email: psempertegui@hotmail.es]

**Pablo Hidalgo Lascano**, Director del Proyecto. Ingeniero en Electrónica y Telecomunicaciones de la Escuela Politécnica Nacional. Profesor principal de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional.
[email: phidalgo@ieee.org]