

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Módulo Estadísticas del Sistema de
Información Hospitalaria alas HIS**

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

Autores: William Armando Pérez Romera

Kenia Reyes Mosqueda

Tutora: Ing. Kenia Fernández Parra

Ciudad de la Habana, Junio de 2009

“Año del 50 aniversario del triunfo de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.
Para que así conste firmamos la presente a los 24 días del mes de junio del año 2009.

Firma del autor:

William Armando Pérez Romero

Firma de la autora:

Kenia Reyes Mosqueda

Firma del tutora:

Ing. Kenia Fernández Parra

Datos de contacto

Ing. Kenia Fernández Parra

Instructor recién graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor de la Facultad # 7. Ha impartido las asignaturas Matemática Discreta, Gestión de Software, Sistemas de Bases de Datos y Ética Informática. Se desempeñó por un período de 6 meses como líder del área temática productiva Gestión Hospitalaria (GEHOS), actualmente se desempeña como analista en la misma.

Correo electrónico: kfernandezp@uci.cu



Aquel que duda y no investiga, se torna no sólo infeliz, sino también injusto.

Blaise Pascal

Agradecimientos

William....

Primero que todo quisiera agradecerles a mis padres por todo el amor y el apoyo que me han brindado, no sólo a lo largo de estos cinco años sino en todo mi trayecto de vida. Aquí está su regalo.

A mi hermano, por sacar siempre lo mejor de mí y no perder nunca la fe en su hermano mayor.

A mi tutora Kenia y a mi oponente Alain, por la ayuda que supieron brindarme.

A mis compañeros de aula y amigos que siempre estuvieron junto a mí en las buenas y malas e hicieron posible que este sueño se hiciera realidad, especialmente a Franco, chichí, Yusdelis, Liannis, Juan Manuel, Aliuska. En fin para no olvidar a nadie, a todos

Muchas gracias.

Kenia...

Hay una nube de agradecimientos en mi vida, muchos merecen estar en el primer lugar de estas líneas, si escribiera sus nombres, probablemente fueran miles. Pero hay alguien que siempre ha estado a mi lado y que ha hecho invisible la distancia para estar junto a mí, es el Espíritu Santo, el ha hecho posible cosas imposibles de mi vida...le agradezco con todo mi corazón.

A mi papá, por haber trabajado arduamente estos cinco años para darme apoyo,

A mi mamá, por ser muy cariñosa y ayudarme a toda costa,

A mi abuelita por mostrarme desde pequeña el camino a seguir,

A Deivis, por haberme formado docentemente,

A mi familia y amigos por haber estado durante todos los días de mi vida,

A mi tutora Kenia, y mi oponente Alain por ser un apoyo importante en la realización del trabajo...

Dedicatoria

Dedico este trabajo primeramente a Dios por todas las cosas lindas que me ha permitido vivir durante estos cinco años al conocer personas maravillosas a las cuales también dedico mi esfuerzo. A mis padres: Beatriz y Onel, a mis abuelitas: Braulía y Vicenta y mi abuelito: Claudio, a mis tíos especialmente: Eugenia, Israel, Raúl, Luisa, China, y Demetrío, a mis amigos de la vida, especialmente a: Deivís, Mileidís, Leyanís y Adrián.

Kenia Reyes Mosqueda

A mis padres, mi hermano, a mis amistades, y a todos los que confiaron en mí.

William Armando Pérez Romero

Resumen

Las instituciones hospitalarias tienen diferentes departamentos, áreas y servicios, los que en un período de tiempo definido emiten consolidados de la información que gestionan. A pesar de los avances tecnológicos que se han alcanzado, el garantizar una correcta gestión de esta información estadística no ha sido solucionado aun.

Actualmente, existen instituciones que cuentan con sistemas informáticos que son costosos y basados en tecnología propietaria. Además, sólo abarcan las áreas administrativas o contables y excluyen los totales asociados a los departamentos asistenciales, así como las áreas de toma de decisión. Por las razones antes descritas, el objetivo de este trabajo de diploma es desarrollar el módulo de Estadísticas del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

El desarrollo del sistema estuvo guiado por la metodología Rational Unified Process, se utilizó UML como lenguaje de modelado y Visual Paradigm para el modelado visual. En la Base de datos se utilizó PostgreSQL como gestor, y Pgadmin para la administración de la base de datos. La implementación estuvo sustentada sobre el entorno de desarrollo Eclipse unido a Java como lenguaje de programación.

Con el desarrollo del sistema propuesto se espera garantizar el control estadístico del trabajo realizado en la institución, gestionar y alcanzar una utilización óptima de los recursos de salud así como la obtención de datos estadísticos en tiempo real que posibilita realizar estudios científicos que repercutan en una futura toma de decisiones.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1. Conceptos asociados al dominio del problema	5
1.2. Sistemas automatizados existentes vinculados al campo de acción	6
1.3. Tendencias y tecnologías actuales a considerar	10
Capítulo 2: Características del sistema	24
2.1. Objeto de estudio.....	24
2.2. Objeto de automatización	27
2.3. Información que se maneja.....	29
2.4. Propuesta del sistema	30
2.5. Modelo de Negocio.....	30
2.6. Especificación de los requerimientos de software	40
2.7. Modelo de Casos de Uso del Sistema	44
Capítulo 3: Análisis y Diseño.....	53
3.1. Descripción de la arquitectura.....	53
3.2. Modelo de diseño	56
Capítulo 4: Implementación.....	77
4.1. Modelo de Datos.....	77
4.2. Implementación	87
Conclusiones	95
Recomendaciones	96

Referencias bibliográficas	97
Bibliografía	99
Anexos.....	101
Glosario de términos	105

Introducción

El ser humano desde sus primeros años de existencia y con el afán de mejorar su calidad de vida, ha creado nuevas herramientas que le permitan satisfacer sus necesidades. Para ello ha desarrollado instrumentos que le facilitan el trabajo y hacen su vida mucho más fácil y sencilla. Entre los mejores inventos de todos los tiempos y uno de los que mayores beneficios aporta a la sociedad, es la computadora. Esta surge como resultado de la evolución constante de la tecnología y motivada por las necesidades de manejar la información.

El surgimiento de las computadoras no fue un hecho aislado, ya que la aparición de la primera no fue producto de las ideas o ambición de un científico, sino que se originó de la evolución y desarrollo de los medios de cálculo. La computación se ha desarrollado en paralelo con las crecientes necesidades del hombre para realizar los cálculos de forma más rápida y con mayor precisión. Por lo que se pueden encontrar sus orígenes y antecedentes miles de años atrás. Los ordenadores están dotados de una gran capacidad de trabajo, que posibilitan interpretar y procesar de manera muy rápida y eficiente la información.

Todo lo anterior se logra mediante la combinación de hardware y software (programas y sistemas) que hacen de la computadora un instrumento de vital importancia, que se ha insertado en las diferentes actividades de la vida del hombre.

Uno de los sistemas, que ha logrado su inserción dentro de la sociedad; específicamente en el área de la salud, son los Sistemas de Información Hospitalaria (SIH o HIS por sus siglas en inglés). Estos se han convertido en aplicaciones de alto valor para las instituciones hospitalarias, que posibilitan una gestión de la información en los distintos departamentos que conforman una institución hospitalaria.

Su impacto ha sido positivo ya que persiguen elevar la calidad de los servicios prestados y aplicar la información obtenida a las diferentes áreas de la investigación, la docencia, la administración, la clínica, por lo que contribuyen a minimizar los costos y elevar la calidad de atención al paciente.

Actualmente el proceso de garantizar una correcta gestión de la información, no ha sido solucionado correctamente, a pesar de todos los avances tecnológicos que se han alcanzado. Cada día es mayor la cantidad de información que se necesita procesar, debido a que los sistemas que el hombre utiliza para controlar los datos generados por algunos procesos de gestión y control se han hecho más complejos.

Una de las áreas de vital importancia, donde se genera la información del entorno hospitalario que tiene como principal función obtener datos tangibles acerca de los logros o dificultades existentes en los departamentos del hospital, es el área de las estadísticas. Este departamento dentro de la entidad hospitalaria, es el encargado de la gestión de los resúmenes estadísticos. Los mismos permiten la mejora de los procesos y una pronta toma de decisiones efectivas de acuerdo a los indicadores resultantes de cada departamento, en los resúmenes que pueden ser obtenidos.

Actualmente la mayoría de las instituciones hospitalarias no cuentan con un sistema estadístico capaz de gestionar uniformemente la información generada en las distintas áreas del hospital, integrado en un sistema único, que relacione toda la información del paciente. En las áreas de la salud resulta engorroso gestionar sus resultados estadísticos. Estas no cuentan con una información centralizada de las distintas áreas para poder emitir un consolidado. Los resúmenes emitidos, dependen de que la información sea entregada en tiempo por parte del personal que labora en los departamentos del hospital.

Es necesario que el proceso de actualización de la información se realice eficientemente, para garantizar una correcta sincronización de la información que reflejan. El control de las estadísticas, está totalmente influenciado por el personal que labora en el hospital, en ocasiones el flujo de datos puede afectar el trabajo del personal pues se incrementan las atenciones durante el trabajo diario y en otras, falta de preparación sobre las nuevas tecnologías.

Los informes estadísticos se realizan de forma manual, lo que trae como consecuencia que ocurran errores en su recolección, puedan deteriorarse o perderse. Estos indican el comportamiento de áreas como: Laboratorio, Anatomía Patológica, Admisión, Consulta externa, Hospitalización, Bloque Quirúrgico y otras. Representan en un pequeño por ciento la cantidad de camas disponibles por departamento, cantidad de admisiones, cantidad de transferencias realizadas, registros con la cantidad de pacientes ingresados y egresados, cirugías que han sido canceladas, entre otras. Los informes contienen información consolidada de cualquier departamento del hospital, en dependencia de las variables a ser representadas en el reporte.

Lo anteriormente descrito evidencia el incremento en el volumen de páginas de los registros, además de no posibilitar la generación de gráficos dinámicos, que permitan a los directivos y trabajadores, tener un consolidado actualizado y al alcance de todos. Lo que en ocasiones, imposibilita la generación de estadísticas confiables y actualizadas.

Este elevado número de inconvenientes y situaciones en los hospitales repercuten en el correcto funcionamiento de la entidad. Lo que ocasiona que no existan informes con datos estadísticos que permitan hacer estudios y análisis de la población atendida, enfermedades más comunes y otros reportes. Además, la carencia de información estadística en tiempo real a los niveles de departamento, sala, salones quirúrgicos y consultas a nivel de hospital, posibilita la demora de la información por los niveles de gerencia administrativa.

Después de realizar un profundo análisis de la situación actual en los centros hospitalarios específicamente en el área de Estadísticas, se define como **problema a resolver**: ¿Cómo facilitar la gestión de información relacionada con los procesos en el área de Estadísticas de las instituciones hospitalarias?

El **objeto de estudio** lo constituye el proceso de gestión de información en las instituciones hospitalarias, cuyo **campo de acción** se enmarca en el proceso de gestión de información en el área de Estadísticas de las instituciones hospitalarias.

Basado en esa idea, se define como **objetivo general** del presente trabajo: desarrollar el módulo de Estadísticas del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **tareas**:

1. Evaluar las tendencias actuales en el mundo de los Sistemas de Información Hospitalaria relacionado con los procesos estadísticos.
2. Analizar los procesos de negocio asociados al área de Estadísticas de los hospitales.
3. Asimilar la arquitectura definida por el Área Temática Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Analizar la herramienta definida para la generación de reportes.
5. Obtener mediante el Proceso Unificado de Desarrollo el Modelo de Diseño a partir del modelo de análisis existente.
6. Implementar los reportes propuestos del módulo Estadísticas del Sistema de Información Hospitalaria Alas HIS.

El desarrollo del módulo de Estadísticas del Sistema de Información Hospitalaria alas HIS, proporcionará un grupo de beneficios entre los que pueden ser mencionados los siguientes:

- ❖ Poner a disposición del profesional médico herramientas de gestión clínica y gestión administrativa que den respuesta a sus necesidades reales en el área de Estadísticas.
- ❖ Determinar el flujo de pacientes que transita por los distintos departamentos de la institución hospitalaria, evidenciándose el comportamiento de los ingresos, egresos y transferencias realizadas.
- ❖ Llevar el control estadístico del trabajo realizado en las distintas áreas, entre las que se destacan Consulta externa y Emergencias.
- ❖ Mantener un control de gestión y utilización óptima de recursos de salud.
- ❖ Garantizar que toda la información generada durante el trabajo en los hospitales sea almacenada en bases de datos que permitan hacer uso de ellas en cualquier momento.
- ❖ Obtención de datos estadísticos en tiempo real para la realización de estudios científicos acerca de las enfermedades más comunes por áreas residenciales o departamentos de la institución hospitalaria.
- ❖ Elaborar partes informativos, ayudados de gráficos estadísticos.

El documento está compuesto por cuatro capítulos, el primero de ellos, **Fundamentación teórica**, en el cual se analizan los conceptos y sistemas relacionados con los procesos de gestión hospitalaria, además de ubicar al lector en el ambiente de desarrollo, justificándose el uso de tecnologías, metodologías y herramientas para el desarrollo de la solución propuesta.

Seguidamente, el capítulo **Características del sistema**, en el cual se plasma el marco conceptual asociado a la información que será manipulada por el sistema, llegándose a un acuerdo sobre las funcionalidades, requerimientos deseados y el objeto de automatización.

El tercer capítulo **Diseño del sistema** se centra en la modelación detallada y la construcción de la estructura de la aplicación.

En el cuarto y último capítulo **Implementación**, se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión eficiente de los requerimientos previamente identificados.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se abordan los conceptos fundamentales que posibilitan un mejor entendimiento de los procesos vinculados al campo de acción. De igual forma se analizan los diferentes sistemas de información hospitalaria a nivel nacional e internacional. Se incluye la fundamentación de la selección de un grupo de herramientas, tecnologías y metodologías utilizadas para el desarrollo de este tipo de sistemas.

1.1. Conceptos asociados al dominio del problema

Las instituciones hospitalarias incluyen un departamento de estadísticas, el mismo se encarga de procesar la información del hospital y emitir resúmenes estadísticos. Estos *resúmenes o informes* constituyen la información sintetizada de cada servicio del hospital, de vital importancia para una futura toma de decisiones de la misma. Estos informes estadísticos incluyen el *movimiento hospitalario del hospital* contemplándose dentro de este toda la información relacionada con el ingreso del paciente ya sea electivo o de urgencia, reflejándose la hora, el día, la sala y el servicio donde fue ingresado. De igual forma se mantiene el control de los egresos y traslados hechos en el hospital.

Relacionado al movimiento hospitalario, se contemplan un grupo de indicadores. Los cuales se clasifican en:

Índice ocupacional: Expresa qué porcentaje de las camas fue utilizado en el período.

Promedio de estancia: Es el número de días de asistencia hospitalaria que como promedio recibió cada paciente en un período de tiempo.

Índice de rotación: Se interpreta como el número de personas que pasó por una cama durante un período.

Intervalo de sustitución: Es el tiempo promedio que una cama permanece desocupada entre el egreso de un paciente y el ingreso de otro.

Días de ocupación de la cama: Expresa el número de días que como promedio estuvo ocupada una cama durante el año.

Morbilidad hospitalaria: representa el estudio de los efectos de una enfermedad en una población en el sentido de la proporción de personas que enferman en un sitio y tiempo determinado.

Mortalidad hospitalaria: Es la relación entre egresos fallecidos del hospital y el total de ingresos.

Tasa de mortalidad neta: Este indicador mide la calidad de la atención hospitalaria.

Mortalidad bruta: Es el indicador que mide la acción de la muerte sobre la población hospitalaria e indica de esa población egresada cuantos fueron fallecidos en total.

Promedio de camas reales: Se interpreta con el número de personas que pasan por una cama real durante un período de tiempo determinado. Una cama real es aquella que se encuentra instalada realmente y disponible las 24 horas del día para recibir una persona esté ocupada o no.

Vinculado a la generación de resúmenes estadísticos se encuentra el *área de Consulta externa*, de la cual se lleva el control diario de la cantidad de casos vistos ya sean programados o no programados, dígame por programados, los pacientes que acuden a la consulta con una cita previamente asignada y los no programados aquellos pacientes que son atendidos y no poseen una cita. La cita no es más que la asignación de turnos a determinadas consultas.

Se define además la información relacionada al *área de emergencias*, procesándose la información de los pacientes vistos de medicina y cirugía, las remisiones, ingresos y casos vistos en observación. Los pacientes vistos en medicina son aquellos atendidos normalmente en la consulta y los vistos en cirugía son los que directamente pasan a dicha área.

1.2. Sistemas automatizados existentes vinculados al campo de acción

Existe un amplio grupo de empresas especializadas en la concepción, desarrollo e implantación de soluciones de software vinculadas al área de la salud, las que tienen un vínculo estrecho con las instituciones hospitalarias. Dentro de dichas instituciones se ha ampliado el marco de desarrollo hacia cada una de las áreas, servicios y departamentos que estas entidades incluyen, específicamente el departamento de estadísticas, en aras de optimizar los recursos humanos, materiales e incrementar la calidad de la atención en los servicios de la salud.

1.2.1. Sistemas de Información Hospitalaria desarrollados internacionalmente

A nivel internacional existe un grupo de Sistemas de Información Hospitalaria que incorporan un módulo para la gestión de la información estadística de las instituciones hospitalarias, entre los que se destacan:

Kewan – Cosmosalud, desarrollado por la empresa Dimensión Informática S.L, una de las entidades españolas con mayor experiencia en el desarrollo, mantenimiento e implantación de soluciones en el ámbito de la salud. Esta aplicación posibilita: obtener indicadores de gestión, pero no genera otros reportes estadísticos para la gestión de áreas básicas como admisión, archivo y facturación.

Fue diseñado para ser utilizado en cualquier tipo de organización ya sea hospitales públicos o privados, de atención primaria o atención especializada. Es un sistema multicentro y multidioma, que integra todo tipo de información, imágenes, voz, video, etc. Contiene una interfaz gráfica que permite visualizar la ocupación de quirófanos, agendas, entre otras funcionalidades. Además de incorporar las últimas tecnologías como la firma electrónica, reconocimiento de huella, reconocimiento de voz, códigos de barras, facturación electrónica, servicio de mensajes cortos (SMS del inglés Short Message Service). (1)

X-HIS desarrollado por la empresa española ISOFT, elimina la navegación por menús e incorpora una interfaz que facilita el uso de la aplicación a cualquier tipo de usuario. Este software brinda un browser asistencial que le permite al usuario navegar por todo el historial clínico. Tiene una interfaz parecida a la del Outlook. Es multiplataforma, corre sobre los principales sistemas operativos como: Unix, Windows, Linux y otros, tiene independencia de base datos de MS SQL server, Oracle, Informix, Adaptive server y permite la conexión a través de redes de áreas locales, wireless, correo electrónico, fax, web etc.

Este software, permite crear un archivo de historias clínicas de forma digital para gestionar de manera informatizada el historial de los pacientes, además de que mantiene los informes clínicos ordenados y clasificados en expedientes clínicos electrónicos. Sin embargo, aunque cuenta con estas ventajas, no tiene un módulo que se especialice en la generación de los reportes estadísticos que sean de interés para los hospitales donde se encuentre instalado dicho sistema, pues sólo aporta datos relacionados con la facturación. (2)

Sistema de Información para la Gerencia Hospitalaria (**SIGHO**) utilizado en algunos de los hospitales de México. Es un software basado en la Norma Oficial Mexicana NOM-168-SSA1-1998 referente al

resguardo y uso del expediente clínico electrónico para facilitar las actividades de gerencia dentro del hospital. (3)

Se apoya en estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC. Está conformado por 14 módulos fundamentales, algunos de ellos son: Agenda, Consulta Externa, Admisión, Hospitalización, Urgencias o Enfermería, Toco-cirugía, Cirugía, Trabajo Social e Imagenología. Para su desarrollo se utilizó como lenguaje de programación ASP.NET y el Framework Microsoft .NET, como Gestor de Base de Datos SQL Server 2000 ya que permite brindarle información actualizada al usuario mediante consultas y procedimientos.

La principal información que se obtiene a través del SIGHO corresponde a la Hoja diaria de consulta externa con diagnósticos codificados en base a la Clasificación Internacional de Enfermedades CIE-10, los formatos intermedios del Sistema de Información Hospitalaria, el concentrado semanal del SUIVE, los tarjeteros de diabetes mellitus e hipertensión arterial, la Historia Clínica Prenatal Simplificada (HCPS), el seguimiento de los programas de Línea de Vida, la coordinación en el Sistema de Referencia y Contrarreferencia, así como el manejo en el registro de los resultados de Laboratorio y Gabinete e inventario de medicamentos.

Sin embargo, aunque cuenta con ciertas ventajas, a partir de la información obtenida proveniente de cada una de las áreas de donde se nutre de los datos de interés, no realiza la generación de reportes estadísticos provenientes de cada uno de los departamentos, o de la institución hospitalaria en general. Lo cual supone que no se puedan realizar estudios científicos de las enfermedades más comunes.

Open HIS, Sistema de información Hospitalaria desarrollado por el Grupo BioLinux, el cual es distribuido libremente bajo la Licencia Pública General versión 2.0 (GPL 2.0) de la Fundación de Software Libre. Este un sistema modular, escalable y cumple con los estándares abiertos predefinidos para el correcto funcionamiento de un sistema hospitalario. Fue realizado con PHP como lenguaje de programación, AJAX para la comunicación con el servidor, y MySQL y/o PostgreSQL como sistemas gestores de base de datos. Sin embargo este sistema es poco abundante en cuanto a los servicios que brinda en pos de la atención del paciente, pues sólo presenta unas pocas opciones encaminadas a informatizar algunos servicios, que no son suficientes para hacerlo totalmente operativo. (4)

Health3000®, es un sistema de información para el manejo y administración de Hospitales y Clínicas. Producido por la empresa Health Care International Services, la cual está establecida en la Ciudad de Miami, desde 1993, y fue fundada con la finalidad de abastecer a Laboratorios Clínicos, de todas sus necesidades en insumos y equipos para su normal funcionamiento. Este producto ha sido desarrollado

con una arquitectura abierta, maneja diferentes módulos, tales como farmacia, laboratorio, radiología, consulta externa, citas, admisión y descarga de pacientes, transferencia de datos a contabilidad entre otros. (5)

Neuron, fue producido por la empresa colombiana Neuron Ltda. Es una aplicación web que organiza el trabajo en 5 módulos que engloban a los demás, los mismos son: operación y transacción, apoyo a operaciones, administración, gestión y web. Específicamente en el módulo de Cirugía de manera especial garantizan la información para la facturación, estadística, programación de cirugías, e historia clínica. El módulo de Programación de cirugías controla la programación de cirugías y permite la integración con el módulo de Admisiones, Facturación, y Estadística.

En el módulo de Administración, de este sistema, se tienen en cuenta los módulos de Administración de información estadística, Administración de propiedad planta y equipo, Control de elementos de consumo, Administración de glosas, Administración de tratamientos odontológicos, Administración de epicrisis, Administración de atención de urgencias, Administración de farmacia, Administración de laboratorios, Tesorería, Facturación, Cartera, Administración de autorización de servicios, Promoción y prevención, Cartas de compromiso, Contratación, Vacunación, Auditoría médica, Prestadores, Nómina orden de prestación de servicios y Nómina planta.

1.2.2. Sistemas de Información Hospitalaria desarrollados a nivel nacional

A nivel nacional se encuentran un pequeño número de sistemas, que aunque no en su totalidad fueron creados para entidades hospitalarias, incluyen funcionalidades vinculadas al objetivo que se analiza, entre ellos se encuentran:

Galen hospital, producido por la empresa cubana Softel, entidad que cuenta con experiencia en el desarrollo de aplicaciones informáticas. El sistema está principalmente dirigido a la gestión de los medios de diagnóstico y consultas.

Este software facilita la solicitud de exámenes, y evaluación de los resultados además de brindar otras funcionalidades tales como: establecer los médicos de guardia, agregar consulta realizadas en el hospital, buscar la historia clínica de un paciente, emitir un resumen de los casos atendidos entre otras.

La aplicación gestiona la información requerida para la actividad gerencial a todos los niveles y facilita la elaboración de reportes estadísticos. Este proyecto incluye varias áreas como la de registro de pacientes, hospitalización, gestión de medios de diagnósticos y consulta, además que incluye conexión

con equipos de diagnóstico y auto-analizadores. También permite la emisión de informes de resultados y estadísticas.

Fue desarrollado en plataforma Windows (Windows NT y Windows 98) con 32 bits, presenta una configuración Cliente/Servidor y usa el gestor de bases de datos relacional SQL Server 7.0 como repositorio de la información. Para la construcción de la base de datos se utilizó el Sistema Gestor de Bases de datos Relacional SQL Server 7.0.

Aunque este es uno de los sistemas más completos, en el área de estadísticas sólo se emite reportes de cirugía, morbilidad y movimiento hospitalario, y se deja fuera importantes informaciones provenientes de otras áreas de la institución hospitalaria. Estas áreas, están relacionadas con la atención al paciente e influyen en la realización de estudios por parte del personal de salud.

Galenlab, fue producido por la empresa cubana Softel, con el objetivo de ser utilizado por los especialistas para la gestión de pacientes, gestión de turnos, controlar los resultados de exámenes de laboratorio, permitir el enlace de los laboratorios con los autoanalizadores y la administración de autoanalizadores. Está dirigido a la gestión de los medios de diagnóstico y consultas, facilita la solicitud de exámenes, registro y evaluación de los resultados obtenidos y la generación de información estadística. Además de que presenta varios componentes como son: Gestión de servicios, Consulta, Agenda, Autoanalizadores, Imágenes y trazados gráficos. (6)

Aunque este software genera información estadística, sólo se centra en los resultados de los exámenes de los pacientes, y deja a un lado importantes informaciones relacionadas con los servicios de atención al paciente, algunas de ellas son: las causas de cancelación de intervenciones quirúrgicas, y el comportamiento de la población atendida por la institución ya sea la morbilidad o mortalidad por patología.

1.3. Tendencias y tecnologías actuales a considerar

El crecimiento en la industria de desarrollo de software impulsa cada vez más el desarrollo de nuevas tecnologías y metodologías que se utilizan con este propósito. Actualmente existen a nivel mundial herramientas que permiten la realización de grandes sistemas en tiempos relativamente cortos.

Una de las principales acciones a desarrollar por cualquier empresa desarrolladora de software es definir el conjunto de herramientas y tecnologías de desarrollo de software a utilizar, lo cual constituye el resultado de la investigación realizada previo a la selección.

Como parte de dicha investigación son analizadas las características, ventajas y desventajas que poseen cada una de estas tecnologías, para determinar la factibilidad de su uso durante el desarrollo del sistema.

A continuación se presenta el resumen realizado durante la investigación, y la propuesta tecnológica realizada para el desarrollo del sistema.

1.3.1. Sistemas distribuidos. Modelo Cliente Servidor

Un sistema distribuido es una colección de computadores autónomos conectados por una red, y con el software distribuido adecuadamente para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar todas sus funcionalidades. (7)

Modelo Cliente Servidor

El Modelo Cliente Servidor es una arquitectura donde la capacidad de proceso está repartida en dos o más computadoras que cooperan entre sí. Los clientes se convierten en la parte activa y realizan peticiones al servidor, el cual es la parte pasiva, encargándose de atender, procesar y dar respuestas a estas peticiones. (8)

Ventajas que presenta la arquitectura Cliente/Servidor

- ❖ Facilita la integración entre sistemas diferentes, y permite que las máquinas ya existentes puedan ser utilizadas con el uso de interfaces más amigables al usuario.
- ❖ Facilita el suministro de información a los usuarios, ya que proporciona una mayor consistencia a la información de la empresa al contar con un control centralizado de los elementos compartidos, además de que facilita la construcción de interfaces gráficas interactivas.
- ❖ Permite utilizar máquinas considerablemente menos costosas que las requeridas por una solución centralizada debido a la disminución de los precios de hardware.

Desventajas de la arquitectura Cliente/Servidor

- ❖ El mantenimiento de los sistemas se hace difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- ❖ Los clientes y los servidores deben utilizar el mismo mecanismo (por ejemplo sockets o RPC), lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas.
- ❖ La congestión en la red puede causar un mal desempeño del sistema.

1.3.2. Patrones de arquitectura y diseño

Los patrones del diseño tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. (9) Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes. Existen muchas formas de implementar patrones de diseño. Los detalles de las implementaciones son llamadas estrategias.

Fue por los años 1994, que apareció el libro "Design Patterns: Elements of Reusable Object Oriented Software" escrito por Gang of Four, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Luego de la publicación de ese libro fue que empezó a difundirse con más fuerza la idea de patrones de diseño. Ellos recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos.

El grupo de GoF clasificó los patrones en 3 grandes categorías: creacionales, estructurales y de comportamiento.

- Creacionales: Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- Estructurales: Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.
- Comportamiento: Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Patrones de la arquitectura J2EE

Con la aparición de la arquitectura J2EE, también vino un nuevo catálogo de patrones de diseño. Desde que J2EE es una arquitectura que involucra otras, incluyendo servlets, JavaServer Pages, Enterprise Java Beans, y otras, tiene su propio conjunto de patrones específicos para diferentes aplicaciones empresariales, que les permiten por ejemplo: Alta disponibilidad, Redundancia a Fallas, Alto desempeño, y Balanceo de Cargas.

Existen 15 patrones J2EE que están divididos en 3 de las 5 capas de la arquitectura: presentación, negocios e integración.

Capa de Presentación

- ❖ Decorating Filter / Intercepting Filter: Un objeto que está entre el cliente y los componentes Web. Este procesa las peticiones y las respuestas.

- ❖ Front Controller/ Front Component: Un objeto que acepta todos los requerimientos de un cliente y los direcciona a manejadores apropiados.
- ❖ View Helper: Un objeto helper que encapsula la lógica de acceso a datos en beneficio de los componentes de la presentación.
- ❖ Composite view: Un objeto vista que está compuesto de otros objetos vista. Por ejemplo, una página JSP que incluye otras páginas de ese tipo y HTML usan la directiva include o el action include.
- ❖ Service To Worker: Es como el patrón de diseño Modelo Vista Controlador que ayuda en el manejo de la vista.
- ❖ Dispatcher View: realiza muy poco trabajo en la vista. El manejo de esta, es realizado por sus mismos componentes.

Capa de Negocio

- ❖ Business Delegate: Un objeto que reside en la capa de presentación, en beneficio de los otros componentes llama a métodos remotos en los objetos de la capa de negocios.
- ❖ Value Object/ Data Transfer Object/ Replicate Object: Un objeto para la transferencia de datos sobre la red.
- ❖ Session Façade/ Session Entity Façade/ Distributed Façade: para encapsular la complejidad de las interacciones entre los objetos del negocio y participantes en un flujo de trabajo, para manejar los objetos del negocio, lo que proporciona un servicio de acceso uniforme a los clientes.
- ❖ Aggregate Entity: Para la agregación entre entidades.
- ❖ Value Object Assembler: Un objeto que reside en la capa de negocios y crea otros objetos cuando es requerido.
- ❖ Value List Handler/ Page-by-Page Iterator/ Paged List: Es un objeto que maneja la ejecución de consultas SQL, caché y procesamiento del resultado.
- ❖ Service Locator: Consiste en utilizar un objeto para abstraer y ocultar las complejidades de la creación del contexto inicial, de búsqueda y recreación de objetos, y reduce la complejidad del código, lo que proporciona un punto de control.

Capa de Integración

- ❖ Data Access Object Service Activator: Consiste en utilizar un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos.

- ❖ Service Activator: Se utiliza para recibir peticiones y mensajes asíncronos de los clientes. Cuando se recibe un mensaje, se invocan a los métodos de los componentes de negocio necesarios para cumplir la petición de forma asíncrona.

1.3.3. Tecnologías existentes

Java, lenguaje de programación

Java es un lenguaje de programación orientado a objeto, el cual contribuye al desarrollo de software, permite realizar cualquier tipo de programa. Está compuesto básicamente por 2 elementos:

- ❖ Plataforma: la máquina virtual de Java
- ❖ El lenguaje: desarrollado por Sun Microsystems a principios de los años 90. La sintaxis es muy parecida a la utilizada en C y C++, aunque presenta un modelo de objetos más simple y elimina herramientas de bajo nivel que suelen inducir muchos errores, como la manipulación directa de punteros o memoria.

Este lenguaje se caracteriza por ser:

- ❖ Compilado: Genera ficheros de clases compiladas, pero estas son en realidad interpretadas por la máquina virtual de java. Dicha máquina virtual es quien mantiene el control sobre las clases que se ejecutan.
- ❖ Multiplataforma: El mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual de java.
- ❖ Seguro: La máquina virtual, al ejecutar el código, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.
- ❖ Distribuido: Fue construido con extensas capacidades de interconexión TCP/IP. Tiene librerías de rutinas para acceder e interactuar con protocolos como http y ftp, lo que permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Eclipse como herramienta de desarrollo

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que se llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo (IDE), como el entorno de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se integra como parte de Eclipse (y que son usados también para

desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Eclipse fue desarrollado originalmente por IBM y es ahora desarrollado por la Fundación Eclipse, una organización independiente que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. Presenta una arquitectura basada en plug-ins que le permite integrar diversos lenguajes sobre un mismo IDE, además es soportado por los principales sistemas operativos como Linux, Windows, Solaris 8 (SPARC/GTK 2) y Mac OSX-Mac/Carbon.

La última versión Eclipse Europa, está compuesta por un total de 21 subproyectos que le permiten adaptarse a las necesidades de cualquier programador. Esta versión incorpora algunas novedades entre las que se encuentra una mejora en el framework, para el desarrollo de interfaces para otros lenguajes interpretados, además facilita el desarrollo de aplicaciones con MinGW en Windows o un editor visual para HTML y JSP.

Visual Paradigm como herramienta de modelado

Visual Paradigm, es una herramienta profesional que utiliza el Lenguaje Unificado de Modelado (UML, como sus siglas en inglés lo indican), la cual soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación, pruebas y despliegue. Esta importante herramienta ampliamente utilizada para el modelado, agiliza el proceso de construcción de aplicaciones de calidad a un menor coste. (10)

Visual Paradigm se puede integrar fácilmente con varios IDEs, entre ellos Eclipse y NetBeans. Posibilita la generación de código desde diagramas y la generación de documentos en varios formatos como HTML, Microsoft Word y PDF. De igual forma facilita la generación de código y la ingeniería inversa en lenguajes como Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python.

JasperReports e iReport, para la generación de reportes

JasperReports es una librería de clases de Java de código abierto diseñada para facilitar la creación de reportes en las aplicaciones Java. Esta puede ser utilizada tanto en aplicaciones web como de escritorio, la misma posibilita incluir en los reportes imágenes y gráficos. Esto permite que los datos se puedan presentar de manera textual o a través de gráficos en dependencia de las necesidades del usuario. (11)

Entre sus principales ventajas se encuentran:

- ❖ Permite una diagramación flexible de los reportes, lo cual posibilita que estos se puedan dividir en secciones opcionales como son título, encabezado y detalles.

- ❖ Posibilita la creación de sub-reportes dentro de reportes facilitando el diseño de los mismos y ayudando a que la información esté mejor organizada.
- ❖ Permite exportar los reportes a una multitud de formatos como PDF, XLS, RTF, HTML, XML, CVS y texto plano.
- ❖ Se integra con JFreeChart que es una librería libre para la generación de todo tipo de gráficas.

Para obtener un reporte se debe definir primeramente el archivo XML, el cual contendrá las particularidades del reporte. En este se especifica la estructura del informe, definiendo la organización de los datos en el mismo.

Luego para cada reporte se implementa una clase java la cual es la encargada de realizar las consultas a la base de datos asignando los valores a este reporte. La salida de esta puede ser en diferentes formatos tales como: PDF, XML, HTML, CSV, XLS, RTF y TXT.

Para definir los XML de los reportes fue utilizada la herramienta iReport, esta es un editor gráfico implementado en java perteneciente al ámbito del software libre, el cual se integra perfectamente con el JasperReports. La misma permite corregir visualmente informes complejos, caracterizándose por:

(12)

- ❖ Manejar la mayoría de las etiquetas de JasperReports
- ❖ Permitir utilizar gráficos estadísticos, imágenes, subreportes y crosstab en la creación de reportes.
- ❖ Los reportes pueden ser exportados a PDF, RTF, XML, XLS, CSV, HTML.

Concepción arquitectónica

La arquitectura en tres capas es un estilo de programación que se basa en la división en el nivel de acceso a datos, nivel de lógica de negocio y nivel de presentación o aplicación. Esta presenta varias ventajas (13):

- ❖ El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto permite reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- ❖ La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- ❖ Al separar la aplicación de la base de datos, se hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.

- ❖ La interfaz del usuario es separada de la aplicación, esto libera de gran procesamiento a la estación de trabajo.

Capa de Presentación

Esta es la capa del sistema con la que interactúa el usuario, brindándole la información que el mismo necesita. Esta capa se comunica únicamente con la capa de negocio y es conocida como interfaz gráfica, su característica principal es la de ser amigable para el usuario.

En esta capa se utiliza Java Server Faces (JSF), framework para aplicaciones Java basadas en servicios web, el cual simplifica el desarrollo de interfaces de usuario en aplicaciones Java. JSF usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas. Además, cuenta con RichFaces, una rica biblioteca de componentes que posee un avanzado marco para integrar fácilmente capacidades AJAX en el desarrollo de aplicaciones de negocios.

Para comprender mejor las ventajas que proporciona AJAX (Asynchronous JavaScript And XML), que por consiguiente son mejoras con las que cuenta JSF, se puede afirmar que la misma es un conjunto de tecnologías de desarrollo Web para crear aplicaciones interactivas sin la necesidad de cargar la página completamente entre una solicitud y otra. Las páginas se ejecutan en el navegador, y mantienen comunicación asíncrona con el servidor en segundo plano mediante un canal de conexión, debido a que la página Web no se refresca completamente, sino que solamente se acceden a elementos de la misma mediante JavaScript u otro lenguaje del lado del cliente, que permita ganar en rapidez y eficiencia.

Capa de Negocio

Es la capa donde se reciben las peticiones del usuario y se envían las respuestas luego de procesarlas, es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para acceder a los datos.

En esta capa se utiliza el framework JBoss Seam, el cual está integrado por un conjunto de elementos que le permiten controlar el acceso a datos, la lógica del negocio y la interfaz de usuario. Seam es un framework Open Source desarrollado por la empresa Jboss (también propietaria de otras tecnologías populares en el mundo Java como Hibernate), con el fin de unir diferentes tecnologías y estándares de Java en un solo framework.

Seam permite usar EJB3 (Enterprise Java Beans en su versión 3)+JSF (Java Server face) de una forma muy sencilla, además de añadirle herramientas útiles para el desarrollo de aplicaciones Web. Todo esto está unido en un solo framework bien acoplado y basado en estándares ampliamente utilizados y probados. Sus principales ventajas son:

- ❖ Permite agregar anotaciones propias a los objetos Entidad y Session de EJB, logra con esto escribir menos código Java y XML
- ❖ Permite hacer validaciones en los POJOs (Plain Object Java), así como manejar directamente la lógica de la aplicación y de negocios desde las secciones beans.
- ❖ Se integra perfectamente con otros frameworks como: RichFaces, ICEFaces(los cuales soportan Ajax) MyFaces, Hibernate y Spring.

JBoss Application Server

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java pero, puede ser utilizado en cualquier sistema operativo que soporte java, implementa todo el paquete de servicios de J2EE. (14)

JBoss fue el primer servidor de aplicaciones de código abierto disponible en el mercado, ofrece una plataforma de alto rendimiento para aplicaciones de negocios. Combina una arquitectura orientada a servicios con una licencia de código abierto, presenta la ventaja de ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia.

Principales características de JBoss:

- ❖ Producto de licencia de código abierto sin coste adicional.
- ❖ Cumple los estándares.
- ❖ Confiable a nivel de empresa
- ❖ Incrustable, orientado a arquitectura de servicios.
- ❖ Flexibilidad consistente
- ❖ Servicios del middleware para cualquier objeto de Java
- ❖ Soporte completo para JMX

Jboss tools

Jboss tools es una colección de plugins que le son añadidos al IDE, los cuales le incorporan una serie de funcionalidades, además de poder ser añadidos a diferentes servidores de aplicación. Posee

un editor gráfico para la configuración de archivos Seam y soporta la realización de pruebas de integración de Seam desde el Eclipse.

Capa de Datos

La capa de datos es donde están almacenados los datos persistentes de la aplicación, es la encargada del almacenamiento y manipulación en la base de datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Una base de datos (BD) se puede definir como el conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. Es decir, una BD puede considerarse una colección de datos variables en el tiempo.

El software que permite la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina Sistema de gestión de bases de datos (SGBD). El objetivo fundamental de estos sistemas consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de manera que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

PostgreSQL

PostgreSQL es un sistema de base de datos relacional perteneciente al ámbito del software libre que se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros. (15)

Las principales mejoras que PostgreSQL incluye son:

- ❖ La sustitución de los bloqueos de tabla por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite realizar copias de seguridad al instante, mientras la base de datos permanece disponible para consultas.
- ❖ La mejora de los tipos internos, incluyendo nuevos tipos fecha/hora de rango y soporte para tipos geométricos adicionales.

Además se utiliza Hibernate para mapear las tablas de la base de datos con las entidades del negocio. Permite así diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos de una manera muy rápida y optimizada. El mismo brinda la posibilidad de generar bases de dato en cualquiera de los entornos soportados: Oracle, MySql; soporta la generación automática de claves primarias y realiza la persistencia de forma transparente sin procesamiento de código de bytes. (16)

1.3.4. Metodologías de desarrollo

Una metodología es un proceso utilizado en el desarrollo de un software para definir Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Estas no existen universalmente, pues las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo del inglés RUP, es un proceso de software genérico que puede ser utilizado para una gran variedad de sistemas de software, así como diferentes áreas de aplicación, tipos de organizaciones, niveles de competencia y tamaños de proyectos. (17) Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo, asegura la producción de software de muy alta calidad y permite satisfacer las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

En el Proceso Unificado de Desarrollo las actividades se agrupan en grupos lógicos para lo cual se definieron 9 flujos de trabajo principales los cuales son: Modelado del negocio, Requerimientos, Análisis y diseño, Implementación, Pruebas y Despliegue, Administración de la configuración y cambios, Administración de proyecto y Entorno. Además, está compuesto por cuatro fases que son: Inicio, Elaboración, Construcción y Transición, las cuales permiten la realización de iteraciones durante el ciclo de vida del desarrollo del software. Ver Figura 1.1

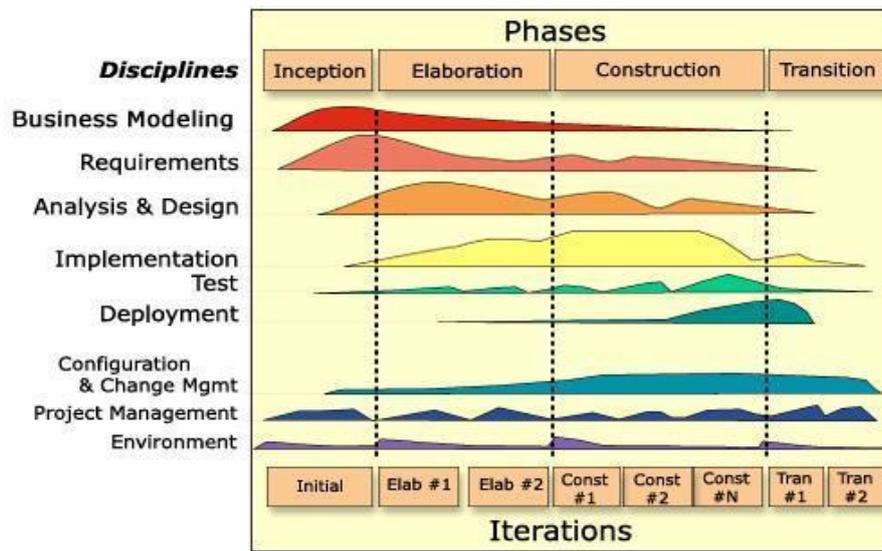


Figura 1.1. Proceso Unificado de Desarrollo

Los aspectos esenciales que definen el proceso unificado se resumen en tres características fundamentales: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Notación para el Modelado de Procesos de Negocio

BPMN constituye el estándar de modelado de procesos de negocio, que permite representar gráficamente las diferentes etapas de cada proceso de negocio. Esta notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes, para que haya un mejor entendimiento entre los involucrados e interesados del negocio. (18)

Dicha notación se inserta en el flujo de trabajo Modelado del negocio, propuesto por el Proceso Unificado de Desarrollo explicado anteriormente, que permite describir los procesos del negocio que representan de forma legible. Un Proceso de Negocio es una colección de actividades donde son tomadas una o varias clases de entradas, para crear una salida que tiene valor para un cliente. El principal objetivo de esta notación es lograr una correcta administración de los procesos del negocio asegurándose que los mismos sean modelados, automatizados, integrados y optimizados de forma continua, para lograr la eficiencia del grupo de desarrollo.

La correcta administración de los procesos permite reducir tanto el tiempo total utilizado para la realización del producto como los gastos, además de reducir los riesgos que atentan contra el correcto desarrollo del sistema. BPMN, fue inicialmente desarrollada por la organización Business Process Management Initiative, y es actualmente mantenida por el OMG (Object Management Group), luego de la fusión de las dos organizaciones en el año 2005. Su versión actual es la 1.1 y hay una versión futura propuesta, la 2.0.

La importancia de este estándar radica en que introduce el modelado de procesos en general, y apoya la comunicación y el entendimiento entre las personas. La Notación para el Modelado de Procesos de Negocio presenta eventos de inicio, intermedio y final. Por cada uno de estos se pueden utilizar varios símbolos en dependencia de la situación que sea necesaria modelar.

Lenguaje Unificado de Modelado

UML, es usado para especificar, visualizar, construir y documentar artefactos de un sistema de software, también para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir e incorporar las mejores prácticas actuales en un acercamiento estándar. (19)

Entre sus características principales se encuentran:

- ❖ Modela estructuras complejas.
- ❖ Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- ❖ Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.
- ❖ Emplea operaciones abstractas como guía para variaciones futuras, y añade variables si es necesario.
- ❖ Reemplaza a decenas de notaciones empleadas con otros lenguajes.

Al concluir este capítulo se llegan a las siguientes conclusiones:

Se investigaron los Sistemas de Información Hospitalaria, lo que permite concluir que en su mayoría son propietarios, con altos costos de licencia. Además, sólo abarcan las áreas administrativas o contables; excluyen los departamentos asistenciales, servicios de atención al paciente, así como las áreas de dirección o gerencias.

Estos sistemas no contemplan un módulo que gestione la información estadística vinculada a cada departamento y excluyen la generación de gráficos que muestren el comportamiento estadístico del hospital en dependencia de los indicadores seleccionados. Por lo que se necesita construir un software estándar que garantice el éxito de la calidad de atención al paciente.

Fueron analizadas un conjunto de herramientas, metodologías y tecnologías que han sido utilizadas para el desarrollo de un sistema de información hospitalaria que solucione las problemáticas existentes. Dicha selección está compuesta por: Eclipse como IDE de desarrollo, PostgreSQL Maestro como aplicación cliente para el manejo de la base de datos, Pgadmin como herramienta para el manejo de la base de datos, Visual Paradigm como herramienta case para el modelado y JasperReports e iReport para la generación de reportes.

Capítulo 2: Características del sistema

En este capítulo son abordados aspectos relacionados con el negocio y el sistema, sobre la base de una amplia descripción de los procesos asociados a los mismos, así como el análisis de todo lo concerniente a las actividades a automatizar, y los requerimientos del software vistos tanto desde el punto de vista funcional como no funcional.

2.1. Objeto de estudio

2.1.1. Objetivos estratégicos de la organización

El objetivo básico del Servicio de Estadística en un Hospital es obtener datos concretos de todos los procesos que se llevan a cabo en cada una de las unidades del mismo.

Objetivos estratégicos específicos

Constituye una estrategia el objetivo de alcanzar la disponibilidad y accesibilidad a los datos esenciales generados como consecuencia de un grupo de acciones llevadas a cabo en otras unidades que producen un resultado observable para el equipo de salud, lo que garantiza la seguridad de los reportes, eficiencia y calidad en la atención al paciente, y promueve el autocontrol de las actividades que se realizan en cualquier centro donde se encuentre el Sistema de Información Hospitalaria, que actualmente se encuentra en desarrollo, con el se contribuiría a elevar la calidad de atención al paciente al contar con reportes estadísticos relacionados además con el entorno que les rodea en las instituciones hospitalarias. (20)

Objetivos estratégicos generales

- ❖ Contribuir eficazmente al tratamiento seguro y efectivo de pacientes, al contar con un sistema que satisfaga la demanda continua y cada vez mayor que permita llevar a cabo evaluaciones de riesgos, gestionar la información disponible, ofrecer una respuesta rápida sobre el terreno y que esté basado en el uso de tecnología de la información innovadora.
- ❖ Identificar la necesidad de generar datos estadísticos en un determinado momento de un área o departamento específico del hospital, lo cual permite la prevención, el control y la vigilancia de las enfermedades transmisibles.
- ❖ Formular, documentar y llevar a cabo un registro para satisfacer las necesidades del personal de salud que labora con cada paciente, en contribución con un programa de investigación

llevado a cabo por profesionales de la salud, que tenga un impacto innovador y sostenible en el control de las enfermedades, mediante la mejora, creación y evaluación de nuevos instrumentos, intervenciones y estrategias

- ❖ aconsejar al equipo de salud acerca de cómo obtener mejores resultados en la búsqueda de un servicio determinado con el fin de establecer una relación coherente y así mejorar su adecuación al sistema.
- ❖ atender y satisfacer las solicitudes de cada unidad del hospital que requiera de los servicios estadísticos.
- ❖ recopilar la información generada en las unidades del centro de salud que cuente con el Sistema de Información Hospitalaria.
- ❖ llevar a cabo la supervisión de los datos de cada unidad para la elaboración de gráficos y tablas con los mismos actualizados.

2.1.2. Flujo actual de los procesos involucrados en el campo de acción

El departamento de Estadísticas de las instituciones hospitalarias tiene como objetivo principal emitir consolidados estadísticos como resultado de la información procesada proveniente de las distintas áreas de la institución.

El técnico de registro y estadísticas de salud, es el máximo responsable en el proceso de la información obtenida de las áreas del hospital, como Radiología, Consulta externa, Anatomía patológica, Bloque quirúrgico, Emergencias, Admisión, entre otras. Los trabajadores de cada una de dichas áreas deben entregar diariamente el parte informativo al departamento de Estadísticas. Para que el técnico de registro y estadísticas de salud proceda a evaluar cada uno de los datos emitidos, conformar los reportes totalizados y generar los indicadores hospitalarios, específicos por cada área y en un intervalo de tiempo determinado. Como resultado de dicho análisis se identifica el proceso Generación de reportes estadísticos a partir del cual se obtienen un conjunto de reportes como:

- ❖ Reportes estadísticos de Consulta Externa, Servicios de apoyos y Emergencia que contiene la información de los totales asociados a las áreas de Cirugía, Laboratorio, Radiología, Consulta externa, Obstetricia, Anatomía patológica y Emergencias.
- ❖ Reporte estadísticos Movimiento mensual según departamentos clínicos, Consolidado de departamentos clínicos de la institución, Movimiento mensual de la Emergencia, Consolidado de la emergencia, Movimiento mensual de la UCI y Consolidado de la UCI, que contienen la

información de los totales asociados al Registro de admisiones, transferencias, egresos y el Censo diario de los departamentos y servicios provenientes del área de Admisión.

- ❖ Reporte Movimiento mensual de la UCI y Consolidado de la UCI que contienen la información de los totales asociados al Registro de admisiones de la UCI, Registro de egreso de la UCI y Registro de transferencias de la UCI proveniente del área Unidad de cuidados intensivos (UCI).
- ❖ Reporte Movimiento mensual de la Emergencia y el Consolidado de la Emergencia que contienen la información del total asociado al Movimiento de pacientes de emergencia proveniente del área de Emergencias.
- ❖ Reporte de Morbilidad y Mortalidad hospitalaria a partir del diagnóstico clínico que emite el médico a pacientes hospitalizados, que contiene la información asociada a la Historia clínica Hoja frontal y el informe de biopsia proveniente del área Anatomía patológica.
- ❖ Reporte de Morbilidad de la Emergencia a partir del diagnóstico clínico que emite el médico a pacientes atendidos en este servicio, estos contienen información asociada a la historia médica de Emergencia pediátrica y de adultos.
- ❖ Reporte de Morbilidad y Mortalidad de la UCI a partir del diagnóstico clínico que emite el médico a pacientes atendidos en esta unidad.

Una vez generados los reportes que resumen la información total de cada área, se generan los Indicadores hospitalarios de interés para el ministerio y el hospital, pues estos responden a la productividad, eficiencia, disponibilidad, accesibilidad, actividades, uso, cobertura, eficacia, efectividad o impacto, utilización de los recursos y la atención a pacientes.

El técnico de registro y estadísticas de salud procede además a hacer uso de los parámetros definidos en los reportes, a identificar los criterios de interés para la dirección de la institución y elaborar diferentes gráficos que visualicen el comportamiento de la entidad, los cuales son informados a la institución y ministerio. De existir la necesidad de graficar otra información, esta debe ser analizada manualmente en cada uno de los partes emitidos. Así como determinar los totales, definir los indicadores y determinar los valores a tomar por cada criterio a representar en el gráfico.

2.1.3. Causas que originan la situación problemática

Al realizar el análisis del flujo de actividades que tiene lugar en el proceso Generación de reportes estadísticos, identificado en el departamento de Estadísticas de las instituciones hospitalarias, han sido

identificadas un conjunto de irregularidades que atentan contra una óptima obtención de los consolidados estadísticos.

La información almacenada se recopila en forma de documentos en los archivos, para lo que es preciso contar con tiempo suficiente para buscarlo y organizarlo una vez consultado cada vez que se requiera, además de que esta información corre gran riesgo de perderse. Los datos relacionados con los reportes de las unidades en los hospitales se guardan en papeles, deteriorándose con el tiempo. Los reportes se realizan sin la seguridad de saber que realmente sean estos los datos generados en las unidades.

Además Los reportes estadísticos de las unidades internas se realizan en documentos de papel, de forma personal es la entrega, pudiéndose utilizar ese tiempo de trabajo necesario, para la realización de otras actividades. La información de los reportes emitidos genera un gran cúmulo de datos registrados en documentos de papel, sin la elaboración de gráficos que ayuden a la comprensión del balance. Los pedidos de las estadísticas de cada unidad se solicitan y archivan en grandes cantidades de archivos en papel, lo cual dificulta la búsqueda de alguna información requerida, lo que hace más complicado el proceso de pedido.

2.2. Objeto de automatización

Debido a la situación problemática descrita, se propone el desarrollo de un sistema informático que permita automatizar el proceso para la generación de reportes estadísticos que garantice un óptimo registro y generación de dichos datos de un hospital determinado. Las actividades principales que requieren ser automatizadas son:

- Elaborar reporte de movimiento de hospitalizados a partir de las estadísticas recibidas del área de admisión: Se desean automatizar la elaboración de reportes de los movimientos que se le realizan a los pacientes en las unidades del hospital a partir de los criterios que necesite el usuario, los cuales pueden ser departamento o servicio en un rango de meses o en un mismo mes, con los cuales se genera el movimiento mensual de hospitalizados por departamentos clínicos y servicios.
- Generar reportes de Morbilidad hospitalaria: se desea automatizar la generación de los reportes confeccionados a partir del diagnóstico clínico que emite el médico a pacientes hospitalizados, lo cual brinda la posibilidad de elegir criterios como el mes, el nivel de diagnóstico y la procedencia, que viene dada en dependencia de si el usuario quiere obtener los datos

estadísticos de enfermos en una población determinada por una enfermedad específica, o la ocurrencia, la cual representa el estado de la estadística de enfermos por una enfermedad específica en un área de la institución hospitalaria.

- Generar reportes estadísticos de Consulta Externa: se desea automatizar la generación de reportes estadísticos provenientes del área de consulta externa, lo cual permite obtener por cada mes seleccionado o por la selección de varios meses a la vez cuantas consultas de uno o varios tipos hubo en ese periodo.
- Generar reportes estadísticos de Emergencia: se desea automatizar la generación de reportes estadísticos provenientes del área de emergencias, los cuales muestran por mes y en grupos lógicos el reporte según los criterios previamente seleccionados por el usuario. Para su la generación de dicho reporte se consulta la hoja de emergencia.
- Generar reporte de mortalidad hospitalaria: se desea automatizar la generación de los reportes confeccionados a partir del diagnóstico clínico que emite el médico de sus pacientes fallecidos, lo cual brinda la posibilidad de elegir criterios como el mes, el nivel de diagnóstico y la procedencia, que viene dada en dependencia de si el usuario quiere obtener los datos estadísticos de fallecidos en una población determinada por una enfermedad específica, o la ocurrencia, la cual representa el estado de la estadística de fallecidos por una enfermedad específica en un área de la institución hospitalaria.
- Generar reporte de natalidad hospitalaria: se desea automatizar la generación de los reportes confeccionados a partir de los criterios seleccionados por el usuario, los cuales pueden ser: año, mes, sexo, residencia, edad de la madre, talla, peso y edad gestacional. Es consultada la entidad Libro de la sala de partos con los criterios antes mencionados se genera el reporte de información.
- Generar comportamiento de la morbilidad o mortalidad por patología: se desea automatizar la generación de los reportes confeccionados a partir de los criterios seleccionados por el usuario, los cuales pueden ser: rango de años, patología, morbilidad y mortalidad. Una vez seleccionados se obtiene el reporte que representa el comportamiento de una enfermedad en una población determinada según afectaciones de enfermos y fallecidos, se muestran los atributos años, tasas e índice ordenados ascendentemente por el año, así como la cantidad de elementos configurados para mostrar por páginas, lo cual permite navegar por el resultado.
- Generar reporte de causas de cancelación de intervenciones quirúrgicas: se desea automatizar la generación de los reportes confeccionados a partir de un rango de años seleccionado por el usuario, el cual depende de la información proveniente de las entidades causas de cancelación

de cirugías electivas y causas de cancelación de cirugías de emergencia. Una vez consultadas dichas entidades se genera el reporte, lo cual muestra por causa de suspensión la cantidad de intervenciones suspendidas.

- Generar Consolidado de departamentos clínicos y servicios: se desea automatizar la generación de los reportes confeccionados a partir de los criterios seleccionados por el usuario, los cuales pueden ser años, departamentos y servicios, los cuales se completan al consultar las entidades registro de admisiones, registro de egresos y registro de transferencias para generar el reporte.
- Generar reportes estadísticos de Servicios de apoyo: se desea automatizar la generación de los reportes confeccionados a partir de los criterios seleccionados por el usuario, los cuales pueden ser: mes o meses, año y radiología. Para la generación de dicho reporte se necesita consultar entidades como solicitud de biopsia, solicitud de citología, estadísticas totales acumuladas.

Estas actividades necesitan ser automatizadas debido a la necesidad de informatización de varios procesos que se llevan a cabo en las estadísticas de los hospitales. Se realizan los trámites de los documentos de forma manual, lo cual aumenta la posibilidad de que existan alteraciones como consecuencia de equivocaciones producto del gran cúmulo de datos a recoger, además de que constantemente deben ser cambiados, sin disponer de tiempo suficiente para generarlos en un corto plazo.

2.3. Información que se maneja

El área de estadística hace uso de la información generada en los distintos departamentos de las instituciones hospitalarias, ya que para generar los reportes se necesita obtener información proveniente de las distintas áreas del hospital como radiología, consulta externa, anatomía patológica, laboratorio, admisión y banco de sangre. Entre la información que se recibe se encuentran:

- Registro de egreso: el cual contiene la información referente a los egresos que se han realizado en la institución como son tipo de egreso, motivos, fecha en que se realizó y causas del mismo. Esto permite conocer la cantidad de disfunciones realizadas y las causas de las mismas.
- Registro de admisiones: registra la información de los pacientes ingresados, la fecha en que ocurre el ingreso, motivos, servicio que atiende al paciente, diagnóstico y ubicación del paciente. Permite obtener la cantidad de ingresos realizados por mes, y los movimientos realizados dentro de las instituciones.

- Solicitud de Intervención Quirúrgica: contiene la información de las cirugías realizadas, lo que permite conocer si las mismas han sido o no de emergencias, y si el resultado de las mismas ha sido el esperado. Esta información es importante porque permite conocer si las cirugías planificadas en las instituciones se realizaron y en caso de no ser posible cuales fueron las causas que propiciaron que esto sucediera.
- Solicitud de Transfusión: Recoge toda la información referente a una solicitud de transfusión. Contiene una sección de datos del paciente (Nombre y apellidos del paciente, número de historia clínica del paciente, Edad, Diagnóstico, Ubicación, Hemoglobina, Tipo y factor de la sangre del paciente).
También recoge otros datos de interés tales como: una sección de datos de la Transfusión (Carácter de la transfusión, Transfusión anterior, Cuando, Reacciones, Tipo de reacción, Tratamiento, Acto operatorio, Tipo de intervención), una sección de datos de los funcionarios (Nombre del Médico, Número de registro del ministerio de salud, Número del colegio de medicina, Cédula, Recibido, Muestra tomada por Aceptación, Fecha, Hora, Observaciones) y una sección para el Personal de Banco de Sangre (Grupo-Rh del receptor, Serial de las bolsas de componentes, Grupo-Rh, Segmento, Componente, Fecha, Hora, Pruebas Cruzadas, Responsable), esta información permite determinar la cantidad de transfusiones realizadas.

2.4. Propuesta del sistema

El objetivo es implementar un sistema que gestione toda la información estadística de un hospital con el fin de obtener resultados estadísticos. El Módulo Estadística del Sistema de Información Hospitalaria alas HIS facilitará la gestión de la información a través de reportes y gráficos, además será integrado con los restantes módulos desarrollados que conformarán un Sistema de Información Hospitalaria que permita la gestión de la información en los demás departamentos del hospital. Sus características básicas estarán centradas en:

- ❖ Generar los resúmenes o reportes estadísticos.
- ❖ Visualizar la información estadísticas de forma rápida y comprensible.

2.5. Modelo de Negocio

Un paso inicial para el desarrollo de cualquier sistema, es el análisis de los procesos de negocio. Estos describen las actividades que se llevan a cabo y su interacción con elementos externos al proceso

(actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios.

El Modelo de negocio es el primer flujo de trabajo propuesto por la metodología RUP, tiene mayor peso durante la fase de inicio, debido a los propósitos que el mismo tiene establecido, entre los que se encuentran:

- ❖ Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- ❖ Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- ❖ Derivar los requerimientos del sistema que va a soportar la organización.
- ❖ Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer.

Para alcanzar un mayor entendimiento del negocio actual, se decide modelar los procesos de negocio identificados con el uso de Business Process Management (BPM). Este último es la tecnología que permite diseñar, implementar, simular, ejecutar y modificar los procesos de negocios de una organización.

Dentro de un proceso de negocio se identifican un conjunto de elementos básicos que permiten conocer y entender cada proceso.

Dentro de cada proceso se identifican:

- ❖ Actores: Elementos encargados de realizar cada actividad. Pueden ser individuos, grupos de personas o departamentos organizacionales, que además pueden clasificarse según el negocio en diferentes roles.
- ❖ Actividades: Acciones que se realizan dentro de un proceso.
- ❖ Recursos: Todo aquello que es usado o afectado por las actividades. Son considerados entradas, salidas o resultados de una actividad.

Las actividades tienen condiciones previas (entradas a una actividad) y efectos (salidas de una actividad). Las condiciones previas o entradas son los recursos requeridos, consumidos o utilizados por las actividades, los cuales tienen asociado un estado. Se muestra además la sincronización requerida entre las diferentes actividades.

Para ganar una mayor comprensión por parte de los beneficiarios en dicho modelo, se decide modelar los procesos de negocio identificados, con el uso del estándar BPMN).

Mediante este estándar se realizan los diagramas de procesos de negocio, donde son estructurados con un grupo de elementos gráficos.

Las cuatro categorías básicas de los elementos son:

- ❖ Objetos de flujo, dentro del cual se identifican los eventos, actividades y decisiones.
- ❖ Objetos de conexión que incluyen los flujos de secuencia, flujos de mensajes y asociaciones.
- ❖ Objetos de carriles (swimlanes), que constituye un mecanismo de organización de actividades en categorías visuales separadas, para ilustrar las diferentes capacidades funcionales o responsabilidades, estos incorporan los llamados pool (carril) y lane (carriles).
- ❖ Artefactos, estos incluyen los objetos de datos, los grupos y anotaciones, lo cual brinda una representación más apropiada del contexto del negocio.

2.5.1. Actores involucrados en el proceso de negocio

Roles	Funciones
Técnico de registros y estadísticas de salud (TRES)	Confecciona las estadísticas generales relacionadas con la información que generan los servicios de apoyo, consulta externa y admisión.
Ministerio de salud	Solicita y recibe los reportes estadísticos de alguna de las áreas de las instituciones hospitalarias.

Tabla 1.1. Actores involucrados en el proceso de negocio

2.5.2. Diagrama de procesos de negocio

2.5.2.1. Proceso de negocio: Generar reportes estadísticos

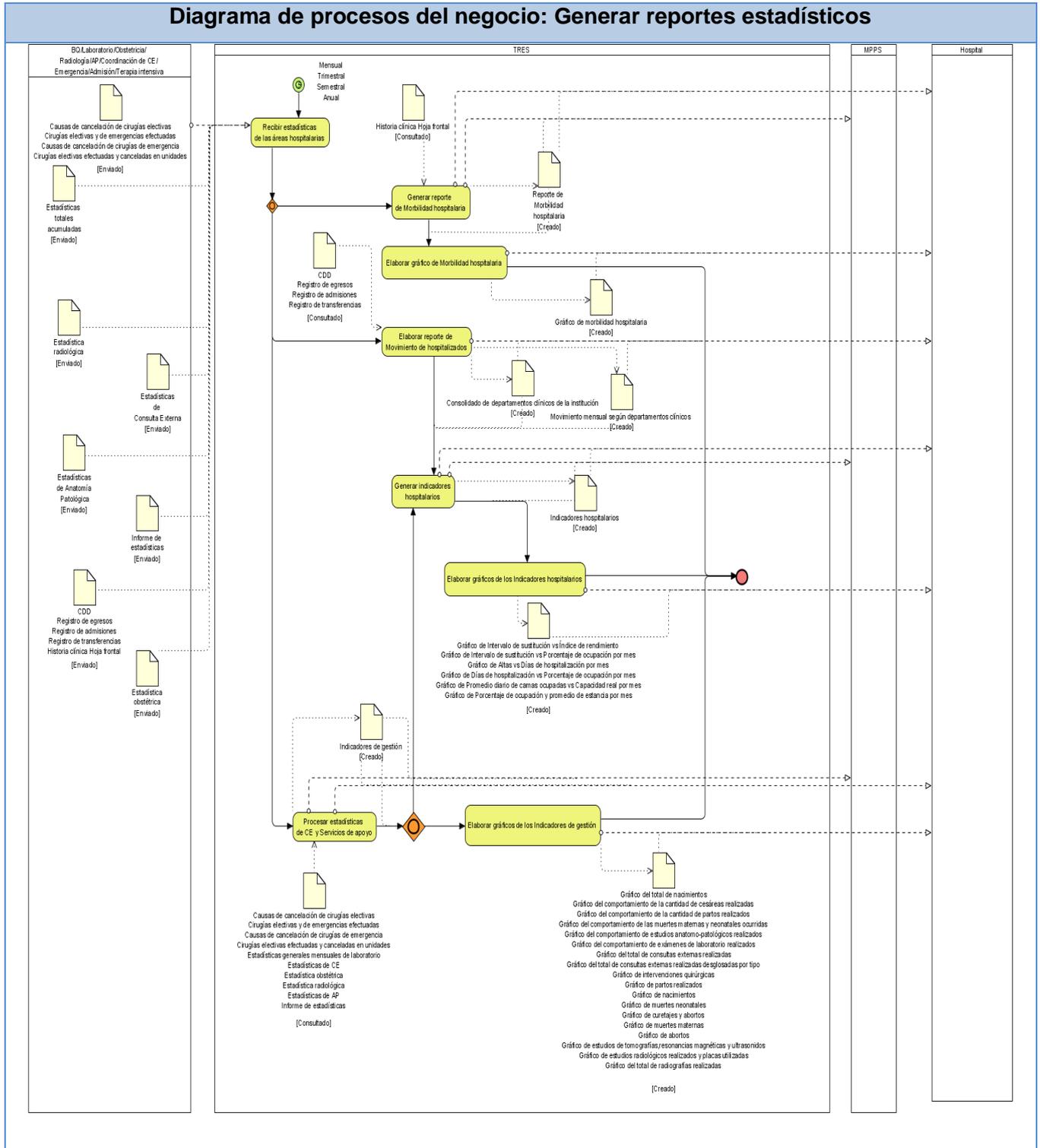


Figura 1.2. Diagrama de procesos del negocio: Generar reportes estadísticos.

2.5.2.2. Descripción textual de los procesos del negocio

PROCESO:	Generar reportes estadísticos			
Misión:	Obtener las estadísticas generales de interés para el Ministerio del Poder Popular para la Salud y el hospital.			
Responsable:	Especialista funcional del área de Estadísticas			
ACTORES INVOLUCRADOS				
Rol		Funciones		
Técnico (a) de registros y estadísticas de salud (TRES)		Confeccionar las estadísticas generales relacionadas con la información que generan los servicios de apoyo, consulta externa y admisión.		
ACTIVIDADES				
Actividad:	Recibir estadísticas de las áreas hospitalarias.			
Flujo de Información				
El (la) técnico(a) de registros y estadísticas de salud recibe las estadísticas que se generan en cada área del hospital para procesarlas.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Causas de cancelación de cirugías electivas	BQ	TRES	Papel	Mensual
Cirugías electivas y de emergencias efectuadas	BQ	TRES	Papel	Mensual
Causas de cancelación de cirugías de emergencia	BQ	TRES	Papel	Mensual
Cirugías electivas efectuadas y canceladas en unidades	BQ	TRES	Papel	Mensual
Historia clínica Hoja frontal	Admisión	TRES	Papel	Diario
Registro de admisiones	Admisión	TRES	Digital	Mensual
Registro de transferencias	Admisión	TRES	Digital	Mensual
Registro de egresos	Admisión	TRES	Digital	Mensual
Censo diario de los departamentos y servicios (CDD)	Admisión	TRES	Digital	Diario
Estadísticas totales acumuladas	Laboratorio	TRES	Papel	Mensual
Estadísticas de Consulta Externa	Consulta externa	TRES	Papel	Mensual
Estadística obstétrica	Obstetricia	TRES	Papel	Mensual

Estadística radiológica	Radiológica	TRES	Papel	Mensual
Estadísticas de Anatomía Patológica	Anatomía Patológica	TRES	Papel	Mensual
Informe de biopsia	Anatomía Patológica	TRES	Papel	Mensual
Movimiento de pacientes de emergencia	Emergencia	TRES	Papel	Mensual
Causas de atención en el servicio de emergencia	Emergencia	TRES	Papel	Mensual
Causas de atención por áreas del servicio de emergencia	Emergencia	TRES	Papel	Mensual
Historia Médica de Emergencia de Adultos	Emergencia	TRES	Papel	Mensual
Historia Médica de Emergencia Pediátrica	Emergencia	TRES	Papel	Mensual
Registro de admisiones de la UCI	UCI	TRES	Papel	Mensual
Registro de egreso de la UCI	UCI	TRES	Papel	Mensual
Registro de transferencias de la UCI	UCI	TRES	Papel	Mensual
Actividad:	Generar reportes estadísticos de CE, Servicios de apoyos y Emergencia.			
Flujo de Información				
El (la) TRES introduce la información estadística recibida de las áreas del hospital en hojas de cálculo y genera con estas los reportes estadísticos de CE, Servicios de apoyos y Emergencia.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Reportes estadísticos de CE, Servicios de apoyo y Emergencia	TRES	MPPS Hospital	Digital Papel	Mensual Trimestral Semestral Anual
Cirugías electivas y de emergencias efectuadas	BQ	TRES	Papel	Mensual
Cirugías electivas efectuadas y canceladas en unidades	BQ	TRES	Papel	Mensual
Estadísticas totales acumuladas	Laboratorio	TRES	Papel	Mensual
Estadísticas de Consulta Externa	Consulta externa	TRES	Papel	Mensual

Estadística obstétrica	Obstétrica	TRES	Papel	Mensual
Estadística radiológica	Radiológica	TRES	Papel	Mensual
Movimiento de pacientes de emergencia	Emergencia	TRES	Papel	Mensual
Estadísticas de Anatomía Patológica	Anatomía Patológica	TRES	Papel	Mensual
Actividad:	Elaborar reporte de departamentos clínicos.			
Flujo de Información				
El (la) TRES a partir de las estadísticas recibidas del área de admisión genera el Movimiento mensual según departamentos clínicos y el Consolidado de departamentos clínicos de la institución.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Movimiento mensual según departamentos clínicos	TRES	MPPS Hospital	Digital Papel	Mensual
Consolidado de departamentos clínicos de la institución	TRES	MPPS Hospital	Digital Papel	Mensual Trimestral Semestral Anual
Movimiento mensual de la Emergencia	TRES	TRES	Digital	Mensual
Consolidado de la emergencia	TRES	TRES	Digital	Mensual
Movimiento mensual de la UCI	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Consolidado de la UCI	TRES	TRES	Digital	Mensual
Registro de admisiones	Admisión	TRES	Digital	Mensual
Registro de transferencias	Admisión	TRES	Digital	Mensual
Registro de egresos	Admisión	TRES	Digital	Mensual
Censo diario de los departamentos y servicios (CDD)	Admisión	TRES	Digital	Diario
Actividad:	Elaborar reporte de Movimiento de hospitalizados en la UCI.			
Flujo de Información				
El (la) TRES a partir de las estadísticas recibidas de la Unidad de Cuidados Intensivos genera el Movimiento mensual de la UCI y el Consolidado de la UCI.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Registro de admisiones de la UCI	UCI	TRES	Papel	Mensual
Registro de egreso de la UCI	UCI	TRES	Papel	Mensual

Registro de transferencias de la UCI	UCI	TRES	Papel	Mensual
Movimiento mensual de la UCI	TRES	Hospital MPPS	Digital Papel	Mensual
Consolidado de la UCI	TRES	Hospital MPPS	Digital Papel	Mensual Trimestral Semestral Anual
Actividad:	Elaborar reporte de movimientos en la Emergencia.			
Flujo de Información				
El (la) TRES a partir de las estadísticas recibidas de la Emergencia genera el Movimiento mensual de la Emergencia y el Consolidado de la Emergencia.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Movimiento de pacientes de emergencia	Emergencia	TRES	Papel	Mensual
Movimiento mensual de la Emergencia	TRES	Hospital MPPS	Digital Papel	Mensual
Consolidado de la Emergencia	TRES	Hospital MPPS	Digital Papel	Mensual Trimestral Semestral Anual
Actividad:	Elaborar gráficos.			
Flujo de Información				
Luego de haber generado los reportes de interés para el hospital y el MPPS, el (la) TRES elabora diferentes gráficos con el uso de los parámetros definidos en los reportes.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Gráficos	TRES	Hospital MPPS	Digital Papel	Semanal Mensual Semestral Trimestral Anual
Reportes estadísticos de CE, Servicios de apoyo y Emergencia	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Indicadores hospitalarios	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Consolidado de la UCI	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Movimiento mensual de la UCI	TRES	TRES	Digital	Mensual Trimestral Semestral Anual

Capítulo 2 Características del sistema

Consolidado de departamentos clínicos de la institución	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Movimiento mensual según departamentos clínicos	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Consolidado de la Emergencia	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Movimiento mensual de la Emergencia	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Reporte de Morbilidad hospitalaria	TRES	TRES	Digital	Mensual Anual
Reporte de Morbilidad de la Emergencia	TRES	TRES	Digital	Mensual Anual
Reporte de Morbilidad de la UCI	TRES	TRES	Digital	Mensual Anual
Causas de cancelación de cirugías de emergencia	BQ	TRES	Papel	Mensual
Causas de cancelación de cirugías electivas	BQ	TRES	Papel	Mensual
Actividad:	Generar reporte de Morbilidad y Mortalidad hospitalaria.			
Flujo de Información				
El (la) TRES confecciona el reporte de Morbilidad y Mortalidad hospitalaria a partir del diagnóstico clínico que emite el médico a pacientes hospitalizados.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Reporte de Morbilidad hospitalaria	TRES	Hospital MPPS	Digital	Mensual Anual
Reporte de Mortalidad hospitalaria	TRES	Hospital MPPS	Digital	Mensual Anual
Historia clínica Hoja frontal	TRES	TRES	Papel	Mensual
Informe de biopsia	Anatomía patológica	TRES	Papel	Mensual
Actividad:	Generar reporte de Morbilidad y Mortalidad en Emergencia.			
Flujo de Información				
El (la) TRES confecciona el reporte de Morbilidad de la Emergencia a partir del diagnóstico clínico que emite el médico a pacientes atendidos en este servicio.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Reporte de Morbilidad de la Emergencia	TRES	Hospital MPPS	Digital	Mensual Anual

Capítulo 2 Características del sistema

Reporte de Mortalidad de la Emergencia	TRES	Hospital MPPS	Digital	Mensual Anual
Historia Médica de Emergencia de Adultos	Emergencia	TRES	Papel	Mensual
Historia Médica de Emergencia Pediátrica	Emergencia	TRES	Papel	Mensual
Actividad:	Generar reporte de Morbilidad y Mortalidad en la UCI.			
Flujo de Información				
El (la) TRES confecciona el reporte de Morbilidad de la UCI a partir del diagnóstico clínico que emite el médico a pacientes atendidos en esta unidad.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Reporte de Morbilidad de la UCI	TRES	Hospital MPPS	Digital	Mensual Anual
Reporte de Mortalidad de la UCI	TRES	Hospital MPPS	Digital	Mensual Anual
Historia clínica	UCI	TRES	Papel	Mensual
Actividad:	Generar indicadores hospitalarios.			
Flujo de Información				
El (la) TRES a partir de los Indicadores de gestión, el Movimiento mensual según departamentos clínicos y el Consolidado de departamentos clínicos de la institución genera los Indicadores hospitalarios de interés para el ministerio y el hospital.				
Artefacto	Emisor	Receptor	Formato	Frecuencia
Movimiento mensual según departamentos clínicos	TRES	TRES	Digital	Mensual
Consolidado de departamentos clínicos de la institución	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
Indicadores hospitalarios	TRES	Hospital MPPS	Papel Digital	Mensual Trimestral Semestral Anual
Reportes estadísticos de CE, Servicios de apoyo y Emergencia	TRES	TRES	Digital	Mensual Trimestral Semestral Anual
REGLAS DEL NEGOCIO				
Regla		Actividad		
No existe		No existe		
REFERENCIAS A PROCEDIMIENTOS				
Procedimiento		Descripción		
No existe		No existe		

2.6. Especificación de los requerimientos de software

Los requerimientos son la descripción de los servicios y obligaciones que debe cumplir un software determinado, estos pueden ser funcionales, que representan las funcionalidades con que debe contar el producto una vez concluido; y no funcionales, que son las cualidades o propiedades que debe poseer el sistema. (21)

2.6.1. Requerimientos funcionales

Los requerimientos funcionales surgen a partir de un análisis profundo de la situación problemática, así como del resultado de varias entrevistas realizadas al personal de salud, con el objetivo de satisfacer completamente las necesidades de los usuarios finales. Como resultado del estudio del proceso de negocio expuesto con anterioridad y de las actividades definidas para automatizar; a continuación se listan los requerimientos funcionales correspondientes al proceso Generar reportes estadísticos:

RF 1. Generar Movimiento mensual según departamentos clínicos y servicios: Permite generar el movimiento mensual desglosado por departamentos y servicios, lo cual permite manejar los datos de las admisiones, transferencias, egresos, días de hospitalización, defunciones e infecciones intrahospitalarias.

RF 2. Generar reportes estadísticos de Consulta Externa: Permite generar un reporte con la cantidad de pacientes atendidos en Consulta Externa y desglosado por tipo de consulta.

RF 3. Generar reportes estadísticos de Servicios de apoyo: Permite generar los principales reportes estadísticos de los servicios de Laboratorio, Imagenología, Radiología, Anatomía Patológica, Quirófano y Sala de partos desglosados por meses.

RF 4. Generar reportes estadísticos de Emergencia: Permite generar un reporte con la cantidad de pacientes atendidos en la emergencia y la cantidad de referencias y contrarreferencias emitidas.

RF 5. Generar reporte de morbilidad hospitalaria: Permite generar un reporte con la morbilidad en dependencia de los criterios seleccionados, y devuelve la morbilidad a partir de los diferentes niveles para el diagnóstico del paciente y desglosado por el departamento o servicio seleccionado como criterio, así como por grupos de edad, sexo o procedencia. Devuelve las 30 causas de mayor incidencia a partir de los criterios seleccionados y permite consultar el resto de las causas de morbilidad.

RF 6. Generar reporte de mortalidad hospitalaria: Permite generar un reporte con la mortalidad en dependencia de los criterios seleccionados, desglosado por el departamento o servicio, así como por

grupos de edad, sexo o procedencia. Devuelve las 30 causas de muerte de mayor incidencia a partir de los criterios seleccionados y permite consultar el resto de las causas de mortalidad.

RF 7. Generar reporte de natalidad: Permite generar un reporte de natalidad en dependencia de los criterios seleccionados, devuelve la edad gestacional y la edad de la madre, así como datos del recién nacido.

RF 8. Generar comportamiento de la morbilidad o mortalidad por patologías: Permite generar para un lapso de tiempo dado en años, las tasas para cada patología y los índices para cada año. Permite visualizar la variación ya sea por disminución o aumento durante el lapso de tiempo.

RF 9. Generar reporte de causas de cancelación de intervenciones quirúrgicas: Permite generar un reporte donde se refleja por cada causa de cancelación la cantidad de intervenciones quirúrgicas canceladas.

RF 10. Generar consolidado de departamentos clínicos y servicios: Permite generar un consolidado en dependencia de los criterios seleccionados. Refleja de forma acumulativa datos de las admisiones, altas, transferencias, e indicadores hospitalarios, desglosados por los departamentos o servicios especificados como criterios para generar el consolidado.

RF 11. Generar comportamiento de la natalidad: Permite generar para un lapso de tiempo dado en años, las tasas de nacidos vivos o muertos, en dependencia de la selección, así como los índices para cada año. Permite visualizar la variación ya sea por disminución o aumento durante el lapso de tiempo.

2.6.2. Requerimientos no funcionales

Los requisitos no funcionales son los que definen propiedades y restricciones del sistema. Existen múltiples categorías para clasificar los mismos. A continuación se muestran los requisitos con los que cumple el sistema:

Usabilidad: El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido. Debe brindar comodidad a la hora de acceder a las diferentes funcionalidades que proporciona la aplicación mediante teclas de acceso rápido, la navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario. Reutilizará diseños de aplicaciones comúnmente usada por los usuarios finales con vistas a aprovechar la experiencia de usuario.

Fiabilidad: En los servidores de los hospitales se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se garantizarán además, políticas de respaldo a toda la información, para que no ocurran pérdidas en caso de desastres ajenos al sistema.

Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros hospitales por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas. Se mantendrá seguridad y control a nivel de usuario, lo cual garantiza el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.

Se registrarán todas las acciones que se realizan, lo cual permite tener el control de las actividades de cada usuario en todo momento. El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, lo que proporciona un registro de actividades (log) de cada usuario en el sistema.

El sistema implementará un control de cambios a determinados campos de información (seleccionados por su importancia), de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.

Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista. Además, permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

Eficiencia: El Centro de Datos permitirá agregar recursos para aumentar el poder de procesamiento y almacenamiento sin afectar los sistemas, lo que garantiza las expansiones motivadas por futuros requerimientos.

El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén en uso, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

Soporte

- Seguridad de acceso y administración de usuarios

Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP.

- **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, lo cual permite archivar 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.

Requisitos para la documentación de usuarios en línea y ayuda del sistema: Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

Interfaz (Interfaces de usuario): Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegables que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.

Además en esta se incorporarán asistentes que faciliten el uso del sistema por los usuarios, en procesos con determinado nivel de complejidad, que lo guíen paso a paso para minimizar la posibilidad de errores. El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, para minimizar los pasos a dar en cada proceso. Se diseñarán salidas de información en forma de gráficos, estadísticas, pronósticos y análisis comparativos que puedan ser usados como soporte para la toma de decisiones. El sistema incluirá reportes estándares y parametrizables que permitirán al usuario configurar la información de salida y el orden en que aparecen los datos. Las salidas se podrán generar en el formato de fichero PDF.

El sistema usará el formato estándar WSDL para la descripción de los servicios web. Además de que implementará mecanismos de encriptación de datos para el intercambio de información con sistemas

externos, utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

2.7. Modelo de Casos de Uso del Sistema

2.7.1. Definición de actores

Actores del sistema: Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él.

Actor	Funciones
Técnico de registros y estadísticas de salud (TRES)	Genera las estadísticas generales relacionadas con la información que se obtiene de los servicios de apoyo, consulta externa y admisión.
ALAS-RIS	Sistema externo con el que se comunica el módulo para la gestión de la información de los estudios radiológicos e imagenológicos.
SIPOL	Sistema externo con el que se comunica el módulo para la gestión de la información de las autopsias forenses.

Tabla 1.2. Actores del sistema

2.7.2. Vista global de actores

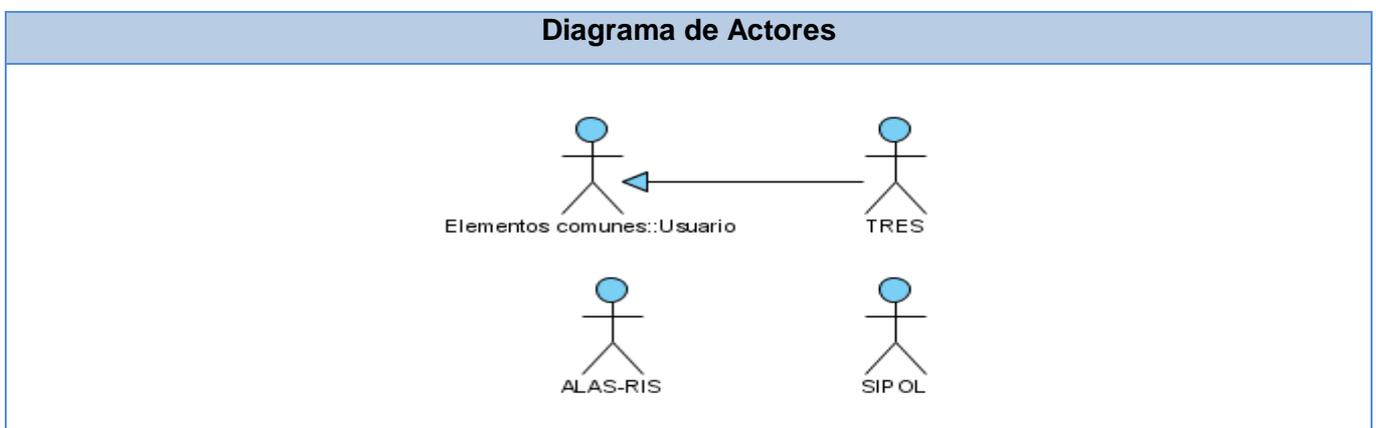


Figura 1.3. Vista global de actores

2.7.3. Diagramas de Casos de Uso

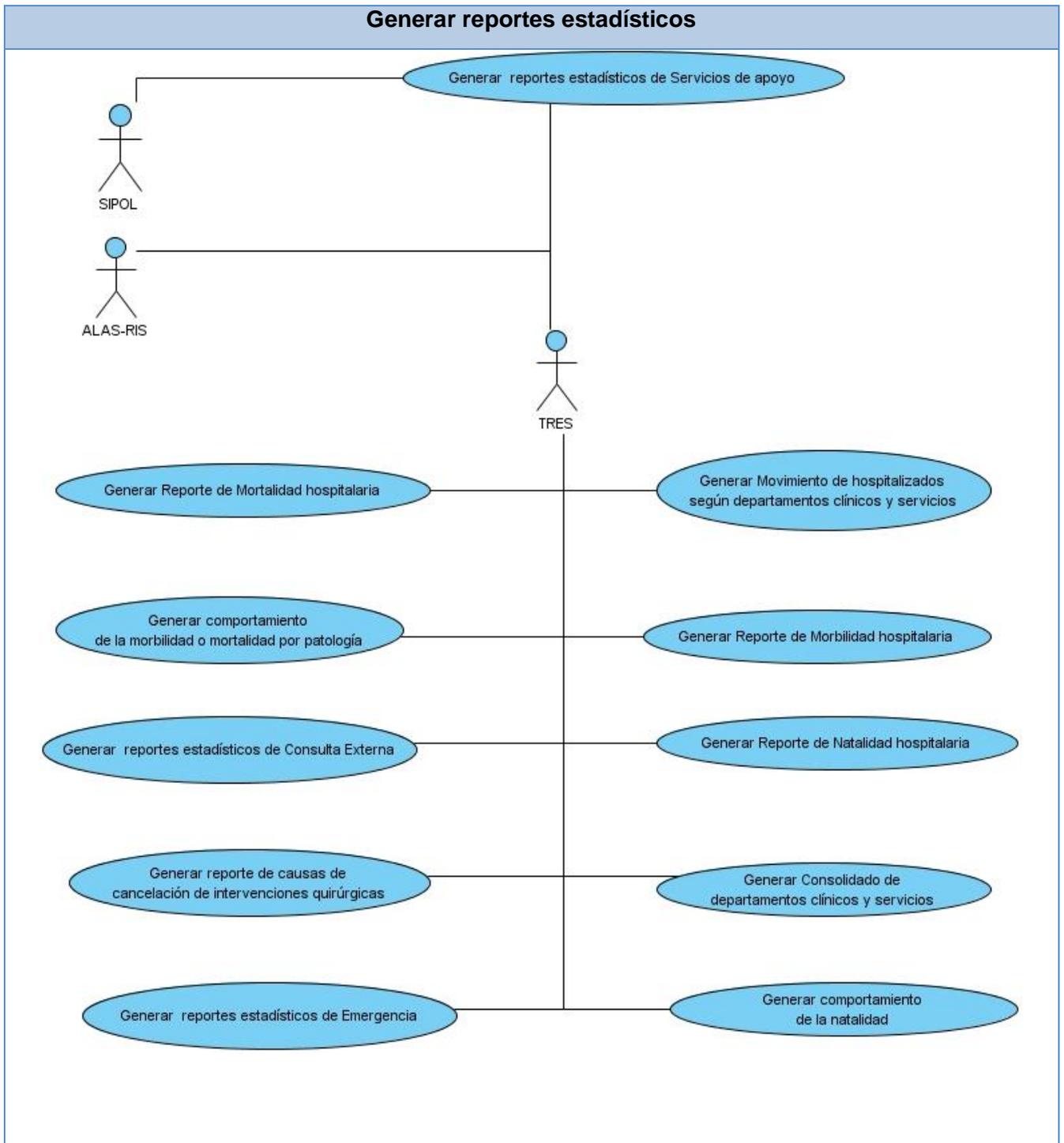


Figura 1.4. DCU Generar reportes estadísticos

2.7.4. Descripción Textual de los Casos de Uso

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Un caso de uso describe un proceso de principio a fin, es decir, una secuencia de eventos, las acciones y las transacciones que se requieren para realizarlo y producen un resultado observable para un actor.

CU_ Generar Movimiento de hospitalizados según departamentos clínicos y servicios	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Movimiento de hospitalizados según departamentos clínicos y servicios, el sistema brinda la posibilidad de generar el reporte, consulta las entidades correspondientes, y genera el reporte de información, además de brindar la opción de imprimir el reporte. El caso de uso termina
Precondición:	No existe.
Poscondición:	Se generó un reporte de Movimiento de hospitalizados según departamentos clínicos y servicios dado criterios.
Referencias:	RF 1

Tabla 1.3. CU_ Generar Movimiento de hospitalizados según departamentos clínicos y servicios

CU_ Generar reportes estadísticos de Consulta Externa	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar reportes estadísticos de Consulta Externa. El sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados, consulta las entidades correspondientes y genera el reporte de información. El sistema brinda la opción de imprimir el reporte y generar el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte estadístico de Consulta Externa.
Referencias:	RF 2

Tabla 1.4. CU_ Generar reportes estadísticos de Consulta Externa

CU_ Generar reportes estadísticos de Servicios de apoyo	
Actores:	TRES(Inicia), ALAS-RIS, SIPOL
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar reportes estadísticos de Servicios de apoyo. El sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados, consulta las entidades correspondientes y se comunica con los sistemas ALAS-RIS y SIPOL, genera el reporte de información. El sistema brinda la opción de imprimir el reporte y genera el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte estadístico de Servicios de apoyo.
Referencias:	RF 3

Tabla 1.5. CU_ Generar reportes estadísticos de Servicios de apoyo

CU_Generar reportes estadísticos de Emergencia	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar reportes estadísticos de Servicios de apoyo. El sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados, consulta las entidades correspondientes y se comunica con los sistemas ALAS-RIS y SIPOL, genera el reporte de información. El sistema brinda la opción de imprimir el reporte y genera el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un Reporte estadístico de Emergencia.
Referencias:	RF 4

Tabla 1.6. CU_Generar reportes estadísticos de Emergencia

CU_Generar reporte de Morbilidad hospitalaria	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Morbilidad hospitalaria, el sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados consulta la entidad Registro de egresos y genera el reporte de información. El sistema permite imprimir el reporte y generar el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de Morbilidad hospitalaria dado criterios.
Referencias:	RF 5

Tabla 1.7. CU_Generar reporte de Morbilidad hospitalaria

CU_ Generar reportes estadísticos de Mortalidad hospitalaria	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Mortalidad hospitalaria, el sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados consulta la entidad Registro de egresos y genera el reporte de información. El sistema permite imprimir el reporte y generar el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de Mortalidad hospitalaria dado criterios
Referencias:	RF 6

Tabla 1.8. CU_ Generar reportes estadísticos de Mortalidad hospitalaria

CU_ Generar reportes estadísticos de Natalidad hospitalaria	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Reporte de Natalidad hospitalaria, el sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados consulta la entidad Libro de la sala de partos y genera el reporte de información. El sistema permite imprimir el reporte y generar el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de natalidad hospitalaria dado criterios
Referencias:	RF 7

Tabla 1.9. CU_ Generar reportes estadísticos de Natalidad hospitalaria

CU_ Generar comportamiento de la morbilidad o mortalidad por patología	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar comportamiento de la morbilidad o mortalidad por patología, el sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados consulta la entidad Registro de egresos y genera el reporte de información. El sistema permite imprimir el reporte. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó el comportamiento de la morbilidad o mortalidad por patología, dado criterios.
Referencias:	RF 8

Tabla 1.10. CU_ Generar reportes estadísticos de la morbilidad o mortalidad por patología

CU_ Generar reporte de causas de cancelación de intervenciones quirúrgicas	
Actores:	TRES(Inicia)
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar reporte de causas de cancelación de intervenciones quirúrgicas, el sistema brinda la posibilidad de generar el reporte después de consultar la entidad Causas de suspensión de intervenciones quirúrgicas. El sistema permite imprimir el reporte y generar el gráfico correspondiente. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó un reporte de causas de cancelación de intervenciones quirúrgicas.
Referencias:	RF 9

Tabla 1.11. CU_ Generar reportes estadísticos de causas de cancelación de intervenciones quirúrgicas

CU_ Generar Consolidado de departamentos clínicos y servicios	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar Consolidado de departamentos clínicos y servicios, el sistema brinda la posibilidad de generar el reporte una vez consultadas las entidades Registro de admisiones, Registro de transferencias y Registro de egresos. El sistema permite imprimir el reporte. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó el Consolidado de departamentos clínicos y servicios.
Referencias:	RF 10

Tabla 1.12. CU_ Generar Consolidado de departamentos clínicos y servicios

CU_ Generar comportamiento de la natalidad	
Actores:	TRES
Resumen:	El caso de uso inicia cuando el actor accede a la opción Generar comportamiento de la natalidad, el sistema brinda la posibilidad de seleccionar los criterios para generar el reporte, el actor selecciona los datos que considera como criterios para generar el reporte, el sistema a partir de los criterios seleccionados consulta la entidad Libro de sala de parto y genera el reporte de información. El sistema permite imprimir el reporte. El caso de uso termina.
Precondición:	No existe.
Poscondición:	Se generó el comportamiento de la natalidad, dado criterios.
Referencias:	RF 11

Tabla 1.13. CU_ Generar comportamiento de la natalidad

En este capítulo se especificaron los procesos de negocio actuales en el área de Estadísticas de las instituciones hospitalaria y se realizó un análisis crítico de los mismos. Se propuso la concepción del sistema, apoyado en la especificación de los requerimientos funcionales y no funcionales. De igual forma fueron identificadas detalladamente las acciones a ejecutar en el sistema por cada uno de los actores. Dichas acciones fueron descritas mediante una descripción textual, además de estar apoyadas en la representación gráfica, que incorpora los actores y la relación entre éstos y los casos de usos definidos.

Capítulo 3: Análisis y Diseño

En el presente capítulo se describe la concepción arquitectónica del Sistema de Información Hospitalaria alas HIS y su aplicación en el módulo de estadísticas, con el objetivo de entender la estructura y dinámica de la organización del mismo. Se especifica el modelo de diseño del sistema y se realiza el refinamiento de las clases definidas, agrupadas en paquetes. De igual forma son descritas cada una de las clases y diagramas de clases, correspondientes al proceso involucrado, justificándose además el uso de los diferentes patrones.

3.1. Descripción de la arquitectura

Para el desarrollo del presente sistema fue puesto en práctica el uso del patrón arquitectónico Modelo-Vista-Controlador (MVC), dada la ventaja que presenta el mismo de separar los datos de una aplicación, la interfaz del usuario, y la lógica de control en tres componentes diferentes. Este patrón garantiza una cierta independencia entre los distintos componentes que permiten realizar cambios en cualquiera de ellos sin que los mismos afecten las otras capas. (22)

El uso de este patrón se pone en práctica mediante el uso del framework JSF o Java Server Face como sus siglas en inglés lo indican, el cual garantiza la recogida y representación de la información al usuario, e implementa de esta forma la vista; además permite controlar las acciones que el usuario desea realizar mediante las funcionalidades implementadas en las clases controladoras o Beans, las cuales se encargan de manipular la información contenida en el modelo.

La vista está desarrollada básicamente con JSF o Java Server Face como sus siglas en inglés lo indican y usan la librería de componentes Richfaces 3.2.0 G.A., la cual se integra fácilmente con el framework Seam. Además en esta capa están presentes controles out-of-the-box AJAX-ready, el framework de extensión AJAX, y Ajax4Jsf para los controles JSF básicos.

Las tecnologías utilizadas para el desarrollo de la aplicación, están directamente relacionadas con los frameworks, los cuales son de vital importancia para la integración de las herramientas usadas en la realización del sistema. La utilización del directorio jerárquico permitió encapsular los recursos compartidos, tales como una biblioteca dinámica, archivos de imagen, cadenas locales, archivos de cabecera, y documentación de referencia en un solo paquete, mediante los cuales el software fue organizado y desarrollado. Dicho framework brindó al sistema soporte de programas, bibliotecas, un lenguaje interpretado, entre otros software que ayudaron a desarrollar y unir los diferentes componentes del proyecto.

El uso de los frameworks permitió el hecho de que varias aplicaciones puedan utilizar todos estos recursos de manera simultánea, el sistema carga en memoria cuando es necesario y comparte la copia de uno de los recursos entre todas las solicitudes siempre que sea posible; provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio; además de que las distintas versiones se pueden incluir en el mismo paquete, lo que hace posible que sea compatible con los programas más antiguos.

En la vista, como se planteó anteriormente, el framework utilizado fue Seam. Esta potente plataforma de desarrollo de código abierto permitió la integración en el sistema, de tecnologías como Java Script asíncrono y XML con el uso de AJAX, Java Server Faces, Java Persistence, Enterprise Java Beans en su versión 3.0 y Business Process Management en una única solución completa con sofisticadas herramientas.

Seam fue escogido para la aplicación con el fin de eliminar la complejidad tanto en la arquitectura como en los niveles de la API, lo que permite a los desarrolladores ensamblar la compleja aplicación web con el uso de simples clases Java, un rico conjunto de componentes de la interfaz de usuario, y muy poco de XML. Además, Seam brindó un singular apoyo a las conversaciones y la administración del estado declarativo lo cual dio lugar a introducir una experiencia de usuario más sofisticada.

En la vista también se utilizó el Lenguaje de Marcado de Hipertexto Extensible, con el objetivo de reemplazar a HTML ante su limitación de uso de las cada vez más abundantes herramientas basadas en XML, lo que aprovecha además la ventaja de este lenguaje, de permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella.

Otro de los lenguajes utilizados en la capa de presentación es el Java Script, gracias a su característica de ser un lenguaje de programación interpretado, es decir, que no requiere compilación; lo cual permite ejecutar código en el lado del cliente contribuyendo con la rapidez en la conexión que se establece entre el cliente y el servidor. (23)

También en la vista son utilizadas las Hojas de Estilo en Cascada, por su capacidad de identificar cómo se va a mostrar un documento en la pantalla, cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en el documento, a través de un dispositivo de lectura permite, a los desarrolladores controlar el estilo y el formato de múltiples páginas Web al mismo tiempo, así como lograr la organización y legibilidad que se ha logrado en el sistema.

Igualmente, otro de los lenguajes vistos en la interfaz es el facelets, usado para construir un árbol de componentes que le permite al sistema ser reutilizable, de manera que se definieron componentes como composición de otros componentes.

JSF o Java Server Faces es un framework para aplicaciones Java basadas en web el cual se utilizó para la realización de las interfaces de usuario del software en desarrollo, y permitió hacer más simplificado el desarrollo de las mismas para la realización del prototipo no funcional. JSF también dio la posibilidad de hacer el despliegue de las páginas, incluyendo plugins que les fueron incluidos gracias a su compatibilidad con los mismos.

Entre las principales ventajas que brindaron dichos plugins se citan algunas como: un conjunto de APIs para representar los componentes de las interfaces de usuario realizadas y administrar su estado, manejar eventos, validar la entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad; asimismo un conjunto por defecto de componentes para las interfaces de usuarios realizadas; y por último dos librerías de etiquetas personalizadas para Java Server Pages que le permitieron al sistema expresar una interfaz Java Server Faces dentro de una página Java Server Pages.

Richfaces es un framework desarrollado para potenciar Java Server Faces con el cual se puede crear un ambiente AJAX de manera fácil, rápida y limpia (24). Es decir, gracias a esta técnica de desarrollo web la aplicación se hizo interactiva sin un elevado esfuerzo de manos de obra y en un tiempo corto.

En el sistema que se desarrolla están unificadas las tecnologías estándares utilizadas: Java Server Faces, Enterprise Java Beans en su versión 3.0 y Java Persistence API, además de la metodología Business Process Management, gracias al framework de integración Seam, lo cual permite crear una capa consistente que impide todo tipo de errores.

En el controlador además se encuentra la Notación para el Modelado de Procesos de Negocio en Java, de inglés JBPMN, que ha estado presente en los diagramas hechos representados en el entorno de desarrollo Eclipse, lo que ha permitido que dichos diagramas sean más legibles y mejores de entender, al ser expresados a través de una amplia gama de elementos que manifiestan su utilidad en la detallada descripción presentada.

Para el acceso a datos, en el modelo, se utiliza la implementación de Java Persistence API de Hibernate 3.3, y son usados los servicios del contenedor de Enterprise Java Beans en su versión 3.0 y/o los contextos de persistencias administrados por Seam. En esta capa están presentes varios

patrones arquitectónicos utilizados en el desarrollo del producto, los citados son: Active record, Identity map, Identity field, Foreign key mapping, Association Table, mapping, Lazy load y Query object.

Hibernate en su versión 3.3, es el framework de persistencia utilizado, este hizo más sencillo el trabajo con las bases de datos relacionales de la aplicación, lo cual permite a los desarrolladores tener una simulación como si la base de datos trabajara directamente con objetos, lo que ha contribuido a que el equipo de trabajo del complejo sistema actualmente en desarrollo, se preocupe por el negocio de la aplicación solamente y no del cómo se persiste, recupera y elimina la información de la base de datos.

El sistema a su vez puede generar completamente las sentencias SQL de inserción, modificación, borrado y cargado de los objetos de la clase, lo que permite que las clases persistidas pudiesen ser simples POJOs. Al utilizar esta tecnología se minimizaron por un lado las configuraciones en XML sin chequeo de tipos y por otro lado el uso de los servicios del contenedor de Enterprise Java Beans en su versión 3.0, se eliminó gran parte del código “infraestructural” en cuanto a transacciones, transmisión del contexto de persistencia, etc. Además de que se pudieron establecer validaciones end-to-end gracias a los Hibernate Validators o “validadores de hibernate”.

3.2. Modelo de diseño

El modelo de diseño representa la primera entrada para el desarrollo de la implementación. Tiene como objetivo producir un modelo lógico del sistema a implementar. Incluye los requisitos no funcionales y restricciones impuestas por el lenguaje de programación. En el mismo se identifican las clases del diseño y la realización de los casos de usos, constituyendo estos en su conjunto el modelo de diseño.

Para garantizar una óptima representación del modelo de diseño, se realizan tres actividades fundamentales, como: diseñar la arquitectura, diseñar casos de usos (identificar clases del diseño e interacciones entre objetos) y diseñar clases del diseño (identificar operaciones, atributos y relaciones). La concepción del modelo de diseño, está en total correspondencia con los patrones a utilizar.

3.2.1. Fundamentación del uso de patrones

Desde el punto de vista general, un patrón es un modelo que se puede seguir para la realización de una actividad determinada. Estos surgen de la experiencia de los seres humanos motivados por tratar de lograr ciertos objetivos. Existen distintos tipos de patrones, que son aplicables en dependencia de lo

que se desea, en este caso particular serán abordados un conjunto de patrones entre los que se encuentran, los patrones de arquitectura y diseño.

Específicamente en el contexto que se analiza, un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular recurrente que surge en un sistema determinado.

Dichos patrones arquitectónicos, presentan características que contribuyen a obtener una visión completa de las funcionalidades representadas que son traducidas en ventajas. Las mismas se enuncian a continuación:

- ❖ Documentan experiencias de diseño existentes y bien probadas.
- ❖ Identifican y especifican abstracciones de alto nivel.
- ❖ Proveen un vocabulario común y comprensible.
- ❖ Son una forma de documentar la arquitectura del software.
- ❖ Facilitan la construcción de software con propiedades definidas.
- ❖ Ayudan a construir arquitecturas heterogéneas y complejas.
- ❖ Atacan problemas recurrentes que ocurren en situaciones específicas y dan una solución.

Por otra parte un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Estos identifican: clases, instancias, roles, colaboraciones y distribución de responsabilidades.

Ventajas:

- ❖ Proponen una forma de reutilizar la experiencia de los desarrolladores.
- ❖ Están basados en la recopilación del conocimiento de los expertos en desarrollo de software.
- ❖ Es una experiencia real, probada y que funciona por lo que ayuda a no cometer los mismos errores.

El patrón arquitectónico utilizado en el desarrollo de la solución, fue el Modelo Vista Controlador (MVC), el mismo separa los datos de la aplicación, la interfaz de usuario y la lógica de control entres componentes distintos: modelo, vista y controlador.

El modelo representa las estructuras de datos, casi siempre contiene funciones para consultar, insertar y actualizar información de la base de datos; la vista es la información que se presenta al usuario, puede ser una página web o una parte de una página; y el controlador es quien actúa como intermediario entre el modelo, la vista y cualquier otro recurso necesario para crear una página web.

Para el diseño del sistema se tuvieron en cuenta los patrones GRASP, encargados de asignar las responsabilidades a las clases presentes en el software. En el sistema se le asignó a cada clase las funcionalidades que podía realizar según la información que las mismas contienen para, de esta forma, no sobrecargar las clases y que cada una realice las operaciones que les corresponde. Se le da la responsabilidad de crear instancias de otras clase solamente a aquellas que contengan a las mismas, para definir, de esta forma quienes son los responsables de crear un objeto de otra clase, poniéndose de manifiesto los patrones experto y creador.

De igual forma fueron modeladas las clases controladoras las cuales son las encargadas de realizar las operaciones del sistema. Dichas operaciones reflejan los procesos del sistema por lo que no es factible manejarse en la capa de interfaz o datos. Además el diseño permite que las distintas clases del sistema interactúen entre sí, sin afectar esto la reutilización de las mismas o el correcto funcionamiento de las clases por separado, con el objetivo de cumplir, de esta forma, con los patrones de bajo acoplamiento y alta cohesión.

3.2.2. Estructura del diseño

En vistas a garantizar una mayor organización en el modelo de diseño, se estableció el agrupamiento por paquetes. Estos agrupan las clases y realizaciones de los casos de usos con las relaciones existentes entre ellos.

Repositorio de clases: En este paquete están contenidas las clases definidas en el diseño de acuerdo a la tecnología, que serán usadas en la implementación del módulo.

Entidades: Contiene las entidades autogeneradas y personalizadas de la base de datos.

- Autogeneradas: Entidades que se autogeneran desde la base de datos (generadas con el ORM Hibernate).

- Personalizadas: Entidades personalizadas, que son modificadas. Estas pueden heredar de las entidades autogeneradas (Las relaciones pueden ser de herencia o composición).

Sesiones: Contiene las clases controladoras autogeneradas, personalizadas y las específicas o propias creadas para la implementación del proceso.

- Controladoras autogeneradas: Clases controladoras autogeneradas por el entorno de desarrollo.
- Controladoras personalizadas: Clases controladoras personalizadas que son modificadas.
- Controladoras del proceso: Clases controladoras propias del proceso.

Vistas: Contiene las vistas del diseño

Generar reporte Estadísticos: Representa el proceso generar reportes estadístico y dentro del paquete se representan los paquetes que contienen las realizaciones de los casos de usos que contiene el proceso, el nombre de cada paquete se corresponde con los casos de uso del mismo.

El siguiente diagrama representa la estructura concebida en el diseño.

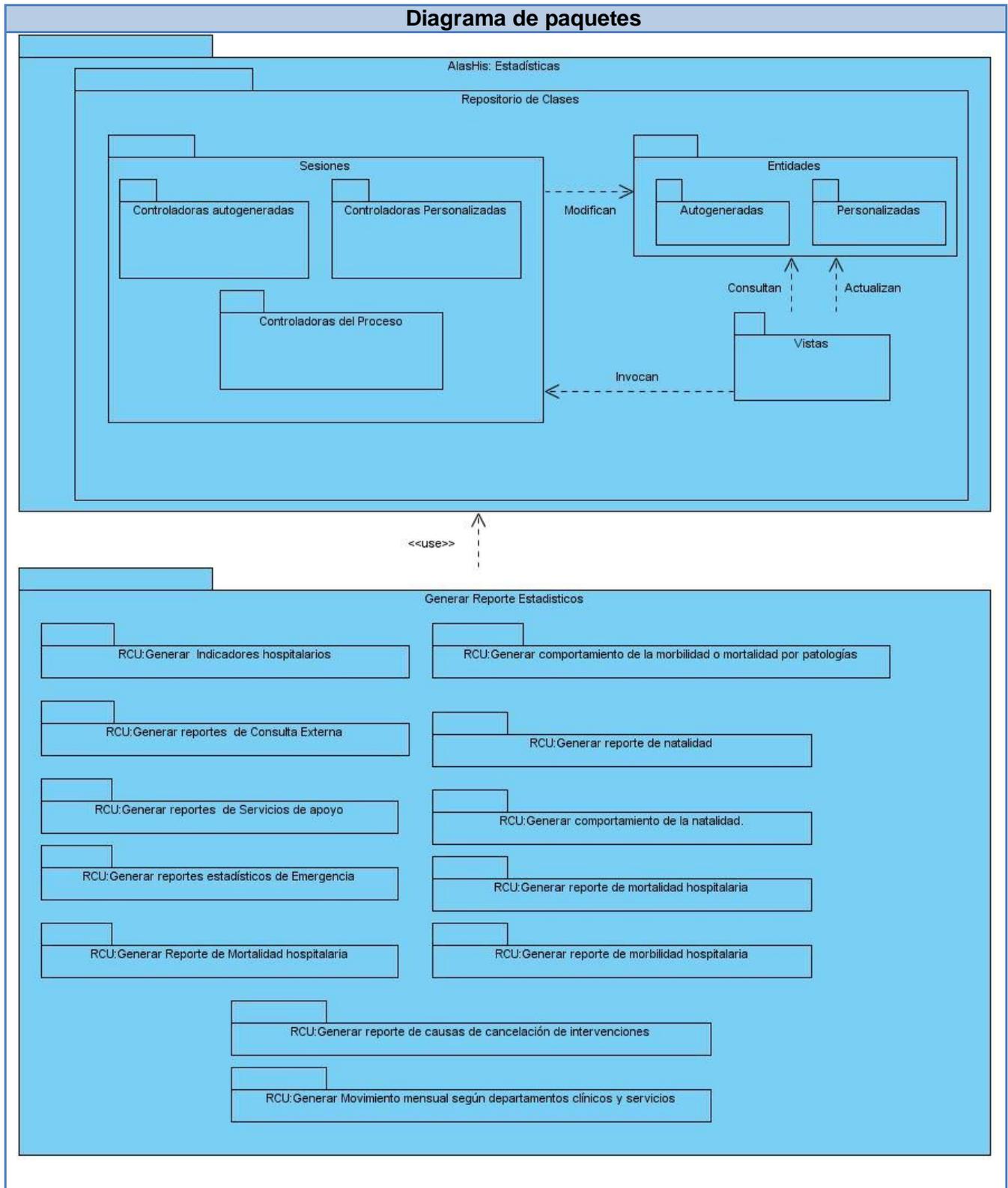


Figura 1.5. Diagrama de paquetes

3.2.3. Definición de elementos de diseño

En la realización de los diagramas de clases del diseño se representan tres estereotipos de clases fundamentales: Server Page o página servidora, la cual se encarga de construir o generar la página cliente; las Client Page o páginas clientes encargadas de mostrar la información al usuario e interactuar con el mismo, las cuales son creadas por una única Server Page y pueden presentar algún vínculo con otras páginas clientes y los Form o formularios que son los que se encargan de enviar los datos a la página servidora para que sean procesados los pedidos, además estos están contenidos en las Client Page.

Con el objetivo de lograr una mayor organización en el diseño, se define la siguiente nomenclatura para nombrar las clases del mismo:

Nomenclatura de las clases del diseño	
Página cliente	Página CL_<NombreClaseCliente>
Formulario	Fomulario_<NombreFormulario>
Página servidora	SP_<NombreClaseServidora>

Tabla 1.14. Nomenclatura de las clases del diseño

A continuación se muestra una tabla con las posibles relaciones existentes entre las distintas clases del diseño:

Desde	Hasta	Client Page	Form	Server Page
Client Page (Página cliente)		<<Link>> , <<redirect>>	Contiene	<<Link>>, <redirect>>
Form (Formulario)		Agregado por.	---	<<Submit>>
Server Page (Página servidora)		<<Build>>, <<redirect>>		<<Redirect>>

Tabla 1.15. Relaciones entre clases del diseño

3.2.4. Diagramas de clases del diseño

Un diagrama de clases de diseño muestra la especificación para las clases de una aplicación, donde se incluye: clases, asociaciones y atributos; interfaces con sus operaciones y constantes; métodos; navegabilidad y dependencias. A diferencia del modelo conceptual, un diagrama de clases de diseño muestra definiciones de entidades software más que conceptos del mundo real.

A continuación son mostrados varios diagramas de clases del presente flujo de trabajo correspondientes a varios casos de uso:

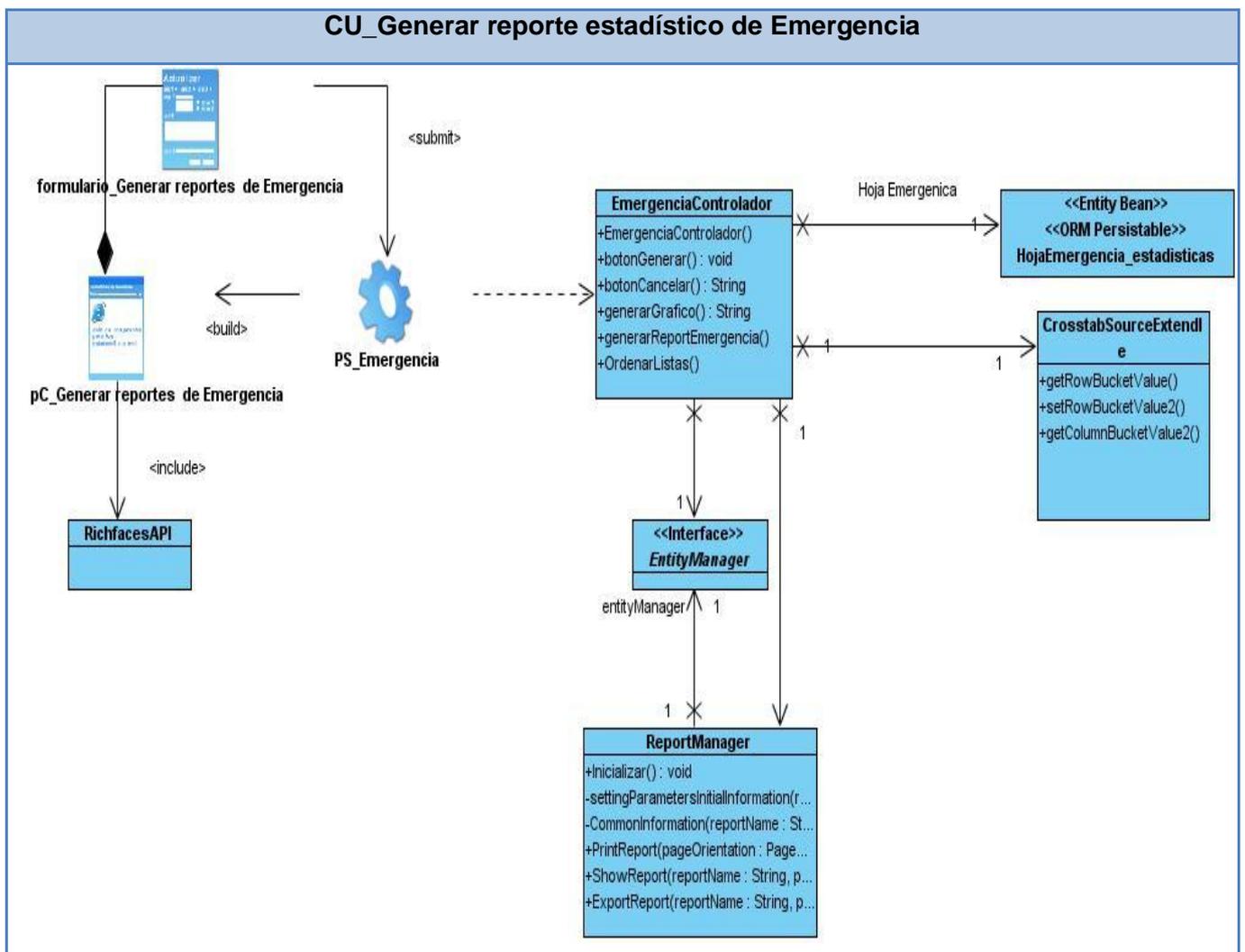


Figura 1.6. Diagrama de Clases del Diseño. CU_ Generar reporte estadístico de Emergencia.

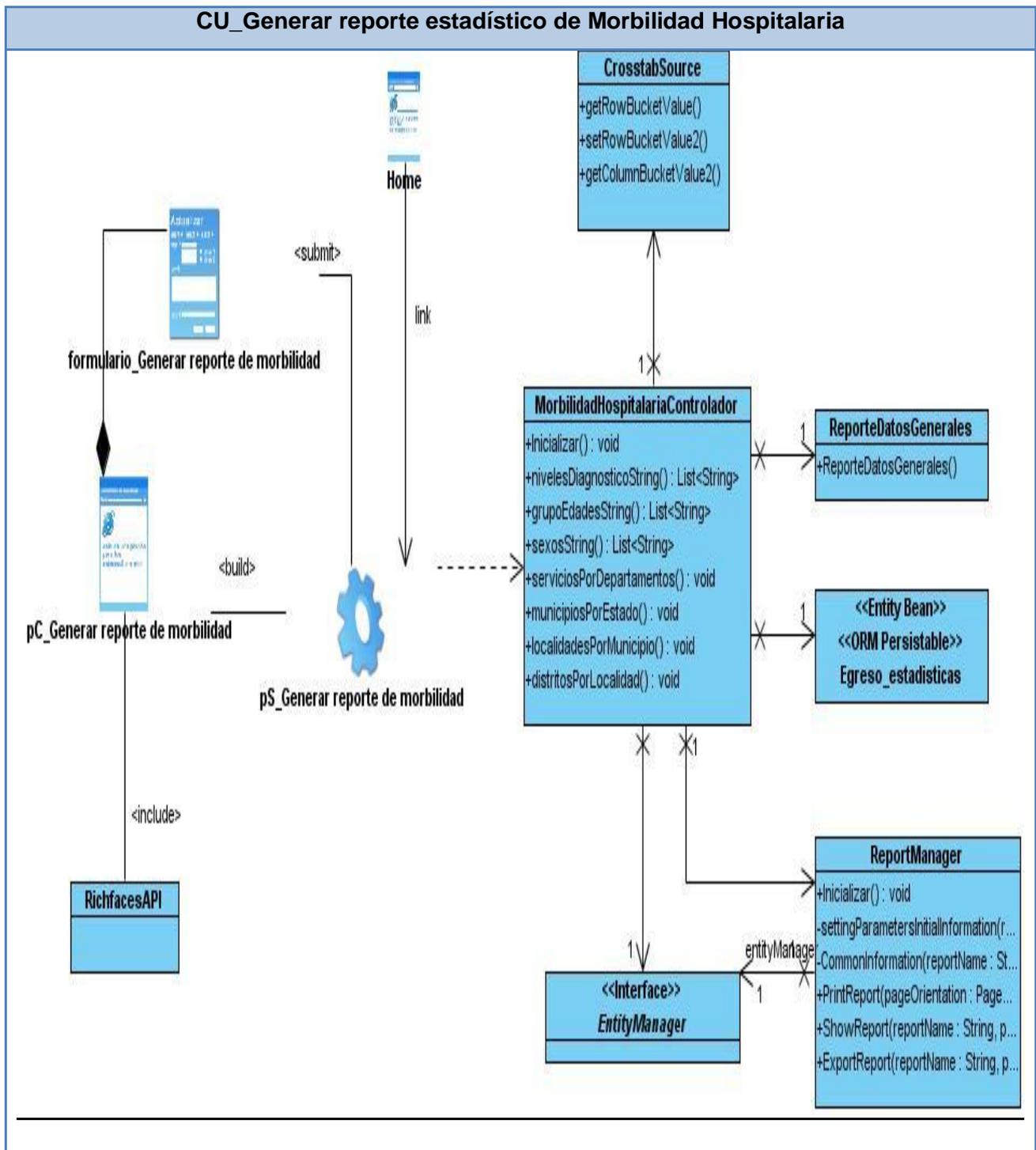


Figura 1.7. Diagrama de Clases del Diseño. CU_Generar reporte estadístico de Morbilidad Hospitalaria.

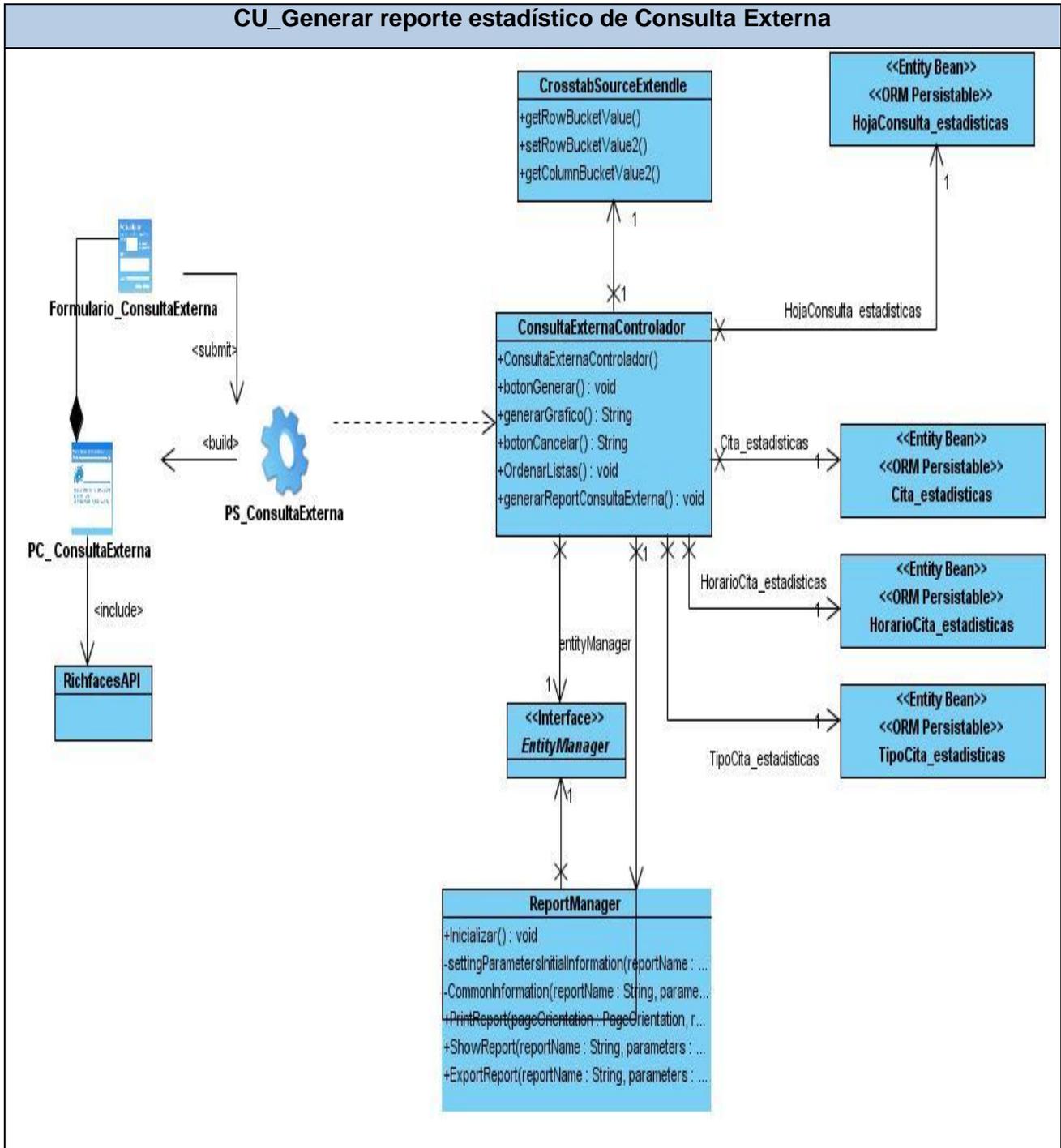


Figura 1.8. Diagrama de Clases del Diseño, CU_Generar reporte estadístico de Consulta Externa.

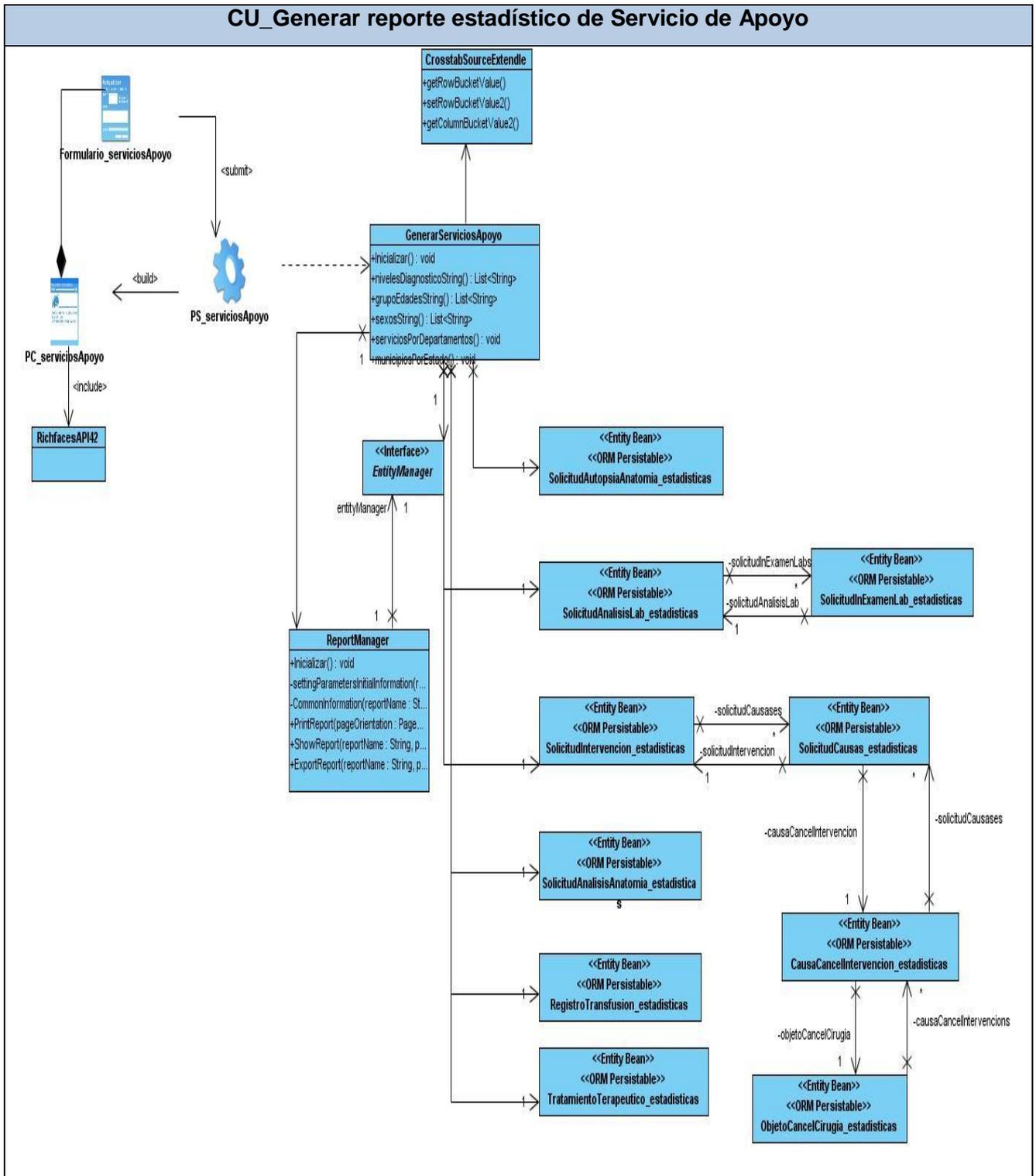


Figura 1.9. Diagrama de Clases del Diseño. CU_ Generar reporte estadístico de Servicio de Apoyo.

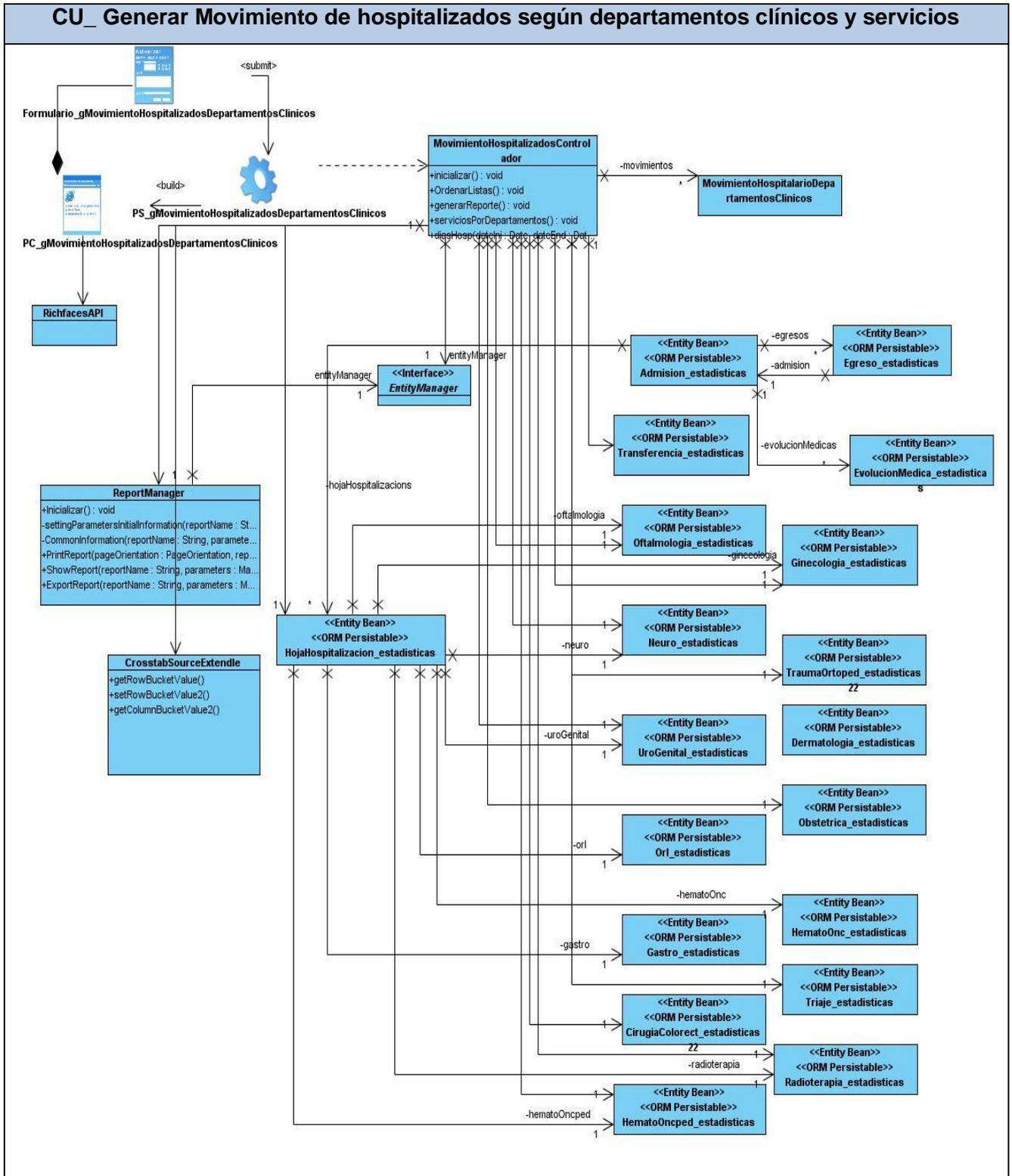


Figura 1.10. Diagrama de Clases del Diseño. CU_ Generar Movimiento de hospitalizados según departamentos clínicos y servicios.

3.2.5. Diagramas de interacción

Los diagramas UML de secuencia y de colaboración (diagramas de interacción) se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. El diagrama de secuencia se centra en el tiempo, mientras que el diagrama de colaboración se centra en el espacio.

A continuación son mostrados varios diagramas de secuencia correspondientes a las realizaciones de casos de usos descritas:

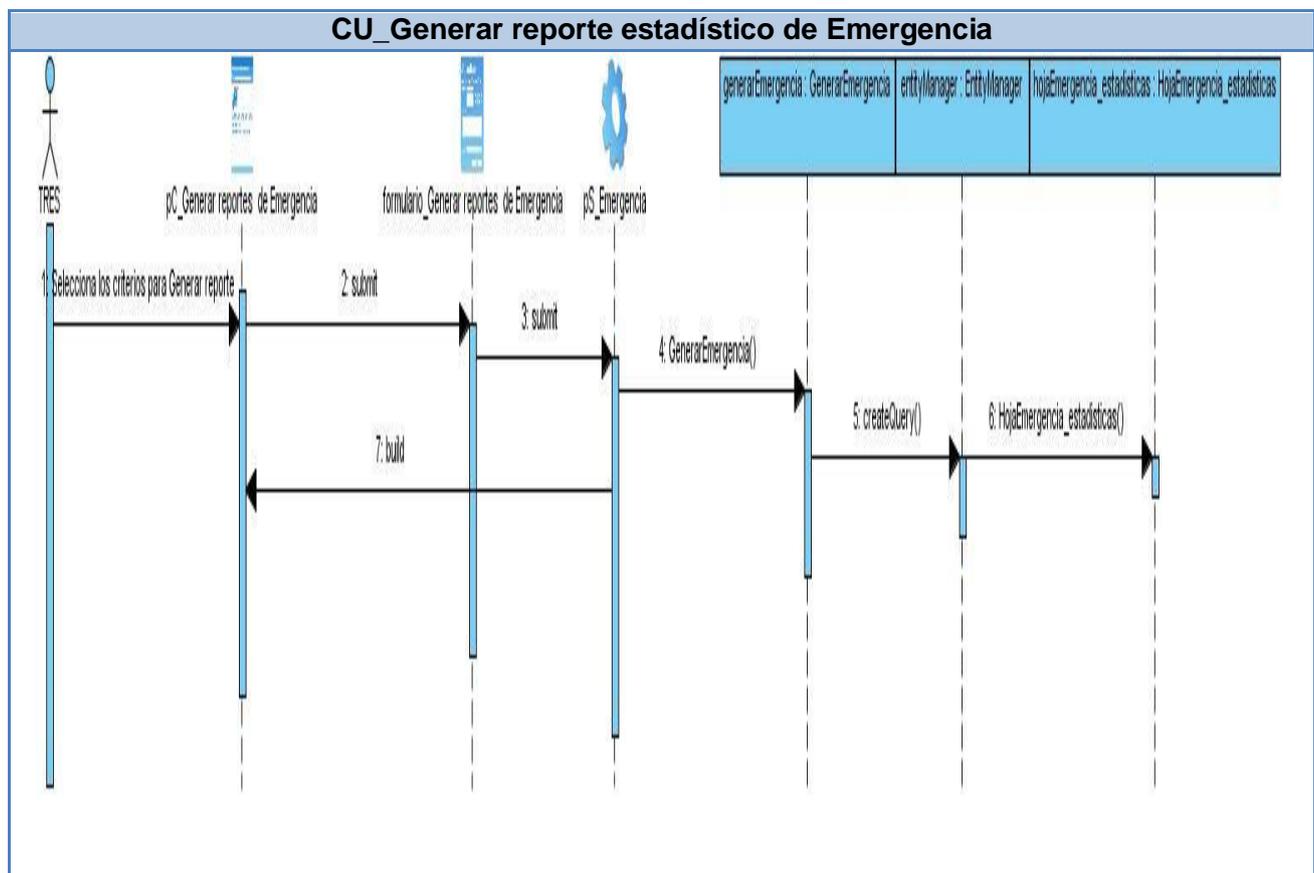


Figura 1.11. Diagrama de secuencia. CU_Generar reporte estadístico de Emergencia.

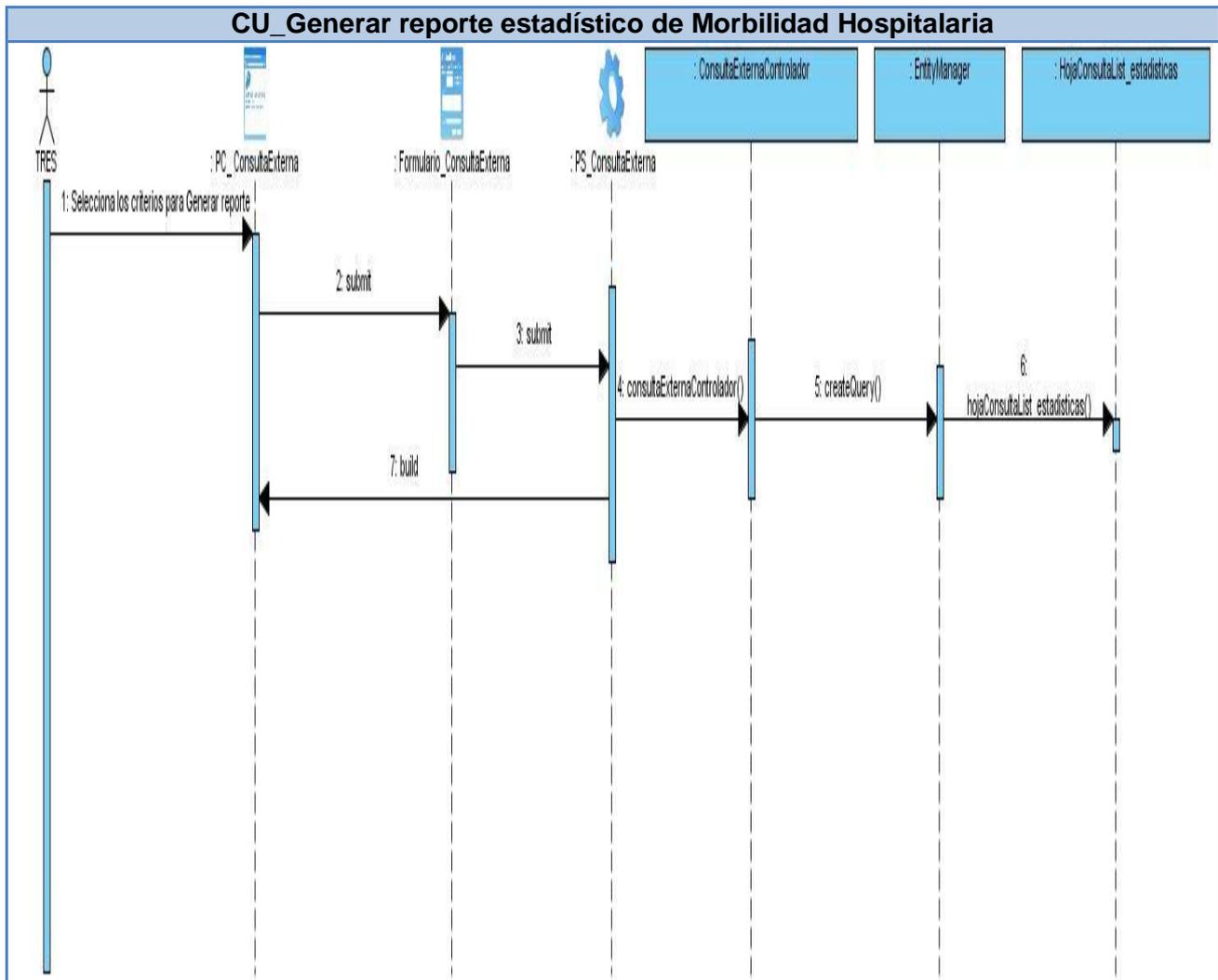


Figura 1.12. Diagrama de secuencia. CU_Generar reporte estadístico de Morbilidad Hospitalaria.

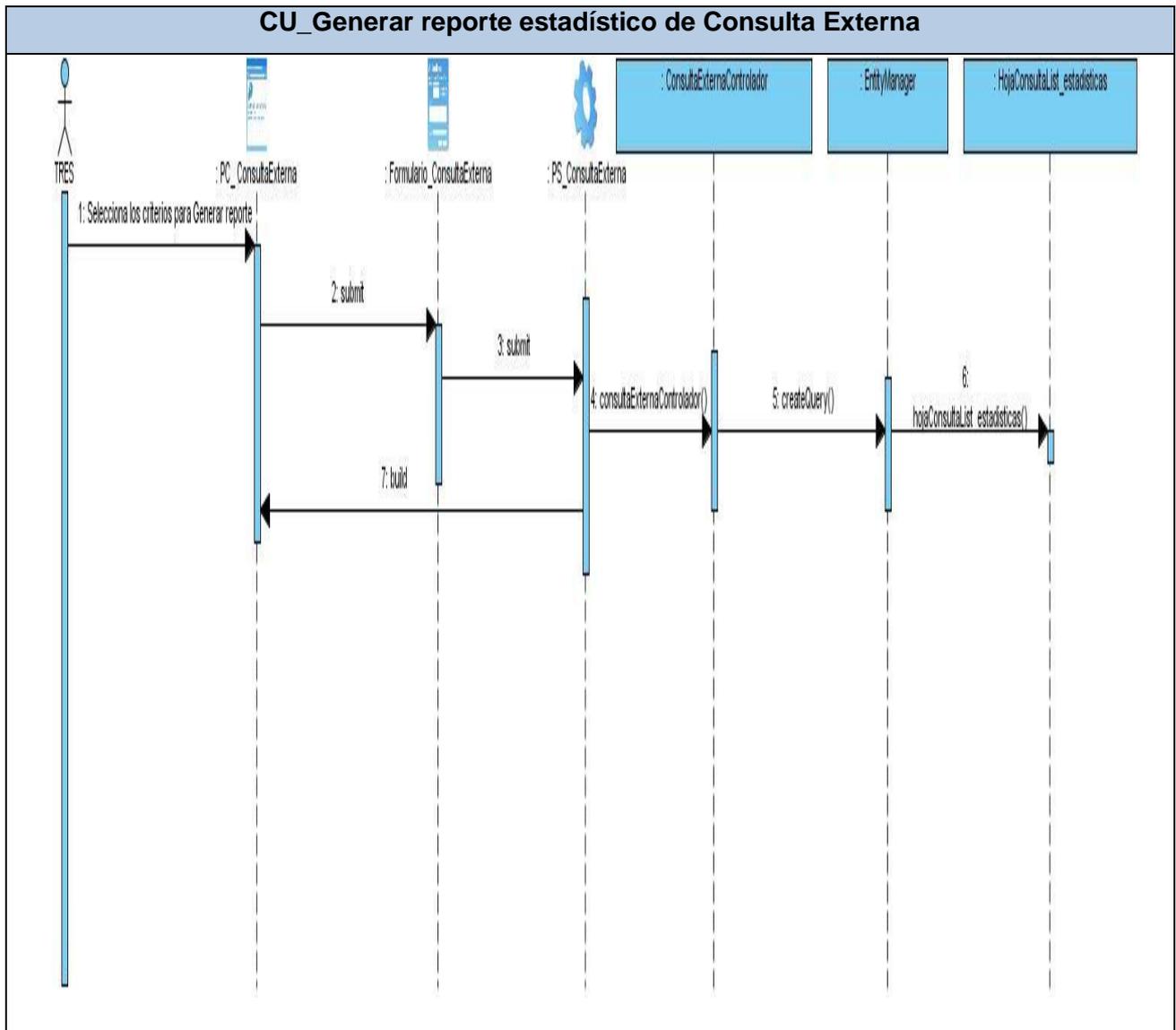


Figura 1.13. Diagrama de secuencia. CU_Generar reporte estadístico de Consulta Externa.

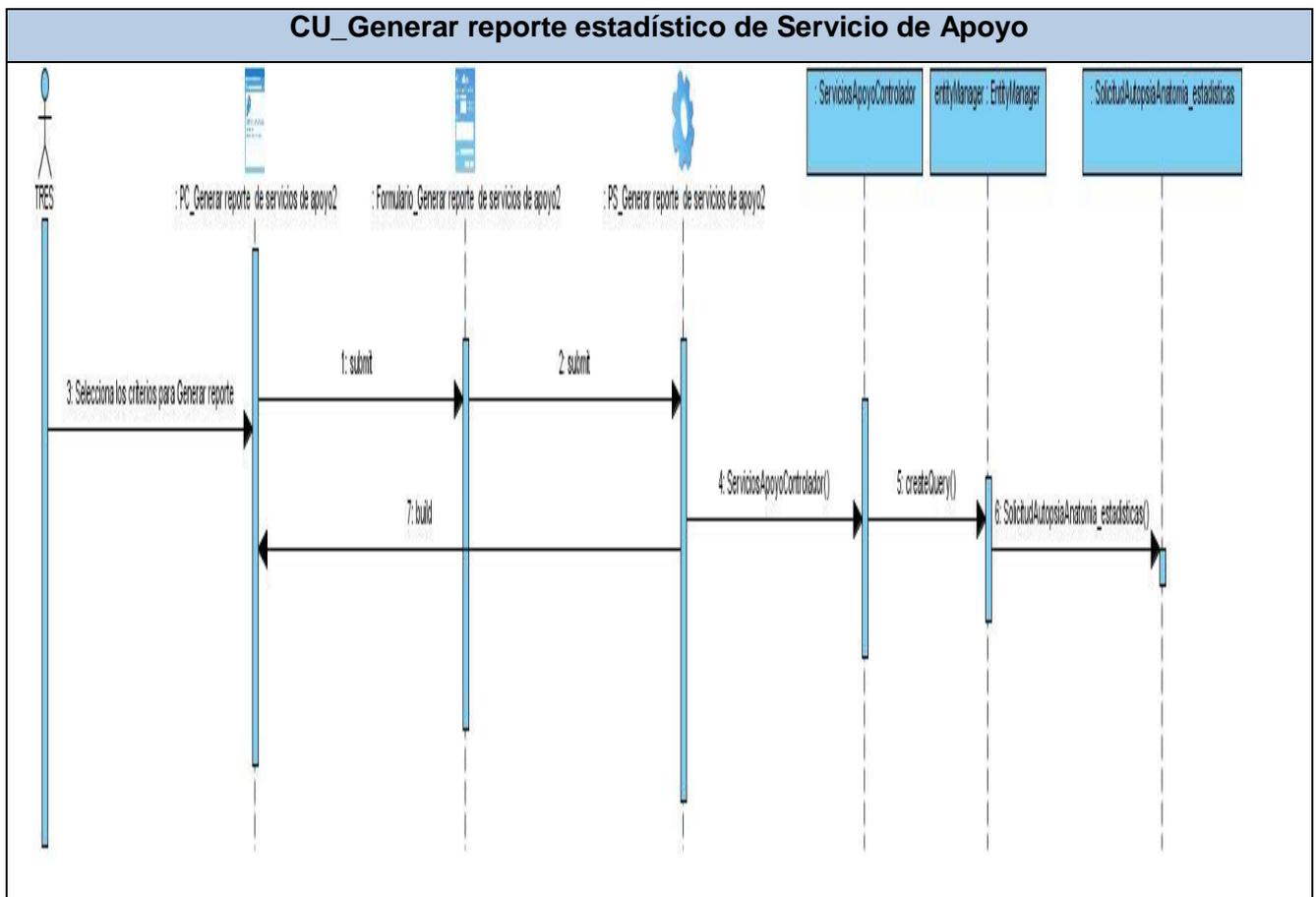


Figura 1.14. Diagrama de secuencia. CU_ Generar reporte estadístico de Servicio de Apoyo.

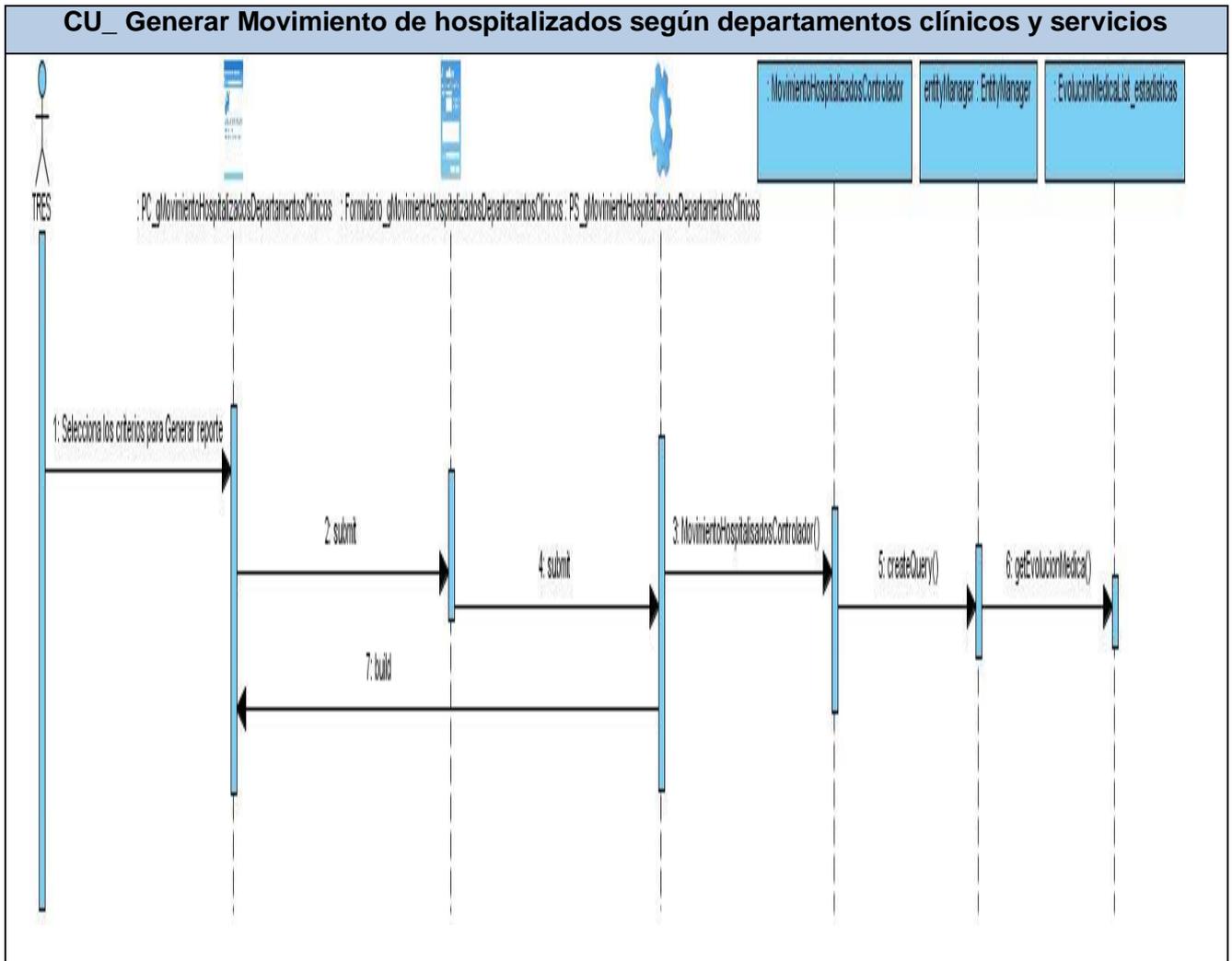


Figura 1.15. Diagrama de secuencia. CU_ Generar Movimiento de hospitalizados según departamentos clínicos y servicios

3.2.6. Descripción de las clases

3.2.6.1. Interfaces

Nombre: Emergencias
Tipo de clase (interfaz)
Descripción General: Le permite al usuario seleccionar los criterios por los cuales desea realizar el reporte de emergencia y le muestra los resultados obtenidos en el mismo, además le brinda la opción de imprimir el reporte y visualizar el grafico correspondiente.

Tabla 1.16. Clase interfaz Emergencias

Nombre: gMorbilidadHospitalaria
Tipo de clase (interfaz)
Le permite al usuario seleccionar los criterios por los cuales desea realizar el reporte de la morbilidad hospitalaria y le muestra los resultados obtenidos en el mismo, además le brinda la opción de imprimir el reporte.

Tabla 1.17. Clase interfaz gMorbilidadHospitalaria

Nombre: ConsultaExterna
Tipo de clase (interfaz)
Descripción General: Le permite al usuario seleccionar los criterios por los cuales desea realizar el reporte de consulta externa y le muestra los resultados obtenidos en el mismo, además le brinda la opción de imprimir el reporte.

Tabla 1.18. Clase interfaz ConsultaExterna

Nombre: serviciosApoyo
Tipo de clase (interfaz)
Descripción General: Le permite al usuario seleccionar los criterios por los cuales desea realizar el reporte de los servicios de apoyo y le muestra los resultados obtenidos en el mismo, además le brinda la opción de imprimir el reporte.

Tabla 1.19. Clase interfaz serviciosApoyo

Nombre: gMovimientoHospitalizadosDepartamentosClinicos
Tipo de clase (interfaz)
Descripción General: Le permite al usuario seleccionar los criterios por los cuales desea realizar el reporte de los movimientos de los hospitalizados por todos los departamentos clínicos y le muestra los resultados obtenidos en el mismo, además le brinda la opción de imprimir el reporte.

Tabla 1.20. Clase interfaz gMovimientoHospitalizadosDepartamentosClinicos

3.2.6.2. Controladoras

Nombre: GenerarEmergencia	
Tipo de clase (controladora)	
Atributo	Tipo
targetMeses	List<String>
sourceMeses	List<String>
fecha	Date
rptEmergencias	Boolean
reporteEmergencia	ReporteIndicadoresHospitalarios
Para cada responsabilidad:	
Nombre:	botonGenerar()
Descripción:	Manda a ejecutar el reporte
Nombre:	getReporteEmergencia()
Descripción:	Llena la lista con los valores que va a mostrar el reporte y los retorna
Nombre:	generarGrafico()
Descripción:	Redirecciona para la página de generar gráfico y envía los parámetros.

Tabla 1.21. Clase controladora GenerarEmergencia

Nombre: GenerarMorbilidadHospitalaria	
Tipo de clase (controladora)	
Atributo	Tipo

reporteMorbilidadhospitalaria	Boolean
fecha	Date
procedencia	String
ocurrenciachecked	Boolean
procedenciachecked	Boolean
Para cada responsabilidad:	
Nombre:	botonGenerar()
Descripción:	Manda a ejecutar el reporte
Nombre:	generarGrafico()
Descripción:	Redirecciona para la pagina de generar grafico y envía los parámetros.

Tabla 1.22. Clase controladora GenerarMorbilidadHospitalaria

Nombre: ReporteMorbilidadHospitalaria	
Tipo de clase (controladora)	
Atributo	Tipo
morbilidad	List<MorbilidadHospitalaria>
total	MorbilidadHospitalaria
Para cada responsabilidad:	
Nombre:	getMorbilidad()
Descripción:	Llena la lista con los valores que va a mostrar el reporte y los retorna.

Tabla 1.23. Clase controladora ReporteMorbilidadHospitalaria

Nombre: GenerarEstConsultaExterna	
Tipo de clase (controladora)	
Atributo	Tipo
targetMeses	List<String>>
sourceMeses	List<String>
fecha	Date

reporteConsultaExterna	ReporteIndicadoresHospitalarios
reporteCE	Boolean
Para cada responsabilidad:	
Nombre:	getReporteConsultaExterna()
Descripción:	Llena la lista con los valores que va a mostrar el reporte y los retorna.

Tabla 1.24. Clase controladora GenerarEstConsultaExterna

Nombre: GenerarServiciosApoyo	
Tipo de clase (controladora)	
Atributo	Tipo
reporteRadiologia	ReporteIndicadoresHospitalarios
reporteImagenologia	ReporteIndicadoresHospitalarios
reporteAnatomiaPatologica	ReporteIndicadoresHospitalarios
reporteLaboratorio	ReporteIndicadoresHospitalarios
reporteQuirofono	ReporteIndicadoresHospitalarios
reporteSalaPartos	ReporteIndicadoresHospitalarios
reporteBancoSangre	ReporteIndicadoresHospitalarios
Para cada responsabilidad:	
Nombre:	getReporteRadiologia
Descripción:	Llena la lista con los valores de radiología que va a mostrar el reporte y los retorna.
Nombre	getReporteImagenologia
Descripción:	Llena la lista con los valores de imagenologia que va a mostrar el reporte y los retorna.
Nombre	getReporteAnatomiaPatologica
Descripción:	Llena la lista con los valores de anatomía Patológica que va a mostrar el reporte y los retorna.
Nombre	getReporteLaboratorio
Descripción	Llena la lista con los valores de laboratorio que va a mostrar el reporte y los retorna.

Tabla 1.25. Clase controladora GenerarServiciosApoyo

Nombre: ReporteMovHospDepartClinicos	
Tipo de clase (controladora)	
Atributo	Tipo

reportMovimientos	ReporteMovHospDepClinicos
Para cada responsabilidad:	
Nombre:	getReportMovimientos
Descripción:	Llena la lista con los valores que va a mostrar el reporte y los retorna.

Tabla 1.26. Clase controladora ReporteMovHospDepartClinicos

En este capítulo se describió la arquitectura del sistema, así como los patrones arquitectónicos y de diseño utilizados en el desarrollo de la solución. Fue especificada la estructura del diseño, la cual contempla las realizaciones de los casos de usos y las clases correspondientes. De estas últimas, se realizó una breve descripción según las clasificaciones establecidas.

Capítulo 4: Implementación

En el presente capítulo son vistos los principales elementos de la implementación del sistema. Se obtienen los principales artefactos de los diagramas de Despliegue, Componentes. Así como el Modelo de datos, este muestra las principales tablas de la base de datos que utiliza el área de estadísticas del Sistema de Información Hospitalaria alas HIS.

4.1. Modelo de Datos

Un modelo de datos es básicamente una "descripción" de lo que se conoce como contenedor de datos (donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no describen estructuras físicas, sino que son abstracciones que permiten la implementación de un sistema de base de datos eficiente. (25) (Ver Figura 1.13)

4.1.1. Descripción de las tablas

Las tablas de la bases de datos, designan un tipo de modelado de datos, donde se guarda la información recogida por el sistema, especialmente las estadísticas provenientes de cada una de las áreas del hospital. Las tablas se componen de dos estructuras:

- ❖ Campo o atributo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.
- ❖ Registro: Corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

En cada campo, existe un nombre único, con su tipo de dato correspondiente, esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí. Cada tabla creada contiene un nombre único en la base de datos, haciéndola accesible mediante su nombre. La estructura de estas se define por la forma de un archivo plano. Ver Tabla 1.29

En la siguiente tabla, se muestran un conjunto de campos que son comunes a todas las tablas de la base de datos del Sistema de Información Hospitalaria en desarrollo:

Atributo	Tipo	Descripción
version	int	Contiene un identificador de la versión, para que no ocurran errores al trabajar varias personas a la vez con el sistema.
eliminado	bool	Contiene la información booleana que declara si aparece o no la información de manera visible para el usuario.
cid	int	Contiene el id de la conversación, permite, de esta manera, darle un rango de vida a la sección en la cual se trabaje.

Tabla 1.27. Entidades comunes

Nombre: Hospital		
Descripción: Contiene elementos de información administrativa, así como el tipo de hospital, su ubicación y el área de población que atiende.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
nombre	String	Nombre de la institución hospitalaria.
poblacion_area_influencia	int	Cantidad de personas que atiende la institución.
direccion	string	Dirección del lugar de ubicación donde se encuentra el hospital.
telefonos	string	Número telefónico de la institución hospitalaria.
fax	string	Fax de la institución hospitalaria.
correo	string	Dirección de correo de la institución hospitalaria.
fecha_apertura	date	Fecha de apertura de la institución hospitalaria.
camas_arquitectonicas	integer	Cantidad de camas arquitectónicas existentes en la institución hospitalaria.
camas_presupuestadas	integer	Cantidad de camas presupuestadas existentes en la institución hospitalaria.
es_publico	bool	Contiene información afirmativa si la institución hospitalaria es pública y negativa en caso contrario.

Tabla 1.28. Entidad Hospital

Nombre: Servicio		
Descripción: Contiene información acerca de cada servicio que brinda la institución hospitalaria.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
nombre	String	Contiene el nombre de un servicio.
tiene_consulta_externa	bool	Contiene información acerca de si el servicio presenta consulta externa o no.
tiene_hospitalizacion	bool	Contiene información acerca de si el servicio presenta hospitalización o no.
activo	bool	Contiene información acerca de si el servicio está disponible o no.
codigo	String	Este es un identificador utilizado para saber la tabla a la cual se hace referencia en caso de ocurrir algún tipo de cambio, especialmente de idioma.

id_servicio_padre	int	Contiene información acerca del id que posee el padre del servicio en caso de que la institución tenga uno o varios servicios que a la vez contengan otros.
-------------------	-----	---

Tabla 1.29. Entidad Servicio

Nombre: Admision		
Descripción: Contiene toda la información necesaria concerniente a la admisión del paciente.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
fecha	date	Contiene información acerca de la fecha en que fue realizada la admisión.
observaciones	text	Contiene las observaciones que emite el técnico acerca del paciente.
peso_ingreso	BigDecimal	Brinda información acerca del peso que posee el paciente al ingresar.

Tabla 1.30. Entidad Admisión

Nombre: Diagnostico_medico		
Descripción: Contiene la información referente al diagnóstico realizado al paciente por parte del médico.		
Atributo	Tipo	Descripción
id	int 4	Identificador de la tabla.
fecha_emision	date	Contiene información acerca de la fecha en que es emitido el diagnóstico médico.
hora_emision	time	Contiene información acerca de la hora en que es emitido el diagnóstico médico.

Tabla 1.31. Entidad Diagnostico_medico

Nombre: Diagnostico_medico_enfermedad		
Descripción: Contiene la información referente a las enfermedades existentes.		
Atributo	Tipo	Descripción
id	int 4	Identificador de la tabla.
codigo_enfermedad	string(10)	Contiene el código de la enfermedad que se que se atiende.
descripcion_enfermedad	text	Contiene una descripción de la enfermedad que

		se atiende.
--	--	-------------

Tabla 1.32. Entidad Diagnostico_medico_enfermedad

Nombre: Egreso		
Descripción: Contiene toda la información relacionada con el egreso del paciente.		
Atributo	Tipo	Descripción
id	int 4	Identificador de la tabla.
admision	Admision_estadisticas	Es el objeto que almacena los datos del paciente cuando es ingresado en el hospital.
diagnosticoMedico	DiagnosticoMedico_estadisticas	Es el objeto que almacena los diagnósticos emitidos al paciente en el periodo que fue atendido.
tipoEgreso	TipoEgreso_estadisticas	Se almacena el tipo de egreso que le fue realizado al paciente, el cual puede ser por fallecimiento, o por alta.
fecha	date	Fecha en la que fue emitido el egreso.
observaciones	String	Este campo es para almacenar algunas observaciones que desee emitir el médico.
hora	Date	Hora en la que fue emitido el egreso.

Tabla 1.33. Entidad Egreso

Nombre: Fallecimiento		
Descripción: Contiene toda la información relacionada con el fallecimiento del paciente.		
Atributo	Tipo	Descripción
id	int 4	Identificador de la tabla.
fecha	date	Contiene la información referente a la fecha de fallecimiento del paciente.
hora	time(0)	Contiene la información referente a la hora de fallecimiento del paciente.

Tabla 1.34. Entidad Fallecimiento

Nombre: Hoja_consulta		
Descripción: Contiene toda la información relacionada con la consulta realizada al paciente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.

motivo	text	Contiene las causas por las que fue realizada la consulta.
enfermedad	text	Contiene la información referente al nombre de la enfermedad que posee el paciente en la consulta.

Tabla 1.35. Entidad Hoja_consulta

Nombre: Hoja_emergencia		
Descripción: Contiene toda la información necesaria referente al paciente emergente.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
fecha	date	Contiene la información referente a la fecha de fallecimiento del paciente.
traido_por	string	Contiene la información referente al nombre de la persona que llevó al paciente a la sala de emergencias.
nivel_de_gravedad	string	Contiene la información referente al nivel de gravedad del paciente expresado en colores.
motivo_consulta	string	Contiene la información referente al motivo por el cual fue realizada la consulta.
hora	time(0)	Contiene la información referente a la hora en que llegó el paciente a la sala de emergencia.

Tabla 1.36. Entidad Hoja_emergencia

Nombre: Hoja_frontal		
Descripción: Entidad que almacena los elementos que identifican un paciente en un hospital.		
Atributo	Tipo	Descripción
numeroHc	String	Número de la historia clínica.
foto	String	Dirección donde se encuentra la foto.
nombre	String	Nombre del paciente.
apellido1	String	Primer apellido del paciente.
apellido2	String	Segundo apellido del paciente.
fechaNacimiento	Date	Fecha de nacimiento del paciente.
correoElectronico	String	Correo electrónico del paciente.

Tabla 1.37. Entidad Hoja_frontal

Nombre: Horario_cita		
Descripción: Contiene una descripción detallada del horario establecido para el encuentro del médico con los pacientes.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
hora_inicio	time(0)	Contiene la información referente a la hora de inicio de la cita.
hora_fin	time(0)	Contiene la información referente al fin de la cita.
cant_pacientes	integer	Contiene la información referente a la cantidad de pacientes planificados para la cita.
fecha_inicio	date	Contiene la información referente a la fecha de inicio de la cita.
fecha_fin	date	Contiene la información referente a la fecha de fin de la cita.

Tabla 1.38. Entidad Horario_cita

Nombre: Mes		
Descripción: Almacena los meses.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
mes	integer	Contiene el mes al cual se hace referencia, este es expresado en número.
valor	string	Contiene el nombre del mes al cual se hace referencia.
codigo	string	Este es un identificador utilizado para saber a cual tabla se hace referencia en caso de ocurrir algún tipo de cambio, especialmente de idioma.

Tabla 1.39. Entidad Mes

Nombre: Objeto_cancel_cirugia		
Descripción: Contiene toda la información necesaria para conocer las causas por las que se cancela una cirugía.		
Atributo	Tipo	Descripción
id	integer	Identificador de la tabla.
valor	string	Contiene la información de las causas por las que se cancela una cirugía.

Tabla 1.40. Entidad Objeto_cancel_cirugia

Nombre: Solicitud_intervencion		
Descripción: Entidad que almacena las características de la solicitud de intervención quirúrgica		
Atributo	Tipo	Descripción
fechaSolicitud	Date	Fecha en la cual se realiza la solicitud de intervención quirúrgica.
hora	Date	Hora en la cual se realiza la solicitud de intervención quirúrgica.
electiva	boolean	Indica si la intervención solicitada es electiva.
duracionAproximada	Date	Duración aproximada de la intervención solicitada.
fechaFijada	Date	Fecha para la cual se solicita la intervención quirúrgica.
aprobada	boolean	Indica si la solicitud de intervención quirúrgica es aprobada.
emergencia	boolean	Indica si la intervención solicitada es de emergencia.
causaModificacionQx	String	Causa por la cual se modifica el quirófano. asignado a la solicitud de intervención quirúrgica.
causaRechazoSolicitud	String	Causa por la cual se rechaza la solicitud de intervención quirúrgica.
otrosAnestesiologos	String	Anestesiólogos adicionales involucrados en la futura intervención quirúrgica.
otrasEnfermeras	String	Enfermeras adicionales involucradas en la futura intervención quirúrgica.

Tabla 1.41. Entidad Solicitud_intervencion

Nombre: Cita		
Descripción: Almacenar las citas existentes que han sido planificadas a los médicos.		
Atributo	Tipo	Descripción
id	int	Id de la cita.
fecha	Date	Fecha en que se crea la cita.
hora_entrada	Time	Hora de entrada de la cita.
eliminado	Boolean	Si la cita está eliminada o no.
cid	Integer	Para trabajar con la bitácora.
hora_salida	Time	Hora de salida de la cita.

Tabla 1.42. Entidad Cita

Nombre: Tipo_cita		
Descripción: Almacenar los tipos de citas existentes a los cuales se les hará la planificación.		
Atributo	Tipo	Descripción
id	int	Id del tipo de cita.
valor	String	Nombre del tipo de cita.
codigo	String	Código para estandarizar los nombres de los tipos de citas.

Tabla 1.43. Entidad Tipo_cita

Nombre: Causas_cancel_intervencion		
Descripción: Contiene información relacionada con las causas de cancelación de una intervención quirúrgica.		
Atributo	Tipo	Descripción
id	int	Id de la causa.
valor	String	Descripción de las causas de cancelación de una intervención quirúrgica.

Tabla 1.44. Entidad Causas_cancel_intervencion

Nombre: Orden_admision		
Descripción: Contiene información relacionada con la orden de ingreso del paciente.		
Atributo	Tipo	Descripción
id	integer	Id de la orden de admisión.
observaciones	String	Contiene las observaciones que emite el técnico acerca del paciente.
Fecha_elaboracion	Date	Contiene información relacionada con la fecha de elaboración de la orden de admisión.
Fecha_admision	Date	Contiene información relacionada con la fecha en

		que fue ingresado el paciente.
--	--	--------------------------------

Tabla 1.45. Entidad Orden_admision

Nombre: Estado_evolution_paciente		
Descripción: Contiene información relacionada con la evolución del paciente.		
Atributo	Tipo	Descripción
id	integer	Id de la evolución del paciente.
valor	String	Contiene información relacionada con la descripción del estado de evolución del paciente.

Tabla 1.46. Entidad Estado_evolution_paciente

4.2. Implementación

4.2.1. Diagrama de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes, objetos y procesos (caso particular de un objeto), muestra las relaciones físicas entre los componentes de hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

La arquitectura en tiempo de ejecución del Sistema de Información Hospitalaria alas HIS se modeló con la configuración de 3 elementos de hardware (nodos): una computadora cliente y dos servidores, un servidor de aplicación y un servidor de base de datos, además de disponer de la conexión con un dispositivo, en este caso la impresora.

La ruta de comunicación entre los nodos está regida por los protocolos de comunicación HTTP y TCP/IP, para la asociación entre la computadora cliente y el servidor de aplicación y la conexión de este con el servidor de base de datos respectivamente.

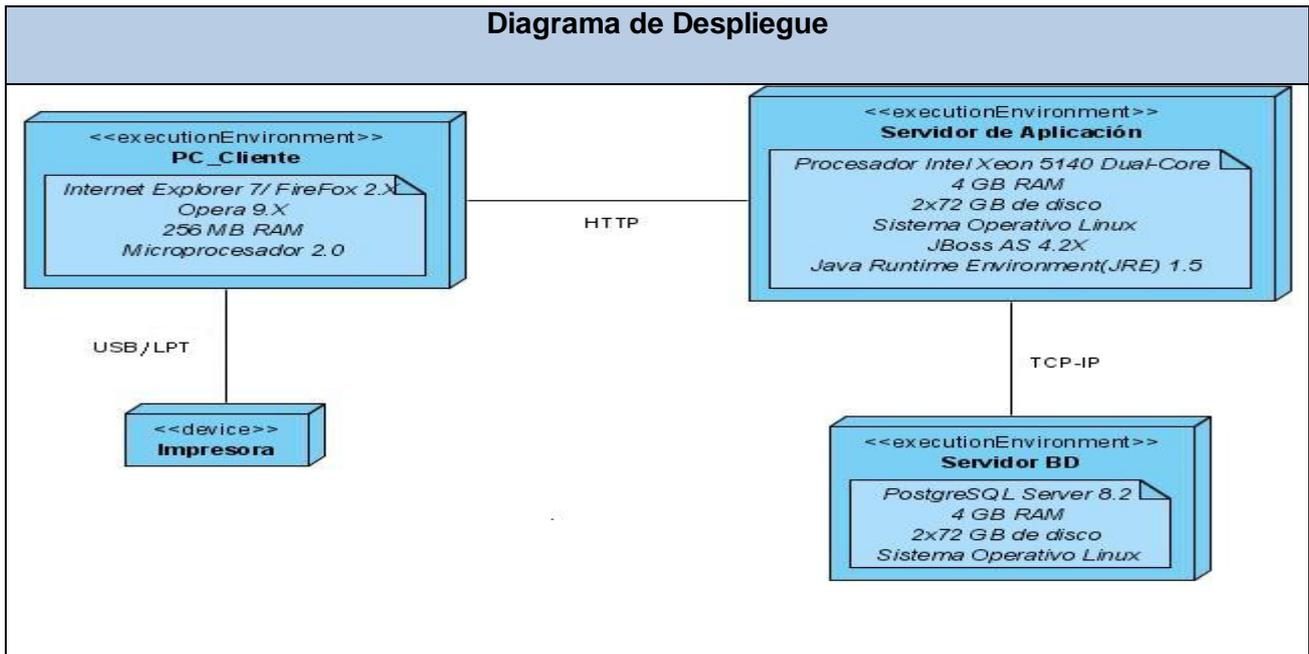


Figura 1.17. Diagrama de Despliegue

4.2.2. Diagrama de Componentes

Un diagrama de componentes ilustra los fragmentos de software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase, usualmente un componentes se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, así eventualmente un componente puede comprender una gran porción de un sistema.

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

Los componentes fundamentales lo conforman instancias de los componentes que representan manifestaciones del código en tiempo de compilación de la interfaz, los datos y el control de las clases que se utilizan. Ver Figura 1.18

Este proceso se realiza en todas las porciones de código donde pueda ocurrir un error, fundamentalmente en aquellas donde se realicen transferencias desde y hacia la base de datos. Además se realiza tratamiento de excepciones en las interfaces de usuario con el objetivo de validar los datos entrados por los mismos.

El sistema cuenta con un archivo denominado `page.xml`, el cual engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, este archivo especifica hacia dónde va a ser dirigido el usuario si ocurriese alguna excepción. Por otra parte se hace uso del componente `Seam FacesMessages`, para capturar cualquier mensaje de error o notificación proveniente de los controladores y posteriormente mostrarlos en las vistas o interfaces de usuario.

4.2.4. Seguridad

Para que cualquier sistema esté definido como seguro, se debe lograr mantener un equilibrio entre las siguientes características:

Integridad: La información sólo puede ser modificada por quien está autorizado y de manera controlada.

Confidencialidad: La información sólo debe ser legible para los autorizados.

Disponibilidad: La información debe estar disponible cuando se necesita.

Irrefutabilidad (No repudio): El uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

En correspondencia con lo que se debe cumplir para lograr desplegar un sistema seguro, a continuación aparecen relacionados un conjunto de acciones llevadas a cabo con el objetivo de que el usuario final tenga el disfrute de contar con un software seguro.

- ❖ Debe ser introducido el nombre y contraseña del usuario que accederá al sistema. (Ver Figura. 1.19). El acceso al mismo está restringido para cada usuario en dependencia de los permisos establecidos de acuerdo al rol que desempeña en la institución hospitalaria, verificándose que los datos introducidos sean válidos.
- ❖ El sistema tiene un estricto control de acceso que posibilita al usuario visualizar únicamente las opciones relacionadas con las actividades que desarrolla.

- ❖ La interfaz relaciona a todos los módulos del sistema alas HIS; desde la misma puede accederse a cada uno de ellos, en dependencia del usuario y los permisos establecidos para el mismo.
- ❖ El administrador del sistema es quien asigna y deniega permisos a roles y usuarios en las funcionalidades de los módulos.
- ❖ Las contraseñas sólo pueden ser cambiadas por el usuario o por el administrador del sistema.
- ❖ Se define un segundo nivel de seguridad en las estaciones de trabajo, para garantizar solo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- ❖ El sistema permitirá llevar una traza de todas las operaciones llevadas cabo por cada usuario mediante un registro de actividades por cada uno, pudiéndose realizar en cualquier momento.
- ❖ Todo el intercambio entre el sistema alas HIS y otros sistemas que soliciten información desde cualquier hospital, se realizará de forma cifrada, para evitar posibilidades de acceso o modificación de la misma.



Figura 1.19. Acceso al sistema

4.2.5. Estrategias de codificación. Estándares y estilos a utilizar.

Se denomina estándar de codificación al conjunto de reglas que se siguen para la escritura del código fuente, de manera que a otros programadores se les facilite entender e identificar las variables y las funciones o métodos. En particular el lenguaje Java, tiene reglas como: Las clases inician en mayúsculas, los atributos y métodos inician con minúsculas, las constantes son todas en mayúsculas. También existe una forma de abrir y cerrar bloques de código y la forma de poner comentarios.

Para el desarrollo del sistema presentado, se decide utilizar un estándar de codificación, para garantizar un estilo de programación homogéneo durante la implementación del sistema, lo cual posibilita que este código pueda ser comprendido fácilmente por otros. A continuación se describen algunos de los elementos que demuestran lo antes enunciado:

- ❖ Idioma: Se utiliza como idioma el español, las palabras no se acentuarán.

La indentación tiene como objetivo lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento, en la misma se definieron los siguientes aspectos:

- ❖ Inicio y fin de bloque: se dejan 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: No se usa el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla. Los inicios `{}` y cierre `}` de ámbitos están alineados debajo de la declaración a la que pertenecen, evitándolos si hay sólo una instrucción. Nunca se coloca `{}` en la línea de un código cualquiera, esto se hace en una línea propia.

También fueron tenidos en cuenta los comentarios, separadores, líneas, espacios en blanco y márgenes para establecer un modo común al comentar el código, de manera que sea comprensible con sólo leerlo una vez. Relacionados con los cuales aparecen los siguientes aspectos:

- ❖ Ubicación de comentarios: los comentarios están:
 - Al final de cada bloque de código.
 - Al inicio de la clase o función se especifica el objetivo de la misma así como los parámetros que usa (se especifican los tipos de datos y objetivo del parámetro) entre otros aspectos.
- ❖ Líneas en blanco: las líneas en blanco se emplean:
 - Antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
- ❖ Espacios en blanco: los espacios en blanco se encuentran entre:
 - Operadores lógicos y aritméticos, para lograr una mayor legibilidad en el código.

Ejemplo: producto = nomproducto

- ❖ Aspectos generales: se tuvieron en cuenta aspectos generales citados a continuación:
 - Sobre el comentario: Se evita comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones es porque está 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
 - Sobre los espacios en blanco: No se usan espacios en blanco después del corchete abierto y antes del cerrado de un arreglo. Después del paréntesis abierto y antes del cerrado, así como antes del punto y coma.

Además, fueron tenidos en cuenta a la hora del trabajo con variables y constantes los siguientes aspectos:

- ❖ Apariencia de variables:
 - El nombre que se le da a las variables comienza con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamellCasing. Ejemplo: nombrePaciente.
- ❖ Apariencia de constantes:
 - La declaración de constantes es con todas sus letras en mayúscula.
- ❖ Aspectos generales:
 - El nombre empleado, permite que con sólo leerlo se conozca el propósito de la misma.

Para el trabajo de las clases y los objetos se definieron las normas presentadas a continuación, con el objetivo de nombrar las clases e instancias de la misma manera para todas las aplicaciones:

- ❖ Apariencia de clases y objetos:
- ❖ Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se emplea la notación PascalCasing. Ejemplo: MiClase ().

❖ Apariencia de atributos:

- El nombre que se le da a los atributos de las clases comienza con la primera letra en minúscula, en caso de ser un nombre compuesto se emplea la notación CamellCasing.

❖ Apariencia de las funciones. Para nombrar las funciones se define:

- La Primera letra en mayúscula.
- Verbos que denoten la acción que hace la función. Se emplea la notación PascalCasing. Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.

❖ Declaración de parámetros en funciones:

- Los parámetros se declaran agrupados por tipos: primero los string y después los numéricos. Además son agrupados según valores por defecto para cada tipo de dato utilizado en el lenguaje.

❖ Aspectos generales:

- Para las clases, los objetos, los atributos y las funciones, el nombre empleado permite que con sólo leerlo se conozca el propósito de los mismos.

Se define como convención interna en el proyecto para los nombres de las entidades que se usan en varios módulos: [nombre de la entidad]_[nombre del módulo].

En este capítulo fueron relacionados los principales elementos de la implementación del sistema presentado. Se definió la propuesta para el despliegue y la estructura del diagrama de componentes del mismo. Además fue descrita la concepción del tratamiento de errores, la seguridad y estrategias de codificación, así como de los estándares y estilos utilizados.

Conclusiones

Con el desarrollo de la presente investigación, fueron analizados los procesos de negocio asociados al área de Estadísticas de las Instituciones hospitalarias, lo que evidenció la necesidad de desarrollar un sistema informático. El desarrollo de las tareas de la investigación posibilitó dar cumplimiento al objetivo general y arribar a las siguientes conclusiones:

- ❖ Fueron definidos los procesos de negocio asociados al área de estadísticas de las instituciones hospitalarias.
- ❖ Se demostró la necesidad existente de desarrollar un sistema de información hospitalaria que incluya la generación de reportes estadísticos.
- ❖ La asimilación de la arquitectura posibilitó una representación lógica y óptima de los componentes de la solución.
- ❖ Se definió la plantilla que regulará la información a mostrar en cada reporte.
- ❖ Se obtuvo el modelo lógico del sistema en términos de diseño, en correspondencia con la arquitectura, facilitando la obtención del módulo Estadísticas del Sistema de Información Hospitalaria alas HIS que garantiza la generación de los reportes estadísticos propuestos.

Recomendaciones

Este trabajo fue concebido para darle solución a la problemática existente en el área de Estadísticas de las instituciones hospitalarias. Debido a la importancia que tiene el sistema obtenido y en vistas de mejorar futuras versiones del mismo, se recomienda:

- Rediseñar la información contenida en los módulos del Sistema de Información Hospitalaria alas HIS que permita completar los datos a mostrar en los reportes estadísticos.
- Implementar la generación de gráficos dinámicos que representen el comportamiento de los resultados estadísticos.

Referencias bibliográficas

1. Santoja, Joaquín Herrero. *La experiencia de Kewan - Cosmosalud*. España : s.n., 2008.
2. García Linares, Reche Martínez, D. Richarte Reina. *Un nuevo concepto en Sistemas de Información Hospitalarios*. Madrid : Comunicaciones S.A., 2003.
3. SIGHO. Sistema de Información para la Gerencia Hospitalaria. [En línea] <http://www.sigho.gob.mx/quees.htm>.
4. OpenHIS. Openhis. [En línea] <http://www.openhis.com.ar/release/>.
5. Health Care International Services . [En línea] <http://www.hcisonline.com/html/Software12.htm>.
6. Softel. *Propuesta para la Gestión de los Donantes de Sangre y Distribución de Sangre y Componentes Sanguíneos*. .
7. Introducción a los sistemas distribuidos. [En línea] 2007. <http://www.augcyl.org/?q=glob-intro-sistemas-distribuidos>.
8. José Guillermo. Arquitectura Cliente-Servidor. [En línea] 2005. <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
9. Patrones de Diseño en aplicaciones Web con Java J2EE. [En línea] 2007. http://java.ciberaula.com/articulo/disenio_patrones_j2ee.
10. Visual Paradigm for UML. [En línea] 2008. <http://www.versionzero.com/noticia/210/visual-paradigm-for-uml>.
11. Javi Sanromán. [En línea] <http://www.jsanroman.net/2007/11/20/%C2%BFque-es-jasper-reports-2/>.
12. Ing. Delia Medina. manual de ireport y jasper. [En línea] http://es.geocities.com/mdelia_99/ireport/manual_de_ireport_y_jasper.html.
13. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal John. *Pattern-Oriented Software Architecture*. 1996.
14. Dr. Juntao Yuan. On the road to simplicity. [En línea] 2008. <http://www.javaworld.com/javaworld/jw-02-2005/jw-0221-jboss4.html>.
15. PostgreSQL About. [En línea] 2008. <http://www.postgresql.org/about>.
16. Bauer, Christian and King, Gavin. *Java Persistence with Hibernate*. 2007.
17. Soluciones y propuestas Rational. [En línea] <http://www.rational.com.ar/herramientas/rup.html>.
18. JBOSS. [En línea] 2009. <http://www.jboss.com/products/jbpm/>.
19. UML: Casos de Uso. Desarrollo de software orientado a objetos. [En línea] 2003. <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>.
20. Plan estratégico a plazo medio. [En línea] 2008.

http://apps.who.int/gb/ebwha/pdf_files/MTSP2009/MTSP3-sp.pdf.

21. Chaves, Michael Arias. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 2006. http://www.intersedes.ucr.ac.cr/10-art_11.html.

22. [En línea] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.

23. web estilo. [En línea] <http://www.webestilo.com/javascript/>.

24. ALBET. *Documento de Arquitectura de Software alas HIS*. 2008.

25. Dr Marqués, María Mercedes. Modelo de Datos . [En línea]

<http://www3.uji.es/~mmarques/f47/apun/node32.html>.

Bibliografía

1. ALBET. *Documento de Arquitectura de Software alas HIS*. 2008.
2. Apps.Plan estratégico a plazo medio. [En línea] 2008. http://apps.who.int/gb/ebwha/pdf_files/MTSP2009/MTSP3-sp.pdf .
3. Augcyl.Introducción a los sistemas distribuidos. [En línea] 2007.
4. Bauer, Christian and King, Gavin. *Java Persistence with Hibernate*. 2007.
5. BOOCH, GRADY, RUMBAUGH, JAMES and JACOBSON, IVAR. *El lenguaje Unificado de Modelado*. 2000.
6. Blog.Jboss,<http://blog.wixel.es/2008/07/desarrollos-con-jsf-y-jboss-seam-i-introduccion/>.
7. Buschmann Frank, Meunier Regine, Rohnert Hans, Sommerlad Peter, Stal John Michael. *Pattern-Oriented Software Architecture*. 1996.
8. Dr. Juntao, Yuan. On the road to simplicity. [En línea] 2008.
9. Dr Marqués, María Mercedes. Modelo de Datos . [En línea].
10. García Linares, Reche Martínez, D. Richarte Reina. *Un nuevo concepto en Sistemas de Información Hospitalarios*. Madrid : Comunicaciones S.A., 2003.
11. Genbeta.Eclipse Europa, nueva versión del entorno de programación. [En línea] 2007.
12. Hcisonline. Health Care International Services . [En línea] <http://www.hcisonline.com/html/Software12.htm>.
13. Ingenierosoftware. UML: Casos de Uso. Desarrollo de software orientado a objetos. [En línea] 2003. <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>.
14. Javi Sanromán. [En línea] <http://www.jsanroman.net/2007/11/20/%C2%BFque-es-jasper-reports-2/>.
15. Ing. Delia Medina. manual de ireport y jasper. [En línea] http://es.geocities.com/mdelia_99/ireport/manual_de_ireport_y_jasper.html.
16. JBOSS. JBPM, 2009. .
17. La web del programador. [En línea]
18. Maestros del Web. [En línea] 2009. <http://www.maestrosdelweb.com/editorial/ajax/>.
19. Media, O'Reilly. XML. [En línea] .
20. OpenHIS. Openhis. [En línea] <http://www.openhis.com.ar/release/>..
21. Patrones de Diseño en aplicaciones Web con Java J2EE. [En línea] 2007. http://java.ciberaula.com/articulo/disenio_patrones_j2ee.
22. PostgreSQL. PostgreSQL About. [En línea] 2008. <http://www.postgresql.org/about>.

23. PostgreSQL. PostgreSQL Awards. [En línea] 2008. <http://www.postgresql.org/about/awards>.
24. Proactiva <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
25. Salaboy.Jboss <http://salaboy.wordpress.com/2008/03/02/algunos-conceptos-de-jboss-jbpm/>.
26. Santoja, Joaquín Herrero. *La experiencia de Kewan - Cosmosalud*. España : s.n., 2008.
27. SIGHO. Sistema de Información para la Gerencia Hospitalaria. [En línea].
28. Softel. *Propuesta para la Gestión de los Donantes de Sangre y Distribución de Sangre y Componentes Sanguíneos*.
29. Soluciones y propuestas Rational. [En línea] <http://www.rational.com.ar/herramientas/rup.html>.
30. Sun Microsystems. Java Server Faces [En línea] 2009.
31. UCI. El Proceso Unificado de Desarrollo (RUP). [En línea] 2008. http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008.
32. Versioncero. Visual Paradigm for UML. [En línea] 2008.
33. Webtaller. Manual de Java. [En línea].
34. web estilo. [En línea] <http://www.webestilo.com/javascript/>.

Anexos

❖ Anexo 1

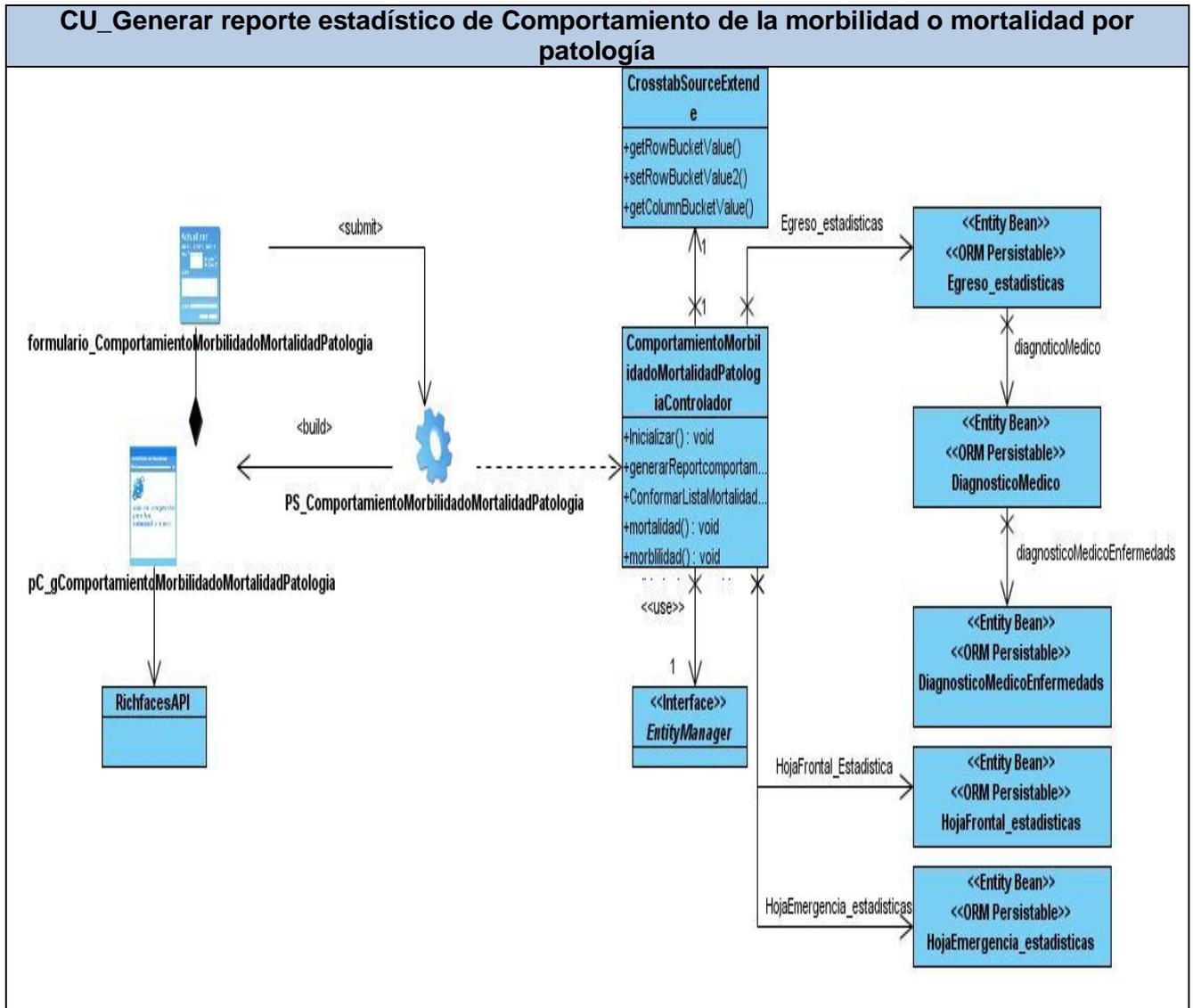


Figura 1.20. Diagrama de clases del diseño. CU_Generar reporte estadístico de la morbilidad o mortalidad por patología.

❖ Anexo 2

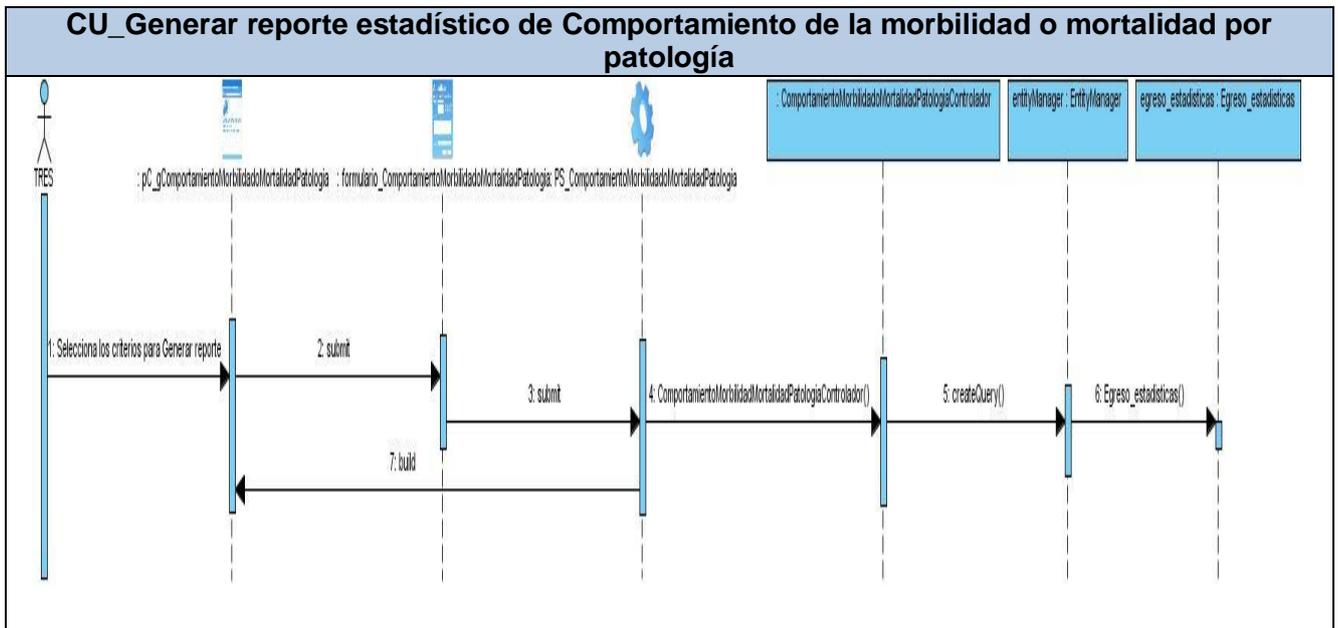


Figura 1.21. Diagrama de secuencia. CU_Generar reporte estadístico de la morbilidad o mortalidad por patología.

❖ Anexo 3

Reporte estadístico de Servicios de apoyo



Reporte estadístico de servicios de apoyo

Servicios de apoyo: Servicios de apoyo: Laboratorio, Banco de sangre; Meses: Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre Anatomía patológica, Laboratorio, Banco de sangre; Meses: Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre

Sábado, 13 de junio de 2009

		May	Jun	Jul	Nov	Total
Anatomía Patológica	Autopsias	0	1	0	0	1
	Citologías	1	1	0	0	2
Banco Sangre	Trasfusiones	0	0	2	0	2
	Tratamientos terapéuticos	0	0	2	0	2
Laboratorio	Total analisis	0	1	0	1	2
	consulta externa	0	2	0	0	2
	emergencia	0	2	0	0	2
	hospitalización	0	0	0	2	2

Figura 1.22. Reportes estadísticos de Servicios de apoyo

❖ Anexo 4

Reporte Comportamiento de la Natalidad

Generar comportamiento de la natalidad Q Buscar...

Seleccione los parámetros deseados «

Rango de años:

Desde: Hasta:

Generar comportamiento de la natalidad

Años	Tasas	Índice	Variación promedio anual: 33.33333333333334
2008	0.011998656150511143	100.0	Cambio porcentual: 33.33333333333334
2009	0.007999104100340762	66.66666666666666	

Figura 1.23. Reporte Comportamiento de la Natalidad

Glosario de términos

ALAS-RIS: Sistema externo con el que se comunica el módulo para la gestión de la información de los estudios radiológicos e imagenológicos.

API: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

BPMN: Business Process Modeling Notation que significa Notación para el Modelado de Procesos de Negocio.

Browser: Término del idioma inglés, utilizado para referirse a navegador.

Business Process Management: BPM son sus siglas en inglés, lo que significa Gestión de Procesos del Negocio.

Componente: término utilizado para representar archivos de código fuente, código binario, ejecutables o DLL, conectados mediante relaciones de dependencia.

Dependencias: relación entre dos elementos que indica que los cambios efectuados en el destino pueden ocasionar cambios en el de origen.

Despliegue: se refiere a la instalación y aplicación del Sistema de Información Hospitalaria alas HIS.

Diagrama: Presentación gráfica de un conjunto de elementos y sus relaciones.

Diagrama de componentes: diagramas de implementación que muestran la estructura del código.

DLL: bibliotecas de vínculos dinámicos.

Framework: Librería de código (posiblemente ya compilado) que sirve como base para el desarrollo de un tipo determinado de aplicaciones.

Gang of Four: GoF, que en español es la pandilla de los cuatro.

GRASP: Siglas en inglés correspondientes a General Responsibility Assignment Software Patterns, lo cual significa Patrones Generales de Asignación de Responsabilidades.

HIS: Sistemas de Información Hospitalaria

Hojas de Estilo en Cascada: Del inglés Cascading Style Sheets (CSS).

HTML: Del inglés XHTML (eXtensible Hypertext Markup Language).

Informática: Disciplina que estudia el tratamiento automático de la información con el uso de dispositivos electrónicos y sistemas computacionales.

Instituciones hospitalarias: Se refiere a hospitales o entidades de salud donde los pacientes se atiendan integralmente.

Interfaz: medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

Java Persistence API: Del inglés JPA, es la API de persistencia desarrollada para la plataforma Java EE.

JSP: Siglas en inglés de Java Server Pages.

Notación CamellCasing: Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas e inician cada palabra con letra mayúscula excepto la primera palabra, que debe iniciar con minúscula. Ejemplo: notacionCamelCasing.

Notación PascalCasing: Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, cada una inicia con letra mayúscula. Ejemplo: NotacionPascalCasing.

Open Source: Significa código abierto, indica que el código fuente de una aplicación es de libre difusión.

Lenguaje de Marcado de Hipertexto Extensible: Del inglés XHTML (eXtensible Hypertext Markup Language).

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

POJOs: Siglas en inglés correspondientes a Plain Old Java Object.

Plugins: Complementos que dan mayor usabilidad al software.

SIPOL: Sistema externo con el que se comunica el módulo para la gestión de la información de las autopsias forenses.

Servidor: Son ordenadores que actúan como "almacenes" de información. Esta información es solicitada por los ordenadores cliente y el servidor responde a tales peticiones con la devolución de los datos solicitados.

TRES: Técnico de registros y estadísticas de la salud.

Underscore: Carácter ASCII representado como "_".

XML: Siglas en inglés de Extensible Markup Language que significa Lenguaje de marcas.