

Universidad de las Ciencias Informáticas

Facultad 7



**Implementación de los procesos gineco - obstétricos del Sistema
de Información Hospitalaria alas HIS**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores: Rubén Alejandro Piñeiro Delgado
Boris Luis Vasallo Lominadze

Tutor: MSc. Karel Gómez Velázquez

Cotutor: Ing. Reinel Muñoz Pérez

La Habana, 30 junio de 2011

“Año 53 de la Revolución”

DATOS DE CONTACTO

Tutor

MSc. Karel Gómez Velázquez: Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2007. Posee la Categoría Docente de Profesor Asistente. Ha impartido asignaturas de Práctica Profesional 1, Introducción a la Programación, Programación 1 y Principios de Algoritmización, así como cursos optativos o de postgrado. Ha participado en varios proyectos de desarrollo vinculados específicamente con el perfil de salud. Actualmente labora en el Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como Jefe de departamento y Líder de desarrollo del Sistema de Información Hospitalaria alas HIS. Egresado de la Maestría en Informática Aplicada.

Correo electrónico: kgomez@uci.cu

Cotutor

Ing. Reinel Muñoz Pérez: Profesor Instructor, graduado en el año 2009, de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor vinculado a la Facultad 7 y miembro del Departamento de Sistemas de Gestión Hospitalaria.

Correo electrónico: rmunoz@uci.cu

AGRADECIMIENTOS

Agradecemos a nuestros familiares por siempre apoyarnos a lo largo de nuestra vida y sobre todo por estos últimos años.

Agradecemos a todos los profesores de nuestro proyecto productivo que de una forma u otra nos han ayudado a formarnos como profesionales, en especial a nuestros tutores, a la profesora Lorena y a nuestro oponente.

A todos nuestros amigos por siempre apoyarnos y estar ahí cuando los necesitamos.

A Isnel por habernos brindado su ayuda en todo momento.

A Mirna por todos los malos ratos que le hicimos pasar durante este año.

DEDICATORIA

De Boris

A mi madre y mi padre que para mí son lo más grande en este mundo y siempre han apoyado todas mis decisiones. Les doy gracias por ayudarme a ser la persona que soy y por creer en mí.

A mi hermano y mi cuñada por apoyarme a pesar de que no se lo pongo fácil.

A todos mis demás familiares por su interés y apoyo en todas mis cosas.

Mis amistades también tienen un gran peso para mí y les agradezco a todos ellos su apoyo en todas las etapas de mi carrera, en especial a Nuria, Yadia, Yasmany, al Primo, Eddy y Dany que vienen conmigo desde primer año aguantándome todo este tiempo y son casi de mi familia. No mencionar al piquete de hospitales sería olvidarme de los compañeros de lucha, le agradezco a todos, a Ray, Noslen, Cuquin, Ariel, Carlitos, el mamacó, a Piedra, a Leo.

Al Maino que siempre nos tendió la mano cuando necesitamos ayuda, junto con todo el grupo del laboratorio 79105, ustedes son los mejores.

A todos los que no he mencionado pero que también han estado presentes para mí en todo momento, a todos ellos gracias.

De Rubén

A mi papá que siempre me ha brindado su apoyo.

A todas mis amistades que de una forma u otra han contribuido a mi formación.

A las viejas que siempre me han guiado por el camino correcto y que tanto me quieren.

A Robert por ser un padre maravilloso, con el que siempre he podido contar, sin importar lo difícil de la situación.

A toda mi familia que me aguantado tantos años de trabajo y sacrificio.

A mi Tata, Pelly, Tito, Migue y mi hermanita hermosa por quererme y siempre estar al tanto de mis estudios.

A mi CT por estar siempre ahí sin importar lo complicado de la situación, y a los caballos de hospitales: Nos, Mamacó, los Carlays, Rams, Piedruscony, Ray, Chals y Yayi.

A mi madre que es mi inspiración y ejemplo a seguir, quien siempre ha estado compartiendo cada tristeza, alegría y por apoyarme en cada decisión de mi vida, gracias

RESUMEN

El procesamiento de los datos en el servicio de Ginecología-obstetricia se realiza de forma manual, se evidencian retrasos en la atención al paciente, principalmente a la hora de obtener sus datos, pues no es posible gestionar los formatos de parto, aborto, legrado y evoluciones de preparto o postparto así como la Hoja de Recién Nacido, implicando por consiguiente retrasos en la generación de los reportes estadísticos en este servicio.

El presente trabajo de diploma se propone como objetivo implementar los procesos del servicio de Gineco-obstetricia del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la información en esta área de las instituciones hospitalarias

Para llevar a cabo el sistema anteriormente descrito se propone utilizar el Proceso Unificado de Desarrollo (RUP) y se basa en tecnologías libres, multiplataforma y sobre una arquitectura en capas. El mismo utiliza Java como lenguaje de programación e implementa el patrón de arquitectura Modelo Vista Controlador (MVC). Como Sistema de Gestión de Bases de Datos es usado el PostgreSQL y como servidor de aplicaciones el JBoss Application Server. Para obtener una interfaz visual moderna y realizar eficientemente las peticiones al servidor son utilizadas las librerías RichFaces.

La implementación de los procesos gineco-obstétricos permitirá que la información de las embarazadas dentro de las instituciones hospitalarias sea fiable y de fácil acceso, llevando además toda la gestión de los antecedentes obstétricos de cada paciente. Se podrán obtener reportes estadísticos más específicos del servicio de Gineco-obstetricia.

Palabras claves: *Ginecología, Obstetricia, preparto, parto, postparto, aborto, legrado, HIS*

Introducción	1
CAPÍTULO 1. Fundamentación teórica	6
1.1. Conceptos básicos asociados al problema	6
1.2. Sistemas automatizados existentes vinculados al campo de acción	7
1.3. Tendencias y tecnologías actuales a considerar	10
1.4. Tecnologías y herramientas utilizadas en el proceso de desarrollo	11
1.5. Tecnologías y metodologías horizontales	16
1.6. Herramientas	18
CAPÍTULO 2. Descripción de la arquitectura	20
2.1. Requerimientos no funcionales.....	20
2.2. Descripción de la arquitectura	23
2.3. Análisis de componentes reutilizables.....	24
2.4. Seguridad.....	25
2.5. Vista de Despliegue	26
2.6. Estrategias de codificación. Estándares y estilos a utilizar.....	27
CAPÍTULO 3. Descripción y Análisis de la solución propuesta	29
3.1. Valoración crítica del diseño propuesto por el analista	29
3.2. Descripción de las nuevas clases u operaciones necesarias.....	39
3.3. Modelo de Datos.....	49
3.4. Breve valoración de las técnicas de validación.....	62
3.5. Vista de Implementación	63
CAPITULO 4. Modelo de pruebas	65
4.1. Prueba de caja negra.....	65
4.2. Descripción de los casos de pruebas.....	67
Conclusiones	82
Recomendaciones	83
Referencias bibliográficas	84
Bibliografía	87

Introducción

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) comienzan a desarrollarse sistemas informáticos encargados de gestionar la información, los cuales se han aplicado en la mayoría de los sectores que proveen servicios a la sociedad y en específico al sector de la salud.

La medicina manipula mucha información, la cual debe estar disponible en el momento de la toma de decisiones, tanto diagnósticas, terapéuticas como administrativas. Actualmente se aplica cada vez más tecnología médica digital, incluyendo sistemas de información. Los sistemas de información son un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, desarrollados para resolver una necesidad.

Entre estos sistemas se encuentran los Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés). Un HIS es un sistema de información orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria. Estos posibilitan la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes burocráticos que enfrentan los pacientes. [1]

El sector de la salud siempre ha sido priorizado por su impacto en la calidad de vida de la población, lo que conlleva a reforzar y fomentar la implementación de nuevas tecnologías que permitan un desarrollo creciente del mismo, para lograr una mejora en la calidad de vida de la población. Los HIS permiten que a partir de ellos se puedan obtener reportes e informes estadísticos de las diferentes áreas o servicios de las instituciones hospitalarias; trayendo consigo una mejora en el funcionamiento de las mismas. [2]

Acorde al tratamiento que necesita cada paciente o al estado en que se encuentran los mismos existen varios niveles de atención: [3]

Asistencia primaria: Son los cuidados que proporcionan la atención básica a la salud de la población, servicios terapéuticos y coordinación de todas las necesidades y servicios comunitarios. Constituye el primer contacto del usuario con el sistema sanitario. Por ella se accede a niveles asistenciales más complejos. Los centros de atención primaria, deben solucionar el 70 u 80% de los problemas asistenciales cotidianos. Incluye el médico de cabecera, pediatra, asistencia social, radiología, laboratorio y urgencias.

Asistencia secundaria: Segundo nivel asistencial, que incluye operaciones sencillas, exploraciones y otras tareas del policlínico que no implican larga hospitalización, aunque sí (en algunos casos) tratamiento en camas hospitalarias de este nivel o atención médica domiciliaria. Comprende la atención de naturaleza especial, generalmente compleja y que no puede ser administrada por el médico general.

Atención terciaria: El nivel terciario debe abarcar alrededor del 5% de los problemas de salud, relacionados con secuelas o aumento de las complicaciones de determinadas dolencias. Se brindan servicios de muy alta complejidad, con la óptima utilización de los recursos y medios existentes y el desarrollo de la investigación. A este nivel pertenecen los institutos y hospitales especializados.

Una de las áreas de mayor importancia dentro de un hospital es la de hospitalización, porque es la que se encarga de la gestión de la información de los pacientes admitidos. Esta área está compuesta por diferentes servicios médicos que son los que les prestan la atención a los pacientes hospitalizados. Uno de ellos es el de Gineco-obstetricia, en el cual se le brinda atención especializada a las embarazadas y los recién nacidos durante las fases de parto, parto y postparto.

Las madres y los niños son los más vulnerables en términos de contagio de enfermedades y mortalidad, puesto que ambos, después del parto, se encuentran en una situación delicada de salud y necesitan las mejores atenciones por parte de personal especializado.

Por estas razones los procesos gineco-obstétricos son de gran importancia ya que recogen toda la información de las embarazadas y de los recién nacidos en las diferentes salas de este servicio. Por la condición delicada que presentan las embarazadas y los recién nacidos en esta etapa de la vida es de gran importancia la búsqueda de la información de los antecedentes personales y familiares de la madre, ya que son vitales para lograr una atención eficiente a los mismos.

El área de Sala de Parto permite recoger todos los datos del recién nacido y de la embarazada durante el parto, parto y postparto. Además se contemplan las emergencias obstétricas y las cesáreas, con sus respectivas gestiones quirúrgicas relacionadas con estos procedimientos. También en esta área se realizan los abortos y legrados, los cuales se registran y pasan a formar parte de la Historia Clínica del paciente. Las cesáreas son tramitadas en los servicios de Cirugía, Enfermería y Farmacia.

Existe información referente a la embarazada, mediante el seguimiento del peso, la tensión arterial y la talla que se recogen desde la primera consulta en su área de salud o policlínico para garantizar un buen término del embarazo, información a tener en cuenta para un posible riesgo en el parto o cesárea. La falta de comunicación entre las áreas de salud y las instituciones hospitalarias trae consigo que se tenga que interrogar a la embarazada una vez llegue a dicha institución para recoger toda la información necesaria que ya está recogida en las áreas de salud, por lo que se trata de información duplicada que entorpece la atención al paciente. [4]

El procesamiento de los datos en el servicio de gineco-obstetricia se realiza de forma manual, lo que puede traer consigo la pérdida de información, por deterioro o de los archivos físicos. Así como en la generación de datos duplicados, por lo que no se garantiza una adecuada gestión de la información en dicho servicio. También se evidencian retrasos en la atención al paciente, principalmente a la hora de obtener sus datos, pues estos están en archivos físicos y también existen retrasos en la generación de los reportes estadísticos.

En este servicio, además, se presta atención a mujeres que no están embarazadas, pero que presentan problemas ginecológicos como pueden ser las alteraciones en el ciclo menstrual, escozor en los genitales externos, pérdida de la orina de forma espontánea, entre otros. Toda la información que se genera es recogida e incorporada al historial clínico de cada paciente.

Una vez conocida la situación problemática actual que afecta el desarrollo de los procesos gineco-obstétricos dentro del área de Hospitalización el **problema a resolver** consiste en: ¿Cómo facilitar la gestión de la información de los procesos del Servicio de Gineco-obstetricia en las instituciones hospitalarias?

Teniendo en cuenta el problema planteado se define como **objeto de estudio**: el Proceso de gestión de la información en las instituciones hospitalarias. El **campo de acción** del presente trabajo está centrado en el Proceso de gestión de la información del servicio de Gineco-obstetricia en las instituciones hospitalarias.

Para resolver el problema identificado anteriormente se propone como **objetivo general**: Implementar los procesos del servicio de Gineco-obstetricia del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la información en esta área de las instituciones hospitalarias.

Para lograr el objetivo general se trazaron las siguientes tareas de investigación:

- Refinar los procesos de negocio asociados al servicio de Gineco-obstetricia de las instituciones hospitalarias.
- Evaluar las tendencias actuales de los Sistemas de Información Hospitalaria, vinculados con el servicio de Gineco-obstetricia.
- Aplicar la arquitectura definida para el desarrollo de sus aplicaciones por el Departamento de Sistemas de Gestión Hospitalaria.
- Obtener los artefactos correspondientes a los flujos de trabajo de “Implementación” y “Pruebas”.
- Implementar los procesos gineco-obstétricos para el Sistema de Información Hospitalaria alas HIS.
- Obtener el acta de validación de la documentación y de la implementación, emitida por el grupo de calidad del Departamento de Sistemas de Gestión Hospitalaria.

La implementación de los procesos gineco-obstétricos en el Sistema de Información Hospitalaria alas HIS, brindará una serie de mejoras a la atención al paciente que aumentarán la calidad del servicio prestado al mismo, entre ellas se encuentran:

- Integración y sincronización de la información obstétrica de la embarazada.
- Creación de la Historia Clínica del recién nacido, así como la incorporación en la misma de los antecedentes familiares, prenatales, perinatales y postnatales.
- Generación de reportes asociados con la atención a pacientes en área de Obstetricia, mejorando el control de los partos mediante el libro de Sala de partos y haciendo más completos los reportes que actualmente brinda el sistema.
- La información clínica del paciente será de fácil y rápido acceso para facilitar los procesos de diagnóstico, tratamiento y seguimiento.
- Se genera información única en todo el proceso de nacimiento del recién nacido eliminando los datos duplicados que se recogen en las diferentes áreas por la que transitan los pacientes.

El documento se encuentra estructurado en cuatro capítulos cuyo contenido se describe a continuación:

CAPÍTULO 1: Fundamentación teórica: Son expuestos los diferentes fundamentos teóricos asociados con el campo de acción de la investigación, haciendo además un análisis valorativo de algunos sistemas que constituyen antecedentes para el desarrollo de los procesos gineco-obstétricos de las instituciones hospitalarias. También son justificadas las tecnologías, metodología y herramientas utilizadas para el desarrollo de estos procesos.

CAPÍTULO 2: Descripción de la arquitectura: Especifica los requerimientos no funcionales y se describe la concepción arquitectónica definida por el Departamento de Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones. Se definen las estrategias de codificación, así como los estándares y estilos a utilizados.

CAPÍTULO 3: Descripción y Análisis de la solución propuesta: Se describe cómo fue implementada la propuesta de solución para una gestión informatizada de los procesos hospitalarios vinculados al servicio de Gineco-obstetricia.

CAPÍTULO 4: Modelo de Pruebas: Se describen los casos de prueba, el método de pruebas a utilizar y se obtiene la lista de no conformidades y los flujos de procesos de integración.

CAPÍTULO 1. Fundamentación teórica

En el presente capítulo se presentan los conceptos asociados al problema identificado y los sistemas existentes vinculados al campo acción, se demuestra la necesidad de desarrollar una nueva solución de acuerdo a la situación problémica actual del Servicio de Gineco-Obstetricia en las instituciones hospitalarias. Como aspectos básicos se abordan los principales conceptos, los antecedentes de sistemas similares y se hace referencia a las diferentes tecnologías y herramientas que serán utilizadas en el desarrollo de la solución que se propone.

1.1. Conceptos básicos asociados al problema

Para una mejor comprensión de la situación asociada al problema a solucionar, se explican a continuación diferentes términos propios del entorno vinculado al campo de acción. [2]

Analgesia: Falta o supresión de toda sensación dolorosa, sin pérdida de los restantes modos de la sensibilidad.

Anestesia: Falta o privación general o parcial de la sensibilidad.

Cirugía: Parte de la medicina que tiene por objeto curar las enfermedades por medio de intervenciones quirúrgicas.

Cirujano: Médico especialista en Cirugía que realiza las intervenciones quirúrgicas.

Stock: Cantidad de mercancías que se tienen en depósito.

Shock: Estado en el que puede entrar la paciente luego de ser intervenida quirúrgicamente.

Psicoprofilaxis: Preparación que se le brinda a la embarazada en los centros de atención primaria durante el período gestacional.

Presentación: Posición en que se encuentra el feto antes de nacer.

1.2. Sistemas automatizados existentes vinculados al campo de acción

Un HIS posee varios módulos, cada uno está orientado a las diferentes áreas dentro de una institución hospitalaria. Dentro de estas áreas se encuentra el área de Hospitalización, el cual se divide en servicios médicos, uno de ellos el de Gineco–Obstetricia brinda la atención a las mujeres embarazadas en todo el período de gestación y a pacientes femeninos con problemas ginecológicos, además de gestionar toda la información que se genera en este proceso.

Como parte de la investigación sobre los antecedentes a nivel mundial de los HIS se obtuvieron los resultados que se muestran a continuación:

METIS HIS

Es un sistema Web que permite el acceso tanto al paciente como al personal clínico, personal administrativo y dirección de la institución.

Consta con una serie de módulos clínicos específicos, entre los que se encuentran: Historia clínica de emergencia, Historia clínica electrónica de policlínica (Web), Sanatorio y Block quirúrgico y Coordinaciones, especializado para el registro del acto quirúrgico y las coordinaciones de las intervenciones. Gestiona, aunque no de manera especializada, los servicios gineco-obstétricos de las instituciones, pero no un seguimiento completo de las embarazadas.

Está desarrollado con GeneXus (versión 9.0) y Pattern PXTools. Fue generado en Java 3 capas. Los reportes dinámicos y datawarehouse fueron desarrollados con GXplorer utilizando la prestación FULL WEB. El producto funciona en los navegadores Internet Explorer y Mozilla Firefox. Es un sistema propietario aunque permite interacción con sistemas de terceros. [4]

SIGHO

Es un sistema que facilita las actividades de gerencia dentro de un hospital. Se compone por 18 módulos. Este permite realizar registros individuales alrededor de la Historia Clínica Electrónica. Está diseñado para un hospital de 120 camas, aunque puede ser adaptado para hospitales más pequeños (30 a 60 camas). Puede ser instalado de forma progresiva y escalonada al interior de las instituciones hospitalarias.

Este es una solución que se apoya en estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos, tales como el CIE-10 y CIE-9MC. Para su desarrollo se utilizaron las mejoras realizadas en los Servicios de Internet Information Server 6.0 (IIS 6.0), Microsoft ASP.NET y Microsoft NET Framework. Como gestor de base de datos utiliza SQL Server 2000.

El primer componente que debe tener instalado el equipo Servidor es el Sistema Operativo Windows y en una versión Server 2000 o superior.

Dentro de sus funcionalidades se encuentran: seguimiento de valoración gineco-obstétrica ambulatoria, seguimiento del embarazo, seguimiento a terminación del embarazo, registro de datos de nacimiento y seguimiento de hospitalización. Su principal desventaja es que no se gestionan los antecedentes obstétricos, las evoluciones de parto y postparto. [5]

SALGEN

El Sistema de Atención Longitudinal a las Gestantes y el Neonato (SALGEN) permite una alta eficiencia en el análisis estadístico de las pacientes así como darle seguimiento y conocer su situación en cualquier momento. Para ello se toman en cuenta datos de consultas y exámenes genéticos en tiempo real, con vistas a actuar ante cualquier tipo de alteración de los análisis. Este sistema facilita, además, la adopción de medidas de salud necesarias para trabajar con un enfoque preventivo.

Fue implementado en las áreas de atención de los ocho municipios de la provincia Sancti Spíritus y tiene posibilidades de ser aplicado en cualquier otra región. Para la realización de este producto se utilizaron herramientas de sistemas informáticos de código abierto. Se utilizó como lenguaje de programación PHP, como gestor de base de datos PostgreSQL. Como resultado se obtuvo un sistema multiusuario y multiplataforma. Como desventaja presenta que al solo gestionar la información de las pacientes durante el embarazo no maneja lo referente a la Historia Clínica del paciente así como los antecedentes familiares, además tampoco se gestiona la información de los servicios de Gineco-obstetricia que se prestan en las instituciones hospitalarias a mujeres que no están embarazadas, pero que presentan problemas ginecológicos. [6]

CONTROL SALUD

Es un sistema de gestión integral multiusuario desarrollado para consultorios de ginecología y obstetricia. Consta con diferentes módulos entre los que se encuentran el Pediátrico, Oftalmológico y el Gineco-obstétrico.

Dentro de sus funcionalidades se encuentran el seguimiento durante el embarazo, los antecedentes ginecológicos de la gestante, examen materno fetal, ecografía obstétrica, gráficas de controles maternos, gráficas de control del feto y datos del parto. Es un sistema privativo y además es una aplicación de escritorio lo que hace más complicado el despliegue de dicho sistema en una institución hospitalaria así como la comunicación con sistemas de terceros. [7]

CTN PACIENTES HIS

Sistema de información integrado, totalmente modular, escalable y flexible. Está diseñado para integrar el ciclo de atención del paciente frente a la prestación de servicios médicos, terapéuticos y diagnósticos. Tiene como eje central para efectuar sus consultas, una Historia Clínica única, dinámica, digital, integrada y adaptable a todas las especialidades médicas y ambientes de atención como hospitalización, donde permite tener un registro de diagnósticos de salida, estado a la salida (defunciones) y los nacimientos.

Es una aplicación de escritorio que además brinda la funcionalidad de definir el horario de los médicos para la asignación de citas, logrando la optimización del tiempo, y ofreciendo a sus pacientes una mejor oportunidad del servicio.

El sistema controla la gestión de los pacientes en el servicio de Gineco-obstetricia en el área de Hospitalización, sin embargo no gestiona los antecedentes obstétricos de las pacientes y no gestiona la información que se genera en el quirófano, información importante sobre el parto de las pacientes que resulta vital en el tratamiento postoperatorio y en las salas de postparto. [8]

El estudio de los diferentes sistemas en el mundo vinculados al campo de acción demuestra que la gestión de los procesos gineco-obstétricos no cumple con todos los requisitos, debido a que no son desarrollados con tecnologías libres, algunos de estos son aplicaciones de escritorio que no son multiplataforma y sobre todo que no se pueden integrar con el Sistema de Gestión Hospitalaria alas HIS. Se llega a la conclusión de que es necesario desarrollar una solución que sea fácil de integrar,

con una arquitectura robusta y flexible, que asegure la calidad del producto final y cumpla con la arquitectura definida por el Departamento de Sistemas de Gestión Hospitalaria.

1.3. Tendencias y tecnologías actuales a considerar

El desempeño de un proyecto está antecedido por la investigación de las tecnologías de punta que se utilizan, así como las ventajas y desventajas que trae su aplicación. Teniendo en cuenta las características del entorno donde se desplegará el sistema a desarrollar y los antecedentes encontrados, se han definido un conjunto de tecnologías a utilizar, permitiéndole al mismo, que esté libre de costos adicionales relativos a pago de licencias de software, además de que sea multiplataforma.

A continuación se mencionan las definiciones de las tecnologías seleccionadas, así como sus ventajas:

Arquitectura en tres capas

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica lo que provee una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, la responsabilidad indica la funcionalidad que está siendo desarrollada. [9]

Esta describe el bajo acoplamiento que debe existir entre los componentes de manera tal que la mayoría de la interacción ocurra solamente en capas vecinas. Las capas que conforman una aplicación pueden residir en la misma máquina física o pueden estar distribuidas sobre diferentes computadoras. Los componentes en las capas se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.

Como su propio nombre lo indica se pueden identificar tres elementos esenciales: la Capa de presentación, que reúne los diferentes aspectos que tienen que ver con las interfaces, la interacción entre los diferentes tipos de usuarios y el manejo y apariencia del entorno visual de la aplicación; la Capa de negocio, que posee todos los aspectos del software que automatizan los procesos de negocio que son ejecutados por los usuarios y la Capa de datos, que es la encargada de los aspectos de software que tienen que ver con los datos persistentes.

Patrón de arquitectura Modelo-Vista-Controlador (MVC)

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control, en tres componentes distintos. El estilo de llamada y retorno MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML (HyperText Markup Language) y el código que provee de datos dinámicos a la página. El modelo lo constituyen los datos que gestiona la aplicación y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción del patrón

Modelo: Se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema de modelo.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca peticiones al modelo y, probablemente, a la vista.

La unión entre la capa de presentación y capa de negocio representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos. [10]

1.4. Tecnologías y herramientas utilizadas en el proceso de desarrollo

La arquitectura en capas y el patrón MVC, pueden relacionarse lógicamente mediante cada uno de sus elementos, donde la Capa de presentación podría corresponderse con la Vista, la Capa de negocio con el Controlador y la Capa de datos con el Modelo. A continuación se realiza una descripción de las tecnologías utilizadas en el proceso de desarrollo, agrupadas por cada uno de los elementos del patrón anteriormente descrito.

1.4.1. Vista

Lenguaje de Marcado de Hipertexto Extensible (XHTML)

Acrónimo inglés de eXtensible Hypertext Markup Language (Lenguaje Extensible de Marcado de Hipertexto). Es la versión XML (eXtensible Markup Language) de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium (W3C) de lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas. [11]

JSF

Java Server Faces (JSF) es un framework basado en el patrón MVC. Ofrece una clara separación entre el comportamiento y la presentación. Une los componentes de la interfaz de usuario (UI, por sus siglas en inglés) con los conceptos de la capa-web sin limitarse a una tecnología de script o lenguaje de marcas particular. [12]

JSF brinda un conjunto de componentes de interfaz de usuario del lado del servidor para aplicaciones Web basadas en Java. Permite el manejo de estados y eventos, así como la asociación entre datos de la interfaz y los datos de la aplicación Web.

Ajax

Son las siglas de Asynchronous JavaScript y XML. No es un lenguaje de programación sino un conjunto de tecnologías que permiten hacer las páginas Web más interactivas. Entre estas tecnologías podemos mencionar XHTML y CSS (Cascading Style Sheets) para crear una presentación basada en estándares; DOM (Document Object Model) para la interacción y manipulación dinámica de la presentación; XML, XSLT (eXtensible Style Sheet Language Transformation) y JSON (JavaScript Object Notation) para el intercambio y manipulación de la información; XMLHttpRequest para el intercambio asíncrono de información y JavaScript como lenguaje interpretado del lado del cliente. [13]

RichFaces

RichFaces es una librería de componentes visuales para JSF. Posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

RichFaces se integra perfectamente en el ciclo de vida de JSF, incluye funcionalidades Ajax, provee varias librerías de componentes como son Core Ajax y UI, así como administración avanzada de recursos como imágenes, código JavaScript y Hojas de Estilo en Cascada (CSS). Es posible crear interfaces de usuario de manera rápida y eficiente, basado en componentes que están listos para usar y son altamente configurables. RichFaces, además, es un proyecto que fue desarrollado con una arquitectura abierta para que fuera compatible con la mayor cantidad de entornos. [14]

Ajax4JSF

Es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones automáticas al servidor, control de cualquier evento de usuario, etc. Esta librería permite dotar a la aplicación JSF de contenido mucho más profesional con muy poco esfuerzo. [15]

Extended EL

Extended EL provee una extensión al estándar Lenguaje de Expresión Unificado (EL) aumentando su expresividad y su poder. Este estándar adopta las características ofrecidas por el lenguaje de expresión de JSF. Mediante su utilización se puede reducir drásticamente la cantidad de código en las páginas creadas, lo que aumenta la productividad y hace más fácil el mantenimiento y más pequeña la curva de aprendizaje en términos del desarrollo.

Seam UI

Son una serie de controles JSF que adicionan varias mejoras desde validación, expresiones Extended EL e integración de la navegación en la interfaz de usuario basada en flujo de páginas o procesos del negocio. [16]

Facelets

Es un framework para plantillas centrado en la tecnología JSF, que permite el uso de plantillas en aplicaciones. Las principales ventajas de Facelets son la construcción de interfaces basadas en plantillas, una rápida creación de componentes por composición y fácil creación de funciones y librerías de componentes. [17]

1.4.2. Controlador

JBoss Seam

Es una poderosa plataforma de desarrollo de código abierto para la construcción de aplicaciones empresariales en Java. Integra tecnologías como Ajax, JSF, Java Persistence API (JPA), Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM).

Fue diseñado para posibilitar a los desarrolladores de aplicaciones empresariales ensamblar complejas aplicaciones usando simples clases Java, un rico conjunto de componentes de interfaces de usuario y muy poca configuración mediante archivos XML. Brinda un soporte único para las conversaciones, entiéndase por conversación en esta plataforma algo muy parecido a lo que hacen las personas donde existe un lugar, un tiempo y una serie de objetos implicados y la administración de estado declarativa puede introducir una experiencia de usuario más. [18]

Drools

Es un Sistema de Administración de Reglas de Negocio (BRMS) con un motor de reglas basado en una adaptación orientada a objetos del algoritmo Rete. Permite expresar de una forma más natural las reglas de negocio interactuando con los objetos de negocio. Provee separación de lógica del negocio y los datos.

También provee soporte para la programación declarativa y es lo suficientemente flexible para expresar la semántica del problema con un Lenguaje Específico de Dominio (DSL, por sus siglas en inglés). Cuenta con la implementación completa de la JSR-94 Rule Engine API. [19]

1.4.3. Modelo

Java Persistence API (JPA)

Es una API de persistencia de POJOs (Plain Old Java Object), es decir, objetos simples que no heredan ni implementan otras clases (como los EJBs). En su definición, ha combinado ideas y conceptos de los principales frameworks de persistencia, como Hibernate, Toplink y JDO (Java Data Object) y de las versiones anteriores de EJB. Es una especificación de Oracle para la persistencia de objetos Java a cualquier base de datos relacional.

Proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos. Esta API fue desarrollada para la plataforma JEE. [20]

Enterprise JavaBeans (EJB3)

Una de las metas de la arquitectura EJB es la de poder escribir de manera fácil aplicaciones de negocio orientadas a objetos y distribuidas, basadas en el lenguaje de programación Java. El propósito de EJB3 es el de proveer el soporte de la arquitectura de EJB y al mismo tiempo reducir la complejidad para el desarrollo de aplicaciones empresariales. [21]

Hibernate

Es una herramienta de Mapeo Objeto Relacional (ORM, por sus siglas en inglés) y un generador de sentencias SQL (Structured Query Language). Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. [22]

1.5. Tecnologías y metodologías horizontales

PostgreSQL

Es un Sistema de Gestión de Base de Datos (DBMS) objeto relacional de código abierto, distribuido bajo la licencia BSD (Berkeley Software Distribution). Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL es desarrollado y mantenido por una amplia comunidad. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Es el DBMS de código abierto más potente del mercado.

Utiliza el lenguaje SQL para la construcción de las consultas. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos, donde cada conexión tiene su propio proceso desde el punto de vista del servidor, para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

PostgreSQL es un sistema objeto relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. [23]

Java

Es uno de los lenguajes de Programación Orientado a Objetos (POO) más completos y utilizados para el desarrollo de software. Permite a los desarrolladores desarrollar en un sistema operativo y ejecutarlo en prácticamente cualquier otro, crear programas para que funcionen en un navegador Web y mediante servicios Web, combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados, así como desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo costo y prácticamente cualquier tipo de dispositivo digital. [24]

JBoss Application Server (JBoss AS)

Es un servidor de aplicaciones Java 2 Platform, Enterprise Edition (J2EE) multiplataforma de código abierto desarrollado en Java. Implementa todo el paquete de servicios de J2EE y es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 5.0. Combina una

Arquitectura Orientada a Servicios con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. [25]

Proceso Unificado de Desarrollo (RUP)

Es el resultado de varios años de trabajo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML (Unified Modeling Language), y trabajo de muchas metodologías utilizadas por los clientes. En RUP se han agrupado las actividades en grupos lógicos en los que se definen nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. Dentro de estos seis dedicados a la ingeniería, los de Implementación y Pruebas son los empleados para el desarrollo de este trabajo investigativo. El ciclo de vida de RUP se caracteriza por ser dirigido por caso de uso, centrado en la arquitectura, iterativo e incremental. [26]

Lenguaje Unificado de Modelado (UML)

Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a objetos. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso utilizar.

Este lenguaje de modelado formal permite tener un mayor rigor en la especificación, realizar una verificación y validación del modelo desarrollado, automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto último permite que el modelo y el código estén actualizados. [27]

Notación para el Modelado de Procesos de Negocio (BPMN)

El objetivo principal de esta notación es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua. A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos.

La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de ellos. La administración de los procesos permite asegurar que los mismos se ejecuten eficientemente, y la obtención de información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que puede identificar posibles ineficiencias en los mismos, y actuar para optimizarlos. [28]

1.6. Herramientas

Con los elementos expuestos anteriormente se pueden definir las herramientas que conforman el Ambiente de Desarrollo de los procesos gineco-obstétricos para el Sistema de Información Hospitalaria alas HIS.

Se utilizará el Visual Paradigm 6.0 como herramienta CASE (Computer Aided Software Engineering), esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Se utiliza la notación BPMN (Business Process Modeling Notation) para generar la documentación del negocio y el lenguaje UML 2.0 para crear los diagramas de casos de uso del flujo de requerimientos, así como los correspondientes a los de los flujos de diseño e implementación. También es utilizada para administrar la base de datos pgAdmin III, debido a que es una herramienta, relacional, muy poderoso y puede ser ejecutado sobre la mayoría de los sistemas operativos que existen hoy en día.

Finalmente se selecciona el Eclipse, un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) que cuenta con la posibilidad de añadir nuevas funcionalidades al editor, a través de nuevos módulos, lo que es beneficioso para la utilización de JBoss Tools, conjunto de herramientas que posibilitan el desarrollo desde el IDE Eclipse con RichFaces, Seam e Hibernate, además de realizar la administración y configuración del servidor JBoss AS.

En este capítulo se realizó un estudio sobre los principales sistemas existentes que gestionan la información de los procesos gineco-obstétricos en las instituciones hospitalarias y se determinó que no cumplen con todos los requisitos necesarios.

Por lo que se evidenció la necesidad de desarrollar un sistema en el que el usuario tenga una mayor accesibilidad, sea adaptable a diferentes instituciones hospitalarias, que esté debidamente documentado para desarrollos futuros del mismo y se encuentre desarrollado con tecnologías y herramientas no privativas. Se justificó la aplicación de los patrones de diseño y el empleo de las tecnologías y herramientas propuestas para la obtención del sistema.

CAPÍTULO 2. Descripción de la arquitectura

El software tradicional, como único proceso ejecutándose en un único ordenador, no supone arquitecturas complejas ni grandes riesgos. Sin embargo, los sistemas actuales que permiten ejecución remota, sistemas distribuidos, nuevos y diversos lenguajes de programación, entornos de alojamiento y otras dependencias externas convierten a la arquitectura de un sistema en su producto técnico más crítico.

De acuerdo con la utilización del patrón de arquitectura MVC, el modelo de diseño tiene una estructura lógica, donde las páginas clientes y formularios se crean a partir de código XHTML que construyen las páginas servidoras, cuyas clases controlan las vistas. Estas invocan a las clases controladoras las cuales se encargan de ejecutar la lógica del negocio y modifican las clases entidades.

2.1. Requerimientos no funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

Muchos requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo. Esto significa que a menudo son más críticos que los requerimientos funcionales particulares. Mientras que el incumplimiento de este último degradará el sistema, una falla en un requerimiento no funcional del sistema lo inutiliza. [29]

RNF Usabilidad

El diseño del sistema está dirigido a potenciar la adquisición de habilidades por parte de los usuarios en un tiempo reducido:

Para el nivel Elemental asociado al dominio del sistema serán necesarios 20 días de preparación, obteniendo la categoría de Usuarios normales.

Para el nivel Avanzado, asociado al dominio del sistema, serán necesarios 30 días de preparación, obteniendo la categoría de Usuarios avanzados.

RNF Soporte

Seguridad de acceso a administración de usuarios:

- Se mantendrá seguridad y control entre usuarios, garantizando el acceso solo a los niveles establecidos de acuerdo con la función que realizan. Las contraseñas pueden ser cambiadas por los propios usuarios o por los administradores del sistema.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- La información que haya sido agregada al sistema no será eliminada físicamente de la base de datos.
- El sistema permitirá la recuperación de la información de la base de datos a partir de respaldos o salvadas realizadas.

Monitoreo de funcionamiento:

Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación.

Respaldo y recuperación de base de datos:

Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados.

Auditoría:

Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema, para esto debe existir un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso como mínimo: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información contenida en el registro modificado.

Configuración de parámetros:

Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

RNF Restricciones de diseño

La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo Web se manejará de forma declarativa. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

RNF Interfaz

Interfaces de usuario:

- Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La entrada de datos incorrecta será detectada claramente e informada al usuario.
- El sistema debe permitir la internacionalización de toda la información mostrada en las interfaces visuales.

Interfaces de comunicación:

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven).

RNF Requerimientos de Rendimiento

- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
- El sistema respetará las buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

RNF Requerimientos de hardware

Estaciones de trabajo:

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo, que cuente con un navegador actualizado y que siga los estándares Web, se recomienda IE 7, Firefox 3 o versiones superiores. Por lo anterior se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.

Servidores:

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

RNF Requerimientos de software

El sistema puede ser desplegado en sistemas operativos Windows, Unix y Linux, utilizando la plataforma Java (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador Web, estos pueden ser IE 7, Opera 9, Google Chrome 11 y Firefox 3 o versiones superiores de estos.

2.2. Descripción de la arquitectura

La arquitectura de software es un conjunto de patrones que sirven de guía para la elaboración de un sistema, ofrece además la descripción del software en subsistemas y componentes, dejando bien claro la forma en que estos interactúan.

Es de suma importancia a la hora de tomar decisiones y al mismo tiempo, proporciona un lenguaje común que permite una mejor comunicación entre los equipos que participen en un proyecto.

Con el objetivo de lograr una mejor organización de los componentes que conforman los procesos a implementar y siguiendo las características del patrón MVC, se agruparon dichos componentes en tres capas fundamentales: presentación, negocio y acceso a datos. Esta estructura garantiza que el cambio o sustitución de un elemento correspondiente a una de las capas no altere el funcionamiento de otro que lo utilice y sea parte de alguna de las capas restantes.

La Capa de presentación está formada por páginas XHTML, las cuales están compuestas por formularios y controles JSF y RichFaces, estos se encargan de validar y mostrar los datos haciendo uso de componentes de interfaz gráfica de usuario.

La Capa de negocio está integrada por clases controladoras, que encierran la lógica del negocio del módulo. A las cuales, mediante anotaciones que provee el framework Seam, se les especifica el contexto en que se encuentran, los cuales definen el estado de los datos y las entidades que manejan.

Para el acceso a datos se usa la implementación de JPA de Hibernate, minimizando por un lado las configuraciones en XML sin chequeo de tipos. Por otro lado utiliza los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam.

Se elimina gran parte del código en cuanto a transacciones, la transmisión del contexto de persistencia y se pueden establecer validaciones mediante los Hibernate Validators.

La comunicación entre los elementos de las capas de presentación, negocio y acceso a datos está regida por el framework Seam. Este permite, mediante anotaciones, que las páginas de la interfaz de usuario referencien las funcionalidades definidas en las clases controladoras, y que estas puedan usar los componentes de acceso a datos y otros de la capa del negocio.

2.3. Análisis de componentes reutilizables

La reutilización de código se refiere al comportamiento y las técnicas que garantizan que una parte o la totalidad de un programa informático existente se puedan emplear en la construcción de otro, lo mismo se puede aplicar a funcionalidades de un mismo programa que podrían ser reutilizadas. De esta forma

se aprovecha el trabajo anterior, se economiza tiempo y se reduce la redundancia. La manera más común fuera copiar el código total o parcialmente, pero sería muy trabajoso mantener tantas copias del mismo en código, por lo que es recomendable dejar el código reutilizable en un mismo lugar y llamarlo desde donde se necesite.

En el proceso de desarrollo que se lleva a cabo, se cuenta con la ventaja de que existan algunos componentes que se utilizan de manera general por todos los módulos del Sistema de Información Hospitalaria alas HIS, como pueden ser: la clase ActiveModule que brinda información sobre qué módulo y en qué entidad se encuentra el usuario que está utilizando el sistema; la clase User que ofrece los datos del usuario que se encuentra autenticado y los componentes principales del Módulo de Hospitalización.

Los procesos gineco-obstétricos abarcan diferentes áreas dentro de las instituciones hospitalarias y dentro del Sistema de Gestión Hospitalaria alas HIS reutiliza diferentes componentes de otros módulos con los que interactúa. Dentro del Módulo de Laboratorio se utiliza la clase CrearSolicitudExamenLaboratorioHospControlador; del Módulo Banco de Sangre CrearSolicitudTransfusionHospControlador, CrearSolicitudExamenTipiajeHospControlador y RegistrarDatosTransfusionControlador; en Bloque Quirúrgico CrearOrdenMedicaHospControlador; de Citas AsignarCitaEstudioHospControlador y otros componentes comunes como son las clases CrearOrdenMedicaHospControlador y CrearSolicitudInterconsultaHospControlador.

2.4. Seguridad

Para fomentar la seguridad en el sistema se propone un control de acceso a nivel de usuarios y contraseñas, así como una diferenciación de los mismos atendiendo al rol que desempeñan en el sistema. Dichas contraseñas solo podrán ser cambiadas por los usuarios o por los administradores del sistema. Las actividades de los usuarios dentro del sistema serán registradas mediante trazas.

Para garantizar la fidelidad de los datos la información que sea intercambiada entre el sistema y otros sistemas debe ser cifrada, asegurando que no se lea ni modifique la información confidencial que se maneja. Además, toda la información ingresada al sistema nunca será eliminada físicamente de la base de datos, permitiendo al sistema recuperarse de posibles errores en su manejo.

2.5. Vista de Despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema y muestra las relaciones físicas de los distintos nodos que lo componen y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. [30]

El despliegue del sistema en cuestión contará con dos servidores, uno de aplicaciones y otro de Base de Datos, conectados entre sí mediante el Protocolo de Control de Transmisión/ Protocolo de Internet (TCP/IP por sus siglas en inglés). Además, contará con estaciones clientes que se conectarán con el servidor de aplicaciones mediante HyperText Transfer Protocol (HTTP, por sus siglas en inglés).

A continuación se muestra el Diagrama de Despliegue y las características de cada uno de los nodos que requiere el sistema.

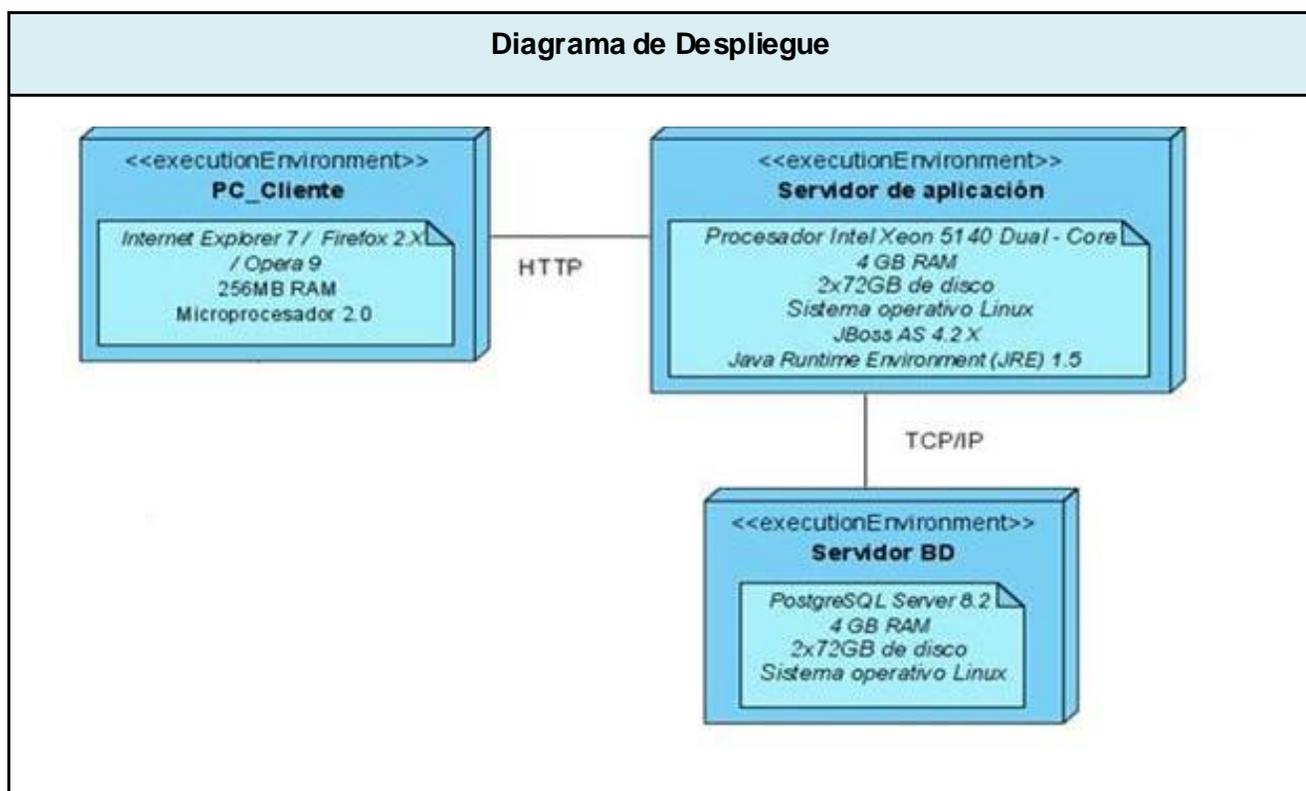


Figura 2.5.1 Diagrama de Despliegue

2.6. Estrategias de codificación. Estándares y estilos a utilizar

Las estrategias de codificación son muy importantes en la implementación de cualquier software, en especial para los involucrados en el mismo. Los software pueden estar la mayor parte de su vida útil en mantenimiento y en la mayoría de los casos dicho mantenimiento es llevado a cabo por el desarrollador, por lo que es importante el empleo de estándares que hagan el código más legible y más fácil de entender por otros programadores.

Como estrategia de codificación del sistema se utiliza el estándar de codificación del lenguaje de programación Java (Java Code Conventions), definido por el propio lenguaje y en español. Para el nombre de las clases se utilizará la notación o estilo Pascal Casing, con la que los nombres quedan compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula y terminando todo el nombre con “_” y el nombre del módulo al que pertenece. Los identificadores de atributos y métodos de la clase se escribirán en notación Camel Casing: nombres compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. Las clases controladoras terminarán con el nombre del módulo al que pertenecen y la palabra “controlador” al final.

Identación

- **Inicio y fin de bloque.**

Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.

- **Aspectos generales.**

El identado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla. Los inicios ({) y cierre (}) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.

Comentarios, separadores, líneas, espacios en blanco y márgenes

- **Líneas en blanco.**

Se emplean antes y después de métodos, clases y estructuras.

Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

- **Espacios en blanco.**

Entre operadores lógicos y aritméticos.

Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: hoja = hoja_hospitalizacion

- **Aspectos generales.**

Sobre el comentario

Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.

Bases de Datos, Tablas, esquemas y Campos

- **Apariencia de los campos**

Todas las letras en minúscula.

El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

- **Aspectos generales**

Sobre las BD, vistas, tablas atributos y procedimientos.

El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos.

En este capítulo se describió la concepción arquitectónica y la seguridad del sistema. Se describieron los requerimientos no funcionales que debe tener el sistema para lograr su objetivo, se elaboró el Diagrama de Despliegue del sistema con el objetivo de visualizar su estructura física. Se definió, además, la estrategia de codificación y los estándares y estilos a utilizar.

CAPÍTULO 3. Descripción y Análisis de la solución propuesta

En el presente capítulo se realiza el análisis y descripción de la solución propuesta, definiendo los diagramas de clases y de interacción que son entradas para el Modelo de Implementación, Modelo de Datos y Diagrama de Componentes.

3.1. Valoración crítica del diseño propuesto por el analista

La etapa de diseño es cuando se traducen los requerimientos funcionales y no funcionales en una representación de software. Es el primer paso en la fase de desarrollo de cualquier producto o sistema, su objetivo es producir un modelo o representación de una entidad que se va a construir posteriormente.

Para lograr un mejor entendimiento del modelo de diseño, se describen a continuación los términos mencionados en el mismo:

Clase de diseño: Es una abstracción de una clase o construcción en la implementación del sistema.

Diagramas de Clases de diseño: Son diagramas estáticos que describen la estructura de un sistema mostrando sus clases y donde se crea el diseño conceptual de la información que se manejará en el sistema.

Diagramas de Interacción: Describen aspectos dinámicos del sistema y muestran gráficamente cómo los objetos se comunican entre sí para cumplir con los requerimientos.

Los Diagramas de Interacción se clasifican en Diagramas de Secuencia y Colaboración. Generalmente los Diagramas de Colaboración se realizan en el análisis y los de secuencia en el diseño.

Diagramas de Secuencia: Es un diagrama de interacción que muestra las interacciones expresadas en función del tiempo, así como los objetos participantes y los mensajes que intercambian entre ellos a lo largo del tiempo.

Para elaborar el Modelo de Diseño se definió una estructura de paquetes que permite dividir el sistema en fragmentos manejables para su implementación. Se emplea además el criterio de empaquetamiento por procesos y por clases, siguiendo la estructura de procesos definidos en el

sistema. Los paquetes que hacen referencia a procesos se componen por subcarpetas que responden a las realizaciones de los casos de uso y los cuales contienen un diagrama de clases y los respectivos diagramas de secuencia. Los diagramas de clases se realizan utilizando estereotipos Web.

Se crea un paquete repositorio de clases que contiene tres subpaquetes, el de las *Vistas*, el de las *Sesiones* y el paquete de las *Entidades*.

El paquete de las *Vistas* contiene los contenidos Web referentes a las páginas clientes y los formularios que la componen.

En el de *Sesiones* se agrupan las clases *Controladoras Autogeneradas* por el entorno de desarrollo y las clases controladoras personalizadas.

El subpaquete de las *Entidades* contiene las *Entidades Autogeneradas* y las *Personalizadas*. Las *Entidades Autogeneradas* son obtenidas desde la base de datos utilizando el ORM Hibernate (permite generar entidades desde la base de datos). Las *Entidades Personalizadas* son las modificadas o entidades que heredan de las autogeneradas y cambiadas para lograr un mejor funcionamiento.

Estos paquetes y subpaquetes guardan relación entre sí, estableciendo las dependencias entre las distintas clases que contienen. A continuación se presenta el diagrama de paquetes del diseño que evidencia lo antes mencionado.

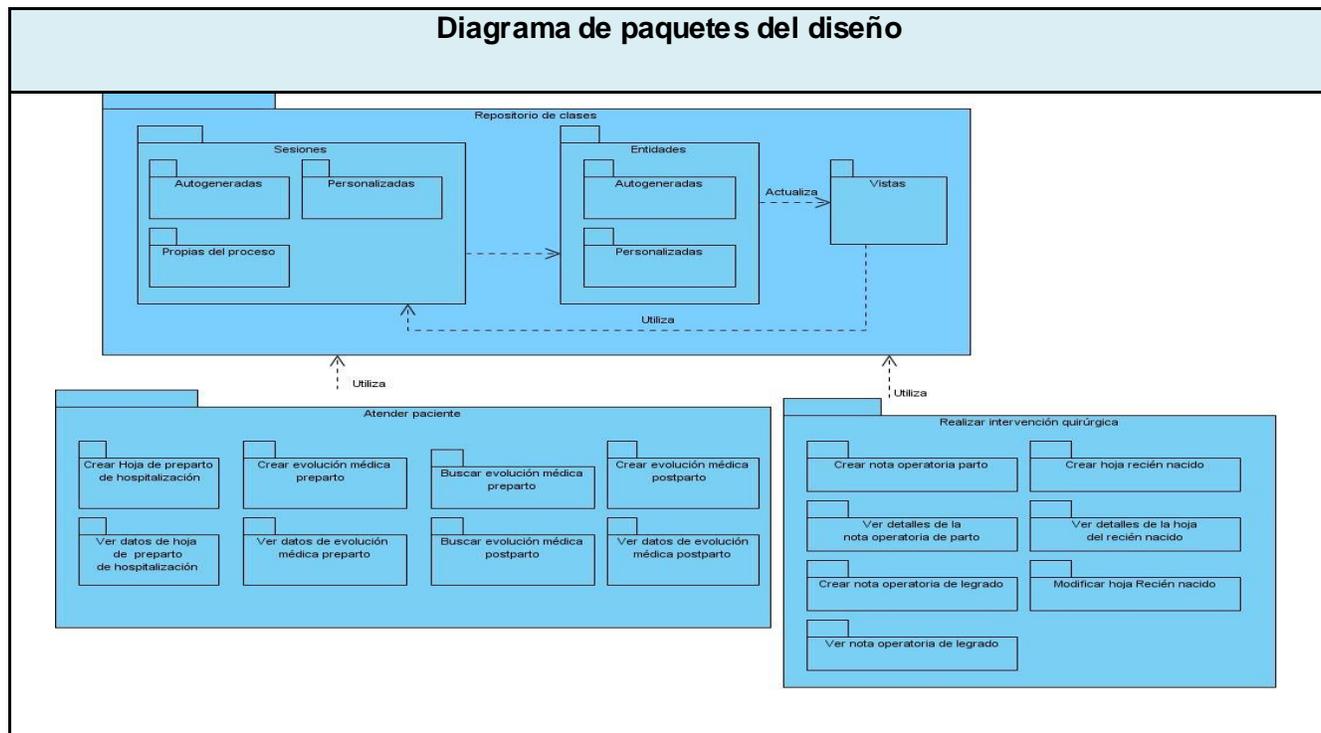


Figura 3.1.1 Diagrama de paquetes del diseño

A continuación se muestran los Diagramas de Clases y Diagramas de Secuencia, para algunos escenarios, de casos de uso del sistema relacionados con los procesos de *Atender paciente* y *Realizar intervención quirúrgica*.

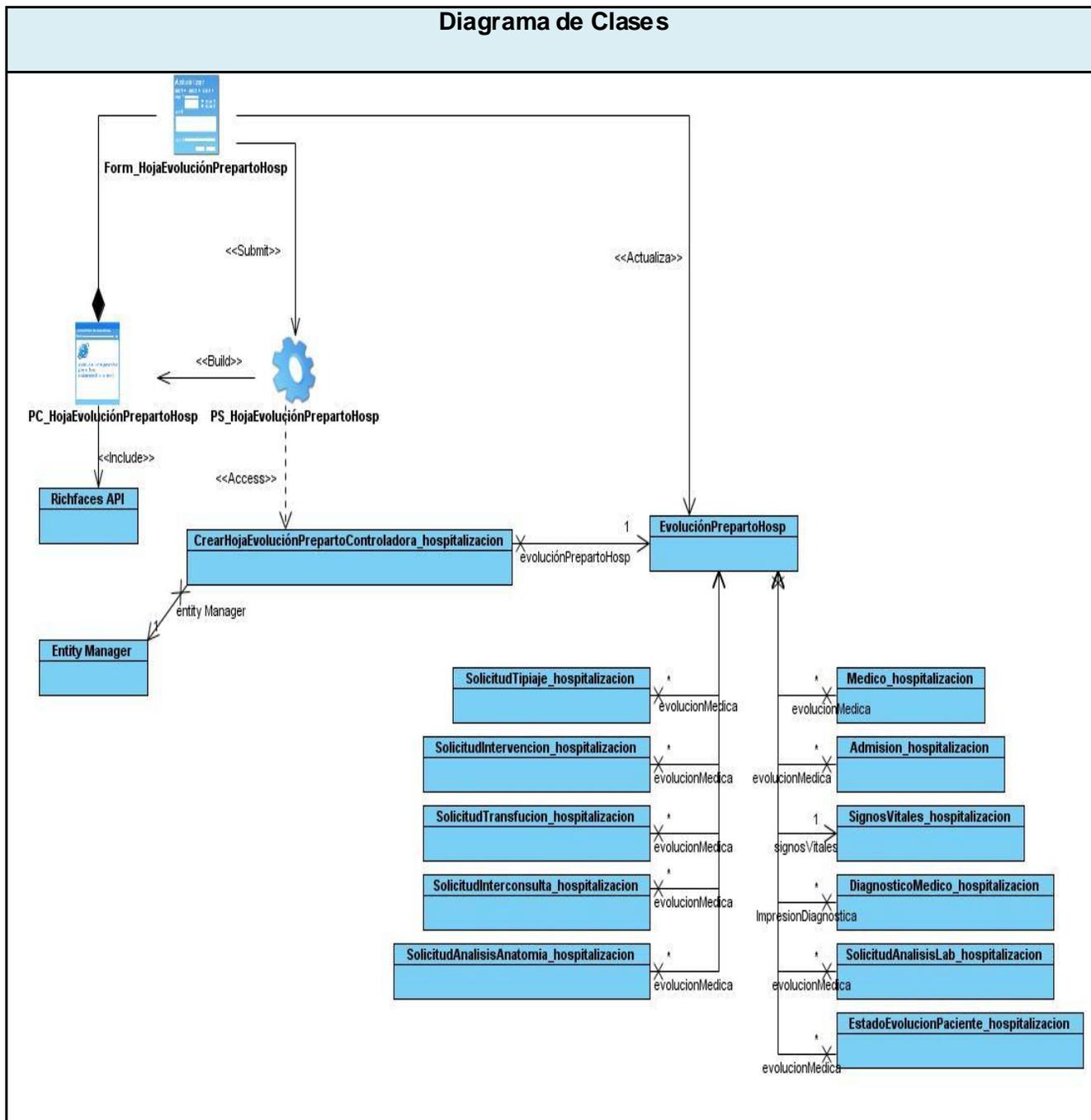


Figura 3.1.1 Diagrama de Clases del Diseño DCD_CrearEvolucionPrepartoHosp

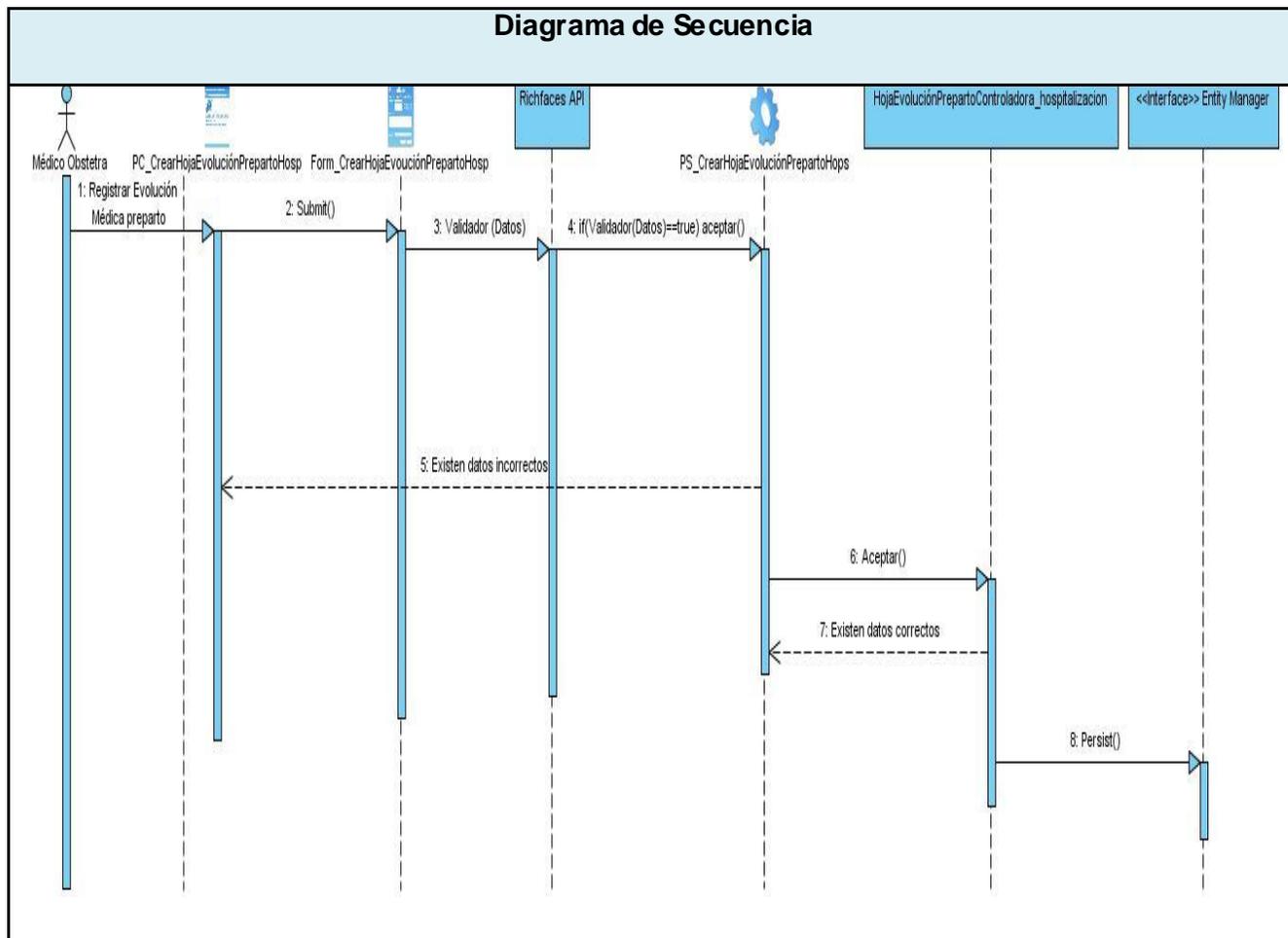


Figura 3.1.2 Diagrama de Secuencia DS_CrearEvolucionPartoHosp

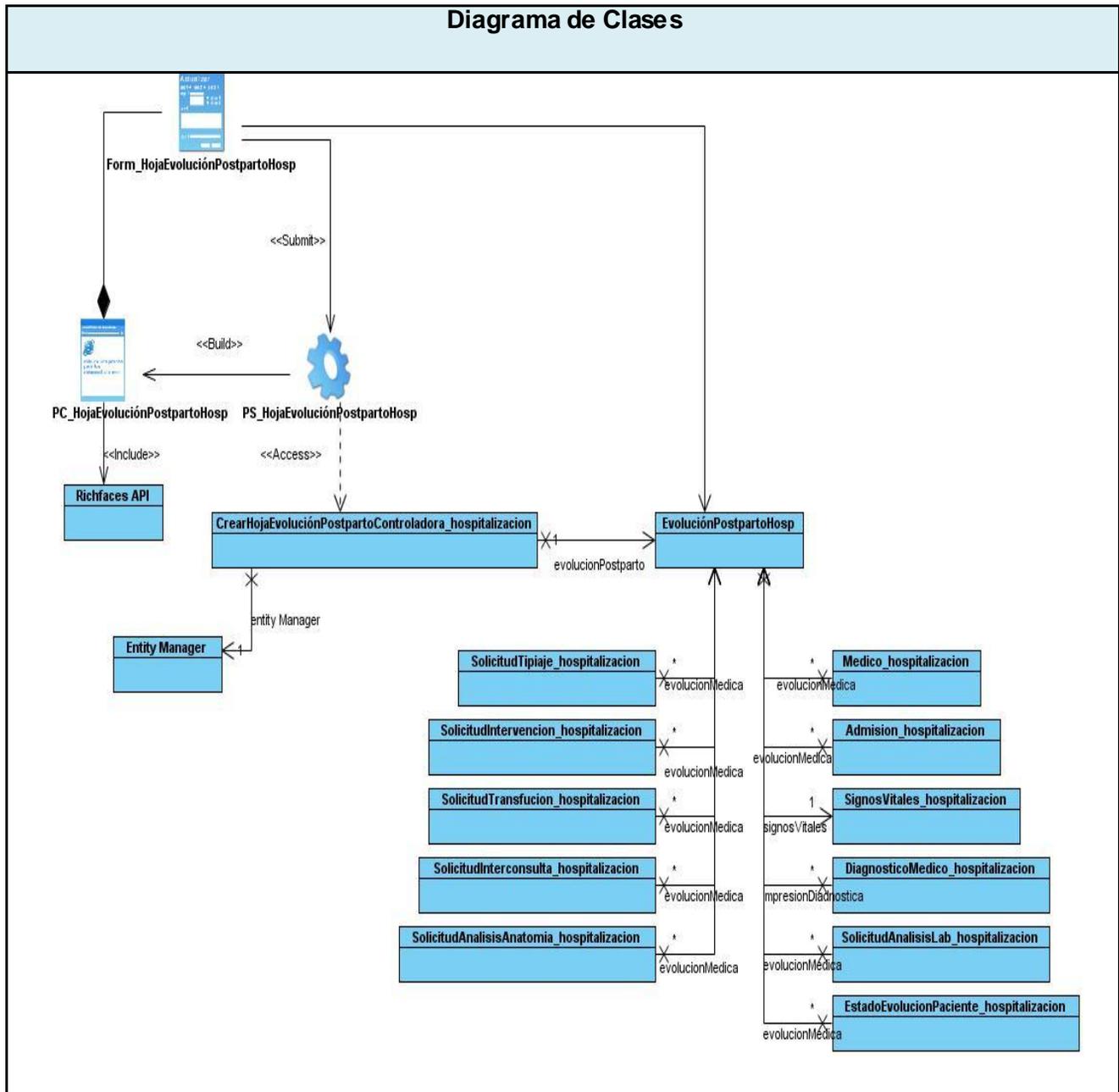


Figura 3.1.3 Diagrama de Clases del Diseño DCD_CrearEvolucionPostPartoHosp

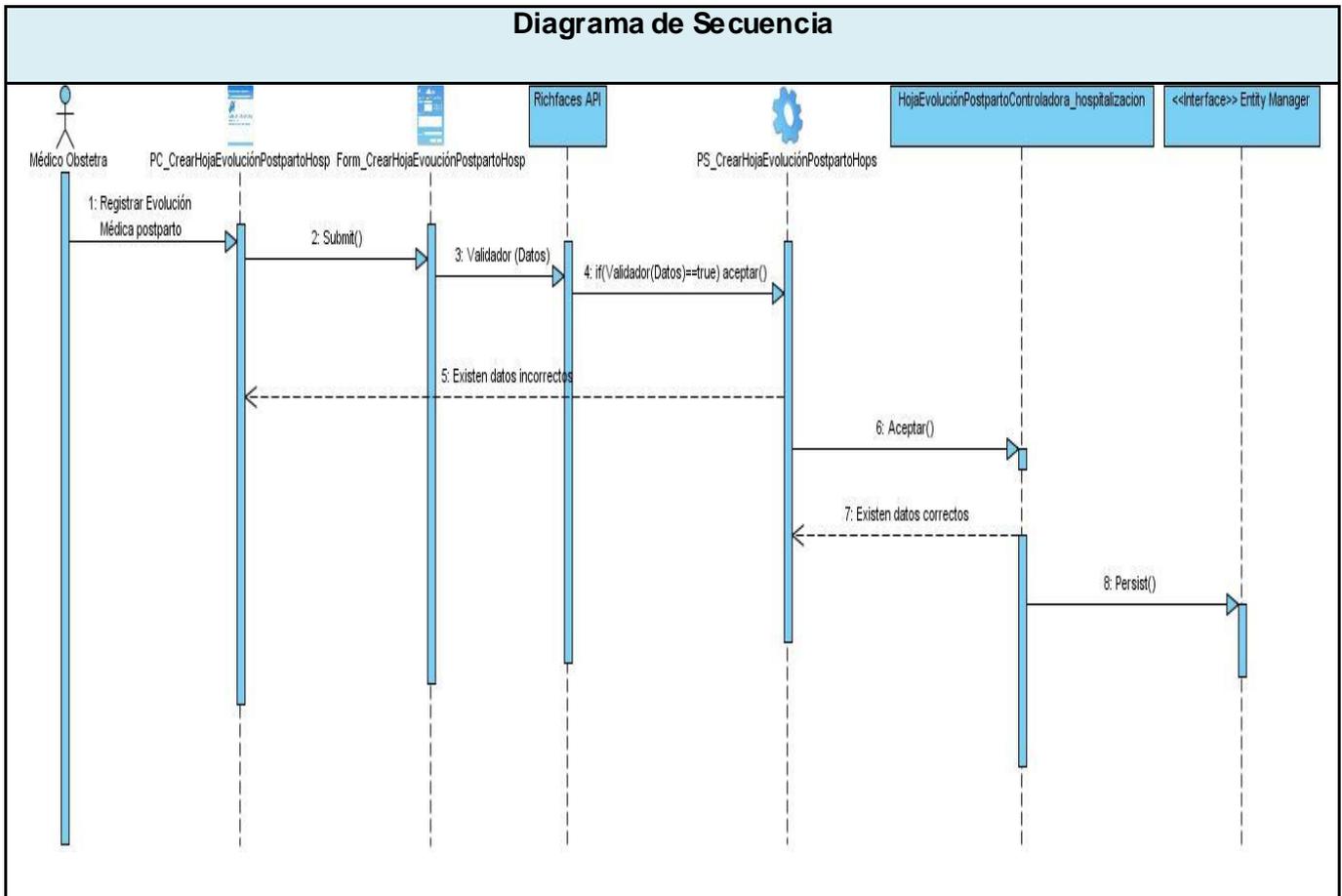


Figura 3.1.4 Diagrama de Secuencia DS_CrearEvolucionPostpartoHosp

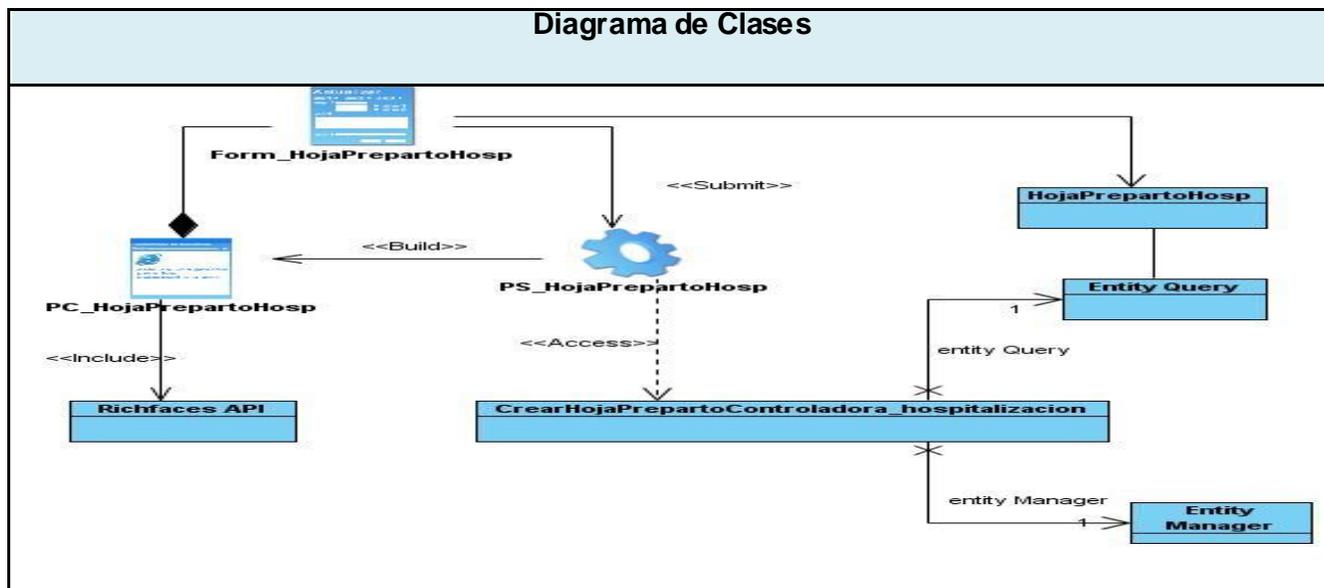


Figura 3.1.5 Diagrama de Clases del Diseño DCD_CrearHojaPrepartoHosp

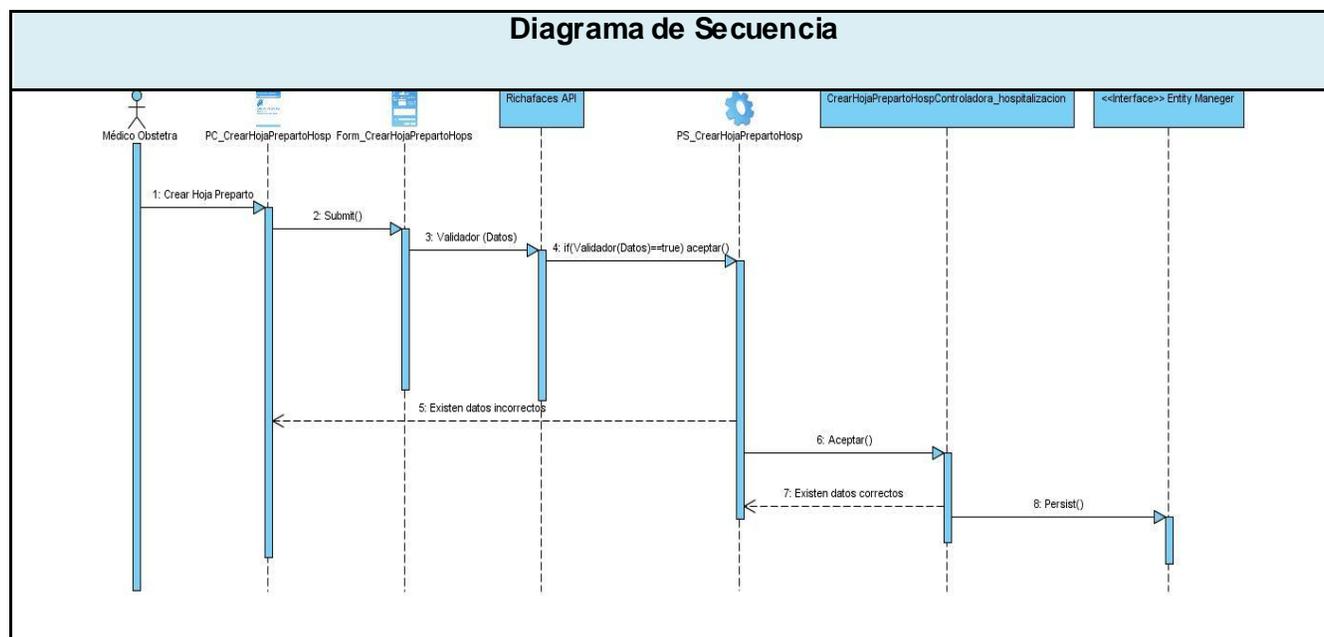


Figura 3.1.6 Diagrama de Secuencia DS_CrearHojaPrepartoHosp

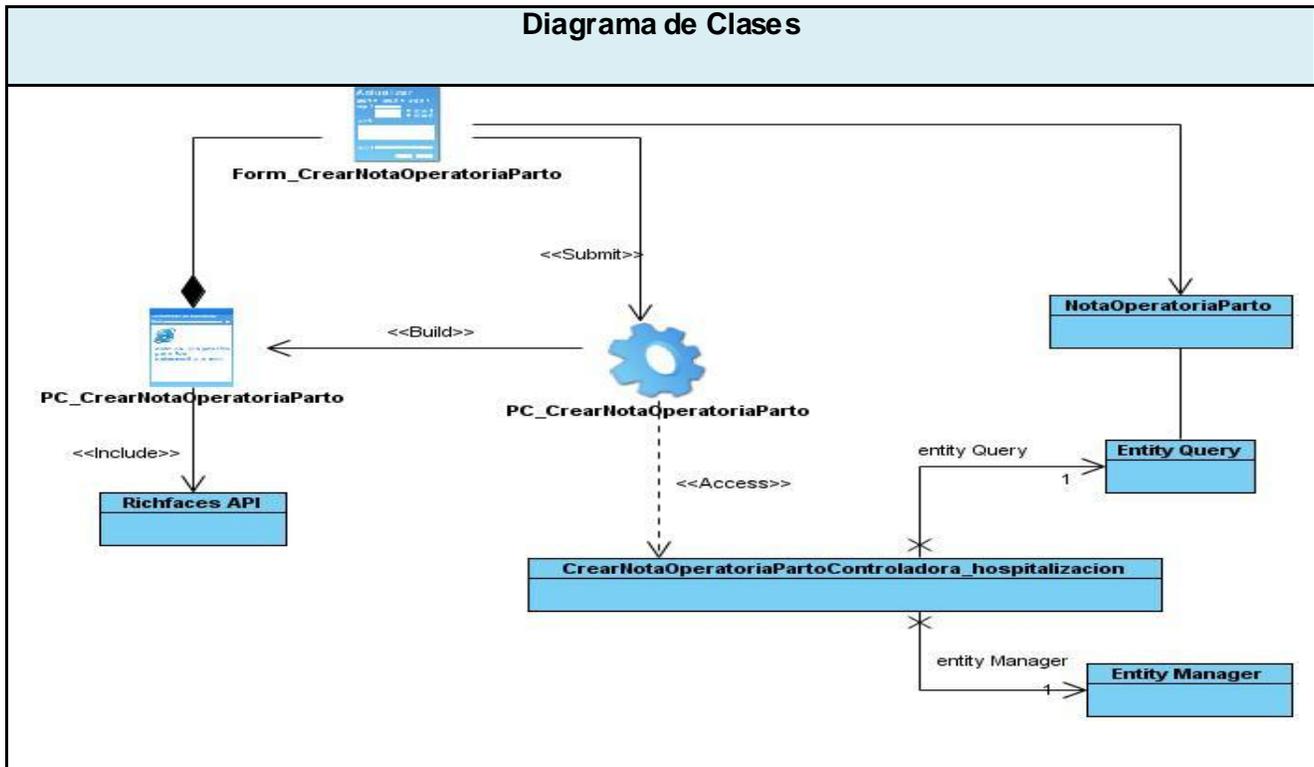


Figura 3.1.7 Diagrama de Clases del Diseño DCD_CrearNotaOperatoriaParto

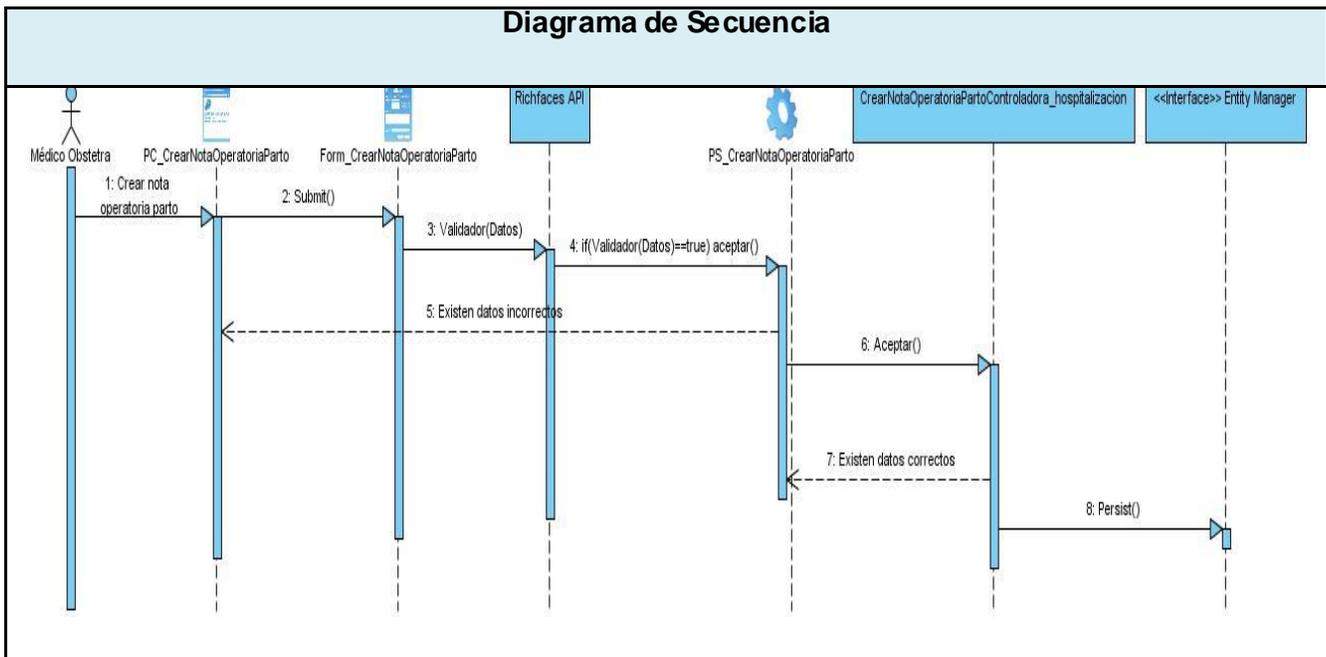


Figura 3.1.8 Diagrama de Secuencia DS_CrearNotaOperatoriaParto

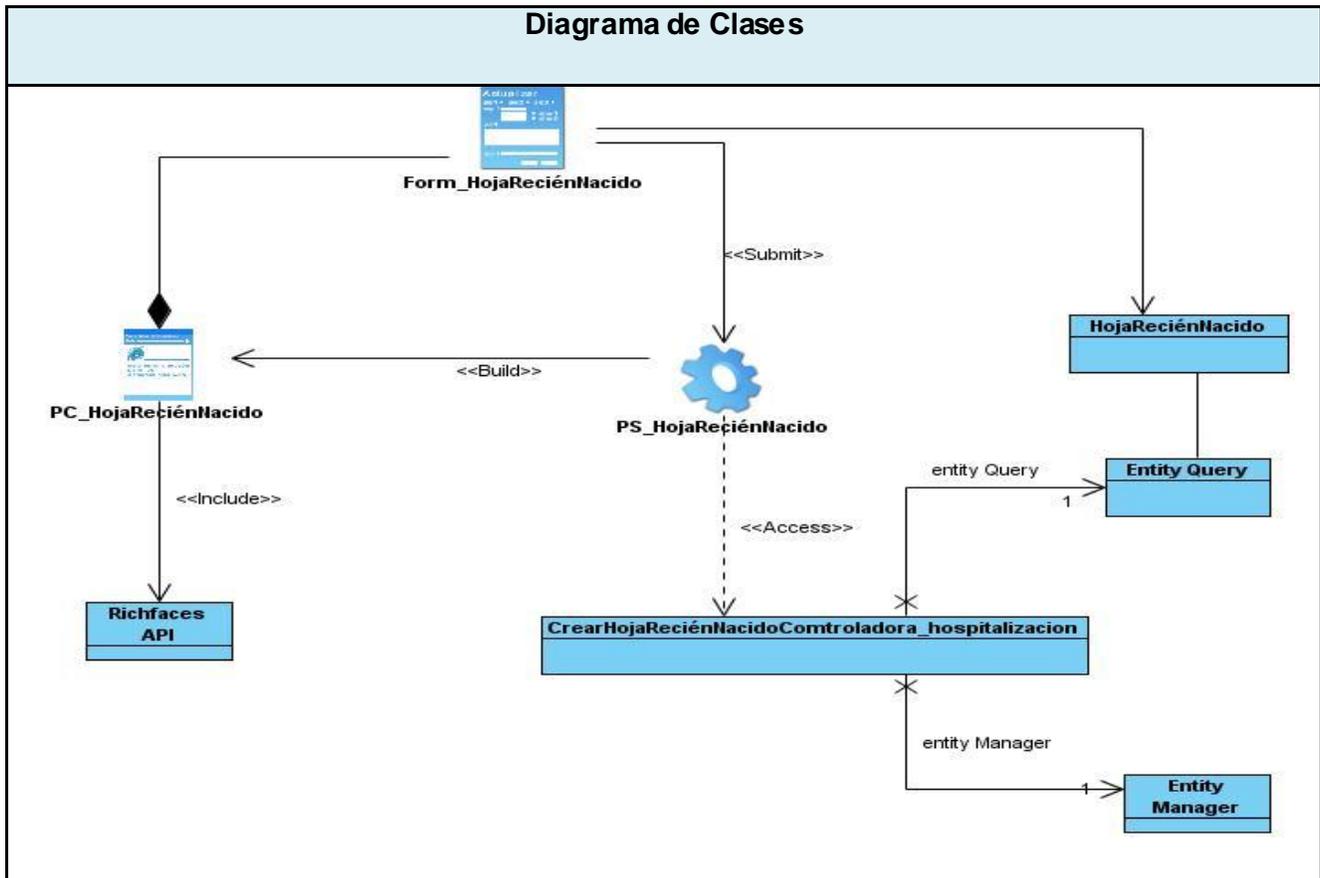


Figura 3.1.9 Diagrama de Clases del Diseño DCD_CrearHojaRecienNacido

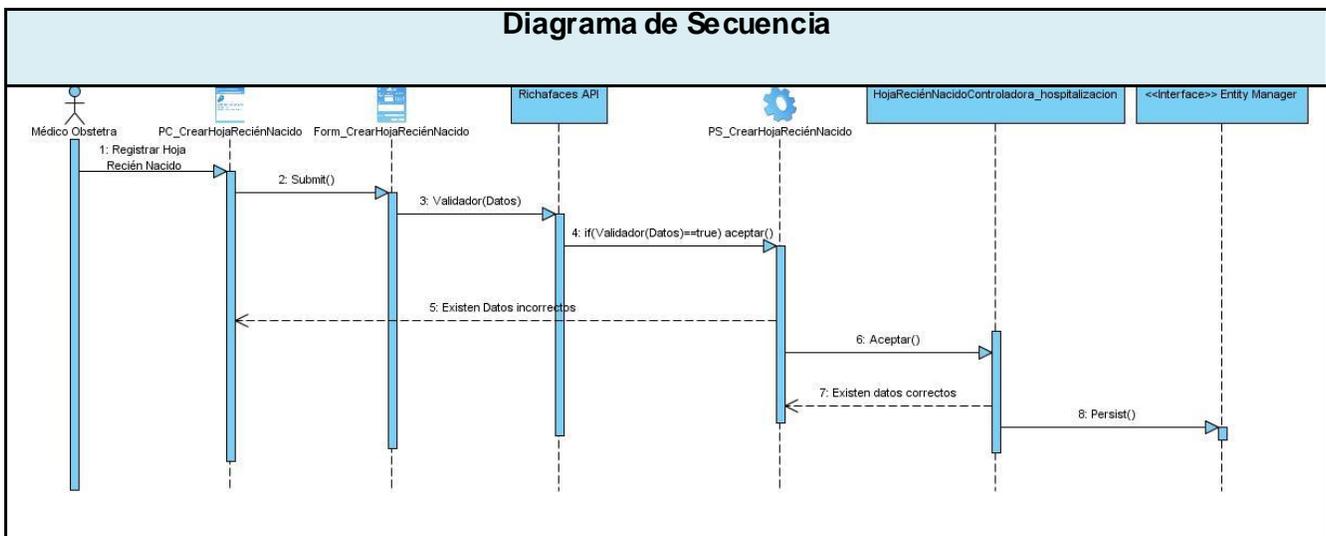


Figura 3.1.10 Diagrama de Secuencia DS_CrearHojaRecienNacido

3.2. Descripción de las nuevas clases u operaciones necesarias.

Nombre: CrearEvolucionMedicaPostPartoHospControlador	
Tipo de clase: Controladora	
Atributo	Tipo
evolucion	EvolucionMedica_hospitalizacion
evolucionMedica	EvolucionMedica
medico	Medico_hospitalizacion
admision	Admision_hospitalizacion
antPrenatObs	AntPrenatObst_hospitalizacion
periodoNeo	PeriodoNeonatal_hospitalizacion
alimentacion	Alimentacion_hospitalizacion
baseComplementoAlimenticio	BaseComplementoAlimenticio_hospitalizacion
desarrollo	Desarrollo_hospitalizacion
estadoEvolucion	EstadoEvolucionPaciente_hospitalizacion
evolucionMedicaPostparto_hospitalizacion	EvolucionMedicaPostparto_hospitalizacion
episiotAnormal	String
heridaQuirurgica	String
sangramiento	String
mucosas	String
masaje_uterino	String
coloracion	String
olor	String
estado_senos	String

caracteristicas_orine	String
forma_diuresis	String
lista_sangramiento	ArrayList<String>
lista_mucosas	ArrayList<String>
lista_coloracion	ArrayList<String>
lista_estado_senos	ArrayList<String>
lista_caracteristicas_orine	ArrayList<String>
lista_forma_diuresis	ArrayList<String>
lista_olor	ArrayList<String>
entityManager	EntityManager
facesMessages	FacesMessages
listarPacientes_hospitalizacion	ListarPacientesControlador
antPersonalesHospList_custom	AntPersonalesCustomList_hospitalizacion
antFamiliaresHospList_custom	AntFamiliaresCustomList_hospitalizacion
habitosPsicoHospList_custom	HabitosPsicoCustomList_hospitalizacion
inmunizacionesHospList_custom	InmunizacionesCustomList_hospitalizacion
crearSolicitudesHospitalizacion	CrearSolicitudesHospitalizacionControlador
modificarSolicitudesHospitalizacion	ModificarSolicitudesHospitalizacionControlador
cieHospList_custom	CieCustomList_hospitalizacion
consultaAccionesHosp	ConsultaAccionesHospControlador
antObtetricosHospList_custom	AntObtetricosCustomList_hospitalizacion
comprobarReglasnegocioHosp	ComprobarReglasNegocioHospControlador
crearDiagnosticoMedicoHosp	CrearDiagnosticoMedicoHospControlador

crearAntecedentesMenorEdadHosp	CrearAntecedentesMenorEdadHospControlador
generadorDocumentoCDAHosp	GeneradorDocumentoCDAHospControlador
hcManager	HCManager
notificationManager	NotificationManager
bitacora	IBitacora
Para cada responsabilidad:	
Nombre:	obtenerEstadosEvolPac
Descripción:	Permite obtener el estado del paciente al cual se le realiza la evolución médica
Nombre:	asignarCampos
Descripción:	Carga los datos de interés del paciente en el sistema necesarios para la evolución médica
Nombre:	verificarConversacionTerminada
Descripción:	Permite controlar el estado de la conversación activa para evitar pérdida de datos
Nombre:	save
Descripción:	Realiza la validación de todos los datos y si es satisfactoria persiste dichos elementos en la base de datos

Tabla 3.2.1 CrearEvolucionMedicaPostPartoHospControlador

Nombre: CrearEvolucionMedicaPrePartoHospControlador	
Tipo de clase: Controladora	
Atributo	Tipo
evolucion	EvolucionMedica_hospitalizacion
evolucionMedica	EvolucionMedica
Medico	Medico_hospitalizacion
admision	Admision_hospitalizacion

antPrenatObs	AntPrenatObst_hospitalizacion
periodoNeo	PeriodoNeonatal_hospitalizacion
alimentacion	Alimentacion_hospitalizacion
baseComplementoAlimenticio	BaseComplementoAlimenticio_hospitalizacion
desarrollo	Desarrollo_hospitalizacion
estadoEvolucion	EstadoEvolucionPaciente_hospitalizacion
evolucionMedicaPreparto_hospitalizacion	EvolucionMedicaPreparto_hospitalizacion
longitud_cuello	String
posicion_feto	String
consistencia_cuello	String
segmento_inferior	String
membranas	String
presentacion_feto	String
funcion_contractil	String
estado_paciente	String
longitudcuello	ArrayList<String>
lconsis_cuello	ArrayList<String>
lposicion_feto	ArrayList<String>
lseg_inferior	ArrayList<String>
lmembrana	ArrayList<String>
lpres_feto	ArrayList<String>
lfun_contractil	ArrayList<String>
lest_paciente	ArrayList<String>

validNullHosp	ValidadorNull_hospitalizacion
estadoEvol	String
longCuello	String
posFeto	String
consCuello	String
segInferior	String
membrah	String
presFeto	String
funContractil	String
puedeSolicInterv	Boolean
esMenorEdad	Boolean
estadosEvol	List<String>
entityManager	EntityManager
facesMessages	FacesMessages
listarPacientes_hospitalizacion	ListarPacientesControlador
antPersonalesHospList_custom	AntPersonalesCustomList_hospitalizacion
antFamiliarsHospList_custom	AntFamiliarsCustomList_hospitalizacion
habitosPsicoHospList_custom	HabitosPsicoCustomList_hospitalizacion
inmunizacionesHospList_custom	InmunizacionesCustomList_hospitalizacion
crearSolicitudesHospitalizacion	CrearSolicitudesHospitalizacionControlador
modificarSolicitudesHospitalizacion	ModificarSolicitudesHospitalizacionControlador
cieHospList_custom	CieCustomList_hospitalizacion
consultaAccionesHosp	ConsultaAccionesHospControlador

comprobarReglasnegocioHosp	ComprobarReglasNegocioHospControlador
crearDiagnosticoMedicoHosp	CrearDiagnosticoMedicoHospControlador
crearAntecedentesMenorEdadHosp	CrearAntecedentesMenorEdadHospControlador
antObtetricosHospList_custom	AntObtetricosCustomList_hospitalizacion
antObstetricosAbortoHospList_custom	AntObstetricosAbortoCustomList_hospitalizacion
datosNinnoList	ListadoControler<AntObstDatosNinno_hospitalizacion>
generadorDocumentoCDAHosp	GeneradorDocumentoCDAHospControlador
hcManager	HcManager
notificationManager	NotificationManager
bitacora	IBitacora
activeModule	IActiveModule
Para cada responsabilidad:	
Nombre:	obtenerEstadosEvolPac
Descripción:	Permite obtener el estado del paciente al cual se le realiza la evolución médica
Nombre:	asignarCampos
Descripción:	Carga los datos de interés del paciente en el sistema necesarios para la evolución médica
Nombre:	verificarConversacionTerminada
Descripción:	Permite controlar el estado de la conversación activa para evitar pérdida de datos
Nombre:	save
Descripción:	Realiza la validación de todos los datos y si es satisfactoria persiste dichos elementos en la base de datos

Tabla 3.2.1 CrearEvolucionMedicaPrePartoHospControlador

Nombre: CrearHojaPreParto	
Tipo de clase: Controladora	
Atributo	Tipo
hojaHosp	HojaHospitalizacion_hospitalizacion
hojaGeneral	HojaHospitalizacion
signosVitales_hospitalizacion	SignosVitales_hospitalizacion
signosVitalesFeto_hospitalizacion	SignosVitalesFeto_hospitalizacion
caracteristicasFR	CaracteristicaFR_hospitalizacion
caracteristicaPulso	CaracteristicaPulso_hospitalizacion
ubicacionPulso	UbicacionPulso_hospitalizacion
examFis	GeneralExamFis_hospitalizacion
examFisPiel	GeneralExamFisPiel_hospitalizacion
examFisCabeza	GeneralExamFisCabeza_hospitalizacion
examFisOjos	GeneralExamFisOjos_hospitalizacion
examFisOidos	GeneralExamFisOidos_hospitalizacion
examFisNariz	GeneralExamFisNariz_hospitalizacion
examFisBoca	GeneralExamFisBoca_hospitalizacion
examFisFaringe	GeneralExamFisFaringe_hospitalizacion
examFisCuello	GeneralExamFisCuello_hospitalizacion
examFisGangLinf	GeneralExamFisGangliosLinf_hospitalizacion
examFisTorax	GeneralExamFisTorax_hospitalizacion
examFisSenos	GeneralExamFisSenos_hospitalizacion
examFisPulmones	GeneralExamFisPulmones_hospitalizacion

examFisCorazon	GeneralExamFisCorazon_hospitalizacion
examFisVasosSAng	GeneralExamFisVasosSang_hospitalizacion
examFisAbdomen	GeneralExamFisAbdomen_hospitalizacion
examFisUrinario	GeneralExamFisUrinario_hospitalizacion
examFisGenitMasc	GeneralExamFisGenitalMasc_hospitalizacion
examFisGenitFem	GeneralExamFisGenitalFem_hospitalizacion
examFisRecto	GeneralExamFisRecto_hospitalizacion
examFisHuesos	GeneralExamFisHuesos_hospitalizacion
examFisExtremid	GeneralExamFisExtremidades_hospitalizacion
examFisNeuro	GeneralExamFisNeuro_hospitalizacion
examFisSensorial	GeneralExamFisSensorial_hospitalizacion
preparto_hospitalizacion	Preparto_hospitalizacion
prepartoDatosRecibimiento_hospitalizacion	PrepartoDatosRecibimiento_hospitalizacion
prepartoDatRecibAtenPrenat_hospitalizacion	PrepartoDatRecibAtenPrenat_hospitalizacion
prepartoDatRecibPeriodoDilat_hospitalizacion	PrepartoDatRecibPeriodoDilat_hospitalizacion
medico	Medico_hospitalizacion
admision	Admision_hospitalizacion
antPrenatObs	AntPrenatObst_hospitalizacion
periodoNeo	PeriodoNeonatal_hospitalizacion
alimentacion	Alimentacion_hospitalizacion
baseComplementoAlimenticio	BaseComplementoAlimenticio_hospitalizacion
desarrollo	Desarrollo_hospitalizacion
estadoEvolucion	EstadoEvolucionPaciente_hospitalizacion

sistolica	Float
diastolica	Float
pulso_valor	Float
fr_valor	Float
temperatura	Float
frecuencia_cardiaca	Float
cir_cefalica	Float
saturacion_oxig	Float
diuresis_horaria	Float
presion_venosa	Float
complicaciones	String
serologia	String
captacion	String
gpeso	String
caracrupturas	String
tiporup	String
caractf	String
caractp	String
ubicp	String
liquidoa	String
lista_complicaciones	ArrayList<String>
lista_serologia	ArrayList<String>
lista_captacion	ArrayList<String>

lista_gpeso	ArrayList<String>
lista_liquidoa	ArrayList<String>
lista_carcrupturas	ArrayList<String>
lista_tiporup	ArrayList<String>
lista_caractf	ArrayList<String>
lista_caractp	ArrayList<String>
lista_ubicp	ArrayList<String>
entityManager	EntityManager
facesMessages	FacesMessages
antPersonalesHospList_custom	AntPersonalesCustomList_hospitalizacion
antFamiliaresHospList_custom	AntFamiliaresCustomList_hospitalizacion
habitosPsicoHospList_custom	HabitosPsicoCustomList_hospitalizacion
inmunizacionesHospList_custom	InmunizacionesCustomList_hospitalizacion
crearSolicitudesHospitalizacion	CrearSolicitudesHospitalizacionControlador
modificarSolicitudesHospitalizacion	ModificarSolicitudesHospitalizacionControlador
cieHospList_custom	CieCustomList_hospitalizacion
consultaAccionesHosp	ConsultaAccionesHospControlador
comprobarReglasnegocioHosp	ComprobarReglasNegocioHospControlador
crearDiagnosticoMedicoHosp	CrearDiagnosticoMedicoHospControlador
crearAntecedentesMenorEdadHosp	CrearAntecedentesMenorEdadHospControlador
validNullHosp	ValidadorNull_hospitalizacion
hcManager	HCManager
esMenorEdad	Boolean

estadoEvolucionPac	String
Para cada responsabilidad:	
Nombre:	asignarCampos
Descripción:	Carga los datos de interés del paciente en el sistema necesarios para la hoja de parto
Nombre:	obtenerEstadosEvolPac
Descripción:	Obtiene el estado actual del paciente al cual se le realizara la hoja de parto
Nombre:	tanner
Descripción:	Crea la entidad tanner correspondiente al paciente según el valor que se le introduce en la hoja de parto
Nombre:	save
Descripción:	Valida la entrada de datos para el paciente y persiste la hoja de parto en la base de datos

Tabla 3.2.3 CrearHojaPreParto

3.3. Modelo de Datos

Describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Consiste en una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Permite describir las estructuras de la base de datos (el tipo de datos que incluye la base de datos y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregar, borrar, modificar y recuperar los datos). [31]

En un enfoque más amplio, un Modelo de Datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan estos elementos entre sí.

A continuación se presenta el diagrama del Modelo de Datos:

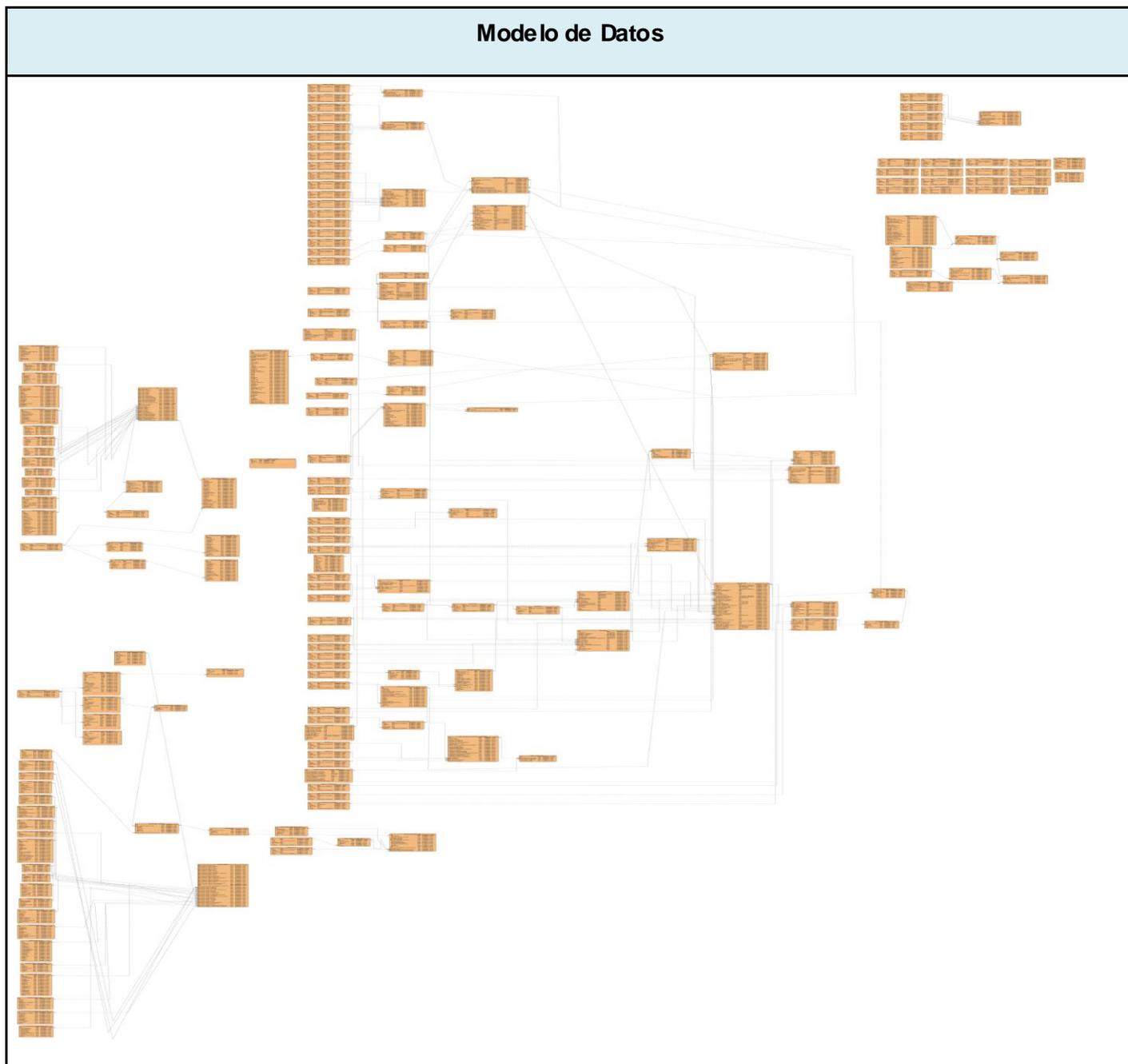


Figura 3.3.1 Modelo de Datos

Modelo de Datos_hoja_recien_nacido

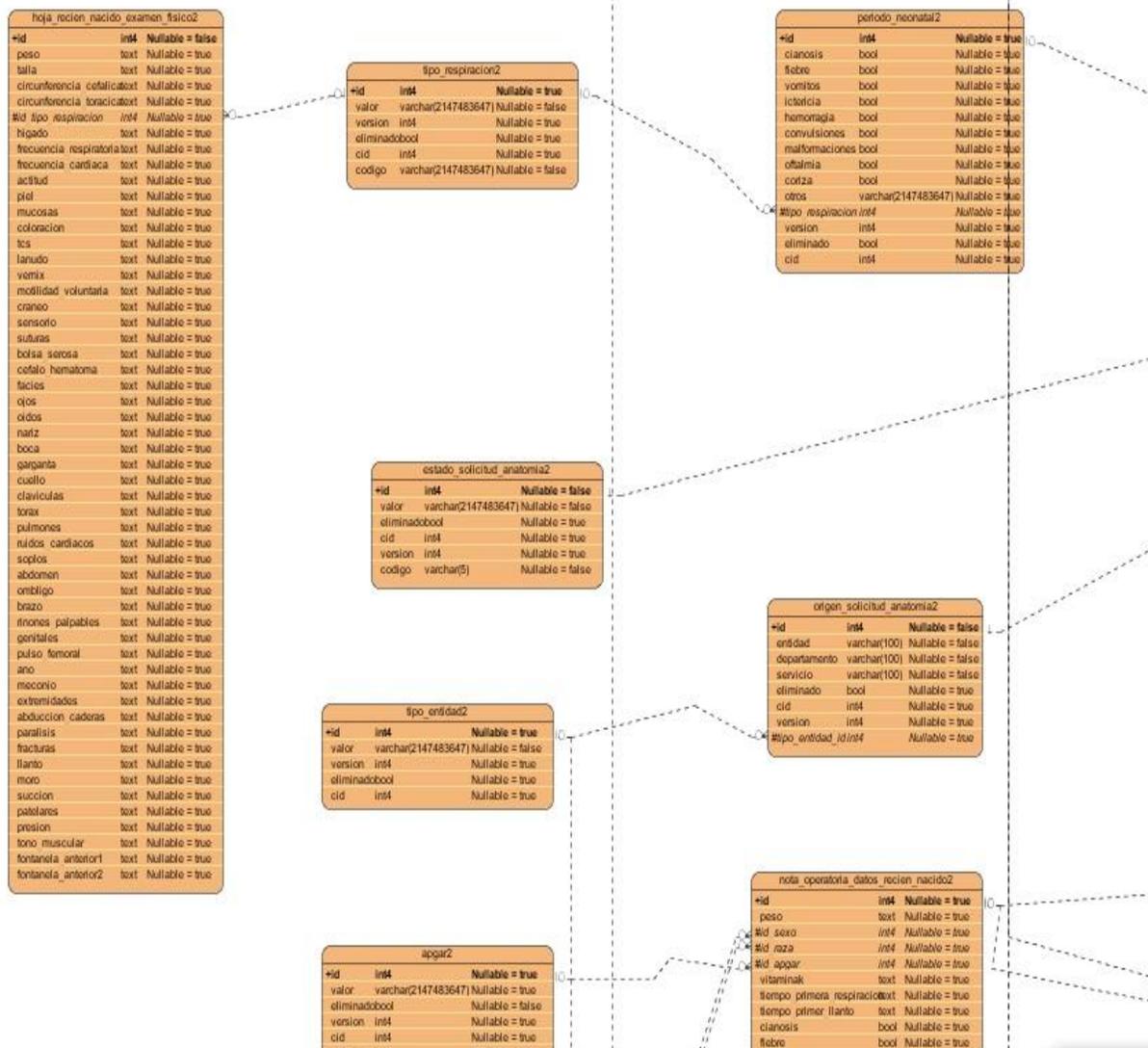


Figura 3.3.2 Modelo de Datos_hoja_recien_nacido

Descripción de las tablas

Nombre: hc_local.evolucion_medica_postparto			
Descripción: Tabla para registrar los datos de las evoluciones médicas en postparto.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador de la evolución médica de postparto. (PK)
id_sangramiento	Integer	si	Identificador de sangramiento. (FK)
gradiente_termico	Boolean	si	Nombre de la evolución.
llene_capilar	Integer	si	Descripción del área.
id_evolucion_medica	Integer	si	Identificador de la evolución médica realizada en la evolución de postparto. (FK)
id_evol_examen_fis_post	Integer	si	Identificador de la evolución del examen físico de postparto. (FK)
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
eliminado	Boolean	si	Permite conocer si la evolución médica de postparto ha sido eliminada.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.1 Hc_local.evolucion_medica_postparto

Nombre: hc_local.evolucion_medica_preparto			
Descripción: Tabla para registrar los datos de las evoluciones médicas en preparto.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador de la evolución médica de preparto.
id_evolucion_medica	Integer	si	Identificador de la evolución médica realizada en la evolución de preparto.(FK)
dilatación	Integer	si	Dilatación que presenta la madre.

foco_fetal	Integer	si	Foco fetal de la evolución médica de parto.
id_funcion_contractil	Integer	si	Identificador de la función contráctil.(FK)
id_presentacion_feto	Integer	si	Identificador de la presentación del feto.(FK)
id_membranas	Integer	si	Identificador de las membranas.(FK)
id_segmento_inferior	Integer	si	Identificador de segmento inferior. (FK)
id_consistencia_cuello	Integer	si	Identificador de la consistencia del cuello. (FK)
id_posicion_feto	Integer	si	Identificador de la posición del feto. (FK)
id_longitud_cuello	Integer	si	Identificador de la longitud del cuello. (FK)
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
eliminado	Boolean	si	Permite conocer si la evolución médica de parto ha sido eliminada.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.2 Hc_local.evolucion_medica_parto

Nombre : hc_local.preparto			
Descripción: Tabla para registrar los datos de la embarazada en el parto.			
Atributo	Tipo	Nulo	Descripción
Id	Integer	no	Identificador de parto.(PK)
id_datos_recibimiento	Integer	si	Identificador de los datos de recibimientos realizados en parto.(FK)
id_signos_vitales	Integer	si	Identificador de los signos vitales realizados en parto.(FK)
id_examen_fisico	Integer	si	Identificador del examen físico realizado en parto.(FK)
hora_recibimiento	Time	si	Hora que se recibe a la madre en parto, se especifica en el formato hh:mm
fecha_recibimiento	Date	si	Fecha que se recibe a la madre en parto, se especifica en el

			formato dd/mm/aaaa.
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
eliminado	Boolean	si	Permite conocer si parto ha sido eliminado.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.3 Hc_local.preparto

Nombre: hc_local.preparto_dat_recib_aten_prenat			
Descripción: Tabla para registrar los datos del recibimiento a la embarazada en la atención prenatal.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador de parto, datos recibimiento, atención prenatal.(PK)
ultrasonidos	Integer	si	Cantidad de ultrasonidos realizados.
altura_uterina	Integer	si	Altura del útero.
formula_menstrual	Real	si	Cantidad de menstruación.
partos_a_terminos	Integer	si	Cantidad de partos a términos.
tiempo_con_anticonceptivo	Integer	si	Duración de la madre con un anticonceptivo.
abortos_provocados	Integer	si	Cantidad de abortos provocados a la madre.
embarazos	Boolean	si	Cantidad de embarazos de la madre.
edad_primera_realaciones sexuales	Integer	si	Edad de la primera relación sexual.
fecha_menarquia	Date	si	Fecha de la menarquía, se especifica en el formato dd/mm/aaaa.
abortos_espontaneo	Integer	si	Cantidad de abortos espontáneos.
edad_gestacional	Integer	si	Semanas de embarazo.
muerdes_neonatales_infantiles	Integer	si	Cantidad de muertes de los niños.
muerdes_fetales	Integer	si	Cantidad de muertes de los fetos.
fecha_prueba_citologica	Date	si	Fecha de la prueba citológica de la madre.

causas_muertes_fetales	Text	si	Causas de muerte de los fetos.
causas_muertes_neonatales_infantiles	Text	si	Causas de muerte de los niños.
resultados_prueba_citologica	Text	si	Resultados de las pruebas realizadas a la madre.
disfucion_placentaria	Text	si	Problemas en la placenta.
ingreso_durante_embarazo	Text	si	Ingreso de la madre durante el embarazo.
hormonas_rad_en_gestacion	Text	si	Hormonas durante la gestación.
id_captacion	Integer	si	Identificador de captación. (FK)
id_serologia	Integer	si	Identificador de serología. (FK)
id_grupo_sanguineo	Integer	si	Identificador del grupo sanguíneo. (FK)
id_ganacia_peso	Integer	si	Identificador de ganancia de peso. (FK)
eliminado	Boolean	si	Permite conocer si los datos de recibimiento de la atención prenatal han sido eliminados.
id_complicaciones	Integer	si	Identificador de las complicaciones. (FK)

Tabla 3.4.4 Hc_local.preparto_dat_recib_aten_prenat

Nombre: hc_local.preparto_dat_recib_periodo_dilat			
Descripción: Tabla para registrar los datos del recibimiento durante el período de dilatación de la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador de preparto, datos recibimiento del período de dilatación.(PK)

hora_primeros_dolores	Time(6)	si	Hora de los primeros dolores de la madre, se especifica en el formato hh:mm:ss.
hora_dilatacion_completa	Time(6)	si	Hora de la dilatación completa de la madre, se especifica en el formato hh:mm:ss.
fecha_ruptura_membrana	Date	si	Fecha en que la madre rompe las membranas, se especifica en el formato dd/mm/aaaa.
id_tipo_ruptura	Integer	si	Identificador del tipo de ruptura. (FK)
id_caracteristicas_ruptura	Integer	si	Identificador de las características de ruptura. (FK)
fecha_primeros_dolores	Date	si	Fecha en que la madre siente los primeros dolores, se especifica en el formato dd/mm/aaaa.
fecha_dilatacion_completa	Date	si	Fecha de la dilatación completa de la madre, se especifica en el formato dd/mm/aaaa.
eliminado	Boolean	si	Permite conocer si los datos de recibimiento del período de dilatación han sido eliminados.
id_liquido_amniotico	Integer	si	Identificador del líquido amniótico. (FK)
hora_duracion_periodo	Time	si	Hora de duración del período, se especifica en el formato hh:mm.

Tabla 3.4.5 Hc_local.preparto_dat_recib_periodo_dilat

Nombre: hc_local.preparto_datos_recibimiento			
Descripción: Tabla para registrar los datos del recibimiento de la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador de preparto datos recibimiento.(PK)

pronostico_materno	Text	si	Pronóstico de la madre.
pronostico_fetal	Text	si	Pronóstico del feto.
id_preparto_atencion_dilatacion	Integer	si	Identificador de los datos de atención a la dilatación realizados en datos recibimientos.(FK)
id_preparto_atencion_prenatal	Integer	si	Identificador de los datos de atención prenatal realizados en datos recibimientos.(FK)
eliminado	Boolean	si	Permite conocer si los datos de recibimiento han sido eliminados.

Tabla 3.4.6 Hc_local.preparto_datos_recibimiento

Nombre: hc_local.obstet_exam_obstetrico_abdomen			
Descripción: Tabla para registrar los datos del examen obstétrico al abdomen de la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador del examen obstétrico al abdomen.(PK)
id_frecuencia_cardiaca_fetal	Integer	si	Identificador de la frecuencia cardíaca fetal. (FK)
id_presentacion	Integer	si	Identificador de la presentación. (FK)
id_dorso	Integer	si	Identificador del dorso. (FK)
encajamiento	Boolean	si	Encajamiento del feto.
eliminado	Boolean	si	Permite conocer si los datos del examen obstétrico del abdomen han sido eliminados.
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.7 Hc_local.obstet_exam_obstetrico_abdomen

Nombre: hc_local.obstet_exam_obstetrico_cuello_vaginal			
Descripción: Tabla para registrar los datos del examen obstétrico al cuello de la vagina en la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador del examen obstétrico al cuello de la vagina.(PK)
id_reconocimiento_especulo	Integer	si	Identificador de los datos al reconocimiento con espéculo realizados en el examen obstétrico al cuello de la vagina. (FK)
id_reconocimiento_tacto_vaginal	Integer	si	Identificador de los datos al reconocimiento con tacto vaginal realizados en el examen obstétrico al cuello de la vagina. (FK)
eliminado	Boolean	si	Permite conocer si los datos del examen obstétrico al cuello de la vagina han sido eliminados.
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.8 Hc_local.obstet_exam_obstetrico_cuello_vaginal

Nombre: hc_local.obstet_exam_obstetrico_cuello_vaginal_rec_especulo			
Descripción: Tabla para registrar los datos del examen obstétrico al cuello de la vagina a través del reconocimiento con espéculo en la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador del examen obstétrico al cuello de la vagina a través del reconocimiento con espéculo.(PK)
id_posicion	Integer	si	Identificador de la posición. (FK)
id_longitud	Integer	si	Identificador de la longitud. (FK)

permeable	Boolean	si	Permite conocer si la vagina se encuentra permeable.
salida_liquido_amniotico	Boolean	si	Permite conocer si existe pérdida de líquido amniótico.
heucorrea	Boolean	si	Permite conocer si la embarazada presenta heucorrea.
cordon_en_vagina	Boolean	si	Permite conocer la situación del cordón en el cuello vaginal.
presencia_tumoraciones	Boolean	si	Permite conocer si existen tumoraciones en el cuello vaginal.
descripción_tumoraciones	Text	si	Permite describir las tumoraciones encontradas en el cuello vaginal.
eliminado	Boolean	si	Permite conocer si los datos del examen obstétrico al cuello vaginal a través del reconocimiento con espéculo han sido eliminados.
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.9 Hc_local.obstet_exam_obstetrico_cuello_vaginal_rec_especulo

Nombre: hc_local.obstet_exam_obstetrico_cuello_vaginal_rec_tacto_vaginal			
Descripción: Tabla para registrar los datos del examen obstétrico al cuello de la vagina a través del reconocimiento con tacto vaginal en la embarazada.			
Atributo	Tipo	Nulo	Descripción
id	Integer	no	Identificador del examen obstétrico al cuello vaginal a través del reconocimiento con tacto vaginal.(PK)
dilatación	Boolean	si	Permite conocer si existe dilatación en el cuello vaginal.
tamaño	Integer	si	Permite conocer el tamaño del cuello vaginal, se mide en

			centímetros (cm).
id_consistencia_cuello	Integer	si	Identificador de la consistencia del cuello. (FK)
id_bordes	Integer	si	Identificador de los bordes. (FK)
id_plano_hodge	Integer	si	Identificador del plano de hodge. (FK)
id_promontorio	Integer	si	Identificador del promontorio. (FK)
id_espina_ciatica	Integer	si	Identificador de la espina ciática. (FK)
angulo_subpubico_recto	Boolean	si	Permite conocer el ángulo subpúbico recto del cuello vaginal.
perine_resistente	Boolean	si	Permite conocer si el periné es resistente.
eliminado	Boolean	si	Permite conocer si los datos del examen obstétrico al cuello vaginal a través del reconocimiento con tacto vaginal han sido eliminados.
version	Integer	si	Indica con cuál versión de la entidad se está trabajando.
cid	Integer	si	Identificador de modificaciones registradas en la bitácora.

Tabla 3.4.10 Hc_local.obstet_exam_obstetrico_cuello_vaginal_rec_tacto_vaginal

3.4. Breve valoración de las técnicas de validación

El proceso de validación consiste en demostrar mediante pruebas y verificaciones, debidamente ejecutados y documentadas, que un equipo o sistema está correctamente calibrado, asegurando que su utilización futura es fiable y está dentro de los límites operacionales especificados.

En la construcción de aplicaciones lo normal es validar la entrada de datos de los usuarios, evitando la entrada de errores o intentos de violar la seguridad de la aplicación. El framework JSF proporciona ayuda para definir de manera fácil este tipo de validaciones en la vista. El problema principal que trae

consigo es tener que repetir las mismas validaciones en diferentes formularios de nuestra aplicación con los mismos objetos de negocio, rompiendo el principio DRY (Don't repeat yourself – no te repitas). Hibernate Validator pretende resolver este problema, de forma que se logre definir, una única vez, las validaciones en los objetos de negocio, e invocaremos dichas validaciones en el punto donde interese. JBoss Seam hace uso de esta capacidad y la integra con JSF.

Principales características de Hibernate Validator:

- Se definen las validaciones fácilmente con anotaciones.
- Se integra directamente con Hibernate, de forma que antes de hacer una inserción o actualización se validarán los objetos.
- Trae consigo un conjunto predefinido de validaciones las que se pueden extender con nuevas validaciones propias.
- Las validaciones que se indiquen se tendrán en cuenta a la hora de generar el DLL (los scripts de creación de la base de datos).

3.5. Vista de Implementación

La Vista de Implementación muestra el empaquetado físico de las partes reutilizables del sistema en unidades sustituibles, llamadas componentes, así como los elementos físicos del sistema, como sus interfaces y dependencia entre componentes. Los componentes son piezas reutilizables de alto nivel a partir de las cuales se puede construir sistemas. Esta vista tiene como propósito definir la organización del código, planificar las integraciones del sistema necesarias en cada iteración e implementar las clases y subsistemas definidos durante el diseño.

Para ilustrar la Vista de Implementación se construyen los Diagramas de Componentes haciendo uso del lenguaje UML. Un Diagrama de Componentes describe la descomposición física del sistema en componentes. Los componentes identifican objetos físicos que hay en tiempo de ejecución, de complicación o de desarrollo y tienen identidad propia como una interfaz bien definida.

A continuación se presenta el Diagrama de Paquetes de Componentes correspondiente al sistema presentado, implementado con tres componentes fundamentales: Vista, Controlador y Modelo, que a su vez contienen otros componentes.

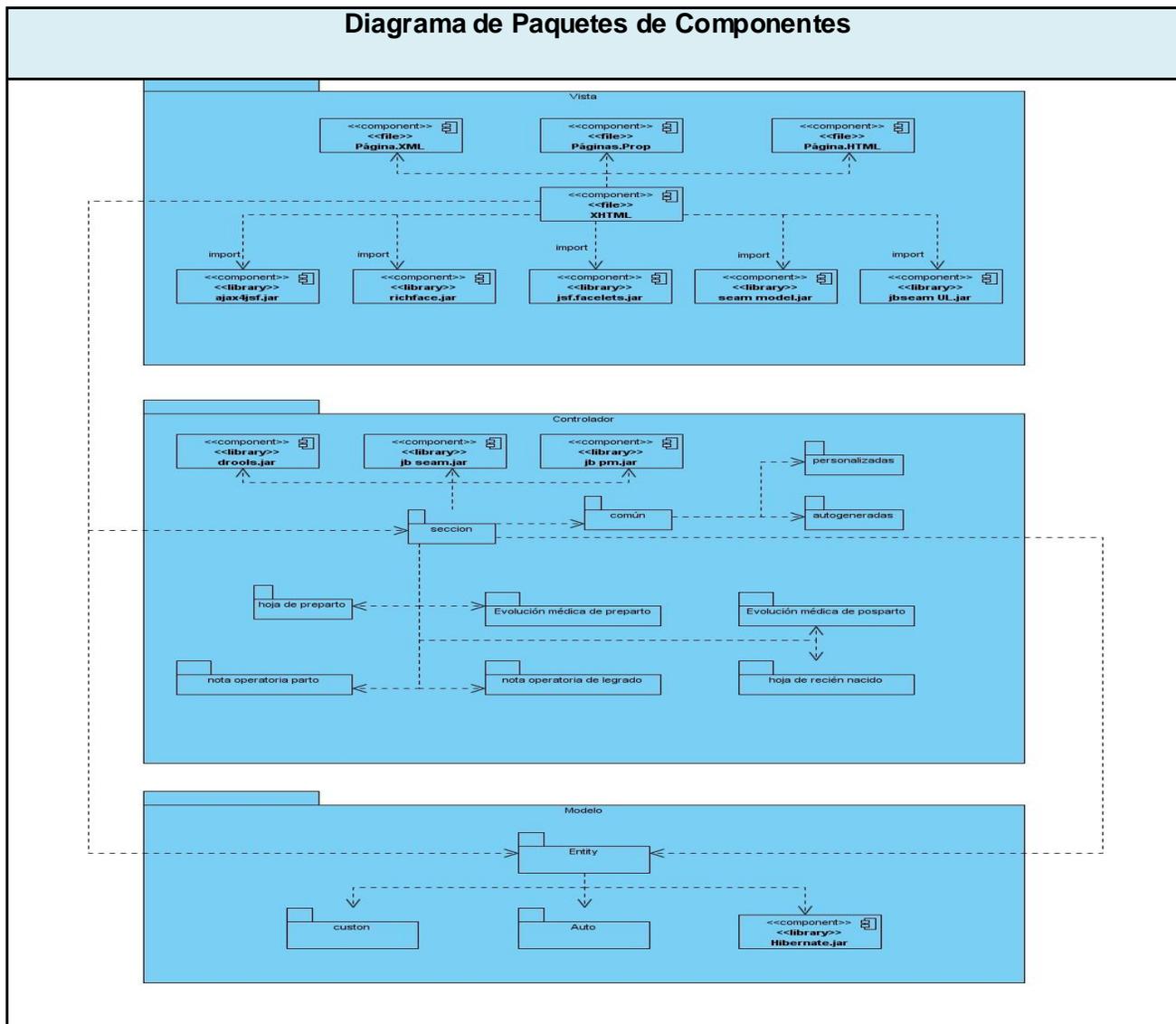


Figura 3.5.1 Diagrama de Paquetes de Componentes

En este capítulo se realizó una valoración del diseño propuesto por el analista y su estructura. De cada clase se realizó una breve descripción de sus atributos y métodos. Se obtuvo el Modelo de Datos del sistema y el Diagrama de Paquetes de Componentes relacionado con el Modelo de Diseño propuesto.

CAPITULO 4. Modelo de pruebas

En el desarrollo de este capítulo se presentará el Modelo de Pruebas, para lo cual se realiza una caracterización de las pruebas de caja negra, el cual se ha determinado como método a utilizar. Al mismo tiempo se definen y describen los casos de prueba.

Para cada fase de desarrollo de un producto reviste gran importancia la ejecución de las pruebas de software ya que son las que permiten verificar la calidad de los sistemas y finalmente se logra un sistema de mayor calidad y por ende tendrá mayor grado de aceptación en el mercado.

Las pruebas de software se dividen en dos tipos: las pruebas de caja blanca y de caja negra. A continuación se explica las características de la prueba de caja negra.

4.1. Prueba de caja negra

Se denomina prueba de caja negra a la acción que se lleva a cabo con el objetivo de demostrar que cada función es completamente operativa. La prueba de caja negra se centra principalmente en los requisitos funcionales del software y permite obtener un conjunto de condiciones de entrada que ejerciten estos requisitos y se ignora la estructura de control. No es más que la prueba que se realiza sobre la interfaz del software. Se conoce también como prueba de caja opaca o prueba funcional. Este tipo de prueba no se aplica a los módulos que trabajan en forma transparente al usuario. [32]

Para llevar a cabo dicha prueba existen varias técnicas, entre las que se pueden citar las siguientes:

- Técnica de la partición de equivalencia: Esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del análisis de valores límites: Esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de grafos de causa-efecto: Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Se utilizará la técnica de la partición de equivalencia permitiendo realizar los diseños de casos de prueba, estos consisten en definir un conjunto de datos de entrada que representan estados válidos y no válidos, para comprobar que la respuesta en ambos casos sea la esperada.

Un Caso de Prueba es un conjunto de entradas de pruebas donde se describen escenarios y se identifican variables. Estas últimas representan a un conjunto de estados válidos o inválidos para las condiciones de entrada para demostrar que las funciones del software son operativas.

Los casos de prueba descritos consisten principalmente en tres partes con subdivisiones:

Introducción o visión general que contiene información general acerca de los Casos de Prueba, consta de:

- Un identificador único para futuras referencias, por ejemplo, mientras se describe un defecto encontrado.
- El dueño o creador, que es el nombre del analista o diseñador de pruebas: quien ha desarrollado pruebas o es responsable de su desarrollo.
- Versión: la actual definición del caso de prueba.
- El nombre del caso de prueba, que debe ser un título entendible por personas, para la fácil comprensión del propósito del caso de prueba y su campo de aplicación.
- Un identificador de requerimientos el cual está incluido por el caso de prueba. También aquí puede ser identificador de casos de uso o especificación funcional.
- El propósito: contiene una breve descripción del objetivo de la prueba, y la funcionalidad que chequea.
- Las dependencias entre los casos de uso.

Actividades de los casos de prueba:

- El ambiente de prueba o configuración: contiene información acerca de la configuración del hardware o software en el cual se ejecutará el caso de prueba.
- La inicialización: es donde se describen acciones que deben ser ejecutadas antes de que los casos de prueba se hayan inicializado. Por ejemplo: no se debe abrir algún archivo.
- La finalización: describe acciones que deben ser ejecutadas después de realizado el caso de prueba. Por ejemplo: Si el caso de prueba daña la base de datos, el analista debe restaurarla antes de que otro caso de prueba sea ejecutado.
- Las acciones o pasos a realizar para completar la prueba.

- La descripción de los datos de entrada.

Resultados

- Los resultados esperados: contienen una descripción de lo que el analista debería ver tras haber completado todos los pasos de la prueba.
- Los resultados reales: contienen una breve descripción de lo que el analista encuentra después de que los pasos de prueba se hayan completado. Esto se sustituye a menudo con un Correcto o Fallido.

4.2. Descripción de los casos de pruebas

Caso de prueba: Crear evolución médica en parto

Escenarios del Crear evolución médica en parto	Descripción de la funcionalidad	Flujo Central
EC 1: Crear evolución médica en parto	Crear la evolución de la paciente en parto satisfactoriamente.	Se selecciona la opción Crear evolución médica parto. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Aceptar. Se regresa a la vista anterior listar paciente.
EC 2: Cancelar operación	Cancelar la opción de Crear evolución médica en parto.	Se selecciona la opción Crear evolución médica parto. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior listar paciente.
EC 3: Existen datos incompletos	Luego de haber introducido los datos el sistema los verifica y valida, de existir incompletos muestra un indicador sobre los	Se selecciona la opción Crear evolución médica parto.

	campos incompletos.	Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.
--	---------------------	--

Tabla 4.2 CP Crear evolución médica parto.

Id del escenario	EC 1	EC 2	EC 3
Escenario	Crear evolución médica en parto	Cancelar operación	Existen datos incompletos
Variable 1 (Longitud del cuello)	V		I
Variable 2 (Posición del feto)	V		I
Variable 3 (Consistencia del cuello)	V		I
Variable 4 (Segmento inferior)	V		I
Variable 5 (Membranas)	V		I
Variable 6 (Presentación del feto)	V		I
Variable 7 (Función contráctil)	V		I
Variable 8 (Foco fetal)	V		I
Variable 9 (Dilatación)	V		I

Variable 10 (Estado del paciente)	V		I
Variable 11 (Observaciones)			
Botón 1 (Aceptar)	NA		NA
Botón 2 (Cancelar)		NA	
Respuesta del Sistema	Crear evolución médica en preparto y regresa a la vista anterior listar paciente.	Regresa a la vista anterior de listar paciente.	Muestra un indicador sobre los campos incompletos.
Resultado de la Prueba			

Tabla 4.3 SC Crear evolución médica preparto.

Caso de prueba: Crear evolución médica en postparto

Escenarios del Crear evolución médica en postparto	Descripción de la funcionalidad	Flujo Central
EC 1: Crear evolución médica en postparto	Crear la evolución de la paciente en postparto satisfactoriamente.	Se selecciona la opción Crear evolución médica postparto. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Aceptar. Se regresa a la vista anterior listar paciente.
EC 2: Cancelar operación	Cancelar la opción de Crear evolución médica en postparto.	Se selecciona la opción Crear evolución médica postparto. Se introducen o seleccionan los datos

		correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior listar paciente.
EC 3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de existir incompletos muestra un indicador sobre los campos incompletos.	Se selecciona la opción Crear evolución médica postparto. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.

Tabla 4.4 CP Crear evolución médica postparto.

Id del escenario	EC 1	EC 2	EC 3
Escenario	Crear evolución médica en posparto	Cancelar operación	Existen datos incompletos
Variable 1 (Sangramiento)	V		I
Variable 2 (Llene capilar)	V		I
Variable 3 (Gradiente térmico)			
Variable 4 (Mucosas)			
Variable 5 (Masaje uterino)			
Variable 6 (Coloración)			

Variable 7 (Olor)			
Variable 8 (Útero contraído)			
Variable 9 (Lactando)			
Variable 10 (Calostro)			
Variable 11 (Estado)			
Variable 12 (Orinó)			
Variable 13 (Características orine)			
Variable 14 (Diuresis)			
Variable 15 (Forma diuresis)			
Variable 16 (Estado del paciente)	V		I
Variable 17 (Observaciones)			
Botón 1 (Aceptar)	NA		NA
Botón 2 (Cancelar)		NA	

Respuesta del Sistema	Crear evolución médica en postparto y regresa a la vista anterior listar paciente.	Regresa a la vista anterior de listar paciente.	Muestra un indicador sobre los campos incompletos.
Resultado de la Prueba			

Tabla 4.5 SC Crear evolución médica postparto.

Caso de prueba: Crear hoja de parto

Escenarios del Crear hoja de parto	Descripción de la funcionalidad	Flujo Central
EC 1: Crear hoja de parto	Crear la hoja de parto de la paciente satisfactoriamente una vez hospitalizada.	Se selecciona la opción Crear hoja de parto. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Aceptar. Se regresa a la vista anterior listar paciente.
EC 2: Cancelar operación	Cancelar la opción de Crear hoja de parto.	Se selecciona la opción Crear hoja de parto. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior listar paciente.
EC 3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de existir incompletos muestra un indicador sobre los campos incompletos.	Se selecciona la opción Crear hoja de parto. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.

EC 4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de existir incorrectos muestra un indicador sobre los campos incorrectos.	<p>Se selecciona la opción Crear hoja de parto.</p> <p>Se introducen los datos incorrectos.</p> <p>Se selecciona la opción Aceptar.</p> <p>Muestra un indicador sobre los campos incorrectos.</p>
---------------------------------	---	---

Tabla 4.6 CP Crear hoja de parto.

Id del escenario	EC 1	EC 2	EC 3	EC 4
Escenario	Crear hoja de parto	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Variable 1 (Fecha de entrada a parto)				
Variable 2 (Hora de entrada a parto)				
Variable 3 (Captación)				
Variable 4 (Serología)				
Variable 5 (Estado del paciente)	V		I	
Variable 6 (Ganancia de peso)				
Variable 7 (Complicaciones)				
Variable 8 (Altura uterina)				

Variable 9 (Ultrasonidos)				
Variable 10 (Muertes neonatales e infantiles)				
Variable 11 (Fecha prueba citológica)				
Variable 12 (Causa muertes neonatales e infantiles)				
Variable 13 (Disfunción placentaria)				
Variable 14 (Resultados pruebas citológica)				
Variable 15 (Fecha primeros dolores)				
Variable 16 (Hora primeros dolores)				
Variable 17 (Fecha dilatación completa)				
Variable 18 (Hora dilatación completa)				
Variable 19 (Fecha ruptura de membranas)				
Variable 20 (Hora ruptura de membranas)				
Variable 21 (Liquido amniótico)				

Variable 22 (Características ruptura)				
Variable 23 (Tipo ruptura)				
Variable 24 (Hora período dilatación)				
Variable 25 (Minutos período dilatación)				
Variable 26 (Pronóstico materno)				
Variable 27 (Pronóstico fetal)				
Variable 28 (Sistólica)	V			
Variable 29 (Diastólica)	V			
Variable 30 (Valor pulso)	V			
Variable 31 (Características pulso)	V			
Variable 32 (Ubicación pulso)	V			
Variable 33 (Características frecuencia respiratoria)	V			
Variable 34 (Valor frecuencia respiratoria)	V			

Variable 35 (Temperatura)	V			
Variable 36 (Frecuencia cardíaca)	V			
Variable 37 (Movimiento fetal)				
Variable 38 (Circunferencia cefálica)				
Variable 39 (Saturación de O2)				
Variable 40 (Diuresis horaria)				
Variable 41 (Presión venosa central)				
Variable 42 (Color de la piel)				
Variable 43 (Erupción)				
Variable 44 (Humedad)				
Variable 45 (Dolor)				
Variable 46 (Fontanelas)				
Variable 47 (Suturas)				

Variable 48 (Conjuntiva)				
Variable 49 (Pupilas)				
Variable 50 (Movimientos)				
Variable 51 (Pabellón)				
Variable 52 (Tímpano)				
Variable 53 (Audición)				
Variable 54 (Fosas nasales)				
Variable 55 (Meatos)				
Variable 56 (Mucosa)				
Variable 57 (Aliento)				
Variable 58 (Lengua)				
Variable 59 (Labios)				
Variable 60 (Amígdalas)				

Variable 61 (Dolor)				
Variable 62 (Adenoides)				
Variable 63 (Perímetro)				
Variable 64 (Ganglios)				
Variable 65 (Vasos)				
Variable 66 (Cervicales)				
Variable 67 (Occipitales)				
Variable 68 (Axilares)				
Variable 69 (Olor)				
Variable 70 (Reflejos)				
Variable 71 (Edema)				
Variable 72 (Sensibilidad)				
Variable 73 (Heces)				

Variable 74 (Vagina)				
Variable 75 (Riñones)				
Variable 76 (Tumoraciones)				
Variable 77 (Pulso)				
Variable 78 (Ruidos)				
Variable 79 (Palpitación)				
Variable 80 (Nódulos)				
Variable 81 (Forma)				
Variable 82 (Observaciones)				
Variable 83 (Conclusiones)				
Botón 2 (Aceptar)	NA		NA	NA
Botón 2 (Cancelar)		NA		
Respuesta del Sistema	Crear hoja de parto y regresa a la vista anterior	Regresa a la vista anterior de listar	Muestra un indicador sobre los campos	Muestra un indicador sobre los campos

	listar paciente.	paciente.	incompletos.	incorrectos.
Resultado de la Prueba				

Tabla 4.7 SC Crear hoja de parto.

A continuación se detallan los elementos probados y su estado final, una vez desarrollado el proceso de pruebas a partir de los Casos de Prueba confeccionados para los diferentes escenarios de los casos de uso asociados a los procesos implementados:

Artefacto	Estado final
Hospitalización, Bloque Quirúrgico	
Primera iteración	34 NC detectadas, 1 NC sin resolver, 2.94 % que representa del total. La NC pendiente constituye una recomendación del Grupo de Calidad del Departamento.
Segunda iteración	9 NC detectadas, 0 NC sin resolver, 0 % que representa del total.

Tabla 4.8 Resumen de No Conformidades por iteración

En la siguiente tabla se desglosan los elementos probados por funcionalidades, interfaces de usuario y pautas de diseño de interfaz de usuario:

Elemento	Cantidad
Funcionalidad	23 NC detectadas, 1 NC sin resolver, 4.34 % que representa del total.
Interfaces de usuario	7 NC detectadas, 0 NC sin resolver, 0 % que representa del total.
Pautas de diseño	12 NC detectadas, 0 NC sin resolver, 0 % que representa del total.

Tabla 4.9 Resumen de No Conformidades por tipo

Mediante el flujo de Pruebas realizado a la implementación de los procesos gineco-obstétricos, a un total de 18 casos de uso, de ellos el 44% de complejidad alta, un 26% de complejidad media y un 30% de complejidad baja se detectaron un total de 23 no conformidades de funcionalidad, 7 de interfaz de usuario y 12 de pautas de diseño de interfaz de usuario, quedando solamente una no conformidad sin resolver pues la misma constituye una recomendación de la presente investigación. Se puede observar que a medida que se avanzó en el desarrollo de las dos iteraciones la cantidad de no conformidades disminuyó considerablemente, así como que la razón de no conformidades por caso de uso estuvo en el orden de uno a uno aproximadamente, por lo antes expuesto se evidencia la calidad de los procesos implementados.

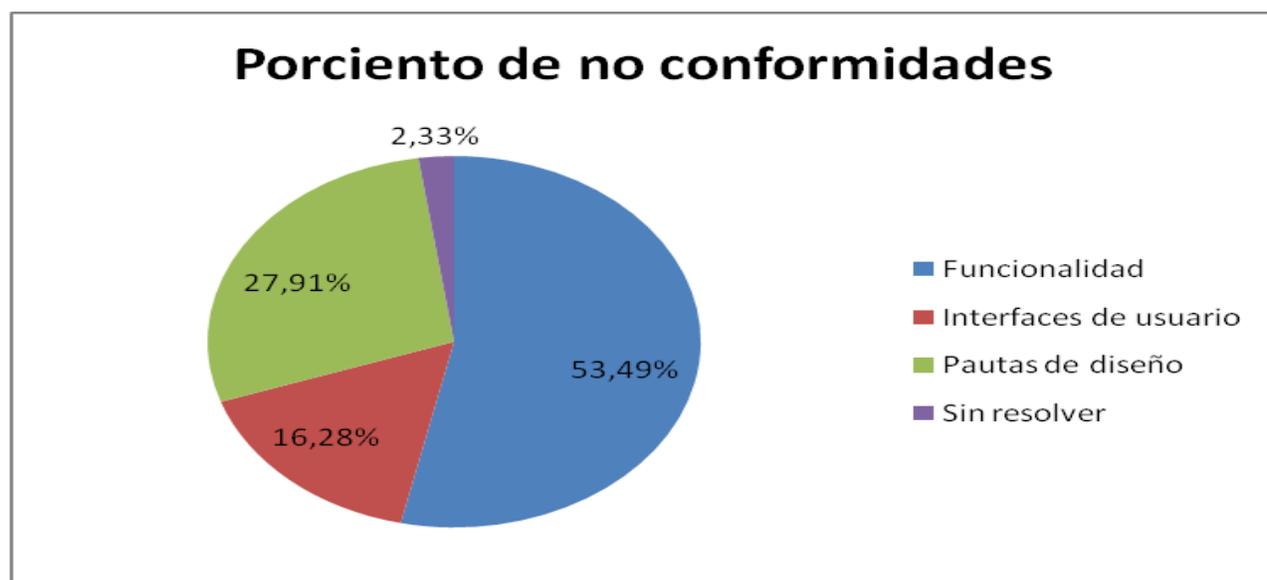


Figura 4.1 Diagrama de porcentaje de NC por tipos con respecto al total

En este capítulo se presentó el Modelo de Pruebas, se realizó una caracterización del método de pruebas de caja negra, el cual se determinó como método a utilizar. Se aplicó la técnica de partición de equivalencia, obteniéndose los casos de prueba como artefactos generados de este flujo de trabajo, siendo estos una guía para las pruebas efectuadas. Finalmente se realiza una valoración de este proceso en función de los resultados obtenidos durante su desarrollo, considerando para ello las no conformidades detectadas por caso de uso, la que permite demostrar la calidad de los procesos implementados.

Conclusiones

Con el desarrollo de los procesos del servicio de gineco-obstetricia del Sistema de Información Hospitalaria alas HIS se arribaron a las siguientes conclusiones:

- El proceso de refinamiento de los requerimientos funcionales evidenció la necesidad de incluir la funcionalidad para la gestión del legrado, no diseñada para el Servicio de Gineco-obstetricia del sistema alas HIS.
- Los sistemas analizados no cumplieron con la totalidad de los requerimientos para la gestión de la información de los procesos gineco-obstétricos que requiere el Sistema de Información Hospitalaria alas HIS.
- La aplicación de la arquitectura propuesta permitió la obtención de un sistema robusto y flexible.
- La implementación de estos procesos permitió un fortalecimiento funcional del sistema alas HIS facilitando la gestión de su información en las instituciones hospitalarias.
- La ejecución de pruebas de caja negra garantizó la calidad necesaria para la futura explotación de las funcionalidades implementadas.

Recomendaciones

Una vez concluida la investigación se recomienda lo siguiente:

- Agregar funcionalidades para desarrollar el proceso de atención a pacientes femeninas embarazadas en el Módulo de Consulta Externa.
- Incluir la gestión de abortos en el servicio de Gineco-obstetricia para el Sistema de Información Hospitalaria alas HIS.
- Permitir la generación de reportes sobre la información generada en el servicio de Gineco-obstetricia.

Referencias bibliográficas

- [1]. **Dr. Cerritos Antonio.** Sistema de Información Hospitalaria. 2003 [Citado el: 25 de Octubre del 2010] <http://educacion.salud.gob.mx/cursos/informatica/HIS/his.pdf>.
- [2]. **Figueredo Gonzalez, Yordanis.** Trabajo de diploma Sala de Parto. 2010. [Citado el: 21 de Septiembre del 2010.]
- [3]. **Niveles de Atención de Salud.** [En línea] [Citado el: 18 de Octubre del 2010.] http://www.bvs.cl/php/decsws.php?tree_id=SP2.001.012&lang=es#.
- [4]. **Metis His.** [En línea]. [Citado el: 22 de Octubre del 2010.] <http://www.apraful.com.uy/soluciones/metis-his.html>
- [5]. **SIGHO.** [En línea] 13 de mayo 2006. [Citado el: 22 de Octubre del 2010.] <http://www.sigho.gob.mx/quees.html>.
- [6]. **SALGEN.** Aplicación Informática para la atención de mujeres embarazadas. [En línea] 8 de junio del 2010. [Citado el: 22 de Octubre del 2010] <http://www.juventudrebelde.cu/cuba/2010-06-29/software-beneficia-programa-materno-infantil/>.
- [7]. **Control Salud.** [En línea]. [Citado el: 6 de diciembre del 2010] <http://www.cristian-silva.com.ar/>.
- [8]. **PACIENTES HIS.** [En línea] 22 de Enero de 2009. [Citado el: 2 de Octubre de 2009.] <http://cnt.com.co/PAGINA/index.asp?id=178>.
- [9]. **Arquitectura basada en capas.** [En línea] 29 de mayo del 2009 [Citado el: 18 de Octubre del 2010] http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10.
- [10]. **BuenMaster.com.** [En línea] 14 de Agosto de 2007. [Citado el: 20 de Octubre de 2010.] <http://buenmaster.com/?a=536>.
- [11]. **Campos, Beatris Prieto.** Diseño y programación de páginas web [En línea] [Citado el: 17 de Octubre de 2010.] <http://atc.ugr.es/~bprieto/paginas-web8/introduccionx.html>.
- [12]. **García, Alejandro Pérez.** desarrolloweb.com. [En línea] 21 de febrero de 2006. [Citado el: 17 de Octubre de 2010.] <http://www.desarrolloweb.com/articulos/2380.php>.

- [13]. **Garrett, Jesse James.** Maestros del web. [En línea] 11 de Junio de 2005. [Citado el: 17 de Octubre de 2010.] <http://www.maestrosdelweb.com/editorial/ajax/>.
- [14]. **JBoss RichFaces.** [En línea] 2009. [Citado el: 20 de Octubre del 2010] <http://labs.jboss.com/portal/jbossrichfaces/>.
- [15]. **JBoss Ajax4jsf** [En línea] 8 de Septiembre del 2010 [Citado el: 21 de Octubre del 2010.] <http://labs.jboss.com/portal/jbossajax4jsf/>
- [16]. **RedHat.** [En línea] 3 de Marzo de 2009. [Citado el: 17 de Octubre de 2010.] http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/seam/Seam_Reference_Guide/A_complete_Seam_application_the_Hotelg_example-e_Seam_UI_control_library.html.
- [17]. **JavaServer Faces y Ajax Parte-1** [En línea] 7 de Agosto del 2007. [Citado el: 21 de Octubre del 2010.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>
- [18]. **Seam Framework** [En línea] 18 de Enero del 2008 [Citado el: 21 de Octubre del 2010.] <http://seamframework.org/Home/WhySeam>.
- [19]. **Mazza, Ramiro Gómez.** DosIdeas. [En línea] 26 de Mayo de 2009. [Citado el: 17 de Octubre de 2010.] <http://www.dosideas.com/java/592-primeros-pasos-con-drools.html>.
- [20]. **Enrique Rodriguez Lasterra.** [En línea] Introducción a Java Persistence API - Presentation Transcript. 2008 [Citado el: 21 de Octubre del 2010.] <http://www.slideshare.net/jamslug/introduccion-a-java-persistence-api>.
- [21]. **Introducción a Enterprise Java Beans.** [En línea] 2002 [Citado el: 21 de Octubre del 2010.] http://www.javahispano.org/contenidos/es/introduccion_a_enterprise_java_beans.
- [22]. **Suárez González, Héctor.** Manual de Hibernate. [En línea] 22 de Abril del 2003 [Citado el: 26 de Octubre del 2010] <http://www.scribd.com/doc/40059777/Manual-Hibernate>.
- [23]. **PostgresSQL.** [En línea] 29 de agosto del 2010. [Citado el: 21 de Octubre del 2010.] <http://www.scribd.com/doc/36570462/postgreSQL-investigacion>.

- [24]. **Paola.** OSUM- Open Source University Meetup. [En línea] 28 de Septiembre de 2009. [Citado el: 17 de Octubre de 2010.] <http://osum.sun.com/group/insptutn/forum/topics/conceptos-clave-del-lenguaje-6>.
- [25]. **Natura Software.** [En línea] 2007. [Citado el: 17 de Octubre de 2010.] <http://www.naturasoftware.com/main.php?f=tecnologia>.
- [26]. **Itera.** [En línea] 2008. [Citado el: 17 de Octubre de 2010.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.
- [27]. **Mora, Francisco.** UML: Lenguaje. [En línea] 2003. [Citado el: 17 de Octubre de 2010.] <http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>.
- [28]. **Milestone consulting.** [En línea] [Citado el: 17 de Octubre de 2010.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.html>.
- [29]. **Especificaciones de Requerimientos.** [En línea] [Citado el: 10 de Enero de 2011.] <http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>.
- [30]. **Diagramas de UML** [En línea] [Citado el: 16 de Marzo de 2011] http://www.programacion.com/articulo/introduccion_a_uml_181/2.
- [31]. **Modelo de datos.** [En línea] [Citado el: 18 de Marzo de 2011] http://www.programacion.com/articulo/modelo_de_datos_178.
- [32]. **Pruebas de caja negra.** [En línea] [Citado el: 19 Marzo del 2011] <http://www.buenastareas.com/ensayos/Pruebas-De-Caja-Negra/1477804.html>.

Bibliografía

1. **Dr. Cerritos Antonio.** Sistema de Información Hospitalaria. 2003 [Citado el: 25 de Octubre del 2010] <http://educacion.salud.gob.mx/cursos/informatica/HIS/his.pdf>.
2. **Figueredo Gonzalez, Yordanis.** Trabajo de diploma Sala de Parto. 2010. [Citado el: 21 de Septiembre del 2010.]
3. **Niveles de Atención de Salud.** [En línea] [Citado el: 18 de Octubre del 2010.] http://www.bvs.cl/php/decsws.php?tree_id=SP2.001.012&lang=es#.
4. **Metis His.** [En línea]. [Citado el: 22 de Octubre del 2010.] <http://www.apraful.com.uy/soluciones/metis-his.html>
5. **SIGHO.** [En línea] 13 de mayo 2006. [Citado el: 22 de Octubre del 2010.] <http://www.sigho.gob.mx/quees.html>.
6. **SALGEN.** Aplicación Informática para la atención de mujeres embarazadas. [En línea] 8 de junio del 2010. [Citado el: 22 de Octubre del 2010] <http://www.juventudrebelde.cu/cuba/2010-06-29/software-beneficia-programa-materno-infantil/>.
7. **Control Salud.** [En línea]. [Citado el: 6 de diciembre del 2010] <http://www.cristian-silva.com.ar/>.
8. **PACIENTES HIS.** [En línea] 22 de Enero de 2009. [Citado el: 2 de Octubre de 2009.] <http://cnt.com.co/PAGINA/index.asp?id=178>.
9. **Arquitectura basada en capas.** [En línea] 29 de mayo del 2009 [Citado el: 18 de Octubre del 2010] http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp#10.
10. **BuenMaster.com.** [En línea] 14 de Agosto de 2007. [Citado el: 20 de Octubre de 2010.] <http://buenmaster.com/?a=536>.
11. **Campos, Beatris Prieto.** Diseño y programación de páginas web [En línea] [Citado el: 17 de Octubre de 2010.] <http://atc.ugr.es/~bprieto/paginas-web8/introduccionx.html>.
12. **García, Alejandro Pérez.** desarrolloweb.com. [En línea] 21 de febrero de 2006. [Citado el: 17 de Octubre de 2010.] <http://www.desarrolloweb.com/articulos/2380.php>.

13. **Garrett, Jesse James.** Maestros del web. [En línea] 11 de Junio de 2005. [Citado el: 17 de Octubre de 2010.] <http://www.maestrosdelweb.com/editorial/ajax/>.
14. **JBoss RichFaces.** [En línea] 2009. [Citado el: 20 de Octubre del 2010] <http://labs.jboss.com/portal/jbossrichfaces/>.
15. **JBoss Ajax4jsf** [En línea] 8 de Septiembre del 2010 [Citado el: 21 de Octubre del 2010.]
 - a. <http://labs.jboss.com/portal/jbossajax4jsf/>
16. **RedHat.** [En línea] 3 de Marzo de 2009. [Citado el: 17 de Octubre de 2010.] http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/seam/Seam_Reference_Guide/A_complete_Seam_application_the_Hotelg_example_e_Seam_UI_control_library.html.
17. **JavaServer Faces y Ajax Parte-1** [En línea] 7 de Agosto del 2007. [Citado el: 21 de Octubre del 2010.] <http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/>
18. **Seam Framework** [En línea] 18 de Enero del 2008 [Citado el: 21 de Octubre del 2010.] <http://seamframework.org/Home/WhySeam>.
19. **Mazza, Ramiro Gómez.** DosIdeas. [En línea] 26 de Mayo de 2009. [Citado el: 17 de Octubre de 2010.] <http://www.dosideas.com/java/592-primeros-pasos-con-drools.html>.
20. **Enrique Rodriguez Lasterra.** [En línea] Introducción a Java Persistence API - Presentation Transcript. 2008 [Citado el: 21 de Octubre del 2010.] <http://www.slideshare.net/jamslug/introduccion-a-java-persistence-api>.
21. **Introducción a Enterprise Java Beans.** [En línea] 2002 [Citado el: 21 de Octubre del 2010.] http://www.javahispano.org/contenidos/es/introduccion_a_enterprise_java_beans.
22. **Suárez González, Héctor.** Manual de Hibernate. [En línea] 22 de Abril del 2003 [Citado el: 26 de Octubre del 2010] <http://www.scribd.com/doc/40059777/Manual-Hibernate>.
23. **PostgresSQL.** [En línea] 29 de agosto del 2010. [Citado el: 21 de Octubre del 2010.] <http://www.scribd.com/doc/36570462/postgreSQL-investigacion>.

24. **Paola.** OSUM- Open Source University Meetup. [En línea] 28 de Septiembre de 2009. [Citado el: 17 de Octubre de 2010.] <http://osum.sun.com/group/insptutr/forum/topics/conceptos-clave-del-lenguaje-6>.
25. **Natura Software.** [En línea] 2007. [Citado el: 17 de Octubre de 2010.] <http://www.naturasoftware.com/main.php?f=tecnologia>.
26. **Itera.** [En línea] 2008. [Citado el: 17 de Octubre de 2010.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.
27. **Mora, Francisco.** UML: Lenguaje. [En línea] 2003. [Citado el: 17 de Octubre de 2010.] <http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>.
28. **Milestone consulting.** [En línea] [Citado el: 17 de Octubre de 2010.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.html>.
29. **Especificaciones de Requerimientos.** [En línea] [Citado el: 10 de Enero de 2011.] <http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>.
30. **Diagramas de UML** [En línea] [Citado el: 16 de Marzo de 2011] http://www.programacion.com/articulo/introduccion_a_uml_181/2.
31. **Modelo de datos.** [En línea] [Citado el: 18 de Marzo de 2011] http://www.programacion.com/articulo/modelo_de_datos_178.
32. **Pruebas de caja negra.** [En línea] [Citado el: 19 Marzo del 2011] <http://www.buenastareas.com/ensayos/Pruebas-De-Caja-Negra/1477804.html>.
33. **Pérez Eguíluz, Javier.** Introducción a AJAX. 7 de junio del 2008.
34. **Farley, Jim.** Practical JBoss® Seam Projects. 2007.
35. **King, Gavin.** Seam - Contextual Components a Framework for Enterprise Java. 2008.
36. **Baver Christian.** Java Persistence with Hibernate. 2009.
37. **Yuan Juntao, Michael.** Jboss seam simplicity and power beyond Java EE. 2008.

38. **Nuñez, Ismael.** Tutorial de Hibernate. 2008.

39. **Kito, Mann.** JAVASERVER FACES IN ACTION. 2007.

40. **Rivera Fernández, Javier.** Modelo de datos. 2009.