

PROTOTIPO DE TELEMEDICINA MÓVIL PARA ASISTENCIA MÉDICA DOMICILIARIA Y REMOTA.

David Andres Roncancio Joya

Universidad Distrital, Bogotá, Cundinamarca, Colombia, david.roncancio@iguanamobile.net

Jair Giovanni Beltran Vera

Universidad Distrital, Bogotá, Cundinamarca, Colombia, giovanny.beltran@iguanamobile.net

Wilmar Yamit Cardenas Mahecha

Universidad Distrital, Bogotá, Cundinamarca, Colombia, yamit.cardenas@iguanamobile.net

Carlos Enrique Montenegro Marin

Universidad Distrital, Bogotá, Cundinamarca, Colombia, cemontenegrom@udistrital.edu.co

Paulo Alonso Gaona Garcia

Universidad Distrital, Bogotá, Cundinamarca, Colombia, pgaonag@udistrital.edu.co

RESUMEN

El proyecto busca como objetivo Analizar, diseñar y desarrollar un prototipo de sistema de información en telemedicina con tecnologías móviles para la asistencia médica domiciliaria y remota, usando el estándar hl7 [12] y servicios Web [9], en este artículo se presentaran los resultados finales del desarrollo del prototipo [1], inicialmente se describirán las herramientas empleadas.

Palabras claves: Movil, Web, Servicios, hl7, Medica.

ABSTRACT

The project seeks to analyze, design and develop a prototype information system in telemedicine with wireless technologies for medical care at home and remotely, using the standard HL7 [12] and Web services [9], this article presents the final results of the prototype development [1], initially described the tools used.

Keywords: Mobile, Web, Services, HL7, Medical

1. INTRODUCCIÓN

El prototipo de telemedicina móvil para asistencia médica domiciliaria y remota a través de servicios Web y tecnologías móviles, abarcara el servicio de acceso a historias clínicas y antecedentes vía web y móvil para apoyar la prestación de una asesoría remota entre médicos.

2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO

Django Framework [7]: Esta herramienta fue seleccionada por su interfaz administrativa “out-of-the-box” que redujo en gran parte el trabajo de la interfaz Web, reduciendo los tiempos de desarrollo especialmente en la generación de formularios, ya que estos se crearon empleado el descriptor del modelo de datos herramienta que ofrece Django para generar formularios a partir del modelo de la base de datos.

Google Code Host [1]: Herramienta empleada para hospedar el código de fácil acceso vía Web con manejo de RSS y usuarios.

Java ME [28]: API empleado para el desarrollo sobre el dispositivo móvil.

LWUIT [6]: Kit de herramientas de peso ligero para interfaces gráficas en Java ME (Lightweight UI Toolkit for Java ME) se utilizó para la creación de interfaces gráficas sobre dispositivos móviles.

Postgresql 8.3 [21]: Manejado Libre de Bases de Datos utilizado en el proyecto.

Floggy [24]: Es un Framework de persistencia para aplicaciones en Java ME.

Perst Lite [24]: Librería para Java Me que funciona como un ODBMS.

Python-HL7 [13]: Librería libre desarrollada por Jhon Paulett, está en una etapa muy inicial de desarrollo sin embargo es la única existente y sirve para parsear mensajes HL7 en arreglos de datos más manejables.

Soaplib / soappy [25][27]: Librerías “open-source” para el manejo de mensajes SOAP en Python, además sirven para generar WSDL de forma automática.

Wireless ToolKit[6]: Es un kit de herramientas que ayudan en el desarrollo de aplicativos basados en Java ME.

3. METODOLOGÍA

Escoger la metodología de desarrollo fue uno de los temas que más ocupó tiempo de investigación en la etapa preliminar del trabajo, puesto que de ella dependía la forma de trabajo durante toda la labor de ingeniería.

Se quería usar una metodología ágil por que se esperaba reducir los tiempos para lograr los objetivos propuestos ya que la labor de investigación de dominio podría retrasar el proyecto, se evaluaron metodologías ágiles como XP (eXtremme Programming) [5], AUP (Agile Unified Process)[5], SCRUM[5] y FDD[2] (Feature Driven Development), siendo esta última la que se decidió usar en este proyecto.

FDD[2] fue escogida por las características especificadas en la documentación, que lo recomienda para equipos de desarrollo pequeños y medianamente expertos en las herramientas de desarrollo, además de no ser necesario un gran conocimiento del dominio.

4. ANÁLISIS DEL DOMINIO DEL SISTEMA

La metodología FDD[2] no hace mención, ni considera una recolección de requerimientos del sistema antes de comenzar con el modelo global del sistema, esto es porque dentro del equipo en FDD[2] debe haber al menos un experto en el dominio del sistema, que será el que participe en la elaboración de los modelos y estará siempre supervisando el correcto enfoque del sistema.

Sin embargo en este proyecto ninguno de los participantes dominaba el tema, por eso fue necesario hacer una recolección de información y seguido de una recolección de requerimientos que se consideró pertinente para el desarrollo del sistema, este documento de requerimientos es el resultado del análisis del dominio del sistema realizado por el equipo para comprenderlo y conocerlo [5].

4.1 REGLAS DEL NEGOCIO

Las reglas del negocio obtenidas en el levantamiento de información fueron:

Table 1: Reglas del Negocio

Identificador	Descripción
RN.1	Únicamente el administrador general podrá crear usuario del tipo administrativo (usuario de entidad prestadora, usuario de la entidad administradora)
RN.2	Todos los usuarios del sistema deberán acceder al mismo a través de su número de identificación y su contraseña

RN.3	Únicamente el administrador de entidad prestadora estará en la capacidad de afiliar nuevo personal médico asociado a la entidad para la cual trabaja (auxiliar, médico)
RN.4	Solamente el administrador de la entidad administradora puede estar en capacidad de registrar y modificar pacientes nuevos
RN.5	Solamente el administrador de la entidad administradora podrá registrar nuevos empleados administrativos (operador de citas)
RN.6	Las entidades administradoras pueden tener contratos con más de una entidad prestadora
RN.7	Una entidad prestadora puede ser contratada por más de una entidad administradora
RN.8	Las entidades prestadoras tienen varias sedes de atención.
RN.9	Únicamente los médicos podrán asignar servicios a sus pacientes
RN.10	Los servicios que se le prestan a un paciente son del tipo formula medica, incapacidad, examen, remisión a especialista
RN.11	Un paciente solo puede estar afiliado a una entidad prestadora.
RN.12	El administrador de citas es el único que puede asignar o modificar las citas medicas
RN.13	La historia clínica podrá ser accedida en su totalidad únicamente un médico que este atendiendo a ese paciente
RN.14	Solo si el paciente es mujer, la historia clínica tendrá el componente ginecobstetrico
RN.15	Toda entidad administradora y prestadora debe tener al menos un administrador asociado
RN.16	Solo se pueden publicar casos anónimos desde el sistema web
RN.17	Una Historia clínica no puede ser modificada, solo actualizada

Para poder ampliar más detalladamente cada una de los objetivos se pueden consultar en [1].

4.2 REQUERIMIENTOS DE INFORMACIÓN

Los requerimientos de información básicamente se refieren al compendio de datos que el sistema contendrá en los diferentes módulos, para consultar estos requerimientos en detalle están en [1].

4.3 RESTRICCIONES DE INFORMACIÓN

A continuación se listan las restricciones detectadas al inicio del desarrollo del prototipo:

Table 2: Restricciones de Información

CRQ-1	Unicidad de usuarios	CRQ-4	Unicidad de las entidades prestadoras
CRQ-2	Alarmas en los pacientes	CRQ-5	Unicidad de las citas
CRQ-3	Unicidad de las entidades administradoras		

Para ampliar al detalle de cada una de las restricciones anteriormente nombradas se pueden consultar en [1].

4.4 ACTORES DEL SISTEMA

El listado de actores detectados para el prototipo es el siguiente:

Table 3: Actores del Sistema

ACT-1	Administrador General	ACT-5	Personal Médico
ACT-2	Administrador Entidad Administradora	ACT-6	Médico General
ACT-3	Administrador Entidad Prestadora de salud	ACT-7	Paciente
ACT-4	Administrador de citas		

Si se desea ampliar al detalle las características de los autores anteriormente listados, se puede consultar en [1].

4.5 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales resultantes del levantamiento de información son:

Table 4: Requerimientos Funcionales

FRQ-01	Consultar agenda médico desde el móvil	FRQ- 18	Administrar citas médicas.
FRQ- 02	Sincronizar Agenda con el servidor	FRQ- 20	Conversión de historias clínicas tradicionales al formato CDA.
FRQ- 03	Ver detalles de cita en la agenda Médica móvil	FRQ- 21	Guardar historias clínicas en formato CDA.
FRQ- 04	Ver detalles de cita en la agenda Médica Web	FRQ- 22	Consultar historias clínicas en formato CDA.
FRQ- 05	Consultar agenda desde Web	FRQ- 23	Realizar orden de incapacidad laboral desde el móvil.
FRQ- 05	Consultar agenda desde Web	FRQ- 24	Generar formula médica desde el móvil.
FRQ- 07	Realizar consulta Médica General desde el móvil	FRQ- 25	Generar orden de remisión a especialista desde el móvil.
FRQ- 08	Realizar petición de exámenes paraclínicos desde el móvil.	FRQ- 26	Consultar la clasificación CIE-10 (reducida) desde el móvil.
FRQ- 09	Realizar consulta de Historia clínica de un paciente desde el móvil.	FRQ- 27	Consultar la clasificación de procedimientos médicos CUPS (reducida) en el móvil
FRQ- 10	Publicar Caso anónimo en el Portal web.	FRQ- 28	Registrar Evento médico desde el móvil
FRQ- 11	Comentar caso anónimo	FRQ- 29	Generar Log de cada suceso en el servidor
FRQ- 12	Autenticar usuarios	FRQ- 30	Actualizar Historia clínica del paciente desde el móvil
FRQ- 13	Guardar Información en el Móvil.	FRQ- 31	Administrar entidades administradoras
FRQ- 14	Registrar personal médico en el sistema.	FRQ- 32	Administrar entidades prestadoras de salud
FRQ- 15	Cambiar contraseña	FRQ- 33	Administrar Entidad Administradora
FRQ- 16	Anexar multimedia	FRQ- 34	Administrar Entidad prestadora de salud
FRQ- 17	Notificar eventos al paciente en el móvil.	FRQ- 35	Usar HL7 messaging como protocolo de intercambio de mensajes

Con el fin de ampliar al detalle los requerimientos funcionales anteriormente listados, se pueden consultar en [1].

4.6 REQUISITOS NO FUNCIONALES

Los requerimientos No funcionales resultantes del levantamiento de información son:

Table 5: Requerimientos No Funcionales

NFR-1	El sistema se debe comunicar con el servidor haciendo uso de HTTPS	NFR-8	La información del sistema se debe mantener en un servidor
NFR-2	El sistema debe usar SOAP como protocolo para comunicación	NFR-9	La información del sistema se debe manejar con el DBMS Postgres
NFR-3	El sistema debe usar HL7 - CDA	NFR-10	La comunicación debe hacerse por medio de web services
NFR-4	El sistema debe usar Java ME	NFR-11	El sistema web se debe hacer usando el framework Django
NFR-5	la interfaz grafica debe ser hecha en LWUIT	NFR-12	El sistema deberá ser probado sobre equipos reales
NFR-6	El sistema debe usar JSR172	NFR-13	El sistema deberá ser sometido a pruebas
NFR-7	El sistema debe usar MMAPAPI(JSR135)		

Para ampliar al detalle los requerimientos No funcionales anteriormente listados, se pueden consultar en [1].

5. DESARROLLO DEL MODELO GENERAL

5.1 MODELO GLOBAL FDD

Conforme a los resultados obtenidos durante el proceso de levantamiento de requerimientos [1] y de acuerdo a la metodología [2], el siguiente paso a dar para la construcción del prototipo es realizar un análisis y una retroalimentación de la documentación del dominio (leyes, estándares, entrevistas, etc.) [10] [11] [12] [13] [14] [17] [19][22] y del documento de requerimientos [1], con base en ellos se procede a diseñar un modelo global del sistema (en este caso un mapa mental) que abarque las características más importantes para el desarrollo del proyecto.

El modelo global resultante de esta fase se puede consultar en [1].

5.2 DIAGRAMA DE CLASES

5.2.1 SISTEMA WEB

Gracias al DBMS que se uso se pueden definir modelos OO y estos se verán especificados en forma física en una base de datos relacional, por este motivo las clases descritas en el modelo E-R de la base datos son diseñadas en del sistema Web y a estas clases además se le agregan los métodos necesarios para lograr las características requeridas, sin embargo existen otras clases emergentes relacionadas sobre todo con la comunicación, como por ejemplo la clase generador_wsdl, que se encarga al momento de ser instanciada de agregar todas las funciones al WSDL.

Gracias a la arquitectura del Framework de desarrollo web, no es necesario clases para la interfaz gráfica, ya que todas las funcionalidades son implementadas a nivel del modelo y la forma de desplegarlas y controlar la información a desplegar es realizada a otro nivel; estas vistas son en primera instancia definidas por una “libreta de direcciones” (urls.py) que describe la función (views.py) encargada de consultar los métodos de la lógica del negocio (models.py), y esta función a la vez se encarga de llamar al template que dibuja la interfaz gráfica.

5.2.2 SISTEMA MÓVIL

Después de un análisis exhaustivo del dominio, de los componentes fundamentales dentro del sistema y de analizar sus relaciones, basados en el modelo global del sistema se realizó una abstracción de todas aquellas

entidades que eran de importancia y se debían modelar dentro del sistema, como resultado de esto se diseñaron las clases que modelan el sistema móvil.

Realizando una separación de clases se modelaron cuatro subsistemas que mostraron ser relevantes: el paquete de *lógica* donde se abstraen y representan la mayoría de entidades del sistema; El paquete de *presentación* que contiene todas aquellas clases que proveen interfaz al usuario para interactuar con el sistema; El paquete de *persistencia* que contiene las clases encargadas de manipular aquella información que el sistema debe almacenar; Y por último el paquete de *conexión*, que contiene las clases que se encargan de manejar los diferentes protocolos de conexión del sistema DoCMobile con DoCWeb.

Aunque se genera un modelo de clases para el proyecto, no significa que sea único e inalterable, es muy probable en este tipo de desarrollos que se presenten cambios al diseño que no fueron contemplados cuando se crearon.

Los demás diagramas se encuentran disponibles en [1].

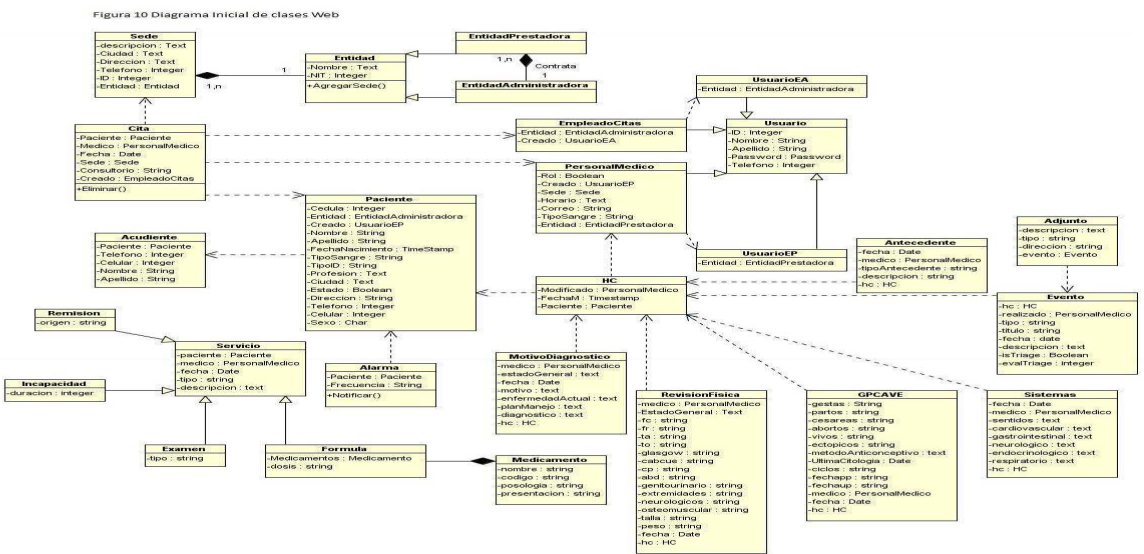


Figure 1: Diagrama Inicial de clases Web

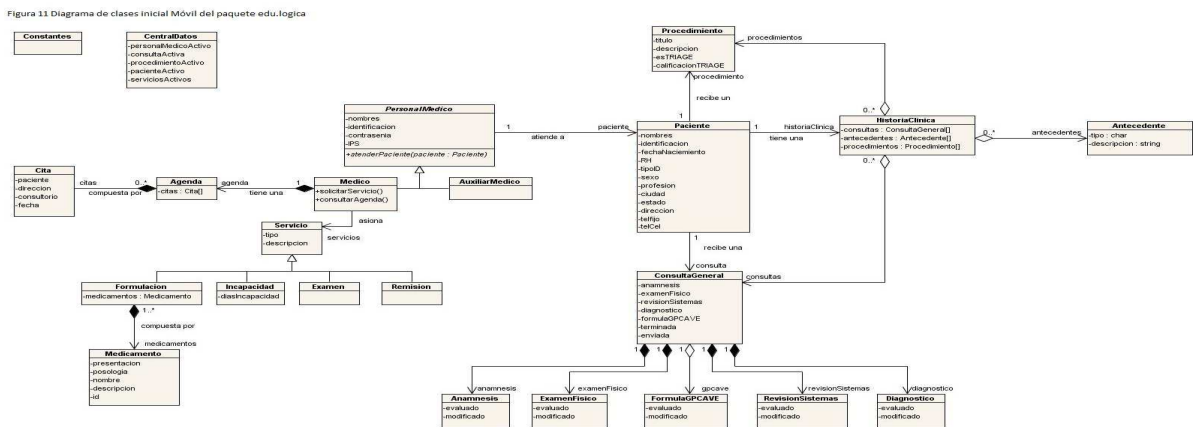


Figure 2: Diagrama de clases inicial Móvil del paquete edu.logica

5.3 DIAGRAMAS DE NAVEGACIÓN

5.3.1 SISTEMA WEB

Para el diseño de la interfaz web lo primero que se hizo fue crear un wireframe (o esqueleto) en papel que definiera como se iban a cumplir todas las funcionalidades de manera coherente, luego se busco un template o conjunto de html / css / js adecuado para representar este bosquejo inicial, posteriormente con ayuda de una herramienta para el desarrollo de wireframes profesionales, se creó uno con los lugares donde irían los menús, las casillas de búsqueda y en general todos los elementos de la interfaz gráfica acorde con las funcionalidades y con el diseño visual escogido. En total se diseñaron 23 pantallas que pueden ser vistas en su totalidad en [1].

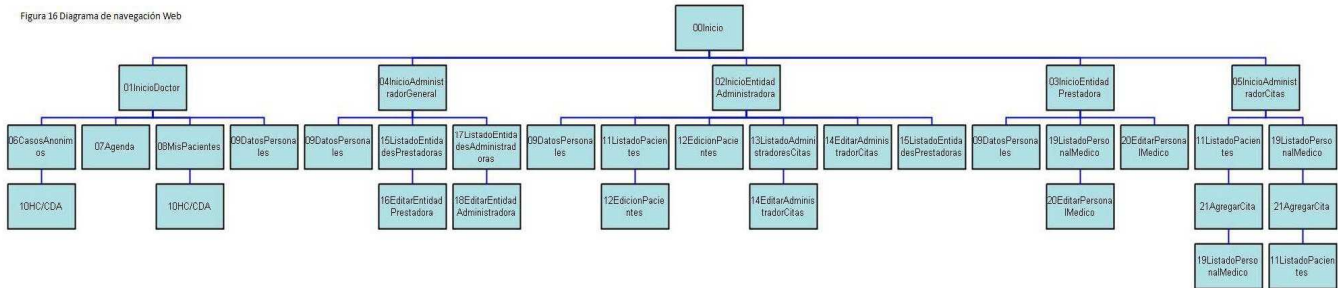


Figure 3: Diagrama de navegación web

5.3.2 SISTEMA MÓVIL

El diseño de la interfaz de usuario para el DoCMobile fue creado teniendo en cuenta las diferentes funcionalidades que tendría el sistema, buscando siempre que fueran lo más simples y rápidas por estar trabajando en un dispositivo Móvil. DoCMobile concibe dos tipos de usuarios distintos, el médico general y el personal médico, para cada uno de ellos existe un mapa de navegación distinto aunque con cierta similitud, uno de los mapas se muestra a continuación y el otro puede ser consultado en [1] :

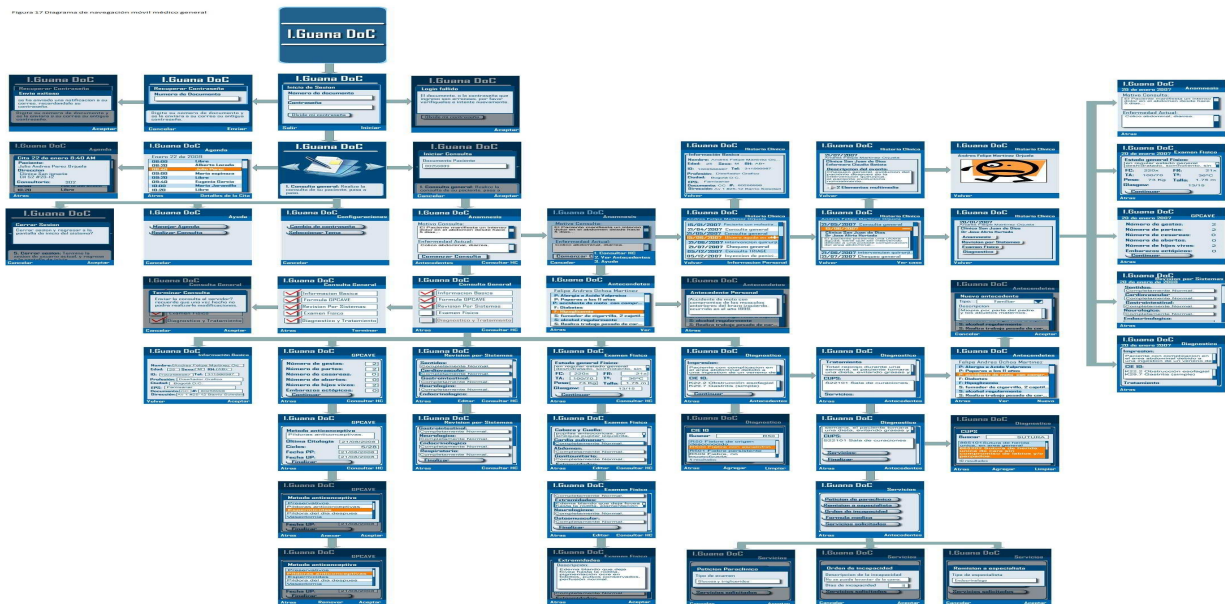


Figure 4: Diagrama de navegación móvil médico general

5.4 DISEÑO DE LA BASE DE DATOS

Basados en la información recogida durante la fase de análisis del dominio, y en los requerimientos de información levantados, se diseñó el modelo Relacional y el modelo Físico de la base de datos, que luego de varias iteración; definió el modelo general, esta información se puede consultar en [1] PROTOTIPO DE TELEMEDICINA MÓVIL PARA ASISTENCIA MÉDICA DOMICILIARIA Y REMOTA.

6. RESULTADO FINAL

El resultado final del proyecto es el prototipo de telemedicina móvil para asistencia médica domiciliaria y remota DoC, Este consistente en dos aplicaciones, un sistema web que hace las veces de servidor y un cliente móvil desarrollado en Java ME. El prototipo cumple las funcionalidades y características planteadas en el proyecto [1].

A continuación se describen las dos aplicaciones que conforman el sistema DoC.



Figure 5: Interfaz de administración Doc Web.

El sistema Web DoCWeb contiene una interfaz administrativa en la cual se pueden administrar las diferentes entidades relacionadas en el área de la salud, EPS e IPS, se pueden hacer relaciones entre las entidades y asignar personal encargado para cada una de ellas; modificar la información de estos usuarios, asignarles nuevos roles e incluso removerlos del sistema. El desarrollo con Django facilitó bastante el trabajo de codificación ya que el Framework viene con algunas plantillas y códigos de ejemplo, además de algunos módulos que ayudan a desarrollar las páginas de manera más sencilla y ágil.

DoCWeb también contiene afiliación de pacientes al sistema y contratación de personal médico por parte de las IPS, toda esta información se almacena en una base de datos para luego ser consultada por el mismo aplicativo o el cliente móvil. El portal ha sido montado en un dominio gratuito <<https://iguanamobile.no-ip.biz/docweb/>> bajo un servidor propio con sistema operativo Ubuntu 8.04, y un servidor de aplicaciones Apache 2.2.8 (Ubuntu). Ha sido probado sin complicaciones en navegadores Internet Explorer 7, Mozilla Firefox 3.0.8 y Opera 9.62. (Si lanza una advertencia de confianza basta con aceptar el certificado del sitio para desplegarlo).

DoCMobile es el cliente que consiste en una aplicación para teléfonos con soporte Java ME, específicamente MIDP 2.0 que es el perfil mínimo usado actualmente por los fabricantes de dispositivos móviles. Gracias al uso de herramientas como Perst y froggy se logró implementar persistencia y manejo de registros muy extensos con un uso similar a un motor de base de datos convencional, con estas ventajas fue posible implementar tres estándares distintos en la medicina nacional comentados en este artículo (POS, CUPS y CIE10) que cuentan con

datos de más de mil registro cada uno. Cada uno de estos estándares es usado en diferentes funcionalidades dentro de la aplicación.



Figure 6: Uso de estándares en Doc Mobile

Otra ventaja, se logro con el Framework móvil LWUIT, esta herramienta permitió desarrollar una interfaz más intuitiva y fácil de manejar, además de solucionar los problemas de resolución para diferentes pantallas.

La conexión con el servidor se hace de manera fácil y transparente haciendo uso de los servicios Web existentes en DoCWeb, se ejecutan las consultas médicas, procedimientos y evaluaciones TRIAGE, que luego son actualizadas en la historia clínica del paciente atendido haciendo uso de HL7-message.

La aplicación móvil fue probada en celulares Nokia, la aplicación corrió perfectamente en el Nokia 5310 XpressMusic y el Nokia 5200. Sin embargo es recomendable usar la aplicación en celulares con buen procesamiento (al menos 200Mhz) por el uso de las librerías externas y las conexiones constantes a internet, en lo posible con un teclado QWERTY o la escritura a mano de los nuevos celulares touch screen, puesto que facilita la escritura, teniendo en cuenta que la aplicación registra mucha información que debe digitarse. Para otros teléfonos de gama media existe la posibilidad de grabar audio para evitar digitar en T9.

REFERENCIAS

- [1] RONCANCIO, David, BELTAN, Jair, CARDENAS, Wilmar, GAONA, Alonso, MONTENEGRO, Carlos, PROTOTIPO DE TELEMEDICINA MÓVIL PARA ASISTENCIA MÉDICA DOMICILIARIA Y REMOTA, Universidad Distrital Francisco Jose de Caldas Colombia 2009.
- [2] CALABRIA, Luis, Metodología FDD, Universidad ORT Uruguay. 2003.
- [3] HUGH Darwen. Foundation for future database systems: the third manifesto: a detailed study of the impact of type theory on the relational model of data, including a comprehensive model of type inheritance. Segunda Edicion. Adisson-Wesley Proffesional xxiii ISBN 0-201-70928-7
- [4] HOLOVATY, Adrian, KAPLAN-MOSS, Jacob. The Django Book.
- [5] DURÁN, Amador, Metodología para la Elicitación de Requisitos de Sistemas Software, Universidad de Sevilla. 2002.
- [6] CARDENAS Patricia, CARDENAS Jorge. j2me java2 micro edition. manual de usuario y tutorial. Alfa-omega. RAMA. Espana 2004. ISBN: 970-15-1022-4
- [7] BENNET, James. Practical Django Projects. E-book. United States of America, 2008 ISBN: 9781590599969.
- [8] GONZALES DUQUE, Raul. Python para todos. España 2007. Tutorial python.
- [9] MORALES MACHUCA Carlos Andres, Estado del arte: Servicios Web. Universidad Nacional de Colombia, Bogota 2008
- [10] Actualización Medicamentos Incluidos en Plan Obligatorio de Salud POS [Web en línea]. <http://www.med-

informatica.com/OBSERVAMED/Deposito_legal/ActualizacionMedPOS_DesdeA226.htm >. [Consulta: 04-06-2009]

- [11] Guía básica para la confección de una Historia Clínica. La Anamnesis Remota. [Web en línea]. <http://www.portalesmedicos.com/publicaciones/articulos/733/1/Guia-http://www.w3schools.com/soap/soap_intro.aspbasica-para-la-confeccion-de-una-Historia-Clinica.-La-Anamnesis-Remota.> Consulta [12-02-2009]
- [12] ATA - American Telemedicine Association - [Web en línea] <<http://www.americantelemed.org/news/definition.html>> [Consulta: 06-06-2008].
- [13] HL7 Book – CDA [Web en línea] <<http://hl7book.net/index.php?title=CDA>> [Consulta: 06-02-2009]
- [14] Clasificación Internacional de Enfermedades 10° CIE 10° REVISION [Web en línea]. <<http://www.revmed.unal.edu.co/obro/subpages/cie10.pdf>>. [Consulta: 04-04-2009]
- [15] the Web framework for perfectionists with deadlines [Web en línea] <<http://www.djangoproject.com/>> [Consulta: 08-01-2008].
- [16] El framework web para perfeccionistas con deadlines [Web en línea] <<http://django.es/>> [Consulta: 08-01-2008].
- [17] TRIAGE de urgencia [Web en línea] disponible en <<http://encolombia.com/medicina/enfermeria/enfermeria5102-triage.htm>> Consulta [2-12-2008].
- [18] feature driven development [Web en línea]. <<http://www.featuredrivendevelopment.com/l/>>. [Consulta: 04-05-2009].
- [19] Standar Organization –HL7 [web en línea] <<http://aspe.hhs.gov/sp/nhii/standards.html#HL7>> Consulta [27-08-2008]
- [20] Migración a PostgreSQL desde otras bases de datos [Web en línea]. <<http://www.dbrunas.com.ar/postgres/migrapg.pdf>>. [Consulta: 04-01-2009].
- [21] Guía del Administrador de PostgreSQL [Web en línea]. <<http://palomo.usach.cl/Docs/postgres/Postgres-Admin.pdf>>. [Consulta: 04-01-2009].
- [22] <<http://www.minproteccionsocial.gov.co/VBeContent/library/documents/DocNewsNo368011.pdf>>. [Consulta: 04-04-2009].
- [23] Bases de datos para móviles (J2ME) Parte III [Web en línea]. <<http://bertiente.files.wordpress.com/2007/09/db2.jpg>>. [Consulta: 04-06-2009].
- [24] Herramientas Para Base de Datos Móviles [Web en línea]. <<http://www.slideshare.net/natalialuva/herramientas-para-base-de-datos-mviles-508288?nocache=4557&src=embed>>. [Consulta: 04-06-2009]
- [25] Introduction to SOAP [Web en línea] <http://www.w3schools.com/soap/soap_intro.asp> Consulta [09-02-2009]
- [26] Tutorial de PostgreSQL [Web en línea]. <<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html>>. [Consulta: 04-01-2009]
- [27] Walsh, Norman. “A technical introduction to XML” [Web en línea] <<http://www.xml.com/pub/a/98/10/guide0.html>> [Consulta: 06-02-2009]
- [28] Sun Developer Network, “Why Java Me Platform?” [Web en línea] <<http://java.sun.com/javame/overview/why.jsp>> [Consulta: 04-04-2009]

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en los procedimientos de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.