

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**GESTIÓN DEL “REGISTRO ANESTÉSICO” MEDIANTE  
SERVICIOS WEB REST Y PLATAFORMA ANDROID  
CASO: SERVICIO DE ANESTESIOLOGÍA DEL  
HOSPITAL DE LA MUJER**

**Proyecto de Grado para obtener el Título de Licenciatura en Informática**  
**Mención Ingeniería de Sistemas Informáticos**

**POSTULANTE: MANUEL HENRY SANCHEZ CARVAJAL**

**TUTOR: M.SC. EDGAR PALMIRO CLAVIJO CARDENAS**

**ASESOR: LIC. MARCELO GERMAN ARQUIPA CHAMBI**

**LA PAZ – BOLIVIA**  
**2016**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## **Dedicatoria**

*A mis padres por su apoyo constante, palabras de aliento e insistencia en lograr todos mis objetivos tanto en mi formación profesional como en mi vida personal.*

## **Agradecimientos**

- Al Dr. Juan Carlos García Caballero, Jefe de Anestesiología y Quirófanos del Hospital de la Mujer por su colaboración y orientación durante el desarrollo del proyecto.
- Al Dr. Juan José Cordón Hinojosa, médico de planta del Servicio de Anestesiología del Hospital de la Mujer, por su apoyo incondicional y predisposición para hacer posible la conclusión de este proyecto.
- A mi tutor M. Sc. Edgar Palmiro Clavijo Cárdenas por sus valiosos consejos y orientación y por estar siempre dispuesto a solventar mis dudas a pesar de su apretada agenda.
- A mi asesor Lic. Marcelo German Aruquipa Chambi por compartir conmigo su amplia experiencia y conocimiento para elaborar y concluir este proyecto.
- A los docentes de la Carrera de Informática de la Universidad Mayor de San Andrés por haber colaborado en mi formación profesional con sus enseñanzas y experiencia.
- A mi familia por su apoyo incondicional tanto en las buenas como en las malas.

## INDICE DE CONTENIDOS

<b>CAPITULO I. LINEAMIENTOS GENERALES.....</b>	<b>1</b>
1.1. <i>INTRODUCCION</i> .....	1
1.2. <i>ANTECEDENTES</i> .....	2
1.3. <i>PLANTEAMIENTO DEL PROBLEMA</i> .....	3
1.3.1. Problema General.....	4
1.3.2. Problemas Específicos. ....	4
1.3.3. Pregunta de Investigación. ....	4
1.4. <i>OBJETIVOS</i> .....	5
1.4.1. Objetivo General.....	5
1.4.2. Objetivos Específicos.....	5
1.5. <i>JUSTIFICACION</i> .....	5
1.5.1. Justificación Social. ....	5
1.5.2. Justificación Económica. ....	6
1.5.3. Justificación Técnica.....	6
1.6. <i>LIMITES Y ALCANCES</i> .....	7
1.6.1. Delimitación del Proyecto de Grado.....	7
1.6.1.1. <i>Delimitación Temática</i> .....	7
1.6.1.2. <i>Delimitación Temporal</i> . ....	7
1.6.1.3. <i>Delimitación Espacial</i> .....	7
1.6.2. Alcances.....	7
1.7. <i>METODOLOGÍA DE LA INVESTIGACIÓN</i> .....	8
1.8. <i>APORTES</i> . ....	8
<b>CAPITULO II. DESCRIPCIÓN DE PROCESOS Y MÉTODOS.....</b>	<b>9</b>
2.1. <i>REGISTRO ANESTÉSICO</i> .....	9
2.1.1. Generalidades.....	9
2.1.2. Registro Pre-Anestésico.....	9

2.1.3. Registro Trans-Anestésico.....	11
2.1.4. Registro Post-Anestésico.....	12
2.2. <i>INGENIERÍA DE SOFTWARE</i> .....	13
2.2.1 Métodos de desarrollo de software.....	14
2.2.2. Métodos tradicionales o clásicas de software.....	14
2.2.3. Métodos ágiles de Software.....	16
2.3. <i>MÉTODO DE DESARROLLO ADAPTABLE DE SOFTWARE ASD</i> .....	17
2.3.1. Características.....	17
2.3.2. Ciclo de Vida.....	17
2.3.2.1. <i>Especular</i> .....	18
2.3.2.2. <i>Colaborar</i> .....	18
2.3.2.3. <i>Aprender</i> .....	19
2.3.3. Ventajas.....	19
2.3.4. Desventajas.....	20
2.4. <i>SERVICIOS WEB REST</i> .....	20
2.5. <i>INGENIERÍA WEB</i> .....	22
2.5.1 Concepto .....	23
2.5.2 Sistema Web .....	23
2.5.3 Arquitectura Modelo Vista Controlador (MVC).....	24
2.5.3.1 <i>Capa del Modelo</i> .....	24
2.5.3.2 <i>Capa de la Vista</i> .....	25
2.5.3.3 <i>Capa del Controlador</i> .....	25
2.6 <i>MÉTODO UWE</i> .....	26
2.6.1 Principales modelos .....	26
2.6.1.1 <i>Modelo de Requerimientos</i> .....	26
2.6.1.2 <i>Modelo de Contenido</i> .....	26
2.6.1.3 <i>Modelo de Navegación</i> .....	26
2.6.1.4 <i>Modelo de Presentación</i> .....	27
2.6.2 Etapas del método UWE.....	27
2.6.2.1 <i>Captura, análisis y especificación de requisitos</i> .....	27
2.6.2.2 <i>Diseño del sistema</i> .....	27

2.6.2.3 Codificación del software .....	27
2.6.2.4 Pruebas .....	27
2.6.2.5 Implementación .....	27
2.6.2.6 Mantenimiento y seguimiento .....	28
2.7 HERRAMIENTAS DE DESARROLLO .....	28
2.7.1 Framework CodeIgniter .....	28
2.7.2 Gestor de base de datos MySQL.....	29
2.7.3 Android Studio.....	30
2.8 CALIDAD DE SOFTWARE ISO 9126.....	31
2.8.1 Funcionalidad.....	33
2.8.2 Fiabilidad .....	33
2.8.3 Usabilidad .....	33
2.8.4 Eficiencia .....	34
2.8.5 Mantenibilidad .....	34
2.8.6 Portabilidad .....	34
2.9 MÉTODO DE ESTIMACIÓN DE COSTO DEL SISTEMA – CRI.....	35
<b>CAPITULO III. INGENIERÍA Y APLICACIÓN DE MÉTODOS .....</b>	<b>37</b>
3.1 INTRODUCCION.....	37
3.2 SITUACION ACTUAL DEL SERVICIO DE ANESTESIOLOGIA.....	37
3.3. DETERMINACION DE REQUISITOS.....	40
3.4 Aplicación Android – Método De Desarrollo Adaptable De Software (ASD).....	41
3.4.1 Fase de Especulación .....	42
3.4.1.1 Identificación de actores .....	42
3.4.1.2 Casos de uso.....	43
3.4.1.3 Resultados de la Fase de Especulación .....	54
3.4.2 Fase de Colaboración.....	56
3.4.2.1 Diagrama de Clases.....	56
3.4.2.2 Diagrama de Secuencias.....	57
3.4.2.3 Resultados de la Fase de Colaboración.....	60
3.4.3 Fase de Aprendizaje.....	60

3.4.3.1 <i>Resultados de la Fase de aprendizaje</i> .....	65
3.5 <i>APLICACIÓN WEB – MÉTODO UWE</i> .....	65
3.5.1 Modelo de Requerimientos – Casos de Uso .....	65
3.5.1.1 <i>Definición de actores</i> .....	65
3.5.1.2 <i>Casos de uso</i> .....	66
3.5.2 Modelo Conceptual – Diagrama de clases.....	75
3.5.3 Modelo Navegacional – Diagrama de navegación.....	76
3.5.4 Modelo de Presentación – Diagrama de presentación .....	77
3.5.4.1 <i>Administración del sistema</i> .....	77
3.5.4.2 <i>Gestión de pacientes</i> .....	78
3.5.4.3 <i>Consulta de registro anestésico</i> .....	78
3.5.4.4 <i>Intervención Quirúrgica</i> .....	79
3.5.5 Implementación de la aplicación web .....	80
3.5.6 Resultados del método UWE .....	82
3.7 <i>SERVICIO REST</i> .....	82
3.7.1 Descripción de los recursos.....	83
3.8 <i>MODELO FÍSICO DE LA BASE DE DATOS</i> .....	84
<b>CAPITULO IV. CALIDAD Y SEGURIDAD .....</b>	<b>86</b>
4.1 <i>CALIDAD DEL SISTEMA</i> .....	86
4.1.1 Funcionalidad.....	86
4.1.2 Fiabilidad .....	91
4.1.3 Usabilidad .....	92
4.1.4 Eficiencia .....	95
4.1.5 Mantenibilidad .....	96
4.1.6 Portabilidad .....	97
4.2 <i>SEGURIDAD DEL SISTEMA</i> .....	97
4.2.1 Autenticación .....	97
4.2.2 Encriptación .....	98
4.2.3 Base de datos.....	99
4.2.3 Hardware.....	99

<b>CAPITULO V. COSTO DEL SISTEMA .....</b>	<b>100</b>
<i>5.1 METODO DE ESTIMACIÓN CRI.....</i>	<i>100</i>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>105</b>
<i>CONCLUSIONES.....</i>	<i>105</i>
<i>RECOMENDACIONES.....</i>	<i>106</i>
<b>REFERENCIAS.....</b>	<b>108</b>
<b>ANEXOS.....</b>	<b>109</b>

## Índice de Figuras

<i>Figura 1.1 Alcances sistema SIGRA</i> .....	8
<i>Figura 2.1 Registro Pre-anestésico</i> .....	10
<i>Figura 2.2 Registro Trans-Anestésico</i> .....	12
<i>Figura 2.3 Registro post-anestésico</i> .....	13
<i>Figura 2.4 Método ASD, Ciclo de Vida</i> .....	18
<i>Figura 2.5 Peticiones sin estado</i> .....	21
<i>Figura 2.6 Caché intermedio</i> .....	21
<i>Figura 2.7 Servicios uniformes REST</i> .....	22
<i>Figura 2.8 Patrón MVC</i> .....	24
<i>Figura 2.9 ISO 9126</i> .....	32
<i>Figura 3.1 proceso de registro anestésico</i> .....	38
<i>Figura 3.3 Diagrama de casos de uso general del sistema</i> .....	43
<i>Figura 3.4 Casos de uso del proceso de registro pre anestésico</i> .....	44
<i>Figura 3.5 Casos de uso del proceso de consulta de registro pre anestésico</i> .....	50
<i>Figura 3.6 Casos de uso del proceso de registro post anestésico</i> .....	51
<i>Figura 3.7 Arquitectura del sistema – Diagrama de clases</i> .....	56
<i>Figura 3.8 Diagrama de secuencias – Registro pre anestésico</i> .....	57
<i>Figura 3.9 Diagrama de Secuencias – Consulta registro pre anestésico</i> .....	58
<i>Figura 3.10 Diagrama de secuencias – Registro Post anestésico</i> .....	59
<i>Figura 3.11 Pantalla inicial en espera de servicio</i> .....	60
<i>Figura 3.12 Pantalla inicial con acceso al servicio</i> .....	61
<i>Figura 3.13 Datos iniciales del paciente</i> .....	61
<i>Figura 3.14 Datos iniciales del paciente completados</i> .....	62
<i>Figura 3.15 Evaluación de patologías del paciente</i> .....	62

<i>Figura 3.16</i> Sumario pre anestésico hoja 1 .....	63
<i>Figura 3.17</i> Sumario pre anestésico hoja 2 .....	63
<i>Figura 3.18</i> Sumario pre anestésico hoja 3 .....	64
<i>Figura 3.19</i> Registro pre anestésico finalizado .....	64
<i>Figura 3.20</i> Diagrama general de casos de uso – aplicación web .....	66
<i>Figura 3.21</i> Consultar registro anestésico.....	67
<i>Figura 3.22</i> Iniciar procedimiento trans anestésico .....	70
<i>Figura 3.23</i> Gestionar personal.....	73
<i>Figura 3.24</i> Modelo Conceptual – Aplicación web.....	75
<i>Figura 3.25</i> Modelo de navegación – Aplicación web.....	76
<i>Figura 3.26</i> Diagrama de presentación – Administración del sistema.....	77
<i>Figura 3.27</i> Diagrama de presentación – Gestión de pacientes.....	78
<i>Figura 3.28</i> Diagrama de presentación – Consulta de registro anestésico.....	78
<i>Figura 3.29</i> Diagrama de presentación – Intervención quirúrgica.....	79
<i>Figura 3.30</i> Administración del sistema .....	80
<i>Figura 3.31</i> Gestión de pacientes.....	80
<i>Figura 3.32</i> Consulta de registro anestésico .....	81
<i>Figura 3.33</i> Intervención quirúrgica - hoja inicial.....	81
<i>Figura 3.34</i> Intervención quirúrgica – hoja anestésica.....	82
<i>Figura 3.35</i> Modelo físico de la base de datos .....	85
<i>Figura 4.1</i> Encriptación del servicio REST .....	98
<i>Figura A.</i> Árbol de problemas causa – efecto.....	109
<i>Figura B.</i> Descripción de clases – Aplicación android.....	110
<i>Figura C1.</i> Cuestionario – Utilidad percibida UP .....	111
<i>Figura C2.</i> Cuestionario – Facilidad de uso percibida FUP.....	111

<b>Figura C3.</b> <i>Cuestionario – Actitud hacia el uso AHU</i> .....	111
<b>Figura D1.</b> <i>Auto completado – Aplicación android</i> .....	112
<b>Figura D2.</b> <i>Auto completado en cuadro de dialogo – Aplicación web</i> .....	112

## Lista de Tablas

<i>Tabla 1.1</i> Insumos para la implementación .....	6
<i>Tabla 2.1</i> Complejidad por tipo de dato.....	36
<i>Tabla 3.1</i> Actividades por fase del método ASD .....	41
<i>Tabla 3.2</i> Definición de actores – aplicación android.....	42
<i>Tabla 3.3</i> Caso de uso “Iniciar sesión” .....	45
<i>Tabla 3.4</i> Caso de uso “Iniciar registro pre anestésico”.....	46
<i>Tabla 3.5</i> Caso de uso “Registrar paciente nuevo”.....	46
<i>Tabla 3.6</i> Caso de uso “Seleccionar historia clínica de paciente”.....	47
<i>Tabla 3.7</i> Caso de uso “Evaluar patologías y antecedentes del paciente”.....	48
<i>Tabla 3.8</i> Caso de uso “Consultar sumario de registro pre anestésico”.....	48
<i>Tabla 3.9</i> Caso de uso “Guardar registro pre anestésico” .....	49
<i>Tabla 3.10</i> Caso de uso “Agregar observaciones adicionales” .....	50
<i>Tabla 3.11</i> Caso de uso “Realizar control de recuperación post anestésica”.....	51
<i>Tabla 3.12</i> Caso de uso “Controlar signos vitales” .....	52
<i>Tabla 3.13</i> Caso de uso “Controlar uso de equipos” .....	52
<i>Tabla 3.14</i> Caso de uso “Controlar escala de Aldrede/Kroulik” .....	53
<i>Tabla 3.15</i> Caso de uso “Prescribir medicamentos”.....	53
<i>Tabla 3.16</i> Caso de uso “Supervisar registro post anestésico”.....	53
<i>Tabla 3.17</i> Caso de uso “Guardar registro post anestésico”.....	54
<i>Tabla 3.18</i> Iteraciones por cada componente .....	55
<i>Tabla 3.19</i> Definición de actores – aplicación web .....	65
<i>Tabla 3.20</i> Caso de uso “Iniciar sesión” .....	67
<i>Tabla 3.21</i> Caso de uso “Seleccionar paciente”.....	67
<i>Tabla 3.22</i> Caso de uso “Consultar registro anestésico”.....	68

<i>Tabla 3.23</i> Caso de uso “ <i>Editar registro de paciente</i> ” .....	68
<i>Tabla 3.24</i> Caso de uso “ <i>Generar reporte impreso</i> ” .....	69
<i>Tabla 3.25</i> Caso de uso “ <i>Dar de alta paciente</i> ” .....	69
<i>Tabla 3.26</i> Caso de uso “ <i>Cerrar sesión</i> ” .....	70
<i>Tabla 3.27</i> Caso de uso “ <i>Iniciar trans anestésico</i> ” .....	71
<i>Tabla 3.28</i> Caso de uso “ <i>Registrar datos iniciales paciente</i> ” .....	71
<i>Tabla 3.29</i> Caso de uso “ <i>Definir equipo médico</i> ” .....	71
<i>Tabla 3.30</i> Caso de uso “ <i>Realizar controles y procedimientos</i> ” .....	72
<i>Tabla 3.31</i> Caso de uso “ <i>Seleccionar usuario</i> ” .....	73
<i>Tabla 3.32</i> Caso de uso “ <i>Gestionar usuario</i> ” .....	73
<i>Tabla 3.33</i> Caso de uso “ <i>Generar reporte de personal</i> ” .....	74
<i>Tabla 3.34</i> Caso de uso “ <i>Gestionar personal</i> ” .....	74
<i>Tabla 3.35</i> Descripción del servicio REST.....	83
<i>Tabla 4.1</i> Aplicación android: Cuenta total – factor de ponderación medio.....	87
<i>Tabla 4.2</i> Aplicación android: Ajuste de complejidad del punto función .....	88
<i>Tabla 4.3</i> Aplicación web: Cuenta total – factor de ponderación medio.....	89
<i>Tabla 4.4</i> Aplicación web: Ajuste de complejidad del punto función .....	90
<i>Tabla 4.5</i> Fiabilidad por componente .....	92
<i>Tabla 4.6</i> Utilidad Percibida.....	93
<i>Tabla 4.7</i> Facilidad de uso percibida.....	94
<i>Tabla 4.8</i> Actitud hacia el uso.....	95
<i>Tabla 4.9</i> Eficiencia del servicio REST.....	96
<i>Tabla 5.1</i> Unidades de trabajo de los campos .....	101
<i>Tabla 5.2</i> Unidades de trabajo de las relaciones.....	102
<i>Tabla 5.3</i> Unidades de trabajo de los impresos .....	103

## Resumen

El presente proyecto de grado tiene como objetivo diseñar e implementar el Sistema de Información para la Gestión del Registro Anestésico SIGRA como solución al problema que atraviesa el Servicio de Anestesiología del Hospital de la Mujer con la elaboración manual del registro anestésico. Este registro es la constancia escrita de las técnicas y procedimientos utilizados durante una intervención quirúrgica, su importancia radica en que sirve como un documento legal para establecer responsabilidades cuando se requiere hacer una investigación por mala praxis, también sirve como un documento estadístico para la investigación de efectividad de nuevas técnicas o procedimientos anestésicos.

A nivel nacional, en los centros de salud, actualmente el registro anestésico sigue elaborándose de forma manual, de modo que no es solamente un problema particular del Servicio de Anestesiología del Hospital de la Mujer.

El sistema SIGRA es un sistema compuesto por una aplicación Web, una aplicación Android y un servicio Web REST que en conjunto brindan una gestión integral del registro anestésico como alternativa a los documentos escritos susceptibles a extravío, adulteración o malinterpretación.

## **Abstract**

The main objective of the current Grade Project is design and implement the Information System for Anesthetic Registry Management SIGRA as a solution to the problem faced by the Anesthetic Service of the Woman's Hospital that currently is the manual elaboration of the anesthetic record. This record is the written proof of the techniques and procedures used in a surgical intervention, its importance resides in the fact that it is used as a legal document to establish responsibilities in an ongoing investigation for a bad practice case, it is also used as a statistical document for research purposes of new techniques and procedures effectiveness.

Currently in clinics on a nacional level, this anesthetic record is done manually, thus the problem faced by the Anesthetic Service of the Woman's Hospital is shared by several other clinics and hospitals.

The SIGRA system is composed by a Web application, an Android app and a Web service fully compliant with REST specifications, these components together offer an integral management of the anesthetic record as an alternative to written documents susceptible of misplacing, modifications or misunderstanding.

## CAPITULO I. LINEAMIENTOS GENERALES

### 1.1. INTRODUCCION

Hoy en día en los centros de salud de tercer y segundo nivel tienen lugar numerosas intervenciones quirúrgicas. Este tipo de operaciones, sin considerar el tipo, grado de riesgo o paciente intervenido, invariablemente es registrado en una bitácora que es conocida como el registro anestésico, el cual es tradicionalmente un documento importante para la recolección de datos durante el acto anestésico.

En la elaboración del registro anestésico, los integrantes del equipo de salud y de la institución hospitalaria describen, explican, comunican, deciden y enfrentan las situaciones y riesgos a los cuales se somete al paciente. Se desprende de lo anterior la obligación que tiene el anestesiólogo de informar al paciente y sus familiares sobre los riesgos que implica la intervención quirúrgica.

El anestesiólogo que no lleva el registro escrito de sus actuaciones se arriesga a cometer errores en el futuro y a no tener ninguna defensa ante los colegas, la institución donde trabaja y ante los tribunales donde sea llamado por errores, complicaciones, accidentes o muertes ocurridas durante el proceso operatorio.

En este aspecto la informática mediante el uso de las tecnologías de la información llega a ser el aliado perfecto para agilizar estos procesos manuales para así conseguir la eficiencia deseada.

Es por esta razón que el Servicio de Anestesiología del Hospital de la Mujer considera necesario e imprescindible informatizar el proceso de registro anestésico, para contar con información oportuna y fiable en el momento debido, a fin de mejorar la calidad del servicio que proporcionan.

El presente proyecto tiene como objetivo diseñar e implementar el sistema de información SIGRA (Sistema de Información para la Gestión del Registro Anestésico) que implemente un cliente Android, un servicio REST y un sistema web que permita agilizar el proceso de registro anestésico y que cumpla con las expectativas y requerimientos del Servicio de Anestesiología del Hospital de la Mujer.

## 1.2. ANTECEDENTES

Los primeros desarrollos de sistemas de información en anestesia se remontan a la década de los 70. En los 90, la Universidad de Duke puso en funcionamiento el Arkive Organizer, un registro electrónico intraoperatorio diseñado en San Diego, California, para ser utilizado en el área quirúrgica. La implementación generalizada de dichos sistemas se caracteriza por ser más precisa, legible, completa y fiable que la gráfica anestésica tradicional.

En la actualidad, a nivel internacional, existen ejemplos de sistemas afines, tal es el caso del SIRUMA (Sistema de Registro Universal de Manejo Anestesiológico) dependiente de la Sociedad Colombiana de Anestesiología y Reanimación (S.C.A.R.E.), también el Hospital Italiano de Buenos Aires, a través de su Departamento de Informática en Salud, presentó en el XVIII Congreso Argentino de Ciencias de la Computación (2013), el desarrollo de una “Ficha Anestésica Web en Áreas Críticas”, que permite una gran escalabilidad y la posibilidad a futuro de uso en dispositivos móviles. Además, indican que en Estados Unidos en 1998 apenas el 1% de los departamentos de anestesia utilizaban sistemas de documentación informáticos en la sala de cirugía, y se estima que actualmente menos del 10 % de todos los hospitales cuentan con este tipo de sistemas.

A nivel nacional, aunque cada institución hospitalaria tiene su propio modelo de registro anestésico (fichas impresas), los datos que en él se consignan son universales.

En la carrera de Informática de la Universidad Mayor de San Andrés se realizaron trabajos que guardan relación con el presente proyecto, también en el seguimiento y gestión de departamentos clínicos, algunos de ellos son:

- “Sistema de Información Hospital de la Mujer Gestión Médica” (2001), en el proyecto se adoptó el paradigma del ciclo de vida clásico, a fin de desarrollar un sistema de información que optimice de forma eficaz y correcta los procesos de registro de pacientes, procesamiento de información estadística, seguimiento de pacientes y recaudaciones.
- “Sistema de Seguimiento Clínico Hospital del Niño ‘Dr. Ovidio Aliaga Uría’” (2005), donde se usó la metodología del Desarrollo Rápido de Aplicaciones RAD (Rapid Application Development) junto con una arquitectura cliente-servidor para la implantación de un sistema de información que gestione el control adecuado y correcto de toda la

información referente a los pacientes, en la administración de los procedimientos de información e indicadores estadísticos.

- “Sistema de Gestión de Internación Quirófano y Enfermería, Caso Hospital Boliviano Holandés” (2008), este proyecto emplea la metodología RUP así como también métricas de calidad para diseñar e implementar un sistema de gestión que permite mejorar el control y manejo de información en los departamentos de internación, quirófano y enfermería, logrando también una comunicación entre hospitalización y áreas necesarias.
- “Sistema de Control y Seguimiento de Pacientes Hospital Arco Iris” (2011), en el que hacen uso de la metodología RUP y el lenguaje de modelado UML para diseñar y desarrollar un sistema gestione los registros del personal, pacientes, asignación de infraestructura (piso, sala, cama) con el objetivo de mejorar el acceso y la manipulación de dicha información.
- “Sistema Web de Seguimiento y Control de Pacientes internos, Caso: Hospital Arco Iris” (2014), que se basa en la metodología de proceso unificado ágil AUP (Agile Unified Process) para desarrollar un sistema que contribuye a la transición de la historia clínica tradicional hacia su sustitución con soportes digitales que permiten almacenar y procesar gran cantidad de datos para el control y seguimiento.

### **1.3. PLANTEAMIENTO DEL PROBLEMA.**

En la actualidad el Servicio de Anestesiología del Hospital de la Mujer elabora el registro anestésico manualmente, en sus tres etapas: pre, trans y post anestésico, en formularios o fichas impresas que con el tiempo son susceptibles a deterioro o alteración, además de conllevar todas las desventajas que implican la manipulación de documentos escritos:

- Grandes volúmenes de documentación.
- Difícil organización.
- Los registros anestésicos (especialmente el trans-anestésico) requieren demasiado tiempo para ser llenados.
- Hay casos en los que un registro anestésico por parte de un doctor no es legible por otro.
- Los registros van perdiendo legibilidad con el paso del tiempo.
- Hay casos en los que existe pérdida de un registro anestésico o adulteración del mismo.

Debido a esto, se presentan complicaciones al momento de brindar el servicio, o cuando se requiere una revisión de una intervención anterior, hay que considerar que el registro anestésico está compuesto de tres fichas y la pérdida de cualquiera supone la imposibilidad de establecer un estudio retrospectivo del mismo.

### **1.3.1. Problema General.**

El Servicio de Anestesiología del Hospital de la Mujer afronta como principal problema la elaboración manual del registro anestésico en sus etapas pre, trans y post anestésico, lo que genera dificultad en la movilidad, diagnóstico y planeación oportuna de las intervenciones quirúrgicas.

### **1.3.2. Problemas Específicos.**

- El registro pre-anestésico y post-anestésico es realizado de manera informal, al no disponer el hospital de una herramienta o sistema adecuado.
- El registro trans-anestésico se realiza en el mismo quirófano, el cual consume tiempo imprescindible para el paciente.
- El conjunto de los tres registros no siempre se archiva conjuntamente, habiendo trasapelados que dificultan el seguimiento.
- Debido su formato físico, el registro anestésico puede verse deteriorado o alterado de forma irreversible.
- En caso de necesitarse reportes de una intervención, el proceso es moroso debido a la complejidad del registro anestésico.

### **1.3.3. Pregunta de Investigación.**

¿Cómo se puede mejorar el llenado manual del registro anestésico de modo que se evite la pérdida o alteración de datos y también automatizar la generación de reportes para así contar con información precisa, oportuna y útil en el servicio de anestesiología del Hospital de la Mujer?

## **1.4. OBJETIVOS.**

### **1.4.1. Objetivo General.**

Diseñar e implementar el Sistema de Información para la Gestión del Registro Anestésico SIGRA que gestione eficientemente el proceso de registro anestésico mediante el uso de servicios web y dispositivos móviles para proporcionar información precisa, oportuna y útil al Servicio de Anestesiología del Hospital de la Mujer.

### **1.4.2. Objetivos Específicos.**

- Desarrollar una aplicación móvil Android que permita realizar el registro pre y post anestésico de manera fácil e intuitiva, a la vez permitiendo la movilidad requerida al no contarse con un consultorio dedicado.
- Diseñar la arquitectura Cliente – Servidor para una aplicación Web que permita el registro trans-anestésico de manera rápida y precisa.
- Desarrollar un servicio web basado en REST para permitir la comunicación e intercambio de información entre la aplicación Android y la aplicación Web, a fin de consolidar los datos de los tres registros anestésicos.
- Implementar un acceso seguro y diferenciado al sistema para evitar que personas ajenas alteren los datos del registro anestésico.
- Diseñar reportes en base a los tres registros anestésicos.
- Diseñar e implementar una Base de Datos normalizada para los tres tipos de registros anestésicos.

## **1.5. JUSTIFICACION.**

### **1.5.1. Justificación Social.**

Podemos justificar el sistema en función al beneficio social que proporcionará. Al permitir una gestión eficiente y fiable del registro anestésico, el Servicio de Anestesiología del Hospital de la Mujer será un referente respecto a las demás entidades de salud, que aún realizan de forma manual el registro, además fomentará la buena praxis médica en beneficio de los pacientes, al registrarse en detalle y de forma clara los procedimientos empleados en la intervención quirúrgica.

### 1.5.2. Justificación Económica.

Se cuenta con un equipo de escritorio, una laptop y una red de cableado estructurado y acceso a internet de banda ancha. La inversión a realizarse será la siguiente:

**Tabla 1.1** Insumos para la implementación

Detalle	Costo Bs.
• Contratación de hosting (un año)	350 Bs.
• Tablet Android de 7''	2000 Bs.
• Router wifi	200 Bs.
<b>TOTAL</b>	<b>2550 Bs.</b>

La implementación del sistema reducirá los costos en tiempo tanto del llenado del registro anestésico como de las consultas y generación de reportes, por tanto se justifica económicamente considerando que existen los medios y recursos económicos para el desarrollo del proyecto.

### 1.5.3. Justificación Técnica.

El proyecto contempla el desarrollo de una aplicación Android, un servicio REST y una aplicación web, de modo que se usarán herramientas de licencia abierta. Las herramientas principales para desarrollar e implementar el sistema web será el lenguaje PHP (con licencia de software libre), el sistema gestor de bases de datos MySQL (licencia GNU GPL), además del uso de HTML, CSS y javascript para el diseño responsivo de la interface de usuario. Se utilizará Android Studio (publicado de forma gratuita a través de la licencia Apache 2.0) para el desarrollo de la plataforma Android.

El sistema a implementar podrá ejecutarse en los equipos que dispone el hospital ya que no requiere muchos recursos de hardware y se tomará en cuenta la funcionalidad, confiabilidad, usabilidad y portabilidad.

## **1.6. LIMITES Y ALCANCES**

### **1.6.1. Delimitación del Proyecto de Grado**

#### **1.6.1.1. Delimitación Temática**

El presente proyecto de grado se enmarca dentro de los sistemas de servicio web basados en REST para proporcionar un servicio en el área de medicina.

Podemos diferenciar tres componentes principales:

- i) La aplicación Android, que permite el registro pre anestésico (realizado por un doctor); y el registro post anestésico (puede ser doctor o interno) y también permite consultas de registros pre anestésicos (solamente doctores).
- ii) El sistema web, que registra el proceso trans operatorio, las consultas de reportes y gestión del registro anestésico.
- iii) El servicio REST encargado de proveer los recursos a petición de la aplicación Android.

#### **1.6.1.2. Delimitación Temporal.**

Se tomará en cuenta desde la implementación y puesta en marcha del sistema, ya que no existen antecedentes oficiales sobre el presente proyecto de grado.

#### **1.6.1.3. Delimitación Espacial.**

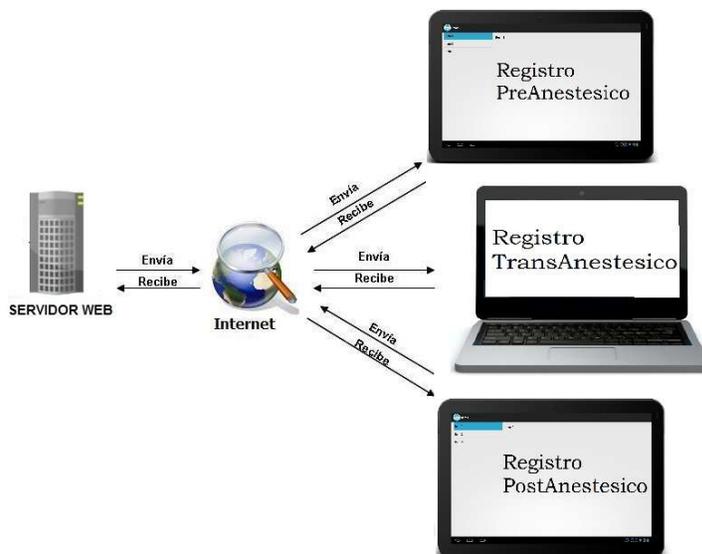
Inicialmente el sistema entrará en servicio en el Hospital de la Mujer para dar soporte al Servicio de Anestesiología.

### **1.6.2. Alcances**

Los alcances propuestos para este sistema son:

- Tener una base de datos centralizada para los tres tipos de registros (pre, trans y post anestésico).
- Disponer de un sistema web con acceso diferenciado a los usuarios para el registro trans-anestésico y la gestión de pacientes y registros.
- También disponer de una interface amigable e intuitiva que permita la consulta e impresión de los registros anestésicos.

- Disponer de una aplicación Android para los registros pre y post anestésicos así como para la consulta de registros pre anestésicos.



**Figura 1.1** Alcances sistema SIGRA

## 1.7. METODOLOGÍA DE LA INVESTIGACIÓN

Es necesario conocer los métodos que rigen el desarrollo de un sistema, y que se aplican en todos los sistemas de información automatizados.

En este sentido, para el desarrollo del proyecto de grado se utilizará el enfoque cualitativo de manera que se considera la investigación exploratoria y descriptiva y el método ASD para el ciclo de vida del proyecto.

Para la recopilación de información se realizarán entrevistas con profesionales del Servicio de Anestesiología del Hospital de la Mujer.

## 1.8. APORTES.

El sistema propuesto permitirá cubrir las falencias a la hora de realizar y gestionar los registros anestésicos, teniendo siempre que se necesite, un reporte detallado.

El aporte del presente proyecto de grado en el área de Informática será el desarrollo de un API REST que hará posible el intercambio de información entre dispositivos Android y sistemas web.

## CAPITULO II. DESCRIPCIÓN DE PROCESOS Y MÉTODOS

### 2.1. REGISTRO ANESTÉSICO.

#### 2.1.1. Generalidades.

El registro anestésico es la representación escrita y gráfica de lo que sucede en la sala de operaciones cuando un paciente es intervenido. Es también un documento médico, legal, de investigación, docente, estadístico y epidemiológico.

El registro anestésico es un documento médico ya que en él se anotan los datos de la identificación del paciente, el estado de sus condiciones físicas y mentales, los riesgos a los que está sometido, la clasificación del estado físico, las drogas que le han sido administradas con sus respectivas dosis y horarios, las técnicas anestésicas aplicadas, las complicaciones presentadas y las decisiones y observaciones pertinentes del equipo de salud. En el registro anestésico debe constar la información que sea importante para tomar decisiones en un momento dado, prever las posibles consecuencias del tratamiento instaurado, comunicar a otros lo que se hizo y planificar el tratamiento que se debe seguir.

El registro anestésico es también un documento útil en las investigaciones de drogas, técnicas o procedimientos anestésicos y en asuntos éticos o legales; su utilidad también se extiende al aprendizaje, la autoformación y la autocrítica.

El registro anestésico es un documento estadístico y epidemiológico ya que permite conocer la incidencia de enfermedades y la morbilidad y mortalidad asociadas al uso de determinadas drogas y técnicas anestésicas o quirúrgicas.

El registro se divide en: pre-anestésico, trans-anestésico y post-anestésico.

#### 2.1.2. Registro Pre-Anestésico.

O pre-operatorio, es la parte del proceso anestésico en la cual se realizan la anamnesis (conjunto de datos que se recogen en la historia clínica de un paciente), el examen físico y la estimación de los riesgos; asimismo, se informa al paciente y a su familia sobre el procedimiento y se escoge la técnica anestésica, las drogas y el tipo de monitorización. Se deben anotar los siguientes datos:

- Apellidos y nombres completos.
- El número de la historia clínica hospitalaria del paciente.

- El peso en kilogramos.
- La clasificación del paciente hospitalizado en ambulatorio o urgente.
- La clasificación del estado físico y el carácter electivo o urgente de la cirugía.
- Los resultados de los exámenes de laboratorio de: hemoglobina, hematocrito, glicemia, urea, creatinina y examen de orina.
- Los resultados del examen físico y, si es del caso, su registro en la historia clínica del paciente.
- La terapia previa a la cirugía.
- La patología asociada al problema por el cual se opera.
- Las cifras de la presión arterial, el pulso por minuto y la temperatura.
- El diagnóstico de intervención.
- El tipo de cirugía.
- La medicación pre anestésica con la fecha, la hora, la dosis y la vía de administración de las drogas utilizadas.

**EVALUACION PREANESTESICA Y CONSENTIMIENTO**

Nombre: \_\_\_\_\_ Sexo:  Macho  Femenino Sala:  Cama:

Peso	Altura	PA	FC	T	Fr	Servicio
Diagnóstico			Procedimiento			

**Evaluación de Anestésico**

<p><b>Cardio-circulatorio</b></p> <input type="checkbox"/> Hipertensión <input type="checkbox"/> Angina <input type="checkbox"/> Coronariopatía <input type="checkbox"/> Infarto del Miocardio <input type="checkbox"/> Insuf. Cardiaca <input type="checkbox"/> Otras	<p><b>Músculo esquelético</b></p> <input type="checkbox"/> Dolor Lumbar <input type="checkbox"/> Miopatía <input type="checkbox"/> Otras <input type="checkbox"/> Endocrino <input type="checkbox"/> Diabetes <input type="checkbox"/> Tiroides <input type="checkbox"/> Otras	<p><b>Alergias</b></p> <input type="checkbox"/> Negativo <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="font-size: small;">Tipo / Agente</th> <th style="font-size: small;">Reacción</th> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </table> <p>Examen físico</p> <p>Cardíaco: _____</p> <p>Resp: _____</p> <p>Neuro: _____</p> <p>Regional: _____</p> <p>Otro: _____</p> <p>Medicación Previa</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td> </td></tr> <tr><td> </td></tr> <tr><td> </td></tr> <tr><td> </td></tr> </table>	Tipo / Agente	Reacción													
Tipo / Agente	Reacción																
<p><b>Respiratorio</b></p> <input type="checkbox"/> Dependencia de O <sub>2</sub> <input type="checkbox"/> Apnea de Sueño <input type="checkbox"/> IVAS Recientes <input type="checkbox"/> Expectoraiones <input type="checkbox"/> Otras	<p><b>Cancer</b></p> <input type="checkbox"/> Quimioterapia <input type="checkbox"/> Ciras <input type="checkbox"/> Infecciosas <input type="checkbox"/> VIH <input type="checkbox"/> Otras	<p><b>Embarazo</b></p> <input type="checkbox"/> Negativo <p>Edad gestacional: _____ Semanas</p> <p><b>Hábitos sociales</b></p> <input type="checkbox"/> Ant. Patológicos <input type="checkbox"/> Tabaco <input type="checkbox"/> Alcohol <input type="checkbox"/> Cigarros <input type="checkbox"/> Drogas <input type="checkbox"/> Otras															
<p><b>Gastrointestinal/Rep.</b></p> <input type="checkbox"/> Refl. Gastro-esofágico <input type="checkbox"/> Úlcera Péptica <input type="checkbox"/> Diarreas/vómitos <input type="checkbox"/> Hemias <input type="checkbox"/> Obst. Intestinal <input type="checkbox"/> Otras	<p><b>Neurrológico</b></p> <input type="checkbox"/> Convulsiones <input type="checkbox"/> Demencia <input type="checkbox"/> Lesión medular <input type="checkbox"/> Otras	<p><b>Cirugía. Anest. Previa</b></p> <input type="checkbox"/> Negativo <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="font-size: x-small;">Cirugía</th> <th style="font-size: x-small;">Anestésica</th> <th style="font-size: x-small;">Obs.</th> </tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </table> <p>Hist. náuseas/vómitos post-operat.</p> <input type="checkbox"/> Si <input type="checkbox"/> No <p>Hist. Familiar prob. c/anestésica</p> <input type="checkbox"/> Si <input type="checkbox"/> No <p>Laboratorio</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="font-size: x-small;">HD</td> <td style="font-size: x-small;">HI</td> <td style="font-size: x-small;">Ha</td> </tr> <tr> <td style="font-size: x-small;">K</td> <td style="font-size: x-small;">Plaquetas</td> <td style="font-size: x-small;">Glucosa</td> </tr> </table>	Cirugía	Anestésica	Obs.							HD	HI	Ha	K	Plaquetas	Glucosa
Cirugía	Anestésica	Obs.															
HD	HI	Ha															
K	Plaquetas	Glucosa															
<p><b>Renal</b></p> <input type="checkbox"/> IVU <input type="checkbox"/> Insuficiencia renal <input type="checkbox"/> Otras	<p><b>Hematológico</b></p> <input type="checkbox"/> Transfusiones previas <input type="checkbox"/> Plaquetopenia <input type="checkbox"/> Otras	<p><b>Vía Aérea</b></p> <p>Hist. de Vía aérea difícil</p> <input type="checkbox"/> Si <input type="checkbox"/> No <p>Mallampati</p> <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <p><b>Estado Físico</b></p> <input type="checkbox"/> I <input type="checkbox"/> II <input type="checkbox"/> III <input type="checkbox"/> IV <input type="checkbox"/> V															

**COMENTARIOS**

ANESTESIOLOGO: \_\_\_\_\_ FECHA: \_\_\_\_\_

FIRMA DEL PACIENTE: \_\_\_\_\_ FAMILIARES: \_\_\_\_\_

**Figura 2.1** Registro Pre-anestésico

**Fuente:** Servicio de Anestesiología, Hospital de la Mujer

### 2.1.3. Registro Trans-Anestésico.

O trans-operatorio, es la parte del proceso en la cual se lleva a cabo la cirugía o el procedimiento programado según el diagnóstico y la planeación. Durante el trans-anestésico se requiere la presencia constante y vigilante del anestesiólogo para brindar la máxima seguridad al paciente, facilitar la acción del cirujano y procurar el pronto regreso a las actividades cotidianas. Se debe consignar:

- El tiempo, en el registro cada cuatro pequeño equivale a cinco minutos; en la línea superior donde aparece la palabra hora se escribe el momento en el cual comenzó la administración del anestésico.
- La concentración de oxígeno.
- La concentración de óxido nitroso.
- Las convenciones o símbolos internacionales, de ellos los más utilizados son: inicio de anestesia, inicio de la cirugía, presión arterial sistólica, presión arterial diastólica, pulso, frecuencia respiratoria.
- El tiempo de anotación; en el que consta la hora en que se aplica una droga, se realiza un procedimiento u ocurrió alguna complicación. Si es necesario, esta información también se escribe y complementa en la historia clínica.
- El nombre de los anestésicos utilizados.
- El método anestésico utilizado.
- La intubación, en este espacio se debe describir si su realización fue fácil o difícil, nasal o traqueal, así como el tamaño del tubo empleado.
- El nombre del cirujano, anestesiólogo y ayudantes.
- El tipo de cirugía.
- La cantidad y el tipo de líquidos aplicados.
- El tipo de monitorización instrumental que le fue aplicada al paciente.
- La firma del anestesiólogo con el código que le corresponde.



**RECUPERACIÓN POST ANESTESICA**

Entrada a RPA \_\_\_\_\_ Hrs. \_\_\_\_\_ Nombre: \_\_\_\_\_

PRESCRIPCIÓN MEDICA										HORARIO	

CONTROLES								USO DE EQUIPOS			
HORA	SAT	T°	O <sub>2</sub>	Resp.	P.A.	SV	SNG	DNGS	EQUIPAMIENTO	ORANDO	FINAL
									MONITOR		
									P.A.N.I.		
									OXÍMETRO		
									CAPNOGRAFO		
									ASPIRADOR		

ESCALA DE ALDREDE Y KROULIK									
Índice de Recuperación Post - Anestesia	Fin Cirugía	Llegada RPA.	15'	15'	45'	60'	90'	Salida	
<b>ACTIVIDAD</b> 2. Movimiento de 4 extremidades 1. Movimiento de 2 extremidades 0. Sin movimiento									
<b>RESPIRACIÓN</b> 2. Con respiración profunda y tos. 1. Deseño y respiración tímida. 0. Apnea									
<b>CIRCULACIÓN</b> 2. FfO <sub>2</sub> de 25 % de valor pre-anestésico 1. FfO <sub>2</sub> de 25 - 20 % de pre-anestésico 0. FfO <sub>2</sub> de 50 % de pre-anestésico.									
<b>CONCIENCIA</b> 2. Despierto lúcido. 1. Despierto somnoliento 0. No responde.									
<b>SATURACIÓN DE OXÍGENO</b> 2. SatO <sub>2</sub> > 95 % con mezcla atmosférica 1. SatO <sub>2</sub> > 90 % con mezcla atmosférica 0. SatO <sub>2</sub> > e 85 % con O <sub>2</sub> suplementaria.									
<b>TOTAL</b>									

**TODO PACIENTE DEBE SUMAR MAS DE 7 PUNTOS PARA EL ALTA DE RPA**

OBSERVACIONES	

Hora de Alta: \_\_\_\_\_ Responsable: \_\_\_\_\_

**Figura 2.3** Registro post-anestésico

**Fuente:** Servicio de Anestesiología, Hospital de la Mujer

## 2.2. INGENIERÍA DE SOFTWARE

Existen diferentes definiciones sobre el término ingeniería de software, Ian Sommerville lo define como, “La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza”.

Según Sommerville (2005), para muchas personas el software son solo programas de computadora, sin embargo nos comenta que son todos aquellos documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera adecuada. Estos productos de software se desarrollan para algún cliente en particular o para un mercado en general. Para el diseño y desarrollo de proyectos de software se aplican métodos, modelos y técnicas que permiten resolver los problemas. En los años 50 no existían métodos de desarrollo, el desarrollo estaba a cargo de los propios programadores. De ahí la importancia de contar con analistas y diseñadores que permitieran un análisis adecuado de las necesidades que se deberían de implementar.

El objetivo principal que busca la ingeniería de software es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad y satisfaga las necesidades y expectativas del cliente. Según Gacitúa (2003), la Ingeniería de Software es un proceso intensivo de conocimiento, que abarca la captura de requerimientos, diseño, desarrollo, prueba, implantación y mantenimiento. Generalmente a partir de un complejo esquema de comunicación en el que interactúan usuarios y desarrolladores, el usuario brinda una concepción de la funcionalidad esperada y el desarrollador especifica esta funcionalidad a partir de esta primera concepción mediante aproximaciones sucesivas. Este ambiente de interacción motiva la búsqueda de estrategias robustas para garantizar que los requisitos del usuario serán descubiertos con precisión y que además serán expresados en una forma correcta y sin ambigüedad, que sea verificable, trazable y modificable.

### **2.2.1 Métodos de desarrollo de software.**

Una parte importante de la ingeniería de software es el desarrollo de métodos y modelos. En la actualidad ha habido muchos esfuerzos que se han encaminado al estudio de los métodos y técnicas para lograr una aplicación más eficiente de los métodos y lograr sistemas más eficientes y de mayor calidad con la documentación necesaria en perfecto orden y en el tiempo requerido. Gacitúa (2003), plantea que una metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema.

Los métodos de software tradicionalmente se dividen en métodos clásicos y métodos ágiles.

### **2.2.2. Métodos tradicionales o clásicas de software.**

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Métodos Tradicionales o Pesados.

Estos métodos tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso,

mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, los métodos tradicionales no se adaptan adecuadamente a los cambios, por lo que no son recomendables cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Entre los métodos tradicionales o pesados podemos citar:

- RUP, el proceso racional unificado RUP (*Rational Unified Process*) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, junto con el lenguaje unificado de modelado UML, constituye el método estándar más utilizado para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (productos tangibles del proceso) y roles.
- Microsoft Solutions Framework (MSF), es un conjunto de ingeniería de software de procesos, principios y prácticas probadas para lograr el éxito en el desarrollo del ciclo de vida del software. MSF proporciona una guía adaptable, basado en experiencias y mejores prácticas dentro y fuera de Microsoft, para aumentar las posibilidades de realización con éxito de una solución de tecnología de información para el cliente, con el objetivo de realizar el trabajo rápido, disminuyendo el número de personas en el equipo del proyecto, evitando los riesgos, al mismo tiempo generar una alta calidad de los resultados. Un proyecto MSF es regido por ciclos o iteraciones. Cada ciclo, cada componente del equipo ejecuta sus funciones y actualiza el resultado de su trabajo conforme la necesidad. Los ciclos se repiten hasta que el proyecto sea concluido o cada versión sea lanzada. Cada componente del equipo será responsable por uno o más papeles, dependiendo del tamaño o de la complejidad del proyecto.
- Iconix, es un método pesado-ligero de desarrollo del software que se halla a medio camino entre RUP y XP. Su fundamento es el hecho que en RUP un 80% de los casos pueden ser resueltos tan solo con un uso del 20% de UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico de UML, de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente. Se guía a través de casos de uso y sigue un

ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final.

### 2.2.3. Métodos ágiles de Software.

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de métodos de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de los denominados métodos ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.

Tras esta reunión se creó *The Agile Alliance*, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía "ágil", de éste los puntos más destacados son:

- Se valora al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funciona más que conseguir una buena documentación, aunque se parte de la base de que el software sin documentación es un desastre.
- Se valora la colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Aunque los creadores e impulsores de los métodos ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios enunciados anteriormente, cada método tiene características propias y hace hincapié en algunos aspectos más específicos. Entre los métodos conocidos como ágiles se pueden mencionar:

- Extreme Programming (XP)
- Scrum
- Familia de métodos Crystal

- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Lean Development (LD)
- Adaptative Software Development (ASD)

## **2.3. MÉTODO DE DESARROLLO ADAPTABLE DE SOFTWARE ASD.**

El método Desarrollo Adaptable de Software ASD (Adaptive software Development) fue desarrollado por Jim Highsmith y Sam Bayer a comienzos de 1990. Este método se adapta al cambio en lugar de luchar contra él. Se basa en la adaptación continua a circunstancias cambiantes. El desarrollo de software adaptable (Adaptive Software Development - ASD) es un método de desarrollo que hace énfasis en aplicar las ideas que se originaron en el mundo de los sistemas complejos, con una adaptación continua del proceso al trabajo.

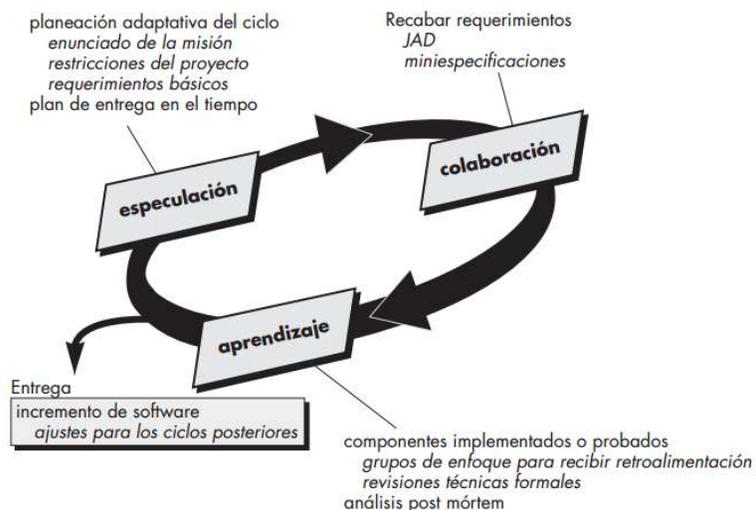
### **2.3.1. Características.**

Sus principales características son:

- Iterativo
- Orientado a los componentes de software (la funcionalidad que el producto va a tener, características, etc.) más que a las tareas en las que se va a alcanzar dicho objetivo.
- Tolerante a los cambios.
- Guiado por los riesgos
- La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

### **2.3.2. Ciclo de Vida.**

ASD utiliza un “cambio orientado hacia el ciclo de vida”, que tiene tres componentes que son: especular, colaborar y aprender.



**Figura 2.4** Método ASD, Ciclo de Vida

**Fuente:** Ingeniería de Software, un enfoque práctico. Pressman (2010)

### 2.3.2.1. Especular.

Una primera fase de iniciación para establecer los principales objetivos y metas del proyecto en su conjunto y comprender las limitaciones (zonas de riesgo) con las que operará el proyecto.

En ASD se realizan estimaciones de tiempo sabiendo que pueden sufrir desviaciones. Sin embargo, estas son necesarias para la correcta atención de los trabajadores que se mueven dentro de plazos de forma que puedan priorizar sus tareas.

Se decide el número de iteraciones para consumir el proyecto, prestando atención a las características que pueden ser utilizadas por el cliente al final de la iteración. Son por tanto necesarios, marcar objetivos prioritarios dentro de las mismas iteraciones.

Estos pasos se pueden volver a examinar varias veces antes de que el equipo y los clientes están satisfechos con el resultado.

### 2.3.2.2. Colaborar.

Es la fase donde se centra la mayor parte del desarrollo manteniendo una componente cíclica. Un trabajo importante es la coordinación que asegure que lo aprendido por un equipo se transmite al resto y no tenga que volver a ser aprendido por los otros equipos.

### **2.3.2.3. Aprender.**

La última etapa termina con una serie de ciclos de colaboración, su trabajo consiste en capturar lo que se ha aprendido, tanto positivo como negativo. Es un elemento crítico para la eficacia de los equipos.

Jim Highsmith identifica cuatro tipos de aprendizaje en esta etapa:

- Calidad del producto desde un punto de vista del cliente. Es la única medida legítima de éxito, pero, además, dentro de los métodos ágiles, los clientes tienen un valor importante.
- Calidad del producto desde un punto de vista de los desarrolladores. Se trata de la evaluación de la calidad de los productos desde un punto de vista técnico. Ejemplos de esto incluyen la adhesión a las normas y objetivos conforme a la arquitectura.
- La gestión del rendimiento. Este es un proceso de evaluación para ver lo que se ha aprendido mediante el empleo de los procesos utilizados por el equipo.
- Situación del proyecto. Como paso previo a la planificación de la siguiente iteración del proyecto, es el punto de partida para la construcción de la siguiente serie de características.

### **2.3.3. Ventajas.**

La tercera fase del ciclo de vida, revisión de los componentes, sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. Apunta hacia el Rapid Application Development (RAD), el cual enfatiza velocidad de desarrollo para crear un producto de alta calidad, bajo mantenimiento involucrando al usuario lo más posible.

Utiliza información disponible acerca de cambios para mejorar el comportamiento del software, promulga colaboración, la interacción de personas, anticipa cambios y trata automáticamente con ellos dentro de un programa en ejecución, sin la necesidad de un programador.

Es importante destacar que la filosofía ASD es meritoria sin importar el modelo del proceso empleado. La dinámica de la organización propia, los equipos, la colaboración interpersonal y el aprendizaje individual conducen a los grupos de proyectos de software con una mayor posibilidad de éxito.

#### 2.3.4. Desventajas.

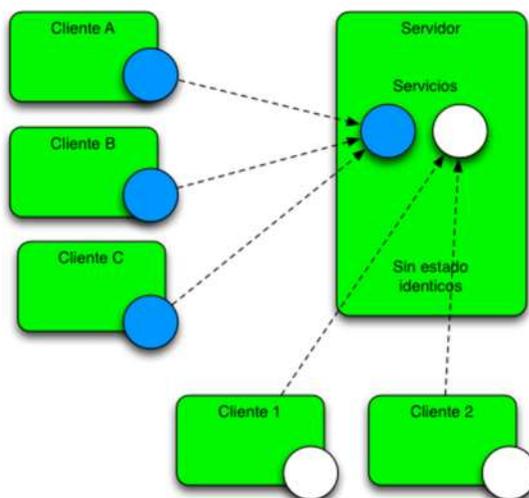
Aunque el ciclo entre el aprendizaje y la especulación es bueno permitiéndonos entregar productos con alta calidad, la prolongación de dicho ciclo por errores o cambios que no son detectados en reuniones anteriores afecta tanto a la calidad del producto como a su costo total.

Dado a que es un método ágil implica no realizar procesos que son requeridos en los métodos tradicionales o por lo menos no realizarlos en procesos diferentes, lo cual implica que empresas grandes las cuales necesitan llevar un mayor control a procesos y personas, tener tareas asignadas a un estado o proceso específico, y en las cuales dicho incremento de procesos no afectan en gran medida al costo final del producto, para dichas empresas el elegir un método tradicional resulta mucho más rentable tanto por el gran volumen de personal, de productos, y de costos que se manejan y para los cuales se tendrá un mayor control.

#### 2.4. SERVICIOS WEB REST.

Los servicios web REST (Representational State Transfer, Transferencia de estado representacional) son un estilo de arquitectura para desarrollar servicios que deben cumplir con las siguientes premisas:

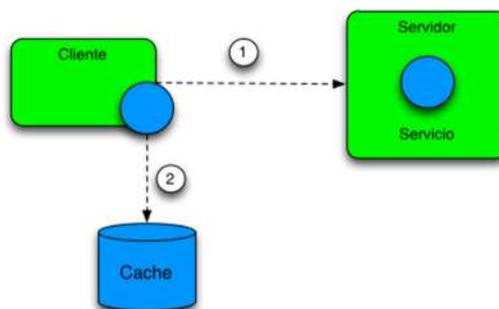
- **Cliente/Servidor:** como servicios web son cliente servidor y definen una interface de comunicaciones entre ambos separando las responsabilidades entre ambas partes.
- **Sin estado:** son servicios web que no mantienen estado asociado al cliente. Cada petición que se realiza a ellos es completamente independiente de la siguiente. Todas las llamadas al mismo servicio serán idénticas.



**Figura 2.5** Peticiones sin estado

**Fuente:** Arquitectura JAVA

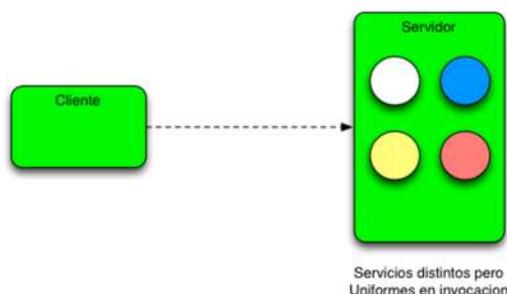
- **Cache:** El contenido de los servicios web REST se puede cachear de tal forma que una vez realizada la primera petición al servicio, el resto puedan apoyarse en la cache si fuera necesario.



**Figura 2.6** Caché intermedio

**Fuente:** Arquitectura JAVA

- **Servicios Uniformes:** Todos los servicios REST compartirán una forma de invocación y métodos uniforme, utilizando los métodos GET, POST, PUT, DELETE.



**Figura 2.7** Servicios uniformes REST

**Fuente:** Arquitectura JAVA

- **Uso de métodos HTTP de manera explícita:** Una de las características claves de los servicios web REST es el uso explícito de los métodos HTTP, siguiendo el protocolo definido por RFC 2616. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:
  - Se usa POST para crear un recurso en el servidor.
  - Se usa GET para obtener un recurso.
  - Se usa PUT para cambiar el estado de un recurso o actualizarlo.
  - Se usa DELETE para eliminar un recurso.
- **Arquitectura en Capas:** Todos los servicios REST están orientados hacia la escalabilidad y un cliente REST no será capaz de distinguir entre sí está realizando una petición directamente al servidor, o se lo está devolviendo un sistema de caches intermedio o por ejemplo existe un balanceador que se encarga de redirigirlo a otro servidor.

## 2.5. INGENIERÍA WEB

Con la difusión y avance que obtuvo la World Wide Web e internet, se ha logrado que numerosas actividades giren entorno y dependan de los servicios que ofrecen estas, como ser variedad de contenido y funcionalidades, que responden a las necesidades de los usuarios.

Con este cambio, dentro de la administración de empresas se vio un giro trascendental, como ser, la interacción con los clientes sin la necesidad de atenderlos personalmente y a la vez reduciendo costos en diversos aspectos dentro de la empresa. De esta manera, las empresas han optado por la implementación de sistemas de información en la web.

### 2.5.1 Concepto

La ingeniería web al igual que la ingeniería de software, aplica tanto métodos, técnicas y herramientas para el desarrollo de la solución a un problema, pero a diferencia de la ingeniería de software, cumple características para el desarrollo de sistemas web de gran complejidad y dimensión.

Si bien la ingeniería de software brinda fases y pasos a seguir para el desarrollo de un sistema de información o software; un sistema web requiere un trato diferente, por las especificaciones y características que conlleva. Powell (1998 citado en Pressman, 2010) resume que los sistemas web “implican una mezcla de publicación impresa y desarrollo de software, de marketing e informática, de comunicaciones internas y relaciones externas, y de arte y tecnología”.

### 2.5.2 Sistema Web

Un sistema web o WebApp, es una aplicación que no depende de una plataforma, sino que está desarrollada para ser implementada en un servidor web con características y capacidades mayores a una página web.

R. Pressman (2010), señala que las características de una WebApp son las siguientes:

- Inmediatez, Las aplicaciones basadas en Web tienen una inmediatez que no se encuentra en otros tipos de software. Es decir, el tiempo que se tarda en comercializar un sitio Web completo puede ser cuestión de días o semanas. Los desarrolladores deberán utilizar los métodos de planificación, análisis, diseño, implementación y comprobación que se hayan adaptado a planificaciones apretadas en tiempo para el desarrollo de WebApps.
- Seguridad, Dado que las WebApps están disponibles a través del acceso por red, es difícil, si no imposible, limitar la población de usuarios finales que pueden acceder a la aplicación. Con objeto de proteger el contenido confidencial y de proporcionar formas seguras de transmisión de datos, deberán implementarse fuertes medidas de seguridad en toda la infraestructura que apoya una WebApp y dentro de la misma aplicación.
- Estética, Una parte innegable del atractivo de una WebApp es su apariencia e interacción. Cuando se ha diseñado una aplicación con el fin de comercializarse o vender productos o ideas, la estética puede tener mucho que ver con el éxito del diseño técnico.

### 2.5.3 Arquitectura Modelo Vista Controlador (MVC)

En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.



**Figura 2.8** Patrón MVC

**Fuente:** DesarrolloWeb.com

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, MVC es un patrón que ya tiene varias décadas y fue presentado incluso antes de la aparición de la web. No obstante, en los últimos años ha ganado mucha fuerza y popularidad gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

#### 2.5.3.1 Capa del Modelo

El modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.

A primera vista los objetos del modelo pueden ser considerados como la primera capa de la interacción con cualquier base de datos que podría estar utilizando la aplicación. Pero en general representan los principales conceptos en torno a los cuales se desea implementar un programa.

### ***2.5.3.2 Capa de la Vista***

La vista hace una presentación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.

Por ejemplo, como la capa de modelo devuelve un conjunto de datos, la vista los usaría para hacer una página HTML que los contenga. O un resultado con formato XML para que otras aplicaciones puedan consumir.

La capa de la Vista no se limita únicamente a HTML o texto que represente los datos, sino que puede ser utilizada para ofrecer una amplia variedad de formatos en función de sus necesidades tales como videos, música, documentos y cualquier otro formato que puedas imaginar.

### ***2.5.3.3 Capa del Controlador***

La capa del controlador gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

Los controladores pueden ser vistos como administradores cuidando de que todos los recursos necesarios para completar una tarea se deleguen a los trabajadores más adecuados. Espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista.

## 2.6 MÉTODO UWE

El método UWE (Ingeniería Web basada en UML), pertenece a los métodos de apoyo a la ingeniería de software y si bien está basado en el modelado UML, tiene como objetivo y ventaja la adaptación de este modelado a sistemas o aplicaciones web, además de utilizar múltiples herramientas o componentes que pertenecen a UML.

UWE, es un proceso interactivo e incremental, que se complementa con los procesos o fases de UML y comparten ciertos elementos de los diagramas que se utilizan en el modelado del diseño; en otras palabras, se puede decir que es una extensión que apoya al diseño basado en la web, de esta forma describe el método UWE el Instituto de Informática, Ludwig Maximilians University Munich (2012).

### 2.6.1 Principales modelos

Los modelos principales en los que se basa este método son:

#### 2.6.1.1 *Modelo de Requerimientos*

Es el encargado de la captura y procesamiento de requerimientos funcionales de la aplicación web, es por esta razón que el método UWE propone dos partes importantes para el modelado de los requerimientos entre los cuales tenemos a los casos de uso y los diagramas de actividades en caso que sea necesario mayor detalle de un caso de uso mediante un proceso.

#### 2.6.1.2 *Modelo de Contenido*

Este modelo es utilizado para representar los conceptos que son relevantes en el dominio de la aplicación y las diferentes relaciones de los mismos, se basa en el análisis de requisitos reflejados en los casos de uso.

#### 2.6.1.3 *Modelo de Navegación*

Este modelo representa los segmentos de navegación de la aplicación web y los enlaces entre ellos. Para este modelo se utiliza un diagrama de clases el cual utilizará estereotipos específicos del método.

#### **2.6.1.4 Modelo de Presentación**

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

### **2.6.2 Etapas del método UWE**

#### **2.6.2.1 Captura, análisis y especificación de requisitos**

Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

#### **2.6.2.2 Diseño del sistema**

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

#### **2.6.2.3 Codificación del software**

Durante esta etapa se realizan las tareas que se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

#### **2.6.2.4 Pruebas**

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

#### **2.6.2.5 Implementación**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

### **2.6.2.6 Mantenimiento y seguimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

Una vez puesta la Pagina Web a Disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio.

## **2.7 HERRAMIENTAS DE DESARROLLO**

Escoger las herramientas adecuadas es de suma importancia para el desarrollo de un proyecto de software, se debe considerar la disponibilidad, portabilidad, eficiencia y seguridad que pueden aportar.

Considerando las delimitaciones y entorno de trabajo del presente proyecto se optó por el uso de las siguientes herramientas de desarrollo:

### **2.7.1 Framework CodeIgniter**

CodeIgniter es una aplicación web realizada en el lenguaje de programación PHP con el objetivo de servir como marco de trabajo para construir otras aplicaciones web bajo una cierta arquitectura y un lenguaje de programación como PHP. Siguiendo las principales características de los framework de desarrollo, CodeIgniter está compuesto por un conjunto de librerías que se utilizan para construir las aplicaciones siguiendo los patrones que establece el framework para conseguir mayores resultados en las mismas.

Algunos de los factores más importantes que definen a CodeIgniter son:

- Patrón de diseño MVC, CodeIgniter se basa en el patrón Modelo Vista Controlador que provee una forma eficiente para desarrollar aplicaciones separando la lógica del proyecto de la gestión de datos y presentación de interfaces.

- Versatilidad, CodeIgniter funciona correctamente en disímiles ambientes de despliegue, incluso en ambientes compartidos, donde solo tenemos acceso mediante FTP y no se posee una administración total del mismo.
- Compatibilidad, el framework es compatible con las principales versiones de PHP, lo cual permite poder desplegado sin problemas en diferentes ambientes.
- Facilidad de instalación, la instalación en los diferentes ambientes se realiza solamente copiando los archivos fuentes del proyecto hacia el ambiente de despliegue. Una vez copiado dichos ficheros, se modifica el fichero de configuración de la base de datos y automáticamente, la aplicación queda lista para ser utilizada.
- Flexibilidad, es un framework de desarrollo extremadamente flexible, es decir, si se quiere seguir las pautas que define el framework, logrará un desarrollo guiado y robusto, pero si se quiere tener toda la libertad posible, el framework le permite utilizar las pautas que defina conveniente.
- Ligereza, el núcleo de CodeIgniter es extremadamente ligero, lo cual posibilita que los tiempos de respuesta sean sumamente atractivos. A partir de exámenes de velocidad de respuesta que se le han realizado al framework en diferentes situaciones, se ha llegado a la conclusión de que CodeIgniter es uno de los framework de desarrollo web sobre PHP más rápido que existen.
- Documentación, la documentación de CodeIgniter como framework de desarrollo web es sumamente fácil de acceder, podemos encontrarla fácilmente en toda la web o más específicamente en su sitio oficial (<http://www.codeigniter.com>). Una gran ventaja con que cuenta la documentación oficial del framework, es que posee una guía totalmente enlazada y permite acceder a sus principales tópicos a través de búsquedas inteligentes.

### **2.7.2 Gestor de base de datos MySQL**

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Entre sus principales características podemos mencionar:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Es una de las herramientas más utilizadas por los programadores orientados a Internet.
- Fácil instalación y configuración.

Además ofrece:

- Escalabilidad y flexibilidad
- Alto rendimiento
- Alta disponibilidad
- Robusto soporte transaccional.
- Fortalezas en web y Data Warehouse.
- Fuerte protección de datos.

### 2.7.3 Android Studio

Android Studio es un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno Eclipse, hasta ahora el IDE más utilizado.

Al crear un nuevo proyecto en Android Studio, la estructura del proyecto aparece con casi todos los archivos dentro del directorio SRC, un cambio a un sistema de generación basado Gradle

que proporcionará una mayor flexibilidad para el proceso de construcción. Además, gracias a su sistema de emulación integrado, Android Studio permite ver los cambios que realizamos en nuestra aplicación en tiempo real, pudiendo además comprobar cómo se visualiza en diferentes dispositivos Android con distintas configuraciones y resoluciones de forma simultánea.

Entre las muchas características de Android Studio destacan sus herramientas de empaquetado y etiquetado de código para organizarnos al implementar grandes cantidades de código, sirviéndose además de un sistema drag & drop para mover los componentes a través de la interfaz de usuario. Además, este nuevo entorno cuenta con Google Cloud Messaging, una funcionalidad con la que podremos enviar datos desde el servidor a terminales Android a través de la nube, siendo esta una forma de enviar notificaciones Push a nuestras apps. Por otro lado, también nos ayudará en la localización de nuestras aplicaciones, dándonos una forma más visual de seguir programando y controlar el flujo de la aplicación.

¿Qué ofrece Android Studio?

- Un entorno de desarrollo claro y robusto.
- Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- Asistentes y plantillas para los elementos comunes de programación en Android.
- Un completo editor con muchas herramientas extra para agilizar el desarrollo de nuestras aplicaciones.

## 2.8 CALIDAD DE SOFTWARE ISO 9126

ISO 9126 es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

Enfatiza tres puntos importantes:

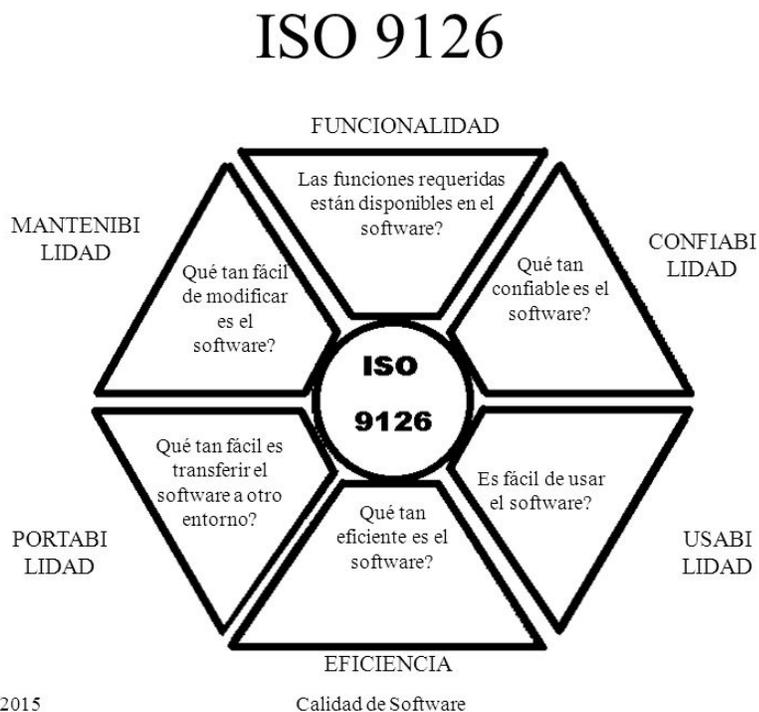
- Los requisitos del software constituyen el fundamento para medir la calidad. La carencia de conformidad con los requisitos es carencia de calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la manera en que el software se somete al trabajo ingenieril. Si no se siguen los criterios, la carencia de calidad será un resultado casi seguro.

- Existe un conjunto de requisitos implícitos que a menudo no se mencionan (por ejemplo, mantenibilidad). Si el software se conforma con los requisitos explícitos pero falla en atender los requisitos implícitos, la calidad del software es sospechosa.

Debido a que la calidad se compone, realmente, de muchas características, la noción de calidad se captura, usualmente, en un modelo que retrata las características compuestas y sus relaciones.

Los modelos antiguos fueron McCall (1977) y Boehm et al. (1978). Ambos modelos son jerárquicos con refinamientos sucesivos que culminan en primitivas que se pueden medir directamente.

ISO 9126 define la Calidad del Software como: “La totalidad de características de un producto de software que se manifiesta en su habilidad para satisfacer necesidades establecidas o implícitas”. Se relaciona con seis factores: funcionalidad, fiabilidad, eficiencia, usabilidad, mantenibilidad y portabilidad.



**Figura 2.9** ISO 9126

**Fuente:** SmartSys TIC's para PYME's

### 2.8.1 Funcionalidad

Conjunto de atributos que soporta la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son tales que satisfacen las necesidades implícitas o establecidas.

Subcaracterísticas:

- Idoneidad, hace referencia a que si el software desempeña las tareas para las que fue desarrollado.
- Exactitud, evalúa el resultado final que obtiene el software y si tiene consistencia a lo que se espera de él.
- Interoperabilidad, consiste en revisar si el sistema puede interactuar con otro sistema independiente.
- Seguridad, verifica si el sistema puede impedir el acceso a personal no autorizado.

### 2.8.2 Fiabilidad

El conjunto de atributos que soporta la capacidad del software para mantener su nivel de rendimiento bajo condiciones establecidas por un periodo de tiempo establecido.

Subcaracterísticas:

- Madurez, se debe verificar las fallas del sistema y si muchas de estas han sido eliminadas durante el tiempo de pruebas o uso del sistema.
- Recuperabilidad, verificar si el software puede reasumir el funcionamiento y restaurar datos perdidos después de un fallo ocasional.
- Tolerancia a fallos, evalúa si la aplicación desarrollada es capaz de manejar errores.

### 2.8.3 Usabilidad

El conjunto de atributos que soporta el esfuerzo necesario para el uso y la evaluación individual de tal uso mediante un conjunto de usuarios establecidos e implícitos.

Subcaracterísticas:

- Aprendizaje, determina que tan fácil es para el usuario aprender a utilizar el sistema.

- Comprensión, evalúa que tan fácil es para el usuario comprender el funcionamiento del sistema.
- Operatividad, determina si el usuario puede utilizar el sistema sin mucho esfuerzo.
- Atractividad, verifica que tan atractiva se ve la interfaz de la aplicación.

#### **2.8.4 Eficiencia**

El conjunto de atributos que soporta las relaciones entre el nivel de rendimiento del software y el monto de recursos empleados, bajo condiciones establecidas.

Subcaracterísticas:

- Comportamiento en el tiempo, verifica la rapidez en que responde el sistema.
- Comportamiento de recursos, determina si el sistema utiliza los recursos de manera eficiente.

#### **2.8.5 Mantenibilidad**

El conjunto de atributos que soporta el esfuerzo necesario para realizar modificaciones especificadas.

Subcaracterísticas:

- Estabilidad, verifica si el sistema puede mantener su funcionamiento a pesar de realizar cambios.
- Facilidad de análisis, determina si la estructura de desarrollo es funcional con el objetivo de diagnosticar fácilmente las fallas.
- Facilidad de cambio, verifica si el sistema puede ser fácilmente modificado.
- Facilidad de pruebas, evalúa si el sistema puede ser probado fácilmente.

#### **2.8.6 Portabilidad**

El conjunto de atributos que soporta la habilidad del software para transferirlo de un entorno a otro.

Subcaracterísticas:

- Capacidad de instalación, verifica si el software se puede instalar fácilmente.

- Capacidad de reemplazamiento, determina la facilidad con la que el software puede reemplazar otro software similar.
- Adaptabilidad, el software se puede trasladar a otros ambientes.
- Co-existencia, el software puede funcionar con otros sistemas.

## 2.9 MÉTODO DE ESTIMACIÓN DE COSTO DEL SISTEMA – CRI

Existen diversos métodos tradicionales de estimación de costos, como el cálculo de puntos función que determinan el costo en base a criterios como entradas externas, salidas externas, archivos internos, etc.

Sin embargo Sergio Gomez (2013) propone el método CRI, el cuál toma en cuenta los siguientes elementos:

- Características de los campos de la base de datos (C)
- Relaciones entre tablas (R)
- Cantidad de vistas, consultas, procedimientos almacenados, para visualizar datos en pantalla o en impresiones (I)

El costo es un resultado de la sumatoria de las unidades de trabajo que requiere cada elemento multiplicado por el valor monetario por hora de trabajo pretendido por el desarrollador.

Definiendo la unidad de trabajo UT como una medida de esfuerzo y tiempo, entonces tenemos:

$$\text{Costo} = (UT_C + UT_R + UT_I) * \text{Valor monetario}(Bs/h)$$

Donde:

a)  $UT_C$ : Unidades de trabajo de los campos

Se define mediante la fórmula:

$$UT_C = \sum_i^n \text{Complejidad del campo } i$$

**Tabla 2.1** Complejidad por tipo de dato

Tipo de dato	Complejidad [1:Normal 2:Media 3:Alta]
VARCHAR	1
INT, DOUBLE	2
DATE, TIME	3

b)  $UT_R$ : Unidades de trabajo de las relaciones

Definido como:

$$UT_R = \sum_1^m (Nro \text{ de tablas de la entidad compuesta}_m)$$

- Donde m es el número de entidades compuestas
- Estas entidades compuestas darán lugar a las relaciones entre tablas

c)  $UT_I$ : Unidades de trabajo de los impresos

Definido como:

$$UT_I = \sum_1^r (Nro \text{ de tablas en el reporte}_r)$$

Donde r es el número de reportes requeridos.

## CAPITULO III. INGENIERÍA Y APLICACIÓN DE MÉTODOS

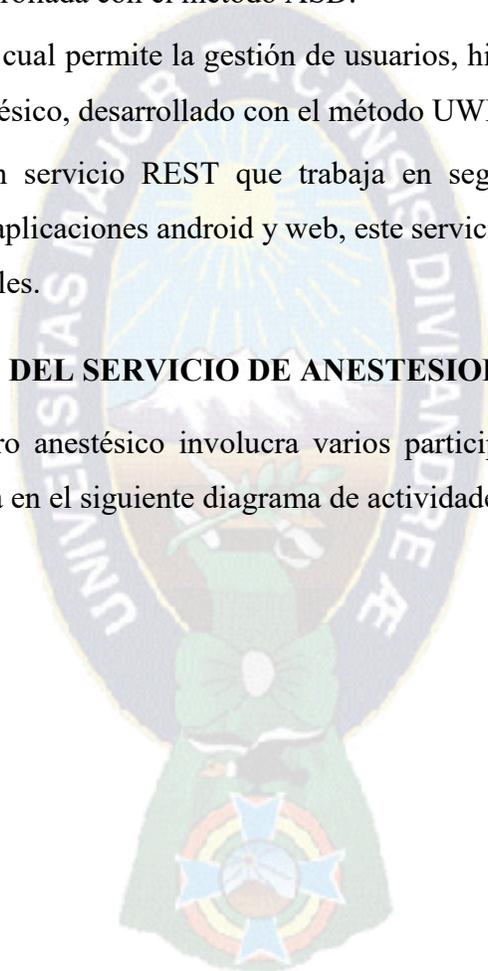
### 3.1 INTRODUCCION

Se procederá a detallar las diferentes etapas y actividades realizadas para el diseño y la implementación del SIGRA, que integra los siguientes componentes principales:

- a) La aplicación android encargada del registro pre y post anestésico y las consultas a registros pre anestésicos, desarrollada con el método ASD.
- b) La aplicación web la cual permite la gestión de usuarios, historiales y reportes y también el registro trans anestésico, desarrollado con el método UWE.
- c) También se tiene un servicio REST que trabaja en segundo plano para permitir la comunicación de las aplicaciones android y web, este servicio se desarrolló como una API restful y url's amigables.

### 3.2 SITUACION ACTUAL DEL SERVICIO DE ANESTESIOLOGIA

El proceso de registro anestésico involucra varios participantes con diferentes roles y actividades, como se muestra en el siguiente diagrama de actividades:



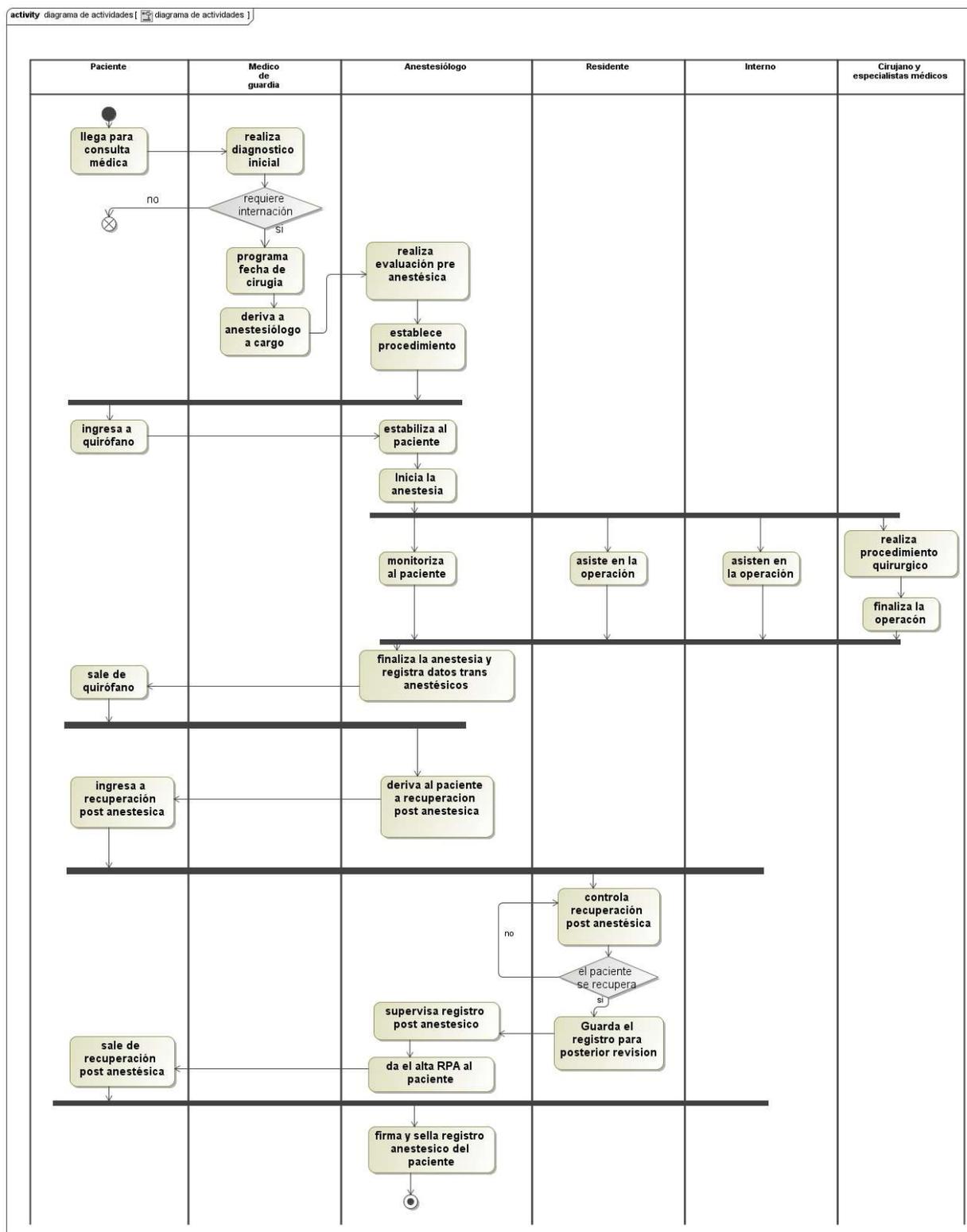


Figura 3.1 proceso de registro anestésico

El servicio de anestesiología se compone de médicos anesestesiólogos, médicos residentes e internos. El resto del personal de quirófano lo complementan: el personal de instrumentación, enfermería, el cirujano, los asistentes y especialistas.

El procedimiento se desarrolla de la siguiente manera:

1. El paciente llega para consulta externa o por Unidad de emergencia.
2. El médico de guardia realiza una valoración inicial del paciente.
3. Según el estado del paciente, el médico de guardia lo deriva a un especialista que determina si es necesario programar una cirugía, lo cual también implica iniciar el registro anestésico del paciente.
4. Se programa la fecha para la intervención quirúrgica y el paciente pasa a internación o directamente a quirófano dependiendo de la gravedad de su condición.
5. El anesestesiólogo a cargo registra las patologías, antecedentes y otras observaciones en el registro pre anestésico, esto a fin de establecer el grado de riesgo y las consideraciones especiales del paciente. También pueden requerirse exámenes adicionales.
6. Se procede a la operación quirúrgica donde intervienen las enfermeras, el anesestesiólogo, el médico residente, el instrumentador, el cirujano y los especialistas necesarios para el caso.
7. Durante la operación el anesestesiólogo va consignando en el registro trans anestésico los procedimientos utilizados, el uso de medicamentos, las observaciones y los signos vitales del paciente.
8. Una vez concluida la operación, el paciente pasa a la sala de recuperación donde el anesestesiólogo, el médico residente o un interno registran el control post operatorio (registro post anestésico), cabe aclarar que sólo el anesestesiólogo a cargo puede certificar el alta del paciente.
9. Posteriormente, una vez que el paciente es dado de alta de la recuperación post anestésica, el anesestesiólogo a cargo emite su firma y sello para validar el registro anestésico del paciente con lo cual ya pasa a ser un documento legal y se procede al traslado del paciente a la unidad de internación con el propósito de que continúe su recuperación final y alta correspondiente.

### 3.3. DETERMINACION DE REQUISITOS

Considerando la situación actual del servicio de Anestesiología se contempló el diseño de dos tipos de aplicaciones: una aplicación android para los registros pre y post anestésicos que también permita consultar otros registros pre anestésicos y una aplicación web para el registro trans anestésico que además pueda gestionar los registros anestésicos y proporcionar reportes impresos. Se contempla también el uso de un servidor para brindar acceso constante a la aplicación y la implementación de un servicio web basado en REST para permitir la comunicación de ambas plataformas.

Los requisitos funcionales que debe contemplar el sistema se resumen en:

- El sistema debe proveer autenticación de la identidad de un usuario.
- El acceso al sistema debe ser diferenciado.
- Los internos deben ingresar con una clave común.
- El proceso de registro pre anestésico debe incluir una evaluación completa de patologías.
- El sistema debe permitir al anestesiólogo realizar la evaluación pre anestésica ya sea que el paciente ingrese por internación o por unidad de emergencia.
- En el registro trans anestésico debe registrarse los procedimientos y técnicas utilizadas, los participantes, su especialidad y el rol que cumplen en la intervención quirúrgica.
- En el registro post anestésico se deben registrar los signos vitales, los equipos usados, las medicaciones utilizadas, la hora de ingreso, hora de control, hora de alta y la valoración del paciente en las escalas de evaluación post anestésica (Aldrede/Kroulik, Bromage, ambulatorio)
- El sistema debe permitir la gestión de usuarios y administradores.
- El sistema debe permitir la gestión de registros históricos del paciente.
- El sistema debe registrar intervenciones de pacientes antiguos.
- El sistema debe permitir consultar, gestionar o imprimir reportes de pacientes o usuarios del sistema.

Los requisitos no funcionales son:

- El sistema debe facilitar la movilidad al realizar la evaluación pre anestésica.

- El sistema debe facilitar la movilidad al realizar la evaluación de recuperación post anestésica.
- En el registro trans anestésico debe haber flexibilidad para registrar la hora y fecha de control.
- Para agilizar el proceso de registro anestésico, el sistema debe mostrar sugerencias de auto completado para diagnósticos, técnicas, procedimientos, uso de agentes y observaciones.
- La gestión de usuarios, pacientes e intervenciones debe ser clara, sin dar lugar a ambigüedades.
- Los reportes impresos deben contener la misma información que los visualizados al llenar el registro anestésico.

### 3.4 Aplicación Android – Método De Desarrollo Adaptable De Software (ASD)

El método ASD utiliza un ciclo de desarrollo dinámico conocido como Especulación – Colaboración – Aprendizaje, cada ciclo está dedicado a un constante aprendizaje y una estrecha colaboración con el Servicio de Anestesiología del Hospital de la Mujer.

La tabla 3.1 describe las actividades realizadas para lograr la creación de un registro pre anestésico, post anestésico y consultas de otros registros pre anestésicos:

**Tabla 3.1** Actividades por fase del método ASD

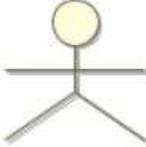
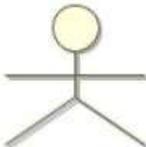
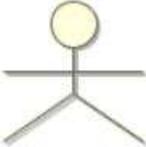
Especulación	Colaboración	Aprendizaje
<ul style="list-style-type: none"> <li>• Se identifican los actores y el grado de interacción con el sistema</li> <li>• Se determina los requisitos y condiciones mediante casos de uso.</li> <li>• En base al aprendizaje de los ciclos anteriores se definen las iteraciones necesarias para el proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Se desarrolla la arquitectura del sistema.</li> <li>• Se desarrollan incrementalmente 3 módulos para las funcionalidades de registro pre anestésico, registro post anestésico y consultas de registro pre anestésico.</li> </ul>	<ul style="list-style-type: none"> <li>• Se realizan pruebas funcionales al término de la iteración para probar la calidad con el cliente.</li> <li>• Se establecen pautas para mejorar las interfaces de usuario.</li> <li>• Se prepara el sistema para su implementación.</li> <li>• Se despliega la aplicación.</li> </ul>

### 3.4.1 Fase de Especulación

#### 3.4.1.1 Identificación de actores

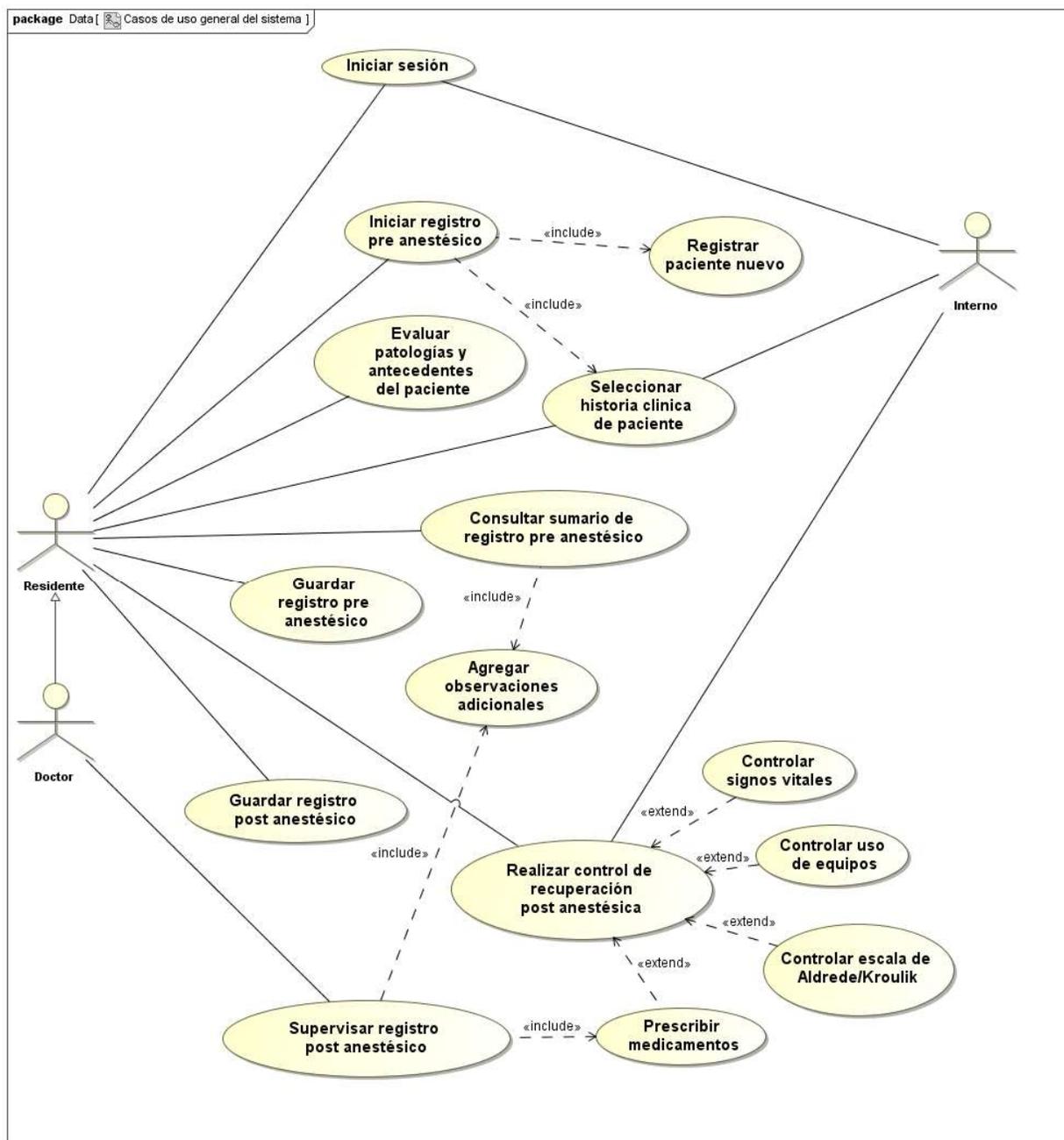
Se pueden identificar tres tipos de usuarios que interactuarán con el sistema como se muestra en la tabla 3.2:

**Tabla 3.2** Definición de actores – aplicación android

Actor	Descripción
 <p><b>Interno</b></p>	<p>Se refiere a los estudiantes de medicina que hacen su último año de prácticas, este actor solo puede evaluar la recuperación post anestésica del paciente pero no puede dar el alta.</p>
 <p><b>Residente</b></p>	<p>Se refiere a los profesionales médicos que hacen su residencia previo a obtener la especialidad en anestesiología, este actor puede crear y hacer consultas de registros pre anestésicos y evaluar la recuperación post anestésica del paciente pero no puede dar el alta.</p>
 <p><b>Doctor</b></p>	<p>Se refiere a los profesionales médicos con especialidad en anestesiología, este actor realiza las mismas actividades que el actor residente además de supervisar las evaluaciones post anestésicas de los residentes e internos. Solo el usuario Doctor puede certificar y dar el alta.</p>

### 3.4.1.2 Casos de uso.

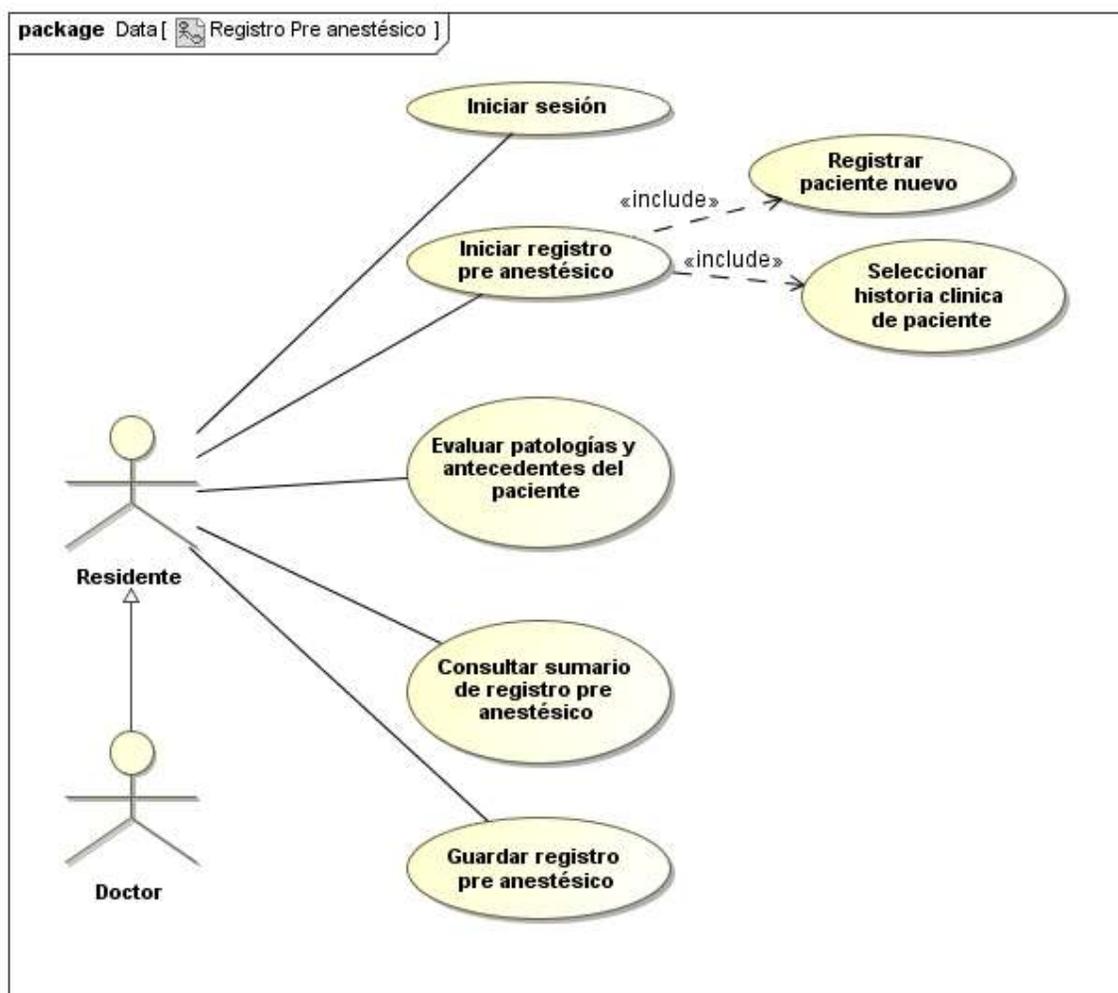
La figura 3.3 muestra el diagrama de casos de uso general del sistema, el cual describe de forma global las actividades llevadas a cabo por los actores.



**Figura 3.3** Diagrama de casos de uso general del sistema

A continuación se muestran los casos de uso que participan en los diferentes componentes de la aplicación android:

a) **Proceso de registro pre anestésico:**



*Figura 3.4* Casos de uso del proceso de registro pre anestésico

Descripción de casos de uso:

**Tabla 3.3** Caso de uso “Iniciar sesión”

<b>Escenario</b>	Caso de uso “Iniciar sesión”
<b>Código</b>	CU-PRE-01
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Los usuarios previamente deben tener asignados sus claves de acceso.</li> <li>• La aplicación tiene acceso al servicio REST para obtener el listado de usuarios permitidos y sus claves de acceso.</li> </ul>
<b>Flujo Normal</b>	<p>El usuario selecciona su nombre de usuario correspondiente e ingresa su clave asignada. Si se trata de un usuario Interno sólo debe introducir su clave de acceso de internos.</p> <p>Si la clave es correcta el usuario puede proceder a elegir una actividad:</p> <ul style="list-style-type: none"> <li>• Crear un registro pre anestésico</li> <li>• Consultar un registro pre anestésico</li> <li>• Iniciar control de recuperación post anestésica</li> </ul> <p>Además si el usuario es Doctor también puede supervisar y dar de alta un registro post anestésico previamente llenado.</p>
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• Si el usuario introduce una clave errónea, la aplicación denegará el acceso y mostrará un mensaje de error.</li> <li>• Si no hay acceso al servicio REST, la aplicación intentará conectarse 4 o 5 veces al cabo de los cuales mostrará un mensaje que no hay conexión y se terminará la ejecución.</li> </ul>
<b>Postcondición</b>	Se concede acceso al usuario, el cual puede proceder con las siguientes actividades.

**Tabla 3.4** Caso de uso “Iniciar registro pre anestésico”

<b>Escenario</b>	Caso de uso “Iniciar registro pre anestésico”
<b>Código</b>	CU-PRE-02
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario ha sido validado correctamente.</li> <li>• Se cuenta con acceso al servicio REST para obtener los recursos para proceder al registro.</li> </ul>
<b>Flujo Normal</b>	El usuario accede a una interfaz donde puede elegir entre registrar un paciente nuevo o seleccionar la historia clínica de un paciente antiguo, luego se procede a llenar los datos iniciales de evaluación como ser: presión arterial, edad, peso, diagnóstico, especialidad, etc.
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• Si el usuario introduce valores no permitidos o que no están en rango, la aplicación resaltará en rojo los campos no válidos antes de poder proceder con el registro.</li> <li>• Si no se puede acceder al servicio REST, no estará disponible la opción de continuar con el proceso y la aplicación terminará.</li> </ul>
<b>Postcondición</b>	Se tienen los datos iniciales y se puede proceder a la evaluación de patologías y antecedentes del paciente.

**Tabla 3.5** Caso de uso “Registrar paciente nuevo”

<b>Escenario</b>	Caso de uso “Registrar paciente nuevo”
<b>Código</b>	CU-PRE-03
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario ha iniciado un registro pre anestésico.</li> <li>• Se cuenta con acceso al servicio REST para validar que no se repita el número de historia clínica y que el paciente no ha sido previamente registrado.</li> </ul>
<b>Flujo Normal</b>	El usuario registra los datos personales del paciente y su número de historia clínica (que ya fue asignado en admisiones).

	Si no se cuenta con los datos personales del paciente se marca como “N.N.”
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• Si se intenta registrar un paciente o una historia clínica existente, la aplicación mostrará una advertencia para verificar los datos.</li> <li>• Si no se puede acceder al servicio REST, no se podrá validar si el paciente existe o no y no se podrá continuar con el proceso.</li> </ul>
<b>Postcondición</b>	Se tiene registrado un paciente nuevo y se puede continuar con el proceso.

**Tabla 3.6** Caso de uso “Seleccionar historia clínica de paciente”

<b>Escenario</b>	Caso de uso “Seleccionar historia clínica de paciente”
<b>Código</b>	CU-PRE-04
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario Doctor o usuario Residente ha iniciado un registro pre anestésico.</li> <li>• El usuario Doctor o usuario Residente está haciendo una consulta de registro pre anestésico.</li> <li>• El usuario Interno ha sido validado correctamente para hacer el control de la recuperación post anestésica.</li> <li>• Se cuenta con acceso al servicio REST para obtener el listado de pacientes e historiales existentes.</li> </ul>
<b>Flujo Normal</b>	El usuario selecciona un paciente y su historial clínico registrado previamente.
<b>Flujo Alternativo</b>	Si no se puede acceder al servicio REST, no estará disponible el listado de pacientes.
<b>Postcondición</b>	Se tiene seleccionado un paciente para proceder a la evaluación.

**Tabla 3.7** Caso de uso “Evaluar patologías y antecedentes del paciente”

<b>Escenario</b>	Caso de uso “Evaluar patologías y antecedentes del paciente”
<b>Código</b>	CU-PRE-05
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario ha registrado correctamente los datos iniciales del pre anestésico.</li> <li>• Se cuenta con acceso al servicio REST para obtener los recursos necesarios para proceder a la evaluación.</li> </ul>
<b>Flujo Normal</b>	El usuario va marcando las diferentes patologías que pueda presentar el paciente, a través de varias pantallas clasificadas por categorías, también registra los resultados del laboratorio de estar disponibles.
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• Si no se puede acceder al servicio REST, no se podrá continuar con el proceso y la aplicación terminará.</li> </ul>
<b>Postcondición</b>	Se completó la evaluación de patologías correctamente y puede proceder al sumario del registro pre anestésico.

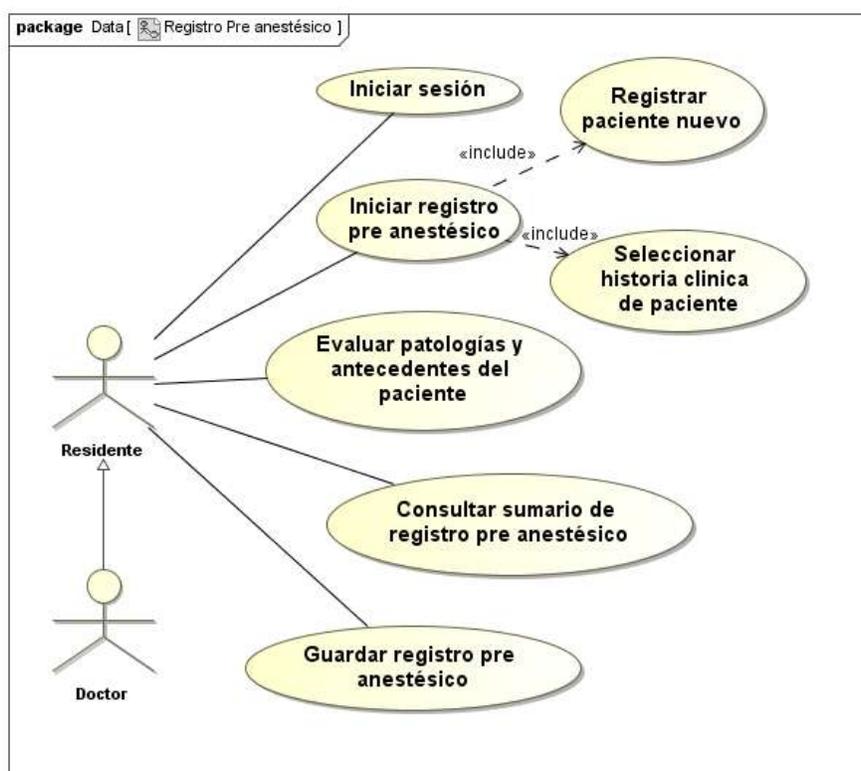
**Tabla 3.8** Caso de uso “Consultar sumario de registro pre anestésico”

<b>Escenario</b>	Caso de uso “Consultar sumario de registro pre anestésico”
<b>Código</b>	CU-PRE-06
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario ha completado correctamente el registro pre anestésico.</li> <li>•</li> </ul>
<b>Flujo Normal</b>	El usuario accede a una interfaz donde se presenta un sumario de todos los datos consignados en el registro pre anestésico para su verificación y también puede ingresar observaciones pertinentes.
<b>Flujo Alternativo</b>	El usuario tiene la posibilidad de volver atrás para corregir algún dato erróneo.
<b>Postcondición</b>	Se tienen validados correctamente los datos del proceso de registro pre anestésico del paciente

**Tabla 3.9** Caso de uso “Guardar registro pre anestésico”

<b>Escenario</b>	Caso de uso “Guardar registro pre anestésico”
<b>Código</b>	CU-PRE-07
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario validó correctamente los datos del registro pre anestésico.</li> <li>• Se cuenta con acceso al servicio REST para poder registrar la evaluación pre anestésica en el servidor.</li> </ul>
<b>Flujo Normal</b>	El usuario registra en el servidor los datos de la evaluación pre anestésica.
<b>Flujo Alternativo</b>	Si no se puede acceder al servicio REST, se mostrará un mensaje para que el usuario intente de nuevo la conexión.
<b>Postcondición</b>	<ul style="list-style-type: none"> <li>• La evaluación pre anestésica ha sido registrada correctamente en el servidor.</li> <li>• Se cierra la sesión del usuario, devolviéndolo a la pantalla inicial.</li> </ul>

## b) Proceso de consulta de registro pre anestésico



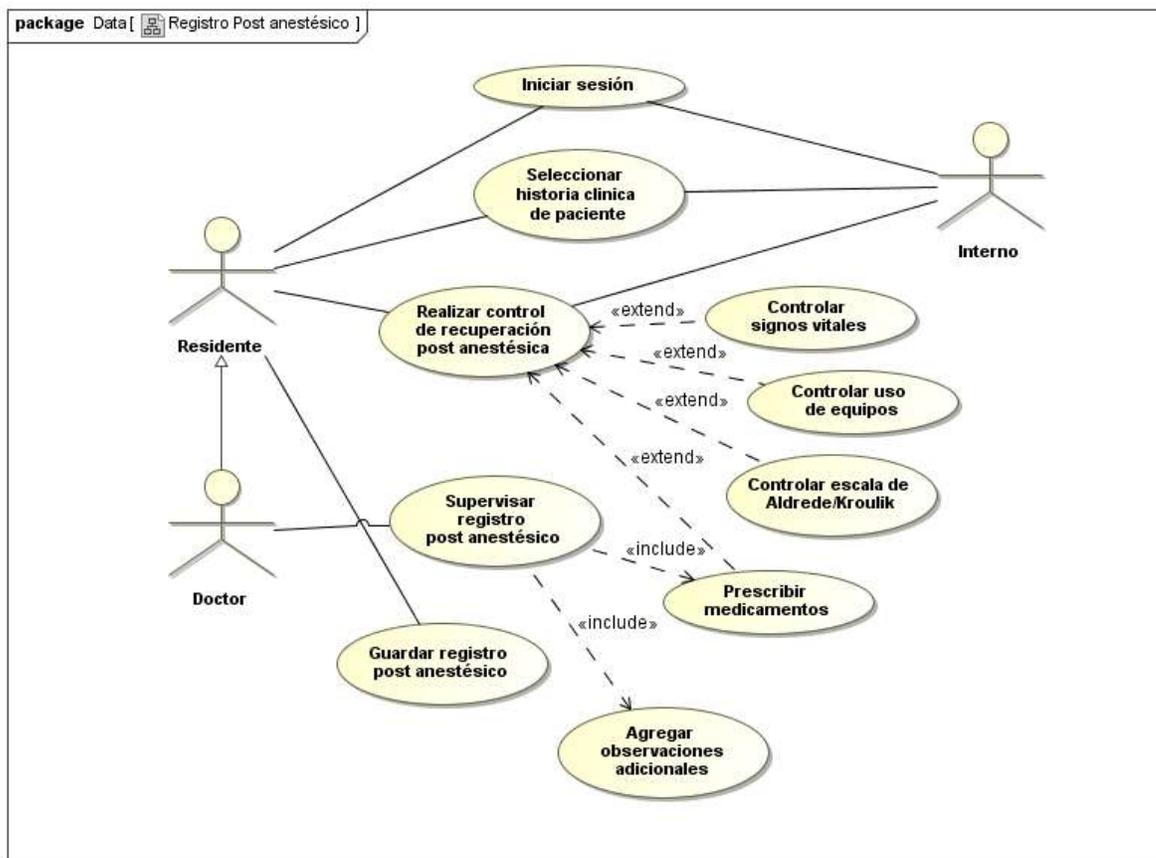
**Figura 3.5** Casos de uso del proceso de consulta de registro pre anestésico

Descripción de los casos de uso:

**Tabla 3.10** Caso de uso “Agregar observaciones adicionales”

<b>Escenario</b>	Caso de uso “Agregar observaciones adicionales”
<b>Código</b>	CU-CON-08
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El usuario está consultando un registro pre anestésico.</li> <li>El usuario Doctor está supervisando un registro post anestésico.</li> </ul>
<b>Flujo Normal</b>	El usuario añade observaciones adicionales al registro anestésico.
<b>Flujo Alternativo</b>	
<b>Postcondición</b>	Se registra la observación, el id del usuario y la fecha.

## c) Proceso de registro post anestésico (control de recuperación post anestésica)



**Figura 3.6** Casos de uso del proceso de registro post anestésico

Descripción de los casos de uso:

**Tabla 3.11** Caso de uso “Realizar control de recuperación post anestésica”

<b>Escenario</b>	Caso de uso “Realizar control de recuperación post anestésica”
<b>Código</b>	CU-POST-09
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario seleccionó un registro anestésico para hacer el control.</li> <li>• El paciente tiene completo su registro trans anestésico.</li> <li>• Se tiene acceso al servicio REST para obtener los recursos necesarios para el control.</li> </ul>
<b>Flujo Normal</b>	El usuario vigila la evolución del paciente anotando la hora en que realiza los controles.

<b>Flujo Alternativo</b>	Si el usuario no registra ciertos parámetros críticos, la aplicación mostrará un mensaje de advertencia.
<b>Postcondición</b>	Se tienen los datos de control de la recuperación post anestésica.

**Tabla 3.12** Caso de uso “Controlar signos vitales”

<b>Escenario</b>	Caso de uso “Controlar signos vitales”
<b>Código</b>	CU-POST-10
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	El usuario se encuentra controlando la recuperación post anestésica.
<b>Flujo Normal</b>	El usuario registra la hora de control y los parámetros como presión arterial, etc. El usuario puede realizar varios controles según la evolución del paciente.
<b>Flujo Alternativo</b>	Si el usuario no toma ningún control, la aplicación mostrará un mensaje de advertencia.
<b>Postcondición</b>	Se registran los controles de signos vitales.

**Tabla 3.13** Caso de uso “Controlar uso de equipos”

<b>Escenario</b>	Caso de uso “Controlar uso de equipos”
<b>Código</b>	CU-POST-11
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	El usuario se encuentra controlando la recuperación post anestésica.
<b>Flujo Normal</b>	El usuario registra los equipos utilizados para la recuperación del paciente.
<b>Flujo Alternativo</b>	No es obligatorio que haya equipos en uso.
<b>Postcondición</b>	Se registra el uso de equipos

**Tabla 3.14** Caso de uso “Controlar escala de Aldrede/Kroulik”

<b>Escenario</b>	Caso de uso “Controlar escala de Aldrede/Kroulik”
<b>Código</b>	CU-POST-12
<b>Actor (es)</b>	Doctor, Residente, Interno
<b>Precondición</b>	El usuario se encuentra controlando la recuperación post anestésica.
<b>Flujo Normal</b>	El usuario controla el nivel de recuperación del paciente según la escala de Aldrede, anotando además la hora de control. Este control puede realizarse varias veces hasta verificar que el paciente recupere la consciencia.
<b>Flujo Alternativo</b>	Si no se toma ningún control, la aplicación mostrará un mensaje de error.
<b>Postcondición</b>	Se registra el último control cuando ya puede darse el alta según la escala de Aldrede.

**Tabla 3.15** Caso de uso “Prescribir medicamentos”

<b>Escenario</b>	Caso de uso “Prescribir medicamentos”
<b>Código</b>	CU-POST-13
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	El usuario se encuentra controlando la recuperación post anestésica.
<b>Flujo Normal</b>	Se registra la hora y la prescripción hecha al paciente según el caso.
<b>Flujo Alternativo</b>	No es obligatorio prescribir medicamentos.
<b>Postcondición</b>	Se registran la hora y las prescripciones.

**Tabla 3.16** Caso de uso “Supervisar registro post anestésico”

<b>Escenario</b>	Caso de uso “Supervisar registro post anestésico”
<b>Código</b>	CU-POST-14
<b>Actor (es)</b>	Doctor
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Previamente ya se registró un control de recuperación post anestésico.</li> </ul>

	<ul style="list-style-type: none"> <li>Se cuenta con acceso al servicio REST para recuperar los datos del registro post anestésico.</li> </ul>
<b>Flujo Normal</b>	<p>El usuario valida los controles registrados, añadiendo observaciones de ser necesario.</p> <p>El usuario es el encargado de dar de alta al paciente y debe registrar la fecha y hora del alta.</p>
<b>Flujo Alternativo</b>	Si el usuario no definió la fecha y hora del alta, la aplicación mostrará un mensaje de alerta.
<b>Postcondición</b>	El registro post anestésico se encuentra completo.

**Tabla 3.17** Caso de uso “Guardar registro post anestésico”

<b>Escenario</b>	Caso de uso “Guardar registro post anestésico”
<b>Código</b>	CU-POST-17
<b>Actor (es)</b>	Doctor, Residente
<b>Precondición</b>	Se completó el registro post anestésico
<b>Flujo Normal</b>	Se registran los datos del control post anestésico en el servidor.
<b>Flujo Alternativo</b>	Si falta algún parámetro obligatorio, la aplicación indicará que datos faltan para proceder al registro.
<b>Postcondición</b>	<ul style="list-style-type: none"> <li>Se completa correctamente el registro anestésico del paciente en sus tres etapas: pre, trans y post anestésico.</li> <li>Queda pendiente imprimir la documentación del paciente en la aplicación web.</li> </ul>

### 3.4.1.3 Resultados de la Fase de Especificación

Al término de esta fase se identificaron los actores y los casos de uso necesarios para las funcionalidades requeridas.

Se definieron también las iteraciones necesarias por cada componente, que se ilustran en la tabla 3.18:

**Tabla 3.18** Iteraciones por cada componente

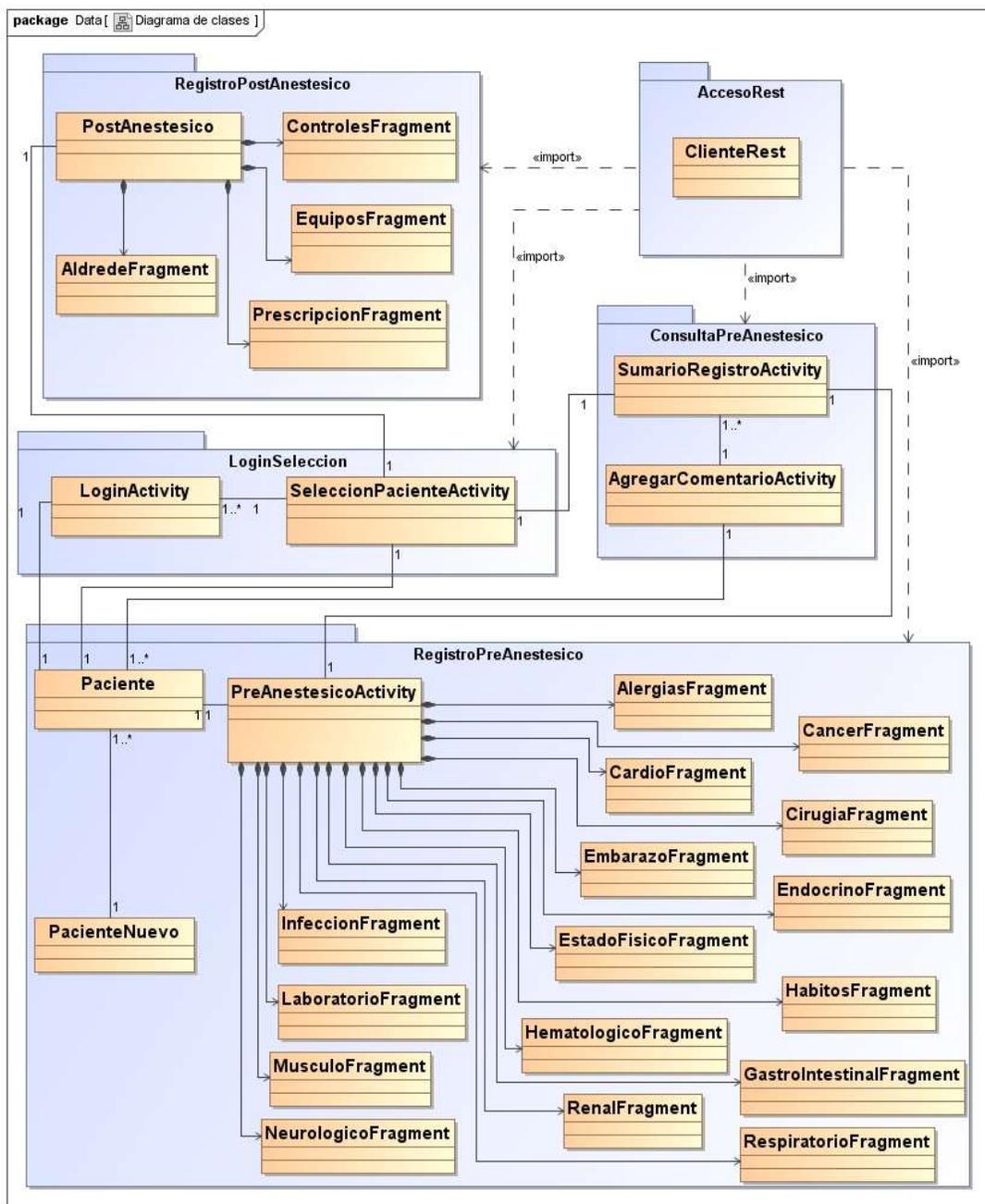
<b>Componente</b>	<b>Iteraciones requeridas</b>	<b>Observaciones</b>
Registro pre anestésico	5	Es el que más complejidad tiene, debido a las diferentes patologías a considerar, los análisis de laboratorio y las posibles intervenciones anteriores que tenga el paciente.
Consultas a registro pre anestésico	2	Se necesitará una iteración para el desarrollo (ya que se reutilizaran submodulos) y una adicional para ajustar las observaciones del cliente.
Registro post anestésico	3	Es el segundo más complejo después del registro pre anestésico, se necesita la participación constante del cliente.



### 3.4.2 Fase de Colaboración

#### 3.4.2.1 Diagrama de Clases

Mediante el diagrama de clases definimos la arquitectura del sistema, tal como se muestra en la figura 3.7



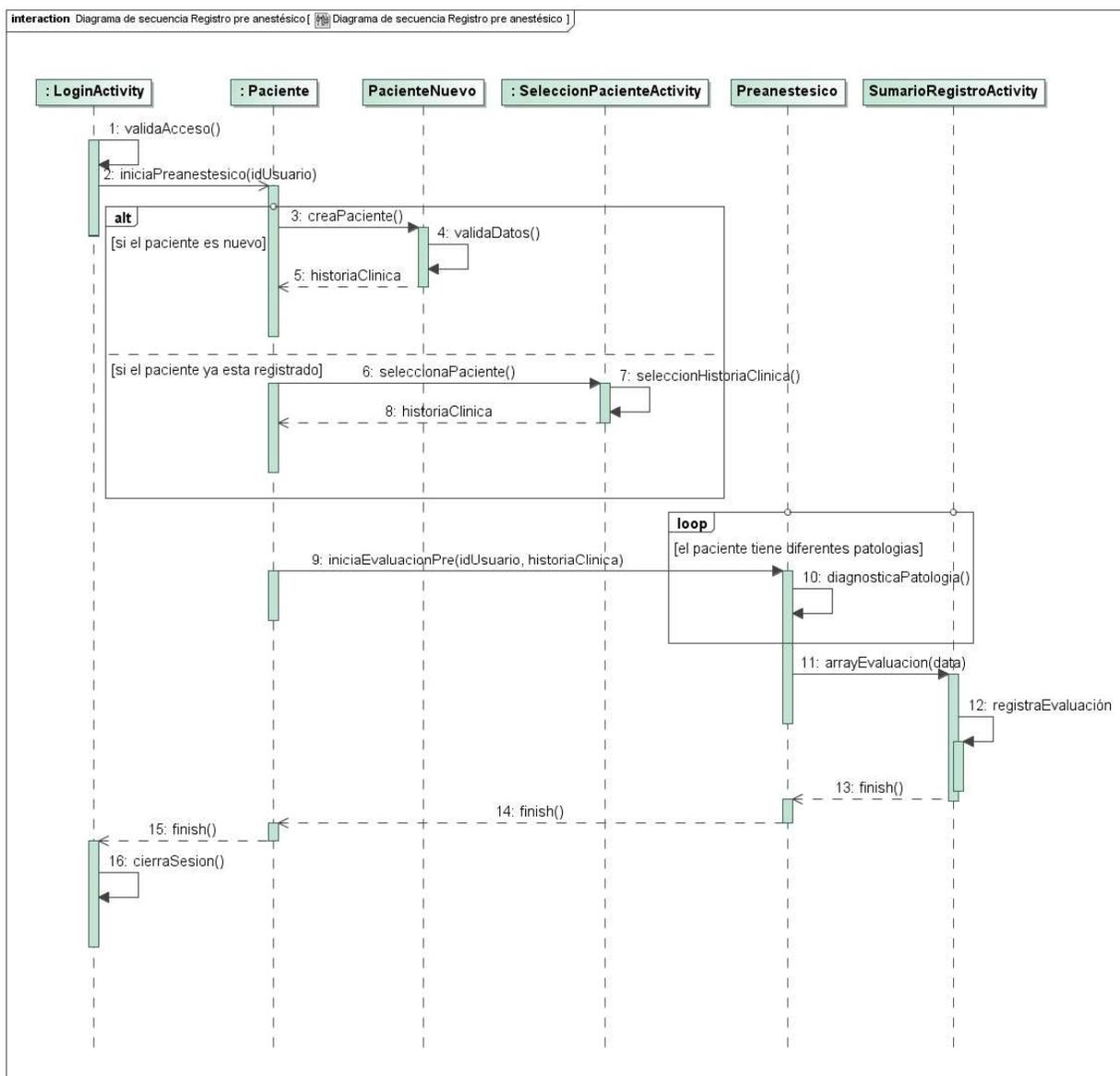
**Figura 3.7** Arquitectura del sistema – Diagrama de clases

Los detalles de cada clase como ser atributos y métodos están incluidos en el Anexo B.

### 3.4.2.2 Diagrama de Secuencias

Para ver cómo estas clases interactúan en el tiempo, se modelan los diagramas de secuencias para los escenarios de registro pre anestésico, consulta de registro pre anestésico y registro post anestésico.

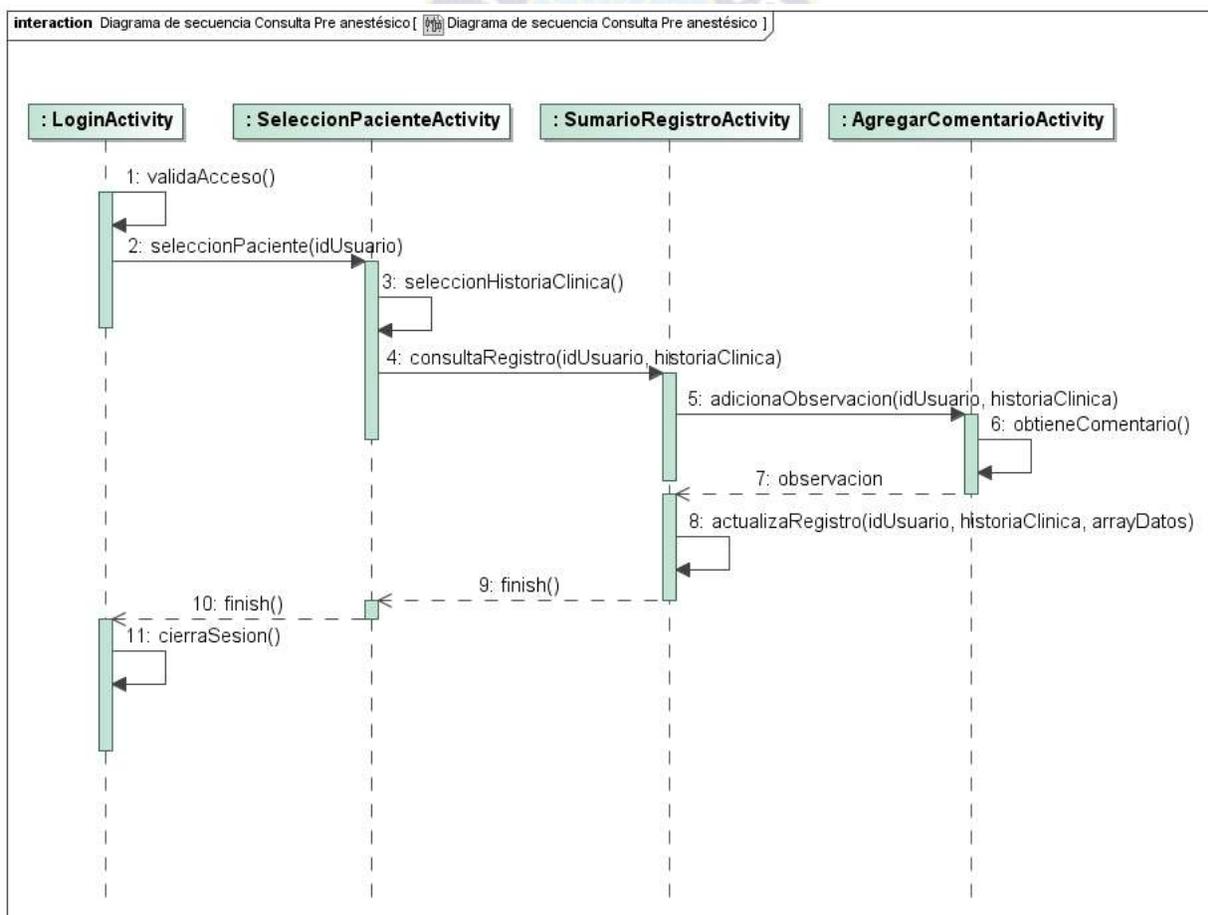
#### a) Registro pre anestésico



**Figura 3.8** Diagrama de secuencias – Registro pre anestésico

En la figura 3.8 se muestra la clase InicioActivity iniciando el proceso cuando el usuario se autentica y procede a seleccionar un paciente existente o registrar los datos de un paciente nuevo mediante la interacción de las clases SeleccionPacienteActivity y PacienteNuevo respectivamente. En ambos casos se dispone del id de usuario y la historia clínica del paciente, con lo cual se procede a iniciar el registro anestésico donde se van consignando los datos de evaluación según las patologías del paciente para luego ingresar al sumario donde el usuario revisa el global de los datos registrados, de no haber observaciones se procede a guardar el registro en la base de datos del servidor.

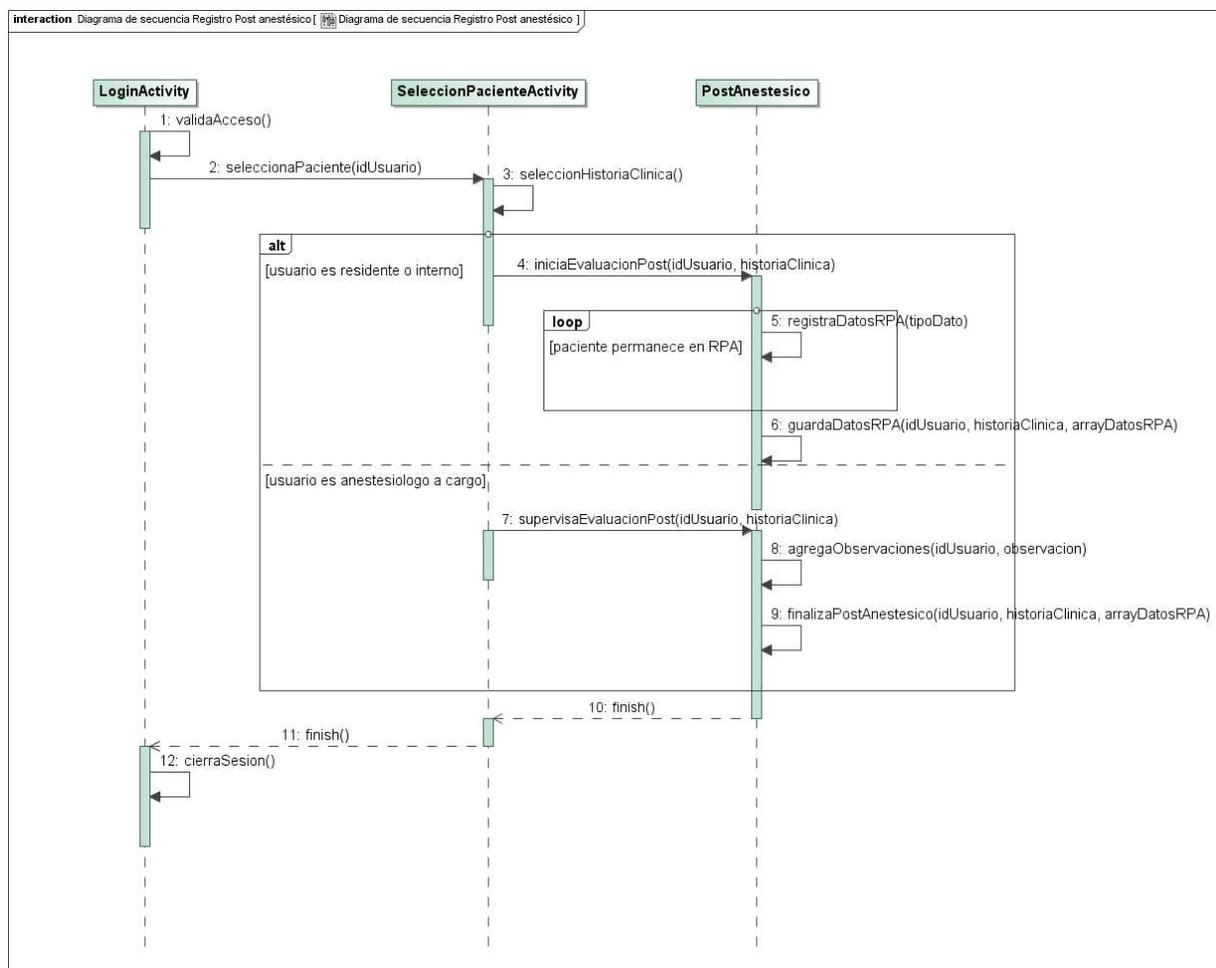
### b) Consulta de registro pre anestésico



**Figura 3.9** Diagrama de Secuencias – Consulta registro pre anestésico

La figura 3.9 muestra la consulta de un registro pre anestésico por un usuario, la secuencia de pasos es similar a la figura 3.8, con la diferencia que no se realiza la evaluación pre anestésica y se pueden agregar observaciones mediante la clase `AgregarComentarioActivity`.

### c) Registro post anestésico



**Figura 3.10** Diagrama de secuencias – Registro Post anestésico

El registro post anestésico inicia con el usuario autenticándose y seleccionando una historia clínica, si se trata de un residente o un interno procede a registrar la recuperación post anestésica del paciente, anotando todos los datos identificados en los requisitos y guardando el registro para una posterior revisión del anesthesiólogo a cargo. Si el usuario tiene los permisos adecuados (es el anesthesiólogo a cargo) procede

a seleccionar el registro que necesita revisión para dar las últimas observaciones y cerrar el registro post anestésico de la historia clínica seleccionada.

### 3.4.2.3 Resultados de la Fase de Colaboración

Se diseñó e implementó la arquitectura y se desarrollaron las funcionalidades según los requisitos identificados. Se cumplieron el número de iteraciones establecidas en la fase de especulación y se diseñó la interfaz de usuario, cuyas características como la usabilidad y funcionalidad fueron considerados en la siguiente etapa del método ASD.

### 3.4.3 Fase de Aprendizaje

Esta fase comprende la evaluación del producto conjuntamente con el cliente para identificar los factores a mejorar durante la siguiente iteración.

El resultado de esta fase es el producto completado con las debidas revisiones de las funcionalidades de la aplicación a fin de garantizar la usabilidad por parte del cliente.

Se mostrará la funcionalidad del sistema tal como se planteó en la fase de colaboración, producto de constantes mejoras a sugerencia del Servicio de Anestesiología del Hospital de la Mujer.



**Figura 3.11** Pantalla inicial en espera de servicio

Iniciando la aplicación se espera a recibir el listado de usuarios autorizados y claves de acceso al sistema.

Registro Anestésico

**Hospital de la Mujer**  
Atención integral para toda la familia

**Inicie Sesión**

Usuario: Dr. Eduardo Alvarez

Clave: ...

Seleccione actividad

- Realizar registro Preanestésico
- Consultar registro Preanestésico (no incluye pacientes antiguos)
- Realizar registro Postanestésico

INGRESO

**Figura 3.12** Pantalla inicial con acceso al servicio

El usuario selecciona su cuenta e introduce su clave para elegir una actividad: “Realizar registro Preanestésico”.

Datos del Paciente

Pulse en 'Paciente' para seleccionar o registrar un nuevo paciente

Paciente: no seleccionado | Historial Clínico: —

Seleccione Sala: | Seleccione Cama: |

Peso: [ ] [Mts]

PA / PA: [ ] [latidos/minuto]

T: [ ] [Resp./minuto]

servicio: especifique el servicio

Diagnóstico: diagnóstico del paciente

Procedimiento: procedimiento a realizar

PROCEDER

**Datos de Paciente**

Pacientes Agregar Paciente

**Figura 3.13** Datos iniciales del paciente

Al comenzar el registro pre anestésico se debe seleccionar al paciente o adicionarlo en caso que no esté registrado previamente.

**Datos del Paciente**

Paciente: juan perez      Historial Clínico: pj150485

Seleccione Sala      Seleccione Cama

65 [Kg]      1.75 [Mts]

150 / 100 [mmHg]      130 [Latidos/minuto]

36.7 [°C]      120 [Resp./minuto]

Servicio: Cardiología

Diagnóstico: arritmia cardiaca

Procedimiento: ablacion por cateter

PROCEDER

**Figura 3.14** Datos iniciales del paciente completados

**Evaluacion PreAnestesia**      REGISTRAR

Hipertensión       Angina

Arritmia       Angioplastía

Infarto del Miocardio       Insuficiencia Cardíaca

AAA

CARDIO CIRCULATORIO

MUSCULO ESQUELETICO

SISTEMA ENDOCRINO

RESPIRATORIO

CÁNCER

INFECCIOSAS

GASTROINTESTINAL

REP.

**Figura 3.15** Evaluación de patologías del paciente

Se van clasificando las patologías del paciente en diferentes categorías para un fácil acceso.

**Verificación de Datos**

**Evaluación PreAnestésica**

**Anestesiologo** Dr. Sergio Cabrera

**Fecha** 29/10/2016 12:31

**Paciente** Fernando Agullar      **Historial Clínico** AF0012

**Sala** Seleccione Sala Cama      Seleccione: Cama

Peso	Altura	PA	FC	Temp	Fr
70 Kg.	1.7 Mts.	150/120 mmHg	120 Lat/Min	36.7 °C	140 Respi/Min

**Especialidad** Cardiología

**Diagnóstico** arritmia cardiaca

**Procedimiento** ablacion por cateter

**Cardio - Circulatorio**      **Músculo Esquelético**

Hipertensión       Artritis

Angina

Arritmia

**Sistema Endocrino**      **Respiratorio**

Negativo       Dependencia de O2

Asma

**Cancer**      **Infecciones**

Negativo       Negativo

**Gastrointestinal / Rep.**      **Neurológico**

Negativo       Convulsiones

**Figura 3.16** Sumario pre anestésico hoja 1

El último paso es verificar el sumario pre anestésico donde el usuario tiene la opción de volver atrás para corregir los datos de ser necesario.

**Verificación de Datos**

**Renal**      **Hematológico**

Negativo       Transfusiones previas

**Embarazo**      **Hábitos Sociales**

Negativo       Negativo

**Antecedentes Patológicos**

**Enfermedades previas:**

---

**Cirugías/Anestesis previas**

---

**Declaradas por el paciente:**

---

**Historial de Intervenciones:**

---

Historial Nauseas/vómitos post Op.       Historial Familiar problemas c/Anest.

**Laboratorio**

Hemoglobina: 1.2 [g%]  
Hematocrito: 50 %  
Actividad de Protrombina: 70 %  
Tiempo de Coagulación: 5 [minutos]  
INR: 0.5  
Potasio (K): 12  
Sodio (Na): 15  
Plaquetas: 2.3 [x10<sup>3</sup>]  
Glucosa: 1000

**Figura 3.17** Sumario pre anestésico hoja 2

**Verificación de Datos**

BEecf: 25 mmHg I/L  
HCO3: 44 mmHg I/L  
TCO2: 23.5 mmHg I/L  
sO2: 36 %

Alergias	
Tipo / Agente	Reacción
Sin Alergias	—
historial de Agentes:	historial de reacciones:

**Examen Físico**

Cardíaco: Negativo

Respiratorio: Normal

Neuro: Patológico

Regional: Normal

Otro: Negativo

**Medicación Previa**

Negativo

Hist. Vía Aérea difícil: si  
Mallampati: Mallampati 3  
Estado Físico ASA: IV

**Comentarios**

se programa la intervención para la proxima semana

REGISTRAR      VOLVER

*Figura 3.18* Sumario pre anestésico hoja 3

**Registro Anestésico**

**Hospital de la Mujer**  
Atención integral para toda la familia

**Inicie Sesión**

Usuario: Dr. Sergio Cabrera

Clave: \*\*\*

**Seleccione actividad**

Realizar registro Preanestésico

Consultar registro Preanestésico (no incluye pacientes antiguos)

Realizar registro Postanestésico

INGRESO

Los datos se registraron exitosamente!

*Figura 3.19* Registro pre anestésico finalizado

El registro pre anestésico es almacenado en la base de datos y la aplicación cierra automáticamente la sesión, mostrando un mensaje del estado del registro.

### 3.4.3.1 Resultados de la Fase de aprendizaje

Al término de la fase de aprendizaje tenemos los siguientes resultados:

- Se desarrolló la aplicación android con las funcionalidades especificadas en los requisitos.
- Se realizaron pruebas con el cliente para verificar la calidad de la aplicación.
- Se adaptó la interfaz gráfica de manera que sea fácil e intuitiva.
- Se desplegó la aplicación para su utilización con los demás componentes del sistema.

## 3.5 APLICACIÓN WEB – MÉTODO UWE

El Sistema de Información para la Gestión del Registro Anestésico - SIGRA cuenta con una componente web que fue desarrollado mediante el método UWE generando en cada uno de los modelos uno o varios artefactos que representan a la aplicación web en escenarios específicos.

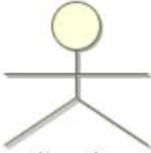
### 3.5.1 Modelo de Requerimientos – Casos de Uso

Para poder obtener una definición clara de los requerimientos de esta aplicación se identifican actores, se generan casos de uso para cada subsistema y se detallan los escenarios específicos.

#### 3.5.1.1 Definición de actores

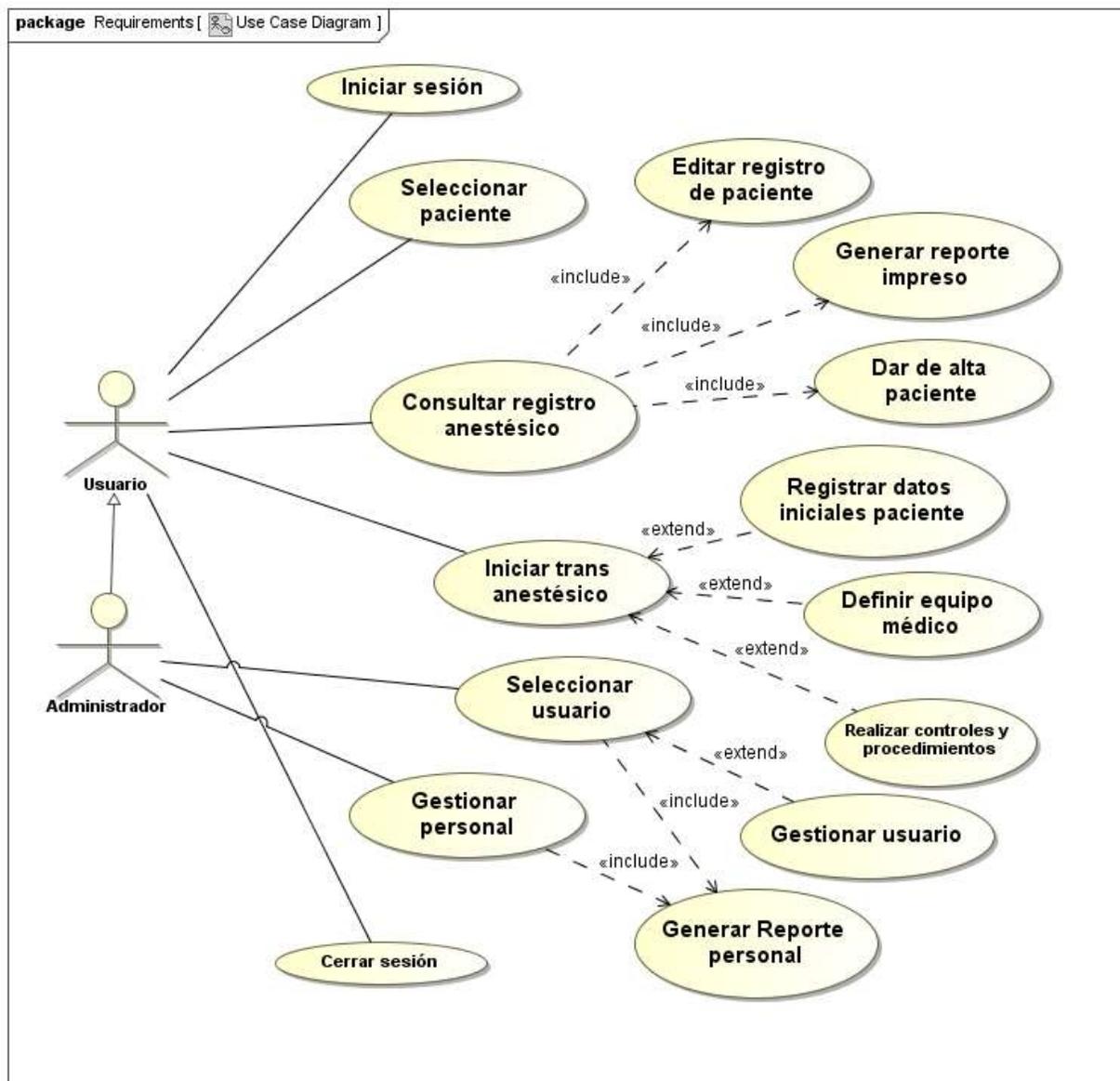
Como inicio del modelo de requerimientos se realizó un análisis de los usuarios que interactuarán con la aplicación como se muestra en la tabla 3.19:

**Tabla 3.19** Definición de actores – aplicación web

Actor	Descripción
 <p data-bbox="475 1535 553 1556">Usuario</p>	<p data-bbox="834 1381 1419 1612">Se refiere a los médicos del Servicio de Anestesiología, este actor puede realizar el registro trans anestésico, consultar y generar reportes de historiales de pacientes y pacientes antiguos.</p>
 <p data-bbox="443 1839 586 1860">Administrador</p>	<p data-bbox="834 1644 1419 1875">Se refiere al jefe del Servicio de Anestesiología, este actor realiza las mismas funciones del actor Usuario además de gestionar usuarios y otros administradores, este actor también puede consultar y generar reportes de personal.</p>

### 3.5.1.2 Casos de uso

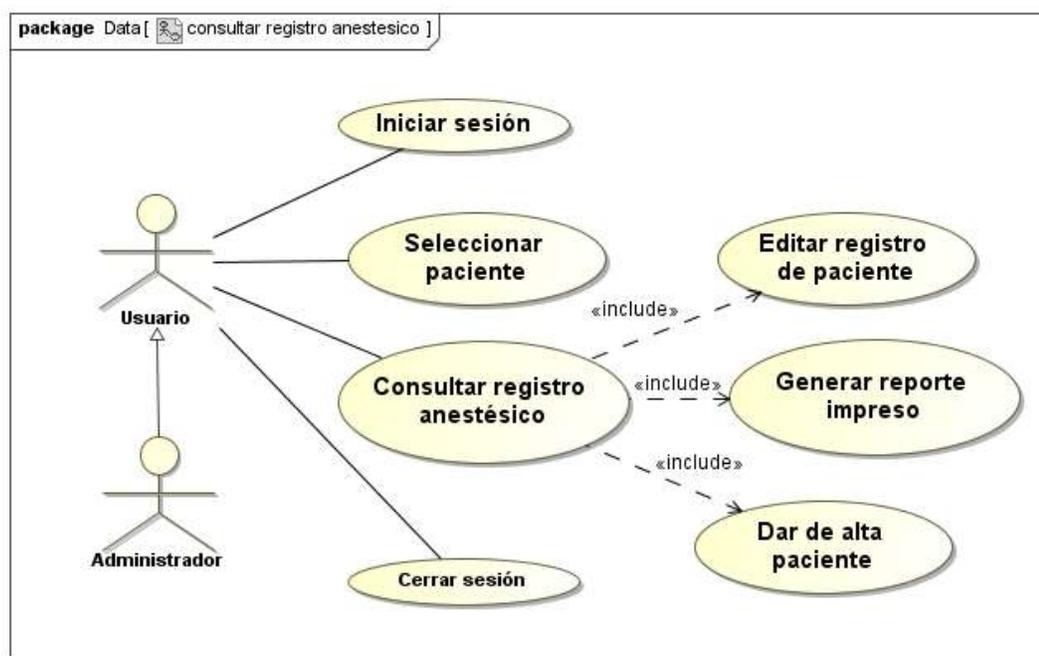
El diagrama general de casos de uso esquematiza los requerimientos de la aplicación web.



**Figura 3.20** Diagrama general de casos de uso – aplicación web

La aplicación web está compuesta de componentes los cuales se identifican con los siguientes casos de uso:

a) Caso de uso – Consultar registro anestésico



**Figura 3.21** Consultar registro anestésico

Descripción de los casos de uso:

**Tabla 3.20** Caso de uso “Iniciar sesión”

<b>Escenario</b>	Caso de uso “Iniciar sesión”
<b>Código</b>	CU-WEB-01
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	El usuario es parte del personal del Servicio de Anestesiología
<b>Flujo Normal</b>	El usuario selecciona su cuenta e introduce su clave de acceso.
<b>Flujo Alternativo</b>	Si los datos son incorrectos, el sistema no permitirá el ingreso.
<b>Postcondición</b>	El usuario tiene iniciada la sesión.

**Tabla 3.21** Caso de uso “Seleccionar paciente”

<b>Escenario</b>	Caso de uso “Seleccionar paciente”
<b>Código</b>	CU-WEB-02
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	El usuario inició sesión correctamente.

<b>Flujo Normal</b>	Se presenta un listado de pacientes activos o pacientes antiguos para ser seleccionados.
<b>Flujo Alternativo</b>	Si no existen registros en la base de datos, el listado estará vacío.
<b>Postcondición</b>	Se tiene un paciente seleccionado para realizar una actividad.

**Tabla 3.22** Caso de uso “Consultar registro anestésico”

<b>Escenario</b>	Caso de uso “Consultar registro anestésico”
<b>Código</b>	CU-WEB-03
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	Se tiene seleccionado un paciente nuevo o antiguo.
<b>Flujo Normal</b>	El usuario consulta el registro anestésico en sus tres etapas si es un paciente antiguo o los registros que el paciente activo tenga completos (pre, trans, post anestésico).
<b>Flujo Alternativo</b>	
<b>Postcondición</b>	

**Tabla 3.23** Caso de uso “Editar registro de paciente”

<b>Escenario</b>	Caso de uso “Editar registro de paciente”
<b>Código</b>	CU-WEB-04
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Se está consultando un registro anestésico.</li> <li>• Hay datos que necesitan ser actualizados</li> <li>• El paciente todavía no fue dado de alta.</li> </ul>
<b>Flujo Normal</b>	El usuario modifica / actualiza los datos del paciente (p.e. nombre en caso que el paciente haya ingresado como “N.N.”)
<b>Flujo Alternativo</b>	El sistema sólo permite la edición de ciertos datos del paciente como nombre, sala, cama, peso, altura, edad.
<b>Postcondición</b>	Los datos del paciente han sido actualizados.

**Tabla 3.24** Caso de uso “Generar reporte impreso”

<b>Escenario</b>	Caso de uso “Generar reporte impreso”
<b>Código</b>	CU-WEB-05
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Se está consultando un registro anestésico.</li> <li>• El registro anestésico a imprimirse está completo.</li> </ul>
<b>Flujo Normal</b>	El usuario genera el reporte impreso correspondiente a un registro anestésico (pre, trans, post anestésico).
<b>Flujo Alternativo</b>	El usuario genera el reporte en formato PDF.
<b>Postcondición</b>	<ul style="list-style-type: none"> <li>• Se genera el reporte anestésico</li> <li>• Se incrementa el contador de reportes y se registra fecha y usuario que genera.</li> <li>• El registro anestésico consultado ya no admite más modificaciones.</li> </ul>

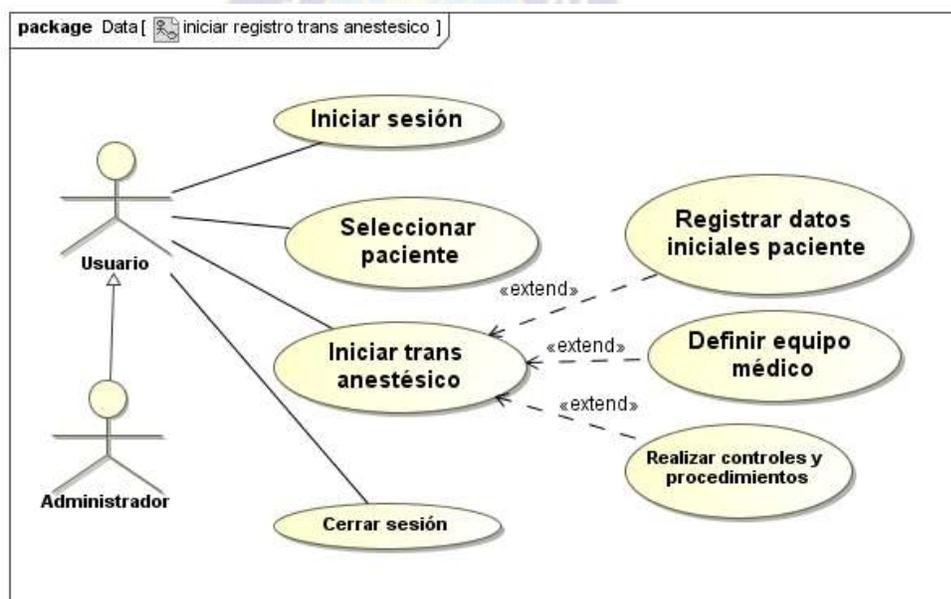
**Tabla 3.25** Caso de uso “Dar de alta paciente”

<b>Escenario</b>	Caso de uso “Dar de alta paciente”
<b>Código</b>	CU-WEB-06
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El paciente completo su registro post anestésico.</li> <li>• El registro anestésico del paciente está completo.</li> </ul>
<b>Flujo Normal</b>	El usuario verifica los últimos detalles del registro anestésico del paciente y da su consentimiento para el alta del paciente.
<b>Flujo Alternativo</b>	De haber alguna observación seria, esta se corregirá antes del alta del paciente.
<b>Postcondición</b>	<ul style="list-style-type: none"> <li>• El registro anestésico consultado ya no admite más modificaciones.</li> <li>• El paciente pasa al histórico de pacientes y deja de ser un paciente activo.</li> </ul>

**Tabla 3.26** Caso de uso “Cerrar sesión”

<b>Escenario</b>	Caso de uso “Cerrar sesión”
<b>Código</b>	CU-WEB-07
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	El usuario inició sesión correctamente.
<b>Flujo Normal</b>	El usuario finaliza la sesión actual.
<b>Flujo Alternativo</b>	
<b>Postcondición</b>	Se guardan los datos de la actividad del usuario.

**b) Caso de uso – Iniciar procedimiento trans anestésico**



**Figura 3.22** Iniciar procedimiento trans anestésico

Descripción de casos de uso:

**Tabla 3.27** Caso de uso “Iniciar trans anestésico”

<b>Escenario</b>	Caso de uso “Iniciar trans anestésico”
<b>Código</b>	CU-WEB-08
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	El paciente tiene completo su registro pre anestésico.
<b>Flujo Normal</b>	El usuario procede a registrar la intervención quirúrgica, la cual el sistema lo divide en tres etapas: registrar datos, definir equipo y realizar controles.
<b>Flujo Alternativo</b>	
<b>Postcondición</b>	El paciente tiene completo su registro trans anestésico y pasa a recuperación post anestésica.

**Tabla 3.28** Caso de uso “Registrar datos iniciales paciente”

<b>Escenario</b>	Caso de uso “Registrar datos iniciales paciente”
<b>Código</b>	CU-WEB-09
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	El usuario ha iniciado el registro trans anestésico
<b>Flujo Normal</b>	El usuario establece los datos iniciales del paciente previo a iniciar la intervención quirúrgica.
<b>Flujo Alternativo</b>	Si no se llenan los campos, el sistema no permitirá avanzar y mostrará un mensaje de alerta.
<b>Postcondición</b>	Se establecen datos del paciente como el índice Glasgow, presión arterial, conteo de hematocritos y glóbulos rojos que se registran en el trans anestésico.

**Tabla 3.29** Caso de uso “Definir equipo médico”

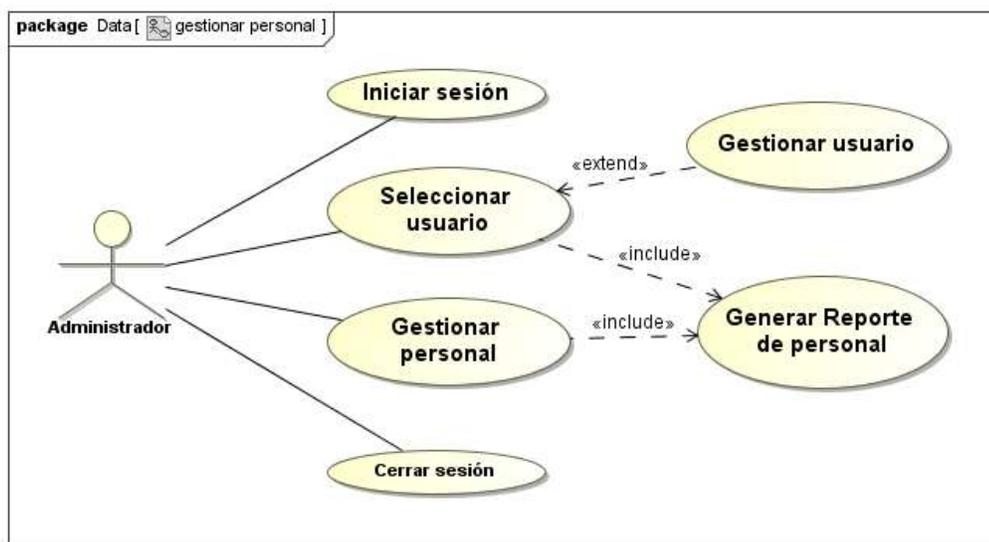
<b>Escenario</b>	Caso de uso “Definir equipo médico”
<b>Código</b>	CU-WEB-10
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	<ul style="list-style-type: none"> <li>El usuario ha iniciado el registro trans anestésico.</li> </ul>

	<ul style="list-style-type: none"> <li>• Se ha completado los datos iniciales para la intervención.</li> </ul>
<b>Flujo Normal</b>	El usuario define el equipo médico que participará en la intervención para establecer responsabilidades y roles
<b>Flujo Alternativo</b>	Si no se llenan los campos, el sistema no permitirá avanzar y mostrará un mensaje de alerta.
<b>Postcondición</b>	Se registran los participantes y sus roles en el trans anestésico.

**Tabla 3.30** Caso de uso “Realizar controles y procedimientos”

<b>Escenario</b>	Caso de uso “Realizar controles y procedimientos”
<b>Código</b>	CU-WEB-11
<b>Actor (es)</b>	Administrador, Usuario
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario ha iniciado el registro trans anestésico.</li> <li>• Se han definido los participantes y roles del equipo médico.</li> </ul>
<b>Flujo Normal</b>	El usuario comienza la intervención quirúrgica y va registrando los diferentes datos, controles y procedimientos utilizados durante la intervención, hasta la conclusión de ésta.
<b>Flujo Alternativo</b>	Queda a criterio del usuario cuando registrar los datos y la hora de control. La prioridad es el paciente, no el sistema.
<b>Postcondición</b>	Se registran los participantes y sus roles en el trans anestésico.

c) *Caso de uso - Gestionar personal*



*Figura 3.23 Gestionar personal*

Descripción de casos de uso:

**Tabla 3.31** Caso de uso “Seleccionar usuario”

<b>Escenario</b>	Caso de uso “Seleccionar usuario”
<b>Código</b>	CU-WEB-12
<b>Actor (es)</b>	Administrador
<b>Precondición</b>	El usuario Administrador inició sesión correctamente.
<b>Flujo Normal</b>	El administrador accede a un listado de los usuarios registrados en el sistema.
<b>Flujo Alternativo</b>	Si no existe el listado, se brindará la posibilidad de registrar nuevo personal.
<b>Postcondición</b>	Se tiene el id del usuario seleccionado

**Tabla 3.32** Caso de uso “Gestionar usuario”

<b>Escenario</b>	Caso de uso “Gestionar usuario”
<b>Código</b>	CU-WEB-13
<b>Actor (es)</b>	Administrador
<b>Precondición</b>	El usuario Administrador seleccionó un usuario.

<b>Flujo Normal</b>	El sistema brinda la posibilidad de gestionar información del usuario o de restablecer su clave de acceso.
<b>Flujo Alternativo</b>	
<b>Postcondición</b>	El estado actualizado del usuario es registrado en el sistema.

**Tabla 3.33** Caso de uso “Generar reporte de personal”

<b>Escenario</b>	Caso de uso “Generar reporte de personal”
<b>Código</b>	CU-WEB-14
<b>Actor (es)</b>	Administrador
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• El usuario Administrador seleccionó un usuario.</li> <li>• El usuario Administrador gestiona información del personal.</li> </ul>
<b>Flujo Normal</b>	El sistema brinda la posibilidad de generar reportes por doctor, turnos, por especialidad y por residente
<b>Flujo Alternativo</b>	De haber alguna inconsistencia en los registros se informará al usuario para corregir los datos.
<b>Postcondición</b>	Se genera el reporte y se registra la actividad del usuario.

**Tabla 3.34** Caso de uso “Gestionar personal”

<b>Escenario</b>	Caso de uso “Gestionar personal”
<b>Código</b>	CU-WEB-15
<b>Actor (es)</b>	Administrador
<b>Precondición</b>	El usuario Administrador inició sesión correctamente.
<b>Flujo Normal</b>	El usuario puede realizar el CRUD en la base de datos para gestionar usuarios.
<b>Flujo Alternativo</b>	No se permite la eliminación de un usuario si tiene intervenciones registradas, en cambio su estado cambia a inactivo.
<b>Postcondición</b>	Se tiene actualizado el registro de personal.

### 3.5.2 Modelo Conceptual – Diagrama de clases

El diagrama de clases plantea las clases que se involucran en el procesamiento de información dentro del sistema, brindando una visión de la arquitectura de la aplicación web.

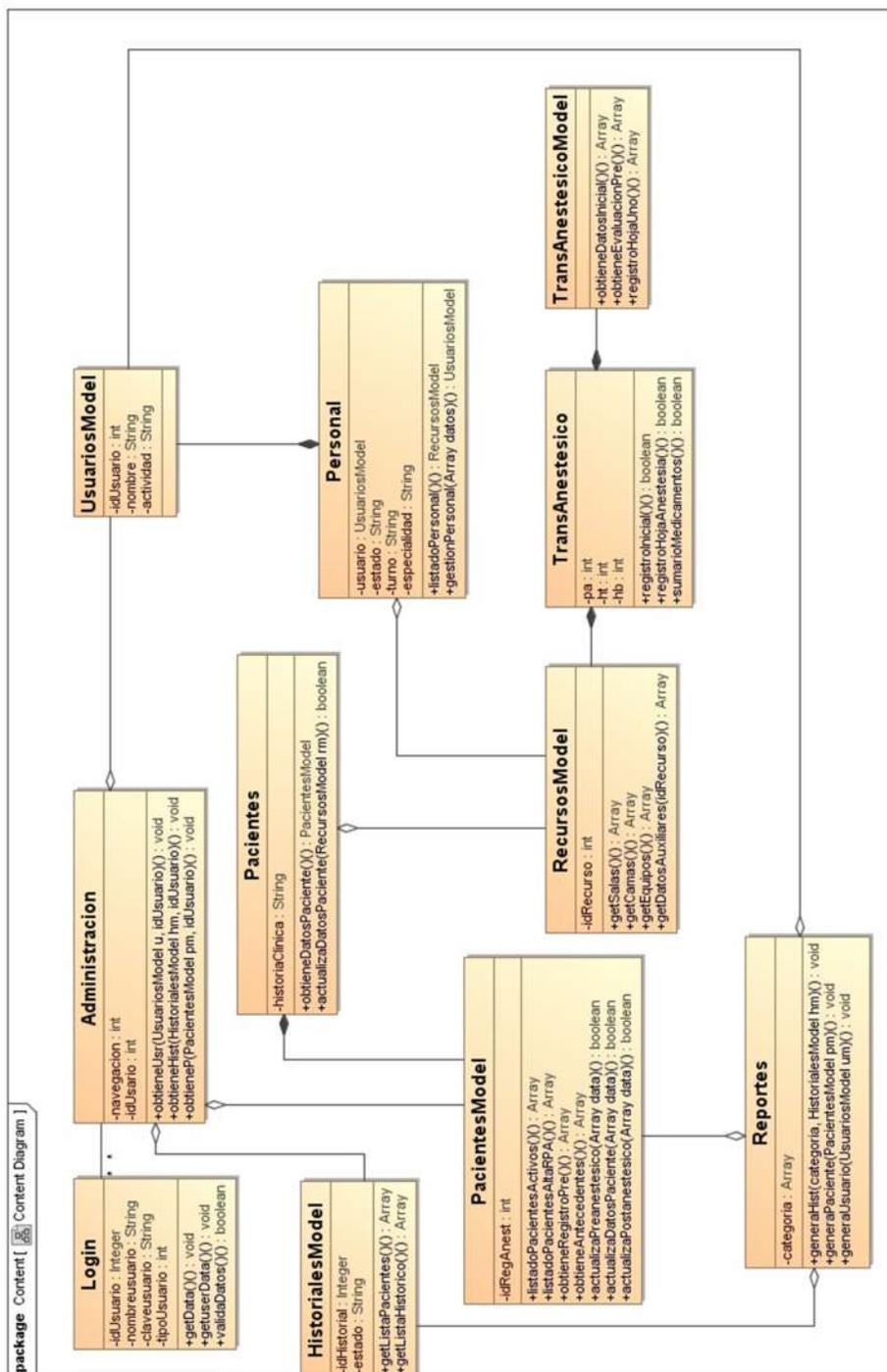


Figura 3.24 Modelo Conceptual – Aplicación web

### 3.5.3 Modelo Navegacional – Diagrama de navegación

El diagrama de navegación es el encargado de mostrar el flujo de navegación entre nodos de la aplicación web mediante enlaces o links como se muestra en la Figura 3.25

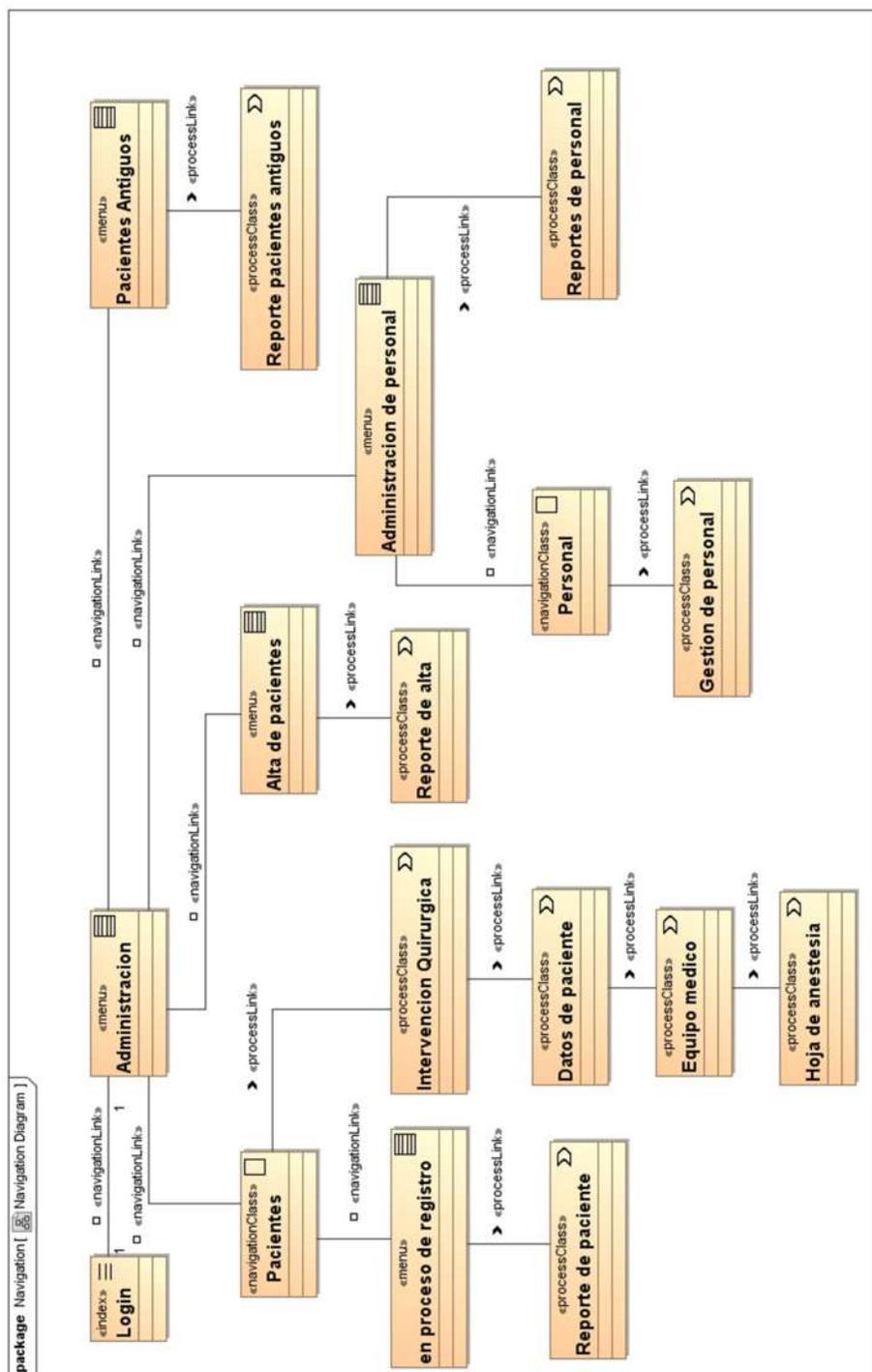


Figura 3.25 Modelo de navegación – Aplicación web

### 3.5.4 Modelo de Presentación – Diagrama de presentación

El diagrama de presentación es el encargado de mostrar un prototipo web mediante un conjunto de estereotipos, los mismos que representan elementos HTML comunes, las ventajas de este artefacto es que los estereotipos utilizados no son dependientes de una tecnología específica y brindan al usuario una idea de las interfaces que se presentarán en cada uno de los componentes de la aplicación web.

#### 3.5.4.1 Administración del sistema

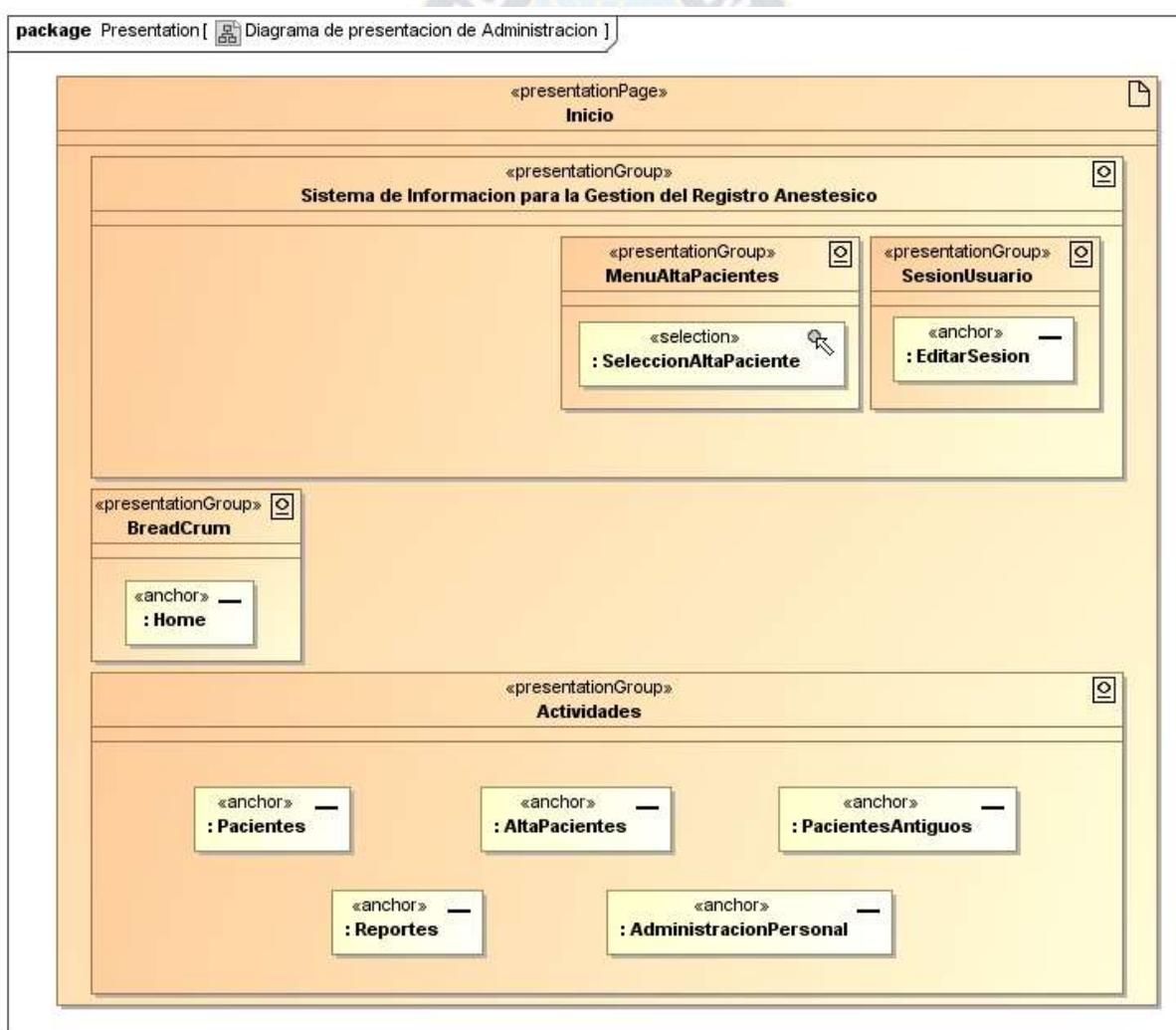


Figura 3.26 Diagrama de presentación – Administración del sistema

### 3.5.4.2 Gestión de pacientes

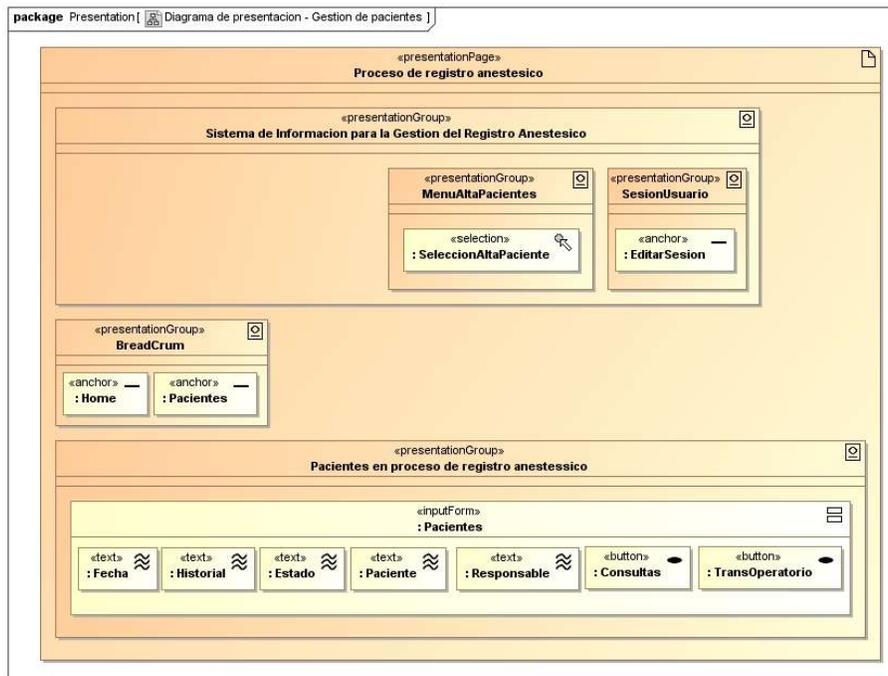


Figura 3.27 Diagrama de presentación – Gestión de pacientes

### 3.5.4.3 Consulta de registro anestésico

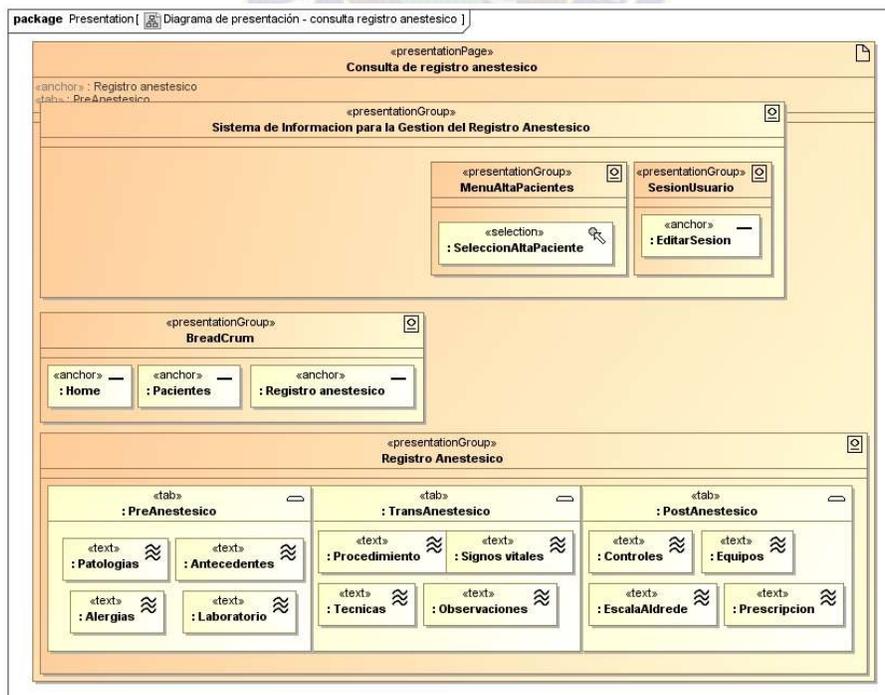


Figura 3.28 Diagrama de presentación – Consulta de registro anestésico

### 3.5.4.4 Intervención Quirúrgica

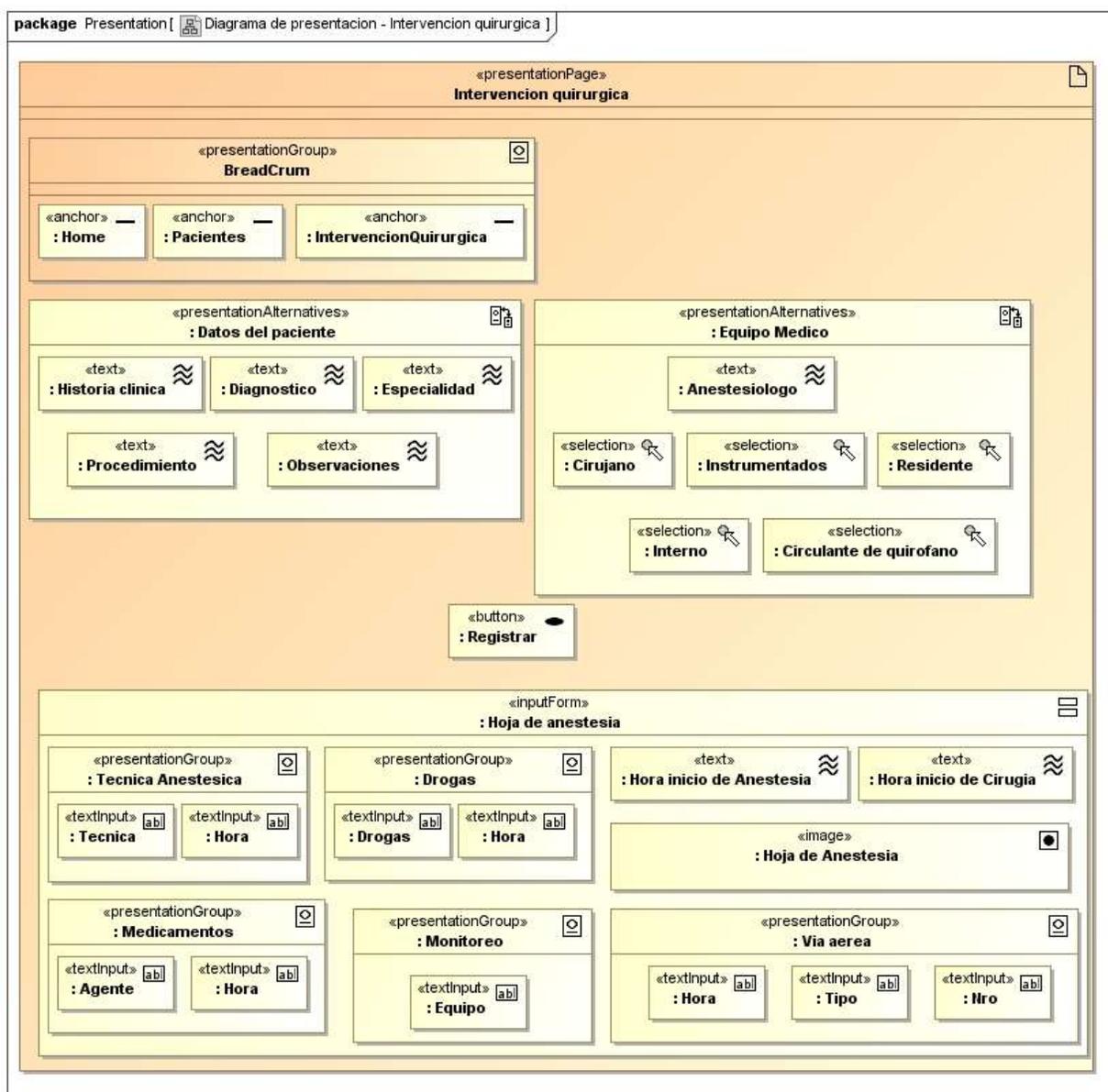


Figura 3.29 Diagrama de presentación – Intervención quirúrgica

### 3.5.5 Implementación de la aplicación web

Como resultado del desarrollo de la aplicación web mediante el método UWE, se presentan a continuación las interfaces mostrando la funcionalidad del sistema.



**Figura 3.30** Administración del sistema

Fecha	Historial	Estado	Paciente	Responsable	Consultas	Iniciar transoperatorio
31/12/1969 20:00	abodef	transoperatorio finalizado	perez, juan	---		
31/12/1969 20:00	AF0012	transoperatorio finalizado	Aguilar, Fernando	Dr. Eduardo Alvarez		
07/10/2016 12:35	me123	RPA evaluado por Interno	machicado, ellas	---		
30/10/2016 22:03	HESD463	evaluacion preanestesia finalizada	N.N.	Dr. Sergio Cabrera		

**Figura 3.31** Gestión de pacientes

REGISTRO ANESTESICO						Preanestésico	Transanestésico	Postanestésico
<b>Evaluación Pre-anestésica y consentimiento</b>						<a href="#">Editar Preanestésico</a> <a href="#">Generar Reporte</a>		
<span style="color: red;">todavía no se generó el REPORTE</span>								
						Fecha preanestésico: 05/10/2016 19:47		
<b>Peso</b>	<b>Altura</b>	<b>PA</b>	<b>FC</b>	<b>T</b>	<b>Fr</b>			
---	---	150/200	100	36	100			
[Kg.]	[Mts.]	[mmHg]	[Lat./Min]	[°C]	[Resp./Min]			
<b>Especialidad</b>		<b>Diagnóstico</b>		<b>Procedimiento</b>				
Cardiología		nuevo diagnostico		Procedimiento 5				
<b>Anestesiólogo:</b> Dr. Sergio Cabrera				<b>Comentarios:</b>				
				primer comentario [05/10/2016 20:14] segundo comentario [05/10/2016 21:09] comentario de residente [05/10/2016 22:24] comentario de otra residente [19/10/2016 18:41] nuevas observaciones				
<input checked="" type="checkbox"/> Cardio - Circulatorio <input checked="" type="checkbox"/> Arritmia <input checked="" type="checkbox"/> Angioplastia				<input checked="" type="checkbox"/> Musculo esquelético <span style="background-color: orange; color: white; padding: 2px;">NEGATIVO</span>				

*Figura 3.32* Consulta de registro anestésico

Inicio > Pacientes > Intervención Quirúrgica

Intervención Quirúrgica - Datos del Paciente						
<b>Paciente</b>	<b>Sexo</b>	<b>Historial Clínico</b>	<b>Peso</b>	<b>Altura</b>	<b>Sala</b>	<b>Cama</b>
N.N.	Femenino	HESD463	0 [Kg.]	0 [M.]	no especificado	no especificado
<b>Edad</b>	<b>Glasgow</b>	<b>PA</b>	<b>Ht.</b>	<b>Hb.</b>		
edad [↕] Años [v] adicional [↕] - edad - [v]	3 [v] /15	[↕] / [↕]	---	---		
<b>Diagnostico Preanestésico</b>	<b>Operacion propuesta</b>	<b>Servicio</b>				
nuevo diagnostico	nuevo procedimiento	nuevo servicio				
<b>Observaciones Preanestésico</b>	<b>Observaciones Adicionales Registro Preanestésico</b>					
bla bla						
Conformación del equipo médico						
Defina los integrantes del equipo médico y su rol (o especialidad), diferenciando entre médicos e internos						
<input type="button" value="Agregar medico"/> 		<b>Anestesiologo</b> Dr. Tomas Vaca			<input type="button" value="Agregar Interno"/> 	
<input type="button" value="Registrar"/>						

*Figura 3.33* Intervención quirúrgica - hoja inicial

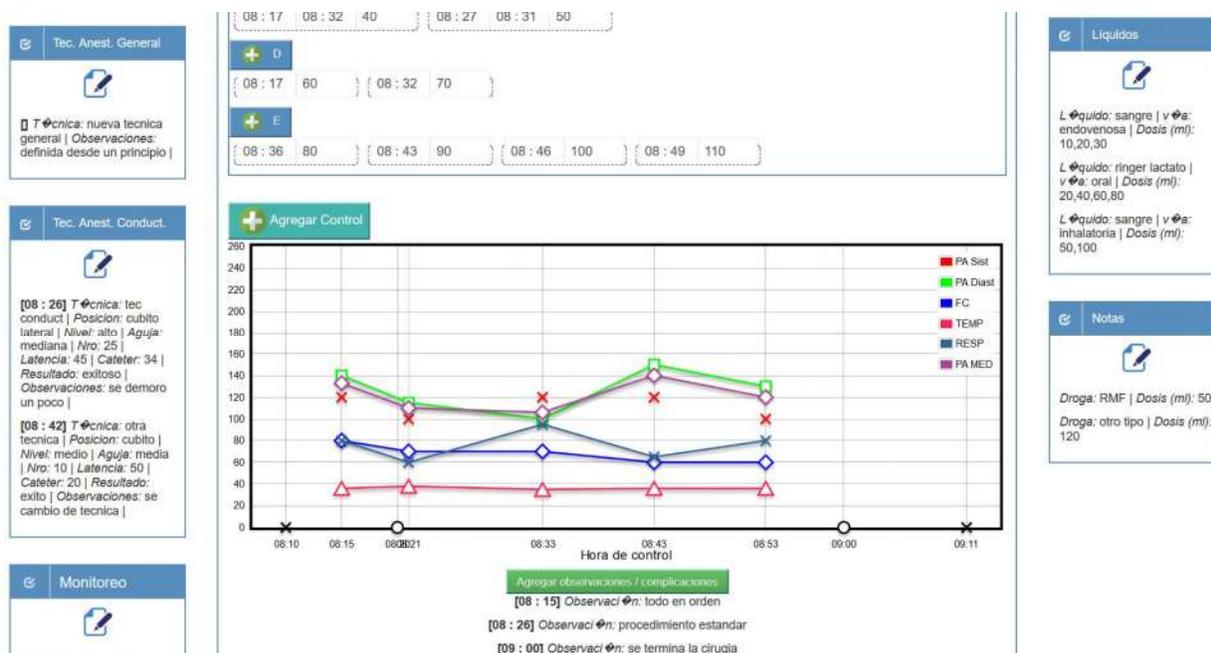


Figura 3.34 Intervención quirúrgica – hoja anestésica

### 3.5.6 Resultados del método UWE

El desarrollo de la aplicación web contempló un análisis completo de requisitos que complementado con los modelos y artefactos desarrollados, llevaron a implementar un sistema enfocado en brindar facilidades al usuario al momento de utilizar las diferentes funcionalidades de la aplicación.

### 3.7 SERVICIO REST

El servicio REST desarrollado permite la comunicación e intercambio de datos entre la aplicación web y la aplicación android. Este servicio cumple con el principio RestFul:

- Los recursos se obtienen explícitamente mediante verbos HTTP
  - GET, obtiene un recurso
  - POST, crea un nuevo recurso
  - PUT, actualiza un recurso
- Son peticiones sin estado (una misma petición devuelve el mismo recurso).
- Sigue una estructura jerárquica e intuitiva de acceso mediante URI's (uniform resource identifier)

- Se utiliza el formato JSON (JavaScript Object Notation) para el intercambio de recursos.

### 3.7.1 Descripción de los recursos

En la tabla 3.35 se describen los recursos proporcionados por el servicio rest:

**Tabla 3.35** Descripción del servicio REST

Método	URI	Descripción
GET	/rest/doctores	Devuelve un listado de doctores, claves y el tipo de doctor (anestesiólogo o residente)
GET	/rest/historial/{id}	Devuelve el historial completo y los antecedentes del id de registro anestésico proporcionado.
GET	/rest/pacientes/historiales/preanestesico	Devuelve listado de historiales clínicos para consulta pre anestésica
GET	/rest/pacientes/historiales/doctor	Devuelve listado de historiales clínicos para que el doctor pueda hacer el seguimiento de la recuperación post anestésica
GET	/rest/pacientes/historiales/interno	Devuelve listado de historiales clínicos para que el interno autorizado pueda registrar la recuperación post anestésica.
GET	/rest/pacientes/{inicialPaterno historiaClinica}	Devuelve un listado de todos los pacientes cuyo apellido paterno o parte de su historial clínico coincida con los parámetros proporcionados.
GET	/rest/paciente/{historiaClinica}	Comprueba si existe el paciente con la historia clínica proporcionada.
GET	/rest/recursos/preanestesico	Devuelve los recursos necesarios para iniciar el registro pre anestésico (listado de especialidades, procedimientos, diagnósticos, etc)

GET	/rest/recursos/postanestesico/{id}	Devuelve los recursos necesarios para iniciar el registro post anestésico del paciente con registro anestésico {id} (listado de equipos, prescripciones, escalas, etc).
GET	/rest/recursos/antecedentes/{historiaClinica}	Devuelve los antecedentes del paciente según su historia clínica.
POST	/rest/registro/preanestesico/{historiaClinica}	Se registra en la historia clínica del paciente la evaluación pre anestésica.
POST	/rest/registro/postanestesico/{id}	Los datos de la recuperación post anestésica se guardan en el registro anestésico {id}
PUT	/rest/registro/preanestesico/{id}	Se actualiza el registro pre anestésico {id} con una nueva observación, la fecha y el doctor responsable.
PUT	/rest/registro/postanestesico/estado/{id}	Se actualiza el estado del registro anestésico a “en RPA”.
PUT	/rest/registro/postanestesico/{id}	Se actualiza el registro post anestésico con las observaciones hechas por el anesthesiólogo a cargo.

### 3.8 MODELO FÍSICO DE LA BASE DE DATOS

La base de datos implementada permite tanto a la aplicación web como al servicio rest gestionar los usuarios, reportes y los datos de registros anestésicos de pacientes. En la figura 3.35 se muestra el modelo físico que se encuentra normalizado hasta la forma normal 2.

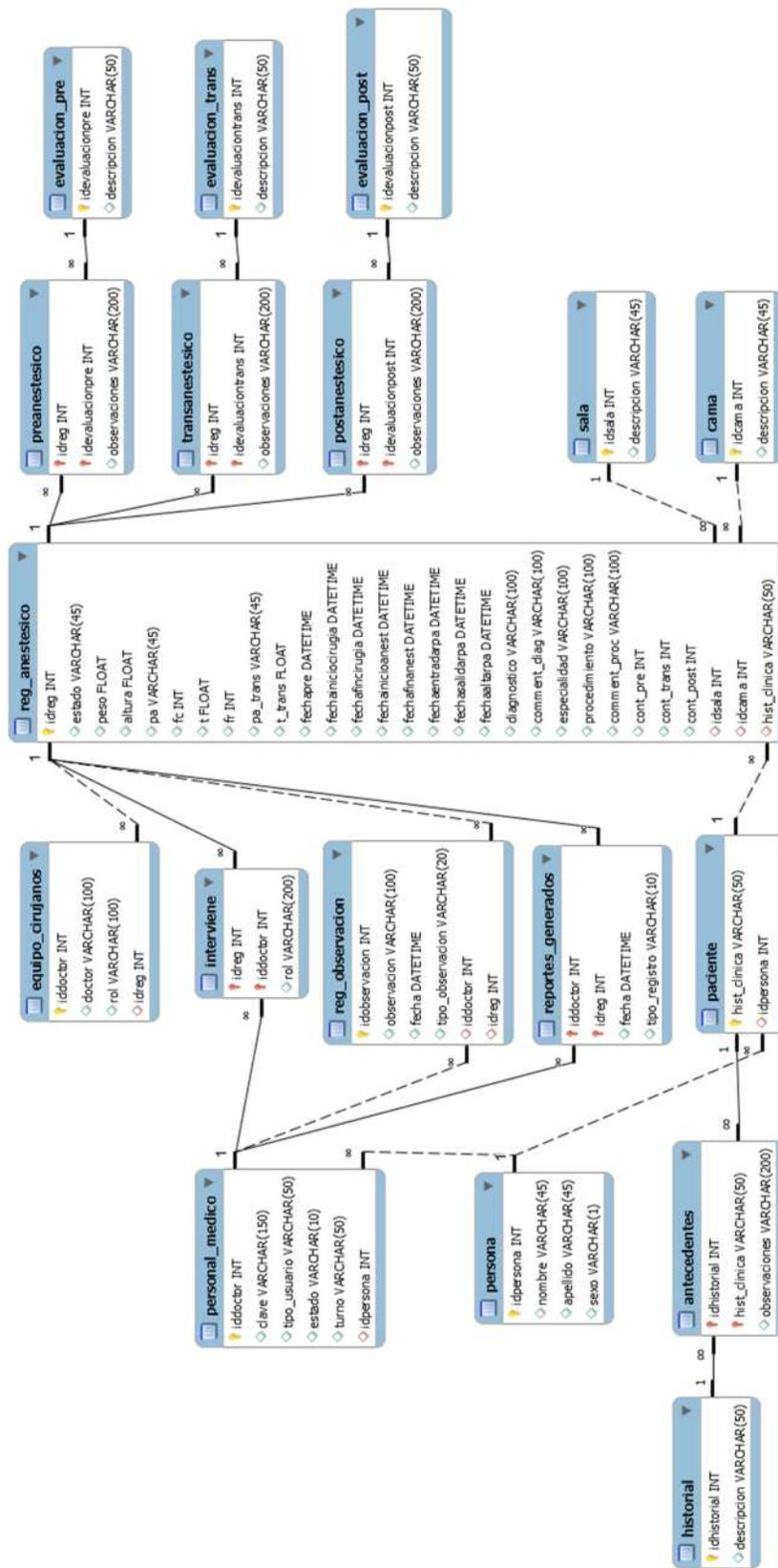


Figura 3.35 Modelo físico de la base de datos

## CAPITULO IV. CALIDAD Y SEGURIDAD

### 4.1 CALIDAD DEL SISTEMA

Se define la calidad en base a un conjunto de propiedades definidos en la norma ISO 9126:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

El objetivo principal es alcanzar la calidad necesaria para satisfacer las necesidades del Servicio de Anestesiología del Hospital de la Mujer.

#### 4.1.1 Funcionalidad

La funcionalidad se mide según la complejidad del conjunto de características y capacidades del sistema.

La métrica de punto función nos permite medir la funcionalidad desde la perspectiva del usuario, dejando de lado los detalles de codificación e implementación para lo cual se debe determinar las siguientes características:

- **Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona datos orientados a la aplicación.
- **Número de salidas de usuario.** Se cuenta cada salida que proporciona información orientada a la aplicación, se refiere a informes, pantallas, mensajes de error.
- **Número de peticiones de usuario.** Se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de archivos.** Se cuenta cada archivo maestro lógico (esto es un grupo lógico que puede ser parte de una gran base de datos o archivo independiente).

- **Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de disco) que se utilizan para transmitir información a otro sistema [Pressman, 2010]

Roger Pressman mide la siguiente relación:

$$(1)PF = CuentaTotal * \left( Grado\ de\ confiabilidad + Tasa\ de\ error * \sum F_i \right)$$

Donde:

*PF:* Medida de funcionalidad

*CuentaTotal:* Es la suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos

*Grado de confiabilidad:* Confiabilidad estimada del sistema, se considera un valor nominal de 65%

*Tasa de error:* Probabilidad subjetiva estimada del dominio de la información, este error es del 1%

*Fi:* Son los valores de ajuste de complejidad

#### a) Aplicación android

Los parámetros de medida para la aplicación android son:

**Tabla 4.1** Aplicación android: Cuenta total – factor de ponderación medio

Parámetro de medida	Cuenta		Factor de ponderación medio	Total
Nº de entradas de usuario	139	*	4	556
Nº de salidas de usuario	30	*	5	150
Nº de peticiones de usuario	10	*	4	40
Nº de archivos	5	*	10	50
Nº de interfaces externas	15	*	7	105
<b>CUENTA TOTAL</b>				901

**Tabla 4.2** Aplicación android: Ajuste de complejidad del punto función

<b>Factores</b>	<b>Significado</b>	<b>Escala F(i)</b>
¿Requiere el sistema copias de seguridad y de recuperación fiables?	MODERADO	2
¿Se requiere comunicación de datos?	ESENCIAL	5
¿Existen funciones de procesamiento distribuido?	MODERADO	2
¿Es crítico el rendimiento?	SIGNIFICATIVO	4
¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	ESENCIAL	5
¿Requiere el sistema entrada de datos interactiva?	SIGNIFICATIVO	4
¿Requiere la entrada de datos interactiva que las transacciones de entradas se lleven a cabo sobre múltiples pantallas u operaciones?	SIGNIFICATIVO	4
¿Se actualizan los archivos maestros de forma interactiva?	MEDIO	3
¿Son complejas las entradas, las salidas, los archivos o las peticiones?	MEDIO	3
¿Es complejo el procesamiento interno?	SIGNIFICATIVO	4
¿Se ha diseñado el código para ser reutilizable?	MEDIO	3
¿Están incluidas en el diseño la conversión y la instalación?	MODERADO	2
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	SIGNIFICATIVO	4
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	SIGNIFICATIVO	4
<b>TOTAL <math>\sum F_i</math></b>		<b>49</b>

Con los valores de ponderación y cuenta total hallados, reemplazamos en (1):

$$PF = CuentaTotal * (Grado de confiabilidad + Tasa de error * \sum F_i)$$

$$PF = 901 * [ 0.65 + 0.01 * 49 ]$$

$$PF = 1027.14$$

Ahora hallamos el punto de valor máximo para comparar los valores del sistema, tomando como referencia:  $\sum F_i = 70$

$$PF_{MAX} = CuentaTotal * (Grado de confiabilidad + Tasa de error * 70)$$

$$PF_{MAX} = 901 * [ 0.65 + 0.01 * 70 ]$$

$$PF_{MAX} = 1216.35$$

Por lo tanto la funcionalidad real de la aplicación android es:

$$FUNCIONALIDAD = \frac{1027.14}{1216.35} * 100\% = 84.4 \%$$

La funcionalidad estimada es del 84.4% tomando en cuenta el punto función máximo

#### b) Aplicación web

Los parámetros de medida para la aplicación web son:

**Tabla 4.3** Aplicación web: Cuenta total – factor de ponderación medio

Parámetro de medida	Cuenta		Factor de ponderación medio	Total
Nº de entradas de usuario	105	*	4	420
Nº de salidas de usuario	85	*	5	425
Nº de peticiones de usuario	25	*	4	100
Nº de archivos	15	*	10	150
Nº de interfaces externas	0	*	7	0
<b>CUENTA TOTAL</b>				1095

**Tabla 4.4** Aplicación web: Ajuste de complejidad del punto función

<b>Factores</b>	<b>Significado</b>	<b>Escala F(i)</b>
¿Requiere el sistema copias de seguridad y de recuperación fiables?	ESENCIAL	5
¿Se requiere comunicación de datos?	MODERADO	2
¿Existen funciones de procesamiento distribuido?	INCIDENTAL	1
¿Es crítico el rendimiento?	SIGNIFICATIVO	4
¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	ESENCIAL	5
¿Requiere el sistema entrada de datos interactiva?	SIGNIFICATIVO	4
¿Requiere la entrada de datos interactiva que las transacciones de entradas se lleven a cabo sobre múltiples pantallas u operaciones?	SIGNIFICATIVO	4
¿Se actualizan los archivos maestros de forma interactiva?	SIGNIFICATIVO	4
¿Son complejas las entradas, las salidas, los archivos o las peticiones?	MEDIO	3
¿Es complejo el procesamiento interno?	MEDIO	3
¿Se ha diseñado el código para ser reutilizable?	MEDIO	3
¿Están incluidas en el diseño la conversión y la instalación?	MODERADO	2
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	SIGNIFICATIVO	4
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	SIGNIFICATIVO	4
<b>TOTAL <math>\sum F_i</math></b>		<b>48</b>

Con los valores de ponderación y cuenta total hallados, reemplazamos en (1):

$$PF = CuentaTotal * (Grado de confiabilidad + Tasa de error * \sum F_i)$$

$$PF = 1095 * [ 0.65 + 0.01 * 48 ]$$

$$PF = 1237.35$$

Ahora hallamos el punto de valor máximo para comparar los valores del sistema, tomando como referencia:  $\sum F_t = 70$

$$PF_{MAX} = CuentaTotal * (Grado de confiabilidad + Tasa de error * 70)$$

$$PF_{MAX} = 1095 * [ 0.65 + 0.01 * 70 ]$$

$$PF_{MAX} = 1478.25$$

Por lo tanto la funcionalidad real de la aplicación web es:

$$FUNCIONALIDAD = \frac{1237.35}{1478.25} * 100\% = 83.7 \%$$

La funcionalidad estimada es del 83.7% tomando en cuenta el punto función máximo

#### 4.1.2 Fiabilidad

Se refiere a la capacidad de la aplicación para mantener su nivel de ejecución bajo condiciones normales en un período de tiempo establecido [Valle, 2009]

Se utilizará la función de distribución exponencial para modelar la fiabilidad, asumiendo una tasa constante de fallos, de esta manera se tiene la función de fiabilidad:

$$(2) R(t) = e^{-\lambda * t}$$

Donde:

$\lambda$  : tasa constante de fallos y está definido como

$$\lambda = \frac{nro. de fallos de acceso}{nro. total de accesos al sistema}$$

t: período de operación en el tiempo

Se realizaron pruebas por un período de 30 días evaluando cada componente del sistema (aplicación android y aplicación web), llegando a obtener los siguientes resultados:

**Tabla 4.5** Fiabilidad por componente

	Aplicación android	Aplicación web
$\lambda$	$\frac{2 [N^{\circ} \text{ de fallos}]}{210 [N^{\circ} \text{ de accesos}]} = 0.0095$	$\frac{3 [N^{\circ} \text{ de fallos}]}{350 [N^{\circ} \text{ de accesos}]} = 0.0086$
t	30 días	30 días
R(t)	0.75	0.77

La función de fiabilidad para componentes en paralelo es:

$$(3) R_S(t) = 1 - \prod_{i=1}^n (1 - R_i(t))$$

Entonces tenemos que la fiabilidad del sistema considerando sus componentes es:

$$R_S(t) = 1 - [ (1 - 0.75) * (1 - 0.77) ]$$

$$R_S(t) = 0.9425$$

Por lo tanto la fiabilidad del sistema en un período de prueba de 30 días es de 94%

#### 4.1.3 Usabilidad

Consiste en la evaluación del esfuerzo necesario que el usuario invertirá para usar el sistema, en base a la percepción y comprensión de la estructura lógica.

Se utiliza el test TAM de apropiación tecnológica donde se evalúan los siguientes aspectos:

- Utilidad percibida (UP)
- Facilidad de uso percibida (FUP)
- Actitud hacia el uso (AHU)

Se realizaron encuestas a los usuarios finales, mostrándose los resultados a continuación:

**Tabla 4.6** Utilidad Percibida

Escala de importancia								
UP	Absolutamente Improbable	Considerablemente Improbable	Poco improbable	Nada	Poco Posible	Considerablemente Posible	Absolutamente Posible	PERCEPCION
Me ayudaría a hacer mis actividades más rápido	1	2	3	4	5	6	7	<b>5</b>
Usar este recurso, mejoraría el desempeño de mi actividad profesional	1	2	3	4	5	6	7	<b>7</b>
Usar este recurso, incrementaría mi productividad	1	2	3	4	5	6	7	<b>7</b>
Usar este recurso aumentaría la efectividad en mis actividades profesionales	1	2	3	4	5	6	7	<b>6</b>
Usar este recurso facilitaría la realización de mis proyectos.	1	2	3	4	5	6	7	<b>6</b>
Encontraría este recurso útil en mis proyectos.	1	2	3	4	5	6	7	<b>6</b>
PROMEDIO: <b>6.2</b>				PROMEDIO PERCENTIL: <b>88.1%</b>				

**Tabla 4.7** Facilidad de uso percibida

<b>Escala de importancia</b>								
<b>FUP</b>	<b>Absolutamente Improbable</b>	<b>Considerablemente Improbable</b>	<b>Poco improbable</b>	<b>Nada</b>	<b>Poco Posible</b>	<b>Considerablemente Posible</b>	<b>Absolutamente Posible</b>	<b>PERCEPCION</b>
Aprender a utilizar este recurso fue fácil para mi	1	2	3	4	5	6	7	<b>5</b>
Mi interacción con este recurso fue claro y entendible	1	2	3	4	5	6	7	<b>6</b>
Encuentro este recurso flexible para interactuar con él	1	2	3	4	5	6	7	<b>6</b>
Sería fácil para mi llegar a ser un experto en el uso de este recurso	1	2	3	4	5	6	7	<b>7</b>
Encuentro a este recurso fácil de utilizar	1	2	3	4	5	6	7	<b>6</b>
<b>PROMEDIO: 6</b>				<b>PROMEDIO PERCENTIL: 85.7%</b>				

**Tabla 4.8** Actitud hacia el uso

Escala de importancia								
AHU	Absolutamente Improbable	Considerablemente Improbable	Poco improbable	Nada	Poco Posible	Considerablemente Posible	Absolutamente Posible	PERCEPCION
El uso de este recurso en mis actividades sería bueno	1	2	3	4	5	6	7	<b>6</b>
El uso de este recurso en mis actividades sería beneficioso	1	2	3	4	5	6	7	<b>7</b>
El uso de este recurso en mis actividades no sería absurdo.	1	2	3	4	5	6	7	<b>6</b>
El uso de este recurso en mis actividades sería positivo	1	2	3	4	5	6	7	<b>6</b>
El uso de este recurso en mis actividades sería placentero	1	2	3	4	5	6	7	<b>6</b>
PROMEDIO: <b>6.2</b>				PROMEDIO PERCENTIL: <b>88.6%</b>				

#### 4.1.4 Eficiencia

Las métricas de eficiencia comprenden tres aspectos:

a) Comportamiento en el tiempo

Podemos considerar el tiempo estimado para completar una tarea, aquí tiene una relevancia importante el servicio rest ya que es muy importante que el tiempo de respuesta sea lo más corto posible. La aplicación android y la aplicación web se desempeñan eficientemente

pero la latencia hasta acceder al servicio rest puede ser costoso en tiempo para completar un registro anestésico.

Producto del uso continuo del sistema se identificó los tiempos promedios para los diferentes tipos de peticiones rest

**Tabla 4.9** Eficiencia del servicio REST

Tipo de petición	Tiempo promedio de respuesta (1 usuario)	Tiempo promedio de respuesta (N usuario > 1)
GET	≈ 15ms	> 20ms
POST	≈ 40ms	> 70ms
PUT	≈ 10ms	≈ 10ms

Cabe aclarar que son tiempos estimados que podrían variar a medida que la base de datos incremente la cantidad de registro a consultar.

b) Utilización de recursos

La aplicación android se ejecuta en dispositivos con un mínimo de 512MBytes de RAM y Ice Cream Sandwich 4.0 como S.O., lo cual hoy en día la mayoría de dispositivos superan ampliamente.

La aplicación web se probó que funciona correctamente incluso en equipos netbooks con 1GigaByte de RAM.

c) Conformidad de la eficiencia

En la tabla 4.6 se mostró que la utilidad percibida por parte del usuario es del 80.9% debido principalmente a la rapidez del sistema.

#### 4.1.5 Mantenibilidad

Una parte importante del ciclo de vida de una aplicación es la implementación y mantenimiento, considerando que a futuro puede ser necesario que otro desarrollador haga mantenimiento del código o implemente nuevas funcionalidades.

El sistema está desarrollado con el estándar PSR-1, lo cual implica que el código es legible, se atiene a reglas de sintaxis y todos los métodos están documentados.

La base de datos se encuentra normalizada y mantiene la integridad referencial.

#### 4.1.6 Portabilidad

La portabilidad se refiere a la habilidad del software de ser transferido de un ambiente a otro. El entorno puede incluir entornos organizacionales de software y hardware.

- A nivel de software. El sistema puede ser puesto en producción en un servidor dedicado o virtual siempre que Apache y el DBMS MySQL estén instalados, también se puede iniciar el sistema en computadoras personales o portátiles ya que existen paquetes gratuitos que configuran el entorno para que el sistema pueda funcionar en los S.O. Windows (Wamp), Linux (Lamp) y Macintosh (Mamp),
- A nivel de hardware. El sistema es portable y adaptable si se cumple con las siguientes características mínimas:
  - CPU: Pentium IV o superior
  - RAM 512 MB
  - HDD: 2GB o más
  - Zona de acceso wifi disponible.

### 4.2 SEGURIDAD DEL SISTEMA

Para garantizar la seguridad se implementó las siguientes medidas en los diferentes componentes del sistema:

#### 4.2.1 Autenticación

Tanto en la aplicación web como en la aplicación android se requiere autenticarse para acceder a las funcionalidades del sistema, además el acceso es diferenciado según el tipo de usuario registrado.

En la aplicación web existe el usuario administrador que es el encargado de gestionar las claves de acceso de los demás usuarios y es el que define la clave de acceso común para internos. Además el framework Codeigniter proporciona de manera nativa protección contra ataques de tipo SQL injection y cross-site scripting XSS.

## 4.2.2 Encriptación

Debido a que la aplicación android será utilizada en un número indeterminado de dispositivos móviles no se crearon tokens para autenticar la conexión al servicio REST, en su lugar se optó por encriptar los recursos para evitar su acceso si son interceptados. La encriptación también se hace efectiva para almacenar las claves de acceso en la base de datos y así evitar que sean legibles en el remoto caso que alguien no autorizado acceda directamente a las tablas.

Se utiliza una encriptación personalizada para evitar alguna posible brecha de seguridad en métodos de encriptación conocidos. Los pasos son los siguientes:

1. Se concatenan los recursos solicitados y se ordenan en una matriz de manera que forme un cuadrado perfecto (el número de filas es igual al número de columnas).
2. En caso que la longitud concatenada de recursos no dé un cuadrado perfecto, se añade un carácter de relleno '@' las veces necesarias.
3. El mensaje se lee de arriba – abajo y de izquierda a derecha.

Se presenta un ejemplo práctico asumiendo que tenemos el nombre de una persona y el número de su tarjeta de crédito:

**Nombre (nom):** Roberto Perez Sanjinez

**Tarjeta de crédito: (tc) :** 145-123-789

**Información concatenada:**

nom::RobertoPerezSanjinez;tc::145-123-789

**Información encriptada:**

nbri:2@oeeen:3@mrze1-@:tSz47@:oa;58@RPnt-9@oejc1@@

Este sería el cuadrado perfecto, se eliminan los espacios y se agregan @:

n	o	m	:	:	R	o
b	e	r	t	o	P	e
r	e	z	S	a	n	j
i	n	e	z	;	t	c
:	:	1	4	5	-	1
2	3	-	7	8	9	@
@	@	@	@	@	@	@

*Figura 4.1* Encriptación del servicio REST

### 4.2.3 Base de datos

Para prevenir una potencial pérdida de información ante una situación imprevista, se generan automáticamente backups periódicos de la base de datos.

### 4.2.3 Hardware

A nivel de hardware el sistema físicamente se encuentra en un servidor de hosting para así evitar el acceso directo al sistema por parte de personal no autorizado, en caso que el sistema se configure en una maquina local, es importante que el acceso sea mediante clave y que la menor cantidad de usuarios tengan acceso al sistema.



## CAPITULO V. COSTO DEL SISTEMA

### 5.1 METODO DE ESTIMACIÓN CRI

El método de estimación de costos CRI propone como métricas las características de los campos de la base de datos (C), las relaciones entre las tablas (R) y la cantidad de vistas, consultas y procedimientos almacenados para visualizar datos en pantalla o en impresiones (I).

Este método propuesto por Gomez (2013) es particularmente útil para determinar el costo de sistemas como el SIGRA compuesto de una aplicación android, una aplicación web y un servicio REST ya que se centra en la base de datos y las propiedades de sus relaciones y complejidad de sus campos y no es necesario estimar el costo de cada componente por separado.

Para estimar el costo del sistema tenemos las siguientes ecuaciones

$$(1) \text{ Costo} = (UT_C + UT_R + UT_I) * \text{ValorMonetario (Bs/Hr)}$$

$$(2) UT_C = \sum_1^n (\text{Complejidad del campo}_i)$$

$$(3) UT_R = \sum_1^m (\text{Nro de tablas de la entidad compuesta}_m)$$

$$(4) UT_I = \sum_1^r (\text{Nro de tablas en el reporte}_r)^*$$

Debemos calcular (2), (3) y (4) en base a la figura 3.35 Modelo físico de la base de datos y la cantidad de reportes contemplados en la aplicación android y la aplicación web.

La tabla 5.1 muestra los resultados de la sumatoria de complejidad de los campos existentes en la base de datos.

**Tabla 5.1** Unidades de trabajo de los campos

<b>Tabla</b>	<b>SubTotal</b>
Antecedentes	4
Cama	3
Equipo_cirujanos	6
Evaluacion_post	3
Evaluacion_pre	3
Evaluacion_trans	3
Historial	3
Interviene	5
Paciente	3
Persona	5
Personal_medico	8
Postanestesico	5
Preanestesico	5
Reg_anestesico	57
Reg_observacion	12
Reportes_generados	8
Sala	3
Transanestesico	5
_anestesia	3
_cirugia	3
_cirujano	3
_comentario	3
_historico_interviene_residente	8
_medicacion_previa	3
_procedimiento	3
_rol	3

_observacion	3
_especialidad	3
<b>TOTAL UT<sub>c</sub></b>	<b>176</b>

En la tabla 5.2 calculamos las relaciones existentes

**Tabla 5.2** Unidades de trabajo de las relaciones

<b>Entidades compuestas</b>	<b>SubTotal</b>
Historia ; antecedentes ; paciente	3
Persona ; paciente	2
Persona ; personal_medico	2
Paciente ; reg_anestesico	2
Persona_medico ; reportes_generados ; reg_anestesico	3
Persona_medico ; reg_observacion	2
Reg_observacion ; reg_anestesico	2
Personal_medico ; interviene ; reg_anestesico	3
Equipo_cirujanos ; reg_anestesico	2
Reg_anestesico ; preanestesico ; evaluación_pre	3
Reg_anestesico ; postanestesico ; evaluación_post	3
Reg_anestesico ; transanestesico ; evaluación_trans	3
Reg_anestesico ; sala	2
Reg_anestesico ; cama	2
<b>TOTAL UT<sub>R</sub></b>	<b>34</b>

Y la tabla 5.3 nos muestra los reportes requeridos

**Tabla 5.3** Unidades de trabajo de los impresos

Aplicación	Reporte	Nro. tablas
Android	Listado de pacientes por historial	5
Android	Listado de pacientes por inicial de apellido	5
Android	Sumario registro pre anestésico	16
Android	Listado de historiales y estado	13
Android	Listado de equipos	4
Android	Listado de escalas y evaluaciones	6
Web	Reporte de pacientes en proceso de registro anestésico	14
Web	Reporte de pacientes con alta RPA	12
Web	Reporte de pacientes antiguos	15
Web	Reporte pre anestésico	16
Web	Reporte post anestésico	15
Web	Reporte trans anestésico	16
Web	Reporte actividades de personal	10
Web	Reporte cirugías por especialidad	16
Web	Reporte cirugías por anestesiólogo y residente	15
TOTAL UT <sub>1</sub>		178

Una vez obtenidos (2), (3) y (4), tenemos en (1), suponiendo el costo por hora 80 Bs:

$$\text{Costo} = (176 + 34 + 178) * 80 \text{ Bs}$$

$$\text{Costo} = 388 * 80 \text{ Bs}$$

$$\text{Costo} = * 80 \text{ Bs}$$

$$\text{Costo} = 31040 \text{ Bs.}$$

El método de estimación CRI nos indica, una vez que se obtienen las unidades de trabajo dentro del sistema, que el costo resultante de desarrollar e implementar el Sistema de Información para la Gestión del Registro Anestésico SIGRA asciende a 31040 Bs. (treinta y un mil cuarenta 00/100 bolivianos)



## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

El Sistema de Información para la Gestión del Registro Anestésico SIGRA fue diseñado, desarrollado e implementado satisfactoriamente en el Servicio de Anestesiología del Hospital de la Mujer, debido a las pruebas de calidad con el cliente se tuvieron varias iteraciones incrementales hasta obtener un producto que cumple con los requisitos y objetivos establecidos para el proyecto. Por tanto se llegan a las siguientes conclusiones respecto a los objetivos específicos:

- El objetivo de desarrollar una aplicación móvil Android se cumplió aplicando el método de Desarrollo Adaptable de Software ASD, teniendo como resultado una aplicación que brinda ayudas visuales y sugerencias de autocompletado en diagnósticos al usuario para que la elaboración de los registros pre y post anestésicos sea fácil e intuitiva.
- El diseño de la arquitectura cliente – servidor de la aplicación Web se realizó aplicando el método UWE. La aplicación web resultante permite al usuario registrar la fecha y hora de los controles de signos vitales, uso de agentes y la dosis exacta, técnicas y procedimientos mediante cuadros de diálogo para no saturar la pantalla de campos preestablecidos y agilizar el registro trans anestésico. De esta manera cumplimos el objetivo específico planteado.
- El objetivo de desarrollar un servicio web basado en REST para comunicar ambas aplicaciones se cumplió desarrollando un cliente REST en la aplicación Android y el servicio REST en la aplicación Web de forma que los datos consignados en ambas aplicaciones pueden ser consolidados para generar los tres registros anestésicos.
- El objetivo de implementar un acceso seguro y diferenciado al sistema se logra mediante el uso de contraseñas y roles de usuarios que tienen diferentes privilegios en ambas aplicaciones, también se encripta el servicio REST para evitar la lectura de los datos (por ejemplo contraseñas) en caso de que estos sean interceptados.
- El objetivo de diseñar reportes en base a los tres registros anestésicos se cumplió implementando un módulo de reportes y consultas en la aplicación Web. Los reportes impresos se obtienen una vez que termina todo el proceso anestésico pero se pueden

consultar cuando se requiera y en ellos están consignados todos los datos del registro anestésico del paciente.

- El objetivo de diseñar e implementar una base de datos normalizada para los tres registros anestésicos se cumplió de forma implícita al desarrollar el servicio REST, la aplicación Android y la aplicación Web. La base de datos resultante se encuentra normalizada hasta la forma normal 2 donde cada atributo que no forma parte de la clave primaria depende si o si de ésta y de esta forma garantizamos la integridad referencial de los datos.

Habiendo cumplido con los objetivos específicos, también cumplimos con el objetivo general del proyecto de grado: “Diseñar e implementar el Sistema de Información para la Gestión del Registro Anestésico SIGRA que gestione eficientemente el proceso de registro anestésico mediante el uso de servicios web y dispositivos móviles para proporcionar información precisa, oportuna y útil al Servicio de Anestesiología del Hospital de la Mujer”

## RECOMENDACIONES

A la conclusión del presente proyecto se tienen las siguientes recomendaciones para garantizar una gestión adecuada del registro anestésico:

- Los usuarios deberían renovar sus claves de acceso por lo menos una vez al mes para reducir los riesgos de acceso no autorizado.
- La clave de acceso común para internos debe cambiarse de ser posible cada día, el Jefe de Servicio es el que debe determinar el mejor momento.
- Para que los datos del registro trans anestésico sean lo más exactos posible, se recomienda que los doctores registren al momento cualquier procedimiento o técnica utilizada durante la intervención.
- Se debe tratar de limitar la cantidad de dispositivos móviles que hacen uso de la aplicación Android, para el caso de los internos es recomendable que el Servicio de Anestesiología cuente con uno o dos dispositivos específicamente para su uso.
- El sistema funciona siempre que el punto de acceso wifi esté activo y los equipos de escritorio cuenten con las condiciones para operar normalmente, se recomienda el uso de UPS u otro tipo de generadores eléctricos para garantizar el uso del sistema.

- Una siguiente versión del sistema puede incluir una interface directa con los equipos de control y monitoreo de quirófano, para esto se recomienda una colaboración estrecha y disponer de acceso abierto para estudiar estrategias de integración.



## REFERENCIAS

### Referencias bibliográficas

Pressman, Roger S. (2010). *Ingeniería del Software*, México: Editorial Mc Graw Hill, Séptima Edición.

Sampieri, Roberto H. (2014). *Metodología de la Investigación*, México: Editorial Mc Graw Hill, Sexta Edición.

### Referencias Web

Aguirre Aguirre, Fernando Patricio. (2013). *Registro Anestésico*. Recuperado de: <http://patricioaguirreaguirre.blogspot.com/2013/03/registro-anestesico.html>

Alvarez Caules, Cecilio. (2013). *Introducción a servicios REST*. Recuperado de: <http://www.arquitecturajava.com/servicios-rest/>

Alvarez, Miguel Angel. *Qué es MVC*, Recuperado de: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>

Marquès, Asier. (2013). *Conceptos sobre API REST*. Recuperado de: <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>

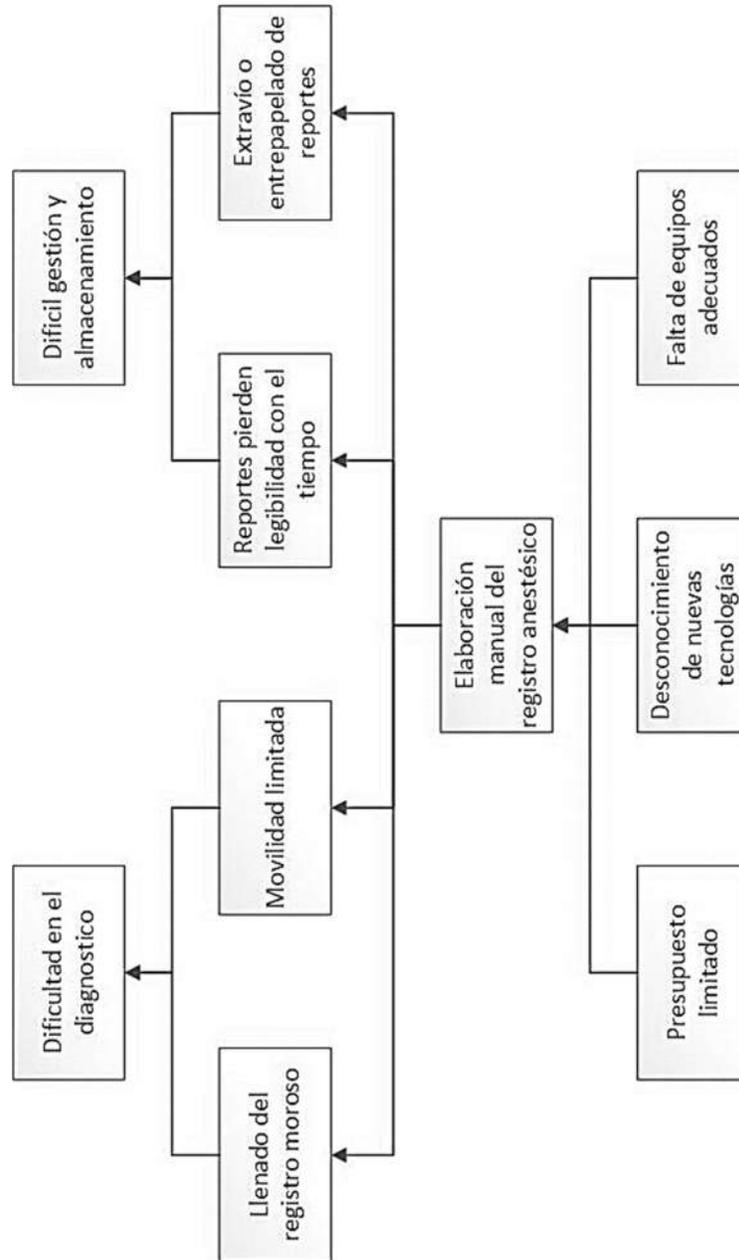
Universidad Unión Bolivariana, Ingeniería de Software, *Método Ágil ASD*. Recuperado de: [http://ingenieriadesoftware.mex.tl/61154\\_ASD.html](http://ingenieriadesoftware.mex.tl/61154_ASD.html)

### Documentos en versión electrónica

Bianco, Gustavo (2013) *Desarrollo de una Ficha Anestésica Web en Áreas Críticas*. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/31848>

## ANEXOS

## ANEXO A



*Figura A.* Árbol de problemas causa – efecto

ANEXO B

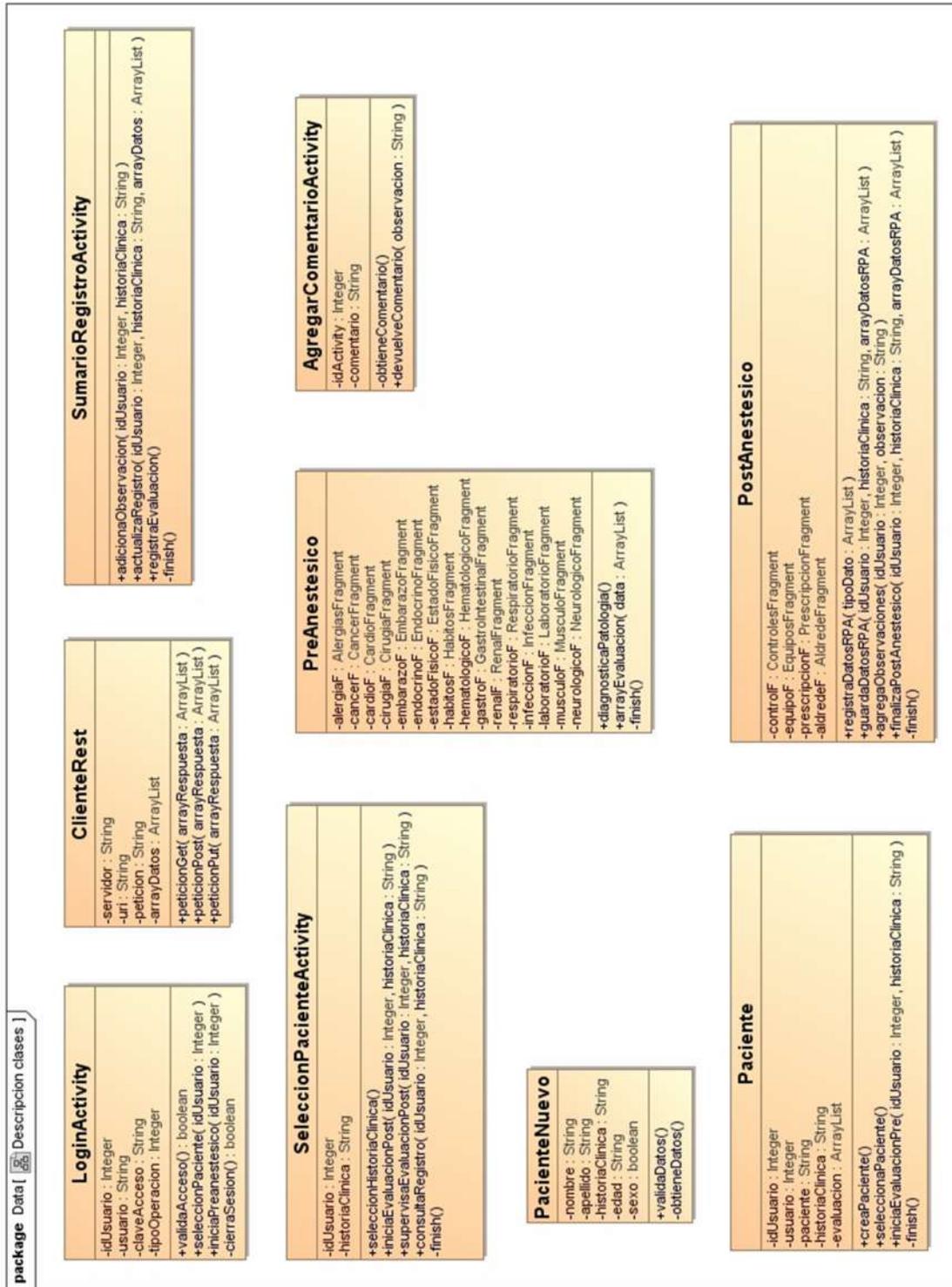


Figura B. Descripción de clases – Aplicación android

## ANEXO C

CUESTIONARIO PARA DETERMINAR LA APROPIACIÓN TECNOLÓGICA								
Nombre del Recurso	SIGRA							
	ESCALA DE IMPORTANCIA							OBSERVACIONES
Utilidad Percibida (UP)	Absolutamente Improbable	Considerablemente Improbable	poco Improbable	Nada	Poco Posible	Considerablemente posible	Absolutamente posible	
Me ayudaría a hacer mis actividades más rápido	1	2	3	4	5	6	7	
Usar este recurso, mejoraría el desempeño de mi actividad profesional	1	2	3	4	5	6	7	
Usar este recurso, incrementaría mi productividad	1	2	3	4	5	6	7	
Usar este recurso aumentaría la efectividad en mis actividades profesionales	1	2	3	4	5	6	7	
Usar este recurso facilitaría la realización de mis proyectos.	1	2	3	4	5	6	7	
Encontraría útil en mis proyectos	1	2	3	4	5	6	7	

Figura C1. Cuestionario – Utilidad percibida UP

	ESCALA DE IMPORTANCIA							OBSERVACIONES
Facilidad de Uso Percibida (FUP)	Absolutamente Improbable	Considerablemente Improbable	poco Improbable	Nada	Poco Posible	Considerablemente posible	Absolutamente posible	
Aprender a utilizar este recurso sería fácil para mí	1	2	3	4	5	6	7	
Mi interacción con este recurso sería clara y entendible	1	2	3	4	5	6	7	
Encuentro este recurso flexible para interactuar con él	1	2	3	4	5	6	7	
Sería fácil para mí llegar a ser un experto en el uso de este recurso	1	2	3	4	5	6	7	
Encuentro a este recurso fácil de utilizar	1	2	3	4	5	6	7	

Figura C2. Cuestionario – Facilidad de uso percibida FUP

	ESCALA DE IMPORTANCIA							OBSERVACIONES
Actitud Hacia el Uso (AHU)	Absolutamente Improbable	Considerablemente Improbable	poco Improbable	Nada	Poco Posible	Considerablemente posible	Absolutamente posible	
El uso de este recurso en mis actividades sería bueno	1	2	3	4	5	6	7	
El uso de este recurso en mis actividades sería beneficioso	1	2	3	4	5	6	7	
El uso de este recurso en mis actividades no sería absurdo	1	2	3	4	5	6	7	
El uso de este recurso en mis actividades sería positivo	1	2	3	4	5	6	7	
El uso de este recurso en mis actividades sería placentero	1	2	3	4	5	6	7	

Figura C3. Cuestionario – Actitud hacia el uso AHU

## ANEXO D



**Datos del Paciente**

Servicio  
Obstetricia

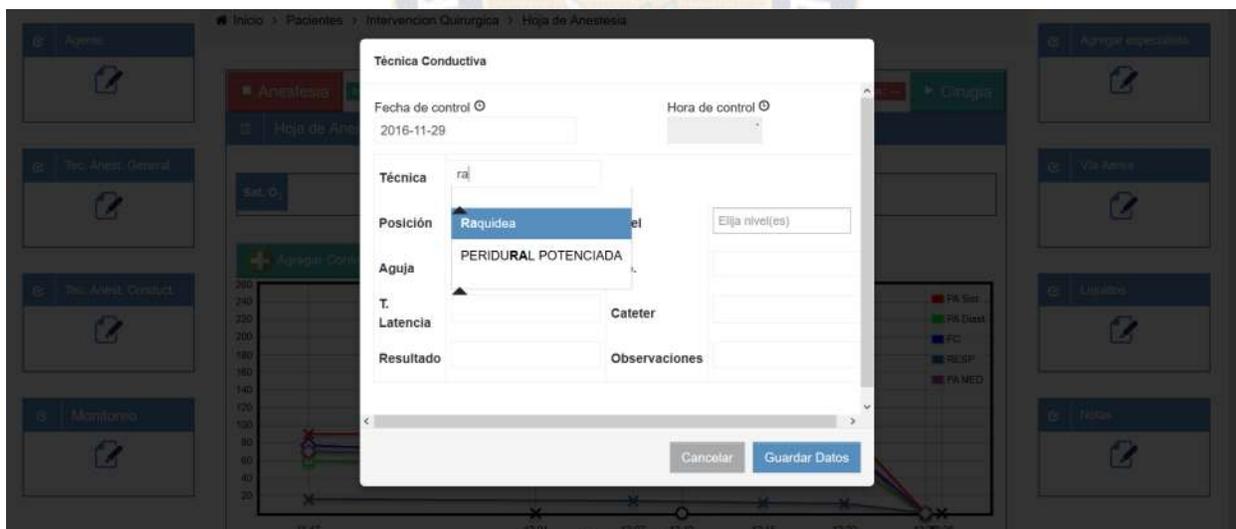
Diagnóstico  
su

Sufrimiento fetal agudo

Cesarea

PROCEDER

*Figura D1.* Auto completado – Aplicación android



Técnica Conductiva

Fecha de control 2016-11-29

Hora de control

Técnica ra

Posición Raquídea

Aguja PERIDURAL POTENCIADA

T. Latencia

Cateter

Resultado

Observaciones

Cancelar Guardar Datos

*Figura D2.* Auto completado en cuadro de dialogo – Aplicación web