

OpenSITEM. Caracterización de nodos en
redes de telesalud

OpenSITEM. Caracterización de nodos en redes de telesalud

Lilia Edith Aparicio Pico
Paulo César Coronado Sánchez





UD
Editorial

E2
ESPACIOS

© Universidad Distrital Francisco José de Caldas
© Centro de Investigaciones y Desarrollo Científico
© Lilia Edith Aparicio Pico, Paulo César Coronado Sánchez
Primera edición, abril de 2019
ISBN: 978-958-787-113-5

Dirección Sección de Publicaciones
Rubén Eliécer Carvajalino C.

Coordinación editorial
Edwin Pardo Salazar

Corrección de estilo
Rodrigo Díaz Lozada

Diagramación y montaje de cubierta
Fabián Camargo Muñoz

Editorial UD
Universidad Distrital Francisco José de Caldas
Carrera 24 No. 34-37
Teléfono: 3239300 ext. 6202
Correo electrónico: publicaciones@udistrital.edu.co

Aparicio Pico, Lilia Edith.

OpenSITEM caracterización de nodos en redes de telesalud /
Lilia Edith Aparicio Pico, Paulo César Coronado. -- Bogotá:
Universidad Distrital Francisco José de Caldas, 2019.

186 páginas ; 24 cm. -- (Colección espacios)

1. Tecnología médica 2. Innovaciones en medicina
3. Telemedicina. I. Coronado, Paulo César, autor. II. Tít. III. Serie.
371.334 cd 22 ed.
A1629154

CEP-Banco de la República-Biblioteca Luis Ángel Arango

Todos los derechos reservados.

Esta obra no puede ser reproducida sin el permiso previo escrito de la
Sección de Publicaciones de la Universidad Distrital.

Hecho en Colombia

Contenido

Presentación	15
Estructura del proyecto	17
Objetivos	17
Objetivo general	17
Objetivos específicos	17
Base metodológica	17
Metodología de recolección de información	18
Modelo de requisitos	18
Establecimiento de nivel de calidad	18
Soporte a decisiones arquitectónicamente significativas	18
Instrumentos empleados para recolección de información	18
Metodología para la estructuración y validación del documento	18
Declaración de situación esperada	19
Requisitos arquitectónicamente significativos	19
Repositorios de código fuente	20
Antecedentes	20
Sistema para la caracterización de nodos potenciales de redes de eSalud	21
SITEM - Fase I	22
SITEM - Fase II	25
OpenSITEM - Fase III	25
Contexto teórico	27

Redes de eSalud	27
Caracterización de nodos para redes de eSalud	28
Subsistemas en una red de eSalud	29
eSalud	30
Componentes principales de eSalud	31
Marco legislativo y normativo	31
En el ámbito de los servicios médicos	31
<i>Ley 1751 de 2015</i>	32
<i>Ley 1419 de 2010</i>	32
<i>Resolución 2182 de julio 9 de 2004</i>	32
<i>Resolución 4678 de noviembre 11 de 2015</i>	33
<i>Resolución 1830 de junio 23 de 1999</i>	33
<i>Resolución 1895 de noviembre 19 de 2001</i>	33
<i>Resolución 1995 de julio 8 de 1999</i>	33
<i>Circular 015 de abril 4 de 2002</i>	33
<i>Decreto 2092 de 2 de julio de 1986</i>	34
En el ámbito del desarrollo de redes teleinformáticas	34
<i>Documento Conpes 3072</i>	34
<i>Documento Conpes 3582</i>	34
<i>Resolución 087 de 1997</i>	34
Manual de Calidad de Servicio	35
En el ámbito del tratamiento de datos e información	35
<i>Leyes 1581 de 2014 y 1266 de 2008</i>	35
Aspectos clave en la gestión y dirección del conocimiento	35
Acerca del conocimiento	35
Ciclo de conocimiento	36
Tecnologías del conocimiento	38
Gestión de conocimiento	39
Sistemas de gestión de conocimiento	40
Sistemas integrados para el soporte de desempeño	41
Ingeniería de software	41
Ingeniería	42

Proceso de desarrollo del software	43
Principios de diseño y desarrollo	43
Proceso de desarrollo de software de código abierto	45
Métodos ligeros	48
Proceso unificado	50
UML: lenguaje de modelado unificado	53
Portales de información y conocimiento	54
Beneficios y obstáculos para la implementación de portales basados en aplicaciones web	55
Ciclo de vida de los portales	58
Presencia emergente	58
Presencia mejorada	58
Interacción	58
Transacción	58
Transformación	59
Aplicaciones web	59
OpenSITEM - Sistema federado de aplicaciones para la caracterización de nodos potenciales de redes de eSalud	60
Módulos funcionales	61
Características de interés de OpenSITEM	62
Descripción de la arquitectura de OpenSITEM	63
Interesados e intereses	63
Punto de vista estructural	64
Punto de vista de comportamiento	67
<i>OpenSITEM-FE: motor de federación de aplicaciones</i>	67
Punto de vista de distribución	67
Gestión de nodos	68
Motor de recomendación	68
Modelo de nodos	69
Prestador de servicio de salud	69
Tecnología de interconexión	70
Equipo	70
Operador de telecomunicaciones	70

Organización	71
Servicios médicos	71
Aplicaciones federadas	72
Marco de desarrollo OpenSARA	73
Patrón frontera-control-entidad	74
Marco de trabajo	75
<i>Bloque</i>	75
<i>Página</i>	75
<i>Clase nuclear</i>	75
<i>Componentes</i>	75
Estructura	75
<i>Bloque</i>	75
Experiencia de desarrollo de OpenSITEM	77
Modelo de requisitos	77
Declaración del problema	78
Responsabilidades del sistema	78
Alcance	78
Definición de actores	78
Casos de uso	79
Arquitectura	80
Modelo de implementación	81
Acerca de la seguridad en el sistema	83
Modelo de datos	85
Interfaz gráfica	85
Arquitectura de la página	86
Entregables del proyecto	87
Conclusiones y recomendaciones	89
Bibliografía	91
Glosario	99
Apéndice A	105
Modelo de requisitos	105
Modelo de casos de uso	105
<i>Características generales de los actores</i>	105
Casos de uso	106

Apéndice B	115
Modelo de análisis y diseño	115
<i>Especificaciones de casos de uso</i>	115
Apéndice C	133
Modelo de implementación	133
Código fuente de OpenSITEM	134
Clase página	135
Apéndice D	137
Modelo de datos	137
Apéndice E	139
Modelo de despliegue	139
Apéndice F	141
Declaración de riesgos	141
Apéndice G	147
Instrumento preliminar para la fase de transición	147
<i>Presentación</i>	147
<i>Indicaciones generales</i>	147
Contexto clínico	148
Unidades de recolección de información	149
Contexto de servicio/relación con el entorno	150
Unidades de recolección de información	150
Consideraciones tecnológicas	152
<i>Unidades de recolección de información</i>	152
<i>Software</i>	153
Consideraciones de calidad	153
Unidades de recolección de información	154
Aceptación del servicio de salud brindado en la modalidad de telemedicina	155
<i>Unidades de recolección de información</i>	155
Aspectos de construcción y persistencia del sistema de telemedicina	156
<i>Unidades de recolección de información</i>	156
Políticas de la organización	157
Consideraciones acerca de la información	157

<i>Unidades de recolección de información</i>	158
Procesamiento de información	158
<i>Unidades de recolección de información</i>	158
Apéndice H	161
Otros entregables nucleares de OpenSITEM	161
Visión-resumen ejecutivo	161
<i>Propósito inicial</i>	161
<i>Alcance inicial</i>	161
Posicionamiento	162
Participantes en el proyecto y usuarios	163
Entorno de usuario	165
Suposiciones y dependencias	165
Descripción global de OpenSITEM	165
Otros requisitos del producto	165
Lineamientos de codificación para la organización de los módulos	166
Apéndice I	167
Análisis de necesidad de un bus de servicio empresarial	167
Capacidad de integración	167
Protocolo de comunicación	167
Capacidades especiales de mensajería	167
Orientación a servicios	167
Escalabilidad vs. Simplificación	168
Apéndice J	169
Aspectos relativos al despliegue	169
Fuentes de información primaria	170
Apéndice K	171
Entrevista base para identificación de interesados	171
Metodología de procesamiento	171

Instrumento	171
Anexo A	173
GNU General Public License	173
Preámbulo	173
<i>Términos y condiciones para copia, modificación y distribución</i>	174
<i>Inexistencia de garantía</i>	178

Presentación

En el marco del proyecto Telemedicina Bogotá 2K, el Grupo de Investigación en Telemedicina de la Universidad Distrital Francisco José de Caldas (GITEM) realizó un estudio de campo para caracterizar las entidades de la Red Distrital de Salud. Este estudio permitió hacer un diagnóstico del estado de la eSalud en la ciudad y refrendar que las acciones de investigación del grupo estaban alineadas con la propuesta que la Organización Mundial de la Salud (OMS) y la Unión Internacional de Telecomunicaciones presentaron en las Herramientas para el desarrollo de estrategias nacionales de eSalud.

Como consecuencia de los hallazgos presentados en los informes del Ministerio de Salud y Seguridad Social (Minsalud), de la Organización para la Cooperación y el Desarrollo Económico (OCDE) y el Observatorio Así Vamos en Salud, el grupo GITEM desarrolló la fase III del Sistema de Información para la Caracterización de Nodos Potenciales de Redes de eSalud con el que se propone un portal de inteligencia analítica que apoye los procesos de definición de capacidad en proyectos de eSalud.

OpenSITEM, como se ha denominado el sistema, es fruto del esfuerzo que realiza el grupo para sistematizar su experiencia de diagnóstico. Busca ofrecer una plataforma alternativa, independiente y emergente que colabore en la superación de las limitaciones que en materia de gestión y análisis de información tienen los equipos de trabajo encargados del diseño y desarrollo de redes de eSalud.

En el dominio técnico, OpenSITEM es un sistema federado de aplicaciones de software libre o de código abierto que provee herramientas para analizar datos e información de los siguientes elementos que son de interés para la descripción y definición de capacidad de nodos potenciales de redes de eSalud: entidades de salud, servicios médicos, tecnologías de interconexión, operadores de telecomunicaciones, equipos médicos, organizaciones, profesionales, estándares, pacientes, enfermedades, medicamentos y proyectos. OpenSITEM *no es un sistema de eSalud*, sino una plataforma para apoyar la definición de la capacidad que tiene un nodo para potencialmente hacer parte de una red de eSalud.

OpenSITEM propone un mecanismo para la integración de actores en el área de *caracterización y análisis de capacidad* de nodos en redes de eSalud. Provee un escenario ubicuo, basado en tecnologías de la información y un modelo de trabajo colaborativo en red que propende por la construcción evolutiva de una base de información y

conocimiento. Siguiendo los lineamientos de apertura en la información y el conocimiento (*open philosophy*), busca elaborar un *recurso público* para ayudar al diseño de proyectos de eSalud en nuestra ciudad.

La arquitectura general del sistema y la experiencia en el desarrollo se presenta en este documento, organizado en las siguientes secciones:

- **Antecedentes:** enumera los aspectos que motivaron el proyecto y un resumen de las fases anteriores.
- **Estructura del proyecto:** muestra un resumen de los aspectos que, a consideración de los autores, son relevantes para medir el impacto presente y la potencialidad del proyecto.
- **Contexto teórico:** recopilación de aspectos teóricos que fueron de interés para abordar el desarrollo del trabajo de investigación.
- **OpenSITEM:** descripción de la arquitectura del sistema resultante haciendo énfasis en sus conceptos y propiedades fundamentales. El producto software que de manera emergente se está creando se centra en esta descripción y guiado por los casos de uso que se presentan en este informe. Esta sección se complementa con los apéndices de los modelos de requisitos, análisis y diseño, implementación, despliegue y datos.
- **Experiencia de desarrollo:** teniendo en cuenta que la plataforma de software que soporta OpenSITEM es de código abierto, se presenta una definición del método de trabajo empleado para que los grupos interesados en participar puedan integrarse a la comunidad de desarrollo. Esta sección hace especial énfasis en los aportes originales del grupo y describe las demás herramientas que se integran al modelo.
- **Conclusiones y recomendaciones:** aspectos y hallazgos que se deben considerar para la consolidación de OpenSITEM así como el desarrollo de nuevos proyectos de investigación.

Lilia Edith Aparicio Pico. Ph. D.

Directora GITEM++

Grupo de investigación en Telemedicina

Universidad Distrital Francisco José de Caldas

Estructura del proyecto

Este capítulo presenta un resumen del proyecto *OpenSITEM: sistema federado de aplicaciones para la caracterización de nodos potenciales de redes de eSalud*.

Objetivos

El proyecto planteó los siguientes objetivos:

Objetivo general

Describir una arquitectura de un sistema software cuya funcionalidad permita caracterizar nodos potenciales de redes de eSalud, considerando aspectos de adaptabilidad e interoperabilidad.

Objetivos específicos

- Identificar, a partir del análisis de los resultados del estudio de campo existente, los tipos de nodos nucleares para la definición de redes de eSalud.
- Producir o adaptar un marco de desarrollo de aplicaciones de código abierto que permita a los integrantes del grupo de investigación desarrollar módulos de software alineados con la descripción de la arquitectura.
- Construir prototipos de software que implementen la funcionalidad de los módulos nucleares del sistema.
- Establecer el nivel base de calidad de la descripción de la arquitectura y de los prototipos desarrollados con base en pruebas automatizadas y valoraciones de parte de los interesados.

Base metodológica

El proyecto se desarrolla con un enfoque mixto cualitativo-cuantitativo, con alcance exploratorio, descriptivo y proyectivo.

Los métodos específicos empleados se decantan de una metodología Architecture Definition Method (ADM) y un método de desarrollo ágil basado en OpenUP, el cual es orientado a la gestión del riesgo, guiado por casos de uso, centrado en la arquitectura, iterativo e incremental.

Metodología de recolección de información

El proyecto recolecta información para la definición del modelo de requisitos, el establecimiento del nivel de calidad y el soporte de las decisiones significativas para la arquitectura. A continuación se presentan los métodos empleados en cada uno de estos dominios.

Modelo de requisitos

Se emplearon varios métodos de los descritos en el OpenUP, en especial las entrevistas a interesados, los talleres de requisitos y el estudio de sistemas similares. Todos y cada uno de ellos se realizaron de acuerdo con los lineamientos de OpenUP, siendo los talleres complementados con las técnicas descritas por Gottesdiener.

Establecimiento de nivel de calidad

Para el establecimiento de nivel de calidad se empleó el análisis de calidad de software empleando la herramienta SonarQube¹, el análisis de logs del sistema y la presentación de prototipos a los interesados, técnica descrita en OpenUP.

Soporte a decisiones arquitectónicamente significativas

Para recolectar información significativa para este dominio se empleó la técnica de reunión de tormenta de ideas y el *mapeo* de historias de usuario.

Instrumentos empleados para recolección de información

Se diseñaron varios instrumentos para recolectar información y se aplicaron en diferentes sesiones de las reuniones de tormentas de ideas, talleres de requisitos, presentación de prototipos y entrevistas a interesados (véase apéndices G y K).

Metodología para la estructuración y validación del documento

La creación del formato de entrevista siguió un método de refinamiento basado en retroalimentación directa, a partir de los pasos definidos por Dave Collingridge. Cada una de las versiones fue desarrollada por el equipo y presentada en los talleres de requisitos para aprobación por parte de los interesados.

El proceso consistió en seis pasos que se aplicaron de manera iterativa hasta obtener una versión estable del documento:

- Validar por revisión de pares: empleando expertos en el dominio —integrantes del grupo de investigación y expertos en la elaboración de instrumentos—. Profesionales adscritos al Comité de Autoevaluación Institucional colaboraron en esta tarea. Se revisó, principalmente, que las preguntas tuviesen claridad en el enunciado, no tuvieran sesgo, no generaran referencias circulares, no fueran de doble cañón, entre otros. En este paso se categorizó cada pregunta de conformidad con el contenido, según la clasificación hecha por García Córdoba.

1 Véase <https://www.sonarqube.org/>

- Automatizar el instrumento: para esto se elaboró un sistema de gestión de encuestas utilizando el marco de desarrollo OpenSARA. Dicho sistema permite crear preguntas de diferentes tipos, agruparlas en instrumentos, definir periodos de aplicación, generar reportes, entre otras funcionalidades².
- Realizar una prueba piloto: se seleccionó un conjunto que no superara el 20 % de la población de interesados y se les aplicó el instrumento. Se pidió que evaluaran la claridad, extensión y dificultad del instrumento.
- Analizar los datos recolectados: teniendo en consideración que los instrumentos están automatizados, el análisis parte de que se tienen datos íntegros. Se revisa por tanto la relación entre la respuesta y el sentido de la pregunta, la validez de los dominios de datos (en especial en preguntas cuantitativas), la pertinencia del tipo de pregunta (selección múltiple, abierta, selección única, etc.) y el tiempo que se tarda en diligenciar el instrumento.
- Análisis de componentes principales: a partir de las respuestas se crea una matriz de correlación de conceptos y se descartan aquellas preguntas que tengan un alto grado de correlación.

Declaración de situación esperada

Con el desarrollo del proyecto se obtiene una descripción de arquitectura, los lineamientos para la construcción progresiva de un sistema software que implemente dicha arquitectura y un conjunto base de prototipos. A partir de la entrega de estos resultados, se espera que los estudios de campo y análisis realizados por el grupo de investigación puedan ser consultados libremente a través de internet.

Requisitos arquitectónicamente significativos

En la descripción de la arquitectura se definen los siguientes requisitos arquitectónicos para la adecuada gestión de riesgo:

- Integrar aplicaciones robustas que en la actualidad estén licenciadas como código abierto.
- Se prioriza la adaptación frente al desarrollo desde cero.
- La interoperabilidad utiliza arquitectura orientada a servicios con modelo de Bus de Servicio Empresarial.
- A nivel de aplicativo, se promueve una arquitectura multicapa-multinivel, en donde la persistencia se tiene en nodos separados.
- El crecimiento de la aplicación se realiza a través de plugins o utilizando las API definidas.

2 Si bien este módulo no hace parte de los objetivos del proyecto, sí sirvió para validar la potencialidad del marco de desarrollo OpenSARA. El sistema de gestión de encuestas está siendo probado a nivel institucional para soportar los procesos de autoevaluación (<http://autoevaluacion.udistrital.edu.co/version3/>).

- Se propende por un sistema federado de aplicaciones.
- Funcionalidad emergente, es decir, prototipos base cuyo acercamiento a la arquitectura se realiza progresivamente. Esto con el fin de gestionar, ser conscientes del riesgo relacionado con el bajo tiempo de permanencia de personal especializado³.

Repositorios de código fuente

Los módulos de los prototipos así como el del marco de desarrollo se han liberado bajo una licencia considerada de código abierto. El código fuente se encuentra en repositorios públicos de la plataforma github:

- Marco de desarrollo de aplicaciones OpenSARA: [frameworksara/openSARA](https://github.com/frameworksara/openSARA).
- Módulo de nodos medicamentos: [giseproi/OpenHealthVademecum](https://github.com/giseproi/OpenHealthVademecum).
- Módulos de prototipo OpenSITEM: [giseproi/openSITEM](https://github.com/giseproi/openSITEM).

Los repositorios son de acceso público y están administrados por el equipo de desarrollo. Los integrantes de este equipo se encargan de analizar las contribuciones para agregarlas a las ramas que vayan surgiendo. Con la entrega de este informe y la transferencia desde el servidor git privado a github, se está promoviendo la participación de nuevos actores que contribuyan a mejorar y hacer crecer el sistema.

Antecedentes

A principios del milenio, el grupo GITEM realizó un estudio para abordar problemas puntuales en la implementación de servicios médicos prestados a través de medios teleinformáticos en el Distrito Capital (Aparicio, 2000, p. 10):

En el momento no existe un diagnóstico real sobre los servicios requeridos en el área de telemedicina, razón suficiente para iniciar un trabajo de campo que establezca la situación actual de servicios médicos y la demanda real, así como la posibilidad de conocer a corto, mediano y largo plazo cuáles serían los costos de inversión que permitirían dar soluciones al problema de cobertura.

La socialización del conocimiento alrededor de las tecnologías aplicadas al desarrollo de la medicina, es uno de los valores que lleva al éxito de soluciones efectivas en el sector salud, por tal motivo es necesario desarrollar un plan de alfabetización en el sector salud y en el sector gubernamental y académico.

En el país no existen estrategias de investigación en esta área del conocimiento para llevar a cabo un estudio real que permita dar el paso a soluciones verdaderas sobre desarrollo tecnológico o experimental para poder implementar centros de investigación en Telemedicina.

3 Debido a la escasa financiación económica, no es posible mantener el personal que se ha especializado en el proyecto. El grupo y sus profesores asociados mantienen un ciclo corto de reclutamiento, capacitación, producción y transferencia.

La Universidad Distrital tiene el recurso humano, el conocimiento y la experiencia científica y tecnológica, capaz de dar soluciones tangibles a estas necesidades; unida al conocimiento y experiencia de entidades como clínicas y hospitales y con la participación de operadores de comunicaciones, puede desarrollar soluciones efectivas a los problemas de salud que afronta la sociedad colombiana.

Los resultados obtenidos fueron recopilados en extensos tomos en formato físico y digital. Luego de los primeros análisis, se observó que los reportes no tenían una estructura documental coherente ni un modelo de información que los caracterizara. Este hecho, a la par con el ingreso de entidades al estudio, aumentó la complejidad a la hora de generar estudios comparativos o de apoyo a la toma de decisiones. Se evidenciaron casos de información faltante o innecesaria para el proyecto. Como consecuencia, al ser el objeto de estudio intrínsecamente dinámico, cualquier cambio de las condiciones iniciales no se ve reflejado y la información queda desactualizada rápidamente. En ese escenario, la labor de articular información de las organizaciones suponía un proceso lento, por lo cual la estrategia de gestión de información empleada en el estudio comenzó a mostrar debilidades.

Como respuesta a este contexto problemático se creó dentro del grupo un proyecto denominado *Sistema para la caracterización de nodos potenciales de redes de eSalud*, que en sus primeras fases de desarrollo ha dado solución parcial al marcar las pautas hacia la integración de información para el GITEM. En paralelo, el grupo de investigación implementó, en convenio con Colciencias, el Sistema de Referencia y Contrarreferencia para el Distrito Capital, utilizando herramientas de desarrollo propietarias, específicamente el middleware.NET de Microsoft, con lo que el grupo adquirió experiencia en la desarrollo de aplicaciones siguiendo metodologías efectivas para la construcción de software. Así, el estudio base generó una “instantánea” de la capacidad de la red de salud del Distrito para el emprendimiento de proyectos de eSalud, pero no se logró un sistema que permitiera observar la evolución de dicha capacidad o tomar otras “instantáneas” a través del tiempo.

Sistema para la caracterización de nodos potenciales de redes de eSalud

El proyecto que se describe en este documento surge de la necesidad de contar con un espacio común de información que permita valorar la capacidad de las instituciones que pertenecen a una red de salud y ofrecer un mecanismo de observación, intercambio, análisis y colaboración entre equipos de consultoría y diseño de redes de eSalud. El proyecto —cuyo nombre de desarrollo es OpenSITEM— pretende brindar una plataforma tecnológica para que grupos de trabajo especializados puedan integrar (cargar, almacenar, analizar, categorizar, comparar) los datos que les permitan responder inquietudes tales como:

- ¿Cuáles son las necesidades de servicios de la población que atiende la institución?
- ¿Cuáles son las características de la redes eléctricas, de telecomunicaciones, de servicios médicos, de profesionales y de equipos médicos de la institución?

- De un grupo seleccionado de instituciones, ¿cuál es la de mayor potencial para la prestación de un servicio de eSalud?
- ¿Cuál es la capacidad de un proveedor de telecomunicaciones para atender la demanda de un determinado servicio en una zona específica?

OpenSITEM se presenta como un sistema federado de aplicaciones de código abierto que, en conjunto, ofrecen herramientas para contestar estas u otras preguntas relacionadas. Si bien el desarrollo de cada componente no ha sido abordado por el grupo, la contribución innovadora radica tanto en la arquitectura como en el motor de integración que conducen a un modelo de utilidad que permite la interoperabilidad novedosa de aplicaciones preexistentes en un nuevo dominio de negocio, con potencial de ser utilizado en procesos de diseño y planeación de sistemas de eSalud.

El proyecto ha pasado por varias fases que han correspondido a etapas dentro de un proceso de investigación emergente. El producto ha evolucionado de un simple sitio web construido ciento por ciento *in-house*, a una solución federada —centrada en una arquitectura y guiada por casos de uso— que recoge los aspectos de calidad definidos por cada uno de sus componentes y del motor de integración. A continuación se describe brevemente cada una de las fases.

SITEM - Fase I

El objetivo de esta fase fue la definición de un modelo de información que permitiera la gestión de resultados del estudio de campo. A partir de dicho modelo se presentó la recomendación de definir un sistema informático que permitiera gestionarlo. Con los resultados de esta fase se encontró *un costo de oportunidad adecuado, debido a que en la actualidad ningún portal se especializa en el proceso de diagnóstico de capacidad para la implementación de proyectos de eSalud.*

Los alcances y logros efectivos de esta fase fueron:

- Descripción base del modelo de información.
- Propuesta de desarrollo de portal especializado.
- Bosquejo de la arquitectura general del sistema.
- Integración del SITEM dentro de los proyectos del grupo, entendiéndolo como una plataforma tecnológica para soportar grupos de trabajo que constituyan un observatorio multitemporal de los nodos que integran las redes de salud, con miras a servir de referente para la medición de capacidad para el despliegue o seguimiento de proyectos de eSalud.
- Estudios sobre filosofía de software libre y el movimiento del código abierto.
- Construcción de un prototipo de baja funcionalidad conocido como SITEM versión 0.1, bautizado internamente como Kauil. Este prototipo se construyó totalmente por los integrantes del grupo y, aunque estuvo al aire durante dos años, nunca alcanzó una versión estable ni sus características de calidad fueron evaluadas.

Para la construcción del prototipo se experimentó un proceso de desarrollo ágil, considerando los principios de Kanban (Anderson, 2016). Con algo de esfuerzo se apropiaron los valores y algunas técnicas de trabajo propuestas por el método. Sin embargo, ciertas especificidades del proyecto condujeron a un entorno de riesgo que no fue posible gestionar. Como principales causas se priorizaron las siguientes:

- Escasos recursos de personal, financieros y técnicos. Durante esta fase el grupo de investigación priorizó el estudio de campo, por lo cual SITEM tenía asignado solo dos integrantes.
- Competencias y habilidades no homogéneas entre participantes. Esto condujo a inversión de tiempo en capacitación, aun cuando un supuesto era que al tener estudiantes de últimos semestres estos tendrían formación en ingeniería de software y desarrollo de sistemas.
- Bajo tiempo de permanencia de los integrantes. Con un promedio de retención inferior a 0,5 años y un nivel de inmersión promedio de 2 horas/día.
- Inexistencia de mecanismos de transferencia de conocimiento, teniéndose como técnica única la capacitación basada en clase magistral por parte de integrantes experimentados.
- Incorrecta disciplina de requisitos y arquitectura.
- Enfoque en donde el código fuente se consideraba el único producto. Esta visión impidió que el grupo de trabajo tuviese un conjunto de artefactos que evolucionara de manera consistente.



Figura 1. Fases transcurridas en el desarrollo del SITEM

Fuente: elaboración propia.

SITEM - Fase II

Conscientes de la oportunidad de negocio, se evaluaron los resultados y experiencias de la primera fase. Un análisis de riesgo permitió definir estrategias de tratamiento que incluyeron los siguientes aspectos:

- Adopción de un método de desarrollo guiado por casos de uso, centrado en la arquitectura, iterativo e incremental.
- Fortalecimiento del modelo de transferencia de conocimiento a partir de la creación de objetos de aprendizaje.
- Uso de marco de trabajo (*frameworks*).
- Definición de lineamientos de trabajo.
- Enfoque en la interoperabilidad.
- Limitación del dominio de negocio a los nodos de entidades de salud en sus redes eléctricas y de telecomunicaciones.
- Liberación del código para facilitar el desarrollo colaborativo.

Con la experiencia se comprobó la máxima de Larman (2003): *Rápido, barato, bueno: elija dos cualquiera*. Dejando a un lado las esperanzas utópicas, sacrificamos la rapidez en que versiones estables del proyecto habrían de ver la luz.

La metodología permitió elegir a OpenUP como el método de trabajo y se declaró la fase II como etapa para desarrollar una visión base, planificar el proyecto, preparar el ambiente de trabajo, definir el modelo base de requisitos y lograr acuerdos acerca del enfoque técnico que debe regular el desarrollo.

Los alcances de esta fase fueron:

- Elaboración de la visión general del proyecto.
- Aprobación del plan general del proyecto para las fases II (inicio), III (elaboración y construcción) y IV (transición).
- Modelo base de requisitos.
- Bosquejo de la arquitectura del sistema.
- Modelo de información - segunda versión.
- Definición de los requisitos mínimos del ambiente de trabajo.

OpenSITEM - Fase III

La fase III corresponde a las fases de elaboración y construcción definidas en OpenUP. Para concretar la visión el grupo toma una decisión arquitectónica importante y concentra su esfuerzo en un motor de integración de aplicaciones, cuyo objetivo es la federación de sistemas existentes de software libre o código abierto (FLOSS). De esta manera se logra un mayor nivel funcional del que se habría poder alcanzado de seguir con un desarrollo ciento por ciento *in-house*. Con esto se busca el aseguramiento de

la calidad en el desarrollo, la interoperabilidad, escalabilidad y el uso extensivo de soluciones consolidadas de software libre o de código abierto.

Para recalcar la adopción de la filosofía del código abierto todo el código generado se gestiona en un repositorio público en la plataforma GitHub. De manera complementaria y considerando el producto de manera integral, se elaboran diferentes artefactos y se formalizan las áreas de capacitación a partir de ciclos genéricos de transferencia de conocimiento apoyados en tecnologías de la información.

Es precisamente esta fase la que da vida a este documento, a una arquitectura emergente y una versión 1.0 del sistema denominada *OpenSITEM*, con la que el grupo entrega un sistema complejo constituido por un motor de federación (basado en OpenSARA, un *framework* de desarrollo para aplicaciones web desarrollado por el grupo) que articula las herramientas externas Knowage, ERPNext, Alfresco Community, CAMUNDA y OpenProject, así como varias herramientas propias tales como:

- Sistema Integrado de Evaluación.
- Sistema de Gestión de Redes de Práctica y Colaboración.
- Sistema de Agentes Notificadores y de Recomendación.
- Sistema de Información Geográfica 3D.

De esta forma, OpenSITEM propone una solución que abarca diferentes dominios, pero con capacidad de adaptación para el propósito específico de las necesidades del proyecto *Sistema federado de aplicaciones para la caracterización de proyectos de eSalud*.



Figura 2. Sistema de información para la caracterización de proyectos de eSalud. Interfaz de usuario en la fase III

Fuente: elaboración propia.

Contexto teórico

El grupo GITEM considera que dos aspectos del proyecto son de impacto para el ámbito colombiano: el modelo de dominio e información para la caracterización de nodos potenciales de redes de eSalud y el modelo de federación de aplicaciones que permite el despliegue de soluciones de alta funcionalidad a bajo costo, aprovechando y colaborando con el movimiento de Software Libre y Código Abierto (FLOSS).

Este capítulo provee la base teórica que sustenta los modelos de dominio e información, el marco legislativo y demás conceptos que son primordiales para determinar las características (atributos, indicadores, relaciones, axiomas, categorías) que definen un nodo dentro de una red de eSalud. Teniendo en cuenta el carácter evolutivo del proyecto software y como referencia para posteriores desarrollos soportados en OpenSITEM, se incluye el proceso de ingeniería que guía la integración del sistema.

Redes de eSalud

Conjunto de nodos relacionados con capacidad operativa de brindar servicios de eSalud, garantizando atributos de calidad, estandarización, interoperabilidad, privacidad y seguridad (Wilson, 2017). La arquitectura base de cada nodo de la red es de tipo agregado, con *micronodos* agrupados funcionalmente para brindar un servicio con capacidad de ser consumido por otros nodos, creando estructuras funcionalmente complejas. En este sistema los nodos de más alto nivel de abstracción son los que tienen capacidad de ofrecer servicios de eSalud.

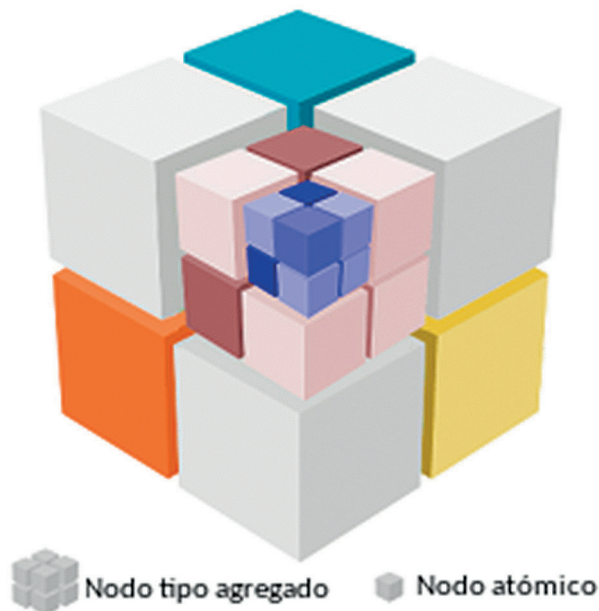


Figura 3. Vista conceptual de la arquitectura de una red de eSalud en OpenSITEM
Fuente: elaboración propia.

De acuerdo con las recomendaciones del grupo GITEM, la descripción de la arquitectura de cada nodo, así como la de la red, puede obtenerse de manera emergente utilizando métodos tales como el Architecture Development Method (ADM) de The Open Group o similares. Se recalca que OpenSITEM no es una plataforma para prestación de servicios de eSalud sino para la descripción de nodos de la red a diferentes niveles de abstracción, a partir del modelo de agregación.

Caracterización de nodos para redes de eSalud

Las redes de eSalud incluyen nodos de diferentes tipos, por ejemplo, médicos, tecnológicos, de recurso humano, financiero, estratégico, de organización, de política y de infraestructura (ITU-OMS, 2012; OMS, 2016; OPS, 2011). En el estudio de campo, el equipo de diseño y planeación de redes de eSalud se encontró con el obstáculo de no disponer de información de calidad de los nodos potenciales que harían parte de la red. Por esta razón y con el objetivo de crear una matriz de comparación, se propuso un *reducido* de nodos⁴ y una descripción de la arquitectura de cada uno de ellos, de tal manera que se pudiese recopilar datos basados en modelos de información.

Al definir el resultado se planteó que un servicio con posibilidad de ser implementado en un modelo de eSalud requiere la integración de nodos. En OpenSITEM existen dos tipos de nodos: (1) atómicos: aquellos que ofrecen un servicio cohesivo que no abarca un caso de uso de dominio y (2) agregados: que articulan nodos (atómicos o agregados) para brindar capacidad funcional a nivel de caso de uso de negocio. Si se considera que el modelo de dominio gira en torno a los servicios de eSalud, el nodo agregado que lo provee se denominará *macronodo*.

Un nodo atómico se caracteriza por atributos (propiedades) e interfaces. Estas últimas son los puntos de acceso donde los servicios del nodo son expuestos para ser consumidos (The Open Group, 2016). Un nodo agregado está definido por el conjunto de atributos de los nodos que lo componen y las interfaces que surgen de la interoperabilidad de dichos nodos. Un nodo puede pertenecer a uno o más agregados, razón por la cual un nodo, atómico o agregado, goza de alta cohesión y bajo acoplamiento como patrón principal⁵.

Otro aspecto que se utiliza para la descripción de los nodos es la categoría o clase a la cual pertenece. Debido a que la arquitectura de la red puede ser abordada desde distintos puntos de vista, un nodo puede tener asociada uno o varias categorías. Por ejemplo, el modelo de componentes que propone la OMS (ver figura 4) podría definir las categorías de clasificación de los nodos. Posteriormente a ello, si surge otro modelo o se edita el actual, los nodos podrían adaptarse para mantener las relaciones y crear unas nuevas. El grupo de investigación desarrolla OpenSITEM enfocado en la caracterización multicategoría de los nodos en redes de salud, tomando como modelo de mayor nivel de abstracción el propuesto por la OMS y uno de baja abstracción con

4 El reducido es un conjunto finito de nodos con capacidad de entregar una funcionalidad contractualmente definida relacionada con la eSalud.

5 En el mayor grado en que se cumplan estos patrones más simple resulta la caracterización de la red.

categorías: médica, tecnológica, administrativa, humana y financiera. Sin embargo, debido al perfil de profesionales que han participado en el desarrollo, los nodos de la categoría tecnológica han sido mejor caracterizados hasta el momento.



Figura 4. Componentes en un sistema de eSalud
Fuente: elaboración propia a partir de WHO (2016).

Subsistemas en una red de eSalud

El grupo GITEM considera que una red de eSalud está compuesta por cuatro subsistemas:

- Subsistema de captura de datos: conformado por los dispositivos de hardware, los protocolos y aplicaciones software que trabajan conjuntamente para transformar información médica en datos susceptibles de ser administrados usando técnicas digitales.
- Subsistema de transmisión de datos: hacen parte de este subsistema los dispositivos de hardware, las tecnologías de interconexión, los protocolos y aplicaciones que permiten estructurar redes de transmisión de datos digitales de una manera fiable en tiempos aceptables para un servicio específico.
- Subsistema de gestión de información: dispositivos de hardware —computadores, sistemas de almacenamiento masivo, etc.— y sistemas de información que almacenan, procesan, distribuyen, analizan, integran y producen información con base en los datos de los subsistemas de captura, históricos y de pronóstico.
- Subsistema de despliegue de información: elementos de hardware (pantallas, transductores, sistemas de audio, etc.), aplicaciones software y protocolos asociados que permiten recibir y reproducir información médica.

Todos ellos sirven como plataforma para determinar cierta organización para la definición de tipos de nodos.

eSalud

Si se tiene en consideración que el servicio de más alto nivel de la red es la eSalud, es necesario abordar su concepto con el fin de tener un marco para la comprensión del modelo de dominio que se presentará en un capítulo posterior.

La eSalud es definida por la organización Mundial de la Salud (OMS) (2016) como:

el apoyo que la utilización costo eficaz y segura de las tecnologías de la información y las comunicaciones ofrece a la salud y a los ámbitos relacionados con ella, con inclusión de los servicios de atención de salud, la vigilancia y la documentación sanitarias, así como la educación, los conocimientos y las investigaciones en materia de salud. (p. 145)

En sí es un término que define un campo multidisciplinar que integra componentes de diferentes áreas del saber, incluidas, entre otras, la medicina, la ingeniería electrónica, la telemática, la informática, la ingeniería de sistemas, la inteligencia artificial, la biónica, la psicología, la sociología, la antropología y las geociencias. Las redes eSalud consideran elementos que van mucho más allá del simple despliegue de redes tecnológicas de intercomunicación y se plantean como redes de interacción social cuyo objetivo primario, mas no el único, es la prestación de servicios médicos y relacionados.

De acuerdo con el grado en que se presente cada uno de los elementos mostrados en la figura 5 y de la mayor o menor correlación entre ellos, se pueden crear sistemas de eSalud que se acerquen al ideal de proveer servicios de salud de alta calidad.

La OMS y la Unión Internacional de Telecomunicaciones (ITU) proveen un método que podría ser utilizado para abordar el desarrollo de proyectos de eSalud complejos, dinámicos y evolutivos (ITU-OMS, 2012).

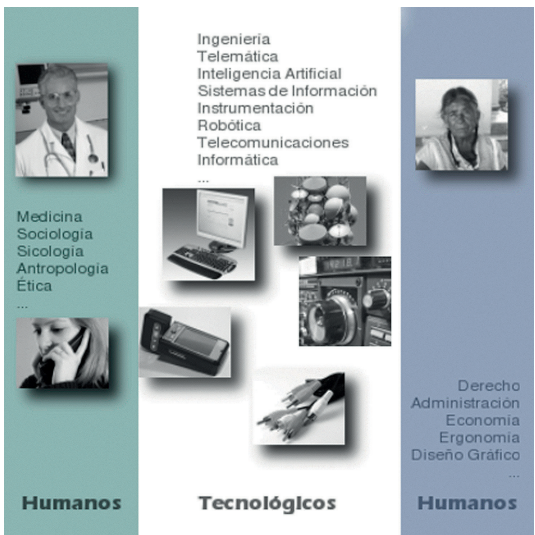


Figura 5. Modelo simple de categorías para nodos de una red de eSalud
Fuente: elaboración propia.

Componentes principales de eSalud

La eSalud comprende áreas que no tienen una división concreta pero se enfocan en ciertos aspectos de interés:

- Telemedicina: provisión remota de servicio clínicos.
- Telesalud: telemedicina, complementada con la prestación remota de otros servicios tales como entrenamiento médico, monitoreo de pacientes, cuidado médico, gestión administrativa, etc.
- mSalud: telesalud con el apoyo de dispositivos móviles.
- Registro médico electrónico: conocido como historia clínica electrónica. Comprende los mecanismos que permiten registrar en un entorno digital seguro, la información sobre los eventos de salud de cada paciente.
- eAprendizaje/eEnseñanza: servicios de enseñanza/aprendizaje de ciencias médicas y afines, en modalidad a distancia o virtual.

Vale la pena aclarar que la pluralidad de componentes no es más que un esfuerzo para abordar en un único modelo todas las tendencias que se han presentado en la evolución de la eSalud. En alguna literatura los términos son intercambiables, dependiendo el enfoque del autor (OMS, 2010).

Marco legislativo y normativo

OpenSITEM es un sistema que gestiona información sobre diferentes áreas del saber, de acuerdo con las categorías de los nodos que se definan. Cada aspecto (atributos, interfaces, interoperaciones) está en un marco legislativo y normativo concreto, o está asociado a estándares y normas de uso extendido o de facto aceptadas en el mundo. La información de OpenSITEM que sea de acceso público no podrá incluir protección por derechos de autor que restrinjan su difusión (Congreso de la República, 1982, 2000). Debido a esto, el código fuente de OpenSITEM es cubierto por una licencia abierta que se ciñe a la normatividad expresada en la Ley 565 de 2000: adopción del Tratado de la OMPI sobre derechos de autor y complementarias, para garantizar que todos los aspectos tanto técnicos como conceptuales estén debidamente registrados (Congreso de la República, 1989, 1993).

OpenSITEM es una plataforma para la definición de nodos y no es posible a priori definir el marco legislativo que regirá cada nodo o categoría. Sin embargo, a continuación se describe un marco relacionado con las categorías base que se han definido en el primer modelo del sistema.

En el ámbito de los servicios médicos

El derecho a la salud ha sido reconocido por normas y pactos internacionales contenidos en tratados sobre derechos humanos, económicos, sociales y culturales (DHESC). Esos acuerdos han sido ratificados por Colombia para su cumplimiento como un derecho de los ciudadanos:

según el inciso segundo del art. 93 de la Constitución, los derechos y deberes consagrados en ésta se interpretarán, de conformidad con los Tratados Internacionales sobre derechos humanos ratificados por Colombia. Ello significa, que para efectos de interpretar los referidos derechos las normas de los tratados en referencia tienen carácter prevalente en el orden interno, formando por lo tanto parte del bloque de constitucionalidad, como lo ha reconocido la Corte en diferentes pronunciamientos. (Corte Constitucional de Colombia, 2001)

Así ocurre con el derecho a la salud. Al Ministerio de Salud y Protección Social le corresponde expedir las normas técnicas y administrativas de obligatorio cumplimiento para las entidades promotoras de salud del régimen contributivo, las instituciones prestadoras de salud del Sistema General de Seguridad Social en Salud, las administradoras del régimen subsidiado y para las direcciones seccionales, distritales y locales de salud en cuanto al objetivo de cumplimiento en el desarrollo de actividades de protección específica, detección temprana y atención de enfermedades de interés en salud pública.

A continuación se registra la normatividad que se tuvo en cuenta al momento de definir los componentes actuales del subsistema de servicios médicos, en cuanto a la relevancia que se tiene tanto para la proyección de nuevas redes de eSalud como para apoyar los sistemas básicos ya existentes. Es de anotar que lo contemplado en las leyes nacionales es, en su mayoría, derivado de normas internacionales que han sido objeto de detallados estudios y reconocidas técnicamente con base en las experiencias vividas por los profesionales de esta área.

Ley 1751 de 2015

Por medio de la cual se regula el derecho fundamental a la salud. Es una ley estatutaria que surge a partir de la debacle del proceso de reforma y tiene como efecto positivo el elevar a la salud como un derecho fundamental. Entró en vigor a partir del año 2017 y da lineamientos para reestructurar el sistema de salud a partir del desarrollo de *redes de servicios* públicos, privados o mixtos. También declara la necesidad de establecer políticas relacionadas con la salud tales como la política para la información, la política de innovación, ciencia y tecnología y la política farmacéutica nacional.

Ley 1419 de 2010

Por la cual se establecen los lineamientos para el desarrollo de la telesalud en Colombia. Define las redes de telesalud y el aprendizaje en telesalud como ejes principales de la gestión del conocimiento en salud. Si bien esta ley obliga a desarrollar el mapa de conectividad, en el 2018 aún no se encontraba uno disponible para los ciudadanos.

Resolución 2182 de julio 9 de 2004

Con esta resolución se definían las condiciones de habilitación para las instituciones que prestan servicios de salud bajo la modalidad de telemedicina. Fue derogada por el artículo 11 de la Resolución 1043 de 2006, con la cual se establecen las condiciones que deben cumplir los prestadores de servicios de salud para habilitar sus servicios e implementar el componente de auditoría para el mejoramiento de la calidad de la atención y se dictan otras disposiciones.

Posteriormente, la Resolución 1448 de 8 de mayo de 2006 regula la prestación de servicios de salud bajo la modalidad de telemedicina y establece las condiciones de habilitación de obligatorio cumplimiento para las instituciones que prestan servicios de salud. Esta resolución aclara que las actuaciones de los médicos en el ejercicio de la prestación de servicios bajo la modalidad de telemedicina se sujetarán a las disposiciones establecidas en la Ley 23 de 1981 y demás normas que la reglamenten, modifiquen, adicionen o sustituyan.

Resolución 4678 de noviembre 11 de 2015

Con esta resolución, modificada por la Resolución 1113 de 2017, el Ministerio de Salud y Protección Social adopta una Clasificación Única de Procedimientos en Salud (CUPS) la cual “[...] corresponde a un ordenamiento lógico y detallado de los procedimientos y servicios en salud que se realizan en el país, en cumplimiento de los principios de interoperabilidad y estandarización de datos utilizando, para tal efecto, la identificación por un código y una descripción validada por los expertos del país” (Ministerio de Salud, 2015). La Clasificación Única de Procedimientos en Salud adaptación para Colombia, se implementa por Resolución 365 de 1999. Su primera publicación se presenta en un solo volumen que contiene la lista tabular y el índice alfabético. A partir de dicha resolución se realizó la primera actualización de la CUPS (1ºA-CUPS) mediante la Resolución 2333 de 2000. En el año 2016, por la Resolución 3804, se establece el procedimiento para la actualización de la CUPS, con lo que el Ministerio espera darle una mayor agilidad al proceso.

Resolución 1830 de junio 23 de 1999

Adopta para Colombia “Las codificaciones únicas de especialidades en salud, ocupaciones, actividades económicas y medicamentos esenciales” para el Sistema Integral de Información del SGSSS – SIIS.

Resolución 1895 de noviembre 19 de 2001

Por la cual se adopta para la codificación de morbilidad en Colombia, la Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud - décima revisión.

Resolución 1995 de julio 8 de 1999

Por la cual se establecen normas para el manejo de la historia clínica. La historia clínica es un documento de vital importancia para la prestación de los servicios de atención en salud y para el desarrollo científico y cultural del sector, *es un documento privado, obligatorio y sometido a reserva*, en el cual se registran cronológicamente las condiciones de salud del paciente, los actos médicos y los demás procedimientos ejecutados por el equipo de salud que interviene en su atención. Dicho documento únicamente puede ser conocido por terceros con previa autorización del paciente o en los casos previstos por la ley.

Circular 015 de abril 4 de 2002

Estándar de historias clínicas y registros, establece la obligatoriedad de definir procedimientos para utilizar una historia única institucional. Ello implica que la insti-

tución cuente con un mecanismo para unificar la información de cada paciente y su disponibilidad para el equipo de salud. No es restrictivo en cuanto al uso de medio magnético para su archivo, y sí es expreso en que debe garantizarse la confidencialidad y el carácter permanente de registrar en ella y en otros registros asistenciales.

Decreto 2092 de 2 de julio de 1986

Por el cual se reglamentan parcialmente los títulos VI y XI de la Ley 09 de 1979, en cuanto a la elaboración, envase o empaque, almacenamiento, transporte y expendio de medicamentos, cosméticos y similares. Se dan las disposiciones generales y definiciones, el registro sanitario de medicamentos, cosméticos y similares.

En el ámbito del desarrollo de redes teleinformáticas

Documento Conpes 3072

Aunque no es una norma regulatoria, es una declaración oficial del Gobierno colombiano acerca de la necesidad de fomentar las tecnologías de la información para potenciar la absorción, creación y divulgación del conocimiento por medio del desarrollo sostenible en las infraestructuras física, de información y social. Según el Ministerio de Comunicaciones (2000): “[...] para que el país pueda ofrecer un entorno económico atractivo y participar en la economía del Conocimiento, resulta indispensable desarrollar una sociedad en la que se fomente el uso y aplicación de las Tecnologías de la Información. A través de estas Tecnologías, se puede efectuar un salto en el desarrollo en un tiempo relativamente breve, mucho menor del que se necesita para superar el déficit de infraestructura física”(p. 6).

El documento Conpes brinda un referente válido pues la mayoría de los objetivos estratégicos del OpenSITEM contienen el espíritu expresado en diferentes partes de este.

Documento Conpes 3582

Política Nacional de Ciencia, Tecnología e Innovación. En el cual se enfatiza el desarrollo de la salud y la tecnología como mecanismos de generación de valor social.

Resolución 087 de 1997

Por medio de la cual se regulan en forma integral los servicios de telefonía pública básica conmutada (TPBC) en Colombia y claramente se expresa:

Los servicios de TPBC deberán ser utilizados como instrumento para impulsar el desarrollo político, económico y social del país con el objeto de elevar el nivel y la calidad de vida de los habitantes en Colombia. Los servicios de TPBC serán utilizados de manera responsable para contribuir a la defensa de la democracia, a la promoción de la participación de los colombianos en la vida de la Nación y la garantía de la dignidad humana y de otros derechos fundamentales consagrados en la Constitución Política, y para asegurar la convivencia pacífica.

Esta resolución reviste particular importancia para la extracción de elementos semánticos y algunos componentes necesarios en las redes de telecomunicaciones basadas

en telefonía conmutada. Algunas heurísticas usadas en el subsistema de consultoría también se basan en apartes de esta resolución.

Manual de Calidad de Servicio

“Con este Manual de calidad de servicio (QoS) se especifican los parámetros de calidad de servicio de red que permiten el suministro de servicios a los clientes y los usuarios, satisfaciendo sus expectativas de calidad de servicio. Estos parámetros tienen que ver tanto con la implementación del servicio como con su utilización continua. Asimismo, la calidad de servicio se relaciona con todos los aspectos relativos a la evaluación y gestión de las redes” (ITU-T, 2004)

Sus principales aspectos son recogidos por la Comisión de Regulación de Telecomunicaciones (2006a, 2006b) y sirven de base para el proyecto de resolución de dicha entidad (Comisión de Regulación de Comunicaciones, 2007) que especifica, entre otros: las definiciones relativas a la QoS, parámetros de medición, variables y propiedades técnicas de diferentes enlaces. Aunque el objetivo principal de este manual es garantizar la QoS en un sistema de telecomunicaciones dado, es importante notar que sus indicaciones deben ser tenidas en cuenta al momento de proyectar los servicios de telecomunicaciones en un sistema de telemedicina dado. En estos aspectos también se consideran dentro del modelado de ciertos componentes del OpenSITEM las recomendaciones de la UIT (ITU-T, 2001a, 2001b), las cuales aún no tienen un equivalente en la normatividad colombiana pero son de uso extendido alrededor del orbe.

En el ámbito del tratamiento de datos e información

Leyes 1581 de 2014 y 1266 de 2008

Estas leyes se relacionan con la protección de los datos personales y el aseguramiento de la privacidad de la información personal.

Aspectos clave en la gestión y dirección del conocimiento

Como se aclaró anteriormente, OpenSITEM es una federación de diferentes tipos de sistemas software y sirve de soporte para el trabajo de los equipos de diseño de redes de eSalud del grupo de investigación GITEM (o similares). En esa línea, parte de su funcionalidad está enfocada en brindar apoyo a los flujos de trabajo de comunidades de práctica, y por tanto es necesario que se aborden aspectos relacionados con la gestión del conocimiento. A continuación se presenta la aproximación conceptual de esta área.

Acerca del conocimiento

En la presente investigación, el *conocimiento* se entiende como un concepto que encierra, entre otras, las siguientes definiciones, que se usarán de acuerdo con su contexto como partes complementarias de una misma meta-definición.

De acuerdo con Gunter Dueck (2001), es un concepto que puede tener componentes en una o varias de las siguientes dimensiones:

- Episteme: *dimensión abstracta o metafísica*, en forma de generalizaciones, bases, leyes y principios científicos.
- Phronesis: *dimensión práctica*, relativa al conocimiento pragmático discernido a través de las prácticas aceptadas por la sociedad.
- Techne: *dimensión técnica*, relativa a la forma de hacer las cosas, de la realización de actividades concretada en la forma de manuales, procedimientos y comunidades de práctica.
- Metis: *dimensión objetiva*, como forma de hacer corpórea, real y sustancial la conjugación de los otros tipos de conocimiento.

Esta concepción multidimensional extiende y explica la taxonomía del conocimiento formulada por los antiguos griegos:

El conocimiento incluye restricciones implícitas y explícitas entre objetos (entidades), operaciones y relaciones, que permiten recoger heurísticas generales y específicas así como los procedimientos de inferencias relacionados con la situación a modelar. (Sowa, 1994, p. 52)

Otros autores despojan del sentido filosófico y colocan su definición en un plano simple y utilitario: “El conocimiento es información organizada y analizada para hacerla comprensible y aplicable a la resolución de problemas y toma de decisiones”. Si bien esta es una definición reduccionista, sirve de base para las propuestas de representación de conocimiento en documentos XML comúnmente denominadas *ontologías*.

Ciclo de conocimiento

Múltiples factores deben ser considerados cuando se trata de capturar, crear y diseminar el conocimiento dentro de un grupo de personas. La no homogeneidad en los medios de almacenamiento de conocimiento es uno de ellos. El conocimiento —en cualquiera de sus formas—, puede estar guardado en diferentes partes, que van desde entidades biológicas —mente humana, genes— a repositorios de conocimiento estructurado tales como ontologías, grafos de relación o mapas mentales.

Otro factor importante es la capacidad de acceso al conocimiento, que es muy diferente al mero hecho de acceder a una fuente de conocimiento dada. Nokata (1995) considera que el conocimiento puede hallarse en dos estadios con propiedades diferentes: tácito y explícito.

- Conocimiento tácito: se corresponde con el conocimiento obtenido a través de la experiencia, conocimiento simultáneo y conocimiento análogo. Esta forma usualmente se encuentra en medios de almacenamiento biológicos como la mente humana.

- Conocimiento explícito: se corresponde con el conocimiento racional, secuencial y digital y se encuentra almacenado en documentos, bases de conocimiento, ontologías o cualquier otro medio abstracto de representación.

Liebowitz (1998) establece un tercer estadio llamado conocimiento implícito.

- *Conocimiento implícito*: acceso directo mediante consulta y discusión. Requiere la localización y comunicación previa de conocimiento informal.

Un ciclo de conocimiento es el proceso por el cual el conocimiento se transforma de tácito a explícito y viceversa, por medio de las siguientes actividades:

- Socialización: compartir conocimiento tácito entre individuos. El conocimiento continúa siendo tácito sin ser transformado en explícito. Este tipo de patrón no es muy interesante, debido a su naturaleza tácita (tácito-tácito).
- Articulación: alguien transforma el conocimiento tácito en explícito (tácito-explícito).
- Síntesis: combinación de conocimiento explícito para crear nuevo conocimiento explícito (explícito-explícito).
- Interiorización: proceso de transformar conocimiento explícito en tácito (explícito-tácito).

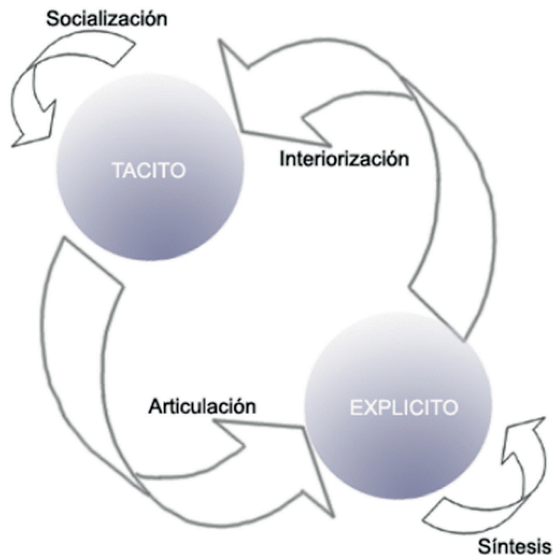


Figura 6. Ciclo de conocimiento
Fuente: elaboración propia.

El flujo de conocimiento organizacional más importante es la transformación del conocimiento tácito en explícito (Davies, 2011), esto es, la *articulación* que se apoya en procesos avanzados de socialización. Esto permite acumular conocimiento explícito que puede ser compartido y accedido por los miembros de la organización. Por el contrario, la interiorización es el proceso natural llevado a cabo a través del aprendizaje individual por parte de los integrantes de la organización, es decir, la asimilación de conocimiento.

El tercer flujo de conocimiento relacionado con la transformación de conocimiento es la *combinación* o síntesis. En este caso se transfiere conocimiento a otra forma explícita de conocimiento. Un ejemplo sería cambiar el formato de una base de conocimiento, agrupar ontologías o refinar heurísticas. Este tipo de flujo de conocimiento es importante para seleccionar, combinar y distribuir el conocimiento existente con diferentes fines. Por ser quizás el flujo de conocimiento más formal en OpenSITEM, algunos componentes específicos implementan flujos de síntesis de conocimiento.

El cuarto flujo del ciclo de vida del conocimiento permite transformar conocimiento tácito en otras formas de conocimiento tácito, mediante procesos de socialización. Un ejemplo de esto es cuando se transfiere conocimiento tácito de un experto a un ingeniero de conocimiento en una entrevista personal.

Tecnologías del conocimiento

En la actualidad, el conocimiento se considera un activo fundamental y, como lo expresa Gang (2007), tiene dos propiedades de vital importancia para contribuir al desarrollo de las organizaciones o comunidades de práctica:

- Es explicable: cuando no se evidencia esta propiedad, el conocimiento permanece tácito y en ninguna medida puede considerarse como perteneciente a la comunidad o la organización. Es entonces una tesis que solo aquel conocimiento que se ha convertido en explícito es el que posee una organización; de otra manera es propiedad exclusiva del individuo.
- Se puede comunicar y compartir: cuando no se toman medidas que formalicen estas actividades, el conocimiento se pierde, es decir, la organización o comunidad debe tener procesos conocidos que garanticen flujos de conocimiento basados en la síntesis, interiorización y socialización.

En general, para que el conocimiento pueda generar ventajas competitivas debe ser gestionado de alguna manera. Esta necesidad implicó el surgimiento de dos áreas de investigación comúnmente agrupadas bajo el concepto de tecnologías de conocimiento: ingeniería de conocimiento y gestión de conocimiento.

Uno de los primeros problemas que deben atacar las tecnologías de conocimiento es el asociado con el modelado de conocimiento. Benafia (2016) expresa algunos principios para tener en cuenta cuando se modela el conocimiento:

- Definición de roles de conocimiento: el conocimiento se puede dividir en unidades atómicas que tienen propiedades irreductibles y que se asocian para lograr funciones que identifican dicha unidad.

- Identificación de tipos de conocimiento: cada unidad de conocimiento debe enmarcarse en uno o varios de los siguientes tipos de conocimiento: de tareas, inferencial, del dominio, ontologías del dominio, modelos del dominio.
- Capacidad de ser compartido y reutilizado: las unidades de conocimiento pueden ser expresadas usando lenguajes y reglas formales, lo que permite que pueda ser entendido por entidades con roles de conocimiento definidos.
- Uso de modelos gráficos: las unidades de conocimiento pueden ser representadas mediante grafos tipo red en los cuales tanto los nodos como las rutas de interconexión son unidades atómicas de conocimiento.

Gestión de conocimiento

La gestión del conocimiento es de esos conceptos *polisémicos* que no pocos autores pretenden consolidar en una sola definición (Girard, 2015). En este proyecto partimos de una visión sistémica que lo trascienda a lo que en investigación holística se comprende como sintagma. La *gestión de conocimiento*, como se percibe en la presente investigación, es “una unión sintagmática de diversos paradigmas” (Hurtado, 2000). Fue Karl Wiig quien usó el término de *gestión de conocimiento* por primera vez, durante una conferencia en Suiza, y a partir de ese momento diversos autores lo han conceptualizado, surgiendo definiciones parciales como:

La Gestión de Conocimiento es la construcción y aplicación sistemática, explícita y deliberada de conocimiento para maximizar la efectividad organizacional con respecto al conocimiento, por lo que usa sus activos de conocimiento. (Wiig, 1993, p. 12)

La Gestión de Conocimiento es el proceso de capturar experiencia colectiva organizacional donde ésta resida (por ejemplo, bases de datos, documentos, mentes humanas) y su distribución allá donde pueda ayudar a mejorar los resultados. (Hibbard, 1997, p. 35)

La Gestión de Conocimiento es la gestión y control explícito del conocimiento en una organización para lograr los objetivos de la organización. (van der Spek y Spijkervet, 1997, p. 7)

Consecuente con la concepción de la gestión de conocimiento como un sintagma, se puede concebir un paradigma asociado a un proceso con ciertas actividades implicadas:

- Identificación y mapeo de bienes intelectuales de la organización.
- Generación de conocimiento nuevo que permita obtener una ventaja competitiva.
- Recopilación accesible de información organizacional.
- Compartir buenas prácticas y tecnología, incluyendo técnicas de trabajo en grupo.

Este paradigma explora el conocimiento técnico, pragmático y objetivo, considerando que la conjugación de leyes rígidas y epistemológicas va en contravía de la dinámica misma de los sistemas. La articulación de este conocimiento debe ser mantenida de alguna forma en la organización, y de ahí surge el concepto de *memoria corporativa*. El saber hacer está completamente diseminado en la organización y debe ser integrado de forma coherente para facilitar su acceso y reutilización, esto es, expresarlo en forma de memoria corporativa.

Las memorias corporativas se consideran un elemento clave para gestionar el conocimiento porque facilitan su conservación, distribución y reutilización. Van Heijst, Van der Spek y Kruizinga (1997) definen la memoria corporativa como una “representación de conocimiento e información organizacional explícita y persistente” (p. 43), mientras que Nagenda y Plaza (1996) lo definen como “los recursos colectivos de datos y conocimiento de una compañía, incluyendo experiencias en proyectos, experiencia en resolución de problemas, etc.” (p. 20). Para Abecker (1998), una memoria corporativa es referida como “un contenedor que integra información contextual, documentos e información no estructurada, que facilita su uso y reutilización” (p. 19).

Sistemas de gestión de conocimiento

En IA, las bases de conocimiento son generadas para ser consumidas en sistemas expertos y basados en conocimiento, donde las computadoras usan inferencias para responder a cuestiones de usuario. Aunque es importante la adquisición de conocimiento para inferencias computacionales, en los desarrollos más recientes en gestión del conocimiento, el conocimiento queda disponible para consumo humano directo o para desarrollar software que procese dicho conocimiento.

Históricamente, la gestión de conocimiento se ha centrado en un único grupo, a través de lo que generalmente se ha conocido como sistema de información ejecutiva (EIS), que contiene un conjunto de herramientas para acceder a bases de datos, generar alertas, etc., para apoyar el proceso de toma de decisiones. Más recientemente, se ha comenzado a diseñar sistemas de gestión de conocimiento para organizaciones completas. Si los ejecutivos necesitan acceder a la información y al conocimiento, es probable que sus empleados tengan interés en esa información.

De acuerdo con O’Leary (citado en Valencia, 2005), las principales funciones de un sistema de gestión de conocimiento son facilitar:

- La conversión de datos y texto en conocimiento.
- La conversión de conocimiento individual y de grupo en conocimiento explícito.
- La conexión de individuos y conocimiento a otros individuos y conocimientos.
- La comunicación de información entre diferentes grupos.
- La creación de nuevo conocimiento útil para la organización.

Sistemas integrados para el soporte de desempeño

De acuerdo con Winslow y Bramer (1994), son sistemas que integran múltiples fuentes y herramientas de gestión de conocimiento en un único ambiente de trabajo para apoyar de una manera más efectiva las tareas relacionadas con:

- Infraestructura: organización y estructura del entorno de trabajo.
- Control: monitorización, coordinación y control.
- Navegación: interacción hombre-máquina.
- Presentación: posibilidad para personalizar datos y servicios.
- Adquisición: captura conocimiento, casos, opiniones, aprendizaje y datos sensoriales en diferentes medios y su transformación en formato interno.
- Consultoría: consultar servicios, asistencia y recordatorios.
- Instrucción: ayuda y entrenamiento.
- Aprendizaje: aplicación de técnicas de descubrimiento de conocimiento y minería de datos.
- Evaluación: valoración y certificación basada en medidas del rendimiento y la calidad.
- Referencia: constituyen fuentes de conocimiento y experiencia para la organización.

En la actualidad, estos sistemas se han convertido en una necesidad debido al crecimiento desmesurado de aplicaciones incompatibles y protocolos no estandarizados.

Ingeniería de software

A medida que el marco conceptual del OpenSITEM crece, el cúmulo de nuevos requerimientos —no vislumbrados en su planteamiento inicial— fomenta que los riesgos asociados al desarrollo de los componentes también crezcan. Lo que en un principio no era más que una “herramienta” para la administración del acervo documental fruto de una investigación, se convirtió en una propuesta de sistema de información y conocimiento que suponía un reto novedoso dentro del grupo de investigación.

Es evidente que se deben sumar nuevos saberes para procurar manejar formalmente el proceso de elaboración del sistema; un punto inicial y obligado de estudio se centró en la ingeniería de software.

Como rama de la ingeniería comparte la definición fundamental que de esta brindó a mediados del siglo pasado el Consejo de Ingenieros para el Desarrollo Profesional (ECPD, por sus siglas en inglés) y que en general propone:

Ingeniería

Es la aplicación creativa de principios científicos para el diseño o desarrollo de estructuras, máquinas, aparatos, procesos de manufactura o sistemas genéricos, para ser usados de forma independiente o combinados; o su construcción y operación de con total conocimiento de su diseño; o el pronóstico de su comportamiento bajo ciertas condiciones de operación. Todo aquello respecto a una funcionalidad esperada asegurando economía en el manejo de los recursos y con seguridad para la vida y la propiedad⁶.

Se concibe entonces a la ingeniería de software como la aplicación de los principios de ingeniería a los sistemas de software, con base en “un acercamiento sistemático, cuantificable y disciplinado del desarrollo, operación y mantenimiento de software”(IEEE, 1990), y ciertamente se fundamenta en actividades interrelacionadas, propias del ser humano cognoscente y creativo que Brugge y Dutoit (2000) identifican como:

- Actividades de modelado: para abstraer la complejidad del dominio del problema en unidades factibles de ser objeto de estudio y análisis. En este contexto, las nociones de contratos funcionales e independencia conceptual desempeñan un rol importante. Se pretende con estas actividades obtener dos modelos, uno de análisis —como abstracción de alto nivel de la realidad (contexto problémico) y uno de diseño que represente el dominio de la solución de algún problema dentro del modelo de análisis.
- Actividades de resolución de problemas: siendo la ingeniería de software un proceso guiado de búsqueda de solución a un problema específico del ser humano que puede ser apoyado por sistemas software. Se concibe actualmente como un proceso investigativo que, de acuerdo con un acercamiento holístico, contempla flujos de trabajo continuos y evolutivos de exploración, descripción, análisis, comparación, explicación, predicción, proposición, modificación, confirmación y evaluación.
- Actividades de adquisición de conocimiento: durante el desarrollo del sistema software, el ingeniero, a partir de un modelo constructivista, recrea constantemente su conocimiento tácito a partir de las nuevas experiencias y el mayor conocimiento del dominio del problema, así como de los diferentes paradigmas usados en la consecución de soluciones óptimas. En realidad, el ingeniero, así como los demás actores que intervienen en el sistema, ven revalidados o reformados sus conocimientos a medida que los requerimientos son cumplidos y los riesgos minimizados.

También contempla actividades propias de trabajo colaborativo que producen integración de saberes en ambientes inter, trans y multidisciplinarios, lo que potencia efectivamente la creación de ciclos de conocimiento que contribuyen al refinamiento continuo del sistema como objeto perfectible, y del conocimiento directo, que tanto

⁶ Adaptación de la definición hecha por los autores.

del sistema como del proceso tienen los actores vistos como sujetos perfectibles, racionales y cognoscentes, dentro de una dinámica de retroalimentación entre el sujeto que crea el sistema y el sistema mismo.

Proceso de desarrollo del software

Un proceso de desarrollo de software puede ser visto como el conjunto de actividades que debe realizar un grupo de personas para dar solución —mediante un sistema software— a un problema cuyas características y condiciones de resolución han sido especificadas. El producto final, el software, es un *sistema*; es decir, un conjunto de componentes funcionales que se relacionan por medio de interfaces definidas logra el objetivo común de solucionar los problemas determinados.

Para determinar el proceso más apropiado según las necesidades y especificidades del proyecto se condujo una metodología⁷ centrada en diferentes modelos ampliamente aceptados en el campo de la ingeniería de software, lo que al final dio lugar a un proceso consolidado de guía para el desarrollo del Sistema de Información para Proyectos de Telemedicina. Se aclara que este modelo, como el sistema y los actores, no es indiferente al proceso evolutivo de adaptación de conocimiento, por lo que en realidad no se considera como una fórmula mágica, sino simplemente como un caso específico que aporta unos lineamientos interesantes para otros proyectos de software similares y sirve de base para los ingenieros de proceso de fases posteriores en el ciclo de vida del macroproyecto OpenSITEM. En últimas, un sistema software exitoso es aquel en el cual todos sus componentes se refinan constantemente y el proceso de desarrollo es un componente nuclear que tiene mayor incidencia.

Existen tantos procesos de desarrollo de sistemas en el mundo que la mera enumeración taxativa podría cubrir cientos de páginas. El dilema de cuál es el mejor de ellos es irresoluble, sin embargo, se pueden definir las características óptimas para un contexto en particular, teniendo en cuenta múltiples indicadores a partir de aspectos tales como:

- Tamaño del grupo de desarrollo.
- Presupuesto.
- Límites de tiempo.

Principios de diseño y desarrollo

A través del tiempo se han decantado ciertas prácticas que son reconocidas como las óptimas cuando se trata de construir un sistema software de gran magnitud, tanto en líneas de código como en funcionalidad y recursos involucrados. Estos principios son tenidos en cuenta con independencia del proceso de desarrollo que se siga. Quizás

7 La metodología no fue exhaustiva y se limitó a un grupo muy reducido de elementos cuya caracterización se basó exclusivamente en indicadores de tipo cualitativo. Se recomienda remitirse a Carty (2006), Pressman (2006), Jacobson (2000) y Koch (2005), entre otros, para detalles de los diferentes procesos.

los de mayor difusión son los *patrones* GRASP (acrónimo de General Responsibility Assignment Software Patterns), que se basan en la asignación precisa de responsabilidades a cada uno de los componentes del sistema software⁸.

En el desarrollo de OpenSITEM se recomienda, como estrategia para mantener la calidad del software, que los integrantes del grupo tengan en cuenta y adquieran competencias en el manejo de los siguientes patrones y principios⁹:

- Modularidad: para facilitar las tareas de mantenimiento, depuración e incremento en la funcionalidad, se requiere que el sistema se implemente con base en componentes que presenten propiedades de *alta cohesión* funcional entre sus elementos y tengan *bajo acoplamiento* entre sí. La alta cohesión funcional tiene en cuenta que el componente realiza solo tareas relacionadas y utiliza un conjunto de datos homogéneo. El bajo acoplamiento se refiere al hecho de que un componente; se relaciona con otro a través de una interfaz estable y definida. Dicha relación no está supeditada a la implementación interna de ninguno de los componentes; con bajo acoplamiento un cambio en un componente no requeriría ningún cambio en la implementación del componente asociado.
- Prueba continua: todos los módulos y sus componentes deben ser probados en cuanto su funcionalidad y el cumplimiento de los demás principios y patrones. Las actividades de prueba podrán ser automatizadas o realizadas manualmente, pero en cualquier caso deben ser formalmente documentadas. Es una recomendación que en lo posible el personal de prueba sea diferente a aquel que ha diseñado o construido el componente.
- Codificar claramente: la forma en que se ingresa el código o se agrupa un conjunto de elementos gráficos en un diagrama debe ser hecha de tal forma que se facilite su comprensión. Se recomienda el uso de comentarios para aquellas partes del código cuya funcionalidad no sea evidente o cuando se evite el tener que analizar piezas de código extensas.
- Abstracción funcional por capas: los diferentes componentes del OpenSITEM deberán centrar su funcionalidad en tres capas principales: datos, aplicación e interfaz. Las unidades que manejen cada una de las capas deben propender por conservar la modularidad.
- Reutilización: los diferentes componentes del OpenSITEM —denominados bloques dentro del modelo de desarrollo— deben estar codificados de tal forma que puedan ser fácilmente adaptados en los diferentes módulos sistema.
- Recreación de componentes: se debe conocer la estructura interna de un determinado componente para poder sugerir mejoras. Este principio no pretende

8 Aún el Object Management Group declara el uso de ciertos principios de diseño en el desarrollo del metamodelo que especifica a UML.

9 Debido a que el proceso general adoptado contiene elementos del desarrollo de software de código abierto, no siempre se obtiene un seguimiento preciso de los patrones por parte de todos los participantes. Refinamientos sucesivos y estrategias de capacitación se despliegan en el interior del grupo para incrementalmente llegar a este objetivo.

desplazar a la reutilización sino que debe complementarla. El contexto definirá cuál de los dos deberá ser usado. El tiempo transcurrido desde la creación y la cantidad de uso del componente son indicadores a tener en cuenta.

- **Controlar las versiones:** debe mantenerse un repositorio que permita recuperar los estados anteriores de cualquier componente dentro del sistema. El incremento general en la funcionalidad, el refinamiento en el desempeño y la experiencia adquirida al desarrollar el software es información que permanece latente en los repositorios. Los repositorios integrados permiten mantener la sincronización de los grupos de trabajo y blindan el hilo estable — “oficial” — de los hilos secundarios en desarrollo o depuración.
- **Documentar:** ya sea empotrado dentro del código, usando lenguajes de modelado o en artefactos independientes se deben documentar las actividades interesantes que se realicen en el desarrollo del sistema. La documentación debe usar estándares multiplataforma para que pueda ser transparentemente visualizados, editados y compartidos entre los integrantes del equipo de desarrollo.

Proceso de desarrollo de software de código abierto

Es indiscutible el papel preponderante que tiene la planificación en el desarrollo de cualquier tipo de sistema, sin embargo, no debe olvidarse que cuando se requiere solucionar un problema no basta con el mero seguimiento de una receta, es aquel “toque” único que brinda el ser humano el que hace que los sistemas de software se diferencien unos de otros. No es por casualidad que en nuestro medio el software se considera un producto que está cubierto por la misma legislación que las obras literarias o musicales.

En los últimos años, apelando a ese recurso intangible llamado pasión, que aún hoy solo es característico de los seres humanos, se ha extendido el proceso de desarrollo de software de código abierto. Todos los paradigmas que las grandes empresas de desarrollo de software se han encargado de poner como las *mejores prácticas*, se han reevaluado, trastocado, pisoteado y sin más ni más un gran cisma apareció en el horizonte. Este renacimiento moderno surge como respuesta humana al gran vacío de satisfacción de necesidades que brindaba el software a principios de los años noventa.

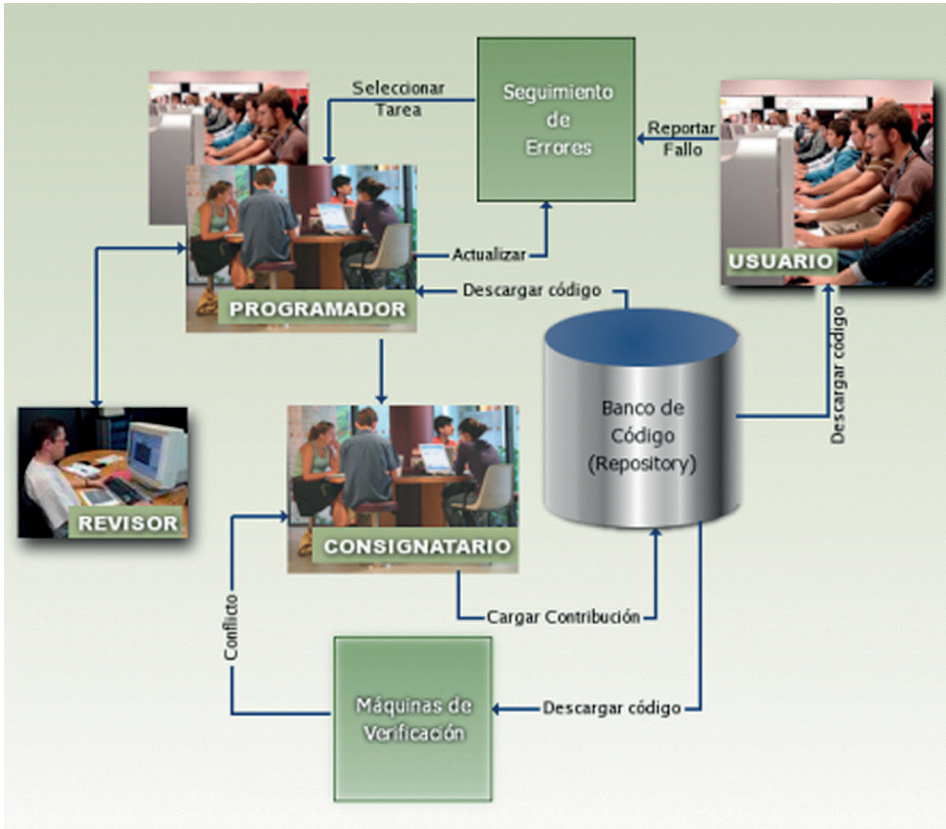


Figura 7. Conceptos básicos en el desarrollo de software de código abierto

Fuente: elaboración propia.

Más que una forma de realizar sistemas, se trata de una visión revolucionaria en torno al software como patrimonio de la humanidad y se une a la filosofía del software libre que expresa Stallman (2002):

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de dichos usuarios:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para esto.

Las aplicaciones más representativas del mundo del software libre, como Apache, Mozilla, MySQL, PostgreSQL y el mismo sistema operativo gnu/Linux, luego de sus etapas primarias adoptaron como proceso de desarrollo uno que contravenía en gran manera los fundamentos del control riguroso y ponía el futuro del sistema en manos de la anarquía¹⁰. Tal como lo expone Linus Torvalds (2001), “la idea es liberar versiones de prueba rápido y a menudo, delegar cuanto sea posible, estar abierto hasta el punto de resultar promiscuo” (p. 227).

Algunos principios fundamentales en este tipo de desarrollo son expuestos por Eric Raymond (2010):

1. Todo buen trabajo de software comienza a partir de las necesidades personales del programador (todo buen trabajo empieza cuando uno tiene que rascarse su propia comezón).
2. Los buenos programadores saben qué escribir. Los mejores, qué reescribir (y reutilizar).
3. “Contemple desecharlo; de todos modos tendrá que hacerlo”.
4. Si tienes la actitud adecuada, encontrarás problemas interesantes.
5. Cuando se pierde el interés en un programa, el último deber es heredarlo a un sucesor competente.
6. Tratar a los usuarios como colaboradores es la forma más apropiada de mejorar el código y la más efectiva de depurarlo.
7. Libere rápido y a menudo y escuche a sus clientes.
8. Dada una base suficiente de desarrolladores asistentes y *beta-testers*, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien.
9. Las estructuras de datos inteligentes y el código burdo funcionan mucho mejor que en el caso inverso.

En general, un proceso de desarrollo de software libre se basa en el hecho de que el programa puede ser usado para cualquier propósito y el código fuente está disponible para cualquier persona. Es decir, la ausencia de barreras en cuanto a la limitación en el uso hace que muchas personas interesadas en la funcionalidad que brinda el software lo descarguen y empiecen a utilizarlo. Y así se da inicio al siguiente ciclo:

1. El grupo inicial de programadores mantiene un sitio en la red para obtener retroalimentación de los usuarios, los cuales reportan fallos, disfuncionalidades y solicitan nuevas características.

¹⁰ Definida en su sentido positivo como la situación humana en donde es innecesaria e indeseable la autoridad, lo que conlleva a una sociedad libre basada en el respeto mutuo de sus miembros y la cooperación voluntaria entre individuos.

2. Un desarrollador, que puede ser uno de los usuarios, revisa la lista de reportes y decide trabajar en uno específico; a tal efecto descarga la última versión del código fuente, la modifica y la envía a un revisor para que este convalide la contribución.
3. La contribución se agrega al código fuente y genera una nueva versión del sistema. Esto se realiza sincronizando el código fuente de desarrollo con aquel existente en la bodega de código fuente (*repository*), la cual normalmente es gestionada por un programa para el control de versiones.
4. Si en algún momento dos programadores están realizando modificaciones a la misma porción de código y pretenden sincronizarlas, ocurre un conflicto que deberá ser resuelto siguiendo reglas definidas que habitualmente contemplan el bloqueo de la versión más reciente, el aviso para resolución entre desarrolladores que causan el conflicto o el descarte de las contribuciones.

De esta forma, se va refinando el software siguiendo el ciclo mostrado en la figura 7. El grupo de desarrollo se ve aumentado cuando usuarios expertos empiezan a proponer y realizar cambios directos en el código; cuando uno de ellos demuestra tener el suficiente interés y respeto hacia los intereses del software se le asigna el permiso para escribir directamente en la bodega de código.

La creación de la documentación, así como de los modelos de requerimientos, análisis, diseño y despliegue sigue el mismo proceso.

Métodos ligeros

A principios del milenio, un grupo de experimentados desarrolladores, entre los que se encontraban Kent Beck, Alistair Cockburn, Martin Fowler y Dave Thomas, redactaron un manifiesto en el que consignaban los elementos de mayor importancia dentro del desarrollo de sistemas software:

Nosotros estamos descubriendo mejores formas de desarrollar software dado que lo creamos y ayudamos a otros a realizar esta tarea. Por medio del trabajo de desarrollo hemos encontrado de gran valor elegir:

- Individuos e interacciones sobre procesos y herramientas
- Software ejecutable sobre documentación profusa
- Colaboración del Cliente en el desarrollo sobre contrato de negocios
- Respuesta al cambio sobre ceñirse a un plan.

Mientras que existe valor en los elementos de la derecha nosotros valoramos más los elementos de la izquierda. (Beck, 1999)

Con esto sentaban las bases para el despliegue de nuevos métodos de realizar software agrupados bajo el nombre genérico de “ágiles”¹¹, que se contraponían a los métodos y procesos tradicionalmente rígidos y altamente planificados.

¹¹ Siendo por definición un método caracterizado por ser liviano y ligero.

Koch (2005) expresa claramente las razones de por qué se desarrollan estos métodos y sus principales características. Los métodos ágiles nacen como respuesta del desarrollador puro al ambiente altamente industrializado y burocrático en el cual transcurren la mayoría de los proyectos de desarrollo de software. En estos ambientes es típico el riguroso control que sobre el cumplimiento de cronogramas, planes de trabajo y presupuestos mantienen los denominados *ingenieros de proceso*. El enfoque tradicional se basa en la planificación, con la que se trata de *predecir* desde las primeras etapas todos los pormenores del ciclo de desarrollo.

Debido a que los métodos tradicionales tienen fundamento en la ingeniería civil y mecánica, tratan de mitigar los riesgos poniendo un especial interés a las actividades de modelo, en especial en las etapas de análisis y diseño; en general, relegan a los desarrolladores a etapas de construcción erróneamente consideradas de *cero esfuerzo* intelectual. Los requisitos del software se tratan de fijar desde los inicios del desarrollo, firmándose usualmente un contrato de aceptación de estos por parte del cliente. Los métodos tradicionales siguen los lineamientos de aseguramiento de la calidad por la cual los procesos son eficientemente documentados, controlados, auditados, vigilados y mejorados. Todos esos aspectos hacen que el elemento clave sea el proceso y se relegue a segundo plano el crear productos que en realidad aporten un nuevo valor al cliente.

Para atacar la abrumadora complejidad que añade el proceso al sistema de software, los métodos ágiles proponen cambiar el paradigma *predictivo* —rígido y resistente al cambio— por uno adaptativo que sea flexible y reaccione rápidamente ante cambios inesperados en los requisitos del software. Aquellos que han desarrollado un software de mediana o alta complejidad conocen de primera mano el hecho de que los requisitos no son estáticos; estos cambian, evolucionan, se transforman, ya que en sí mismos no son sino abstracciones de necesidades del mundo real y este no es estático sino que se caracteriza por una fuerte dinámica.

El *cliente también debe ser adaptable*, en el sentido de que la mayoría de las veces los requerimientos del sistema los va descubriendo a medida que interactúa con él. Una de las premisas de los procesos ágiles es mantener un contacto permanente con el cliente e involucrarlo en todas las fases del desarrollo. Con esto se logra que el cliente obtenga un software que realmente cope sus intereses y (el cliente) sea consciente de los costos asociados a su desarrollo.

Así como los requisitos cambian durante el desarrollo, también lo hacen los recursos y el escenario en el cual se desenvuelve el equipo de trabajo. Para atacar esta característica de los sistemas software, se recomienda *aferrarse a un presupuesto global*, pero distribuyéndolo en pequeños presupuestos que solventen las tareas que a corto plazo realizan los involucrados en el desarrollo.

Es claro con lo expuesto hasta aquí que el enfoque es considerar el desarrollo como una *carrera de 100 metros planos* y no como una maratón. En tal sentido, se deben gestar planes a corto plazo cuyo objetivo principal sea generar versiones del sistema que puedan ser probadas, corregidas e incrementalmente adicionadas en funcionalidad. Cada plan transcurre en lo que se denomina una iteración, la cual usualmente no supera el mes de duración. Algunos recomiendan (Beck, 1999) una duración de dos semanas o menos.

Otra característica de los métodos clásicos, y que atacan los métodos ágiles, es aquella en la cual se considera a las personas como recursos intercambiables, mediante la definición de roles con funciones específicas y predictivas. Esto hace que las personas —cuyo comportamiento es poco predecible y no lineal (Cockburn, 1999)— tengan una moral baja y descienda su productividad; en el mejor de los casos trabajan con esfuerzo y, si sus condiciones son excelsas, rápidamente abandonan el grupo, con lo cual se pierde un activo intangible, lo que repercute negativamente en la calidad global del sistema. Para los metodólogos ágiles, el desarrollo se centra en las personas más que en los procesos, considera a cada miembro del grupo como un *ser creativo e irremplazable*, lo que genera un gran cambio en cuanto al método: *no es estático* ni *recetario*. Evoluciona, se recrea, se adapta y se concreta dentro del grupo de trabajo.

Evidentemente, el proceso de desarrollo de software libre maneja los principios promulgados por los métodos ágiles, los cuales encuentran quizás su máxima expresión en la programación extrema (Beck, 1999).

Proceso unificado

Según lo expresa Alhir (2003):

El Proceso Unificado (UP) es un proceso de desarrollo de software basado en componentes dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental [...] que utiliza la especificación UML dada por el Object Management Group (OMG) para preparar los esquemas del sistema. El Proceso Unificado es aplicable a diferentes tipos de sistemas de software, incluyendo proyectos de pequeña y larga escala; proyectos que tengan varios grados de complejidad técnica y administrativa, a través de diferentes dominios de aplicación y culturas organizacionales.

El PU nace de la unificación, en 1995, de la aproximación sugerida por Rational Software Corporation y el proceso orientado a objetos de la empresa Objectory AB. Se puede considerar al Proceso Unificado como un modelo de ciclo de vida del proyecto que incluye contexto, colaboraciones e interacciones. El UP es documentado totalmente en el libro “The Unified Software Development Process” escrito por Booch, Rumbaugh y Jacobson, y publicado por Addison- Wesley en 1999. (p. 40)

Un sistema, desde que nace hasta que muere, repite el proceso unificado en ciclos de desarrollo constituidos por fases secuenciales cuyo objetivo es la producción incremental de liberaciones del sistema, llamadas comúnmente generaciones del sistema. Cada una de las fases se convierte en un hito principal y está constituida por pequeños microprocesos denominados *iteraciones*. Habitualmente, la numeración de las fases se hace de acuerdo con números enteros, mientras que en las iteraciones se hace en números decimales.

Las fases para el desarrollo de proyectos en el proceso unificado son cuatro, a saber:

- Fase de concepción

También conocida como de inicio. Se centra en el establecimiento de las fronteras, ámbitos, riesgos asociados y visión del proyecto. Determina la viabilidad y los objetivos del proyecto. En esta fase podría tenerse una arquitectura general del sistema que esboce los subsistemas más importantes.

- Fase de elaboración

Se enfoca en la determinación de la arquitectura y requisitos del sistema; de esta forma se establece su viabilidad técnica. Durante esta fase se construyen los casos de uso crítico y se obtiene una arquitectura refinada del sistema.

- Fase de construcción

Es en la que se crea la mayor funcionalidad del producto, finaliza con cierta capacidad operativa. Se centra en la construcción del sistema y la arquitectura de este se considera estable.

- Fase de transición

Concluye con la liberación del producto, centrándose en la transición o distribución del sistema a la comunidad o usuario final.

Dentro de las iteraciones, el grupo de trabajo deberá distribuir sus esfuerzos en áreas estratégicas que conduzcan a la mitigación temprana de los riesgos; estas áreas son conocidas dentro del PU como disciplinas (figura 8):

- Disciplina de administración de cambios en la configuración, la cual se centra en la administración de la configuración del sistema y de las peticiones de cambios en esta.
- Disciplina de administración de proyecto.
- Disciplina de ambiente, que se centra en los ambientes de desarrollo del proyecto, incluyendo los procesos y las herramientas.
- Disciplina de modelado del negocio, focalizado en la comprensión del negocio que está siendo automatizado por el sistema, capturando dicho conocimiento en un modelo del negocio.
- Disciplina de requerimientos, necesaria para entender los requerimientos del sistema que automatiza el negocio y captura dichos requerimientos en un modelo de casos de uso.
- Disciplina de diseño y análisis, centrada en analizar los requisitos y diseñar el sistema, capturando tales conocimientos en un modelo de análisis/diseño.
- Disciplina de implementación, para la implementación del sistema basado en el modelo de implementación.
- Disciplina de pruebas, que maneja las pruebas (evaluaciones) del sistema comparándolas con los requerimientos, basándose primordialmente en el modelo de pruebas.

- Disciplina de distribución, encargada de la distribución del sistema basado en el modelo de distribución.

Durante la etapa de concepción, la mayoría del esfuerzo se distribuye a través del modelo del negocio y la disciplina de requerimientos.

En la fase de elaboración, el esfuerzo se distribuye entre las disciplinas de implementación, diseño, análisis y requerimientos. En la etapa de construcción el esfuerzo se distribuye entre las disciplinas de análisis, diseño, implementación y pruebas.

En la fase de transición, el esfuerzo se distribuye a través de las disciplinas de prueba y distribución. Obviamente, las disciplinas de soporte se distribuyen entre todas las cuatro fases. El objetivo general es producir el sistema, por lo tanto, todas las disciplinas nucleares están comprometidas tanto como sea posible para no introducir riesgos en el proceso; esto es, los practicantes son los responsables de determinar cuáles disciplinas comprometer y en qué momento hacerlo.

En este punto es necesario definir varios conceptos:

- El *Riesgo* en el proceso unificado se concibe como un obstáculo para alcanzar el éxito en la ejecución de una actividad, y puede estar determinado por características del negocio, humanas o técnicas.
- La *Iteración* es un paso o rama a través de una ruta hasta cierto destino. Dicho de otra forma, es un movimiento planeado que puede ser evaluado para demostrar un progreso tangible dentro de una actividad o proceso, además, de acuerdo con lo citado por Jacobson (2005): “Una iteración es iterativa en el aspecto de que es un acto repetitivo que propende la mejora continua del trabajo. Aditiva en el caso de que el resultado es siempre superior al alcanzado con un solo trabajo y paralela ya que el trabajo puede ser concurrente dentro de la iteración” (p. 320).

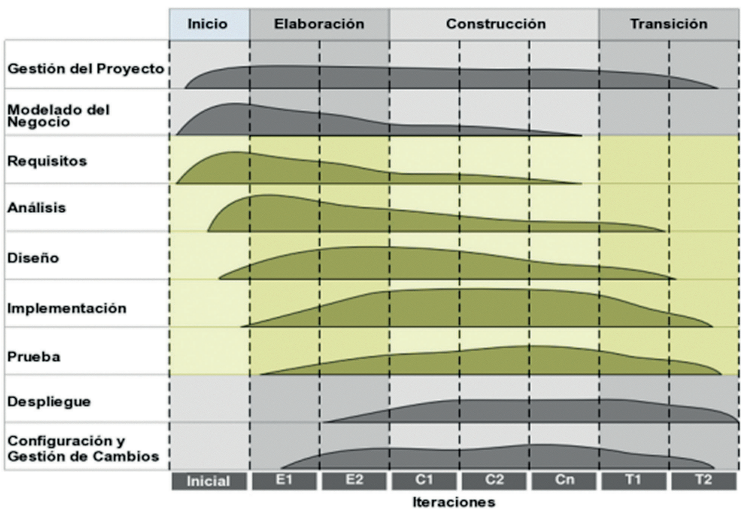


Figura 8. Ciclo de desarrollo de un sistema según el proceso unificado
Fuente: Fundación Eclipse (2018).

Cuando se decantan los requerimientos, aquello que el sistema debe cumplir, se declaran explícitamente los casos de uso, los cuales, dado que el proceso unificado está manejado por ellos, determinan las iteraciones. De la misma forma en que los casos evolucionan en el marco de las disciplinas regidas por un proceso iterativo, los sistemas evolucionan constantemente con base en iteraciones realizados en el marco de su arquitectura, incluyendo en la arquitectura todos los elementos, sus colaboraciones e interacciones.

Resulta pues obvio que la iteración, dado que es un avance demostrable, tiende en últimas a reducir los riesgos inherentes a cada una de las etapas del proceso de desarrollo. Esto ha sido definido por Alhir (2003):

[...] de esta forma las iteraciones confrontan los riesgos derivados de los casos de uso y la arquitectura para alcanzar el éxito en el proyecto, buscando en todo momento reconciliar las fuerza técnicas y del negocio. Una iteración esta acotada en el tiempo con inicio y final fijos en donde una colección de colaboraciones es planeadas, ejecutadas y evaluadas del tal forma que en todo momento se pueda demostrar progreso en el proceso [...] un caso de uso evoluciona a través de un gran número de iteraciones y a través de cualquier número de disciplinas nucleares en una iteración. La experiencia y aprendizaje obtenido en una iteración evidentemente conduce la aplicación de las próximas iteraciones dentro del proceso. (p. 4)

La iteración se convierte en el hito más importante para asegurar el crecimiento continuo y el aseguramiento de la calidad total dentro del sistema que se está desarrollando. Las iteraciones marcan totalmente el ciclo de desarrollo del OpenSITEM que utiliza una aproximación iterativa propuesta por el proceso unificado.

UML: lenguaje de modelado unificado

Con independencia del modelo utilizado para la construcción y gestión del desarrollo del sistema, se requiere que la comunicación entre los diferentes integrantes del grupo de desarrollo sea efectiva. Se hace indispensable que todo el equipo utilice y entienda un lenguaje consistente y unificado, con el cual exprese claramente sus ideas y desde el cual pueda marcar claramente las directrices por seguir. El lenguaje de modelado unificado (UML, por sus siglas en inglés) brinda las características tanto sintácticas como semánticas para lograr caracterizar lógicamente cualquier tipo de software, permitiendo que sea utilizado en cualquier etapa del diseño. Es especialmente útil en aquellos desarrollos enfocados en objetos.

El lenguaje de modelado unificado es definido por el Object Management Group en los siguientes términos¹²:

UML es un lenguaje visual para la especificación, construcción, y documentación de los artefactos de un sistema. Es un lenguaje de modelado de propósito general que puede ser usado con la mayoría de los métodos orientados a objetos y

12 La página del OMG (www.omg.org) describe la organización como un consorcio de la industria de la informática, sin ánimo de lucro, con carácter internacional y de membresía abierta. Los diferentes grupos de trabajo del OMG desarrollan estándares en un rango amplio de tecnologías.

a componentes; que puede ser aplicado a todos los dominios de aplicación (p. e., salud, finanzas, telecomunicaciones, aeroespacial) y plataformas de implementación (p. e., J2EE, .NET). (OMG, 2007, p. 7)

Jacobson (2005) recalca que UML es usado para entender, diseñar, buscar, configurar, mantener y controlar la información acerca de los sistemas.

Con UML se crean artefactos con información acerca de la estructura —o vista estática— y el comportamiento —o vista dinámica— de un sistema. Cada vista del sistema se modela como una colección de objetos que interactúan, es decir, que tienen interfaces y relaciones entre ellos perfectamente definidas. Fruto de tal relación entre objetos, el sistema ofrece una funcionalidad o cumple un objetivo que es de interés.

Portales de información y conocimiento

Antes de la explosión de servicios a través de internet, los portales basados en aplicaciones web estaban recomendados solo para organizaciones que por su complejidad (por tamaño o geografía) necesitaran un sistema tecnológico para que todo su personal pudiese tener acceso a la información en forma compartida y simultánea. Sin embargo, con fundamento en el crecimiento del uso de internet¹³, surge la necesidad de la sociedad de mantener cierto orden en la corriente de bytes, y grupos de usuarios con intereses de información comunes empiezan a conglomerarse alrededor de portales temáticos no organizacionales.

Un portal de información no es, en esencia, una fuente nueva de información; es una vista de la información existente que dispuesta en una forma ordenada se convierte en una herramienta de conocimiento extraordinariamente poderosa, la cual permite poner al descubierto información valiosa que se enmascaraba entre otra no menos interesante (mediante un proceso conocido como *data mining*). Dentro de las múltiples ventajas que ofrece el portal, proporciona la facilidad de obtener información actualizada a muy corto plazo, lo que es indispensable para la óptima toma de decisiones. Dicha información está disponible, en condiciones óptimas, veinticuatro horas al día, trescientos sesenta y cinco días del año, permitiendo así acceder a los datos que se necesitan de acuerdo con la disponibilidad singular de tiempo y apoyar la toma de decisiones en cualquier circunstancia y lugar.

Gran cantidad de organizaciones y grupos de usuarios están explotando el uso de los portales, creando y transformando servicios y procesos tradicionales y convirtiéndolos en servidores de autoservicio, pudiendo así dedicarse a aquellos de mayor valor agregado a la organización, el personal o el grupo de investigación.

13 El porcentaje de la población con acceso a internet en Colombia ha crecido de un 2,1 % en el año 2000 a un 15,8 % en el 2007, según la Comisión de Regulación de Telecomunicaciones.

Beneficios y obstáculos para la implementación de portales basados en aplicaciones web

Las facilidades que proporciona la tecnología permiten que el portal sea accedido a través de numerosas opciones, esto es, mediante computadores de escritorio y portátiles integrados a la red interna de la organización, a través de internet por redes de banda ancha y estrecha y de los diversos medios inalámbricos como son las tecnologías celular, WiFi, WiMax, Bluetooth, por intermedio de PDA, celulares y equipos de cómputo en redes WLAN.

Debido a la estructura del portal, se tiene una fuerte correlación entre diversas aplicaciones que nos permiten analizar interrelaciones que serían realmente complejas y lentas si no se contara con ellos. Sin embargo, es importante recalcar más que los beneficios, los problemas potenciales. De hecho, en el éxito de un portal están enfocados factores clave que tienen beneficios y problemas asociados:

- Factor humano

Los individuos adaptan los procesos de información en diferentes maneras.

- Factor tecnológico

Las intranets pueden ser costosas y poco efectivas si la organización no tiene la tecnología necesaria para construirlas.

La principal ventaja al construir y mantener un portal basado en aplicaciones web es mejorar la eficiencia y efectividad en la comunicación de los miembros de una organización o un grupo de usuarios, lo que aumenta la objetividad en la toma de decisiones y la transferencia de conocimiento. Todo lo anterior se maximiza si el portal se concibe como fruto de un proceso de investigación en donde todos sus componentes y servicios se construyen, mantienen, distribuyen e integran de acuerdo con los requerimientos de los usuarios finales.

Una de las características importantes de los portales es que en un solo lugar —y con un mecanismo de acceso unificado— los usuarios pueden acceder a las aplicaciones. Esta integración con aplicaciones y servicios orientados al trabajo colaborativo hace que trascienda los límites de un mero repositorio organizacional —que permite el autoservicio de requerimientos y extracción de información básica— y lo convierten en una herramienta de administración del conocimiento, útil para la toma de decisiones.

Las novedosas tecnologías que convergen en internet permiten que la información sea personalizada y dirigida, de tal forma que se potencian ciclos de creación, captura y diseminación de conocimiento necesarios para el crecimiento de los activos intangibles de los grupos y organizaciones. Así, los portales convierten la información en valor, ya que eliminan las barreras de distancia y disponibilidad de información, reduciendo costos y conectando a múltiples personas en diversos sitios al mismo tiempo.

La tabla 1 muestra los principales beneficios potenciales de desplegar los portales de información y conocimiento dentro del quehacer de las organizaciones, los grupos de trabajo y las comunidades de práctica.

Tabla 1
Beneficios potenciales al implementar un portal

Beneficios humanos (suaves)
Provee estructura de soporte 24 horas.
Servicio centrado en el usuario
Medio ambiente amigable.
Beneficios físicos y capitales (beneficios fuertes)
Creación de un medio ambiente libre de papel
Mejorar eficiencia y efectividad
Reducción de costos
Menores tiempos en consecución de información
Beneficios estratégicos
Creación de herramientas innovadoras de apoyo
Proveer información a tiempo real
Apoyo a proceso de negocios de reingeniería
Apoyo a los ciclos de creación, captura y diseminación de conocimiento
Aumento de capital intangible
Formalización del <i>know-how</i>

Fuente: elaboración propia.

Los riesgos que se afrontan también son enormes y pueden llevar al traste cualquier política o proyecto de desarrollo. La tabla 2 muestra algunos de los más importantes que deben ser minimizados.

Tabla 2
Riesgos potenciales al implementar un portal

De tipo humano (fuertes)
Indiferencia de la administración
Sobrevaloración del papel de las TIC
Dirección centrada en el capital financiero
Estructuras organizacionales conservadoras
Micropoderes y feudos organizacionales autogestionados
Ignorancia o resistencia respecto al uso de TIC
Resistencia a estandarizar la información
Resistencia a compartir información o conocimiento
Riesgos físicos/capital
Procesos orientados a la técnica y no multidimensionales
Carencia de capital (TIC marginales).
Dificultad de integración de tecnología nueva y la existente
Ausencia de inter, trans y multidisciplinariedad
Riesgos tecnológicos (débiles)
Estándares propietarios

De tipo humano (fuertes)
Redes de interconexión da baja velocidad
Alta relación consumo-adopción de tecnología

Fuente: elaboración propia.

Ciclo de vida de los portales

Los portales como representación sistemática del quehacer de un grupo humano evolucionan en la medida en que dicho grupo mejora su conocimiento de las relaciones entre sus miembros y el entorno que los rodean. En general, pueden determinarse cinco macroetapas que aumentan gradualmente su funcionalidad con base en el conocimiento organizacional y la interrelación de usuarios a través del portal.

Presencia emergente

Esta es la etapa primaria por la que pasa un portal. Su funcionalidad es la de distribuir información interesante para un grupo de usuarios totalmente caracterizado por el grupo de personas que construye —en todas sus dimensiones— el portal. Dichos usuarios no intervienen directamente en la estructura del portal, el cual complementa sus servicios por medio de enlaces y dependencia a otros portales temáticamente relacionados.

Presencia mejorada

Los usuarios pueden determinar en cierto grado la navegación a través de búsquedas en el archivo del sitio. Un portal en esta etapa presentará gran cantidad de información que usualmente se agrupa por áreas temáticas. Los mapas del portal se distribuyen profusamente con el fin de guiar a los usuarios en su tránsito por este y usualmente está disponible un sistema básico de ayuda prediseñada.

Interacción

Los portales registran a sus usuarios. Se implementan herramientas en línea como las salas de charla (chat), las listas de correo y los foros; se realiza capacitación básica por medio de seminarios basados en contenidos y se hace uso extensivo de recursos multimediales. La ayuda es síncrona o asíncrona pero ágil, lo que fomenta una depuración y actualización de la información contenida en el portal.

Transacción

Los usuarios realizan operaciones a través del portal. El comercio electrónico, la gestión de contenidos, la personalización de los ambientes del portal, búsquedas semánticas y el despliegue de servicios avanzados —cursos, blogs, etc.— caracterizan esta etapa.

Transformación

La etapa más avanzada de los portales en la que se han estructurado comunidades de práctica sobre temas concretos que potencian los ciclos de conocimiento mediante las herramientas brindadas por el portal. Se observa una jerarquía *ad hoc* de usuarios con base en su aporte. Ellos mismos generan contenido que es convalidado por la comunidad y los administradores técnicos limitan sus funciones a aquellas relacionadas con mantener operativa la plataforma tecnológica. Las transacciones y el contacto en tiempo real son rutinarios. Las aplicaciones son de conocimiento general y el nivel de inmersión en el portal es alto.

Aplicaciones web

También conocidas como WebApps, son en su concepción más básica aplicaciones que responden a peticiones realizadas por un usuario por medio de un navegador (cliente) y ejecutan la lógica del programa en un servidor. Las aplicaciones web usualmente interactúan con sistemas de bases de datos y distribuyen los resultados de sus operaciones en lenguajes estándar tales como HTML, SMIL, XML, RDF, SVG, etc. Las WebApps también se pueden encontrar en ambientes diferentes al modelo cliente-servidor.

Boss (2004) destaca algunas características de las aplicaciones web:

- No requieren instalación. En general, las aplicaciones web no necesitan ejecutar rutinas de instalación en las máquinas del cliente. Quizás en algunos lenguajes sea necesario la preparación de un ambiente específico de trabajo que la mayoría de las veces es de acceso público.
- Accesibilidad. Las aplicaciones web se despliegan desde de una página web. Los protocolos usados son estandarizados y abstraen fácilmente las capas de aplicación de las de diseño y datos.
- Facilidad en el desarrollo. Los lenguajes usados son de alto nivel, con un buen soporte para cadenas de caracteres, diferentes tipos de datos y con facilidades para la programación orientada a objetos. La mayoría de ellos con sintaxis similares y herramientas de desarrollo gratuitas de fácil adquisición.
- Independencia de la plataforma. Las *WebApps engines* implementan el modelo de capa intermedia, lo que permite que las diferentes WebApps puedan ser desplegadas sobre diferentes plataformas sin detrimento de su funcionalidad. El uso de métodos genéricos definidos en interfaces de programación (API) ayuda en gran manera a garantizar esta característica.
- Seguridad. No obstante la facilidad de acceso de las WebApp, estas pueden implementar rutinas avanzadas que brindan ambientes transaccionales seguros, aislados del sistema de archivos y configuración del sistema en donde se alojan. El intercambio de información cifrada por la red y la integración con la seguridad de los servidores de bases de datos forman un contexto de alta seguridad.

- Privacidad. Las WebApps pueden operar fácilmente sobre una plataforma de preferencias de privacidad, debido a que la mayoría de los motores están habilitados para soportar el protocolo P3P.
- Almacenamiento persistente. Tanto en el cliente —a través de archivos texto para el manejo de sesiones— como en el servidor de base de datos.
- Integración. Las aplicaciones web pueden brindar sus servicios —u obtener uno determinado— a través de interfaces claras y definidas en las denominadas redes de servicios web.

OpenSITEM - Sistema federado de aplicaciones para la caracterización de nodos potenciales de redes de eSalud

OpenSITEM es un sistema federado de aplicaciones de software libre o de código abierto que provee herramientas para analizar datos e información de interés para la descripción y definición de la capacidad de nodos potenciales de redes de eSalud tales como: entidades de salud, servicios médicos, tecnologías de interconexión, operadores de telecomunicaciones, equipos médicos, organizaciones, profesionales, estándares, pacientes, enfermedades, medicamentos y proyectos. Implementa un ambiente para apoyar las tareas de las comunidades de práctica involucradas en la investigación, el diseño, mantenimiento, desarrollo e implementación de redes de eSalud. Tuvo su génesis en la primera fase del Proyecto Telemedicina Bogotá, como solución a la necesidad de administrar los resultados del estudio de campo realizado a las entidades e instituciones de salud y los operadores de telecomunicaciones en la ciudad de Bogotá.

Esta plataforma va generando de manera progresiva un sistema que permita la definición, categorización y caracterización de nodos potenciales de redes de eSalud, para apoyar las actividades básicas de los *trabajadores del conocimiento* en el área de la telemedicina del grupo GITEM¹⁴, ofreciéndoles, además de un repositorio de datos, herramientas que facilitan las tareas de capturar, extraer, organizar, analizar, encontrar, sintetizar, distribuir y compartir información y conocimiento de nodos potenciales. OpenSITEM describe un mecanismo para federar herramientas que sirvan para actualizar el estado de los nodos definidos en el modelo base, así como para la caracterización de nodos y categorías inéditas. El modelo base de nodos y categorías se construye a partir del estudio de campo realizado por el grupo de investigación y se plantea una plataforma para gestionar los datos de nodos potenciales de redes de eSalud, contribuyendo a disminuir el tiempo de adquisición, análisis y despliegue de la información.

14 Conformado por profesionales y estudiantes de la Universidad Distrital así como por profesionales de las diferentes instituciones que han participado en los diferentes estudios de campo.



Figura 9. Imagen en el año 2017 de la página principal de OpenSITEM
Fuente: OpenSITEM [captura de pantalla].

La integración del sistema¹⁵ se basa en métodos conocidos, tal como los descritos por Balduino (2010). Para el diseño de los módulos inéditos se han tenido en consideración varios principios, patrones y antipatrones, tratando de minimizar riesgos que impacten de manera negativa la calidad del software.

OpenSITEM hace uso extensivo de aplicaciones existentes, marcos de trabajo, bibliotecas, APIs, servicios web y plantillas. Esto ha permitido lograr un alto grado de funcionalidad específica. Los módulos propios¹⁶ —aquellos que hacen parte de la suite desarrollada por GITEM— se implementan sobre el *framework* OpenSARA¹⁷.

Módulos funcionales

OpenSITEM, incluyendo las aplicaciones de terceros y las desarrolladas por GITEM, ofrece los módulos base listados en la tabla 3:

15 OpenSITEM es un sistema federado de aplicaciones.

16 En referencia a la autoría, no al carácter de código abierto

17 Diseñado y construido por GITEM. OpenSARA es un producto de este proyecto y se ha usado en otros dominios tanto en la Universidad Distrital (sistema de gestión de inventarios, sistema de consultas a comunidades, sistema de evaluación para acreditación, entre otros), como por algunas empresas del sector TI (OpenKyOS).

Tabla 3
Módulos base de OpenSITEM

Módulo	Elaboración	Nombre
Motor de Federación de Aplicaciones	Propia	OpenSITEM-FE
Motor de recomendación	Propia	OpenSITEM-RS
Gestión de Nodos	Propia	OpenSITEM-NM
Diseñador de Redes	Propia	OpenSITEM-ND
Analizador de Redes	Propia	OpenSITEM-NA
Gestión de Encuestas	Propia	OpenSITEM-PM
Inteligencia de Negocio	Tercero	Knowage
Sistema de Información Geográfica	Tercero/propia	qGIS
Visor geográfico	Tercero	Cesium
Gestión Documental	Tercero	Alfresco
Reportico	Gestión de Reportes	Tercero/Propio

Fuente: elaboración propia.

Características de interés de OpenSITEM

OpenSITEM define algunas características que son de interés en diferentes dominios:

- Dominio de aplicación: desarrolla un motor y un proceso para la federación de aplicaciones.
- Dominio de arquitectura: propone un modelo de arquitectura para la caracterización de nodos potenciales de redes de eSalud.

- Dominio de utilidad: implementa una plataforma para soportar flujos de trabajo de investigadores del GITEM. En el mercado no existe una herramienta que de manera unificada cumpla con el modelo de requerimientos definido.
- Dominio social: tanto el *framework* (OpenSARA) como los módulos de OpenSITEM están disponibles en repositorios públicos y cobijados por licencia de código abierto. El *framework* ya lo han utilizado equipos de trabajo externos al grupo GITEM¹⁸. Se tiene presupuestado que la fase IV del proyecto (transición) permita que los datos también sean abiertos.

Descripción de la arquitectura de OpenSITEM

Como se ha mencionado, OpenSITEM es una aplicación federada de arquitectura orientada a servicios, con características como las descritas por Erl (2017). La integración es realizada por un motor que permite el intercambio de mensajes y la sincronización de sesiones entre los diferentes aplicativos federados, sin llegar a considerar una arquitectura de bus de servicio empresarial (ESB, por sus siglas en inglés)¹⁹.

En esta sección se describe la arquitectura general de OpenSITEM y de los módulos inéditos realizados en el marco del proyecto. Las aplicaciones federadas construidas por terceros se abstraen, y únicamente se visibilizan las interfaces provenientes de las API o de los servicios web utilizados en su adaptación.

La descripción de la arquitectura sistema (AD) se realiza con base en el modelo conceptual definido por la ISO (2011). Dada la extensión de los artefactos, se presentan aquellos apartes de la AD que a juicio de los autores se consideran relevantes para explicar las propiedades fundamentales.

Interesados e intereses

El proyecto considera una aplicación que en su conjunto es inédita y de la cual no se encontró referente de modelo de dominio. En este escenario, es importante la identificación de los interesados, los cuales se enfocan en los investigadores del grupo GITEM que participan en proyectos relacionados con las redes de eSalud. También incluye expertos en el diseño de redes de telecomunicaciones y de prestación de servicios médicos.

A continuación se muestra el resultado del proceso de identificación:

- Asistente de investigación hospital
 - Ingresar y corregir la información recopilada en el estudio de campo.
 - Elaborar informes a partir de la información ingresada utilizando diferentes filtros y formatos.

18 Especialmente en la Universidad Distrital en el sistema de gestión de votaciones, sistema de autoevaluación institucional, sistema de gestión de inventarios, sistema de pago en línea, sistema de costos, entre otros.

19 Aunque existen ESB de corte empresarial tales como Mule, WSO2, Apache Service Mix o similares, el análisis de utilidad mostró que la mayoría de las funcionalidades que ofrecen nunca serían utilizadas y entonces se optó por la simplicidad arquitectónica y evitar la tarea recurrente de configuración y administración del ESB. Véase el apéndice I.

- Generar una plantilla del informe de investigación que cumpla con los parámetros exigidos por la Universidad.
- Registrar las evidencias de las visitas de campo y demás entregables que exige el proyecto.
- Gestionar el proyecto de caracterización de los hospitales.
- Visualizar geográficamente algunos atributos del hospital.
- Analista en proyectos eSalud del GITEM
 - Analizar la información de diferentes nodos.
 - Evaluar la potencialidad de los nodos para pertenecer a una red de eSalud.
- Diseñador de redes de eSalud
 - Diseñar redes de eSalud a partir de la interconexión de nodos.
 - Evaluar la potencialidad de las redes diseñadas.
- Experto en nodo
 - Ingresar y corregir información relacionada con el nodo.
 - Elaborar informes a partir de la información ingresada utilizando diferentes filtros y formatos.

Los principales mecanismos utilizados para la identificación fueron:

- Entrevistas directas: realizadas en su mayoría durante la primera iteración de la fase de inicio. Se definió una entrevista base (ver apéndice K).
- Reunión presencial: de duración fija (30 minutos) y con agenda previa²⁰.
- Matriz de identificación: donde se consignaron las evaluaciones de los atributos de liderazgo, poder de decisión, nivel de interés en el proyecto, conocimiento, actitud ante el proyecto (oposición, neutralidad, soporte), procedencia (externa o interna - GITEM, Universidad).

Punto de vista estructural

La estructura de OpenSITE se presenta principalmente desde tres puntos de vista:

- Punto de vista de negocio (figura 10): con los grupos de actores, roles generales, servicios de negocio y procesos de negocio.
- Punto de vista de aplicación (figura 11): con los grupos de servicios de aplicación y módulos de aplicación (federadas y nativas²¹).

20 La agenda en las reuniones de identificación fue simple. Se buscaba tener una idea de quiénes eran los interesados representativos de los diferentes roles que existen en el grupo.

21 Una aplicación federada es aquella que mayormente ha sido desarrollada por equipos externos, mientras que una aplicación nativa es la que en más alta proporción fue desarrollada por integrantes de GITEM.

- Punto de vista infraestructura (figura 13): con los grupos de servicio de infraestructura y componente de infraestructura.

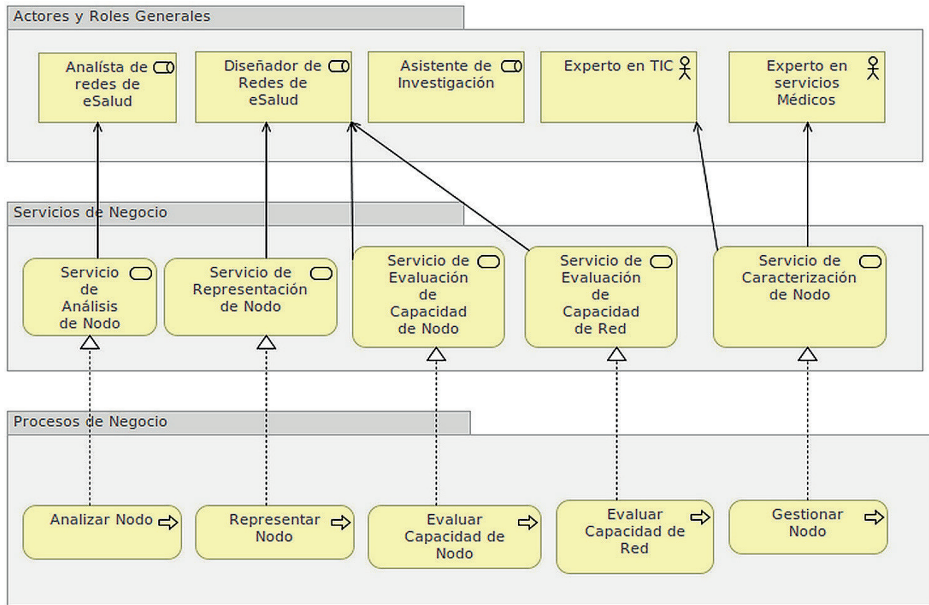


Figura 10. Punto de vista de negocio - vista conceptual por capas
Fuente: elaboración propia.

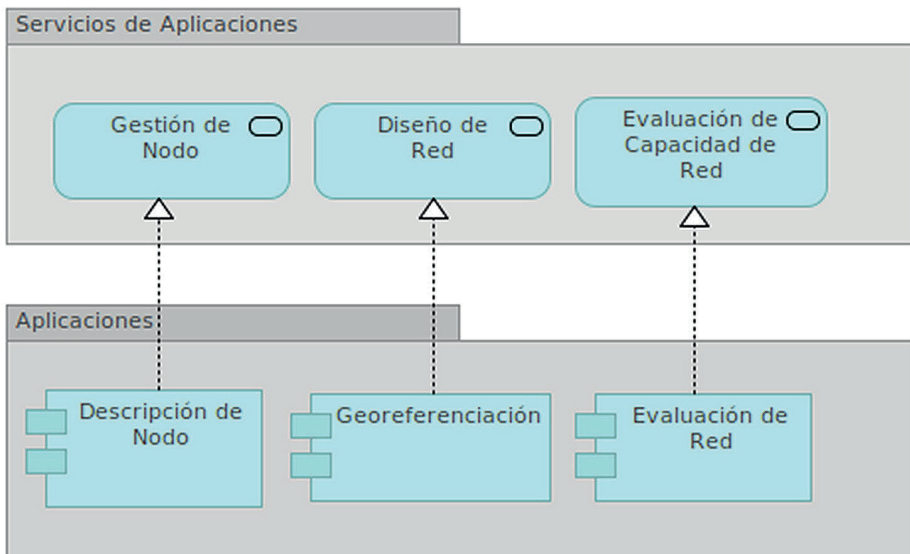


Figura 11. Punto de vista de aplicación - vista conceptual por capas
Fuente: elaboración propia.

Las aplicaciones nativas se denotan por las iniciales de cada módulo. Por ejemplo, el motor de federación (*federation engine*) se conoce como OpenSITEM-FE, el gestor de nodos (*node management*) como OpenSITEM-NM, etc.

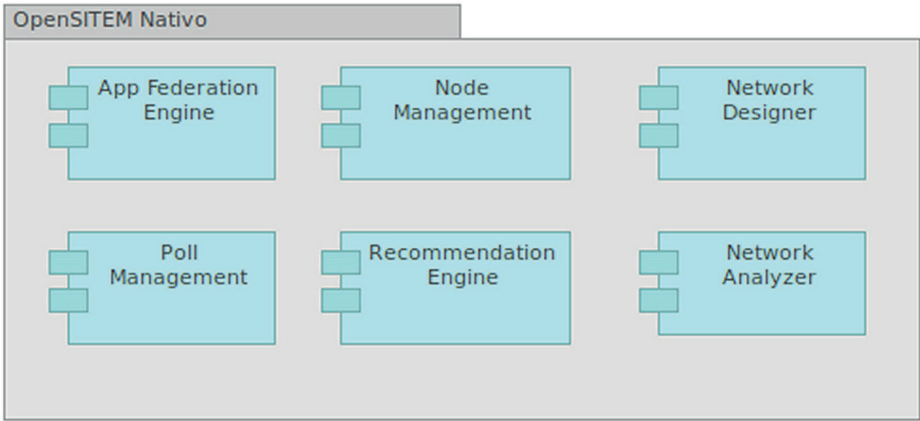


Figura 12. Punto de vista de aplicación - aplicaciones nativas de OpenSITEM
Fuente: elaboración propia.

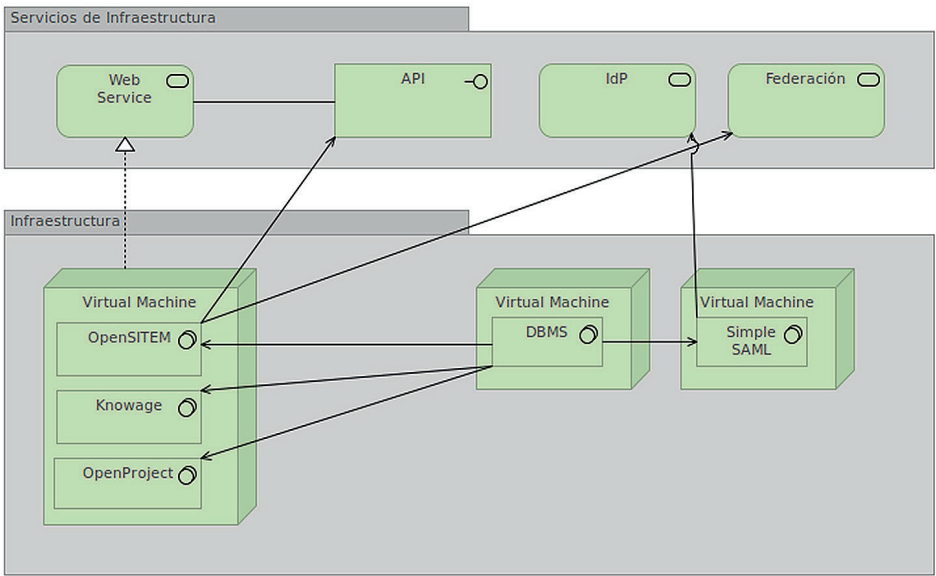


Figura 13. Punto de vista de infraestructura - vista conceptual por capas
Fuente: elaboración propia.

Punto de vista de comportamiento

OpenSITEM-FE: motor de federación de aplicaciones

OpenSITEM no es un sistema de federación universal de aplicaciones. Integra aplicaciones que exponen su funcionalidad a través de interfaz de programación de aplicaciones (API) o de servicios web, utilizando protocolos de la pila TCP/IP con representación basada en XML o JSON. Por esta razón —y por las mostradas en el apéndice I— se prescindió de emplear una solución ESB y se diseñó un motor de integración con la siguiente funcionalidad:

- Evaluación de federación: servicio que se encarga de recibir el mensaje proveniente de una aplicación —a través de un servicio web general— y definir si dicho mensaje hace parte de una transacción que es necesario que sea federada. En el contexto de OpenSITEM, se dice que una transacción es federada cuando luego de su ejecución debe “disparar” transacciones en uno o varios sistemas diferentes.
- Acceso a aplicación: servicio por el cual un usuario adquiere las credenciales para acceder o consumir un servicio específico de una aplicación federada.
- Registro de transacción: servicio específico por aplicación federada que se encarga de consumir un servicio web. Su objetivo es realizar una transacción en una aplicación.

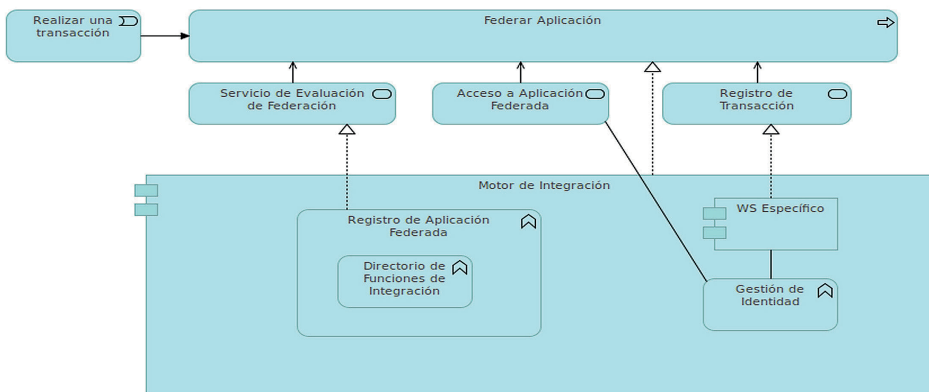


Figura 14. Punto de vista de comportamiento - modelo general del motor de federación de aplicaciones
Fuente: elaboración propia.

Punto de vista de distribución

OpenSITEM y cada una de las aplicaciones federadas están desplegadas en una infraestructura virtualizada. Aunque al momento de elaborar este informe se está trabajando en migrar a contenedores, la tecnología que se emplea está 100 % relacionada con máquinas virtuales.

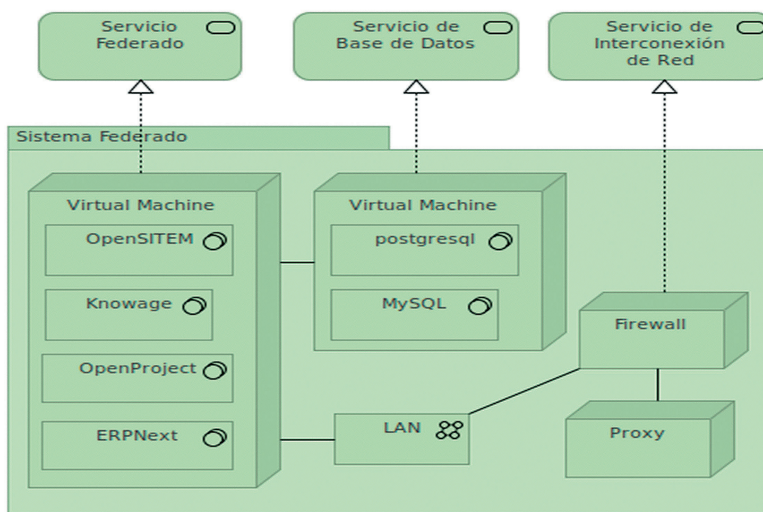


Figura 15. Punto de vista de distribución
Fuente: elaboración propia

En un plano técnico, cada una de las instancias utiliza un sistema operativo GNU/Linux con distribución Centos.

Gestión de nodos

El módulo de gestión de nodos se puede considerar como un directorio enriquecido de los objetos con potencialidad de pertenecer a una red de eSalud. Este directorio se considera el nivel ontológico de mayor abstracción y constituye lo que denominamos el *Catálogo de Nodos*. A nivel arquitectónico, se trata de una clásica aplicación para crear, leer²², actualizar y borrar²³. Los formularios y reportes utilizan plantillas XML, lo que permite hacer cambios sin tener que modificar directamente el código fuente.

Motor de recomendación

El motor de recomendación de OpenSITEM utiliza dos aproximaciones para determinar cuál es el nodo *óptimo* en una determinada búsqueda —cuál se muestra primero, o en una determinada estructura de red— cuál *encaja*. El motor es un módulo de análisis de contenido con un algoritmo que evalúa la similitud que tienen los atributos del nodo respecto al filtro empleado, y un segundo algoritmo que revisa lo *apropiadas* que han sido las clasificaciones anteriores a partir del registro de selecciones, si se trata de búsquedas, o de pertenencia en el caso de estructuras de red.

²² Gestionado por el motor de reportes, el cual es una biblioteca externa basada en la aplicación Reportico.

²³ En realidad en OpenSITEM nunca se borra ni actualiza nada, solamente se replican los registros y se marca su estado. Esto permite tener datos que han demostrado ser útiles en tareas de depuración, control y minería de datos.

Ambos algoritmos están dotados de estrategias de variación de entropía con el objetivo de incrementar las probabilidades de aprendizaje.

Modelo de nodos

Como se presentó anteriormente, el motor de gestión de nodos permite la definición de cualquier tipo de nodo. No obstante, el equipo de trabajo ha definido un modelo base de nodos que se consideran el conjunto reducido que permite describir los elementos primordiales²⁴ de las redes de eSalud.

Los tipos de nodos del modelo base son:

Prestador de servicio de salud

Esta jerarquía de clases (figura 16) se caracterizó a partir de los datos recopilados en el estudio de campo realizado por el grupo GITEM, en el proyecto del Sistema de Gestión de Salud para el Distrito Capital fases I y II. Sus atributos están compuestos por objetos de varias clases: especialidad médica, red de comunicaciones, red eléctrica y red de atención.

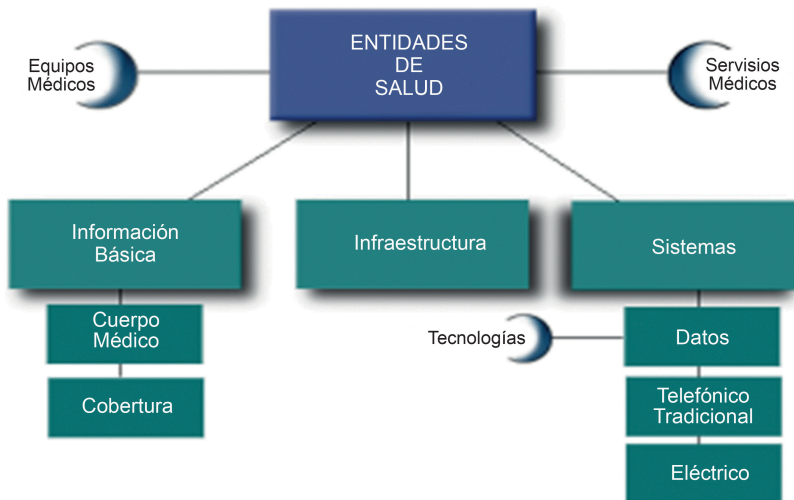


Figura 16. Arquitectura básica de prestador de servicio de salud

Fuente: elaboración propia.

La información de estos nodos puede ser administrada por las entidades prestadoras de servicios de salud, de tal forma que se cree gradualmente un catálogo del estado actual de las entidades y su potencialidad para ser parte en redes de eSalud. Las entidades iniciales pertenecen a la red de la Secretaría de Salud de Bogotá, pero por medio del módulo denominado *redes de atención* se puede crear cualquier prototipo de red jerárquica de atención (Yellowlees, 2005).

²⁴ En este caso el carácter de primordial fue definido por el grupo de interesados del GITEM.

Tecnología de interconexión

Nodos con datos de las tecnologías y protocolos de interconexión disponibles en las redes de acceso y transporte. Estas tecnologías se ordenan principalmente sobre el modelo de referencia OSI, pudiéndose crear —desde el módulo de *Arquitectura*— cualquier otro tipo de modelo. En la actualidad se tiene como alternativa de clasificación el modelo TCP/IP. El conjunto de nodos de esta clase será una guía técnica que muestra la información de las capas físicas, de enlace y de red en formatos unificados²⁵.

Los nodos de tipo tecnología de interconexión proveen información a los analistas para la revisión sistemática de las opciones que brindan los fabricantes y así proyectar redes que sean técnica y económicamente viables²⁶. La información se estructura de acuerdo con indicadores cualitativos y cuantitativos que permiten evidenciar el carácter de interoperabilidad, impacto y permanencia de la tecnología en el mercado.

Equipo

Nodos con datos técnicos de equipos con potencial uso en la telesalud —puede ser desde un equipo de resonancia magnética hasta un tablero interactivo—. Posee atributos que son objetos de las clases proveedor y fabricante, así como atributos de especificaciones técnicas, funcionales y física.

Operador de telecomunicaciones



Figura 17. Arquitectura básica de operador de telecomunicaciones
Fuente: elaboración propia.

25 Aunque existen portales con esta información, por ejemplo tcpipguide, se requieren datos estructurados que faciliten la búsqueda y comparación de tecnologías.

26 En la construcción del modelo de requisitos se encontraron varios casos en los cuales no podía accederse de manera remota a un equipo, porque sus interfaces eran incompatibles con las tecnologías de las instituciones.

Describe nodos de operadores de telecomunicaciones (figura 17) con énfasis en las características técnicas de los servicios que ofrecen, su cobertura y tarifas. Estos nodos brindan a los analistas información de operadores que permite determinar comparativas y evaluar las ventajas y desventajas entre diferentes opciones de interconexión. Este módulo se complementa con la información de dominio público que muestra el Sistema de Información Unificado para el sector de las Telecomunicaciones mantenido por la Comisión Reguladora de Telecomunicaciones.

Organización

Describe las características de organizaciones y grupos de investigación que trabajen en el área de la eSalud. Estos nodos son claves para vislumbrar potencialidades de trabajo en grupo y de sistematizar experiencias adquiridas en los diferentes proyectos desarrollados en el área.

Servicios médico

Este tipo de nodo, cuya arquitectura se muestra en la figura 18, describe servicios médicos. Sus instancias componen un catálogo de servicios y especialidades médicas disponibles. Se pone especial atención en la descripción detallada de los requerimientos técnicos y tecnológicos que requiere cada especialidad así como el perfil de los profesionales y entidades educativas de formación de especialistas.

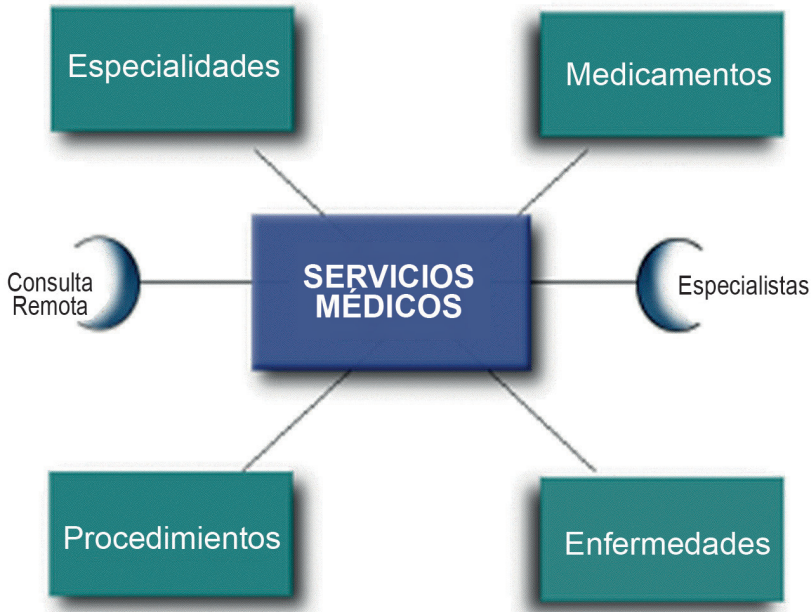


Figura 18. Arquitectura de servicio médico
Fuente: elaboración propia.

Aplicaciones federadas

OpenSITEM integra a su arquitectura soluciones de software libre o de código abierto para proveer un ambiente integrado y aumentar sus prestaciones o implementar nuevos casos de uso. Entre las aplicaciones, mostradas en la figura 19, que contribuyen a OpenSITEM se pueden citar:

- Knowage
Suite de inteligencia analítica y de negocios (Engineering, 2017).
- OpenProject
Sistema de gestión de proyectos (OpenProject, 2017).
- Alfresco
Sistema de gestión documental.
- Cesium
Visor geográfico 3D.
- WordPress
Sistema de gestión de contenido.
- SimpleSAML
Proveedor de identidad.
- Moodle
Es un ambiente integrado de aplicaciones para la creación, organización, mantenimiento y seguimiento de cursos en línea. El fin primordial de sus herramientas es dar soporte a un marco de educación social constructivista.
- MapServer
Es una aplicación desarrollada en Python que proporciona al OpenSITEM los servicios básicos necesarios para la georreferenciación de la información.
- Sistema de Información Unificado del Sector de Telecomunicaciones
Conocido como SIUST, es un aplicativo web desarrollado por la Comisión Reguladora de Telecomunicaciones que contiene información del sector de las telecomunicaciones en Colombia: “[...] información técnica de infraestructura, normatividad del sector, estadísticas comerciales e índices financieros de los prestadores de servicios y los indicadores de gestión del sector, entre otros”²⁷.

27 Tomado del sitio web del SIUST, <http://www.siuist.gov.co/siuist/>

La información, que es de acceso público, permite que OpenSITEM se nutra de ella para complementar y validar sus propias bases de datos en algunos subsistemas.

- Wikipedia

Quizás la fuente de información colaborativa más grande en internet, debido a que sus contenidos son de uso libre, OpenSITEM se nutre de ellos y a su vez los complementa. A partir de información disponible se han editado más de cincuenta artículos en Wikipedia que tienen relación temática con OpenSITEM.



Figura 19. Aplicaciones de software libre o uso libre que complementan a OpenSITEM
Fuente: elaboración propia

Marco de desarrollo OpenSARA

OpenSARA es un marco de desarrollo experimental para aplicaciones PHP orientadas a la web. Se creó con el objetivo de simplificar el desarrollo de OpenSITEM con una orientación hacia programadores. En ese sentido, está pensado para desarrollo

ágil de módulos inéditos y el reciclaje de módulos preexistentes. El código base se distribuye bajo licencia Eclipse Public License o GPL. Los diferentes plugins se distribuyen de conformidad a la licencia definida por sus autores.

OpenSARA es un marco de trabajo basado en el patrón frontera-control-entidad. La conexión a bases de datos se abstrae por medio de un componente que implementa el patrón *Abstract Factory*.

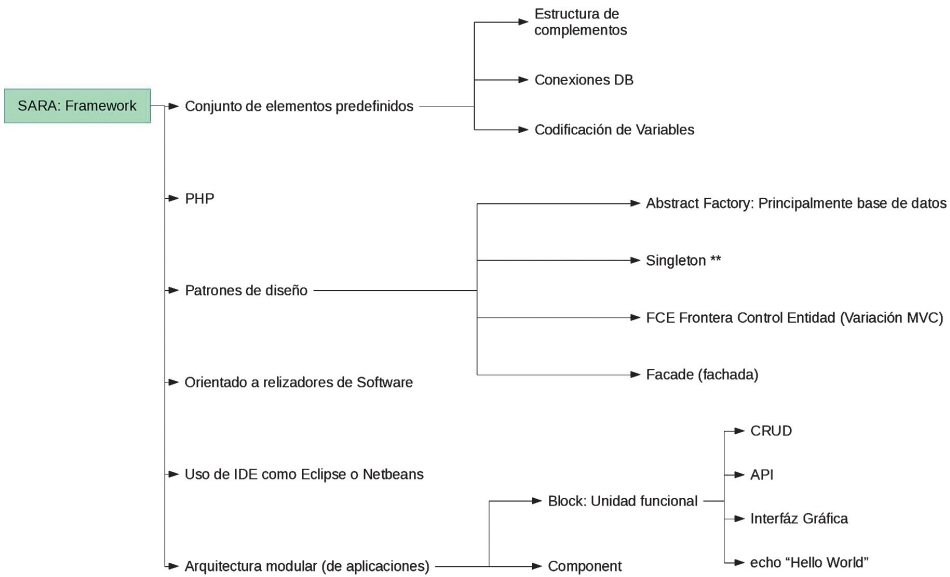


Figura 20. Conceptos principales de OpenSARA
Fuente: elaboración propia.

Patrón frontera-control-entidad

Es una variación del patrón modelo-vista-controlador en donde se hace un mayor énfasis en el bajo acoplamiento.

- Frontera: objetos que actúan como interfaces del aplicativo con sistemas cliente tales como usuario humanos, servicios web, aplicaciones heredadas (*legacy applications*), interfaces de programación de aplicaciones, etc.
- Control: objetos de interoperación entre los objetos frontera y los objetos entidad. Su misión es orquestar la ejecución de la aplicación y la interacción con los sistemas cliente.
- Entidades: objetos que implementan la lógica de negocio, incluyen aquellos que representan el modelo de datos de las aplicaciones.

Los objetos frontera, control y entidad tienen símbolos específicos en el estándar UML.

Marco de trabajo

Bloque

Elemento modular base de OpenSARA. Puede implementar todo un caso de uso empleando patrón FCE o estar constituido por un simple archivo HTML o PHP. Depende del programador decidir las características del bloque de conformidad con los requisitos del proyecto.

OpenSARA recomienda tres modelos de bloques: CRUD completo síncrono, CRUD completo asíncrono (vía AJAX) y servicio web.

Página

Punto de interacción con el usuario. Son generadas en tiempo de ejecución con base en la estructura de bloques declarada a partir de datos o archivos XML. Una página es creada a partir del código fuente de un conjunto de bloques. Es la interfaz gráfica de la aplicación.

Clase nuclear

Conjunto de clases que gestionan la integración de aplicaciones, se dividen en siete categorías que se encargan de:

- Builder: construir las páginas enlazando los diferentes bloques que las constituyen.
- Connection: conectar a bases de datos.
- Crypto: codificar y decodificar cadenas de petición.
- Auth: autenticación, autorización y control de sesiones.
- Manager: gestionar el marco de trabajo.
- Locale: internacionalización.
- General: implementar comportamiento genérico requerido por otras clases nucleares.

Componentes

Bloque especializado. Un componente se construye con el objetivo de que sea consumido por los bloques cuando se requiere un comportamiento genérico (enviar correos electrónicos, generar notificaciones, gestionar logs, etc.). Los componentes definen un contrato a través de las interfaces que declaran.

Estructura

Bloque

Se ubican dentro de la carpeta blocks y se registran en la base de datos o en archivos XML, dependiendo de la configuración del marco de trabajo. Un bloque es una carpeta que debe tener como mínimo un archivo llamado bloque.php, el cual se utiliza como punto de entrada (*entry point*). Aunque esta es la única restricción que impone

openSARA para la estructura, el marco de desarrollo provee dos modelos de bloques enriquecidos que implementan el patrón FCE. El primero de ellos, llamado bloque-Modelo1, está compuesto por:

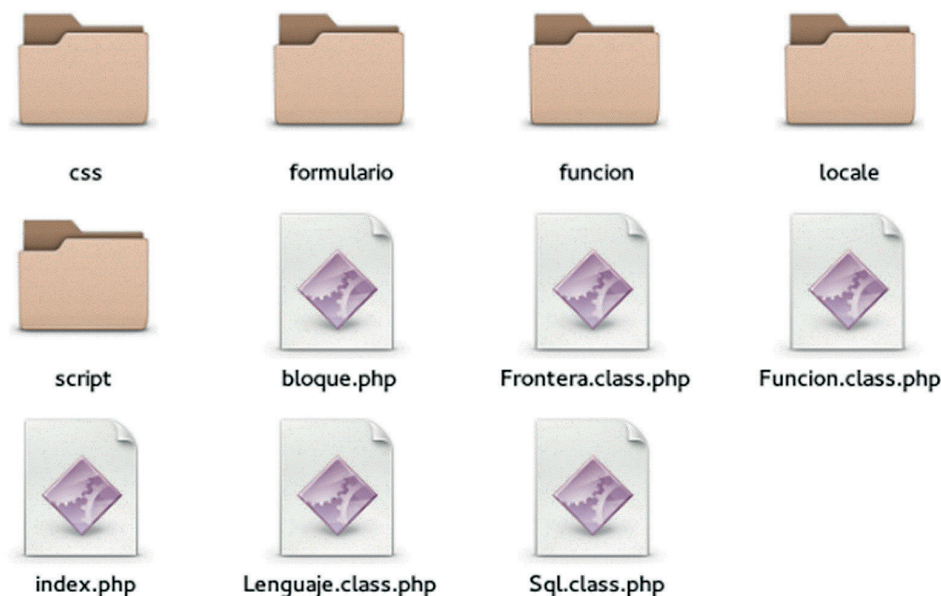


Figura 21. Estructura de un bloque en el sistema de archivos

Fuente: elaboración propia.

- **Bloque.php:** clase bloque. Crea los objetos que procesarán las peticiones. Es el elemento de control en el patrón FCE.
- **Frontera.class.php:** clase frontera. Define la gestión de la interfaz de usuario a partir de los formularios especificados dentro de la carpeta formulario. Implementa el elemento frontera en el patrón FCE.
- **Funcion.class.php:** clase función. Define el procesamiento de los formularios a partir de la gestión de objetos creados con base en las clases de la carpeta función. Implementa el elemento entidad en un patrón FCE.
- **Lenguaje.class.php:** clase lenguaje. Define las funciones de internacionalización específicas para el bloque, describiendo el procesamiento de los archivos de la carpeta locale.
- **Sql.class.php:** clase SQL. Define las cadenas SQL que serán utilizadas en el bloque.
- **Carpeta script:** contiene los archivos js que se utilizarán en el bloque.

- Carpeta css: archivos CSS que definen los estilos que serán aplicados a los elementos del bloque.

Experiencia de desarrollo de OpenSITEM

Como se ha mencionado, OpenSITEM se centra en la arquitectura y se guía por casos de uso. Tanto el director de proyecto como los equipos de trabajo temporales están comprometidos con la tarea de generar un aplicativo funcional que esté alineado de manera emergente con las vistas arquitectónicas planteadas.

Con el método definido, los artefactos se construyen de manera iterativa e incremental. El “orden” en el que aquí se muestran no implica una secuencia de actividades, pues no en pocas ocasiones abordar la disciplina de modelado de dominio se realiza a continuación de, o en paralelo a, un taller de requisitos, y dicho taller surge de una nueva restricción detectada al momento de despliegue. Esta capacidad de adaptación al cambio es lo que ha permitido que OpenSITEM vaya realizando la transición desde un aplicativo específico a una *suite* escalable potenciada por la arquitectura de federación de aplicaciones, orientada a servicios que está implementando.

En la implementación de cada uno de los módulos de OpenSITEM se siguen las fases contempladas en el proceso unificado²⁸ y se desarrollan flujos de trabajo en las disciplinas básicas de:

- Requisitos
- Arquitectura
- Elaboración
- Pruebas
- Despliegue

A partir de iteraciones continuas por las diferentes disciplinas, se refinan constantemente los modelos y se presentan resultados que permiten medir los avances, así como comprobar los niveles de calidad y la validez tanto de los entregables como del método empleado.

Modelo de requisitos

Un requisito es una declaración explícita y aceptada de la funcionalidad que un usuario espera del sistema. Dicha declaración debe ser coherente con el objeto de estudio de OpenSITEM, respetando siempre los principios normativos y de propiedad intelectual. Cada requisito se plasma en un documento denominado caso de uso, el cual describe un conjunto de escenarios de interacción del usuario con el sistema, con el propósito de obtener algo de valor.

²⁸ Inicio, elaboración, construcción y transición.

Declaración del problema

OpenSITEM es una propuesta de solución a los siguientes problemas:

- La información de los estudios de campo no está estructurada, está dispersa, se ha perdido y es difícil de consultar y consolidar.
- El equipo de diseño de GITEM no cuenta con un mecanismo estándar para caracterizar los componentes de las redes de eSalud.
- No es posible determinar el valor de un nodo dentro de una red de eSalud.
- No existe capacidad de análisis multitemporal para determinar la validez de la categorización de nodos de una red de eSalud.

Responsabilidades del sistema

El sistema debe contribuir a que los investigadores:

- Caractericen nodos con potencialidad de pertenecer a redes de eSalud.
- Creen instancias de nodos a partir de la información de los estudios de campo que desarrolló el grupo.
- Comparen nodos de la misma categoría para definir cuál tiene mayor valor relativo.
- Consulten catálogos especializados en cada categoría de nodos.
- Definan modelos de categorías de nodos.
- Evalúen la potencialidad de un nodo para pertenecer a una red de eSalud.

Alcance

OpenSITEM no implementa todo el modelo de requisitos. Los módulos desarrollados son prototipos funcionales que sirven de guía para el desarrollo y evolución posterior de la solución. Si relacionamos esto con el proceso OpenUP, el grupo en la fase III alcanza el final de la fase elaboración.

Definición de actores

El sistema es manejado por varios tipos de usuarios, cada uno con características específicas. Este tratamiento especial tiene que ver con el mantenimiento de la integridad de la información, la cual solo puede ser gestionada por un grupo selecto de usuarios.

Algunos actores esperados en el OpenSITEM son:

- Administrador
- Consultor
- Especialista médico

- Profesional TIC
- Usuario general.

Cada uno de ellos con las características mostradas en el apéndice A.

Casos de uso

Los requisitos funcionales son declarados con diferente nivel de detalle, empleando casos de uso, diagramas de comportamiento y de interacción. Al caracterizar una interacción por medio de un caso de uso, se espera tener información acerca de:

- Nombre del caso de uso.
- Objetivo que se logra al ejecutarse el caso de uso.
- Código que lo identifique unívocamente dentro del banco de artefactos.
- Actores que intervienen al desarrollarse el caso de uso.
- Casos de uso con los que está relacionado.
- Precondiciones. El estado del sistema que debe asegurarse antes de que el caso de uso inicie. Debido a que es responsabilidad del sistema, no se verifica en el caso de uso.
- Postcondiciones. Las características y estado del sistema una vez se haya terminado el caso de uso.
- Flujo de tareas. Flujo principal y flujos alternativos de las actividades que se suceden para lograr la funcionalidad deseada. Se debe mantener claridad en el modelo, por lo que se recomienda utilizar diferentes artefactos para los flujos alternativos cuando esto lo amerite.

La tabla 4 muestra el flujo principal de un caso de uso de OpenSITEM.

Tabla 4
Caso de uso evaluar capacidad de nodo

Caso de uso	
Nombre	Evaluar capacidad de un nodo
Objetivo	El actor asigna un valor y un diagnóstico a un nodo que da cuenta de la capacidad que tiene dicho nodo de pertenecer a una red de eSalud.
Código interno	UC-GENERAL-XXXX
Actores	Consultor.

Caso de uso	
Precondiciones	El nodo se encuentra registrado en el sistema en estado ACTIVO.
Flujo Básico	<ol style="list-style-type: none">1. El consultor selecciona el nodo que desea evaluar.2. OpenSITEM presenta los datos relacionados con el nodo.3. El consultor revisa el historial de valoraciones del nodo.4. El consultor revisa el valor de los atributos del nodo.5. El consultor valora el nodo.6. El consultor argumenta la valoración a través de un diagnóstico.7. OpenSITEM guarda los datos de la valoración.8. OpenSITEM envía una notificación de la valoración a los interesados.
Postcondiciones	Se agregó una valoración argumentada al sistema.
Casos de uso relacionados	Consultar historial de valoración de nodo.

Fuente: elaboración propia.

El modelo actual tiene 80 casos de uso principales definidos²⁹ y más de 230 flujos alternativos, los más relevantes incluidos en el apéndice A. Con esto se concretan los requerimientos de más alto nivel definidos por el grupo GITEM. Se recalca que el grueso de ellos aún no se ha desarrollado por lo que probablemente el modelo se adapte a medida que se progresa en la construcción.

Arquitectura

Los requisitos son insumos para modelar los elementos del sistema y sus relaciones. El equipo de desarrollo se ha apoyado en diferentes diagramas de estructura y de comportamiento e interacción para describir la estructura y el comportamiento del sistema. El apéndice B corresponde a las vistas de la arquitectura que contienen los elementos más interesantes para diferentes partes del sistema. Como puede observarse en dicho apéndice, se han aplicado varios patrones de diseño³⁰, especialmente se presta atención a mantener los principios de:

²⁹ No se incluyen casos de uso implementados en aplicaciones conexas.

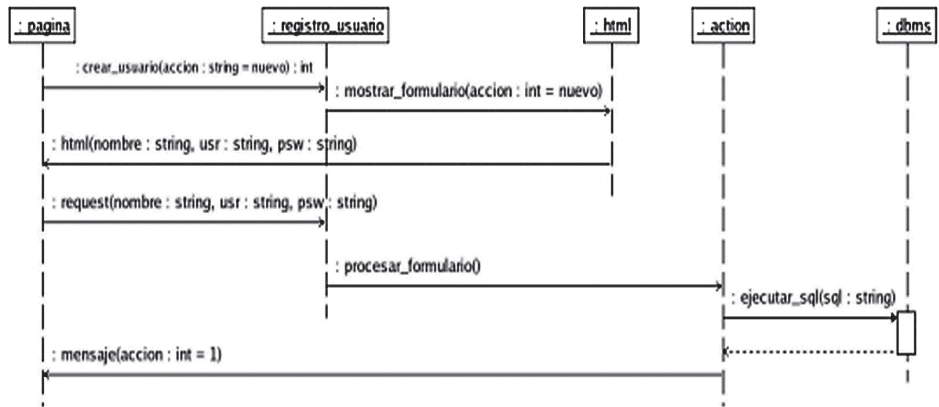
³⁰ Basados en General Responsibility Assignment Software Patterns (GRASP) y GoF.

- Alta cohesión
- Bajo acoplamiento

Y se asignan responsabilidades teniendo en cuenta:

- Experto en información
- Controlador

En la figura 22 se muestra el diagrama de interacción correspondiente a la realización del caso de uso no esencial de registrarse en el sistema.



Proyecto SITEM	Modelo de Diseño Registrar Usuario	Proyectó
Diagrama de Secuencia		Revisó

Figura 22. Realización del caso de uso registrarse en el sistema
Fuente: elaboración propia.

Modelo de implementación

El conjunto de diagramas del apéndice B brinda la información fundamental para el modelo de implementación. En el OpenSITEM se agrupan los diferentes componentes en la jerarquía de carpetas mostrada en la figura 23.

Cada una de las carpetas contiene los ficheros de código fuente del producto:

- Clases: contiene los archivos en PHP que implementan las clases.
- Funciones: grupo de funciones en JavaScript para la validación de información en el nodo de usuario. En la actualidad la fase IV contempla complementar esta aproximación con la utilización de AJAX.

- Configuración: alberga el archivo *config.inc.php* que guarda las variables de ingreso a la base de datos. Dichas variables se encuentran codificadas de acuerdo con el algoritmo que se seleccione (o implemente) desde la clase *codificar*.
- Bloques: agrupa el código fuente de cada bloque desarrollado en el OpenSITEM. Un bloque se define como una unidad de funcionalidad independiente que puede utilizarse en cualquier página.
- Estilo: información acerca de los parámetros generales de estilo —tamaño de fuente, color de bordes, fondos, colores de letras, etc.— para diferentes componentes del OpenSITEM. La modificación o inclusión de parámetros afectará la interfaz global del sistema. Actualmente los estilos en el OpenSITEM se basan en hojas de estilo CSS.
- Gráficos: todos los archivos gráficos usados en el proyecto.
- Documentos: carpeta inicialmente vacía que se utiliza para guardar los archivos que los usuarios carguen a través del protocolo HTTP. Por seguridad se recomienda que esta carpeta se encuentre fuera del directorio donde se encuentra instalada la aplicación.
- Instalar: contiene el instalador del producto. Esta carpeta debería ser retirada una vez el sitio se encuentre en producción.
- Desarrollo: con varios scripts que facilitan la tarea de desarrollo y adaptación de bloques en el sistema. Estos scripts se han construido pensando en plataformas de desarrollo y prueba por lo que se supone no debe encontrarse en plataformas de producción.

En todo caso, durante el proceso de instalación se puede —y recomienda— asociar nuevos nombres a las carpetas, por lo que en teoría ningún desarrollo basado en el OpenSITEM que esté en etapa de producción debería tener nombres de carpetas conocidas.

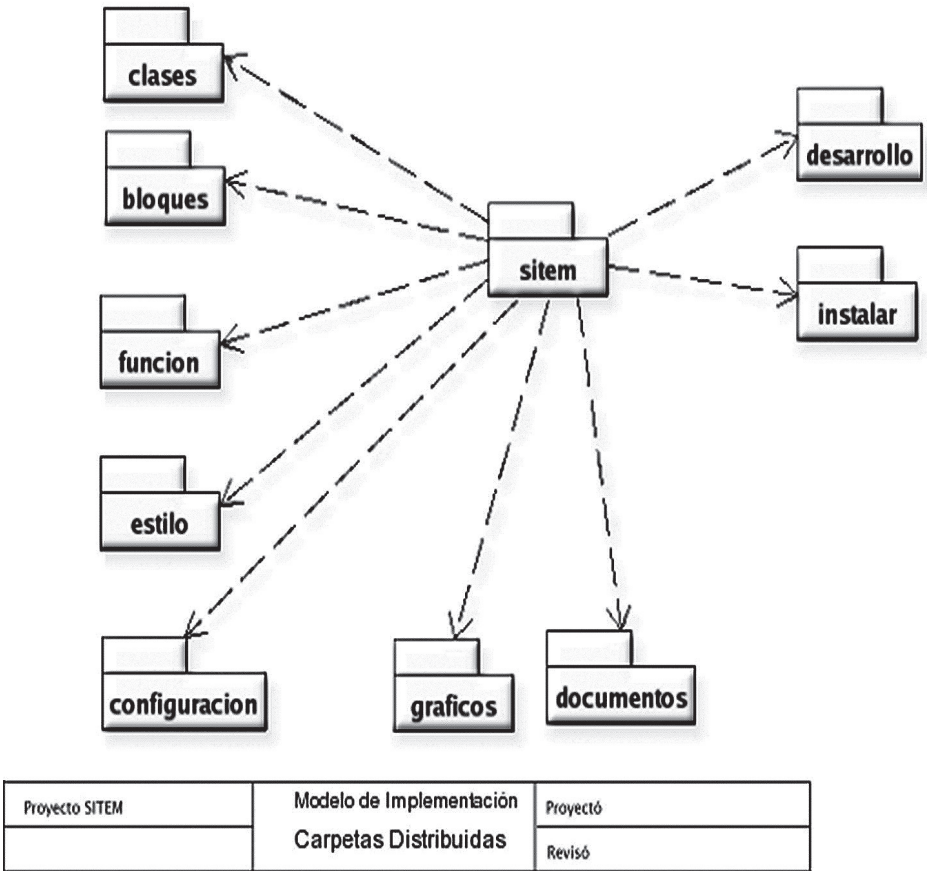


Figura 23. Carpetas que se distribuyen con el OpenSITEM
Fuente: elaboración propia.

Acerca de la seguridad en el sistema

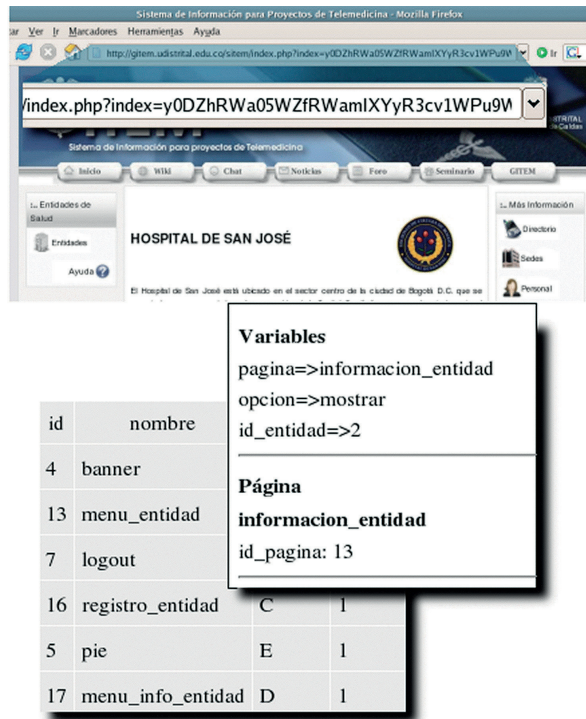
Al tratarse de una aplicación web se presentan grandes retos con la seguridad, se emplea como referencia el top 10 del proyecto OWASP, que para el año 2017 considera que los riesgos más importantes son (OWASP, 2017):

- Inyección
- Fallos en la autenticación
- Exposición de datos sensibles
- Entidades externas XML
- Fallos en el control de acceso
- Fallos en la configuración

- Cross-Site Scripting (XSS)
- De-serialización insegura
- Uso de componentes con vulnerabilidades conocidas
- Gestión y seguimiento ineficiente a los registros del sistema

Para cada uno de estos riesgos el equipo de desarrollo ha estudiado e implementado las técnicas y controles que la misma guía de la OWASP declara. La documentación del modelo de seguridad está siendo elaborada en el marco de una tesis de pregrado, la cual se encuentra en sus primeras etapas.

Como ejemplo, la figura 24 muestra la manera en que OpenSITEM implementa un control sobre A2:2017, usando el gestor de sesiones de lado de servidor para generar un identificador aleatorio para cada una de las peticiones de página. Dicha técnica se asegura de crear cada cierto periodo de tiempo³¹ una llave única que garantiza un alto nivel de entropía en la cadena codificada, lo que implica que cada recurso tendrá un tiempo de expiración.



Variables
pagina=>informacion_entidad
opcion=>mostrar
id_entidad=>2

Página
informacion_entidad
id_pagina: 13

id	nombre		
4	banner		
13	menu_entidad		
7	logout		
16	registro_entidad	C	1
5	pie	E	1
17	menu_info_entidad	D	1

Figura 24. URL encriptada. Con la herramienta Desenlace el desarrollador puede descifrar los datos
Fuente: elaboración propia.

³¹ Predefinido a cinco minutos.

En cuanto a la integridad de los datos, se tiene un modelo de comparación de contenidos que se activa cada periodo de tiempo, el cual es programable, y se propone mantener una copia de respaldo verificada y avalada por el administrador. En caso de corrupción o pérdida de datos, se mantiene una lista completa de los usuarios del sistema de hasta 1'000.000 de sesiones de tipo desplazamiento donde el usuario más antiguo es descartado para la inclusión del nuevo cuando el tamaño asignado es completado, asegurándose el manejo eficiente de disco.

Por ningún motivo se permite el acceso a sitios restringidos a usuarios que no hayan sido plenamente identificados en el sistema. En sitios críticos se hace una revisión de los datos de acceso guardados en *cookies* o se constatan los datos de inicio de sesión. Se ha evitado al máximo el uso de caracter comodín y todos los accesos a la base de datos son validados en su sintaxis. El OpenSITEM actualmente propone el uso de protocolos seguros tales como SSL.

Vale la pena destacar el uso de métodos de autenticación de usuario basado en sesiones y codificación de datos, lo que permite ofrecer un contenido personalizado de acuerdo con el perfil de cada uno de los clientes del sistema.

Modelo de datos

Dentro del proceso de desarrollo del OpenSITEM, el modelado y la elaboración del sistema de bases de datos es una de las partes fundamentales de la propuesta. Se ha venido estructurando el modelo de acuerdo con las necesidades de cada módulo, en particular para garantizar la independencia entre ellos en cada una de las capas, incluyendo la de persistencia.

Se implementan o no políticas transaccionales, dependiendo de las características de cada subsistema. El modelo de seguridad en los datos hereda todos los elementos del servidor tales como bloqueos de puertos, ocultación de ventanas, manejo de sockets, etc.; además, un esquema lógico de validación por conexiones persistentes complementa estas características.

El apéndice D contiene los diagramas de clases que describen la arquitectura de datos del sistema y cada uno de sus subsistemas asociados.

En la actualidad, el OpenSITEM acepta bases de datos PostgreSQL, MySQL y Oracle. La capa de persistencia del hilo principal se despliega sobre un servidor MySQL.

Interfaz gráfica

De acuerdo con los diagramas conceptuales del portal GITEM y de OpenSITEM, se han utilizado para la creación de las páginas los conceptos de diseño web enumerados por Krug (2014), intentando evitar al máximo las páginas sobrecargadas de información. La navegación es guiada mediante enlaces, los cuales están agrupados temáticamente, de manera tal que se logra coherencia en el contenido.

Los gráficos han sido optimizados y su inclusión es necesaria para dar ayuda visual al contenido basado en texto. Teniendo en cuenta que el OpenSITEM está diseñado para interactuar permanentemente y por periodos prolongados con el cliente, se

ha evitado deliberadamente la utilización de componentes dinámicos. Sin embargo, el diseño no pierde atractivo ya que su implementación se fundamenta en bibliotecas de uso extendido.

Para reducir el tiempo de acceso al portal, sobre todo cuando se trabaja con conexiones lentas, se da la posibilidad en algunos subsistemas de descargar en formato PDF todo el contenido del grupo de páginas en donde se esté ubicado.

Arquitectura de la página

Con independencia del subsistema que nos encontremos, las páginas siempre están compuestas por cinco secciones denominadas genéricamente con las letras A, B, C, D y E. En ellas se distribuyen los diferentes bloques que conforman la página en una arquitectura *top-down*. Las páginas que no tienen bloques en todas las secciones colapsan aquellas que no se utilizan para dar una impresión visual consistente. Las figuras 25 y 26 muestran gráficamente el manejo de las secciones en cada página.

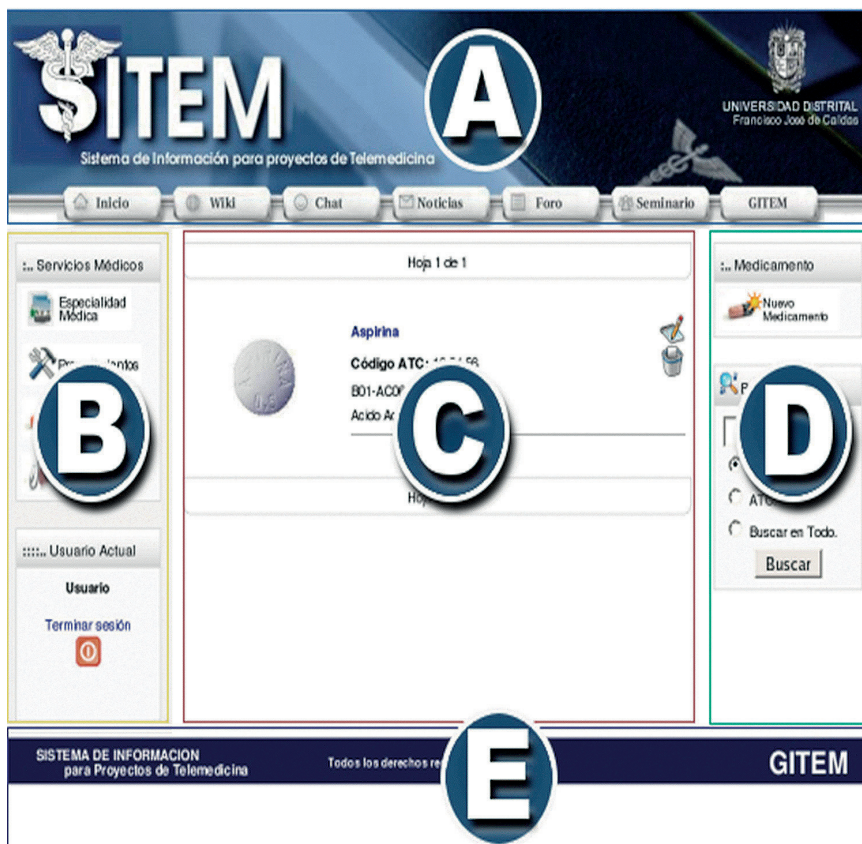


Figura 25. Arquitectura de una página en el OpenSITEM
Fuente: elaboración propia.



Figura 26. Página del OpenSITEM en donde la sección D se ha colapsado

Fuente: elaboración propia.

Entregables del proyecto

Los siguientes artefactos-documentos son generados y utilizados por el proyecto. En el apéndice H se tienen extractos importantes de algunos de ellos.

- **Plan general de trabajo**
- **Modelo de casos de uso.** El cual especifica los requerimientos que debe cumplir el módulo de software y que en últimas constituye los contratos que este, el módulo, tiene con actores externos. Este artefacto estará hecho en su totalidad usando el lenguaje de modelado unificado.

Dentro de este modelo se tienen dos vistas claras: la del negocio y la del sistema. El modelo de casos de uso del negocio ilustra el ámbito del negocio que está siendo modelado. El diagrama contiene actores del negocio y los servicios o funciones que ellos requieren del negocio. El modelo de casos de uso del sistema representa el ámbito de una aplicación. De esto se tiene que un solo modelo de casos de uso del negocio puede tener muchos modelos de casos de uso asociados, donde cada uno representa una única aplicación.

- **Modelo de objetos.** El cual describe la forma en que cada requerimiento o contrato se cumple. Establece las entidades internas, la información que intercambian y los flujos de trabajo que logran el cumplimiento de los requisitos. Los grafos correspondientes podrán incluir diagramas estáticos o interactivos expresados en UML.
- **Glosario.** Que es el único artefacto válido de consulta para la terminología usada en el desarrollo del OpenSITEM (véase el glosario).
- **Visión.** Este documento define la visión del OpenSITEM. Es de todos el que marca las pautas conceptuales.
- **Especificaciones adicionales.** Este documento captura los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto tales como confiabilidad, desempeño, etc., u otros requisitos de ambiente como sistema operativo, requisitos de compatibilidad, etc.
- **Prototipos de interfaces de usuario.** Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Solo los de este último tipo serán entregados al final de la fase de elaboración, los otros serán desechados. Asimismo, este artefacto será desechado en la fase de construcción en la medida que el resultado de las iteraciones vaya desarrollando el producto final.
- **Arquitectura.** Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo con el avance del proyecto (véase el apéndice B).
- **Modelo de datos.** Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un diagrama de clases (donde se utiliza un *profile* UML para modelado de datos a efectos de conseguir la representación de tablas, claves, etc.). El apéndice D contiene los artefactos más importantes del modelo de datos.
- **Modelo de implementación.** Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.
- **Modelo de despliegue.** Este modelo muestra el despliegue de la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes (véase apéndice E).

- **Casos de prueba.** Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba y los resultados esperados. Estos casos de prueba son aplicados como ejercicios de regresión en cada iteración. Cada caso llevará asociado un procedimiento con las instrucciones para efectuarlo y, dependiendo del tipo de prueba, dicho procedimiento podrá ser automatizable mediante un script provisional.
- **Solicitud de cambio.** Los cambios propuestos para los artefactos se formalizan mediante este documento, a través del cual se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación e impacto, y se asegura que estos sean conocidos por el equipo de desarrollo. Los cambios se establecen respecto de la última *baseline* (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso, al final de cada iteración se establecerá una *baseline*.
- **Plan de iteración.** El conjunto de actividades y tareas se ordena temporalmente y se le asignan recursos a corto plazo. Se realiza para cada iteración y en todas las fases.
- **Lista de riesgos.** Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, organizados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación. El apéndice F contiene la declaración de los riesgos más importantes detectados en el proyecto así como estrategias para minimizarlos.

Conclusiones y recomendaciones

La aproximación basada en nodos de las redes de eSalud permite que el grupo de investigación adopte los principios de la orientación a objetos para el diseño de redes de eSalud. La interpretación de los servicios médicos como agregados de nodos ha puesto en evidencia la necesidad de definir las interfaces de cada componente para asegurar su pertenencia a una determinada red. Este enfoque basado en nodos con interfaces definidas ha permitido que el grupo pueda valorar el nivel de capacidad de los nodos y definir cuál de ellos presenta un mayor potencial. Con el desarrollo del proyecto se consolida una descripción de arquitectura para gestionar un modelo de nodos potencialmente útiles en redes de eSalud. También se brinda un conjunto de prototipos que permite al grupo de investigación obtener un ecosistema emergente para la gestión de la información en sus equipos especializados.

La federación de aplicaciones de software libre y de código abierto es una solución óptima para alcanzar funcionalidad invirtiendo bajos recursos. Esto, unido a la decisión de centrar el trabajo en la arquitectura, ha permitido que el desarrollo avance incrementalmente, aunque se tenga un bajo tiempo de retención de personal.

El uso de un marco de desarrollo simple y específico ha demostrado beneficios en comparación con utilizar marcos genéricos. La continuación del uso del marco de trabajo por parte de exintegrantes del equipo en proyectos diferentes indica un adecuado nivel de madurez que potencia la colaboración entre programadores. Se

debe complementar el empleo de un marco de desarrollo con el uso de servicios web y APIs para fomentar la interoperabilidad y adaptabilidad de la solución federada.

La completa inmersión de los interesados en el proceso de desarrollo es clave para garantizar un avance. La gestión de la expectativa es importante, sobre todo en este proyecto en el que la evolución de la descripción de la arquitectura es mucho más rápida que el desarrollo y la puesta en producción de la solución.

La validez de la arquitectura propuesta, así como los módulos prototipo han sido endógenas y se requiere realizar una fase de transición en la cual se pueda alcanzar un mayor grupo de usuarios.

La plataforma que se requiere para desplegar el sistema no es difícil de administrar pues está basada en virtualización. No obstante, aunque en la actualidad grandes proveedores ofrecen infraestructura como servicio (IaaS), por cuestiones de presupuesto y trámites internos no se ha podido explotar esta opción. Por esta razón, la solución aún se encuentra en una intranet.

Desafortunadamente, tareas tan simples como configurar una entrada DNS, crear reglas en un *firewall* u obtener acceso vía SSH se han convertido en verdaderas pesadillas pues el proyecto depende de las unidades especializadas de la institución. Las estrategias de contingencia y recuperación de desastres han demostrado ser efectivas, pues por errores “involuntarios” de los administradores externos de la plataforma, en varias ocasiones ha tocado reconfigurar el sistema a partir de copias de seguridad. En este escenario es claro que la Universidad Distrital Francisco José de Caldas no cuenta con una organización que permita el despliegue de soluciones fruto de investigación en un ambiente de alta disponibilidad.

Bibliografía

- Acero, D. y Ariza, M. (2012). *Requerimientos técnicos y financieros para la implementación de una red piloto de Telemedicina en el Hospital Simón Bolívar*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Alhir, D. (2003). *Understanding the unified process*. Methods and Tools. Recuperado de <http://www.methodsandtools.com/archive/archive.php?id=32>
- Anderson, D. y Carmichael, A. (2016). *Essential Kanban condensed*. Seattle, WA: Lean Kanban University Press.
- Aparicio, L. (2000). *Propuesta de estudio red de telemedicina Bogotá*. Bogotá: Grupo GITEM, Universidad Distrital Francisco José de Caldas.
- Aparicio, L. y Ramírez, J. (2003). *Arquitectura de red de telemedicina*. Bogotá: Centro de Investigaciones y Desarrollo Científico, Universidad Distrital Francisco José de Caldas.
- Ardila, J. y Ardila, M. (2011). *Evaluación y diagnóstico de los servicios básicos y especializados al servicio de la salud de la clínica San Pedro Claver para el desarrollo de la red de Telemedicina de Bogotá*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Balduino, R. (2010). *Introduction to OpenUP (open unified process)*. Fundación Eclipse. Recuperado de <https://www.eclipse.org/>
- Bangeman, C. (1994). *The Bangemann report: Europe and the global information society. Recomendaciones al Consejo Europeo*. Bruselas: Consejo Europeo de Corfú.
- Bashshur, R. (1995). On the definition and evaluation of telemedicine. *Telemedicine Journal*, 1(1), 19-30.
- Bashshur, R. y Lovett J. (1977). Assessment of telemedicine: results of the initial experience. *Aviation, Space and Environmental Medicine*, 48(1), 65-70.
- Beck, K. (1999). *Extreme programming explained: embrace change*. Boston: Addison-Wesley Professional.

- Benafia, A., Mazouzi, S. y Maanru, R. (2016). From linguistic to conceptual: a framework based on a pipeline for building ontologies from texts. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 20(1), 941-960.
- Bergeron, B. (2003). *Essentials of knowledge management*. New Jersey: Jhon Wiley & Sons.
- Boggs, W. y Boogs, M. (2002). *UML with Rational Rose 2002*. Alameda: SyBex.
- Bos, B. (2004). *A proposal for webapps*. W3C Consortium. Recuperado de <https://www.w3.org/People/Bos/webapps-proposal.html>.
- Britannica Concise Encyclopedia. *Anarchism*. Recuperado de <http://search.eb.com/ebc/article?eu=380585>.
- Bruegge, B. y Dutoit, A. (2000). *Object-oriented software engineering*. New Jersey: Prentice Hall.
- Carty, J. (2006). *Dynamics of software development*. Washington: Microsoft Press.
- Cockburn, A. (1999). The Impact of object-orientation on application development. *IBM Systems Journal*, 38, 308-332.
- Congreso de la República. (1982). *Ley 23 de 1982: sobre derechos de autor*.
- Congreso de la República. (1989). *Decreto 1360 de 1989: Inscripción del soporte lógico (software) en el Registro Nacional del Derecho de Autor*.
- Congreso de la República. (1993). *Ley 44 de 1993: Modifica y adiciona la Ley 23 de 1982 y modifica la Ley 29 de 1944*.
- Congreso de la República. (2000). *Ley 565 de 2000: adopción del Tratado de la OMPI sobre Derechos de Autor*.
- Comisión de Regulación de Telecomunicaciones. (2006a). *Condiciones de calidad en servicios de telecomunicaciones*.
- Comisión de Regulación de Telecomunicaciones (2006b). *Indicadores de calidad en servicios de telecomunicaciones*.
- Comisión de Regulación de Telecomunicaciones (2007). *Proyecto de resolución para indicadores de calidad de servicio*.
- Corte Constitucional de Colombia (2003). *Sentencia C-067/03*.
- Craig J. y Patterson, V. (2005). Introduction to the practice of telemedicine. *Journal of Telemedicine and Telecare*, 11(1), 3-9.
- Currie, W. y Seddon, J. (2014). A cross-national analysis of eHealth in the European Union: some policy and research directions. *Information & Management*, 51(6), 783-797.
- Davenport, T. y Prusak, L. (1988). *Working knowledge: how organizations manage what they know*. Boston: Harvard Business School Press.

- Department of Economic and Social Affairs. (2005). *UN Global E-government Readiness Report 2005 From e-government to e-inclusion*. Nueva York: Naciones Unidas.
- Davies, M. (2011). Knowledge - explicit, implicit and tacit: philosophical aspects. En *International Encyclopedia of Social and Behavioral Sciences* (pp. 74-90). San Diego: Elsevier.
- Dueck, G. (2001). Views of knowledge are human views. *IBM Systems Journal*, 40(4), 885-888.
- Duque, J., García J. y Caicedo, D. (2012). *Estudio sobre los requerimientos técnicos y financieros para implementar los servicios telemédicos en el Hospital El Tunal, para el proyecto telemedicina Bogotá*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Fundación Eclipse. *OpenUP Versión 7*. Recuperado de <http://epf.eclipse.org/>
- Earl, T. (2017). *Service-oriented architecture: analysis and design for services and microservices* (2.a ed.). Boston: Prentice Hall.
- Engineering Group. (2017). *Knowage portal*. Recuperado de <https://www.knowage-suite.com/>
- Firestone, J. y McElroy, M. (2011). *Key issues in the new knowledge management. Knowledge and innovation*. Nueva York: Routledge.
- Fundación Eclipse (2018). *Open UP Wiki*. Recuperado de <http://www.eclipse.org/downloads/>
- Gamma, E., Helm, R., Johnson, R. y Vlissides, J. (1994). *Design patterns: elements of reusable object-oriented software*. Boston: Addison-Wesley Professional Computing Series.
- Gang, L. y Yi, L. (2007). *A relational model of knowledge share, knowledge acquisition and product innovation*. Xi'an (China): Universidad Xi'an Jiaotong.
- García, F. (2008). *El cuestionario: recomendaciones metodológicas para el diseño de un cuestionario*. México D. F: Editorial Limusa.
- Girard, J. (2015). Defining knowledge management: toward an applied compendium. *Online Journal of Applied Knowledge Management*, 3(1), 5-20.
- González, O. y Torres, J. (2012). *Evaluación y diagnóstico de los servicios básicos y especializados al servicio de la salud del Hospital de San José para el desarrollo de la red de telemedicina de Bogotá*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Gottesdiener, E. (2002). *Requirements by collaboration: workshops for defining needs*. Boston: Addison-Wesley Professional.
- Guarín, S. García, M. y Torres, L. (2013). *Diagnóstico de las redes eléctrica, telefónica y de datos del hospital Rafael Uribe E.S.E*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.

- Huang, L., Milne, D., Eibe, F. y Witten, E. (2011). *Learning a concept-based document similarity measure*. Hamilton (Nueva Zelanda): Departamento de Ciencias de la Computación, Universidad de Waikato.
- Hurtado, J. (2000). *Metodología de la investigación holística*. Caracas: Sypal.
- IEEE Institute. (1990). *IEEE standard glossary of software engineering terminology - IEEE std 610.12-1990*. Recuperado de <https://ieeexplore.ieee.org/document/159342>.
- International Organization for Standardization. (2011). *ISO 42010:2011 Systems and software engineering - Architecture description*.
- ITU-T (1999). *Telemedicine and developing countries - lessons learned. Document 2/116-E. ITU-D Study Groups. Question 14/2: Fostering the application of telecommunication in health care. Identifying and documenting success factors for implementing telemedicine*. Ginebra: Unión Internacional de Telecomunicaciones.
- ITU-T. (2001a). *Recomendación G.1000 - Calidad de servicio en las comunicaciones: Marco y definiciones*. Ginebra: Unión Internacional de Telecomunicaciones.
- ITU-T. (2001b). *Recomendación G.1010 - Categorías de calidad de servicio para los usuarios de extremo de servicios multimedia*. Ginebra: Unión Internacional de Telecomunicaciones.
- ITU-T. (2004). *Manual Calidad de servicio y calidad de funcionamiento de la red*. Ginebra: Unión Internacional de Telecomunicaciones.
- ITU - OMS. (2012). *National eHealth Strategy Toolkit*. Ginebra: Unión Internacional de Telecomunicaciones.
- Koch, S. (2005). *Free open source software development*. Londres: Idea Group.
- Liebowitz, J. y Beckman, T. (1998). *knowledge organizations: what every manager should know*. Boca Raton: St. Luci Press.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Addison-Wesley.
- Jacobson, I. Booch, G. y Rumbaugh, J. (2005). *The unified modeling language reference manual* (2.a ed.). Boston: Addison-Wesley.
- Jackson, D. (2005). *The W3C workshop on web applications and compound documents*. Cambridge: W3C Consortium.
- Krug, S. (2014). *Don't make me think, revisited: a common-sense approach to web usability*. Nueva York: New Riders.
- Larman, C. (2003). *UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid: Pearson Educación.
- Larman, C. (2004). *Agile and iterative development: a manager's guide*. Boston: Addison Wesley.

- Lewis T., Synowiec, C., Lagomarsino, G. y Schweitzer J. (2012). E-health in low- and middle-income countries: findings from the Center for Health Market Innovations. *Bull World Health Organ*, 90(5), 332–340.
- Martínez, R. y Martín, F. (1998). Landscape: a knowledge-based system for visual landscape assessment. *IEA/AIE*, 2, 849-856.
- Ministerio de Comunicaciones. (2000). *Documento CONPES 3072 - Agenda de conectividad*. Bogotá: autor.
- Ministerio de Salud. (2006). *Resolución 1448 de 2006*. Bogotá: autor.
- Ministerio de Salud. (2015). *Resolución 4678 de 2015*. Bogotá: autor.
- Ministerio de Salud y Protección Social. (2016). *Estudio exploratorio de la situación de la telemedicina en municipios priorizados - Colombia*. Bogotá: autor.
- Muñoz, F. y Barrero, J. (2010). *Evaluación y diagnóstico de los servicios básicos y especializados al servicio de la salud del Hospital La Victoria para el desarrollo de la red de telemedicina de Bogotá*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5, 14-37.
- Nonaka, I. y Takeuchi, H. (1995). *The knowledge creating company: how Japanese companies create the dynamics of innovation*. Nueva York: Oxford University Press.
- Object Management Group. (2007a). *Unified modeling language: specification. Versión 2.1.1*. Object Management Group. Recuperado de <https://www.omg.org/spec/UML/2.1.1>.
- Object Management Group. (2007b). *Unified modeling language: superstructure. Versión 2.1.1*. Object Management Group. Recuperado de <https://www.omg.org/spec/UML>.
- OpenProject (s. f.). *Portal del Sistema OpenProject*. Recuperado de <https://www.openproject.org>
- Open Web Application Security Project. (2017). *OWASP top 10 - 2017 The ten most critical web application security risks*. The OWASP Foundation. Recuperado de https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf
- Organización para la Cooperación y el Desarrollo Económico. (2015). *OECD reviews of health systems: Colombia 2016*. Paris: OECD Publishing,
- Organización Mundial de la Salud. (2010). *Telemedicine: opportunities and developments in member States: report on the second global survey on eHealth 2009*. Ginebra: autor.
- Organización Mundial de la Salud. (2016). *Global diffusion of eHealth: making universal health coverage achievable. Report of the third global survey on eHealth*. Ginebra: autor.

- Organización Panamericana de la Salud. (2011). *Estrategia y plan de acción sobre eSalud. 51 Consejo Directivo*. Washington: autor.
- Observatorio Así Vamos en Salud. (2017). *Tendencias de la Salud en Colombia. Informe Anual 2016*. Bogotá: autor.
- Patton, J y Economy, P. (2014). *User story mapping: discover the whole story*. Sebastopol: O'Reilly Media.
- Pressman, R. (2006). *Ingeniería del software* (6.^a ed.). Ciudad de México: McGraw-Hill.
- Raymond, E. (2001). *The cathedral and the bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly Media.
- Research and technology development on telematics systems in health care: AIM 1993 Annual Technical Report on RTD: Health Care (1993)*. Bruselas: Comisión Europea, Dirección General XIII.
- Roos, I. et al. (2016). *Factors that influence the implementation of e-health: a systematic review of systematic reviews*. Londres: Research Department of Primary Care and Population Health, University College.
- Rozo, O., Valencia, S. y Barahona, F. (2003). *Estudio diagnóstico de las condiciones técnicas y financieras en instrumentos y equipos médicos y de servicios de la clínica San Pedro Claver para la implementación de los servicios telemáticos*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Ruiz, M. y Niño, D. (2012). *Evaluación y diagnóstico de los servicios básicos y especializados al servicio de la salud del Hospital El Tunal para el desarrollo de la red de telemedicina de Bogotá*. Bogotá: Grupo de Investigación en Telemedicina, Universidad Distrital Francisco José de Caldas.
- Salazar, J. y Kopec, A. (2002). *Aplicaciones de telecomunicaciones en salud en la subregión andina - telemedicina*. Lima: Organismo Andino de Salud, OPS.
- Sarmiento, A. (2005). *Issues of human computer interaction*. Londres: IRM Press.
- Sowa, J. (1984). *Conceptual structures: information processing in mind and machine*. Boston: Addison-Wesley.
- Stallman, R. (2002). *Free software, free society: selected essays of Richard M. Stallman*. Boston: GNU Press.
- Tatnall, A. (2005). *Web portals: the new gateways to internet information and services*. Londres: Idea Group.
- The Open Group. (2016). *ArchiMate 3.0.1 specification*- Amersfoort: Van Haren Publishing.
- Torvalds, L. y Diamond, D. (2001). *Just for Fun: The Story of an Accidental Revolutionary*. Nueva York. Harper Collins.
- Turban, E. (1992). *Decision support and expert systems - management support systems*. Sydney: Collier Macmillan.

- Van Heijst, G., Van der Spek, R. y Kruizinga, E. (1997). Corporate memories as a tool for knowledge management. *Expert Systems with Applications*, 13(1), 41-54
- Wielinga, B. et al. (1992). KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition Journal*, 4(1), 5-53.
- Wiig, K. (1993). *Knowledge management foundations*. Arlington: Schema Press.
- Wilson, R. (2017). *Successful digital health systems: guidelines for healthcare leaders and clinicians*. Recuperado de <https://library.e-healthconference.com/>
- World Health Organization. (WHO) (2016). *Integrated care models: an overview*. Working document. Copenhagen: WHO.
- Yellowlees, P. (2005). Successfully developing a telemedicine system. *Journal of Telemedicine and Telecare*, 11(7), 331-335.
- Zavala R. (2000). *Diseño de un sistema de información geográfica sobre internet* (tesis de Maestría en Ciencias de la Computación). Universidad Autónoma Metropolitana-Azcapotzalco. Ciudad de México.
- Zellerino, B., Milligan, S., Brooks, R., Freedenberg, D. y Collingridge, D. (2009). Development, testing, and validation of a patient satisfaction questionnaire for use in the clinical genetics setting. *American Journal of Medical Genetics Part C: Seminars in Medical Genetics*, 151C(3), 191-199.

Glosario

ADSL (Asymmetric Digital Subscriber Line): línea digital asimétrica del suscriptor. Tecnología que permite la transmisión de información digital a altas velocidades de descarga utilizando medios de transmisión convencionales.

Algoritmo: conjunto de reglas o procedimientos que representa la solución a un problema. Un programa de software es la transcripción de un algoritmo.

Ancho de banda: cantidad de datos que pueden transmitirse en determinado periodo de tiempo por un canal de transmisión.

Apache: servidor web de distribución libre, poderoso y flexible. Implementa los últimos protocolos HTTP. Altamente configurable y extensible con módulos de terceros. Fue desarrollado en 1995 y ha llegado a ser el más usado de internet.

API (Application Program Interface): interfaz de programación de aplicaciones.

Aprendizaje por computador: hace referencia al uso de computadores como parte clave en la enseñanza y el aprendizaje haciéndolo parte integrante del entorno educativo.

Aprendizaje en línea: hace referencia a aquel tipo de aprendizaje que se lleva a cabo mediante la utilización de una red o internet.

Artefacto: cualquier tipo de información que es producida o usada en el proceso de desarrollo de software.

Base de datos (motor): aplicación informática que permite la gestión de datos y el manejo de la información.

Call (*Computer-assisted language learning*): es un pseudolenguaje que permite aproximar la enseñanza y el aprendizaje, en el que la tecnología computacional es usada, ya sea como presentación, refuerzo, o acceso a material para aprender. Usualmente incluye elementos interactivos y contenidos multimedia.

Caso de uso: artefacto utilizado para describir la funcionalidad de un sistema desde el punto de vista de los usuarios.

CBR (*Constant Bit Rate*): tasa de bits constante.

CGI (*Common Gateway Interface*): estándar para tender interfaces entre aplicaciones externas y servidores de información, tales como los servidores HTTP.

Cookie: pequeño archivo de texto que se almacena en el disco duro al visitar una página web y que sirve para identificar al usuario cuando se conecta de nuevo a dicha página.

Clase: en el desarrollo de software es la representación mediante un nombre, atributos y operaciones de un objeto o conjunto de objetos.

CRUD, caso CRUD: caso que agrupa los casos de crear, leer, actualizar y borrar registros.

CSS (*Cascading Style Sheet*: hojas en estilo de cascada): reglas de estilo para documentos HTML.

Daemon (*Disk And Execution Monitor*): un programa no invocado explícitamente pero que permanece esperando a que una situación especial ocurra. Los sistemas Unix ejecutan muchos *daemons*, principalmente para manipular solicitudes de servicios desde otro host o desde una red.

DBMS (*Data Base Management System* - sistema de gestión de bases de datos): sistema de software que permite a los usuarios guardar y modificar información.

Depurar: relacionado con el software, detectar, localizar, corregir los problemas en un programa informático.

Desarrollo iterativo: método de construcción de productos cuyo ciclo de vida está compuesto por un conjunto de iteraciones, las cuales tienen como objetivo entregar versiones del software.

Disciplina: conjunto de reglas para mantener el orden y la subordinación entre miembros de un cuerpo.

Dominio: parte del nombre jerárquico con que se conoce a cada entidad conectada a internet. Un dominio se compone de una serie de etiquetas o nombres separados por puntos.

DSL (línea digital de abonado): tecnología de red pública que proporciona ancho de banda amplio sobre el cableado tradicional de cobre en distancias limitadas. Hay cuatro tipos de DSL: ADSL, HDSL, SDSL y VDSL. Todos ellos se aprovisionan a través de pares de módems, estando un módem situado en la oficina central y el otro en el domicilio del abonado.

Entorno de aprendizaje: conjunto de conocimientos, herramientas de enseñanza y aprendizaje, espacios y personas involucradas en un proceso de aprendizaje dado.

Entorno virtual de aprendizaje (*virtual learning environment*): paquete de software cuya función es reemplazar o complementar un entorno de aprendizaje formal y tradicional.

eSalud: de acuerdo con el Instituto de Estándares de Comunicaciones Europeo, es la aplicación de tecnologías de la información y la comunicación en todo el rango de funciones que afecta el sector salud.

Filtro: cualquier función o programa que seleccione información automáticamente con un criterio preestablecido.

GNU – (GNU's Not Unix): sistema operativo, compuesto de pequeñas piezas individuales de software totalmente libre.

Hipertexto: método de preparar y publicar un texto ideal para el computador, con el que los lectores pueden escoger su propia ruta a través del material. La información se descompone en pequeñas unidades y después los hipervínculos se insertan en el texto.

Hipervínculo - enlace: en un sistema de hipertexto, palabra subrayada o destacada que cuando se pulsa en ella lleva a otro documento.

HTML (*Hyper text markup language* - lenguaje de marcado de hipertexto): lenguaje de marcado que define la estructura de páginas web. Utiliza etiquetas para denotar los diferentes objetos que componen una página.

HTTP - (Hyper Text Transfer Protocol - protocolo de transferencia de archivos): estándar para la transferencia de mensajes entre navegadores web utilizando texto plano.

Incremental: que puede instalarse por fases, cada una de ellas añadiendo nuevas funcionalidades.

Infraestructura: conjunto de medios necesarios para el desarrollo de una actividad.

Ingeniería de software: aplicación de un enfoque sistemático al diseño, desarrollo, operación y mantenimiento de software.

Ingeniería inversa: conjunto de etapas desarrolladas para obtener las bases del diseño y la forma de implementación de algún producto de ingeniería a partir de su estado final.

Interconexión: unión o conexión entre sí de dos o más elementos.

Interfaz: parte de un sistema en la que se desarrolla la comunicación o interacción entre el programa y el usuario (siendo este humano u otro sistema).

Interoperabilidad: según la IEEE, es la habilidad de dos o más sistemas para intercambiar mensajes y utilizar la información intercambiada.

IP (*internet Protocol* - protocolo internet): protocolo de la capa de red de la pila TCP / IP que ofrece un servicio de intercambio de paquetes de datos entre host.

Iteración: repetición de un conjunto de pasos.

Kbps: kilobits por segundo.

KBps: kilobytes por segundo.

Método: modo ordenado de proceder para llegar a un resultado o fin determinado.

Metodología: parte de la lógica que estudia los métodos.

Mitigar: moderar, suavizar.

Modelado: descripción, en un lenguaje de máquina, de la forma o características de un objeto o conjunto de objetos.

Módulo: una aplicación software que presenta una funcionalidad específica dentro de OpenSITEM. En el contexto del proyecto es sinónimo de subsistema.

Multiplataforma: las aplicaciones pueden ser vistas en cualquier computador del mundo a través de múltiples plataformas (sistemas operativos, navegadores y hardware).

MySQL: sistema de administración para bases de datos relacionales (rdbms).

NRT-VBR - (*Non-Real-Time Variable Bit Rate*): tasa de bits variable en tiempo no real, esta categoría de servicio se encamina para aplicaciones en tiempo no real, las cuales tienen características de tráfico en ráfaga.

Nodo - componente de una red de eSalud: tiene atributos que definen su tipo y capacidad de interconectarse e interoperar con otros nodos de igual o diferente tipo. Ejemplos de nodos en OpenSITEM: un electrocardiógrafo (tipo equipo médico), un hospital (tipo entidad de salud), un radiólogo (tipo especialista), una toma eléctrica (tipo componente eléctrico), un *smartphone* (tipo dispositivo de comunicación), etc.

NTP (*Network Time Protocol* - protocolo de tiempo de red): sistema de sincronización de tiempo para relojes de computadores a través de internet.

Objeto: un objeto es una instancia, un ejemplar, de una clase. Por extensión del término, en OpenSITEM se considera que un nodo es un ejemplar de una categoría de nodos.

OMG (*Object Management Group*): organización que se encarga de crear y actualizar varios estándares de tecnologías orientadas a objetos incluyendo UML y BPMN.

OSI - *internetworking* de sistemas abiertos: programa internacional de normalización creado por ISO e ITU-T. Para desarrollar estándares para las redes de datos que faciliten la interoperabilidad entre equipos de varios fabricantes.

Passwords - contraseña: método de seguridad empleado para identificar a un usuario. Sirve para dar acceso a personas con determinados permisos. En la actualidad se están generando de manera dinámica y distribuyendo vía correo electrónico o mensajes de texto.

PCR - (*Peak Cell Rate*): tasa pico de celda.

PHP - (*Hypertext Preprocessor*): es un lenguaje interpretado de alto nivel ejecutado al lado del servidor. Según w3techs.com, a octubre de 2017 cerca del 80% del total de sitios web del mundo empleaban este lenguaje.

Plantillas: serie de archivos que definen la apariencia de una aplicación web y permiten cambiar la presentación de la aplicación con solo modificar unos cuantos archivos, sin necesidad de modificar el código que le da funcionalidad a la aplicación.

Plataforma - base o elemento de apoyo: se refiere a una combinación específica de hardware y sistema operativo.

Protocolo: conjunto de instrucciones o lenguaje que utilizan los computadores para comunicarse entre sí.

Prototipo: original ejemplar o primer molde en que se fabrica una figura u otra cosa. Modelo original que sirve como ejemplo para futuros estados o formas.

QoS (*Quality of Service* - calidad de servicio): una medida de rendimiento en el sistema de transmisión que refleja la calidad de la transmisión y la disponibilidad del servicio.

RDSI (red digital de servicios integrados): protocolo de comunicaciones que permite a las redes de las compañías telefónicas transportar datos, voz y otro tráfico.

Registro: pequeña unidad de almacenamiento que contiene cierto tipo de datos. En el ámbito de las bases de datos es cada una de las fichas que forman un conjunto.

RT-VBR (*Real-Time Variable Bit Rate*): está encaminada a aplicaciones que requieren restricción en la variación de retardo (lo mínimo) y variación de retardo, que podrían ser apropiados para voz y video en tiempo real.

Script: pequeños archivos de texto embebidos dentro del documento que ofrecen una forma de extender los documentos HTML, convirtiendo el contenido del documento en una página dinámica; pueden ser ejecutados al presentarse un evento como el movimiento del ratón.

Scorm (*Sharable Content Object Reference Model*): es una colección de normas y especificaciones para aprendizaje basado en la web. Define cómo debe ser la comunicación entre el contenido de lado al cliente y un sistema servidor, tratando aspectos como el empaquetamiento de información en archivos ZIP.

Servidor: dispositivos de un sistema que resuelve las peticiones de información de otros elementos del sistema a los que se denomina clientes.

Sistema de información: conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución.

SQL (*Structured Query Language* - lenguaje estructurado de consultas): lenguaje de programación interactivo y estándar para obtener y actualizar información de una base de datos.

Tasa de bits: velocidad en kilobits por segundo a la que puede transmitir un circuito virtual.

TCP/IP - TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*) - protocolo para el control de la transmisión / protocolo internet: nombre común que se da al paquete de protocolos desarrollados por el DoD en los años setenta con el fin de soportar la construcción de *interworks* a escala mundial.

Tecnología: conjunto de los conocimientos técnicos. Conjunto de los instrumentos y procedimientos industriales de un determinado producto.

Teleasistencia: los sistemas de teleasistencia permiten a los pacientes recoger datos acerca de su enfermedad, transmitir la información a un lugar remoto y usar la videoconferencia para discutir sus padecimientos y tratamientos con el profesional de la salud.

Teleconferencia: es la telecomunicación que se establece entre dos o más especialistas para tratar diversos temas. En el caso de la telemedicina, generalmente tienen que ver con los pacientes o están relacionados con capacitación especializada.

Telemedicina: herramienta tecnológica para brindar servicios médicos a distancia mediante el intercambio de imágenes, voz, datos y video, haciendo uso de las tecnologías de la información y de las comunicaciones, permitiendo el diagnóstico, la opinión de casos clínicos y la provisión de cuidados de salud y educación médica.

Transición: tiempo en el cual se cambia de un estado a otro.

UBR (*Unspecified Bit Rate*): esta categoría está encaminada para aplicaciones en tiempo no real, por ejemplo, aquellas que no requieren restricciones en la variación de retardo (mínimo) y variaciones de retardo.

UML (*Unified Modeling Language* - lenguaje unificado de modelado): brinda todas las características, tanto sintácticas como semánticas, para lograr caracterizar lógicamente cualquier tipo de software, permitiendo ser utilizado en cualquier etapa del diseño y especialmente útil en aquellos desarrollos enfocado a objetos.

UP - Proceso unificado: puede concebirse como una idea, una plantilla que provee una infraestructura para ejecutar proyectos, pero no tiene en cuenta todos los detalles requeridos en dicha ejecución.

Usuario: dicho de una persona que tiene derecho a usar una cosa con cierta limitación.

Videoconferencia: comunicación simultánea bidireccional de audio y video.

WEB: sistema de comunicación para el intercambio de mensajes a través de internet.

WWW (*World Wide Web*): sistema mundial de hipertexto que utiliza internet como mecanismo de transporte. Conjunto de páginas de información con texto, imágenes sonido, etc., enlazadas entre sí.

XML (*Extended Markup Language*): lenguaje de marcado extendido. Permite definir estructuras de información empleando etiquetas organizadas de manera jerárquica.

Apéndice A

Modelo de requisitos

Es un pilar importante para garantizar que la funcionalidad implementada sea de interés para un actor específico. Se emplea el mecanismo de casos de uso para describir la funcionalidad con un alto grado de abstracción técnica.

Modelo de casos de uso

Características generales de los actores

El sistema ofrece funcionalidad que es de interés para alguno de los siguientes actores (figura A1):

- Administrador: se encarga de la gestión de usuarios, el registro de nuevas categorías de nodos y tareas de administración tales como copias de seguridad, corrección de errores, edición de páginas, etc.
- Entidad de salud: para la gestión específica de un nodo tipo entidad de salud. Un usuario de este tipo está confinado a la institución que crea. Para poder gestionar información de otra entidad, deberá solicitar autorización a la instancia de actor dueña del registro.
- Servicios médicos: gestiona la información de nodos tipo servicio médico.
- Especialista: con permisos para gestionar nodos tipo especialista (profesionales de la salud).
- Profesional TIC: actor abstracto que agrupa a los actores que gestionan información de los nodos tipo equipo médico, operador de telecomunicaciones y tecnología de interconexión.
- Usuario general: usuario general de consulta. Puede generar informes pero no se le permite modificar ningún registro.
- Consultor: especificación del usuario general en donde además de consultar información tiene acceso a los módulos de tablas de análisis, agenda y blog.

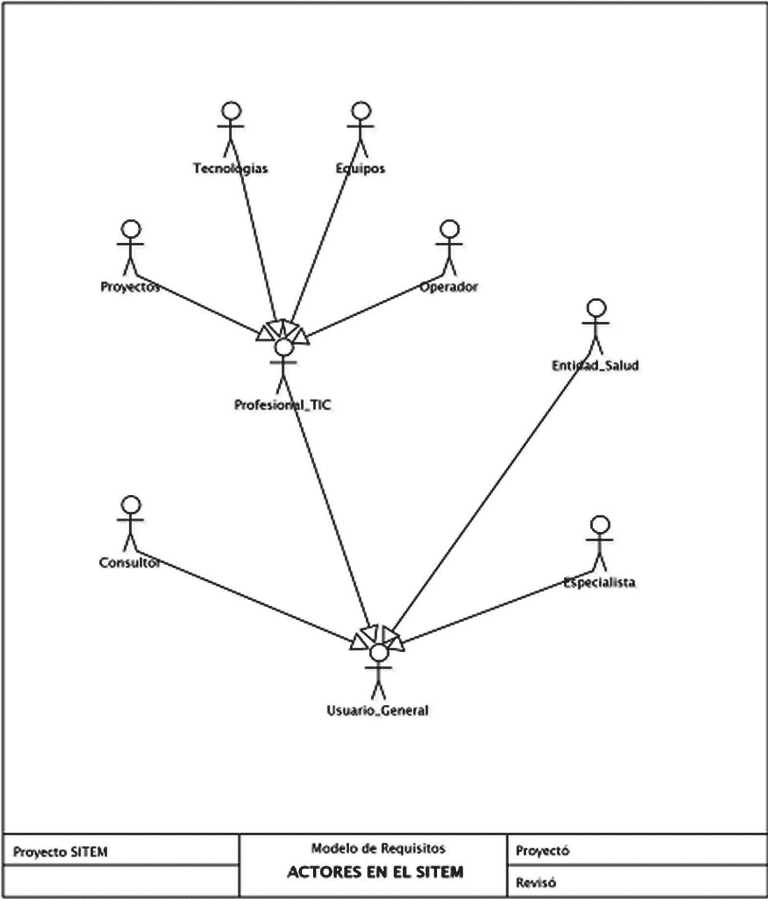


Figura A1. Conjunto de actores de OpenSITE
Fuente: elaboración propia.

Casos de uso

Los artefactos correspondientes a los casos de uso se han empaquetado de acuerdo con el tipo de nodo en el cual se desarrollan. Se incluyen solo los casos que se consideran nucleares y las especificaciones de mayor relevancia e impacto.

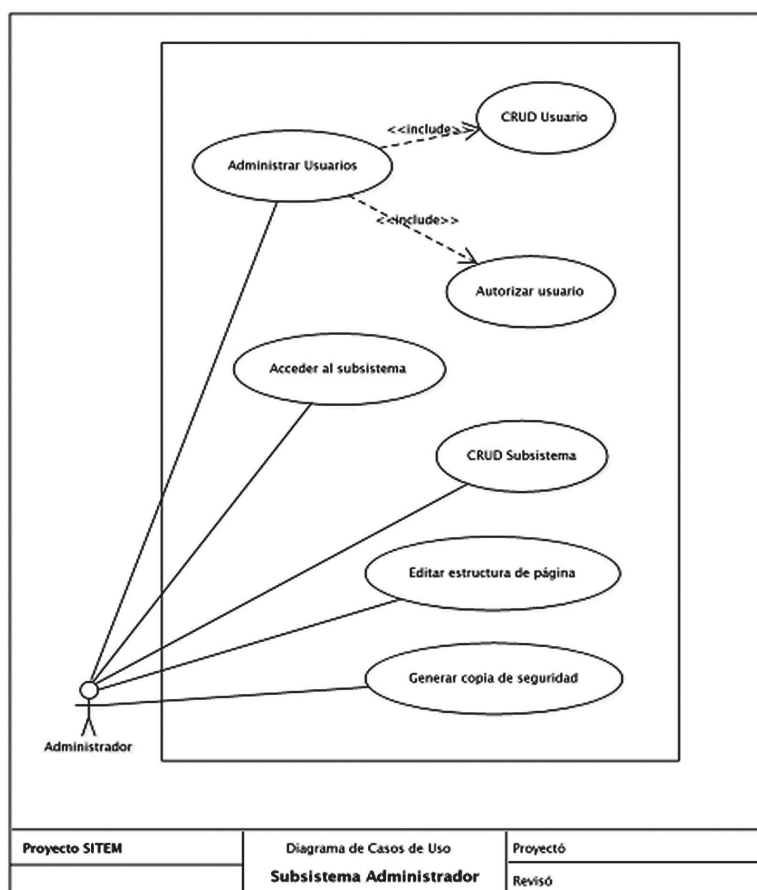


Figura A2. Casos de uso principales del actor administrador
Fuente: elaboración propia.

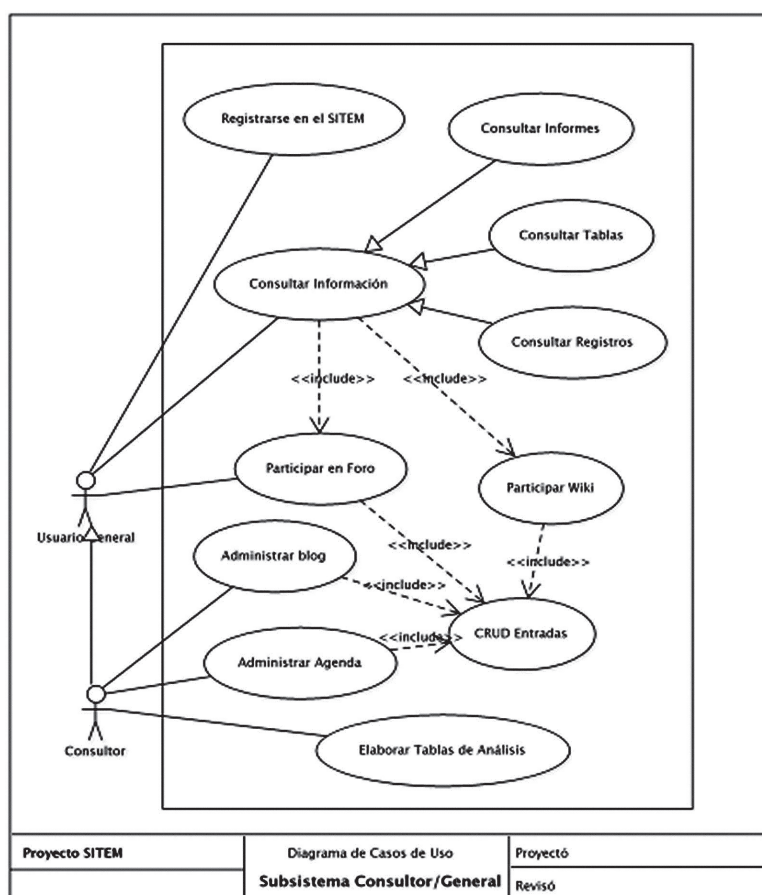


Figura A3. Casos de uso principales del actor consultor/usuario general
Fuente: elaboración propia.

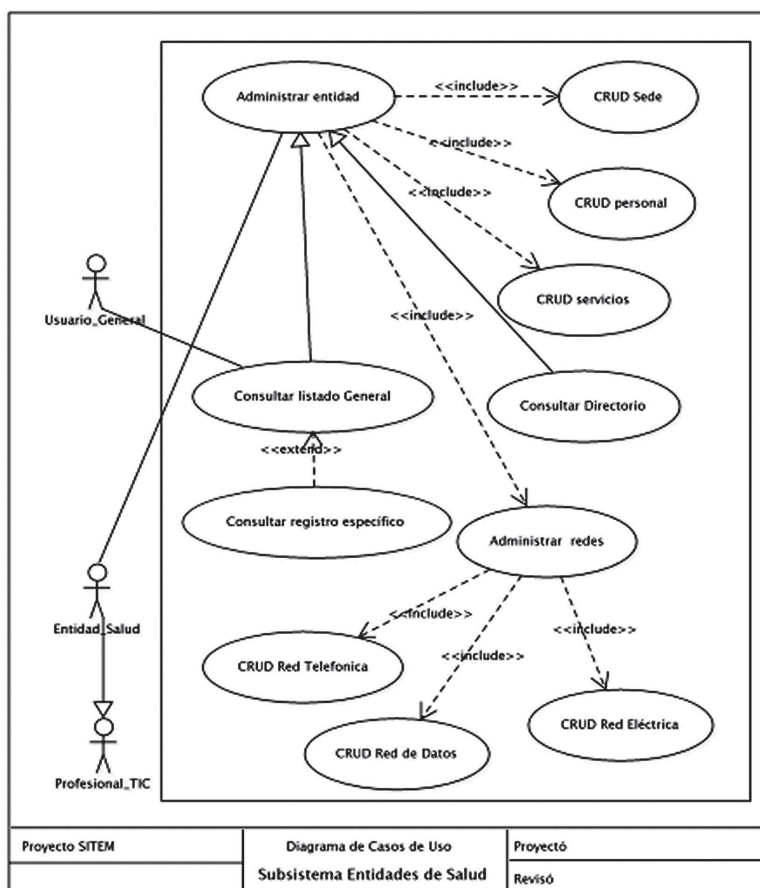


Figura A4. Casos de uso principales del actor entidad de salud
Fuente: elaboración propia.

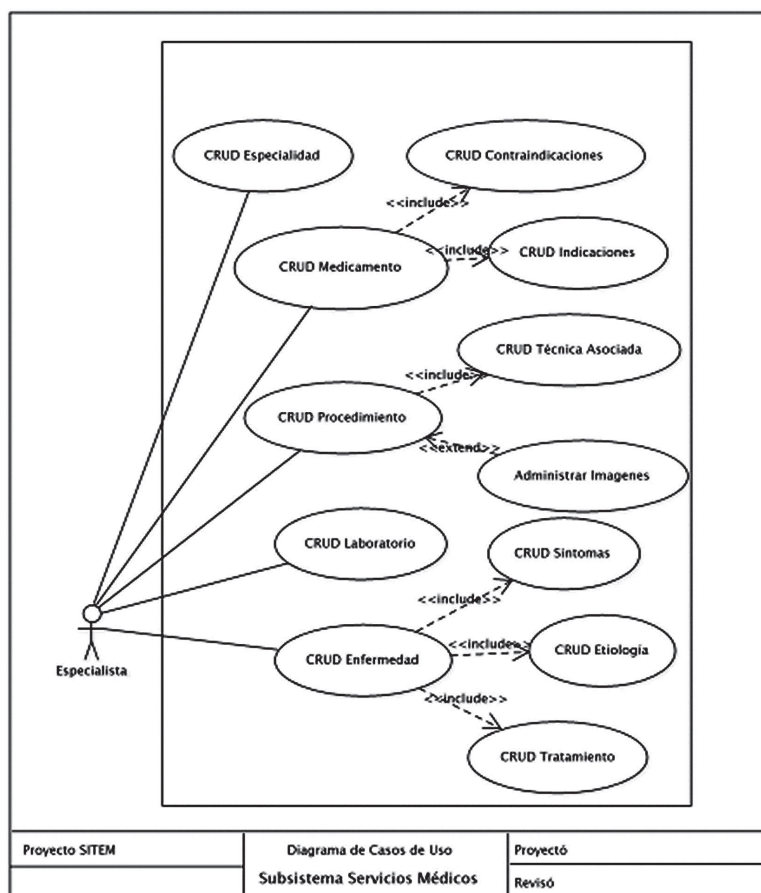


Figura A5. Casos de uso principales del actor servicios médicos
Fuente: elaboración propia.

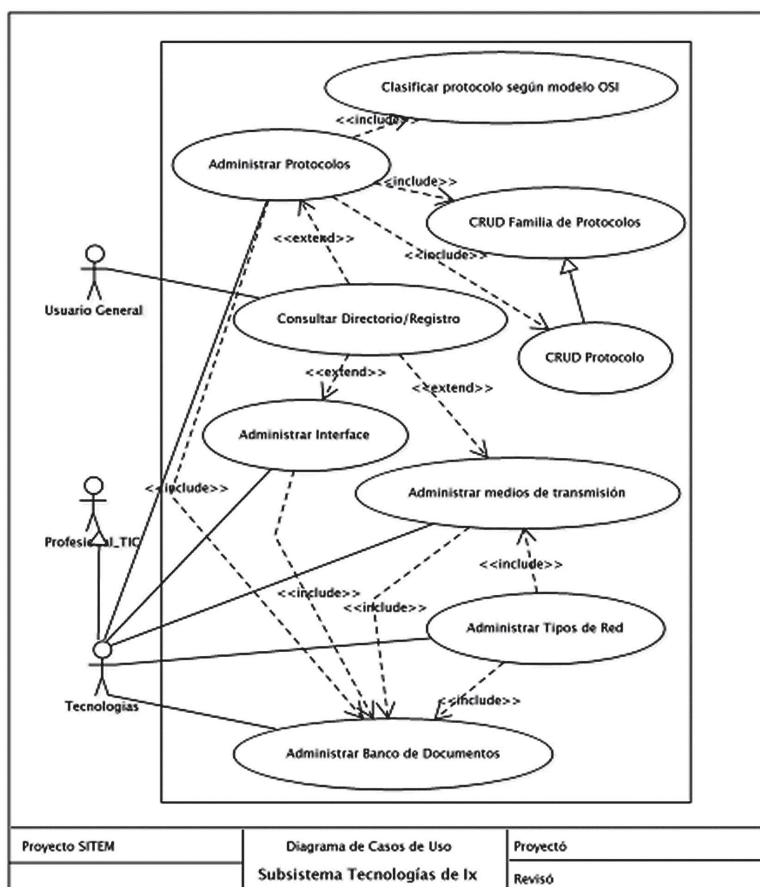


Figura A6. Casos de uso principales del actor tecnologías de interconexión
Fuente: elaboración propia.

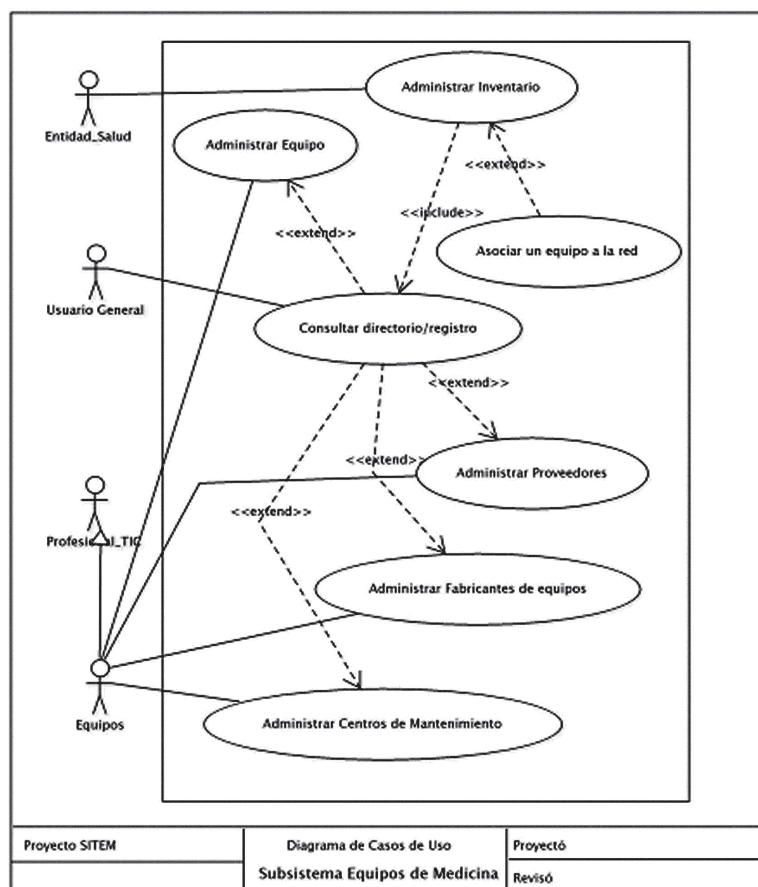


Figura A7. Casos de uso principales en el subsistema equipos de medicina
Fuente: elaboración propia.

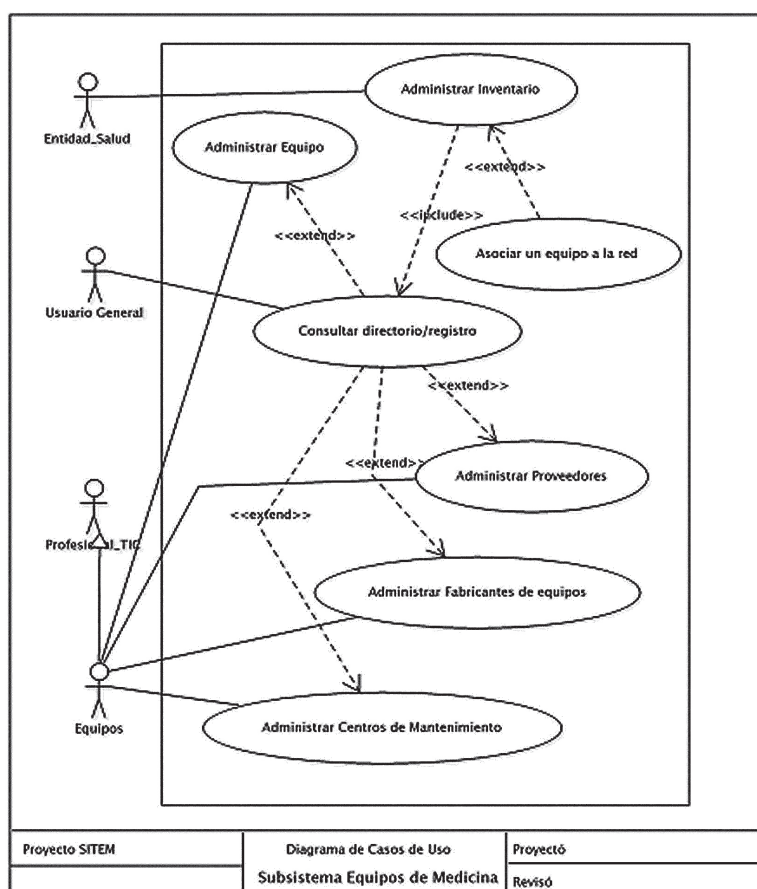


Figura A8. Casos de uso principales en el subsistema de operadores de servicios de telecomunicaciones
Fuente: elaboración propia.

Apéndice B

Modelo de análisis y diseño

Especificaciones de casos de usos

Para analizar en detalle los requerimientos del sistema se especifican mediante plantillas aquellos casos de uso que se consideran de importancia para el desarrollo base, conocidos como casos de uso nucleares. La tabla B1 muestra las secciones básicas que contiene un caso de uso. En aquellos casos de uso en que una o varias secciones carezcan de contenido relevante se omite por completo su declaración.

Tabla B1
Plantilla genérica para la especificación de los casos de uso

Caso de uso	
Nombre	Nombre que identifica el caso de uso, usualmente es el mismo utilizado en el diagrama de casos de uso
Objetivo	Beneficio que obtiene el actor con la ejecución de este caso de uso.
Código interno	Código único que identifica al caso de uso dentro del repositorio de artefactos.
Actores	Usuarios que intervienen en el caso de uso.
Precondiciones	Estado en que debe encontrarse OpenSITEM antes de ejecutarse el caso de uso.
Flujo básico	Flujo principal de actividades. Ambiente ideal.

Caso de uso	
Flujo alternativo y de error	Actividades que bifurcan el flujo básico. Si existe más de un flujo alternativo este debe colocarse en una nueva fila.
Postcondiciones	Estado en que queda OpenSITEM después de ejecutado el caso de uso.
Puntos de extensión	Secuencias de acciones que extienden el flujo del caso de uso.

Fuente: elaboración propia.

Tabla B2
Caso de uso Registrarse en OpenSITEM

Caso de uso	
Nombre	Registrarse en OpenSITEM
Objetivo	El actor logra crear una cuenta en OpenSITEM con un rol específico para poder trabajar en un subsistema dado.
Código interno	UC-GENERAL-001
Actores	Usuario general.
Precondiciones	Ninguna.
Flujo básico	1. El usuario general selecciona la opción de nuevo usuario desde la página principal de OpenSITEM.
	2. OpenSITEM muestra un formulario con los campos: Nombres Apellidos

Caso de uso	
Flujo básico	<p>Correo electrónico Teléfono Nombre de usuario Clave Reescriba la clave Acceso requerido</p>
	3. El usuario diligencia uno a uno los campos requeridos y opcionales.
	4. El usuario envía los datos a OpenSITEM.
	5. OpenSITEM verifica que los datos tengan los formatos esperados.
	6. OpenSITEM ingresa el registro a la base de datos colocando el campo de estado en 1 — registrado sin autorización.
	7. OpenSITEM redirecciona a la página de registro exitoso.
	8. El usuario acepta el mensaje.
Postcondiciones	Se agregó un registro en la base de datos con el campo de estado en 1.
Casos de uso relacionados	Seleccionar rol en OpenSITEM.

Fuente: elaboración propia.

Tabla B3
Caso de uso Autorizar un usuario para acceder a OpenSITEM

Caso de uso	
Nombre	Administrar autorizaciones de usuario
Objetivo	Proveer un mecanismo eficaz para que el administrador general de OpenSITEM pueda gestionar el estado de autorización de los usuarios en los diferentes subsistemas.
Código interno	UC-GENERAL-002
Actores	Administrador.
Precondiciones	Debe existir por lo menos un usuario registrado en el sistema diferente al administrador. El administrador se encuentra correctamente autenticado y autorizado en el subsistema de administrador.
Flujo básico	1. El administrador selecciona la opción Usuarios desde el menú principal del subsistema Administrador.
	2. OpenSITEM muestra un listado con los datos básicos de diferentes usuarios registrados: Nombres Apellidos Correo electrónico Acceso requerido
	3. Punto de extensión 1.
	4. Punto de extensión 2.
	5. El administrador verifica que los datos del usuario son reales.
	6. El administrador seleccionada la casilla de verificación y acepta el trámite.

Caso de uso	
Flujo básico	7. OpenSITEM procesa el formulario colocando el estado del usuario en valor 2 - Registrado y autorizado.
	8. OpenSITEM indica con un mensaje el éxito en la operación de autorización.
	9. OpenSITEM envía un mensaje de texto al usuario indicando que ha sido autorizado.
	8. El administrador acepta el mensaje de éxito.
Postcondiciones	Se cambia el valor en el campo estado del registro correspondiente al usuario.
Puntos de extensión	1: direccion="avanzar" o direccion="retroceder" extend Navegar en listado.
	2: Opción="buscar" extend Búsqueda condicional de registro.

Fuente: elaboración propia.

Tabla B4
Caso de uso realizar Copia de seguridad

Caso de uso	
Nombre	Generar copia de seguridad
Objetivo	Generar una copia de seguridad de los datos contenidos en la base de datos asociada a OpenSITEM.
Código interno	UC-GENERAL-003
Actores	Administrador.
Precondiciones	El administrador se encuentra correctamente autenticado y autorizado en el subsistema de administrador.

Caso de uso	
Flujo básico	1. El administrador selecciona la opción Copia de Seguridad en el menú principal del subsistema Administrador.
	2. OpenSITEM muestra un listado con las tablas opcionales para la copia de seguridad.
	3. El administrador selecciona la casilla de verificación de las tablas que desea sean copiadas.
	4. Punto de extensión 1.
	5. El administrador acepta la selección.
	6. OpenSITEM muestra un listado con las tablas que serán copiadas y un formulario con los campos: Nombre del archivo Ruta de descarga
	7. El usuario diligencia uno a uno los campos requeridos.
	8. El usuario envía los datos a OpenSITEM.
	9. OpenSITEM realiza una copia de los registros escribiéndolos uno a uno en los archivos de destino.
	10. OpenSITEM redirecciona a la página de operación exitosa.
Flujo básico	11. El usuario acepta el mensaje.
Postcondiciones	Se crean n archivos en la carpeta de destino con el contenido de las n tablas seleccionadas para copia de seguridad.
Puntos de extensión	1:opcion="todo" extend Seleccionar todos los cuadros.

Fuente: elaboración propia.

Tabla B5
Caso de uso Elaborar tablas de análisis

Caso de uso	
Nombre	Elaborar tablas de análisis
Objetivo	Obtener un repositorio de análisis de algún componente del modelo jerárquico de seguimiento a proyectos.
Código interno	UC-GENERAL-004
Actores	Consultor.
Precondiciones	Existe en OpenSITEM un modelo jerárquico de seguimiento a proyectos.
Flujo básico	1. El consultor selecciona la opción Seguimiento desde el menú principal del subsistema Consultor.
	2. OpenSITEM muestra el modelo de seguimiento a proyectos con sus componentes de primer nivel.
	3. Punto de extensión 1.
	4. El consultor selecciona la opción de Analizar para un componente.
	5. OpenSITEM muestra un formulario con los campos: Valoración cualitativa Valoración cuantitativa Juicio Diagnóstico Fortalezas Oportunidades Debilidades Amenazas Directrices de mejoramiento Directrices de acción

Caso de uso	
Flujo básico	6. El usuario diligencia uno a uno los campos requeridos y opcionales.
	8. El usuario envía los datos a OpenSITEM.
	5. OpenSITEM verifica que los datos tengan los formatos esperados.
	6. OpenSITEM ingresa el registro a la base de datos.
	7. OpenSITEM redirecciona a la página de registro exitoso.
	8. El usuario acepta el mensaje.
Postcondiciones	Existe un registro de análisis asociado a un componente y un consultor.
Puntos de extensión	1:opcion="mas" extend Mostrar Componentes de Nivel Inferior

Fuente: elaboración propia.

Tabla B6
Caso de uso Ingresar una nueva entidad de salud

Caso de uso	
Nombre	Ingresar una entidad
Objetivo	Registrar una nueva entidad de Salud en OpenSITEM.
Código interno	UC-GENERAL-005
Actores	Entidad Salud.

Caso de uso	
Precondiciones	El usuario entidad salud se encuentra autorizado y autenticado en el subsistema Entidades de Salud.
Flujo básico	1. Entidad salud selecciona la opción Nueva Entidad desde el menú principal del subsistema Entidades de Salud.
	2. OpenSITEM muestra un formulario con los campos: Nombre de la entidad Nombre corto Logosímbolo NIT Fecha de fundación Dirección principal Teléfono principal (PBX) Misión Visión Descripción Comentarios
	3. El usuario diligencia uno a uno los campos requeridos y opcionales.
	4. El usuario envía los datos a OpenSITEM.
	5. OpenSITEM verifica que los datos tengan los formatos esperados.
	6. OpenSITEM comprueba que no exista otra entidad registrada con el mismo NIT.
	7. OpenSITEM ingresa el registro a la base de datos.
	8. OpenSITEM redirecciona a la página de registro exitoso.
	9. El usuario acepta el mensaje.

Caso de uso	
Flujo alternativo	5.A. Los datos no tienen el formato adecuado.
	6.A. OpenSITEM informa el error.
	7.A. Punto de extensión 1.
Flujo alternativo	6.A. Existe una entidad registrada con el mismo NIT.
	7.A. OpenSITEM informa el error.
	8.A. Punto de extensión 1.
Postcondiciones	Existe un registro de una entidad de salud.
Puntos de extensión	1:opcion="corregir" extend Mostrar Formulario Corrección

Fuente: elaboración propia.

Tabla B7
Caso de uso Consultar información básica de una entidad de salud

Caso de uso	
Nombre	Consultar información básica de una entidad de salud
Objetivo	Obtener en pantalla los datos básicos de una entidad de salud.
Código interno	UC-GENERAL-006
Actores	Entidad salud, usuario general.
Precondiciones	El usuario se encuentra autorizado y autenticado en el subsistema Entidades de Salud.

Caso de uso	
Flujo básico	1. Entidad Salud selecciona la opción Entidades desde el menú principal del subsistema Entidades de Salud.
	2. OpenSITEM muestra un listado de 25 entidades ordenadas alfabéticamente por nombre.
	3. Punto de extensión 1.
	4. El usuario selecciona una entidad de salud desde el listado.
	5. OpenSITEM realiza una búsqueda con el id de la entidad.
	6. OpenSITEM muestra en pantalla el menú secundario para solicitar edición y los datos de la entidad: Nombre de la entidad Nombre corto Logosímbolo NIT Fecha de fundación Dirección Teléfono principal (PBX) Misión Visión Descripción
	7. El usuario acepta los datos.
Puntos de extensión	1: direccion="avanzar" o direccion="retroceder" extend Navegar en listado

Fuente: elaboración propia.

Tabla B8
Caso de uso Editar un registro en OpenSITEM

Caso de uso	
Nombre	Editar un registro en OpenSITEM
Objetivo	Editar la información que se encuentra en un registro del SITEM. La actualización puede involucrar más de una entidad en la capa de persistencia.
Código interno	UC-GENERAL-007
Actores	Profesional TIC, entidad salud, administrador, usuario general
Precondiciones	El usuario se encuentra autorizado y autenticado en el subsistema.
Flujo básico	1. El usuario selecciona la opción Editar Registro desde el menú secundario del subsistema.
	2. OpenSITEM muestra un formulario rellenado con los datos del registro correspondiente.
	3. El usuario edita los valores dentro de los controles del formulario.
	4. El usuario envía el formulario a OpenSITEM.
	5. OpenSITEM verifica que los datos editados no violen alguna regla de integridad referencial.
	6. OpenSITEM actualiza los registros en la capa de persistencia.
	7. OpenSITEM muestra al usuario un mensaje de éxito.
	8. El usuario acepta el mensaje.

Caso de uso	
Flujo alternativo	5.A. Existe un error de integridad referencial.
	7.A. OpenSITEM informa el error.
	8.A. Punto de extensión 1.
Puntos de extensión	1:opcion="corregir" extend Mostrar Formulario Corrección
Postcondiciones	Se actualizan los registros correspondientes.

Fuente: elaboración propia.

Tabla B9

Caso de uso Asociar un protocolo de comunicaciones al modelo OSI

Caso de uso	
Nombre	Asociar un protocolo de comunicaciones al modelo OSI.
Objetivo	Asociar un protocolo de comunicaciones al modelo de referencia OSI.
Código interno	UC-GENERAL-008
Actores	Profesional TIC, tecnologías.
Precondiciones	El usuario se encuentra autorizado y autenticado en el subsistema.
	Existe registrado por lo menos un protocolo de comunicaciones en OpenSITEM.
Flujo básico	1. El usuario selecciona la opción más información desde el menú secundario del subsistema.

Caso de uso	
Flujo básico	2. OpenSITEM muestra la información del protocolo asociada por áreas temáticas.
	3. El usuario selecciona la opción Clasificar OSI.
	4. OpenSITEM muestra el gráfico de siete capas del modelo OSI.
	5. El usuario selecciona una o varias capas del modelo.
	6. OpenSITEM asocia el ID del protocolo a cada una de las capas del modelo OSI seleccionadas por el usuario.
	7. OpenSITEM muestra el modelo OSI extendido con los demás protocolos registrados en cada capa.
	8. El usuario acepta el registro.
Flujo alternativo	5.A. El usuario no selecciona ninguna capa.
	7.A. OpenSITEM regresa al punto 2 del caso de uso.
Postcondiciones	El conjunto de protocolos asociados a una capa del modelo OSI se incrementa.

Fuente: elaboración propia.

Tabla B10

Especificación caso de uso para Borrar un registro en OpenSITEM

Caso de uso	
Nombre	Borrar un registro de OpenSITEM
Objetivo	Eliminar un registro en algún subsistema de OpenSITEM garantizando que solo el experto en información lo realice y se mantenga la integridad referencial en los datos.

Caso de uso	
Código interno	UC-GENERAL-009
Actores	Profesional TIC, entidad salud, consultor, administrador.
Precondiciones	El usuario se encuentra autorizado y autenticado en el subsistema.
	Existe un registro en OpenSITEM.
	El usuario ha creado el registro y este no tiene información asociada.
Flujo básico	1. El usuario selecciona la opción Eliminar Registro desde el menú secundario del subsistema.
	2. OpenSITEM muestra un mensaje de conformación de eliminación con los datos básicos del registro.
	3. El usuario selecciona la opción de Confirmar Borrado.
	4. OpenSITEM elimina el registro cumpliendo las restricciones de claves foráneas.
	5. El sistema muestra un mensaje indicando que el registro se ha borrado del sistema.
	6. El usuario acepta el mensaje.
	7. OpenSITEM redirecciona a la página en donde se encontraba el usuario antes del proceso de borrado.
Flujo alternativo	3.A. El usuario no acepta borrar el registro.
	4.A. Continúa en el punto 7 del flujo principal.
Postcondiciones	El registro se borra del sistema.

Fuente: elaboración propia.

Tabla B11
Especificación caso de uso Para acceder a una página del SITEM

Caso de uso	
Nombre	Acceder a una página de OpenSITEM.
Objetivo	Ingresar a una página específica dentro del sistema.
Código interno	UC-GENERAL-010
Actores	Profesional TIC, entidad salud, consultor, administrador.
Precondiciones	El usuario está autorizado para acceder a la página.
	La página se encuentra registrada en OpenSITEM.
	La página tiene uno o más bloques asociados.
Flujo básico	1. El usuario elige un enlace a una página dentro de OpenSITEM.
	2. OpenSITEM verifica que el usuario tiene una sesión válida.
	3. OpenSITEM rescata los valores de la página desde la base de datos.
	4. OpenSITEM verifica que el usuario tenga los privilegios necesarios para ingresar a la página.
	5. OpenSITEM consulta la estructura de la página desde la base de datos.
	6. OpenSITEM envía secuencialmente el código HTML correspondiente a cada uno de los bloques que constituyen la página.
	7. OpenSITEM registra el acceso del usuario en la base de datos.

Caso de uso	
Flujo básico	8. OpenSITEM actualiza la información de sesión.
Flujo alternativo	4.A. El usuario no tiene los privilegios para ver la página.
	5.A. OpenSITEM registra un intento de ingreso ilegal.
	6.A. OpenSITEM muestra un mensaje informando el error.
Postcondiciones	El usuario ingresa a una página dentro de OpenSITEM.

Fuente: elaboración propia.

Apéndice C

Modelo de implementación

El modelo de implementación de OpenSITEM está compuesto básicamente por los siguientes artefactos que en su conjunto representan lo que comúnmente se denomina el aplicativo:

- Bloques: unidad básica de funcionalidad. Se puede pensar en ellos como “instancias” persistentes de clases abstractas. En especial se tienen tres clases:
 - Administración: con atributos y métodos para mostrar directorios de datos.
 - Menú: con métodos especializados en la administración de enlaces dentro de OpenSITEM.
 - Registro: para la realización de casos CRUD.
- Clases: descriptores para varios tipos de objetos entre las cuales se tiene:
 - DBMS: interacción con la bases de datos.
 - Página: describe objetos que realizan la construcción en tiempo de ejecución de las páginas.
 - Encriptar: descriptor para objetos que se encargan de codificar y decodificar los datos en OpenSITEM. El conjunto de operaciones debe ser manipulado en cada implementación del SITEM para garantizar un alto nivel de seguridad.
 - Autenticación: con descripción de atributos y operaciones que controlan las rutinas de AAA en OpenSITEM.
 - Config: clasificador de objetos que manipulan las variables de configuración globales.
 - Html: descriptor de controles de formularios en OpenSITEM.
 - Sesión: operaciones y atributos para el control de sesiones en OpenSITEM luego del proceso de AAA.
 - Mensaje: para objetos que administran mensajes de interacción con los actores.

- Sql: clase para describir objetos especializados en gestionar archivos con extensión SQL.
- Navegación: con operaciones específicas para el control de desplazamiento entre conjuntos de registros.
- Función: métodos JavaScript para la validación y control de navegación en el lado del cliente.
- Estilo: para el control de la capa de interfaz gráfica.

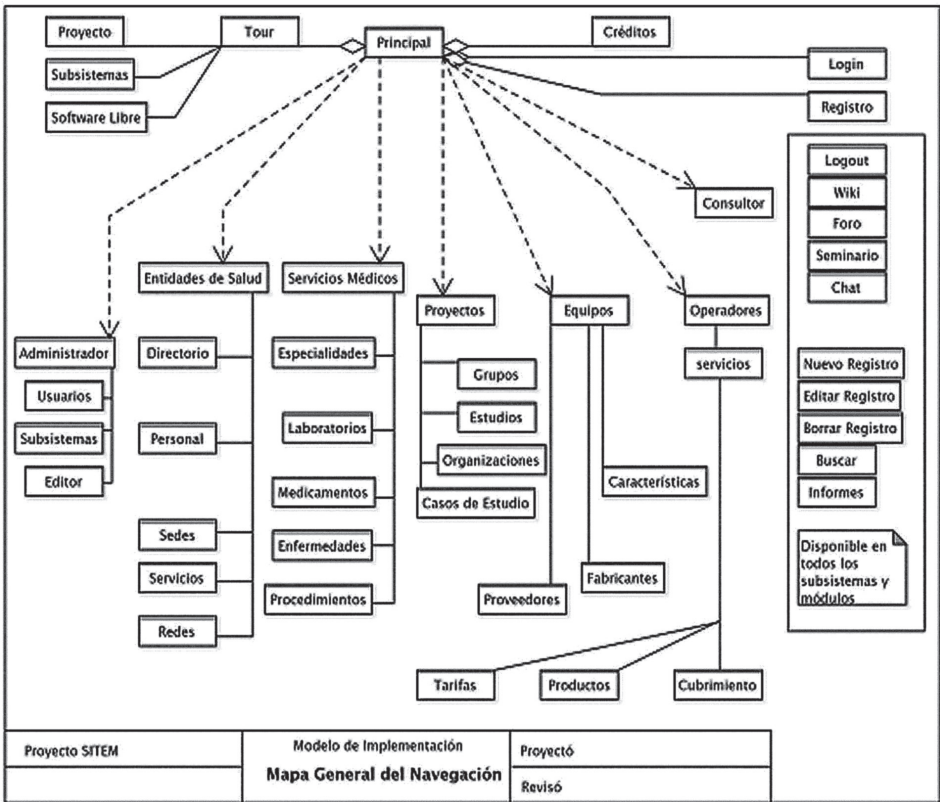


Figura C1. Modelo general de navegación
Fuente: elaboración propia.

Código fuente de OpenSITEM

La siguiente porción de código fuente representa el formato general que se encuentra en OpenSITEM. Por patrón general, se recomienda a todos los grupos que participan en el desarrollo que mantengan un esquema de codificación claro y documentación *in situ* suficiente para aclarar secciones.

Clase página

La clase Página tiene las operaciones:

- Página: constructor.
- Especificar página: inicializar variables privadas.
- Procesar página: controlar redireccionamientos.
- Ancho sección: implementa control de secciones colapsadas.
- Armar sección: para seleccionar los bloques que contiene cada página.

Algunos métodos instancian la clase DBMS y HTML. Estas por lo tanto deben ser visibles.

```
class pagina
//Método constructor function pagina(id_pagina,configuracion)
//Declaración de variable para controlar accesos indebidos
GLOBALS["autorizado"]=TRUE; this->especificar_pagina(id_pagina);
if(!isset(_POST['registro_compuesta']))
if(!isset(_REQUEST['action']))
this->mostrar_pagina(configuracion);
else //echo 'Procesamiento de la pagina'; this->procesar_pagina(configuracion);
else this->mostrar_pagina(configuracion);
//Fin del método constructor
//Metodo especificar_pagina function especificar_pagina(id_pagina)
this->id_pagina=id_pagina;
//Fin del método especificar_pagina . . .
```


Apéndice D

Modelo de datos

La capa de persistencia en OpenSITEM está soportada en una estructura de base de datos de modelo relacional. La forma normal básica - 3NF se garantiza, mientras que otras formas normales pueden ser omitidas en casos puntuales donde se demuestre, por pruebas de desempeño, que no seguirlas redundaría en una mejora significativa de la velocidad en el acceso a los datos sin detrimento de su calidad e integridad. A continuación se listan las formas normales tenidas en cuenta como patrón de diseño en el modelo de datos de OpenSITEM:

- Primera forma normal: cada renglón-columna contiene valores atómicos.
- Segunda forma normal: 1NF y todo campo que no sea clave primaria depende de los campos clave.
- Tercera forma normal: 2NF y no hay dependencias transitivas.
- Forma normal de Boyce-Codd: todos los determinantes de la tabla son clave candidata.
- Cuarta forma normal: una fila no debe contener dos o más campos multivalorados.
- Quinta forma normal: una tabla puede almacenar atributos dependientes a la clave solo por unión.

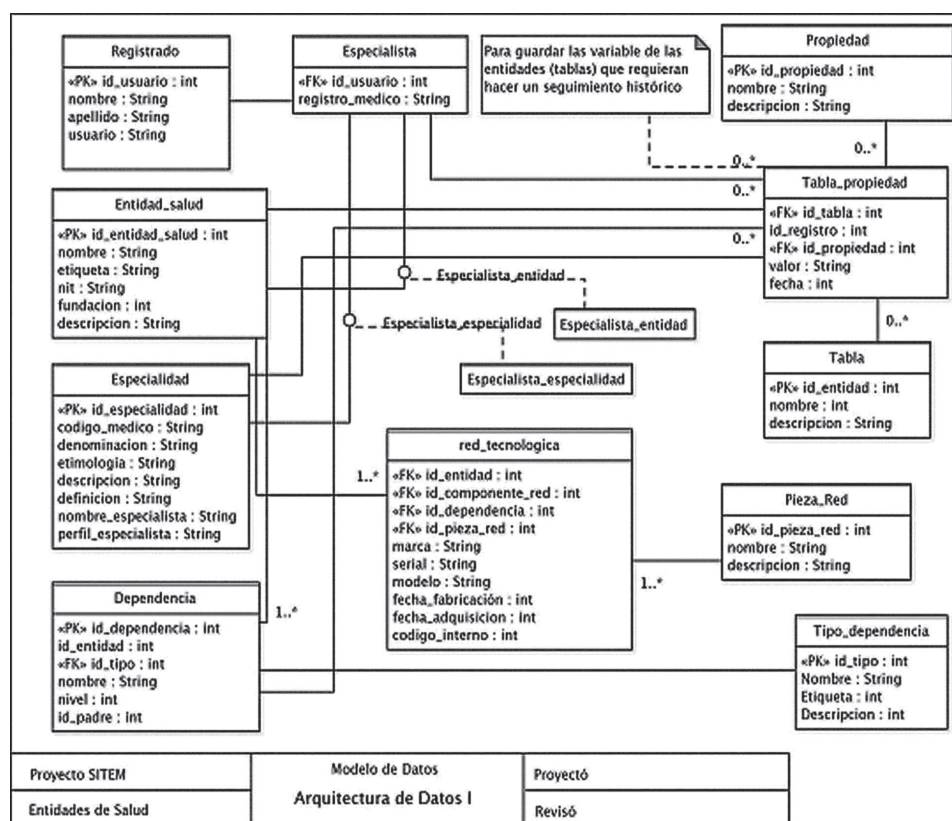


Figura D1. Arquitectura de datos Subsistema Entidades de Salud

Fuente: elaboración propia.

Apéndice E

Modelo de despliegue

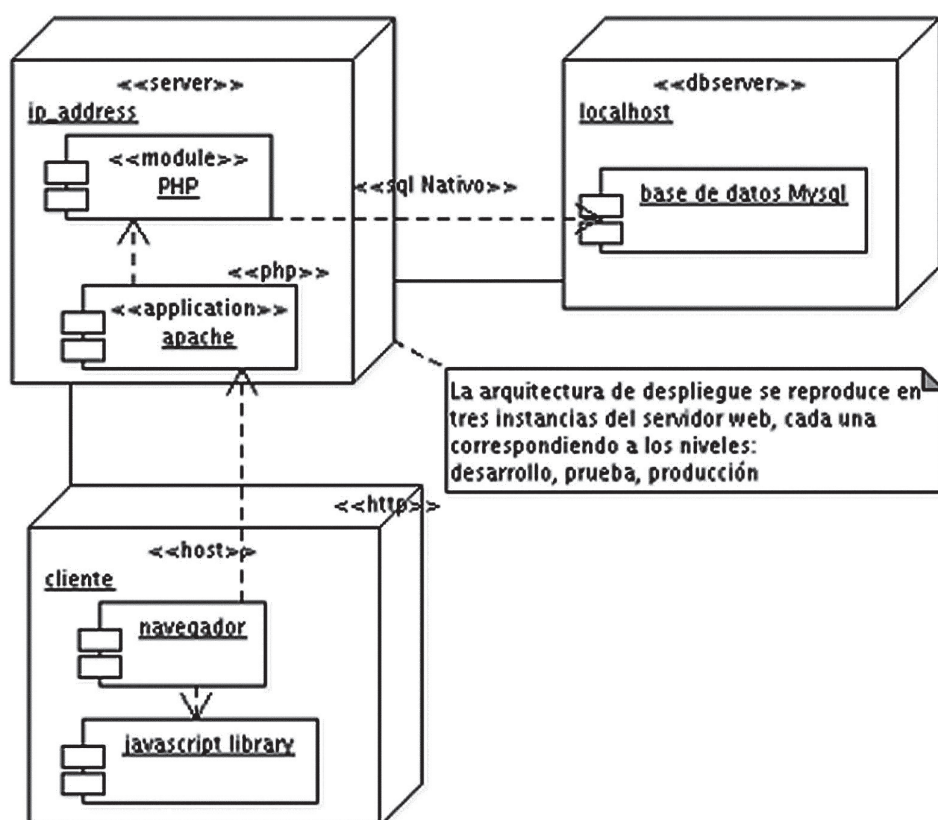


Figura E1. Modelo de despliegue
Fuente: elaboración propia.

OpenSITEM se despliega sobre un servidor que tenga instalado un programa que acepte peticiones usando el protocolo HTTPS - Servidor Web, soporte para el intérprete de PHP y un servidor de bases de datos MySQL o PostgreSQL. La figura E1 muestra el diagrama de componentes correspondiente a la arquitectura propuesta.

La arquitectura mostrada se reproduce en tres instancias del servidor web, cada una de las cuales corresponde a los niveles de:

- Desarrollo: con las versiones más recientes del sistema que se consideran en versión alfa o inestables. Dentro de la codificación son aquellas que muestran incrementos en el cuarto segmento: AA.XX.YY.ZZ
- Prueba: versiones que tienen desarrollos estables pero que aún no han pasado la etapa de revisión de casos de prueba. En la codificación son aquellas que muestran un incremento en el tercer segmento: AA.XX.YY.ZZ
- Producción: versiones que han sido probadas y no tienen errores detectados. De acuerdo con la naturaleza de la funcionalidad alcanzada, estas versiones representan un incremento en los segmentos primero o segundo dentro de la codificación.
- Si es el primer segmento el que incrementa, se alcanza un ciclo de desarrollo y la versión es bautizada con un nombre específico. Para OpenSITEM se usan los nombres de los dioses mayas que participaron en alguno de los tres intentos por crear la humanidad.

Apéndice F

Declaración de riesgos

El proyecto está expuesto a una serie de contingencias que de no ser correctamente manejadas atentan contra la consecución de sus metas. Estos riesgos deben estar detectados de la forma óptima, evaluados y medido su impacto. Cada vez que se avanza en el ciclo de desarrollo unos se vuelven más importantes que otros. A continuación se declaran los más importantes.

- **Baja disponibilidad de tiempo**

La falta de tiempo por parte de los integrantes del proyecto debido al trabajo en paralelo de otros proyectos es un riesgo inminente que de seguro puede derivar en la no consecución de los objetivos. El reclutamiento de integrantes cuyo objetivo primario es el lleno de un requisito para grado pone en riesgo la continuidad en el desarrollo de algún hilo.

Métodos para la reducción del riesgo: es aquí donde los métodos ágiles son de gran ayuda. El grupo de desarrollo fomenta la retroalimentación continua entre los diferentes hilos, creando una especie de competencia sana en donde los componentes y prácticas más eficientes son detectados y propagados en las sesiones de capacitación. A partir de la versión 3.0, actualmente en desarrollo, a los grupos adscritos no se les impone un objetivo específico, sino que las personas interesadas simplemente deciden qué tarea es más relevante y su participación solo se tiene en cuenta a través de la producción-posicionamiento basado en el esfuerzo.

- **Pérdida de tiempo debido a procesos administrativos**

Las cargas administrativas inherentes a los procesos del proyecto consumen gran cantidad de tiempo y en ciertas circunstancias pueden tomar más del esperado y obstruir el desarrollo de las actividades en el proyecto. La dependencia de algunas actividades del formalismo burocrático desenfoca al grupo de trabajo.

Métodos para la reducción del riesgo: el trabajo se realiza en varias áreas y disciplinas de manera paralela, brindando siempre la posibilidad de redistribuir recursos si se requiere esperar a que algún trámite administrativo se concrete. El jefe del proyecto se encarga de realizar todos los trámites y la generación de documentos necesarios con el fin de agilizar los procesos administrativos.

- **Falta de recursos para trabajo**

El mayor riesgo para la elaboración del proyecto y la consecución de sus objetivos se encuentra en no disponer de recursos para dichas tareas. Principalmente, un equipo de cómputo capaz de soportar sistemas operativos Linux, así como de un espacio físico de trabajo donde se encuentre dicho equipo y los integrantes del proyecto puedan realizar sus actividades y centralizar información.

Métodos para la reducción del riesgo: cada uno de los integrantes del proyecto está en capacidad de realizar las tareas que le corresponden por su cuenta, fomentando el teletrabajo. El jefe del proyecto se encargó de realizar la centralización del trabajo y la información por medio del préstamo al proyecto de recursos propios. En fases posteriores, el grupo de investigación recibe apoyo del Centro de Investigaciones y Desarrollo Científico de la Universidad Distrital, con lo que se adquiere un servidor con acceso público a internet.

- **Falta de información y demora en su consecución**

El proyecto requiere y se basa en cierta información proveniente de diversas fuentes, como publicaciones, manuales y estudios realizados dentro y fuera del grupo GITEM; la demora o imposibilidad de consecución de información calificada se convierte en un riesgo para el desarrollo normal del proyecto.

Métodos para la reducción de riesgo: desde el comienzo del proyecto, se realiza un listado de información necesaria e indispensable y se inicia su búsqueda y consecución. El jefe de proyecto lleva a efecto los procedimientos requeridos con el fin de contar con los resultados de estudios previos en el interior del GITEM. En todo momento se mantiene un cúmulo documental disponible para la comunidad de desarrollo, las jornadas de capacitación y el despliegue de presentaciones focalizadas.

- **Falla en la dirección del proyecto**

Existe la posibilidad de que el director del proyecto o alguno de sus ayudantes cometa algún error o falla, atentando de esta manera con el proceso natural de desarrollo.

Métodos para la reducción de riesgo: se realiza un seguimiento continuo a todas las tareas asociadas a la dirección del proyecto. El concepto y objetivos de OpenSITEM plasmados en su documento de visión se han convertido en un artefacto de consulta continua y recurrente.

- **Valoración nula o errónea de los resultados**

Al terminar una iteración dentro del proceso de desarrollo, es posible que no se obtengan los resultados esperados. El exceso de planificación y la rigidez frente al cambio pueden generar que se subvaloren los resultados obtenidos y, con la excusa de no cambiar un documento impreso y empastado, sacrificar nuevas ideas y mejoras.

Métodos para la reducción de riesgo: se ha difundido claramente en el grupo de desarrollo el hecho de que todos los artefactos están en construcción; nunca se valora uno de ellos como la versión definitiva. El uso de términos correctos como

las versiones estables permite que los entes de control dentro del grupo puedan realizar métricas a corto plazo que en sumatoria pueden arrojar datos valiosos en cuanto al avance del proyecto. Los objetivos de los trabajos de grado pueden ser modificados respecto a los anteproyectos, siempre y cuando se conserve su espíritu. En la práctica se utiliza Cervisia, un sistema de CVS que permite el seguimiento del proyecto, sus resultados y errores.

- **Falta de motivación de los integrantes del proyecto**

Debido a la posibilidad de que los integrantes del grupo en ciertos momentos y etapas del desarrollo del proyecto carezcan de la motivación necesaria, se corre el riesgo de que las actividades asignadas no se realicen o se realicen de una forma no satisfactoria. Usualmente esto es fruto de reclutamientos forzosos de personal.

Métodos para la reducción de riesgo: a integrantes de fases preliminares se les dictan capacitaciones en los métodos de desarrollo ágiles y se les descarga en gran medida la responsabilidad de documentar el desarrollo-enfoque 100 % centrado en el código. Para el cumplimiento de las tareas y actividades se asigna un tiempo adicional con el objetivo de que los integrantes del grupo resuelvan sus dudas y preocupaciones sobre el proyecto y así acometer una determinada tarea entre todos los integrantes.

En la actualidad, el proyecto es de carácter abierto y cualquier persona puede, sin una intermediación directa del grupo de investigación, realizar trabajos sobre este. Los desarrollos que deseen ser validados podrán solicitar dicha certificación al grupo de desarrollo principal, el cual sopesará los resultados y en algunos casos concretos el proceso de desarrollo. Es decir, a partir de la versión 3.0 de OpenSITEM ningún nuevo integrante es regulado por las políticas del grupo principal de desarrollo; es una decisión individual el seguimiento o no de los lineamientos.

- **Imposibilidad total de trabajo**

Quizá por motivos externos al proyecto, como paros, vacaciones o cierres, es posible que el trabajo se detenga totalmente.

Métodos para la reducción de riesgo: teletrabajo, depreciación de la comunicación persona a persona y uso intensivo de herramientas de trabajo en grupo dentro de la plataforma de OpenSITEM, especialmente las de primer nivel como correo electrónico, canales IRC y foros. Las de segundo nivel, como chat y videoconferencia en tiempo real, han sido poco utilizadas para minimizar el riesgo colateral de segregación tecnológica.

- **Capacitación deficiente**

Es posible que algún miembro del proyecto asignado a alguna tarea no posea todos los conocimientos necesarios para realizarla.

Métodos para la reducción de riesgo: dentro de OpenSITEM se han creado cursos en línea con contenidos actualizados a los cuales los integrantes podrán acceder de forma asíncrona a través de internet. En los foros se guarda información

relevante a los problemas en el desarrollo de tareas, con el objetivo de que todo el equipo de trabajo pueda encontrar soluciones basadas en la colaboración. El jefe del proyecto puede asignar recursos enfocados en la capacitación de ciertas áreas específicas.

- **Mal funcionamiento de herramientas de software y hardware**

La posibilidad de fallo en los equipos de trabajo y de los recursos utilizados puede causar interrupción de actividades y pérdida de tiempo.

Métodos para la reducción de riesgo: en el grupo de desarrollo principal existe un plan de mantenimiento a corto plazo que contempla actividades para el aseguramiento de la integridad de la plataforma de hardware y de los datos asociados al proyecto. Con la autorización para desplegar el hilo principal en servidores de aplicaciones se tendrá un mayor respaldo.

- **No seguimiento del plan de trabajo**

Las libertades propias de cada integrante del grupo, además del conjunto formado por todos los otros riesgos, pueden causar un desarrollo no planeado. En sí este riesgo solo es evidente cuando se captan recursos que van enfocados al cumplimiento de objetivos específicos. A partir de la versión 3.0, el grupo principal será el encargado de dictaminar los alcances de los nuevos *releases*.

Método para la reducción de riesgo: uno o más de los integrantes deben actuar como ingenieros de procesos y se encargarán de verificar el cumplimiento del plan de trabajo.

- **Desacuerdo entre los integrantes del grupo**

La posibilidad de desacuerdos de trabajo o personales está presente en el interior del proyecto.

Método para la reducción de riesgo: en el caso de desarrollo novedoso, los integrantes en desacuerdo deberán seguir las reglas de contribución concurrente. Para todos los demás casos —procedimiento depreciado— el jefe de proyecto y el ingeniero de procesos deberán encargarse de tomar cualquier decisión con miras al cumplimiento de los objetivos.

- **Desviación de los recursos de trabajo**

Es posible que los recursos obtenidos para el desarrollo del proyecto sean utilizados para otras actividades que no involucren la obtención de objetivos específicos.

Método para la reducción de riesgo: el ingeniero de procesos verifica el correcto cumplimiento del plan de trabajo, así como la correcta utilización de los recursos. Se realiza un historial de utilización de los recursos.

- **Otros riesgos**

Existen otros riesgos como la renuncia al proyecto de algún miembro, el cambio de los objetivos del proyecto, cambio en las características del mercado y eventos externos que afecten el normal desarrollo del proyecto.

Método para la reducción de riesgo: proceso de desarrollo ágil, con elementos claves del proceso unificado, utilización de artefactos, desarrollo centrado en la persona y compromiso con la filosofía general de la libertad y el derecho a la información.

Apéndice G

Instrumento preliminar para la fase de transición

Presentación

El presente instrumento agrupa de una manera lógica los aspectos necesarios para identificar el estado y la potencialidad de los servicios de salud que hacen uso intensivo de las TIC. El contenido se basa en aquel utilizado por el proyecto HERMES, modelo completamente depurado que ha servido de base para la recolección de información de varios proyectos de telemedicina emprendidos en la Comunidad Económica Europea (CEE). Así mismo, se le han adicionado algunos ítems aplicables al contexto colombiano.

Indicaciones generales

El cuestionario deberá ser respondido como fase posterior a la etapa de divulgación del proyecto y una vez se haya conformado un comité, o en su defecto, se haya designado un encargado, por parte de la institución objetivo. Es necesario que en el proceso los miembros del proyecto sirvan de apoyo resolviendo de una manera exacta las dudas que surjan en el ámbito tecnológico (telesalud, estándares de telecomunicaciones y otros). De igual forma, se deben realimentar de la información especializada que puedan capturar de los comités. Para apoyar estas tareas podrán hacer uso de las herramientas de trabajo en grupo que se encuentran en el portal OpenSITEM.

La profundidad con que se respondan las cuestiones garantiza que la institución evalúe su situación actual frente a la posible implementación de servicios de salud prestados en la modalidad de telemedicina.

El cuestionario está disponible para ser diligenciado en línea, de tal forma que podrá ser resuelto en cualquier orden y anexando cualquier tipo de información que se considere conveniente para aclarar o complementar una respuesta.

Las cuestiones deberán ser manejadas por un especialista en la materia, aunque esto conduzca a que los encargados del proyecto desempeñen un papel meramente coordinador de actividades. Todas las cuestiones deben ser tenidas en cuenta, incluso en el caso de que se hayan tratado tácitamente en respuestas anteriores.

Contexto clínico

Las preguntas que tienen que ver con los servicios clínicos están incluidas en este cuestionario, dado que uno de sus objetivos es proporcionar un conocimiento global del sistema de telemedicina y sus características.

El cuestionario puede ser usado para ver el estado actual de los aspectos clínicos y evaluar qué mejoras deben ser hechas. Las preguntas no pretenden definir competencias clínicas y son por lo tanto abiertas y basadas en la familia de estándares de calidad ISO 9000. Estas características están en consonancia con el concepto de *administración con calidad total* (TQM, por sus siglas en inglés):

1. Definición de necesidades
2. Planificación para cubrir necesidades
3. Controles para cubrir los estándares
4. Mejora continua

Un gran número de factores deben ser considerados con respecto a los tópicos relacionados en esta sección.

Los estándares médicos deben definir el contexto en el cual ha de ser introducido el sistema de telemedicina, no solamente las necesidades que deben atenderse por dicho servicio. Es decir, la definición de estándares médicos y de un conjunto de directrices para asegurar la más alta calidad en la atención debe ser una actividad independiente del sistema de telemedicina que se implemente.

Cualquier servicio médico en la modalidad de telemedicina que se quiera implementar en un ambiente donde existe un conjunto aceptado y aplicado de estándares médicos y directrices, debe estar completamente estructurado para que sea compatible. Si no lo es, debe estar lo suficientemente documentado y la necesidad de su aplicación claramente dilucidada.

Es necesario tener en cuenta las siguientes definiciones:

- Evento de atención en salud: un contacto para el cuidado de la salud que requiere que un servicio clínico sea aplicado.
- Servicio clínico: uno o más planes de salud autorizados e implementados para poder llenar los requerimientos del evento de atención en salud.
- Plan de salud: uno o más procedimientos clínicos asociados a estándares médicos y directrices que se utilizan para poder cubrir un servicio médico.
- Procedimiento clínico: una o más actividades de tratamiento en salud.
- Directriz médica: un informe sistemáticamente desarrollado, el cual asiste en la toma de decisiones acerca de los procedimientos por seguir para una condición clínica específica.
- Estándar médico: un documento, establecido por consenso y aprobado por un cuerpo consultor organizado, el cual provee para el uso común y repetido de las pautas o las características de actividades de cuidados clínicos (o sus resultados).

Su objetivo principal es alcanzar el grado óptimo de orden en un contexto de medicina dado.

Unidades de recolección de información

1. ¿Cuáles son los servicios de salud por modalidad telemedicina que deben ser implementados?

Describe sus ideas/necesidades para el uso de la telemedicina en su ambiente de trabajo

- ¿Cuáles son los servicios clínicos que serán afectados o reemplazados por el nuevo sistema de telemedicina?
- ¿Cuáles organizaciones/departamentos clínicos directamente involucrados en el servicio?

2. ¿Cuáles son los requerimientos de los usuarios?

- ¿Quiénes son los usuarios?
- ¿Quién está involucrado?
- ¿Qué hacen ellos? Roles/actividades, usuarios cliente (pacientes), usuarios participantes (doctores, enfermeros, paramédicos, personal de soporte, etc.)
- ¿Qué método era usado para obtener los requerimientos clínicos de los usuarios?
Ejemplo: ¿Existe un grupo de usuarios, incluyendo un representante del paciente, donde se use un cuestionario o se realice una encuesta?
- ¿Cuáles son los eventos/episodios clínicos que serán dirigidos en el sistema de telemedicina?
- ¿Qué señales, incluyendo signos vitales, necesitarán ser monitorizadas?
- ¿Qué equipo es necesario para monitorizar dichas variables?
- ¿Qué procedimientos (incluyendo entrenamiento) son requeridos para implementar el monitoreo de las variables?
- ¿Quién interpretará la información monitorizada? (especificar el nivel profesional).
- ¿Qué factores se requieren para garantizar la seguridad del paciente, exactitud y funcionalidad de los equipos, y aceptación por parte del usuario en el monitoreo y otros procedimientos?

3. ¿Cuáles son los procedimientos de servicios clínicos que podrán verse afectados por la implementación de la telemedicina?

Ejemplo: el acceso al servicio, referencia, admisión, descarga y procedimientos de seguimiento.

- Describir los eventos y procedimientos existentes, además de las particularidades del ambiente local (específicas).

- Describir una aproximación al nuevo sistema de telemedicina (tomando en consideración condiciones y directrices locales).
 - Describir la estrategia con la cual se obtendrá la aprobación local y extendida de los cambios.
4. ¿Cuáles son los estándares/ pautas relacionadas con el sistema de telemedicina?
- Pautas existentes que deben ser conservadas en el sistema de telemedicina.
- Pautas que deben ser reemplazadas.
- ¿Existen nuevas directrices que deban ser creadas para implementar el sistema de telemedicina?
 - ¿Qué proceso de desarrollo de directrices debería ser usado?
5. ¿Cuáles son los recursos humanos requeridos por el sistema de telemedicina?
- ¿El sistema de telemedicina puede ser implementado y operado por el personal existente?
 - ¿Se debe entrenar nuevo personal?
6. ¿Cuáles son las necesidades de entrenamiento para el sistema de telemedicina?
- En telemática, telemedicina, procedimientos de calidad, análisis de requerimientos, indicadores de resultados y desempeño, etc.
 - Tipo de entrenamiento requerido.
 - Tipo y grado del personal.

Contexto de servicio/relación con el entorno

Es necesario definir una estrategia de desarrollo para poder garantizar que la solución en telemedicina encaje perfectamente en el ambiente existente de servicios en salud. Los servicios telemédicos usualmente se implementan en alguno de los siguientes escenarios: creación de nuevos servicios telemédicos o de un servicio de este tipo que reemplazará a un servicio diferente existente.

Unidades de recolección de información

- **Restricciones legales.**

La pregunta explora las restricciones legales, éticas y sociales sobre el servicio y aquellos que participan en él. Las restricciones éticas pueden estar enmarcadas en leyes, pero no siempre sucede; por ejemplo, los médicos tienen límites éticos promulgados por organismos internacionales, pero hay razones culturales que pueden flexibilizar o endurecer dichas directrices.

1. ¿Cuáles son las restricciones legales de los servicios telemédicos?
- Con respecto a las transmisiones de datos.
 - Con respecto a la protección de datos.

- Con respecto a la responsabilidad en los productos y el seguimiento de estándares.
 - Con respecto a la libertad de información/privacidad.
 - Con respecto a la responsabilidad personal y organizativa.
 - Con respecto a las entidades en las cuales se siguen responsabilidades legales.
 - ¿Qué organismos son responsables de la dirección legal?
2. ¿Cuáles son las consideraciones éticas?
- ¿A quién están dirigidas las restricciones éticas?
 - ¿Con respecto a qué se definen dichas restricciones?
 - ¿Qué organismos son los encargados de la dirección en cuanto a cuestiones éticas?
3. ¿Cuáles son los factores culturales y sociales que han de ser considerados?
- Consideraciones relacionadas con la raza.
 - Consideraciones relacionadas con creencias culturales.
 - Consideraciones relacionadas con el nivel educativo de los usuarios.
 - Consideraciones relacionadas con el grupo socioeconómico al cual pertenece el usuario.
 - Consideraciones relacionados con la ubicación geográfica de los usuarios.
 - Consideraciones relacionadas con la edad de los usuarios.
 - Consecuencias organizativas del sistema de telemedicina.

La siguiente pregunta tiene que ver con los efectos que tendrá la introducción de un servicio de salud por modalidad de telemedicina en la organización.

1. ¿Qué consecuencias tendrá en su organización la introducción del sistema de telemedicina?
- Sobre la práctica actual del trabajo.
 - Sobre los estándares locales.
 - Sobre las inversiones existentes.
2. ¿Qué características técnicas u organizativas requieren adaptación?
- Práctica actual del trabajo.
 - Estándares locales.
 - Inversiones existentes.
 - Políticas administrativas.
 - Maquinaria consultiva.

3. ¿Qué organizaciones/contratistas deben interactuar para proveer el servicio de salud por modalidad de telemedicina?
- Organizaciones gubernamentales
 - Administración regional de salud
 - Organizaciones públicas
 - Organizaciones profesionales
 - Organizaciones voluntarias (Cruz Roja, Defensa Civil, etc.)
 - Compañías de telecomunicaciones (proveedores de servicios de interconexión)
 - Compañías de seguros
 - Otras compañías

Consideraciones tecnológicas

Esta sección se relaciona con la definición de los recursos técnicos y de infraestructura necesarios para la provisión del sistema de telemedicina.

Unidades de recolección de información

- Requerimientos de hardware
 - ¿Cuáles plataformas de hardware son requeridas? (tanto para proveer el servicio de salud en la modalidad de telemedicina como para su utilización).
Por ejemplo: PC, UNIX.
 - ¿Qué tipo de almacenamiento se requiere?
De acuerdo con la cantidad de información.
De acuerdo con las políticas de seguridad.
De acuerdo con la velocidad de búsqueda de la información.
 - ¿Qué tecnología de visualización se requiere?
 - ¿Cuáles son los requerimientos de procesamiento y cómo puede ser medido?
Por ejemplo: velocidad, procesamiento en paralelo, medición en MFlops, Mips, Whetstones, Dhrystones u otras medidas estándar.
 - ¿Qué requerimientos especiales deben tener los equipos del usuario final?
Por ejemplo: unidades móviles u otros dispositivos de usuario específicos (para captura de información, análisis, etc.).
 - ¿Qué puertos de comunicaciones deberán estar disponibles?
Por ejemplo: Ethernet, V32, SCSI, RS 232, USB, conectores especiales a otros equipos.

- ¿Qué tecnologías de adquisición, entrada o salida de datos son requeridas?
Por ejemplo: dispositivos de monitoría, módems, impresores, OCR, etc.
- ¿Existen en su institución dispositivos que cumplan los requerimientos?

Software

- ¿Cuáles sistemas operativos se requieren?
 - ¿Qué herramientas de desarrollo son requeridas para desarrollar aplicaciones específicas para el sistema de telemedicina?
Por ejemplo: lenguajes de programación, compiladores, enlazadores, depuradores, editores, herramientas CASE.
1. ¿Qué infraestructura de comunicaciones se requiere? Por ejemplo: hardware de red, software de red.
 2. ¿Qué enlaces de comunicaciones se requieren? Por ejemplo: hardware (módems, bridges, switch, router, etc.)
 - ¿Qué formatos?
 - ¿Qué velocidad en los tiempos de respuesta?
 - ¿Qué estándares/protocolos?
 - ¿Qué ancho de banda?
 3. Software aplicativo
 - ¿Cuáles son los requerimientos para KBS, DSS?
 - ¿Cuáles son los requerimientos para los sistemas de bases de datos?
 - ¿Cuáles son los requerimientos para otro tipo de software aplicativo? Por ejemplo: ¿existe software comercial, GNU, GPL que cubra las necesidades?
 4. Requerimientos en infraestructura
 - ¿Qué mobiliario se requiere?
 - ¿Qué requerimientos eléctricos deben ser cumplidos?
Voltaje, corrientes, regulación, aislamiento, sistemas de tierra, etc.
 - ¿Qué temperatura, iluminación o humedad se requieren?
 - ¿Qué cambios deben realizarse a los sitios existentes? Por ejemplo: cableado, requerimientos de espacio, etc.
 5. ¿Cuáles metodologías están implementadas (o deben ser adoptadas) para cumplir los requerimientos tecnológicos y de infraestructura del servicio?

Consideraciones de calidad

Esta sección contiene preguntas que deben ser contestadas para garantizar un servicio que tome en cuenta los factores de calidad en todas las etapas del proyecto.

Unidades de recolección de información

1. Aseguramiento de la calidad.

- ¿Cuáles son las consideraciones de calidad que tener en cuenta en el sistema telemédico?

Por ejemplo: mejoramiento de los mecanismos de cuidados para el paciente, mejoramiento de la eficiencia, reducción de costos, etc.

- ¿Cuáles métodos de administración con calidad total (TQM) existen o deben ser implementados?
- ¿Quién es responsable de la supervisión de los servicios médicos/de enfermería, servicios técnicos, servicios administrativos y cuáles son sus roles en el servicio telemédico?

- ¿Cómo podría ser validado el sistema de telemedicina?

Definir factores, variables, criterios e indicadores.

- ¿Existen manuales y políticas de calidad implementados para el sistema de telemedicina?

- ¿Cómo podría obtenerse aprobación de la autoridades?

Por ejemplo: legislación local, nacional o certificación ISO.

2. Grupo de usuarios

- ¿Está o estará un grupo de usuarios establecido para el sistema de telemedicina?
- ¿Está o estará disponible un manual de usuario para el sistema de telemedicina?
- ¿Existirán enlaces entre los diversos grupos de usuarios nacionales/mundiales de servicios de salud similares a los brindados por la modalidad de telemedicina?

3. Describa las estrategias que se seguirán para dar conocimiento al público del nuevo servicio telemédico.

- Estrategias para promocionar los objetivos del servicio y la forma de acceder a él.
- Estrategias para el material educativo que será usado en el servicio.
- Estrategia para la mejora, promoción de resultados e impacto del servicio.

4. ¿Cómo podrían evaluarse las variables de calidad?

5. ¿Cuáles son los valores aceptables para las variables de calidad seleccionadas?

6. ¿Quién es el responsable de definir las estrategias de calidad?

7. ¿Qué tipo de documentación y procedimientos son usados (o se necesitan) para asegurar que la calidad total esté implementada?

8. ¿Qué criterios pueden ser utilizados para demostrar que la calidad en el servicio es máxima?

Aceptación del servicio de salud brindado en la modalidad de telemedicina

Unidades de recolección de información

1. ¿Qué tipos de intereses (beneficios) son inherentes al establecimiento de un sistema de telemedicina y qué variables deben ser consideradas?

Tipos de interés cuyos valores deben ser considerados:

- Clínicos: doctores, enfermeras, etc.
 - Tecnológicos: técnicos, investigadores, doctores, desarrolladores, etc.
 - Éticos
 - Económicos
 - Industriales: empresas, institutos de investigación/universidades, sociedad en general
2. ¿Qué hace que el sistema de telemedicina sea aceptado por las personas involucradas en el proyecto?
 3. ¿Cuál es el costo proyectado para establecer el sistema de telemedicina?
 - Costos capitales tangibles: equipos, instalación/pruebas, cambios estructurales (edificios/cuartos), cambios en la organización (procedimientos), depreciación.
 - Costos operativos: personal, servicio de comunicaciones, mantenimiento y servicio, control de calidad, capacitación de personal, jornadas de socialización.
 4. ¿Quién debería pagar por el establecimiento/implementación del sistema de telemedicina?

Entre otros puede estar:

- Hospital, centros atención de salud
 - Universidades
 - Empresas de telecomunicaciones
 - Fundaciones de investigación
 - Proveedores de servicios de salud bajo la modalidad de telemedicina, compañías de aseguramiento
5. ¿Cómo debe ser evaluado el sistema de telemedicina?
 - Evaluación económica
 - Evaluación de impacto social
 - Evaluación de impacto médico
 - Evaluación clínica
 - Evaluación a las comunidades de práctica

6. ¿Qué técnicas de evaluación están disponibles?

- Investigación evaluativa
- Evaluación integral
- Etnografía
- Análisis de costos
- Análisis costo/beneficio
- Ingeniería económica

Aspectos de construcción y persistencia del sistema de telemedicina

Unidades de recolección de información

1. Tiempo de vida del proyecto

- ¿Cuál es el tiempo de vida proyectado para el sistema de telemedicina?
- ¿Qué etapas pueden identificadas en dicho tiempo de vida?
- ¿Cuánto tiempo tardaría en implementarse el servicio en una nueva ubicación?
- ¿Cuándo y cuáles mejoras pueden ser anticipadas/planeadas?

2. ¿Qué recursos son o deben estar disponibles para establecer el sistema de telemedicina?

- Recursos de información (manuales, educación, soporte)
- Recursos financieros
- Personal (habilidades, incentivos, educación, entrenamiento)
- Espaciales (arquitectónicos)

3. ¿Qué recursos deben ser reevaluados?

- Financieros
- De personal
- Espaciales
- De cultura en la organización
- De mecanismos de interacción con el entorno

4. Disponibilidad de los recursos para la operación del sistema de telemedicina

- ¿En qué sitios?
- ¿En qué tiempo?
- ¿Quién los suple?
- ¿A quién o qué va dirigido?

Políticas de la organización

Esta sección se relaciona con la formulación de políticas administrativas con respecto a la prestación de servicios telemédicos.

Las políticas se refieren a los objetivos de la compañía, prácticas estandarizadas, regulaciones, estatutos, manuales de procedimientos, manuales de seguridad, códigos de ética, etc.

1. ¿Qué políticas/regulaciones administrativas pueden tener efecto sobre el sistema de telemedicina?
 - Con relación a la disponibilidad y distribución de recursos
 - Con relación a las estructuras de la organización
 - Con relación a las personas que son afectadas
 - Con relación a la calidad (norma ISO 9000 o similares)
2. ¿Qué implicaciones pueden tener para el sistema de telemedicina?
 - Con relación con la disponibilidad y distribución de recursos
 - Con relación a las estructuras de la organización
 - Con relación a las personas que son afectadas
3. ¿Cómo deben definirse las políticas de seguridad, manejo de información y calidad para el sistema de telemedicina?
 - Con respecto a políticas administrativas
 - Con respecto a la disponibilidad de fuentes de información
 - Con respecto a la libertad de uso de la información, a la protección de datos y la confidencialidad
 - Con respecto a la legislación
 - Con respecto a consideraciones éticas y sociales
 - Con respecto a la distribución de privilegios en el acceso al sistema de telemedicina
 - Con respecto a las políticas de calidad definidas

Consideraciones acerca de la información

Esta sección cubre la información que ha de ser parte del sistema de telemedicina, incluye cuestiones sobre flujo de datos, estructura de datos y archivos y almacenamiento de datos, entre otros.

Las necesidades de información del sistema y el procesamiento que se haga de ella tienen una gran implicación al elegir la tecnología que debe ser utilizada.

Unidades de recolección de información

1. ¿Qué tipos de información son necesarios para el sistema telemédico?
 - Texto, numérico, imagen, video
 - Imágenes de rayos X, MRI
2. Formatos de archivos y estándares de codificación
 - ¿Cuáles estándares de codificación deberán ser usados para tipos genéricos de datos?
Por ejemplo: mpeg, jpeg, mheg, IA5, etc.
 - ¿Qué estándares de codificación médica deberán ser usados para los elementos de información?

Por ejemplo: códigos READ, ICD9, ICD10, WHO

3. ¿En qué actividades está involucrada la información? ¿Cómo y dónde está siendo utilizada?
4. Calidad de la información
 - ¿Con qué calidad la información debe ser utilizada en las diferentes etapas del servicio (recolección, procesamiento, transmisión y visualización)?
 - ¿Qué tan relevante es la calidad en esas etapas para la exactitud de la conclusión final?
 - ¿Cómo puede ser maximizado el nivel de exactitud del diagnóstico final?
5. ¿Qué procedimientos de control de calidad en la información son necesarios?
6. ¿Cómo es el flujo de la información en el sistema de telemedicina (dentro de las fronteras del sistema)?
7. ¿Cuál es la información que fluye desde y hacia el sistema de telemedicina?
8. ¿Qué pasos deben seguirse para garantizar la seguridad de los datos?

Procesamiento de información

Unidades de recolección de información

1. ¿Qué conversiones en la información son (o deberían ser hechas) antes de su presentación? (si se transmiten los datos originales, entonces ninguna conversión es necesaria)

Por ejemplo: decomprensión de imágenes, decodificación, etc.

2. ¿Qué procesamiento de señal se requiere?
3. ¿Cuáles son las implicaciones de tales transformaciones/procesamientos?
 - Con respecto a responsabilidades legales
 - Con respecto a consideraciones de calidad

4. ¿Cómo deberá ser presentada la información al usuario?
 - Requerimientos en interfaz de usuario

Apéndice H

Otros entregables nucleares de OpenSITEM

Visión-resumen ejecutivo

Propósito inicial

OpenSITEM es un *portal web* especializado en la gestión de datos e información de diferentes componentes estructurales de los sistemas de telemedicina. Provee un ambiente de apoyo a las tareas de las comunidades de práctica involucradas en la investigación, el diseño, mantenimiento, desarrollo e implementación de redes de telemedicina. Tuvo su génesis conceptual en el año 2000, en la primera fase del Proyecto Telemedicina Bogotá, como solución a la necesidad de administrar los resultados del estudio de campo realizado a las entidades e instituciones de salud y los operadores de telecomunicaciones en la ciudad de Bogotá.

Su principal objetivo es apoyar las actividades básicas de los denominados *trabajadores del conocimiento* en el área de la telemedicina, ofreciéndoles, además de un repositorio de datos, herramientas que facilitan las tareas de capturar, extraer, organizar, analizar, encontrar, sintetizar, distribuir y compartir información y conocimiento. El ideal es actualizar el estado de ciertos nodos interesantes del Sistema de Salud de Bogotá Distrito Capital, haciendo énfasis en la posibilidad de interacción a escala nacional e internacional y en los requerimientos que en telemedicina tengan las diferentes entidades que participan o no en el proyecto de telemedicina auspiciado por el grupo GITEM.

Alcance inicial

OpenSITEM es principalmente *un concepto*, su estado actual es una representación del potencial real del sistema que debe ser socializado y entregado a la comunidad. La base de desarrollo principal es el grupo GITEM y será responsable de la versión oficial del producto. Sin embargo, dada la dinámica en el mundo del software libre, el grupo GITEM no limitará el trabajo independiente que sobre su desarrollo realice cualquier persona o grupo de personas. En este sentido, la funcionalidad original del sistema podrá ser modificada pero no avalada directamente por el grupo³².

32 Salvo en casos en que no se trasgreden directamente los objetivos primarios del desarrollo. En tales casos las contribuciones serán asociadas al hilo oficial de desarrollo.

OpenSITEM ha sido creado con el fin de apoyar a los grupos de trabajo que realizan labores en el área de proyección de sistemas de telemedicina. La información que en él se encuentra debe ser ingresada por personas autorizadas para asegurar en un alto grado su veracidad e idoneidad. Sin embargo, no se puede garantizar —y no se garantiza— la exactitud, disponibilidad, integridad y oportunidad de dicha información: LA INFORMACIÓN CONTENIDA EN OpenSITEM NO ES UNA FUENTE OFICIAL DE DATOS. Su uso es responsabilidad de quien lo realiza. La información que se encuentre en OpenSITEM no necesariamente ha sido revisada por expertos profesionales. Todos los contenidos que se ingresen a OpenSITEM deben ser de licencia pública o de libre uso; los que no cumplan estos criterios serán eliminados.

Posicionamiento

- Definición del problema

La mayoría de los estudios base de conocimiento se encuentran disgregados y en idiomas diferentes al español, por lo cual su consulta es compleja, y no existe un mapa seguro de navegación que guíe al investigador hacia las fuentes confiables de información.

- Afecta a

Investigadores, consultores, usuarios y proveedores de servicios en el área de la salud.

- El impacto asociado es

Estudios abandonados e inconclusos, junto con la complejidad innecesaria del proceso de determinación del estado del arte, están abocando a los grupos universitarios a competir codo a codo —a pesar de todas sus limitaciones— con grandes empresas multinacionales interesadas en “sacar del camino” a estos facilitadores de procesos.

- Una solución parcial adecuada sería

Un sistema informático que en un ambiente integrado ofrezca posibilidades a los usuarios para la administración de información sobre varios componentes tecnológicos de las redes de telemedicina, así como la posibilidad de realizar seguimiento al cumplimiento de ciertos indicadores en los proyectos de telemedicina.

Un sistema que sea fácilmente adaptable a las necesidades novedosas y que esté basado en software libre para concentrar la inversión en su desarrollo y no en el pago de licencias de uso o de compra de herramientas de programación.

- Para

Investigadores, estudiantes, usuarios, prestadores de servicios de salud, prestadores de servicios de telecomunicaciones, programadores.

- **Quiénes**

Son los beneficiarios directos del despliegue de servicios médicos por la modalidad de telemedicina.

- **Nuestro producto**

Sistema de Información para el Apoyo de Grupos de Trabajo en Proyectos de Telemedicina (OpenSITEM).

Disminuye el tiempo de adquisición, análisis y despliegue de la información. Es construido siguiendo adaptaciones de procesos de desarrollo ampliamente conocidos y de acuerdo con el paradigma de la orientación a objetos, con lo que se garantiza su facilidad de mantenimiento, escalabilidad e indirectamente su permanencia en el medio.

Contiene módulos para la generación de estadísticas e informes pormenorizados de cada uno de los componentes y logra obtener en unos pocos segundos los datos necesarios para apoyar la labor de análisis, diseño e implementación de proyectos telemédicos o de telesalud. Usa un esquema modular de crecimiento a la medida en donde el esfuerzo para la creación de instrumentos nuevos de consulta se minimiza por el uso de plantillas prediseñadas. En lugar de ser un sistema estático, OpenSITEM contiene características de adaptación dinámica para cubrir las necesidades que tengan los próximos proyectos emanados del GITEM y otras entidades que hagan uso del sistema.

Participantes en el proyecto y usuarios

Perfil de los participantes de OpenSITEM. En nuestro desarrollo nos unimos al manifiesto de los metodólogos ágiles, manteniendo ciertas pautas del proceso unificado para poder dar fe de la calidad en el proceso y el producto.

Tabla H1
Perfil de los participantes del SITEM

Nombre	Descripción	Responsabilidades
Director de proyecto	Directora Grupo GITEM	Garantiza el flujo de recursos para el desarrollo del proyecto.
		Seguimiento del desarrollo del proyecto.
		Aprueba requisitos y funcionalidades generales.

Nombre	Descripción	Responsabilidades
Arquitectos del Sistema	Se encarga de la definición, modelado del problema - arquitectura del Sistema Solución. Engloba las funciones de los antiguos analistas, diseñadores e ingenieros de proceso.	Caso de desarrollo aplicando en parte el proceso unificado.
		Determinar las necesidades de los usuarios del sistema.
		Generar los niveles más altos de requerimientos del sistema.
		Asegurar los criterios de consistencia, pertinencia y completitud del modelo de requisitos.
		Particionar OpenSITEM en subsistemas y componentes.
		Generar artefactos del modelo de requisitos, análisis y diseño.
Ingenieros de prueba	Se encargan de desplegar los casos de prueba para garantizar que los ejecutables cumplen con los requisitos de los usuarios.	Diseñar casos de prueba.
		Realizar pruebas.
		Proponer modificaciones.
		Depurar componentes.
Programador	En OpenSITEM representa el integrante de mayor jerarquía dentro del proceso de desarrollo. Engloba las funciones asociadas a los demás participantes.	Desarrollar componentes.

Fuente: elaboración propia.

Entorno de usuario

El usuario opera una interfaz web a través de un navegador HTTP, con soporte para HTML 5.0, XML 2.0, javascript, XSL y Cascada Style Sheet 2.0.

Para acceder a las diferentes secciones de OpenSITEM se requiere que el usuario ejecute un proceso de autenticación, autorización y registro (AAA) asociado a una sesión. Hasta la versión 3.0 se mantendrá un entorno gráfico básico centrado en hipervínculos y con diseño gráfico mínimo.

Suposiciones y dependencias

- La plataforma tecnológica sobre la que se implementa el módulo tiene una disponibilidad superior al 99% del tiempo.
- Las herramientas de desarrollo son software libre.
- OpenSITEM podrá integrar en su arquitectura otras aplicaciones de software libre o público.
- El hilo principal de desarrollo estará en la Universidad Distrital pero no se restringirá la distribución del producto a usuarios interesados.
- El grupo de participantes en OpenSITEM es indefinido. Los procesos se potenciarán en la medida que se produzcan “explosiones” de desarrollo fomentadas por usuarios interesados.

Descripción global de OpenSITEM

OpenSITEM es implementado sobre una arquitectura multicapa que distribuye los diferentes componentes en tres capas principales: presentación, aplicación y datos, y está presente una capa transversal tácita de seguridad. A nivel de usuario, OpenSITEM está compuesto por siete subsistemas autónomos que prestan servicios a sus pares. Estos agrupan seis componentes clave en todo proyecto de telemedicina: entidades de salud, operadores de telecomunicaciones, tecnologías de interconexión, equipos y tecnologías de captura de datos, proyectos e instituciones relacionadas con la telemedicina y servicios médicos, incluyendo módulos de vademécum, consulta de procedimientos, enfermedades y especialidades médicas.

Otros requisitos del producto

Estándares aplicables

- Unified Process
- Unified Modeling Language versión 2.0
- Extensible Markup Language Versión
- SOAP
- OWL

El sistema debe ser:

- Multiplataforma
- Multiusuario

Requisitos de desempeño

- Velocidad de acceso promedio interior a 10 s
- Ayudas contextuales y contenidos autoexplicativos
- Disponibilidad superior al 99 %
- Manejo de conexiones concurrentes
- Integridad referencial en la capa de persistencia

Lineamientos de codificación para la organización de los módulos

Para asegurar una codificación eficiente, que permita realizar búsquedas rápidas dentro de la organización documental, se tienen las siguientes reglas de obligatorio cumplimiento en todos los artefactos:

El nombre del artefacto deberá estar antecedido de un identificador del tipo:

aaa-bbb-ccc inicialesmódulo-tipoartefacto-versiónartefacto.

Así, para la primera versión del documento de especificaciones de casos de uso, del módulo de administración de instrumentos para la recolección de información ha de tenerse una codificación similar a:

MAI-ECU-001

Siendo MAI y ECU los identificadores únicos tomados del artefacto códigos.

Apéndice I

Análisis de necesidad de un bus de servicio empresarial

Para determinar si el proyecto se beneficiaba de un bus de servicio empresarial, se respondió la lista de chequeo propuesta por Ross Mason del proyecto Mule:

Capacidad de integración

¿Está integrando tres (3) o más aplicaciones o servicios?

Respuesta: SÍ.

¿Tiene más de diez (10) aplicaciones para integrar?

Respuesta: NO

Protocolo de comunicación

¿Requiere usar más de un protocolo de comunicación?

Respuesta: NO.

Capacidades especiales de mensajería

¿Necesita capacidad para el enrutamiento de mensajes tales como bifurcación o agrupación de flujos de mensajes o enrutamiento basado en el contenido?

Respuesta: NO.

Orientación a servicios

¿Necesita publicar servicios que deban ser consumidos por otras aplicaciones?

Respuesta: SÍ

Escalabilidad vs. simplificación

¿Realmente necesita la escalabilidad provista por un ESB?

Respuesta: NO DETERMINADO.

¿Conoce el alcance de su aplicación?, ¿puede iniciar sin requerir un ESB?

Respuesta: Sí.

Apéndice J

Aspectos relativos al despliegue

El despliegue de la solución requiere una plataforma tecnológica adecuada, tal como se muestra en el apéndice B, para garantizar un óptimo servicio a los potenciales usuarios³³. Las herramientas básicas se encuentran disponibles para que el grupo de investigación convoque a las entidades de salud, profesionales en el área de la tecnología, especialistas en medicina y fabricantes —distribuidores— de dispositivos médicos que participaron en el Estudio Red de Telemedicina Bogotá, para que de forma conjunta enriquezcan la base de información en el subproceso de prueba piloto (presupuestada como fase IV).

Con base en el módulo de generación de herramientas para la recolección de información, apéndice C, el grupo aplicará diferentes instrumentos entre los que se incluyen:

- Entrevistas: con el objeto de acordar las directrices por seguir, tanto con los operadores de telecomunicaciones como con los prestadores del servicio de salud que participaron de la fase I de recolección de información, con el propósito de socializar los resultados del proyecto e invitarlos para que editen y actualicen sus datos y obtengan beneficios incrementalmente. Dichos servicios van desde la disponibilidad de un mapa de sedes, servicios y profesionales hasta la consolidación de información para la gestión de sus redes tecnológicas primarias.
- Encuestas: para identificar nuevos requerimientos de servicios que puedan ser desplegados en la plataforma propuesta. Además, estos instrumentos permitirán medir el impacto en la cobertura de los servicios y de participación de las instituciones objeto de la investigación de campo.

Las unidades de recolección de información mostradas en el apéndice G se han adaptado de aquellas propuestas por el proyecto europeo Hermes³⁴. El modelo de

33 La versión actual está desplegada en servidores de la intranet universitaria.

34 Proyecto de investigación a tres años, financiado por la Comunidad Económica Europea y que cumplió sus objetivos hacia principios del milenio dejando como resultado un conjunto de preguntas básicas que apoyan los procesos de implementación de soluciones médicas apoyadas en las TIC.

investigación evaluativa que continúa en la fase IV del proyecto pretende medir la evolución del nivel de servicios de salud prestados con el apoyo de TIC, comparando sucesivamente el modo de operación encontrado entre el año 2000 y el 2005 con aquel encontrado entre el año 2008 y el 2011, luego de que algunas entidades hayan interactuado con OpenSITEM.

Fuentes de información primaria

Para la carga de información inicial en OpenSITEM se utilizan los resultados del estudio de campo realizado en varias instituciones de carácter público y privado. Dichos resultados se encuentran consignados en tesis en formato digital e impresos disponibles en la biblioteca de la Universidad Distrital y en el archivo del grupo de investigación:

- Hospital Rafael Uribe
- Hospital San Pedro Claver
- Hospital Simón Bolívar
- Hospital El Tunal
- Hospital La Victoria
- Hospital San José

Apéndice K

Entrevista base para identificación de interesados

En la elaboración y validación de este instrumento de recolección de la información se siguió el método definido en la sección de *Metodología para la estructuración y validación del documento*, presentado en el capítulo de Estructura del proyecto. Es necesario recalcar que el formato de entrevista es semiestructurado y el entrevistador tendrá las preguntas solo como referencia.

Metodología de procesamiento

El instrumento debe aplicarse en la plataforma de gestión de encuestas. Cada una de las respuestas se analiza de manera separada y se califican los atributos de claridad, pertinencia, independencia (que no tenga similitud con respuestas dadas por otro entrevistado), impacto y completitud.

Las preguntas que tengan mayores índices se procesarán primero y los usuarios que las respondieron se clasificarán como interesados del proyecto. Cada una de las preguntas viene acompañada de su categoría para facilitar la interpretación.

Instrumento

Pregunta

Categoría: hechos. Objetivo: identificar si el entrevistado posee información específica. Enunciado: ¿cuáles son los procedimientos que desarrolla en el interior del proyecto?

Pregunta

Categoría: hechos. Enunciado: ¿qué datos produce al desarrollar su trabajo?

Pregunta

Categoría: opinión. Objetivo: determinar la percepción del entrevistado respecto a un aspecto concreto. Comentario: se utiliza para obtener un criterio base para análisis multitemporal. Enunciado: ¿cuáles son los procedimientos que desarrolla en el interior del proyecto?

Pregunta

Categoría: creencia. Objetivo: adquirir un conocimiento de la creencia, prejuicio o concepto que tiene el entrevistado respecto a un tema. Comentario: se utiliza como insumo para analizar posibles obstáculos o fortalezas que pueda tener una creencia en el desarrollo del proyecto. Enunciado: ¿cree que las tecnologías de información pueden contribuir a mejorar la productividad del grupo?

Pregunta

Categoría: opinión. Enunciado: ¿las herramientas software que apoyan su trabajo dentro del grupo son adecuadas?

Pregunta

Categoría: opinión. Enunciado: ¿las herramientas software que apoyan su trabajo dentro del grupo son suficientes?

Pregunta

Categoría: intención. Objetivo: examinar los mecanismos de optimización de tareas que utilizaría el entrevistado. Enunciado: ¿cómo desarrollaría su trabajo si existiera una herramienta adecuada?

Anexo A

GNU General Public License

Licencia Pública General Versión 2, Junio de 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Traducción no oficial

Toda persona tiene permiso de copiar y distribuir copias fieles de este documento de licencia, pero no se permite hacer modificaciones.

Preámbulo

Los contratos de licencia de la mayor parte del software están diseñados para quitarle su libertad de compartir y modificar dicho software. En contraste, la GNU General Public License pretende garantizar su libertad de compartir y modificar el software, esto para asegurar que el software es libre para todos sus usuarios.

Esta licencia pública general se aplica a la mayoría del software de la FSF Free Software Foundation (Fundación para el Software Libre) y a cualquier otro programa de software cuyos autores así lo establezcan. Algunos otros programas de software de la Free Software Foundation están cubiertos por la LGPL Library General Public License (Licencia Pública General para Librerías), la cual puede aplicar a sus programas también.

Cuando hablamos de software libre, nos referimos a libertad, no a precio. Esta licencia está diseñada para asegurar que:

1. Usted tiene la libertad de distribuir copias del software libre (y cobrar por ese sencillo servicio si así lo desea).
2. Recibir el código fuente (o tener la posibilidad de obtenerlo si así lo desea).
3. Usted puede modificar el software o utilizar partes de este en nuevos programas de software libre.
4. Usted esté enterado de que tiene la posibilidad de hacer todas estas cosas.

Para proteger sus derechos, necesitamos hacer restricciones que prohíban a cualquiera denegarle estos derechos o a pedirle que renuncie a ellos. Estas restricciones se traducen en algunas responsabilidades para usted si distribuye copias del software, o si lo modifica.

Por ejemplo, si usted distribuye copias de un programa, ya sea gratuitamente o por algún importe, debe dar al que recibe el software todos los derechos que usted tiene sobre este. Debe asegurarse también de que reciba el código fuente, o bien que pueda obtenerlo si lo desea. Y, por último, debe mostrarle a esa persona estos términos para que conozca los derechos de que goza.

Nosotros protegemos sus derechos en dos pasos: (1) protegiendo los derechos de autor del software y (2) ofreciéndole este contrato de licencia que le otorga permiso legal para copiar, distribuir y modificar el software.

Además, para la protección de los autores de software y la nuestra, queremos asegurarnos de que toda persona entienda que *no existe ninguna garantía del software libre*. Si el software es modificado por alguien y lo distribuye, queremos que quienes lo reciban sepan que la copia que obtuvieron no es la original, por lo que cualquier problema provocado por quien realizó la modificación no afectará la reputación del autor original.

Finalmente, cualquier programa de software libre es constantemente amenazado por las patentes de software. Deseamos evadir el peligro de que los redistribuidores de un programa de software libre obtengan individualmente los derechos de patente con el fin de volver dicho programa propietario. Para prevenir esto, hemos dejado en claro que cualquier patente deberá ser licenciada para el uso libre de toda persona o que no esté licenciada del todo.

A continuación se describen con precisión los términos y condiciones para copiar, distribuir y modificar el software.

Términos y condiciones para copia, modificación y distribución

Esta licencia aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta GPL. El “Programa”, utilizado en lo subsecuente, se refiere a cualquier programa o trabajo original, y el “trabajo basado en el Programa” significa ya sea el Programa o cualquier trabajo derivado de este bajo la ley de derechos de autor: es decir, un trabajo que contenga el Programa o alguna porción de este, ya sea íntegra o con modificaciones o traducciones a otros idiomas. De aquí en adelante “traducción” estará incluida (pero no limitada a) en el término “modificación”, y la persona a la que se aplique esta licencia será llamado “usted”.

Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance. El acto de ejecutar el programa no está restringido, y la salida de información del programa está cubierta solo si su contenido constituye un trabajo basado en el Programa (es independiente de si fue resultado de ejecutar el Programa). Si esto es cierto o no depende de la función del Programa.

1. Usted puede copiar y distribuir copias fieles del código fuente del Programa tal como lo recibió, en cualquier medio, siempre que proporcione de manera consciente y apropiada una nota de derechos de autor y una declaración de no garantía, además de mantener intactas todas las notas que se refieran a esta licencia y a la ausencia de garantía, y que le proporcione a las demás personas que reciban el programa una copia de esta licencia junto con el Programa. Usted puede aplicar un cargo por el acto físico de transferir una copia, y ofrecer protección de garantía por una cuota, lo cual no compromete a que el autor original del Programa responda por tal efecto.
2. Usted puede modificar su copia del Programa o de cualquier parte de él, formando así un trabajo basado en el Programa, y copiar y distribuir tales modificaciones o bien trabajar bajo los términos de la sección 1 arriba descrita, siempre que cumpla con las siguientes condiciones:
 - A. Usted debe incluir en los archivos modificados notas declarando que modificó dichos archivos y la fecha de los cambios.
 - B. Usted debe notificar que ese trabajo que distribuye contiene totalmente o en partes al Programa, y que debe ser licenciado como un conjunto sin cargo alguno a cualquier otra persona que reciba sus modificaciones bajo los términos de esta Licencia.
 - C. Si el programa modificado lee normalmente comandos interactivamente cuando es ejecutado, usted debe presentar un aviso, cuando el programa inicie su ejecución en ese modo interactivo de la forma más ordinaria, que contenga una noticia de derechos de autor y un aviso de que no existe garantía alguna (o que sí existe si es que usted la proporciona) y que los usuarios pueden redistribuir el programa bajo esas condiciones, e informando al usuario como puede ver una copia de esta licencia (excepción: si el Programa en sí es interactivo pero normalmente no muestra notas, su trabajo basado en el Programa no tiene la obligación de mostrar tales notas).

Estos requerimientos aplican al trabajo modificado como un todo. Si existen secciones identificables de tal trabajo que no son derivadas del Programa original, y pueden ser razonablemente consideradas trabajos separados e independientes como tal, entonces esta licencia y sus términos no aplican a dichas secciones cuando usted las distribuye como trabajos separados. Pero cuando usted distribuye las mismas secciones como parte de un todo que es un trabajo basado en el Programa, la distribución del conjunto debe ser bajo los términos de esta licencia, cuyos permisos para otras personas que obtengan el software se extienden para todo el software, así como para cada parte de él, independientemente de quién lo escribió.

No es la intención de esta sección reclamar derechos o pelear sus derechos sobre trabajos hechos enteramente por usted, en lugar de eso, la intención es ejercer el derecho de controlar la distribución de los trabajos derivados o colectivos basados en el Programa. Adicionalmente, el simple agregado de otro trabajo NO basado en el Programa al Programa en cuestión (o a un trabajo basado en el Programa) en algún medio de almacenamiento no pone el otro trabajo bajo el alcance de esta licencia.

3. Usted puede copiar y distribuir el Programa (o un trabajo basado en él, bajo la Sección 2) en código objeto o en forma de ejecutable, bajo los términos de las secciones 1 y 2 arriba descritas siempre que cumpla los siguientes requisitos:
 - A. Acompañarlo con el correspondiente código fuente legible por la máquina, que debe ser distribuido bajo los términos de las secciones 1 y 2 y en un medio comúnmente utilizado para el intercambio de software, o
 - B. Acompañarlo con una oferta escrita, válida por al menos tres años y para cualquier persona, por un cargo no mayor al costo que conlleve la distribución física del código fuente correspondiente en un medio comúnmente utilizado para el intercambio de software, o
 - C. Acompañarlo con la información que usted recibió sobre la oferta de distribución del código fuente correspondiente (esta alternativa está permitida solo para distribución no comercial y únicamente si usted recibió el Programa en código objeto o en forma de ejecutable con tal oferta de acuerdo a la subsección b anterior).

El código fuente de un trabajo significa la forma preferida de hacer modificaciones a este. Para un trabajo ejecutable, un código fuente completo significa todo el código fuente de todos los módulos que contiene, más cualquier archivo de definición de interfases, más los programas utilizados para controlar la compilación y la instalación del ejecutable.

Sin embargo, como excepción especial, no se requiere que el código fuente distribuido incluya cualquier cosa que no sea normalmente distribuida con las componentes mayores (compilador, kernel, etc.) del sistema operativo en el cual el ejecutable corre, a menos de que una componente en particular acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace ofreciendo acceso a copiar desde un lugar designado, entonces el ofrecer acceso equivalente para copiar el código fuente desde el mismo lugar se considera distribución del código fuente, aunque las demás personas no copien el código fuente junto con el código objeto.

4. Usted no puede copiar, modificar, sublicenciar ni distribuir el Programa a menos que sea expresamente bajo esta licencia. De otra forma, cualquier intento de copiar, modificar, sublicenciar o distribuir el programa es nulo y automáticamente causará la pérdida de sus derechos bajo esta licencia. Sin embargo, cualquier persona que haya recibido copias o derechos de usted bajo esta licencia no verá terminada su licencia ni sus derechos perdidos mientras continúe cumpliendo los términos de esta licencia.
5. Usted no está obligado a aceptar esta licencia, dado que no la ha firmado. Sin embargo, nada le otorga el permiso de modificar o distribuir el Programa ni sus trabajos derivados. Estas acciones están prohibidas por la ley si usted no acepta esta licencia. Sin embargo, modificando o distribuyendo el Programa (o cualquier trabajo basado en el Programa) indica su aceptación de esta licencia y de todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.

6. Cada vez que usted redistribuye el Programa (o cualquier trabajo basado en el Programa), la persona que lo recibe automáticamente recibe una licencia del autor original para copiar, distribuir o modificar el Programa sujeto a estos términos y condiciones. Usted no puede imponer ninguna restricción adicional a las personas que reciban el Programa sobre los derechos que en esta licencia se les otorga. Usted no es responsable de forzar a terceras personas al cumplimiento de esta licencia.
7. Si como consecuencia de un veredicto de un juzgado o por el alegato de infringir una patente o por cualquier otra razón (no limitado solo a cuestiones de patentes) se imponen condiciones sobre usted que contradigan los términos y condiciones de esta licencia, estas no le excusan de los términos y condiciones aquí descritos. Si usted no puede distribuir el producto cumpliendo totalmente con las obligaciones concernientes a la resolución oficial y al mismo tiempo con las obligaciones que se describen en este contrato de licencia, entonces no podrá distribuir más este producto. Por ejemplo, si una licencia de patente no permite la distribución del Programa de forma libre de regalías (sin pago de regalías) por parte de quienes lo reciban directa o indirectamente, entonces la única forma de cumplir con ambas obligaciones es renunciar a su distribución.

Si cualquier parte de esta sección resulta inválida, inaplicable o no obligatoria bajo cualquier circunstancia en particular, la tendencia de esta es a aplicarse, y la sección completa se aplicará bajo otras circunstancias. La intención de esta sección no es la de inducirlo a infringir ninguna ley de patentes, ni tampoco infringir algún reclamo de derechos, ni discutir la validez de tales reclamos; esta sección tiene el único propósito de proteger la integridad del sistema de distribución del software libre, que está implementado por prácticas de licencia pública. Mucha gente ha hecho generosas contribuciones a la amplia gama de software distribuido bajo este sistema, favoreciendo así la constante aplicación de este sistema de distribución; es decisión del autor/donador si su Programa será distribuido utilizando este u otro sistema de distribución, y la persona que recibe el software no puede obligarlo a hacer ninguna elección en particular.

Esta sección pretende dejar muy en claro lo que se cree que será una consecuencia del resto de esta licencia.

8. Si la distribución o el uso del Programa se restringen a algunos países, ya sea por patentes, o por interfases protegidas por derechos de autor, el propietario original de los derechos de autor que ubica su Programa bajo esta licencia puede agregar una restricción geográfica de distribución explícita excluyendo los países que aplique, dando como resultado que su distribución solo se permita en los países no excluidos. En tal caso, esta Licencia incorpora la limitación como si hubiera sido escrita en el cuerpo de esta misma Licencia.
9. La FSF Free Software Foundation puede publicar versiones nuevas o revisadas de la GPL General Public License de uno a otro momento. Estas nuevas versiones mantendrán el espíritu de la presente versión, pero pueden diferir en la

inclusión de nuevos problemas o en la manera de tocar los problemas o aspectos ya presentes.

Cada versión tendrá un número de versión que la distinga. Si el Programa especifica un número de versión para esta Licencia que aplique a él y “cualquier versión subsecuente”, usted tiene la opción de seguir los términos y condiciones de dicha versión o de cualquiera de las posteriores versiones publicadas por la “FSF”. Si el Programa no especifica una versión en especial de esta licencia, usted puede elegir entre cualquiera de las versiones que han sido publicadas por la FSF.

10. Si usted desea incorporar partes del Programa en otros programas de software libre cuyas condiciones de distribución sean distintas, deberá escribir al autor solicitando su autorización. Para programas de software protegidas por la FSF Free Software Foundation, deberá escribir a la FSF solicitando autorización, en ocasiones hacemos excepciones. Nuestra decisión será guiada por dos metas principales:

Mantener el estado de libertad de todos los derivados de nuestro software libre.

Promover el uso comunitario y compartido del software en general.

Inexistencia de garantía

11. DEBIDO A QUE EL PROGRAMA SE OTORGA LIBRE DE CARGOS Y REGALÍAS, NO EXISTE NINGUNA GARANTÍA PARA EL MISMO HASTA DONDE LO PERMITA LA LEY APLICABLE. A EXCEPCIÓN DE QUE SE INDIQUE OTRA COSA, LOS PROPIETARIOS DE LOS DERECHOS DE AUTOR PROPORCIONAN EL PROGRAMA “COMO ES” SIN NINGUNA GARANTÍA DE NINGÚN TIPO, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO NO LIMITADA A, LAS GARANTÍAS QUE IMPLICA EL MERCADEO Y EJERCICIO DE UN PROPÓSITO EN PARTICULAR. CUALQUIER RIESGO DEBIDO A LA CALIDAD Y DESEMPEÑO DEL PROGRAMA ES TOMADO COMPLETAMENTE POR USTED. SI EL SOFTWARE MUESTRA ALGÚN DEFECTO, USTED CUBRIRÁ LOS COSTOS DE CUALQUIER SERVICIO, REPARACIÓN O CORRECCIÓN DE SUS EQUIPOS Y/O SOFTWARE QUE REQUIERA.
12. EN NINGÚN CASO NI BAJO NINGUNA CIRCUNSTANCIA EXCEPTO BAJO SOLICITUD DE LA LEY O DE COMÚN ACUERDO POR ESCRITO, NINGÚN PROPIETARIO DE LOS DERECHOS DE AUTOR NI TERCERAS PERSONAS QUE PUDIERAN MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA COMO SE PERMITE ARRIBA, SERÁN RESPONSABLES DE LOS DAÑOS CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA, SIN IMPORTAR SI SON DAÑOS GENERALES, ESPECIALES, INCIDENTALES O CONSECUENTES CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA (INCLUYENDO PERO NO LIMITADO A LA PÉRDIDA DE INFORMACIÓN O DETERIORO DE LA MISMA AFECTÁNDOLO A US-

TED, A TERCERAS PERSONAS QUE SEA POR FALLAS EN LA OPERACIÓN DEL PROGRAMA O SU INTERACCIÓN CON OTROS PROGRAMAS) INCLUSIVE SI TAL PROPIETARIO U OTRAS PERSONAS HAYAN SIDO NOTIFICADAS DE TALES FALLAS Y DE LA POSIBILIDAD DE TALES DAÑOS.

FIN DE TÉRMINOS Y CONDICIONES

Autores

Lilia Edith Aparicio Pico

Docente de la Universidad Distrital Francisco José de Caldas. Doctora en Ciencias Técnicas, magíster en Teleinformática y especialista en Gerencia de Proyectos Educativos Institucionales. Licenciada en Física y directora del grupo de investigación Gitem++ de la misma universidad.

Paulo César Coronado Sánchez

Docente de la Universidad Distrital Francisco José de Caldas, magíster en Ciencias de la Información y las Comunicaciones e ingeniero Electrónico. Integrante del grupo de investigación Gitem++ de la misma institución.

Este libro se
terminó de imprimir
en abril de 2019
en la Editorial UD
Bogotá, Colombia