

# Aplicación móvil de telemedicina para pacientes hipoglucémicos y diabéticos

M.I. Camargo A. Leonardo<sup>1</sup> | Ing. Sepúlveda M. Sergio<sup>2</sup> | Ing. Castro C. Sergio<sup>3</sup>

Recibido:  
Junio 6 de 2010

Aceptado:  
Octubre 15 de 2010

## Resumen

Se desarrolló un proyecto de telecomunicaciones para la supervisión médica remota de pacientes hipoglucémicos y diabéticos; el prototipo permite a los médicos monitorear a los pacientes con el fin de mantener el control de su salud. El sistema completo consiste de una aplicación móvil instalada en el dispositivo de cada paciente; una base de datos relacional que almacena la información; y una aplicación servidor que utilizan los médicos para diagnosticar con base a la condición de sus pacientes. Para el desarrollo del proyecto se diseñó e implementó la metodología DAM (Desarrollo de Aplicaciones Móviles); siguiendo ésta, se obtuvo como resultado el correcto funcionamiento integrado de cada uno de los elementos del sistema y una evaluación satisfactoria del potencial de éxito de la aplicación móvil.

**Palabras clave:** DBMS, glucosa, Java 2 Micro Edition, SMS.

## Abstract

We developed a telecommunication project for the medical remote supervision of patients while having hypoglycemia or diabetes. The prototype allows physicians to supervise the patients in order to keep control of their health. The whole system consists of a mobile application installed in every patient device; a relational database that saves the information; and a server application which physician uses to take measures about their condition. We designed and implemented MAD (Mobile Applications Development) methodology in order to accomplish this project. As a result, we obtained the integrated functioning of each one of the components and a satisfactory assessment of the mobile application potential to be successful.

**Keywords:** DBMS, glucose, Java 2 Micro Edition, SMS.

<sup>1</sup>Universidad de Magdalena  
Santa Marta - Colombia.  
lcamargo@unimagdalena.edu.co

<sup>2</sup>University of Delaware  
Newark, Delaware - Estados Unidos.  
sergiosm@udel.edu

<sup>3</sup>Universidad Francisco de Paula  
Santander Cúcuta - Colombia.  
sergio.castroc@ufps.edu.co

## I. Introducción

Las Tecnologías de la Información y la Comunicación (TIC), están a disposición de la asistencia médica, trasladando los servicios asistenciales al domicilio del paciente, contribuyendo a un nuevo modelo de salud sostenible y eficiente con el paciente como eje central [1], masificando el término e-Salud (e-Health).

Algunas enfermedades, como la diabetes y la hipoglucemia, requieren que se haga un control periódico de algunas medidas corporales; normalmente las personas que las padecen deben dirigirse a un centro asistencial para conocer su estado y llevar este control. Aunque existen dispositivos para hacer este monitoreo de forma sencilla por el mismo paciente, no hay un sistema que permita centralizar y procesar los datos de diferentes medidas para varias personas, razón por la cual se propone la implementación de un nuevo servicio de e-Salud basado en los beneficios de la tecnología móvil celular “Tiempo, Movilidad y Ubicuidad”.

Para esto se diseñó e implementó un sistema para la supervisión de medidas corporales del ser humano a través de una aplicación móvil. El prototipo llamado Doctor Móvil permite, a los pacientes hipoglucémicos y diabéticos, llevar un control detallado de su estado de salud y de su evolución, utilizando el teléfono móvil celular del paciente. Además, permite enviar los datos de los pacientes desde el teléfono hacia un servidor central donde se almacena toda la información en una base de datos. De esta forma, el médico encargado puede tener acceso a la evolución histórica del paciente y enviarle cualquier clase de indicación y/o sugerencia a través la red del operador de telefonía móvil celular (sistema bidireccional).

Este documento presenta la metodología propuesta utilizada para el desarrollo de la aplicación, los resultados obtenidos describiendo cada componente del sistema,

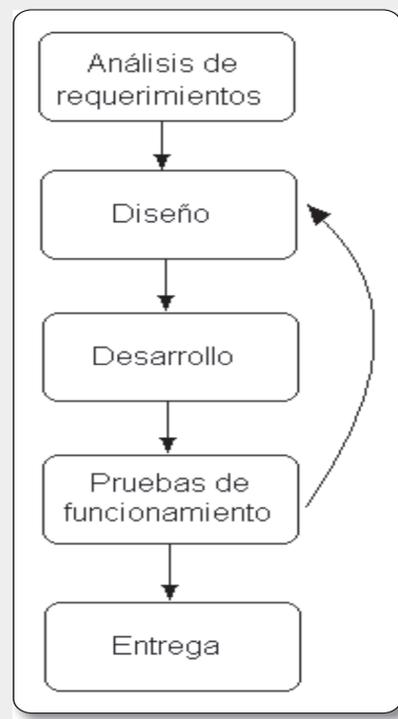
la evaluación practicada al mismo y las conclusiones finales del trabajo realizado.

## II. Metodología

Se propone la “metodología para el Desarrollo de Aplicaciones Móviles (DAM)”. Está basada en “Ingeniería de Software Educativo con modelaje Orientado por Objetos (ISE-OO)” [2] y en la experiencia de investigaciones previas en aplicaciones móviles; además, está fundamentada en la ingeniería de software tradicional [3]. Esta propuesta se encuentra enmarcada en cinco etapas como se muestra en la figura 1.

Algunas de las metodologías empleadas como referencia y soporte son Object Oriented Design[4], Objectory[5] y Object Modeling Technique[6]. A continuación se describe cada una de las etapas que intervienen en el desarrollo de la propuesta.

Figura 1. Etapas de la metodología para el Desarrollo de Aplicaciones Móviles (DAM)



**Análisis de Requerimientos:** en esta etapa se lleva a cabo una recopilación de las peticiones del cliente. Cliente se refiere a la persona o

entidad para la cual se desarrolla la aplicación móvil. Se deben seguir básicamente dos pasos: obtener información del cliente (usuario) y clasificar los requerimientos.

- **Obtener información de cliente.** En este paso se hace una serie de entrevistas en las cuales el cliente le manifiesta al desarrollador las características que debe tener la aplicación.

- **Clasificación de requerimientos.** Una vez se conocen los requerimientos que debe tener el software, se procede a clasificarlos. Dichos requerimientos se pueden clasificar en entorno, interfaz, funcionales, no funcionales y de entrenamiento (manuales).

- **Entorno.** Se refiere a todo lo que rodea al sistema. Por ejemplo, el sistema operativo subyacente, el sistema manejador de base de datos (si se requiere), el formato de archivos, etcétera.

- **Interfaz.** Es la forma cómo interactúan el usuario y la aplicación. Aquí se encuentran los requerimientos de la Interfaz Gráfica de Usuario (IGU), la forma en que el software va a generar los datos de salida, el formato de estos datos, y los demás requerimientos que involucren la comunicación hombre – máquina.

- **Funcionales.** Estos requerimientos son todos aquellos que demandan una función dentro del sistema. Se debe definir claramente cada una de las tareas que debe hacer el software.

- **No funcionales.** Estos requerimientos son la estabilidad, la portabilidad, el rendimiento, el costo, entre otros.

- **Entrenamiento (manuales).** Una aplicación móvil debe constar de un “manual de sistema” donde se indique el proceso de instalación y la atención a posibles fallas en tiempo de ejecución; y además debe existir un “manual de usuario” que especifique paso a paso la forma como debe utilizarse la aplicación desarrollada.

**Diseño:** en esta etapa se plantea la estructura del software a partir de algunos diagramas

de UML[7] (Unified Modeling Language, Lenguaje Unificado de Modelado).

En primer lugar se deben traducir los requerimientos obtenidos de la etapa anterior en un diagrama de casos de uso que describa en forma objetiva cada función que debe implementar el aplicativo.

Después de definir cada caso de uso para la aplicación, se deben seleccionar (si se requieren) los patrones de diseño [8] que serán implementados en la etapa de desarrollo para llevar a cabo cada una de las tareas que se pretenden realizar con el software.

Con base en el diagrama de casos de uso y teniendo en cuenta los patrones de diseño se procede a elaborar el diagrama de clases de la aplicación. En este diagrama se definen cada una de las clases que componen el sistema y se indica, para cada clase, los atributos y métodos correspondientes. Asimismo se debe establecer la relación existente entre cada clase.

Por último, se deben escoger las herramientas que se van a utilizar para el desarrollo de la aplicación móvil. Estas herramientas son, entre otras, el lenguaje de programación, el IDE (Integrated Development Environment, Ambiente Integrado de Desarrollo), la versión del compilador, las APIs (Application Programming Interface, Interfaz de Programación de Aplicaciones) requeridas, el DBMS (DataBase Management System, Sistema Manejador de Base de Datos), los emuladores y todas las demás herramientas que sean necesarias para la etapa de desarrollo.

**Desarrollo:** esta etapa consiste principalmente en codificar, en el lenguaje de programación escogido, cada una de las clases definidas en el diagrama realizado en la etapa de diseño. A medida que se codifica cada clase, se debe redactar su respectiva documentación. La documentación se refiere a la descripción de cada uno de los atributos y métodos de la clase.

Después de codificar y documentar cada clase se deben elaborar los manuales. El manual de sistema debe explicar detalladamente la forma como debe instalarse la aplicación,

ya sea para el dispositivo móvil o para un equipo servidor. En este manual de sistema se deben mencionar las especificaciones técnicas mínimas de hardware y software que requiere el equipo para el funcionamiento adecuado del aplicativo desarrollado.

Debe existir además un manual de usuario que muestre de forma didáctica lo que puede y no puede hacer el usuario con el software desarrollado.

**Pruebas de funcionamiento:** esta cuarta etapa consiste en verificar el funcionamiento de la aplicación. En primer lugar se comprueba el funcionamiento de cada clase en forma individual. Para esto, se pueden probar los métodos utilizando pruebas de unidad con una API adicional (por ejemplo JUnit[9] para Java).

Una vez se hacen las respectivas pruebas de unidad para cada clase, se debe poner a prueba el funcionamiento de toda la aplicación comprobando la interrelación entre todas las clases. Para esto se ejecuta y se observan los resultados obtenidos para compararlos con los esperados. Las pruebas no las debe hacer solo el desarrollador, las deben hacer además otros colaboradores y se deben introducir diferentes datos, inclusive erróneos, para medir el nivel de robustez del software.

Dentro de las pruebas se pueden utilizar emuladores (para aplicación WAP, Midlet, aplicación servidor, entre otras) para comprobar el funcionamiento de todo el sistema. Posteriormente, deben hacerse pruebas de campo en equipos reales para medir el desempeño y el rendimiento del aplicativo.

Si se encuentran fallos en tiempo de ejecución, en la compilación del código, si el software no cumple con los requerimientos especificados, o si el cliente solicita un cambio de última hora, hay que regresar a la etapa de diseño para reestructurar el diagrama de clases y solucionar el inconveniente presentado.

Después de realizar las pruebas de campo y de haber obtenido resultados satisfactorios, se procede a evaluar el sistema. Para la evaluación se plantean dos perspectivas diferentes.

En primer lugar, se busca una muestra de la población objetivo y se les solicita que en una o varias sesiones interactúen con la aplicación y exploren todas sus utilidades. Posterior a esto, se les aplica una encuesta con preguntas objetivas de selección múltiple con única respuesta. Las preguntas deberán estar encaminadas a evaluar el grado de satisfacción del usuario teniendo en cuenta aspectos como la apariencia, la navegación intuitiva, la facilidad del manejo, entre otras.

Por otro lado, se debe buscar un grupo de expertos en el campo del desarrollo de aplicaciones móviles para que utilicen el método de evaluación de las 6M's propuesto por Ahonen, Barret y Golding[10].

Las 6M's consisten en un efectivo método de análisis de los servicios móviles para garantizar que éstos satisfagan las necesidades de los clientes (usuarios) y al mismo tiempo generen ingresos (sean rentables económicamente para el proveedor del servicio). Esta teoría debe su nombre a que consta de 6 atributos: Movement (Movimiento), Moment (Momento), Me (Yo), Multi-user (Multi- usuario), Money (Dinero) y Machines (Máquinas).

Para el proceso de evaluación, a cada una de las 6 M's se le asigna un valor entre 0 y 5, dependiendo que tan bien el servicio cumpla con cada uno de los atributos. Cualquier servicio que brinde un gran valor en cualquiera de las 6 M's tiene un buen potencial para el éxito como servicio móvil. Mientras haya más M's relevantes en el servicio, con una buena puntuación, más probable será que el servicio sea considerado importante por parte de los usuarios. Los servicios con alta puntuación se consideran que tienen un gran potencial en el mercado.

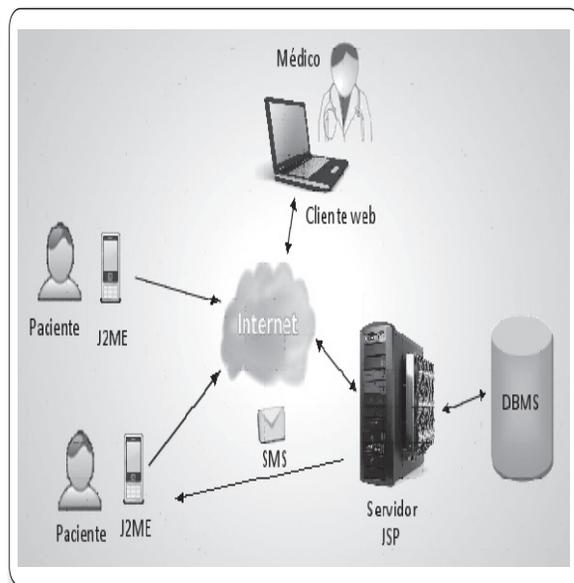
**Entrega:** terminada toda la depuración de la aplicación y atendidos todos los requerimientos de última hora del cliente, se da por finalizada la aplicación y se procede a hacer entrega del ejecutable, el código fuente, la documentación, el manual de sistema y el manual de usuario.

### III. Resultados y discusión

Utilizando la metodología descrita anteriormente se desarrolló el sistema que se aprecia en la figura 2.

El sistema permite que los médicos puedan llevar un seguimiento de la evolución de sus pacientes de forma remota. Para el prototipo implementado se proyecta como población objetivo a los diabéticos, quienes deben llevar un control frecuente del nivel de glucemia que tienen en la sangre. Sin embargo, se hicieron pruebas iniciales de funcionamiento realizando medidas a los investigadores del proyecto.

Figura 2. Arquitectura del sistema



A continuación se describe el funcionamiento global del sistema implementado. Los pacientes tienen un teléfono móvil con una aplicación previamente instalada. Desde esta aplicación ellos registran sus datos personales y las medidas corporales tomadas con su instrumento de medición especial, por ejemplo, los diabéticos medirán su glucosa con el glucómetro. Estos datos

registrados desde la aplicación móvil se envían automáticamente como una petición http hacia el servidor. En este servidor se encuentra una aplicación que se encarga de recibir los datos enviados por el teléfono. Una vez recibidos los datos, se almacenan en una base de datos que reside en el servidor. En cualquier momento, el médico puede ingresar como cliente Web a la información de sus pacientes que se encuentra en la base de datos. Desde el sitio Web, el médico también puede enviar un mensaje de texto hacia un paciente particular.

El sistema consiste básicamente de cuatro componentes. Se desarrolló una aplicación móvil en Java Micro Edition para el teléfono móvil del paciente. Se creó una base de datos relacional utilizando MySQL como sistema administrador de bases de datos (DBMS). Se creó un API en Java Standard Edition para abstraer el acceso a la base de datos y el envío del mensaje de texto. Se implementó también una aplicación servidor con tecnología JSP.

Aplicación móvil: la aplicación móvil o Midlet fue desarrollado en Java Micro Edition utilizando patrones de diseño. Para definir la estructura general del aplicativo se utilizó el patrón de diseño llamado Modelo Vista Controlador[11] que se encarga de separar claramente los datos, la lógica de negocio y la presentación. En la figura 3 se muestra el diagrama de clases del Midlet llamado "DrMovil".

La aplicación DrMovil guarda la información del paciente y la envía al servidor. El Midlet guarda la información en la memoria interna del teléfono al mismo tiempo que la envía al servidor utilizando el protocolo http. Debido a que la memoria de un teléfono móvil es limitada, se almacenan internamente las últimas diez mediciones efectuadas.

El Midlet permite que el paciente revise su historial en modo texto o en modo gráfico. Utilizando las últimas diez mediciones registradas. En la figura 4 se aprecia la gráfica generada por la aplicación móvil.

Aplicación móvil de telemedicina para pacientes hipoglucémicos y diabéticos

Figura 3. Diagrama de clases del Midlet DrMovil

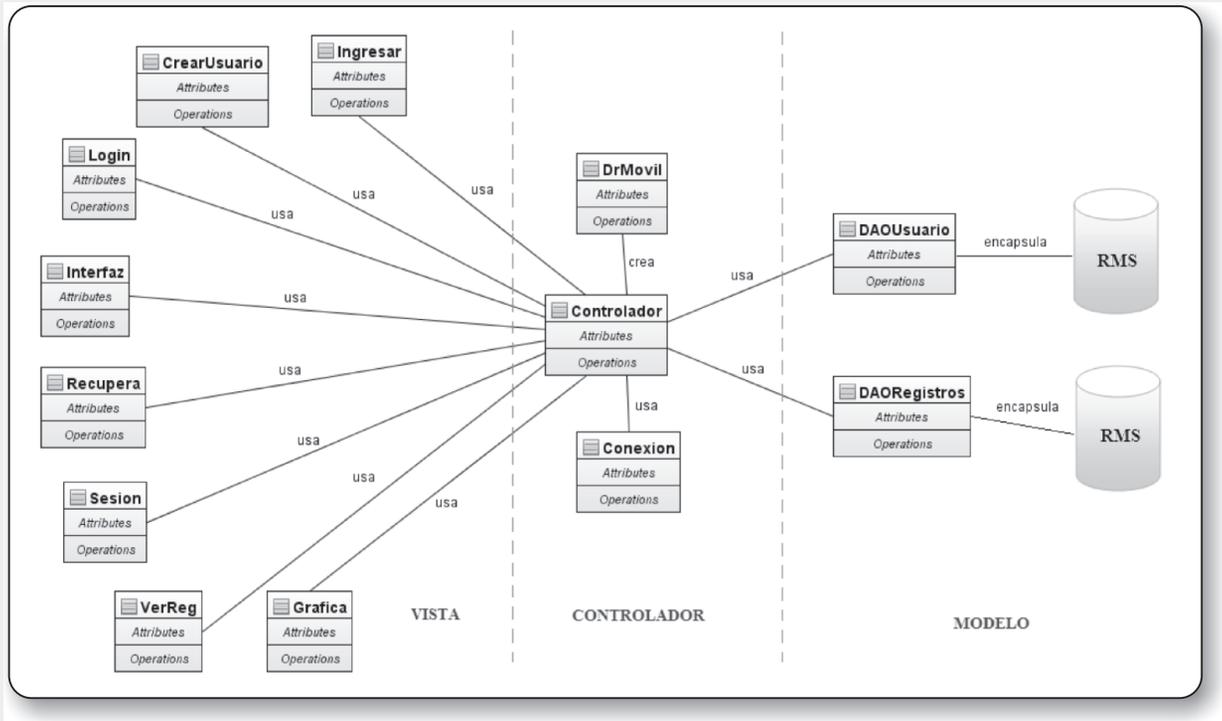


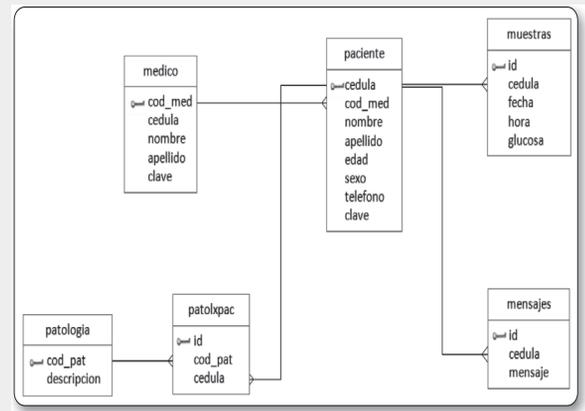
Figura 4. Histórico de registros en modo gráfico



**Base de datos:** se creó una base de datos relacional para almacenar la información de pacientes y médicos. El sistema completo está en capacidad de albergar varios médicos y varios pacientes por cada médico. La base de datos consta de seis tablas que guardan datos de médicos, pacientes, patologías, muestras y

los mensajes enviados por los médicos a sus pacientes. En la figura 5 se presenta el modelo entidad relación de la base de datos creada en MySQL.

Figura 5. Modelo Entidad Relación de la base de datos



**API para acceder a la base de datos:** se desarrolló un conjunto de clases y métodos para abstraer el acceso a la base de datos. Para este API se utilizó el patrón de diseño Data Access Object[12] para acceder a la base de datos. En la figura 6 se muestra el diagrama de clases del API desarrollado.

La clase DAODrMovil que se ve en la figura 6 contiene todos los métodos necesarios para interactuar con la base de datos. De esta manera, si se quiere ingresar un nuevo registro en cualquier tabla o si se desea consultar cualquier dato, solo se debe crear una instancia de esta clase y utilizar sus métodos. Hay una clase para abrir la conexión, otras clases que encapsulan los datos de las tablas y otras clases que acceden directamente a través de comandos SQL a la base de datos.

La clase principal del API contiene un método que se encarga de enviar una petición http para el envío del mensaje de texto. Esta API es utilizada por la aplicación servidor para interactuar con la base de datos y para enviar un SMS en el momento que el médico considere oportuno.

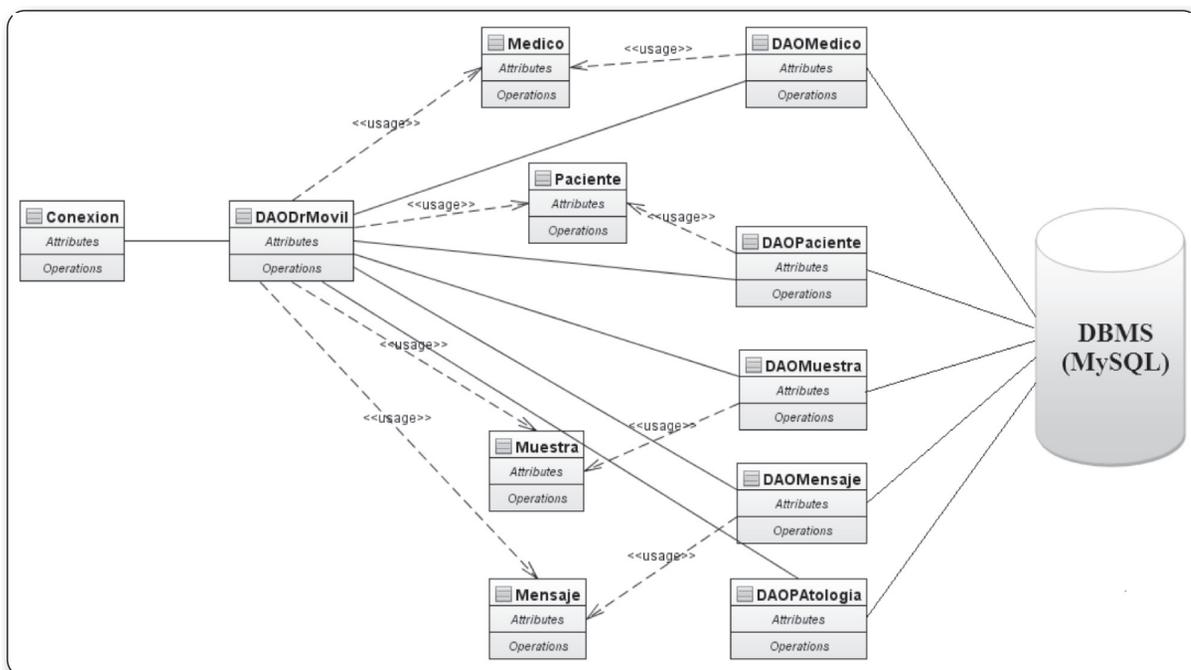
Aplicación servidor: esta aplicación fue creada para que el médico acceda a los datos de sus pacientes. Se utilizó tecnología JSP (Java Server Pages) para la creación de contenidos dinámicos. El médico debe ingresar a la URL (Uniform Resource Locator) correspondiente e iniciar sesión con el número de su cédula y su respectiva contraseña. Luego de identificarse, se le presenta al médico un listado de los

pacientes que tiene a cargo. Desde ahí puede escoger el paciente que quiera para ver su evolución.

La aplicación servidor le permite al médico enviar un mensaje de texto al paciente que desee. Por medio de este mensaje el profesional de la salud envía una retroalimentación al paciente en el momento que crea pertinente. Para el envío de este mensaje se adquirió el servicio con un proveedor internacional de mensajería corta de texto. El API desarrollado para acceder a la base de datos, contiene un método que se encarga de hacer una petición http con ciertos parámetros específicos. Esta petición llega al servidor del proveedor que se encarga posteriormente del envío del mensaje de forma inmediata.

A través de la aplicación servidor el médico gestiona la información de la base de datos. La aplicación utiliza el API desarrollado para validar la identidad del médico, mostrar la información de los pacientes y enviar un mensaje corto de texto. Además, en el servidor se encuentran dos componentes que se encargan de atender la petición http que hace el teléfono móvil. En esta petición se envían, desde el teléfono hacia el servidor,

Figura 6. Diagrama de clases del API desarrollada para acceder a la base de datos



los datos personales del paciente y los valores registrados de cada muestra tomada. En el servidor se recibe esta información, se almacena en la base de datos y se envía un mensaje de confirmación al dispositivo móvil.

## IV. Evaluación

Utilizando la evaluación de las 6M's, y pruebas de funcionamiento correspondientes, se realizó la valoración a la aplicación móvil implementada.

**Evaluación de la 6M's:** a cada una de las 6Ms se le asigna un número entero entre 1 y 5, dependiendo del cumplimiento de la aplicación con cada uno de los atributos. Los atributos con alta puntuación en cualquiera de las 6Ms se consideran que tienen un gran potencial para el éxito del sistema. En la tabla 1 se presentan los resultados de esta evaluación. Como se aprecia en la tabla, la mayor parte de las características evaluadas tiene un alto nivel de aceptación. Esto implica que la aplicación tiene un gran potencial de éxito para ser

comercializada. La aplicación tiene altas probabilidades de aceptación para distribuirse entre las EPS que tengan programas especiales para pacientes de cuidado en casa como los diabéticos o los hipertensos.

**Pruebas de sincronización:** en éstas se comprobó el proceso de actualización de información desde el teléfono móvil al servidor. Para esto se utilizó un entorno simulado (emulador con conexión al servidor) y teléfonos reales conectados a través de Wi-Fi con el servidor; Además de teléfonos reales conectados al servidor utilizando la red de GPRS de los diferentes operadores de telefonía móvil celular. (Movistar, Comcel y Tigo).

En primera instancia se verificó el almacenamiento de un nuevo paciente en la base de datos enviando la información desde el Midlet, y verificándola con una consulta de los pacientes registrados en la base de datos. Para esto, se accedió a la consola de MySQL desde el servidor. Ver figura 7.

Tabla 1. Resultados de la evaluación de las 6M's

M	Calificación	Justificación
Movement	4	El teléfono es móvil para la aplicación, no para realizar una toma de medidas, pues debe estar estático para eso.
Moment	5	Se puede realizar una medición y/o revisar el historial en cualquier momento.
Me	5	Es totalmente personalizado y se pueden encontrar los datos de interés. Los datos suministrados son importantes para el usuario.
Multi-user	3	Solo se puede interactuar con la aplicación del médico y no con otros usuarios.
Money	5	Beneficio económico para el usuario y para la IPS una vez se implemente.
Machines	3	La interfaz es amigable, pero los equipos que se tienen que manipular y conectar al teléfono no los son

Figura 7. Verificación de paciente nuevo insertado

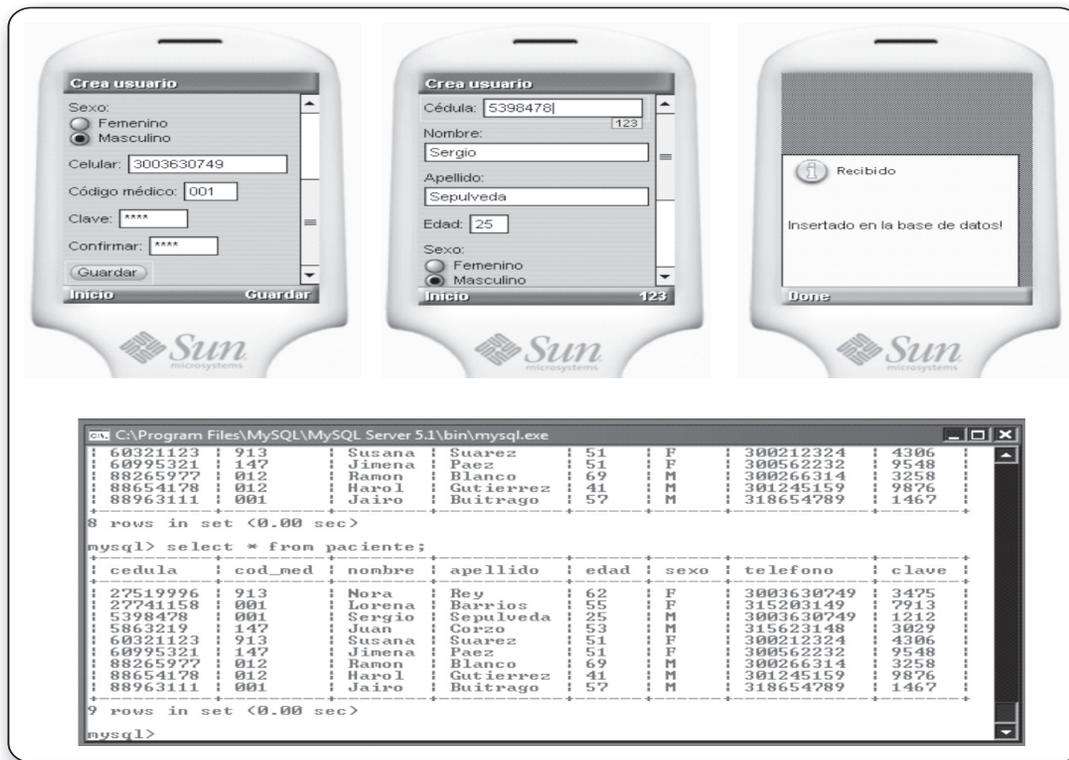
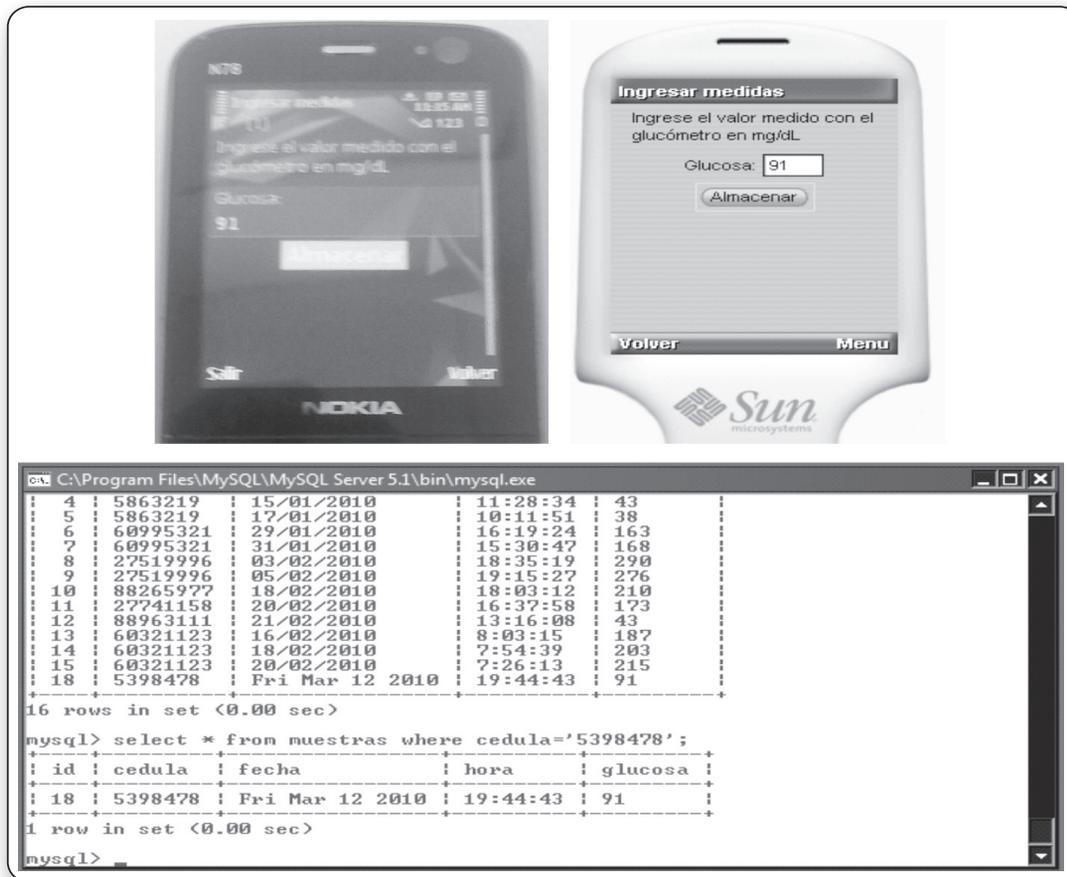


Figura 8. Verificación de muestra almacenada



Por otra parte, se debía corroborar que desde el Midlet se almacenara la muestra correspondiente a la medida corporal que el usuario registrara. La forma de hacer esto es igual a la anterior, a través de una petición http se envían los datos a la aplicación servidor, la cual almacena la información en la base de datos utilizando el API desarrollado para este fin. Ver figura 8.

## V. Conclusiones y recomendaciones

Doctor Móvil no pretende ser el reemplazo de los médicos generales. Lo que busca es ser una ayuda para éstos, además de brindar beneficios económicos a las pacientes que necesitan monitorear constantemente su estado de salud. También pretende ayudar a descongestionar el proceso de solicitud de citas médicas en las IPS.

El sistema desarrollado representa un beneficio en tiempo y costo a los pacientes que lo utilicen. Doctor Móvil les brinda la posibilidad a los pacientes de llevar un autocontrol de su salud y les evita desplazarse hacia un consultorio médico en el caso que no lo necesiten. De esta forma no incurrir en gastos de transporte, de cuota moderadora y les ahorra tiempo.

Con la utilización del prototipo se puede generar un diagnóstico provisional que permite agilizar el trabajo del médico. Es decir, si un paciente sufre alguna anomalía en una de las medidas tomadas del nivel de glucemia en la sangre, el doctor tendrá la posibilidad de conocer esto, antes de recibir su visita; de esta manera tomará una medida correctiva oportuna cuando vea al paciente.

Con el sistema propuesto se consigue una centralización de los datos de los pacientes hipoglucémicos y diabéticos que lo utilicen. Permite almacenar en un mismo lugar los valores medidos del nivel de glucosa en la sangre y alguna información básica para cada

persona, tal como nombre, identificación, sexo, entre otros. Al tener la información centralizada es posible acceder a ella en cualquier momento que se requiera.

El uso de herramientas libres disminuye el costo de implementación del sistema. Esto permite ofrecer a las EPS un servicio eficiente, de calidad y a un costo inferior comparado con el precio del mismo sistema implementado con herramientas propietarias.

La metodología DAM propuesta orienta al desarrollador en una serie de etapas consecuentes para la ejecución de un proyecto en aplicaciones móviles. Esta metodología se convierte en una herramienta importante para el seguimiento y desarrollo estructurado de este tipo de aplicaciones, brindando el soporte de la ingeniería de software tradicional con el uso de diagramas UML y patrones de diseño; además, propone métodos de evaluación cualitativos y cuantitativos que implican un análisis objetivo de las funcionalidades del prototipo desarrollado.

El sistema desarrollado puede adaptarse para registrar cualquier medida corporal. El prototipo propuesto en este documento está diseñado para pacientes con hipoglucemia o diabetes. Sin embargo, con unas mínimas modificaciones puede servir para llevar el control de otras medidas corporales, tales como la presión arterial, la temperatura corporal, la saturación de oxígeno en la sangre, entre otras.

## Bibliografía

- [1] TELEFONICA. (2010). HealthPresence. Dossier de prensa telefónica [en Línea]. [http://pressoffice.telefonica.com/documentos/DossierTelefonica\\_0.pdf](http://pressoffice.telefonica.com/documentos/DossierTelefonica_0.pdf)
- [2] GÓMEZ R. A, GALVIS A. H, MARIÑO O. (1998) Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micromundos interactivos.

- [3] BRAUDE Eric (2005). Ingeniería de software: una perspectiva orientada a objetos. Alfaomega Grupo Editor.
- [4] BOOCH Grady (1993) Object-oriented analysis and design with applications. Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA
- [5] JACOBSON Ivar (2000) The road to the Unified Software Development Process. Cambridge University Press.
- [6] RAMBAUGH James (1991) Object-Oriented Modeling and Design. Prentice Hall.
- [7] MARTIN Fowler, KENDALL Scott (1999), UML Gota a Gota.
- [8] GAMMA Erich, HELM Richard, JOHNSON Ralph, VLISSIDES John (1994). Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley (GoF- Gang of Four).
- [9] "JUnit.org Resources for Test Driven Development" [en línea]. Welcome to JUnit.org!. Disponible en web. <http://www.junit.org> [Consulta: 19 de marzo de 2010].
- [10] AHONEN Tomi, BARRET Joe, GOLDING Paul (2002). "Services for UMTS, creating killer applications in 3G. Wiley.
- [11] SEPÚLVEDA Sergio, CAMARGO Leonardo, MEDINA Byron. Laboratorio de Aplicaciones Móviles. Congreso Regional de Eléctrica, Electrónica y Sistemas, CREES 2009.
- [12] "SUN Developer Network (SDN)". Core J2EE Patterns – Data Access Object. Disponible en Web. <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>. [Consulta: 19 de marzo de 2010].