

Arquitectura para la Creación de Servicios Ubicuos Orientados a Salud¹

Architecture for the Creation of Ubiquitous Services Devoted to Health

Oscar Hernán Mondragón Martínez

Magíster en Sistemas Inalámbricos

Ingeniero en Electrónica y Telecomunicaciones

Docente Universidad Autónoma de Occidente

Grupo de Investigación en Telemática e Informática Aplicada, Facultad de Ingeniería

ohmondragon@uao.edu.co

Zeida María Solarte Astaiza

Magíster en Ingeniería Telemática

Especialista en Redes y Servicios Telemáticos

Ingeniera en Electrónica

Docente Universidad Autónoma de Occidente

Grupo de Investigación en Telemática e Informática Aplicada, Facultad de Ingeniería

zsolarte@uao.edu.co

Recibido Mayo 16 de 2011 – Aceptado Noviembre 30 de 2011

RESUMEN

Los sistemas ubicuos tienen diversas aplicaciones en diferentes campos tales como la industria, la educación y la salud. Sirven de apoyo a los profesionales de la salud para mejorar la atención, a través del acceso eficiente a información fundamental para el diagnóstico, tratamiento y seguimiento del paciente.

La evolución actual de las redes de datos cableadas e inalámbricas en cuanto a confiabilidad y ancho de banda soportado, que permiten usarlas de manera eficiente para el transporte de tráfico multimedia (voz, audio

1 Producto derivado del Proyecto de Investigación “PLATAFORMA PARA EL DESARROLLO DE SERVICIOS UBICUOS ORIENTADOS A SALUD”, proyecto avalado por la Dirección de Investigación y Desarrollo Tecnológico, Universidad Autónoma de Occidente

y video), así como el avance en la construcción de sensores y demás dispositivos inalámbricos cada vez más eficientes en términos de consumo de batería, hacen de éste un momento muy propicio para el desarrollo de sistemas ubicuos.

El presente artículo muestra el diseño de una arquitectura común para el desarrollo de sistemas ubicuos en el dominio de la salud que permite el monitoreo de variables fisiológicas y de contexto de pacientes bajo control clínico.

Palabras clave: computación ubicua, composición de servicios, e-salud, arquitectura de software, servicios web.

ABSTRACT

Ubiquitous systems have several applications in many fields such as industry, education and health. In terms of health, they are used to support both the patient and health professionals in order to improve healthcare, through efficient access to critical information, diagnosis, treatment, and monitoring of the patient.

Ubiquitous computing is a concept that has gotten great importance around the world where major projects about this topic are being developed. Current developments in data wired and wireless networks in terms of reliability and bandwidth support, that have allowed the efficient use of these networks for the transport of multimedia traffic, as well as progress in building sensors and other wireless devices becoming more efficient in terms of battery consumption, have propitiated the development of ubiquitous systems.

This article shows the design of a common architecture for the development of ubiquitous systems in the health domain for monitoring physiological and context variables of patients under clinical control.

Key words: ubiquitous computing, service composition, e-health, software architecture, web services.

1. INTRODUCCIÓN

El vertiginoso avance de las tecnologías electrónicas ha hecho que diferentes procesos en el mundo actual sean cada vez más digitales. En

ellos, gran variedad de dispositivos diseñados para ayudar a automatizar tareas desarrolladas por personas, enriquecen la interacción social.

A nivel internacional el ambiente hospitalario no es ajeno a este avance tecnológico, los sistemas de comunicación inalámbricos han posibilitado el acceso de información clínica a distancia, aumentando la rapidez y disponibilidad de los servicios. Es importante hacer uso de este tipo de tecnologías en el entorno hospitalario nacional y regional, ya que posibilitarían el monitoreo remoto de pacientes y la agilización de los diagnósticos, descongestionando dichos servicios de los centros hospitalarios y teniendo así un impacto positivo en los mismos.

El desarrollo de sistemas ubicuos depende del avance en los diferentes campos que estos involucran, los cuales incluyen hardware, software, redes, protocolos y estándares; sin embargo, cada uno de estos campos avanza a su propio ritmo y genera sus propios productos, haciendo necesario establecer formas apropiadas para su interrelación, es decir, pensarlos como un sistema integrado y no como partes individuales. Es así como la construcción de plataformas para el desarrollo de aplicaciones permite a la comunidad que trabaja en el área avanzar de forma más rápida y efectiva, por cuanto facilita el trabajo colaborativo y genera una base probada para el desarrollo.

Lo anterior hace que surja la posibilidad de integrar sensores y dispositivos móviles en una plataforma ubicua consciente del contexto para el monitoreo, diagnóstico y tratamiento de pacientes con alguna enfermedad, lo que se convierte en el objetivo principal del proyecto ***“Plataforma para la creación de servicios ubicuos orientados a salud”***, cuya arquitectura de referencia se expone en el presente artículo.

2. MARCO TEÓRICO

2.1. Computación ubicua

El término “Computación Ubicua” fue introducido por Mark Weiser en 1991 (Weiser, 1991), quien la describía como computadores muy pequeños con capacidad de comunicación y de computación que se incrustarían de forma casi invisible en cualquier tipo de dispositivo cotidiano. Estos dispositivos se encontrarían por todas partes y se integrarían de forma amigable con los humanos, haciéndolos casi imperceptibles para ellos. Este tipo de computación permitiría que servicios y aplicaciones que

residieran en dispositivos móviles o fijos se ejecutaran de acuerdo al contexto del usuario sin que éste los activara, y que dichos dispositivos tuvieran la capacidad y la inteligencia de regular el procesamiento y el intercambio de información de acuerdo a las necesidades circunstanciales del mismo (Ishii H, 2004).

La visión de Weiser se está convirtiendo en una realidad y se puede ver reflejada en la gran variedad de dispositivos que las personas llevan consigo en cualquier momento, las que permiten acceder a diferentes tipos de aplicaciones y servicios sin importar el lugar, el momento y el dispositivo. Esta visión implica retos, algunos de los cuales ya han sido afrontados en el proceso de maduración que han sufrido sus dos antecesores: Sistemas Distribuidos y Computación Móvil.

Considerando las teorías de Satyanarayanan (2001), las características más relevantes de los sistemas distribuidos que se retomaron en la computación ubicua son la comunicación remota, la tolerancia a fallos, la alta disponibilidad, el acceso a información remota y la seguridad. Aunque en la computación móvil se conservan muchos de estos conceptos, la movilidad trajo consigo nuevos retos, especialmente por las limitaciones en peso, tamaño, consumo de energía y procesamiento que tienen los dispositivos móviles.

Es así como se hace evidente que el desarrollo de la computación ubicua requiere el manejo de una gran cantidad de aspectos relacionados principalmente con la heterogeneidad, el dinamismo de los ambientes y la necesidad de centrarse en los usuarios. De esta manera, uno de los mayores retos a solucionar en este tipo de ambientes, es facilitar a los usuarios la realización de las tareas que tengan relación con los servicios de red, de tal forma que puedan ejecutarse de acuerdo a las especificidades de los ambientes ubicuos (Ben Mokhtar et al., 2006).

Los dispositivos de red deberán tener, por tanto, la capacidad de publicar sus servicios para que puedan ser usados por los usuarios; además, la red debe estar habilitada para localizar, invocar y componer servicios que cumplan con los requerimientos de los usuarios.

De acuerdo a lo anterior se pueden mencionar como características más relevantes de la computación ubicua las siguientes:

- **Transparencia:** la computación ubicua deberá facilitar las tareas de

una forma no intrusiva de tal manera que oculte al usuario toda la tecnología subyacente (Barkhuus, 2002).

- **Movilidad:** entendida como la habilidad para comunicarse con otras personas, acceder y procesar información simultáneamente mientras la persona se encuentra en movimiento; por esto, los servicios ubicuos deben tener en cuenta aspectos como calidad de servicio en las áreas de cobertura y transferencias de código de un dispositivo a otro durante la ejecución de una tarea (Sun and Sauvola, 2002).
- **Sensibilidad al contexto:** se define como la propiedad de un sistema para usar el contexto para proveer información o servicios relevantes al usuario, donde la relevancia depende de las tareas ejecutadas por el usuario (El-Sayed and Black, 2006).

2.2. Composición de servicios

La composición de servicios consiste en la agrupación de un número de servicios existentes para proveer uno nuevo con más funcionalidades, creado de la manera más adecuada para satisfacer los requerimientos de un usuario (Hu, 2004). La aparición de los servicios web ha facilitado el proceso de composición de servicios de valor agregado dadas sus características de interoperabilidad, estandarización, independencia de plataformas; se basan en protocolos ampliamente usados como http y tcp (Alonso, 2004).

Los términos Orquestación y Coreografía describen dos aspectos muy importantes para la composición de servicios (Alvez, 2006). Un proceso se puede considerar una orquestación de servicios cuando es controlado totalmente por una única entidad. Este proceso define completamente las interacciones con los servicios componentes y la lógica requerida para conducir correctamente esas interacciones. Un proceso es una coreografía de servicios cuando define las colaboraciones entre cualquier tipo de aplicaciones componentes, independientemente del lenguaje o plataforma en el que estén definidas las mismas (Alvez, 2006). Un proceso de coreografía no es controlado por uno solo de los participantes.

Una de las principales características de la composición de servicios es que existe una comunidad científica interesada en estandarizar todo los procesos asociados a ella.

Relacionados con los servicios Web, la motivación básica de estándares como XML, SOAP y WSDL es permitir un alto grado de flexibilidad para lograr la composición de servicios complejos. Por su parte, UDDI habilita su descubrimiento, ya sea manual o automático, y facilita la construcción de servicios Web compuestos.

Construidos sobre los estándares mencionados, surge para el proceso de orquestación, el lenguaje Business Process Execution Language for Web Services (BPEL4WS, BPELWS comúnmente denominado BPEL) el cual define una notación estándar para especificar el comportamiento de un proceso de negocio basándose en servicios web (Hackmann, 2007). BPEL combina las características de flujo del Web Service Flow Language (WSFL) de IBM y las características de bloque del Web Services for Business Process Design (XLANG) de Microsoft.

3. ARQUITECTURA PARA EL DESARROLLO DE SERVICIOS UBICUOS ORIENTADOS A SALUD

3.1. Arquitectura lógica

La arquitectura propuesta ofrece componentes básicos reutilizables para la gestión del contexto de los usuarios, la gestión de variables fisiológicas, gestión de acceso a sensores, la gestión de comunicaciones y la gestión de la historia clínica digital, como se muestra en la figura 1.

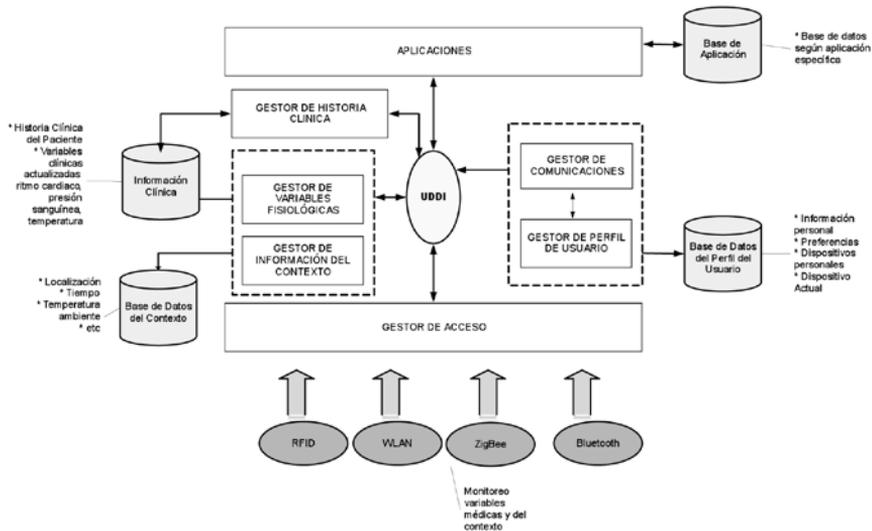


Figura 1. Arquitectura de la plataforma

En el nivel superior de la arquitectura se brindan los mecanismos para la composición de servicios a través de la orquestación de los componentes, lo que permitirá la utilización de estos para configurar aplicaciones avanzadas.

Las funcionalidades de cada uno de estos componentes son:

- **Gestor de acceso:** sirve como interfaz entre los diferentes sensores que monitorean las variables fisiológicas ó de contexto y los componentes que usan dicha información para desarrollar sus funciones.
- **Gestor de variables fisiológicas:** encargado de recibir actualizaciones del estado de las variables fisiológicas del paciente y alimentar la base de conocimiento para tomar decisiones tales como notificar al médico de dichos cambios o alertar al paciente sobre alguna anomalía. Las variables consideradas son: respiración, ECG y temperatura.
- **Gestor de información del contexto:** encargado de recibir las variables del contexto tales como localización, temperatura ambiente, humedad, aceleración, entre otras. Dicha información permitirá a una aplicación basada en la arquitectura propuesta, tomar decisiones oportunas. Un ejemplo es el envío de una ambulancia desde el centro hospitalario más cercano a la ubicación del paciente. Las variables consideradas son: localización, tiempo, temperatura y humedad.
- **Gestor del perfil del usuario:** utilizado para administrar la información del usuario, tal como datos personales y preferencias. Esta información junto con la del contexto permitirá a las aplicaciones personalizar la forma en que interactúan con el usuario del sistema.
- **Gestor de comunicaciones:** dado que en un sistema ubicuo el usuario puede acceder en cualquier momento y desde cualquier dispositivo a sus servicios, este gestor permite enviar la información de la manera más adecuada, dependiendo del dispositivo actual que esté utilizando el usuario.
- **Gestor de historia clínica:** permite el acceso a la historia clínica actualizada del paciente desde cualquier dispositivo, siguiendo los estándares definidos por HL7 (Biron, 2006). La historia clínica sólo podrá ser visualizada / modificada por personal autorizado.

3.2. Arquitectura de despliegue

Se propone una arquitectura de despliegue consistente de los siguientes módulos: red de sensores, coordinador PAN, teléfono, móvl, cliente web (PC/Laptop), servidor web, servidor de aplicaciones, bases de datos, otros servicios web. Ver figura 2.

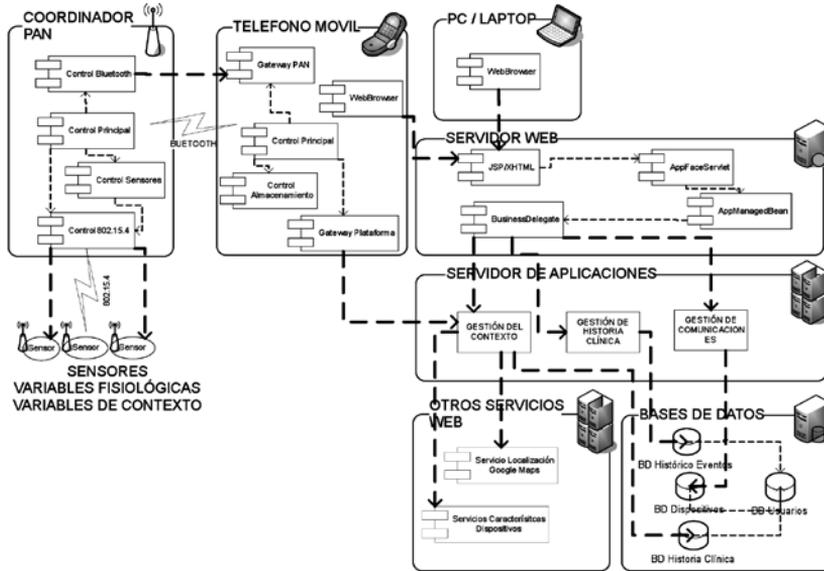


Figura 2. Arquitectura de despliegue

- **Red de Sensores:** conformada por los sensores que hacen mediciones de las variables fisiológicas y de contexto del paciente y las reportan al coordinador de la red de área personal (Coordinador PAN).
- **Coordinador PAN:** módulo que sirve de puente entre la red de sensores y el teléfono móvil. Consta de los siguientes submódulos: Control 802.15.4: encargado de la comunicación con los sensores (transmisión y recepción de información) a través del protocolo 802.15.4. Control Sensores: encargado del envío de comandos hacia los sensores y de la recepción de eventos generados por ellos. Control Principal: módulo encargado de coordinar todas las funciones realizadas por el coordinador. Control Bluetooth: módulo interfaz hacia el dispositivo móvil a través del protocolo 802.15.1.
- **Teléfono móvil:** dispositivo móvil del usuario que interactúa con el coordinador PAN para recibir notificaciones de eventos desde los

sensores y enviar comandos hacia ellos. Consta de los siguientes submódulos: Gateway PAN: interfaz hacia el coordinador de la red de área personal a través de bluetooth. Control de almacenamiento: brinda los mecanismos de almacenamiento de datos relevantes del sistema en caso de pérdida de conectividad hacia la red de sensores. Gateway plataforma: interfaz hacia el servidor de aplicaciones a través de una red inalámbrica ya sea de área local ó extensa. Web browser: interfaz hacia el servidor web.

- **PC/Laptop:** provee un cliente web para el acceso a los servicios y la realización de operaciones de gestión de la información.
- **Servidor Web:** provee la capa de presentación del sistema hacia los clientes web y la conectividad hacia el servidor de aplicaciones para consumir los servicios web que éste ofrece. Se implementa aquí el framework de JSF (Java Server Faces) y los siguientes patrones de diseño: Front controller: este patrón es implementado por un componente web, Java Servlet, el cual es el único punto de entrada para todas las solicitudes HTTP (HyperText Transfer Protocol) que realizan los clientes web de los módulos dispositivo móvil y PC/Laptop.

Dado que se está utilizando JSF como framework para construcción de la capa de presentación, el FrontController estaría representado por un FaceServlet el cual delega el procesamiento de solicitudes a un ManagedBean, quien a su vez realiza los llamados a los servicios de negocio a través de un BusinessDelegate, el cual abstrae la complejidad de la localización de componentes y manejo de excepciones remotas. Sirve para ejecutar los servicios de negocios expuestos por los componentes de fachada en el servidor de aplicaciones. Existe una correspondencia uno a uno entre los métodos definidos en la interfaz remota de la fachada con los del Business delegate.

- **Vista:** ofrece una interfaz gráfica que permite la interacción con los servicios de la plataforma a través de los Managed Beans encargados de recibir las acciones ejecutadas por los usuarios. A este nivel no se implementa lógica de negocio.
- **Servidor de aplicaciones:** es el núcleo de la plataforma. Contiene la lógica del negocio de los servicios que serán ofrecidos para que se construyan aplicaciones a partir de estos. Aloja los gestores del contexto, comunicaciones y de historia clínica. Ver figura 3.

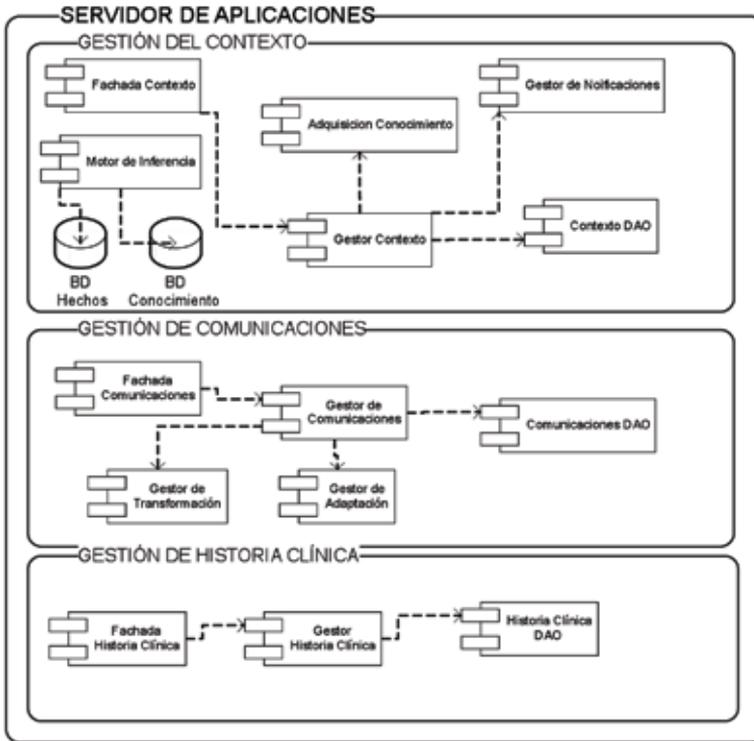


Figura 3. Servidor de aplicaciones

En todos los gestores se implementan los siguientes patrones de diseño:

- **Session Facade:** expone todos los servicios de negocios para encapsular la complejidad de interacciones entre los componentes definidos.
- **Value object:** encapsula la información que se comparte entre los objetos de las diferentes capas de la plataforma.
- **Data access object (DAO):** abstrae y encapsula los detalles de acceso a la fuente de datos a través de un objeto dedicado.
- **Bases de datos:** contiene las entidades que almacenan información útil de la plataforma: base de datos de histórico de eventos, base de datos de dispositivos, base de datos de historia clínica y base de datos de usuarios.

- **Otros servicios web:** este modulo provee conectividad a servicios web externos, tales como el servicio de localización de Google Maps y repositorios de características de dispositivos.

3.3. Escenario de aplicación

Para lograr un mejor entendimiento de la arquitectura propuesta, se hace a continuación, la descripción de un escenario donde se muestran las interacciones entre sus módulos.

Paso 1: una persona conduce su automóvil rumbo al trabajo. Un sensor instalado en su cuerpo detecta una anomalía en la presión sanguínea.

Paso 2: inmediatamente se envía una alarma al smartphone del paciente el cual lo alerta a través de un mensaje de texto. **Paso 3:** se despliega un mapa con los centros asistenciales más cercanos. **Paso 4:** el paciente elige un centro asistencial y se dirige a él. Seguidamente, el sistema envía al centro la información del paciente incluida la anomalía que presenta, datos personales y localización. Finalmente un profesional de la salud apto para tratarlo lo estará esperando a su llegada.

Interacción entre los Módulos de la Arquitectura

Paso 1: **1.** El sensor de presión sanguínea envía el valor medido al paciente a través del protocolo 802.15.4; **2.** El módulo 802.15.4 recibe el dato y usa un método del módulo principal para reportar el evento; **3.** El módulo de control principal solicita al módulo de control bluetooth el establecimiento de la conexión con el teléfono móvil; **4.** El módulo de control bluetooth establece la conexión con el teléfono móvil y envía el dato al Gateway PAN del móvil; **5.** El Gateway PAN recibe el dato a través de bluetooth y usa un método del control principal para reportarlo; **6.** El control principal solicita al Gateway- plataforma la localización del servicio ofrecido por el gestor del contexto; **7.** El Gateway-Plataforma devuelve la referencia remota del componente de gestión de contexto; **8.** Se consume el servicio desde el control principal, pasando el dato medido a través de un método definido en el servicio; **9.** Se registra el evento en la base de hechos; **10.** El evento es analizado por el motor de inferencia para establecer alguna correspondencia: se consultan las reglas de la base del conocimiento y los hechos almacenados en la base de datos de hechos; **11.** En el motor de inferencia del gestor de acceso se dispara una regla y notifica de esto al módulo de suscripción/notificación. En este módulo se verifica qué aplicación se suscribió al evento y ésta es notificada; **12.** Se utiliza el Data

Access Object para acceder a la base de datos de histórico de eventos;
13. Se guarda el evento anormal en la base de datos de histórico.

Paso 2: **1.** La aplicación que recibe el evento localiza la referencia remota del servicio ofrecido por el gestor de comunicaciones a través de un BusinessDelegate; **2.** La aplicación consume el servicio del gestor de comunicaciones a través del cual se enviará el SMS al paciente para alertarlo de la anomalía; **3.** Se usa el DAO de comunicaciones para consultar las preferencias del usuario y establecer a qué dispositivo se debe enviar el mensaje; **4.** Se usa el gestor de adaptación para adaptar el contenido de acuerdo al dispositivo encontrado en las preferencias; **5.** Se envía el mensaje adaptado (SMS, email, etc.)

Paso 3:**1.** La aplicación localiza la referencia remota del servicio ofrecido por el gestor de contexto a través de un BusinessDelegate; **2.** La aplicación consume el servicio ofrecido por el gestor de contexto para solicitar la localización del paciente y de los centros asistenciales más cercanos; **3.** Se usa el GPS del smartphone para determinar las coordenadas del paciente. Se utiliza un servicio de localización basado en las coordenadas del usuario. Por comparación de coordenadas se buscan y se retorna la ubicación de los centros asistenciales más cercanos. **4.** La aplicación localiza la referencia remota del servicio ofrecido por el gestor de comunicaciones a través de un BusinessDelegate; **5.** La aplicación consume el servicio ofrecido por el gestor de comunicaciones y le envía las coordenadas de los centros asistenciales previamente calculadas; **6.** Se usa el DAO de comunicaciones para consultar las preferencias del usuario; **7.** Se consultan las preferencias del usuario para establecer a qué dispositivo se debe enviar el mensaje; **8.** El Gestor de comunicaciones consume el servicio de Google Maps para localizar en un mapa los centros de acuerdo a las coordenadas suministradas; **9.** Se usa el gestor de adaptación para adaptar el contenido de acuerdo al dispositivo encontrado en las preferencias; **10.** Se envía el mensaje adaptado (SMS, MMS).

Paso 4:**1.** La aplicación localiza la referencia remota del servicio ofrecido por el gestor de comunicaciones a través de un Business Delegate; **2.** La aplicación consume el servicio del gestor de comunicaciones para enviar la información del paciente al centro asistencial seleccionado por el mismo; **3.** Se usa el DAO de comunicaciones para consultar las preferencias del centro asistencial; **4.** Se consultan las preferencias del centro asistencial para establecer a que dispositivo se debe enviar el mensaje; **5.** Se usa el gestor de adaptación para adaptar el contenido de acuerdo al dispositivo

encontrado en las preferencias; **6.** Se envía el mensaje adaptado al centro asistencial (SMS, email, etc.); **7.** Finalmente un profesional de la salud apto para tratarlo lo estará esperando a su llegada.

4. CONCLUSIONES

La Arquitectura propuesta en el presente proyecto integra diferentes tecnologías necesarias para el desarrollo de sistemas ubicuos. Trabajos previos presentan cada uno de los componentes diseñados de manera separada, lo que hace difícil y muchas veces ineficiente su integración para la construcción de sistemas de mayor escala. Aunque la arquitectura permite desarrollar cualquier tipo de sistema ubicuo, se especializa en sistemas ubicuos para la salud, lo que garantiza que se tengan en cuenta aspectos de normatividad del sector y el adecuado manejo de las variables fisiológicas medidas en los pacientes. La plataforma desarrollada para validar la arquitectura puede ser aplicada en contextos diferentes al planteado en este proyecto, mediante ajustes relacionados con los nuevos contextos de aplicación, ampliando así las posibilidades del mismo y proveyendo toda una base de experimentación alrededor de la computación ubicua.

Es la primera vez que se desarrolla un proyecto de este alcance en el país en el área de los sistemas ubicuos para salud y con su desarrollo se espera sentar un precedente para trabajos futuros relacionados.

La arquitectura propuesta pretende servir como base para el desarrollo de nuevas aplicaciones que faciliten el acceso a servicios en el campo de la salud, descongestionando los centros asistenciales y permitiendo a los pacientes acceder de manera más oportuna a estos servicios.

El diseño orientado a servicios permite el desarrollo de aplicaciones avanzadas de manera más eficiente mediante la reutilización de funcionalidades comunes.

En cada uno de los niveles de la arquitectura se utilizaron las mejores prácticas de diseño, haciendo uso de patrones de diseño suficientemente probados, lo cual garantiza la robustez del sistema.

5. AGRADECIMIENTOS

Agradecemos a la Dirección de Investigaciones de la Universidad Autónoma

de Occidente y a todo el equipo de trabajo del proyecto de investigación “Plataforma para el desarrollo de servicios ubicuos orientados a salud”, especialmente a los profesores investigadores Diego F. Almario, Lyda Peña, Paulo Calvo, y a los estudiantes y colaboradores que participaron en este proyecto.

BIBLIOGRAFÍA

- Alonso, G., Casati, F., Kuno, H., Machiraju, V. 2004. **“Web Services Concepts, Architectures and Applications”**. ISBN: 3-540-44008-9.
- Alvez, P. Foti P. Proyecto Batuta. 2006. **Generador de Aplicaciones Orquestadoras. Estado del Arte**. Universidad de la Republica.
- Barkhuus, L. (2002). **Ubiquitous Computing: Transparency in Context-Aware Mobile Computing**. In. Proc. Of UbiComp.
- Ben Mokhtar, S., N. Georgantas, and V. Issarny (2006). **Cocoa: Conversation-Based Service Composition in Pervasive Computing Environments**. In Proc. of ICPS.
- Biron Paul V. Shabo Amnon. 2006. **HL7 Clinical Document Architecture, Release 2**. JAMIA. 13:30 – 39.
- El-Sayed, A.-R, and Black J. 2006. **Semantic-based Context-aware Service Discovery in Pervasive-computing Environments**. In Proc. of SIPE.
- Hackmann, G., C. Gill, and G.-C. (2007). **Extending BPEL for Interoperable Pervasive Computing**. In Proc. of ICPS. Roman.
- HU, M. 2004. **Web Services Composition, Partition, and Quality of Service in Distributed System Integration and Re-engineering**. Proceedings by deepX Ltd.
- Ishii, H. 2004. **“Bottles: A Transparent Interface as a Tribute to Mark Weiser”**. IEICE Transactions Inf. And Syst, vol.E87-D, no.6, pp.1299-1311.
- Paradise, R. 2004. **Wearable Health Care System for Vital Signs Monitoring**. Medicon conference.

- Satyanarayanan, M. 2001. ***“Pervasive Computing: Vision and Challenges”***. Personal Communications, IEEE, vol.8, no.4, pp.10 - 17.
- Sun, J., Sauvola J. 2002. ***On Fundamental Concept of Mobility for Mobile Communications. In Proc. Of Mobile Radio Communications, 2002.*** The 13th IEEE International Symposium.
- Weiser, M. 1991. ***The Computer for the Twenty-First Century.*** Scientific American, pp. 94-10.