

SLD017 ARQUITECTURA SOA BASADA EN COMPONENTES DE SOFTWARE DE CODIGO ABIERTO PARA LA IMPLEMENTACION DE HISTORIA CLINICA ELECTRONICA

SLD017 SOA ARCHITECTURE BASED ON OPEN SOURCE SOFTWARE COMPONENTS FOR IMPLEMENTATION OF ELECTRONIC HEATH RECORD

Salvador Rodríguez Loya¹, Yolanda Rodríguez Loya²

1 University Of Sussex, Reino Unido, s.rodriguez-loya@sussex.ac.uk, Shawcross Building, Brighton, BN1 9QT, United Kingdom

2 Instituto Tecnológico de Matamoros, México, yol.nada@hotmail.com

RESUMEN: Los avances médicos han permitido aumentar la esperanza y la calidad de vida. Desafortunadamente, los costos de los servicios de salud se han incrementado en los últimos años y los errores clínicos son peligrosamente comunes. La implementación de sistemas de historia clínica electrónica promete beneficios que incluyen una mejor atención y disminución de costos de atención medica. Dos factores principales que impiden la implementación del historial clínico electrónico son el alto costo de este tipo de sistemas y la falta de interoperabilidad de los sistemas existentes. El presente artículo muestra una arquitectura construida totalmente a partir de software de código abierto que ofrece una solución contra la falta de interoperabilidad y alto costo. La plataforma está basada en los principios SOA (Service Oriented Architecture) e incluye BPM (gestión de procesos de negocio) y sistema de ayuda a la toma de decisiones clínicas (CDSS).

Palabras Clave: Código Abierto, Historial Médico Electrónico.

ABSTRACT: Medical Advances have increased life expectancy and quality of life. Unfortunately, costs of medical services have increased in recent years and clinical errors are dangerously common. The implementation of electronic health record systems promises benefits that includes better care and decreased healthcare costs. Two main factors that impede the implementation of electronic health records are high cost of such systems and the lack of interoperability between existing systems. This paper presents an architecture built entirely from open source software that brings a solution for lack of interoperability and high cost. The platform is base on Service Oriented Architecture (SOA) principles and includes Business Process Management (BPM) and Clinical Decision Support System (CDSS).

KeyWords: Open Source, Electronic Heath Record.

1. INTRODUCCIÓN

En año 2000 *The National Academics* (Instituto líder de investigación médica en los estados unidos) publicó el reporte llamado “*To Err is human*” determinando que los errores clínicos son un problema de gran magnitud, indicando que aproximadamente 98,000 estadounidenses mueren por esta razón cada año [1]. De acuerdo con el reporte “*Health at*

the Glance” publicado por *Organization for Economics Cooperation and Development* (OECD) en el 2011, el gasto por los servicios de salud a nivel mundial ha aumentado más rápido que el Producto Interno Bruto (PIB) [2]. Los gastos en el Reino Unido excedieron el promedio de los miembros de OECD por primera vez en el 2009 y Estados Unidos fue el país con el nivel más alto de gastos de servicios de salud por habitante comparado con cual-

quier otro país [2]. Otros países ricos gastan menos en servicios de salud y aun así logran mantener niveles similares del cuidado de salud y con una mejor calidad en el servicio [3].

Los sistemas de historia clínica electrónica (EHR) prometen atacar los problemas de reducción de costos y mejora del servicio. Existe evidencia que muestra que la implementación de registros computarizados y la automatización de procesos en el área de salud, puede reducir dramáticamente errores y costos [4] [5]. A pesar de que la mayoría de hospitales han automatizado tareas en diferentes departamentos como radiología, laboratorio y farmacia, muy pocos han computarizado la información más importante que poseen: "la historia clínica del paciente" [6]. Comparado con otras industrias el sector salud se ha negado a automatizar sus procesos por décadas [7] [6]. Una de las barreras principales que impiden la implementación de EHR es la falta de interoperabilidad entre los sistemas de información [8].

2. INTEROPERABILIDAD

Las computadoras empezaron a utilizarse en la industria de salud a inicios de 1960 cuando un grupo de hospitales comenzaron a automatizar algunas actividades administrativas principalmente nomina y funciones de control de cuentas de los pacientes [9]. Estos sistemas corrían en modelos centralizados denominados "mainframes". Gradualmente avances en la tecnología permitieron la implementación de sistemas de información en otras áreas de la organización como laboratorio, radiología y farmacia. Un repaso cronológico de los acontecimientos de la evolución de las tecnologías de información provee una buena comprensión de la diversidad de tecnologías y arquitecturas que coexisten hoy en día en la industria. Dos progresiones de arquitectura conceptuales se han identificado: progresión de arquitectura del lado del servidor y progresión de arquitectura del lado del cliente [10]. Eventos importantes de la arquitectura de lado del servidor incluyen la era del mainframe, era del cliente/servidor, era de los sistemas distribuidos y la era del internet. De igual manera eventos relevantes de la arquitectura del lado del cliente incluyen: Terminales, *Thick clients*, *Thin clients*, *Browser clients* y clientes móviles. Además de estas arquitecturas, existe una gran diversidad de aplicaciones desarrolladas en distintos lenguajes como Java, .Net y COBOL, las cuales corren en plataformas incompatibles [11]. Es importante mencionar que muchas de estas tecnologías que hoy se consideran inadecuadas aun continúan en uso debido al alto costo que representa migrar a nuevas plataformas o porque no se ha alcanzado el retorno de inversión [8] [11].

El historial clínico electrónico (EHR) está compuesto de información contenida en diferentes sistemas

distribuidos dentro y fuera de la organización [12]. Uno de los retos a que se enfrenta la industria para lograr implementar del historial clínico electrónico es la falta de interoperabilidad entre la gran cantidad de tecnologías existentes [13] [14]. Se define como Interoperabilidad, a la capacidad de dos o más sistemas de intercambiar y procesar información. De acuerdo con *Health Level Seven International* (HL7), interoperabilidad está compuesta de tres partes: técnica, semántica y procesos. La interoperabilidad técnica está relacionada con el transporte físico de los datos médicos y la seguridad de dicho transporte, la interoperabilidad semántica se relaciona con la comunicación de significado coherente, y la interoperabilidad de procesos se asegura que la información es provista al momento necesario, en una forma de eventos o de una manera secuencial que puede ser utilizada en el flujo actual de trabajo del grupo que provee el servicio de salud [15]

3. ARQUITECTURA PROPUESTA

Service Oriented Architecture (SOA) es un paradigma de arquitectura que facilita la reutilización de componentes de software, al organizar funciones discretas contenidas en aplicaciones, en servicios descubribles, autónomos, interoperables y basados en estándares que pueden combinarse fácilmente para cumplir con la necesidades de la organización [16]. Estos servicios son módulos bien definidos y auto contenidos que proveen funcionalidades estandarizadas de negocios y están ligadas entre sí, para construir un proceso de negocio integral. Organizaciones que adoptan SOA pueden reducir costos, obtener un alto retorno de inversión, reutilizar e integrar servicios y sistemas heredados, reducir tiempo de desarrollo y alinear el negocio con las tecnologías de información (IT) [17].

La siguiente figura muestra la arquitectura SOA propuesta para la implementación del expediente clínico electrónico. Esta arquitectura SOA incluye gestión de negocios (BPM) y sistema de ayuda a toma de decisiones clínicas (CDSS).

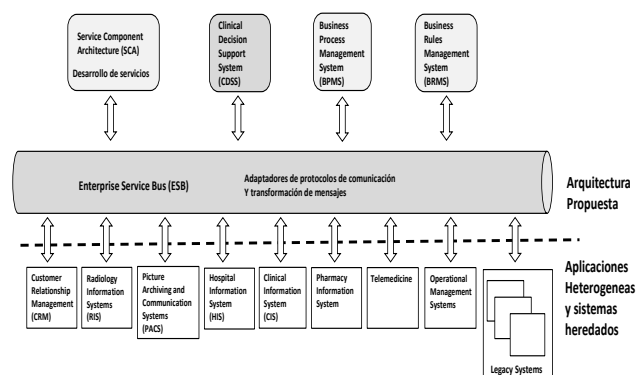


Figura. 1: Arquitectura Propuesta

La figura 1 muestra la arquitectura propuesta para la implementación del expediente clínico electrónico. Las flechas de doble punta representan el flujo de datos, mientras que los rectángulos inferiores de la figura separados por la línea punteada representan los sistemas de información que comúnmente se encuentran en hospitales y organizaciones prestadoras de servicios de salud. En la siguiente sección se da una breve descripción de cada uno de estos sistemas.

3.1 Sistemas de Información que comúnmente se encuentran en las instituciones de asistencia médica

Customer Relationship Management System (CRMS).- Estos sistemas suelen ser utilizados en los hospitales para obtener información sobre las necesidades del paciente y lograr satisfacer sus demandas [18]. De acuerdo a Gartner Group, *Customer Relationship Management* es una estrategia de negocios enfocada al cliente y diseñada para optimizar los ingresos, rentabilidad y satisfacción del cliente [19]. En el punto de vista de negocio, el paciente representa el mayor cliente de la organización prestadora de servicios de salud [20]. De acuerdo con Melissa Lin [21] la mayoría de CRMS ofrece las siguientes funcionalidades:

- Automatización de ventas: Permite automatizar tareas relacionadas a las ventas como la interacción con el cliente, administración de contactos, campañas de ventas y actividades de promoción, monitoreo de ventas, tendencias, control de inventario, etc. La mayoría de CRMS también ofrecen capacidades móviles para proveer al grupo de trabajo soporte en cualquier lugar y en cualquier momento, permitiendo acceso a la información del cliente a través de laptops, teléfonos celulares y otros aparatos móviles.
- Automatización de mercadotecnia.- Permite automatizar tareas como análisis y planeación de mercadotecnia, actividades relacionadas a compañías de mercadotecnia, promoción de productos, tendencias del mercado, etc.
- Servicio al cliente y soporte.- Permite documentar y administrar la información del cliente. El CRMS ayuda a los equipos del centro de llamadas a solucionar y rastrear rápidamente los problemas del cliente desde el momento en que son reportados y hasta que son resueltos.
- Reporte y herramientas de análisis.- Estas herramientas son un grupo de software y tecnologías que ofrecen una gran visibilidad de las ventas, mercadotecnia e información del cliente. Con todo esto, los negocios pueden identificar

las necesidades, preferencias y comportamientos del cliente, lo que permite identificar nuevas tendencias de negocio, oportunidades y cubrir problemas o áreas de mejora.

Radiology Information Systems (RIS).- Estos sistemas están diseñados para el departamento de radiología y permiten la adquisición y análisis de imágenes radiológicas. Los sistemas RIS ofrecen las siguientes funciones [22]:

- Admisión médica: Cuenta con funciones para el registro de pacientes.
- Radiología: Recepción de ordenes
- Exámenes: Asignar ordenes al departamento o tratamiento correspondiente, proveer formatos para la escritura del reporte.
- Agenda de citas: Asignar pacientes a departamentos o tratamientos
- Código de diagnósticos y procedimientos: incluye catálogos de procedimientos de radiología
- Administración de instrumentos médicos: Gestión de tratamientos.
- Agenda de trabajo y administración de tiempos: Permite preparar la agenda de trabajo.

Picture Archiving Communication System (PACS).- De acuerdo con Rodríguez [23] los sistemas PACS se han desarrollado para atacar uno de los problemas de accesibilidad y requerimientos de red para el intercambio de imágenes médicas. Esencialmente, los PACS son sistemas que archivan una gran cantidad de imágenes digitales, las cuales de acuerdo con DICOM (Digital Imaging and Communication in Medicine) están en formato estándar. Estos sistemas pueden ser consultados simultáneamente por una gran cantidad de terminales, sistemas y redes. Se presume que las tecnologías de manejo de imágenes médicas mejoran la calidad del servicio al paciente y facilitan el flujo de trabajo.

Hospital Information System.- Los HIS apoyan las actividades operacionales, tácticas y a niveles estratégicos [23]. Las funciones típicas de los sistemas HIS son: registro de pacientes, ADT (Admisión, Descarga y Transferencia), pagos de los pacientes, índice de historial clínico, copia de seguridad/recuperación y administración de sistema. También pueden incluir módulos de farmacia, laboratorio, radiología, instalaciones, inventario, lavandería, cocina, ambulancia, seguridad, etc.

Clinical Information Systems (CIS).- Representan sistemas separados de departamentos de servicios clínicos especializados. Algunos ejemplos de sistemas CIS incluyen sistemas de monitoreo de pacientes y sistemas de documentación de anestesia. Estos sistemas operan en departamentos como

medicina interna, cardiología, neurología, obstetricia, cirugía y psiquiatría [23].

Pharmacy Information System.- Algunas de las funciones de estos sistemas es mantener información de los medicamentos y registros de los mismos por paciente. Además proveen recetas e información a doctores y enfermeras [23].

Telemedicine.- Este tipo de sistemas facilita la comunicación entre los médicos de atención primaria y especialistas así como con pacientes que se encuentran en lugares alejados. Esto permite a los médicos practicar la medicina a distancia [23].

Billing Systems.- Estos sistemas son utilizados para el control de pagos y envío de reclamos electrónicos a las compañías de seguros. En los Estados Unidos estos sistemas suelen ser distribuidos por Medicare y por compañías de seguros para motivar a los médicos a mandar reclamaciones electrónicas [24]. Además la mayoría de estos sistemas incluyen opciones de agenda electrónica y algunos proveen la habilidad de integrar registros médicos electrónicos si así se desea [24].

Operational Management Systems.- Apoyan actividades que no se relacionan con el cuidado del paciente. Ejemplos son: Sistemas de finanzas, nómina de empleados, control de compras e inventario y tareas de oficina [9].

Legacy Systems.- Se considera *Legacy Systems* (Sistemas heredados) a las soluciones de software y hardware que pertenecen a una generación pasada o era de innovación y que hoy se consideran inadecuados pero que continúan relevantes para la organización [25]. IBM publicó en un reporte del 2007 que 10,000 mainframes aun continuaban en uso a nivel mundial, mitad de los cuales pertenecen al área de finanzas y de seguros [26].

Los rectángulos que se encuentran sobre la línea punteada en la figura 1, representan los componentes principales de la arquitectura propuesta. Cinco tecnologías se han identificado como bloques esenciales para la creación de la arquitectura. Estas tecnologías son: gestión de procesos de negocios (BPM), gestión de reglas de negocio (BRM), bus de servicios de empresa (ESB), arquitectura SCA (*Service Component Architecture*) y sistema de ayuda a la toma de decisiones clínicas (CDSS).

3.2 Gestión de Procesos de negocios

Las organizaciones pueden obtener más beneficios cuando *Service Oriented Architecture* (SOA) es combinado con *Business Process Management* (BPM). Una arquitectura SOA enfocada a procesos permite optimizar continuamente los procesos de negocio, así la tecnología de información (IT) puede ser alineada más estrechamente con los objetivos de la organización. Esta clase de arquitectura in-

crementa la productividad, eficiencia, y flexibilidad a través de la gestión de procesos [27]. Un proceso de negocios es definido como un grupo de tareas estandarizadas y coordinadas para lograr una actividad específica de negocio que suele extenderse a lo largo de la organización [28]. En una arquitectura SOA enfocada a procesos, el sistema de gestión de procesos (BPMS) es utilizado para orquestar y definir la secuencia de los servicios disponibles [29].

Para agregar esta capacidad a nuestra arquitectura hemos elegido a jBPM [30] que es un sistema de gestión de procesos de código abierto que utiliza la notación estándar BPMN 2.0 (*Business Process Management Notation*). Es importante mencionar que existen varios lenguajes para representar los procesos de negocios, actualmente los dos lenguajes principales son BPEL (*Business Process Execution Language*) de OASIS y BPMN de OMG. Ambos fueron creados para atacar diferentes problemas y en ciertas áreas se traslapan. Al momento de esta publicación, BPMN 2.0 parece ser el lenguaje más robusto, de cualquier manera la arquitectura propuesta ofrece la flexibilidad de cambiar de BPMS sin necesidad de cambios mayores.

3.3 Gestión de Procesos de negocios

De acuerdo con la definición de Business Rules Group (BRG) [31] una regla de negocio es una declaración que define o restringe algunos aspectos del negocio. La regla de negocio intenta reafirmar la estructura del negocio, controlar o influenciar el comportamiento del negocio [32]. Un sistema gestor de reglas de negocio (BRMS) es una manera de automatizar y mejorar la toma de decisiones en el negocio. Fundamentalmente es acerca de la extracción de decisiones y reglas que se encuentran embebidas dentro de aplicaciones, personas o grupo de trabajo y sistemáticamente exponerlas como reglas que pueden ser administradas centralmente. El uso de BRMS puede brindar los siguientes beneficios:

- Reducción de costos.- el uso de un motor de reglas puede reducir la complejidad en el desarrollo de software ya que la creación de reglas dentro del BRMS es más fácil que escribirlas dentro del código de la aplicación [33].
- Reducción de costos de mantenimiento.- Cuando las reglas de negocio se encuentran dentro de las aplicaciones, cualquier pequeña modificación incurre en costos de recopilación, prueba e implementación. Mantenimiento de software se refiere a la corrección de errores y a la modificación de las aplicaciones existentes para llevar a cabo nuevas tareas o viejas tareas bajo nuevas circunstancias [34]. De acuerdo a Penny Grubb y otros, las organizaciones gastan

de 60 a 80 por ciento de sus recursos en el mantenimiento de software[35].

- Perpetuidad y compartición de reglas.- Si las reglas de negocio se encuentran dentro del código de una aplicación, estas pueden perderse cuando la aplicación se jubile o es remplazada por nuevas aplicaciones. Por otro lado si las reglas son administradas centralmente, estas se mantienen vivas y pueden ser reutilizadas y compartidas[36].
- Rendimiento.- Los motores de reglas generalmente ofrecen un alto rendimiento al utilizar algoritmos optimizados como el RETE[37] el cual puede filtrar un gran número de reglas a alta velocidad[38], replicar estas habilidades dentro del código de las aplicaciones puede ser retardador y muy costoso[39].
- Cumplimiento con requerimientos gubernamentales: En muchas industrias como el sector salud, los requerimientos de gobierno juegan un papel muy importante. Generalmente mostrar la implementación de controles es una prueba más efectiva de que la organización cumple con las regulaciones, que demostrar sus operaciones diarias[36].
- Calidad del software.- En la mayoría de los casos cuando las reglas de negocio son escritas por los desarrolladores de software, el producto final no refleja las necesidades de los usuarios finales, porque quien entiende el negocio no puede escribir el código de la aplicación, y quien entiende el código de la aplicación no sabe del negocio[40].

Para la arquitectura propuesta se ha escogido el sistema gestor de reglas de código abierto conocido como Drools[41]. Drools es una implementación de motor de reglas desarrollado en lenguaje Java y basado en el algoritmo de Charles Forgy llamado RETE [38].

3.4 Service Component Architecture

Service Component Architecture (SCA) [18] es un grupo de especificaciones para la construcción de aplicaciones distribuidas basadas en *Service Oriented Architecture* (SOA). Estas especificaciones son mantenidas por la organización OASIS (*Organization for the Advancement of Structured Information Standards*) [19]. SCA define servicios como funciones de negocio que pueden ser ensamblados para crear nuevos servicios. Estos servicios compuestos pueden contener nuevos servicios y/o funciones de negocios que se encuentran contenidos en aplicaciones o sistemas existentes. Los servicios pueden ser desarrollados utilizando diferentes lenguajes de programación como Java, C++ o lenguajes especiales como BPEL (*Business Process Execution Language*). Además, los servicios pueden ser desple-

gados en procesos del mismo sistema operativo o en diferentes procesos que se encuentren distribuidos en diferentes máquinas. SCA eleva el nivel de abstracción y se enfoca en solucionar dos problemas del desarrollo de software que son: complejidad y reúso de componentes [20]. También, SCA oculta del código de la aplicación, la definición de seguridad, confiabilidad y otros aspectos de calidad del servicio.

El mecanismo declarativo que ofrece SCA, permite crear servicios independientes al protocolo de comunicación, esto significa que no es necesario definir dentro del código del componente, el protocolo que se utilizara para invocar el servicio. Este mecanismo oculta complejidades presentes en otras tecnologías como Java EE y .NET 2.0. Por ejemplo la creación de una típica aplicación de negocios con capacidad de comunicarse a través de Web Services (WS) en Java EE, requiere conocimientos de JAX-WS y EJB APIs de igual manera escribir una aplicación idéntica en .NET 2.0 requiere entender ASP .NET Web Services, Enterprise Services y .NET messaging APIs. Para la arquitectura propuesta se ha elegido la implementación de SCA de software abierto llamado Apache Tuscany [42].

3.5 Bus de servicio de empresa

SCA no soporta la transformación de mensajes y solo incluye un limitado conjunto de protocolos de comunicación. Además es común encontrar una gran variedad de formatos de datos dentro de las organizaciones de asistencia médica. Esto puede convertirse en un problema cuando se requiere, que varias aplicaciones intercambien información. Con el fin de tener la flexibilidad de comunicarse con sistemas heredados o sistemas que utilizan diferentes protocolos de comunicación, se propone el uso de un Bus de Servicios de Negocio (ESB). El ESB es un software que provee interoperabilidad entre diferentes protocolos de comunicación. La tecnología de ESB es considerada la columna vertebral de SOA y provee capacidades como red, ruteo, transformación de mensajes, calidad de los servicios, seguridad y entrega de mensajes [43]. El objetivo principal de ESB es ayudar a la integración de servicios, pero no participa en la lógica del negocio [16]. La lógica del negocio debe de estar localizada dentro de los servicios, por lo tanto el ESB solo actúa como un mediador en la arquitectura SOA. Para agregar esta capacidad a la arquitectura se ha elegido el ESB de código abierto Apache Synapse [44], el cual cuenta con una gran comunidad de desarrolladores y es utilizado ampliamente por varias empresas como ebay [45] y Prudential [46].

3.6 Sistema de ayuda a la toma de decisiones.

Sistema de ayuda a la toma de decisiones clínicas (CDSS) son principalmente diseñados para apoyar al médico en la toma de decisiones clínicas durante la asistencia médica. Estas herramientas pueden proveer conocimiento pertinente y/o información filtrada acerca del paciente en los momentos necesarios para así mejorar la calidad del servicio [47]. A pesar de que se ha demostrado que CDSS puede mejorar la calidad del servicio considerablemente [48], su implementación no es muy extendida [49][50], muchos sistemas clínicos comerciales incluyen capacidades de ayuda a la toma de decisiones clínicas [51], pero estas capacidades no pueden ser compartidas con otras aplicaciones o sistemas. Por esta razón existe una necesidad por CDSS basado en SOA que pueda proveer servicios de decisión a una gran cantidad de sistemas dentro y fuera de la organización [52]. Para nuestra arquitectura hemos utilizado el CDSS de código abierto llamado OpenCDS. Este proyecto es uno de los mayores esfuerzos en el desarrollo de sistemas de soporte a decisiones basado en estándares y enfocado a SOA. OpenCDS es respaldado por el departamento de *Biomedical Informatics* de la Universidad de Utah y apoyado por una gran cantidad de instituciones alrededor del mundo [53]. La interface de comunicación de OpenCDS está basado en el HL7/OMG estándar *Decision Support Service* (DSS) [54] y el formato de datos se basa en el estándar *Virtual Medical Record* (vMR) [55].

4. CONCLUSIONES

La industria de la salud requiere de una infraestructura que permita cambios ágiles, facilite reutilización de componentes de software y ofrezca flexibilidad para integrar nuevas tecnologías. En este aspecto la aparición de tecnologías como XML, Web Services y SOA ofrecen una buena alternativa, desafortunadamente las soluciones comerciales basadas en estas tecnologías son muy costosas. Existen varias aplicaciones de software abierto que en conjunto pueden proveer una arquitectura suficientemente robusta para la implementación del historial clínico electrónico. Este artículo describe una arquitectura orientada a servicios (SOA) construida por completo a partir de componentes de código abierto que incluye sistema de gestión de procesos (BPMS) y sistema de ayuda a toma de decisiones clínicas (CDSS). La arquitectura intenta dar solución a los problemas de interoperabilidad que impiden la implementación del expediente clínico electrónico (EHR).

5. REFERENCIAS BIBLIOGRÁFICAS

[1] Committee on Quality of Health Care in America and Institute of Medicine, *To err is*

human: building a safer health system. Washington, D.C: National Academy Press, 2000.

- [2] Organization for Economic Cooperation and Development, "Health at the Glance 2011: OECD Indicators," 2011. [Online]. Available: <http://www.oecd.org/dataoecd/6/28/49105858.pdf>.
- [3] Holtz, *Global Health Care: Issues and Policies*. Jones and Bartlett Publishers, 2006.
- [4] S. P, M. SC, and K. EB, "Costs and Benefits of Health Information Technology," *Rockville (MD): Agency for Healthcare Research and Quality (US)*, 2006. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK37988/>.
- [5] D. Blumenthal, C. DesRoches, and K. Donelan, "Health information technology in the United States: the information base for progress," *Robert Wood Johnson Foundation*, 2006. [Online]. Available: <http://www.rwjf.org/files/publications/other/EHRRReport0609.pdf>. [Accessed: 28-Jun-2012].
- [6] A. K. Jha, C. M. DesRoches, E. G. Campbell, K. Donelan, S. R. Rao, T. G. Ferris, A. Shields, S. Rosenbaum, and D. Blumenthal, "Use of Electronic Health Records in U.S. Hospitals," *New England Journal of Medicine*, vol. 360, no. 16, pp. 1628–1638, 2009.
- [7] C. M. DesRoches, E. G. Campbell, S. R. Rao, K. Donelan, T. G. Ferris, A. Jha, R. Kaushal, D. E. Levy, S. Rosenbaum, A. E. Shields, and D. Blumenthal, "Electronic Health Records in Ambulatory Care — A National Survey of Physicians," *New England Journal of Medicine*, vol. 359, no. 1, pp. 50–60, 2008.
- [8] P. G. Goldschmidt, "HIT and MIS: implications of health information technology and medical information systems," *Commun. ACM*, vol. 48, no. 10, pp. 68–74, Oct. 2005.
- [9] G. L. Glandom, D. H. Smaltz, and D. J. Slovensky, *Information Systems for Healthcare Management*. Chicago, IL: Health Administration Press, 2008.
- [10] B. V. Kumar, *Web Services*. New Delhi: Tata McGraw-Hill Education, 2004, p. 20.
- [11] J. Blanton, S. Leski, B. Nicks, and T. Tirzaman, "Making SOA work in a healthcare company," in *Proceedings of the 24th ACM*

- SIGPLAN conference companion on Object oriented programming systems languages and applications*, 2009, pp. 589–596.
- [12] D. Garets and M. Davis, "Electronic Medical Records vs Electronic Health Records: Yes , There Is a Difference," 2006. [Online]. Available: http://www.himssanalytics.org/docs/wp_emr_ehr.pdf. [Accessed: 09-Aug-2012].
- [13] A. Boonstra and M. Broekhuis, "Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions.," *BMC health services research*, vol. 10, p. 231, Jan. 2010.
- [14] N. I. Binti Ismail and N. H. Binti Abdullah, "Developing electronic medical records (EMR) framework for Malaysia's public hospitals," *Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on*. pp. 131–136, 2011.
- [15] Health Level Seven International (HL7), "Health Level Seven International." [Online]. Available: <http://www.hl7.org/>. [Accessed: 28-Jun-2012].
- [16] T. Erl, *Service-Oriented Architecture Concepts, Technology, and Design*. Upper Saddle River, NJ: Prentice Hall, 2005.
- [17] R. Welke, R. Hirschheim, and A. Schwarz, "Service-Oriented Architecture Maturity," *Computer*, vol. 44, no. 2, pp. 61–67, 2011.
- [18] S.-Y. Hung, W.-H. Hung, C.-A. Tsai, and S.-C. Jiang, "Critical factors of hospital adoption on CRM system: Organizational and information system perspectives," *Decision Support Systems*, vol. 48, no. 4, pp. 592–603, 2010.
- [19] Gartner Group, "Customer Relationship Management (CRM)." [Online]. Available: <http://www.gartner.com/it-glossary/customer-relationship-management-crm/>. [Accessed: 24-Aug-2012].
- [20] G. Benz and N. V. Paddison, "Developing patient-based marketing strategies.," *Healthcare executive*, vol. 19, no. 5, pp. 40, 42, 2004.
- [21] M. C. Lin, "A Study of Mainstream Features of CRM System And Evaluation Criteria," in *American Society for Engineering Education Annual Conference & Exposition*, 2003.
- [22] A. Winter, R. Haux, E. Ammenwerth, B. Brigl, N. Hellrung, and F. Jahn, *Health Information Systems: Architectures and Strategies*. London: , 2010.
- [23] J. J. P. C. Rodrigues and B. Vaidya, "Health Information Systems," in *Health Information Systems: Concepts, Methodologies, Tools, and Applications*, New York: , 2010.
- [24] S. Dinehart, "Choosing a computer billing system: avoid these mistakes.," *Seminars in cutaneous medicine and surgery*, vol. 19, no. 3, pp. 177–80, Sep. 2000.
- [25] A. Dedeker, "Improving Legacy-System Sustainability: A Systematic Approach," *IT Professional*, vol. 14, no. 1. pp. 38–43, 2012.
- [26] M. Hegde and J. K. Wall, "Effective Application Modernization with Business Rule Management System." IBM, 2009.
- [27] S. Brahe, "BPM on Top of SOA: Experiences from the Financial Industry," *Springer*, vol. 4714/2007, 2007.
- [28] H. Smith and P. Fingar, *Business Process Management (BPM): The Third Wave*. Tampa, FL: Meghan-Kiffer Press, 2003.
- [29] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, *Applied Soa: Service-Oriented Architecture and Design Strategies*. Indianapolis, Indiana: Wiley Publishing, 2008.
- [30] JBoss Community, "jBPM." [Online]. Available: <http://www.jboss.org/jbpm>. [Accessed: 28-Jun-2012].
- [31] Business Rules Group, "Business Rules Group." [Online]. Available: <http://www.businessrulesgroup.org/>. [Accessed: 28-Jun-2012].
- [32] Business Rules Group, "Defining Business Rules ~ What Are They Really?" [Online]. Available: http://www.businessrulesgroup.org/first_paper/br01c0.htm. [Accessed: 18-Aug-2012].
- [33] F. J. de la Torre and M. G. Perez, "Parallelizing the Design and Development of a Monitoring System," in *Cooperative Design, Visualization, and Engineering 6th International Conference, CDVE 2009 Proceedings*, Heidelberg: Springer, 2009.
- [34] A. April and A. Abran, *Software Maintenance Management: Evaluation and Continuous*

- Improvement*. New Jersey: IEEE Computer Society, 2008.
- [35] P. Grubb and A. A. Takang, "Software Maintenance: Concepts and Practice," London: World Scientific Publishing, 2003.
- [36] G. Witt, *Writing Effective Business Rules*. Waltham, MA: Elsevier, 2012.
- [37] F. Rosenberg and S. Dustdar, "Towards a distributed service-oriented business rules system," *Web Services, 2005. ECOWS 2005. Third IEEE European Conference on*, p. 11 pp., 2005.
- [38] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," *Artificial Intelligence*, vol. 19, no. 1, pp. 17–37, 1982.
- [39] J. Feldman, "Representing and Solving Rule-Based Decision Models with Constraint Solvers 5th International Symposium, RuleMI 2011 - America Ft. Lauderdale, FL, USA, November 2011 Proceedings," in *Rule - Based Modeling and Computing on the Semantic Web*, Berkeley, CA: Springer, 2011.
- [40] J. Taylor and N. Raden, *Smart (Enough) Systems: How to Deliver Competitive Advantage by Automating hidden decisions*. New York: Prentice Hall, 2007.
- [41] JBoss Community, "Drools - The Business Logic integration Platform." [Online]. Available: <http://www.jboss.org/drools/>. [Accessed: 28-Jun-2012].
- [42] The Apache Software Foundation, "Apache Tuscany." [Online]. Available: <http://tuscany.apache.org/>. [Accessed: 28-Jun-2012].
- [43] J. Bean, *SOA and Web Services Interface Design Principles, techniques, and Standards*. Burlington: Elsevier, 2010.
- [44] The Apache Software Foundation, "Apache Synapse Enterprise Service Bus (ESB)." [Online]. Available: <http://synapse.apache.org/>. [Accessed: 28-Jun-2012].
- [45] ebay, "ebay." [Online]. Available: <http://www.ebay.com/>. [Accessed: 28-Aug-2012].
- [46] Prudential, "Prudential." [Online]. Available: <http://www.prudential.com/view/page/public>. [Accessed: 20-Aug-2012].
- [47] J. A. Osherooff, J. M. Teich, B. Middleton, E. B. Steen, A. Wright, and D. E. Detmer, "A roadmap for national action on clinical decision support.," *Journal of the American Medical Informatics Association: JAMIA*, vol. 14, no. 2, pp. 141–5, Jan. 2007.
- [48] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success," *BMJ*, 2005.
- [49] J. A. Lyman, W. F. Cohn, M. Bloomrosen, and D. E. Detmer, "Clinical decision support: progress and opportunities," *Journal of the American Medical Informatics Association: JAMIA*, no. 17, pp. 487–492, 2010.
- [50] R. A. Jenders, J. A. Osherooff, D. F. Sittig, E. A. Pifer, and J. M. Teich, "Recommendations for Clinical Decision Support Deployment: Synthesis of a Roundtable of Medical Directors of Information Systems," *AMIA Annu Symp Proc*, pp. 359–363, 2007.
- [51] G. J. Kuperman, R. M. Reichley, and T. C. Bailey, "Using Commercial Knowledge Bases for Clinical Decision Support," *Journal of the American Medical Informatics Association: JAMIA*, no. 13, pp. 369–371, 2006.
- [52] K. Kawamoto, D. F. Lobach, H. F. Willard, and G. S. Ginsburg, "A national clinical decision support infrastructure to enable the widespread and consistent practice of genomic and personalized medicine," *BMC Medical Informatics and Decision Making*, vol. 9, no. 19, 2009.
- [53] K. Kawamoto, "OpenCDS." [Online]. Available: <http://www.opencds.org/>. [Accessed: 28-Feb-2012].
- [54] Health Level Seven International (HL7), "HL7 Decision Support Service (DSS)." [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12. [Accessed: 26-Jun-2012].
- [55] Health Level Seven International (HL7), "Virtual Medical Record." [Online]. Available: [http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_\(vMR\)](http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_(vMR)). [Accessed: 26-Jun-2012].

6. SÍNTESIS CURRICULARES DE LOS AUTORES

Salvador Rodríguez Loya: Nació en Hidalgo del Parral Chihuahua, México en 1978. El recibió grado como ingeniero en electrónica en el año 2002 y ha trabajado en la industria por 7 años en las áreas de electrónica e informática. Actualmente es estudiante de doctorado de la Universidad de Sussex en el Reino Unido. Sus intereses de investigación son Cloud Computing, Service Oriented Architecture y Clinical Decision Support Systems. Dirección: Shawcross Building, Brighton, BN1 9QT, United Kingdom.

Email: s.rodriguez-loya@sussex.ac.uk,

María Yolanda Rodríguez Loya: Es mexicana y profesora en las carreras de Licenciatura en Informática e Ingeniería en sistemas del Instituto Tecnológico de Matamoros, México. Cuenta con una maestría en sistemas de información de la universidad autónoma de Chihuahua, México y con 14 años de experiencia como catedrático universitario. Dirección: Calle Capricornio 26, Jardines del Satélite, Matamoros Tamaulipas 87456, México.

Email:yol.nada@hotmail.com