

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Desarrollo de los reportes del Módulo Admisión del
Sistema de Información Hospitalaria alas HIS**

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

Autores: Lixandra Cordova Eireos
Leandro Benítez Ortiz

Tutores: Ing. Kenia Fernández Parra
Ing. Yosmel Martínez Díaz

La Habana, junio de 2011

“Año 53 de la Revolución”

Datos de contacto

Ing. Kenia Fernández Parra

Graduada en el año 2007 de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Posee categoría docente de Instructor. Ha impartido las asignaturas Matemática Discreta, Gestión de Software, Sistemas de Bases de Datos, Ética Informática y Metodología de la Investigación Científica. Se desempeñó por un período de 6 meses como líder del área temática productiva Gestión Hospitalaria (GEHOS). Actualmente se desempeña como analista en el proyecto Sistema de Información Hospitalaria alas HIS perteneciente al Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM).

Correo electrónico: kfernandezp@uci.cu

Ing. Yosmel Martínez Díaz

Instructor recién graduado en el año 2009 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor vinculado a la Facultad 7 y miembro del Departamento de Sistemas de Gestión Hospitalaria. Ha impartido la asignatura de Práctica Profesional y cursos optativos definidos por la facultad.

Correo electrónico: ymdiaz@uci.cu

Agradecimientos

De Leandro:

Quiero agradecer especialmente a mis padres por todo el amor y el cariño que me han dado durante toda mi vida, por la educación recibida, por su atención y por ser ellos la motivación que tengo hoy de lograr esta meta.

A mis hermanitas.

A toda mi familia que siempre me ha apoyado y siempre han creído en mí.

A mis amigos que me han soportado en los buenos y malos momentos.

Al profesor Yosmel Martínez Díaz por su preocupación y ayuda en este trabajo, y a aquellos profesores que de una forma u otra también han tenido que ver con este resultado.

Y por último y no menos especial a mi compañera de tesis, de la cual estoy muy orgulloso de tenerla como compañera en este trabajo.

De Lixandra:

Agradecer a mi familia por ser la fuerza que me ayuda a superar cada etapa de mi vida, por estar siempre pendiente de mí, por guiarme, apoyarme y ofrecerme lo mejor en cada momento.

A mi abuela por ser mi razón de ser, gracias a ella he tenido fuerzas para seguir adelante siempre.

A mi mamá por su ayuda y apoyo incondicional, sin ella nunca hubiera podido alcanzar esta meta.

A mi padrastro Luisito por ser muy bueno, preocupado, y ayudarme en todo lo que he necesitado, es más que un padre para mí.

A mi tía Maricel por darme buenos consejos, por ser tan preocupada y por ayudarme tanto.

A mi prima Lisbeth por estar conmigo en todo momento, brindarme su ayuda y ser tan buena y preocupada.

A mi novio Rodney por guiarme, apoyarme siempre y enseñarme a ser mejor en todos los aspectos de mi vida, gracias a él pude traspasar todas las barreras que aparecieron en mi camino.

A mis tutores y a todos los profesores del proyecto porque siempre han estado dispuestos a contribuir y a brindar lo mejor en cada momento para que saliera adelante.

A todas mis amistades y compañeros de estudio porque cada uno de ellos ha puesto su granito de arena para ayudarme.

Dedicatoria

De Leandro:

Este trabajo va dedicado especialmente a mis padres por ser ellos la razón de haber logrado hoy este resultado.

A mis hermanitas.

A toda mi familia porque también se sienten identificados con este resultado, y porque tengo mucha suerte de tenerlos.

De Lixandra:

Dedico este trabajo a mi abuela porque es la persona por la que cada día he tenido fuerzas para superarme en la vida y llegar a ser una profesional, sé que es un sueño para ella.

A mi mamá porque gracias a ella he llegado a cumplir todo lo que me he propuesto.

A mis hermanitos para los que espero ser un ejemplo.

A Rodney por ser mi principal ayuda en estos cinco años de estudio.

A toda mi familia que me ha apoyado siempre y ha contribuido a mi formación como persona y como profesional, este logro es para ella.

Resumen

El Sistema de Información Hospitalaria alas HIS posee un Módulo Admisión, en el cual se emiten una serie de reportes estadísticos que carecen de flexibilidad ya que, su estructura es rígida en cuanto a columnas a mostrar en las tablas generadas y en cuanto a criterios de búsqueda. Además no es posible imprimirlos para disponer de ellos fuera del sistema. Muchos de estos reportes tienen en común varios de los datos que manejan y carecen de gráficos que visualicen la información. Es por esto que se propone el desarrollo de los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS, que optimice la gestión de información en esta área de las instituciones hospitalarias.

El desarrollo del sistema está guiado por el Proceso Unificado de Desarrollo, sobre una arquitectura en capas, es multiplataforma y se basa en tecnologías libres. Se utiliza Java como lenguaje de programación y el patrón de arquitectura Modelo Vista Controlador. Visual Paradigm como herramienta de modelado, PostgreSQL como servidor de base de datos y pgAdmin para su administración. Eclipse como entorno de desarrollo, iReport para el diseño visual de los reportes y JasperReports para la generación de los mismos.

Con la optimización de los reportes se espera proporcionar al usuario una forma más general y fácil de manejar la información, al permitirle seleccionar de los datos adicionales definidos en los mismos, cuáles desea incluir en el reporte a generar, además de facilitar mayor entendimiento de los datos a través de la visualización de gráficos.

Palabras claves

Módulo, admisión, reporte, gráfico.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	6
1.1 Conceptos fundamentales relacionados con el dominio del problema	6
1.2 Situación problemática	8
1.3 Sistemas automatizados existentes vinculados al campo de acción	8
1.4 Tendencias y tecnologías actuales a considerar	12
1.5 Herramientas	21
1.6 Metodologías de desarrollo de software	23
Capítulo 2. Descripción de la Arquitectura	25
2.1 Requisitos no funcionales	25
2.2 Descripción de la arquitectura. Fundamentación.	28
2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados. Estrategias de integración.....	29
2.4 Seguridad	29
2.5 Vista de Despliegue.....	30
2.6 Estrategias de codificación. Estándares y estilos a utilizar.	31
Capítulo 3. Descripción y análisis de la solución propuesta	34
3.1 Valoración crítica del diseño propuesto por el analista	34
3.2 Descripción de las nuevas clases u operaciones necesarias.....	43
3.3 Modelo de datos	50
3.4 Valoración de las Técnicas de validación	52
3.5 Vista de Implementación	58
Capítulo 4. Modelo de prueba	60

4.1 Pruebas de caja negra	60
4.2 Descripción de los casos de prueba	62
Conclusiones	81
Recomendaciones	82
Referencias bibliográficas	83
Bibliografía.....	86
Glosario de términos.....	88

Índice de figuras

Figura 2.1. Diagrama de despliegue	31
Figura 3.1. Diagrama de paquetes	36
Figura 3.2. DCD_Generar reporte de salidas por permisos.....	37
Figura 3.3. DCD_Generar reporte de pacientes referidos	38
Figura 3.4. DCD_Generar reporte de pacientes fuera de servicio	38
Figura 3.5. DCD_Generar reporte de orden de admisión y transferencia sin efectuar.....	39
Figura 3.6. DCD_Generar reporte de auditoría	39
Figura 3.7. DCD_Generar reporte de movimientos hospitalarios por servicio	40
Figura 3.8. DS_Generar reporte de salidas por permisos	40
Figura 3.9. DS_Generar reporte de pacientes referidos.....	41
Figura 3.10. DS_Generar reporte de pacientes fuera de servicio.....	41
Figura 3.11. DS_Generar reporte de orden de admisión y transferencia sin efectuar	42
Figura 3.12. DS_Generar reporte de auditoría	42
Figura 3.13. DS_Generar reporte de movimientos hospitalarios por servicio	43
Figura 3.14. Modelo de datos.....	51
Figura 3.15. Diagrama de componentes	59

Índice de tablas

Tabla 3.1. SalidasPorPermisosControlador.....	44
Tabla 3.2. PacientesReferidosControlador.....	45
Tabla 3.3. PacientesFueraServicioControlador	46
Tabla 3.4. OrdenAdmisionTransfSinEfectuarControlador.....	47
Tabla 3.5. AuditoriaControlador	49
Tabla 3.6. MovimientosHospitalariosPorServicioControlador	50
Tabla 3.7. hc_local.admisión.....	53
Tabla 3.8. hc_local.transferencia	53
Tabla 3.9. hc_local.egreso	54
Tabla 3.10. hc_local.hoja_frontal	56
Tabla 3.11. publico.orden_admision.....	56
Tabla 3.12. publico.orden_transferencia	57
Tabla 3.13. nomencladores.tipo_admision	57
Tabla 3.14. nomencladores.tipo_egreso	57
Tabla 3.15. comun.departamento	57
Tabla 3.16. comun.servicio	58
Tabla 3.17. publico.permiso_salida.....	58
Tabla 3.18. publico.entrada_pac_permiso_salida	58
Tabla 4.1. CP_Generar reporte de salidas por permisos.....	62
Tabla 4.1.1. SC_Generar reporte de salidas por permisos.....	64
Tabla 4.2. CP_Generar reporte de pacientes referidos	65
Tabla 4.2.1. SC_Generar reporte de pacientes referidos	66
Tabla 4.3. CP_Generar reporte de pacientes fuera de servicio	67

Tabla 4.3.1. SC_Generar reporte de pacientes fuera de servicio	69
Tabla 4.4. CP_Generar reporte de orden de admisión y transferencia sin efectuar.....	70
Tabla 4.4.1. SC_Generar reporte de orden de admisión y transferencia sin efectuar.....	73
Tabla 4.5. CP_Generar reporte de auditoría	74
Tabla 4.5.1. SC_Generar reporte de auditoría	77
Tabla 4.6. CP_Generar reporte de movimientos hospitalarios por servicio	78
Tabla 4.6.1. SC_Generar reporte de movimientos hospitalarios por servicio	80

Introducción

En la nueva era de "la Sociedad del Conocimiento", la información y las comunicaciones, son factores claves en los procesos de desarrollo. Las Tecnologías de la Información y las Comunicaciones (TIC), cumplen un papel determinante en esta nueva sociedad, por lo que se hace necesario entender y aprovechar los beneficios potenciales y ventajas competitivas, de estos elementos tecnológicos, para incrementar la productividad, el bienestar y la calidad de vida de los ciudadanos.

Las ciencias de la salud, y la medicina en particular, son uno de los campos del saber más evolucionados y beneficiados por el uso de las modernas tecnologías de la información; el desarrollo tecnológico alcanzado en estas ramas, ha permitido conocer infinidad de procesos que explican el porqué de muchas enfermedades y eventos que ocurren en el organismo humano. Por ello ha crecido de forma exponencial el número de usuarios y de instituciones que se han incorporado a la búsqueda de diferentes medios que permitan una mejor calidad de vida.

Aunque la informática y las comunicaciones en la medicina se han desarrollado considerablemente, en ocasiones solo se emplean sus ventajas en la adquisición y manejo de equipos médicos de alta complejidad, apartando a un lado los beneficios que en el campo de la gestión y administración de recursos estas proporcionan, lo que trae como consecuencia la existencia de problemas administrativos en las instituciones hospitalarias.

La demanda de una información precisa y oportuna está creciendo cada vez más en los hospitales, debido al ambiente tan complejo en el que estos funcionan. Además, estas instituciones requieren acumular y organizar la información para que sea accesible inmediatamente, así como reducir el flujo de papel y la redundancia de información para mejorar la productividad. Igualmente se necesita perfeccionar, innovar y apoyar con el uso de las tecnologías, los procesos de registro, seguimiento y tratamiento del paciente para hacer más eficiente y eficaz las actividades habituales del hospital, centro de salud o clínica. (1)

Para resolver estos problemas se crearon los Sistemas de Información Hospitalaria (más conocidos por sus siglas en inglés como HIS), los cuales tienen como propósito satisfacer las necesidades de generación de información, para almacenar y procesar datos médico-administrativos de cualquier

institución hospitalaria. Permitiendo la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes de trámites administrativos que enfrentan los pacientes. Estos sistemas tienen un fuerte impacto en las instituciones de salud, ya que, buscan elevar la calidad de la atención del paciente, de los servicios brindados y aplicar la información obtenida a las áreas de la investigación, la clínica, la docencia y la administración, además de abatir costos y elevar la productividad. (2)

En un principio, cuando los Sistemas de Información se introdujeron en los hospitales, respondían a las necesidades existentes de mecanización de la información, con el fin de conocer la actividad realizada, y al mismo tiempo, incorporar un cambio de hábitos y relaciones entre los usuarios. Sin embargo, en los últimos años, los sistemas de Información tienen un nuevo empuje, debido a la cultura alcanzada en el manejo de tecnología, al desarrollo de la informática y a la aparición de nuevas herramientas, sobre todo de software. Estos sistemas están orientados hacia los profesionales asistenciales (médicos, enfermeras, asistentes de registros médicos), en contraposición a lo que hasta ahora se venía haciendo.

Los estudios realizados acerca de los Sistemas de Información Hospitalaria han determinado que la mayor parte de estos poseen características comunes, entre las que se encuentran:

- Disponen de enormes volúmenes de datos.
- La información o los datos que manejan no son iguales en todos los sistemas.
- La información obtenida no es comparable con la de otros Sistemas de Información Hospitalaria.
- Los flujos de datos son muy complejos. (3)

Un Sistema de Información Hospitalaria puede estar compuesto por módulos o subsistemas que representan principalmente, cada una de las áreas que se encuentran dentro de las instituciones hospitalarias.

Uno de estos módulos es Admisión, el cual se encarga de gestionar la información de pacientes, desde su ingreso hasta su salida del hospital. El objetivo principal de este módulo es, registrar la admisión de pacientes en alguna de las áreas de la Unidad de Salud; crear su historia clínica o

consultarla en caso de que exista; registrar los traslados internos, externos y el alta del paciente; todo esto a través de la persona encargada en el área Admisión del hospital. Además este módulo, es el responsable de mantener actualizada la información referente a la población de las salas, y la coordinación entre las camas de los pacientes ingresados en el centro médico. (4)

Entre los Sistemas de Información Hospitalaria existentes en el mundo se encuentra alas HIS, producto cubano que cuenta con un Módulo Admisión, en el cual se emiten una serie de reportes o informes relacionados con la estancia del paciente en la institución sanitaria, permitiendo obtener información detallada de cada uno de los movimientos hospitalarios a los que fueron sometidos los pacientes que están ingresados o que en algún momento lo estuvieron, además de proporcionar sus datos personales y clínicos, siendo todo esto de gran importancia en el apoyo a la toma de decisiones y la evaluación de los procesos.

Estos reportes carecen de flexibilidad desde el punto de vista de que su estructura es rígida en cuanto a columnas o campos a mostrar en las tablas generadas y en cuanto a criterios de búsqueda y filtros, además no es posible imprimirlos ni disponer de ellos fuera del sistema, lo que trae como consecuencia que el usuario no pueda realizar otras actividades con estos reportes sin depender de la aplicación, como por ejemplo, enviar por correo, o brindar una copia.

La gran cantidad de reportes existentes actualmente puede dificultar el trabajo de los usuarios, al no tener una manera más general, reducida y fácil de manejar la información. De igual forma muchos de los reportes existentes tienen en común varios de los datos que manejan, siendo repetitivos en algunas ocasiones, careciendo además de gráficos que visualicen el comportamiento de la información. Todo esto hace que la interacción de los usuarios con estas funcionalidades sea engorrosa, imposibilitando la obtención de información estructurada correctamente y complejizando en ocasiones la toma de decisiones.

Basado en lo antes expuesto se tiene como **problema a resolver**: ¿Cómo optimizar la generación de reportes estadísticos relacionados con los procesos del Módulo Admisión del Sistema de Información Hospitalaria alas HIS?

En correspondencia con el problema, el **objeto de estudio** lo constituye, el proceso de generación de reportes estadísticos en el Sistema de Información Hospitalaria alas HIS. El **campo de acción** está centrado en el proceso de generación de reportes estadísticos en el Módulo Admisión del Sistema de Información Hospitalaria alas HIS.

Para resolver el problema identificado se propone el siguiente **objetivo general**: Desarrollar los reportes estadísticos del Módulo Admisión del Sistema de Información Hospitalaria alas HIS, que optimice la gestión de información en esta área de las instituciones hospitalarias.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **Tareas a desarrollar**:

1. Evaluar las tendencias actuales en los Sistemas de Información Hospitalaria, que cuentan con módulos relacionados con los procesos que se llevan a cabo en el área de Admisión de las instituciones hospitalarias y que incluyan la generación de reportes estadísticos y gráficos.
2. Analizar el proceso de generación de reportes asociado al Módulo Admisión del Sistema de Información Hospitalaria alas HIS.
3. Valorar la arquitectura definida por el Departamento de Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
4. Aplicar las pautas de diseño y codificación, definidas por el Departamento de Sistemas de Gestión Hospitalaria.
5. Obtener mediante el Proceso Unificado de Desarrollo, los artefactos generados en los flujos de trabajo de, "Diseño", "Implementación" y "Pruebas".
6. Implementar los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS.
7. Obtener el acta de liberación de los artefactos generados, emitida por el grupo de calidad interno del Departamento de Sistemas de Gestión Hospitalaria.

El desarrollo de los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS, proporciona los siguientes beneficios a partir de su puesta en práctica en las instituciones hospitalarias:

1. Proporcionar al usuario una forma general, reducida y fácil de manejar la información, posibilitando la obtención de reportes estructurados correctamente y la toma de decisiones adecuadas.
2. Permitir al usuario seleccionar de los datos adicionales definidos en los reportes, cuáles desea incluir en el reporte a generar.
3. Facilitar mayor entendimiento de los datos que se manejan a través de la visualización de gráficos que esquematicen su comportamiento.
4. Brindar la posibilidad al usuario de disponer de reportes en diferentes formatos de archivo, lo que posibilita que el mismo no tenga que depender del sistema en otro momento para consultarlos.

El presente documento está estructurado en cuatro capítulos:

Capítulo 1 “Fundamentación Teórica”: brinda un estado del arte de los Sistemas de Información Hospitalaria existentes en el mundo que cuentan con módulos relacionados con los procesos que se llevan a cabo en el área de Admisión de los hospitales y que generan reportes estadísticos. Se estudian las tecnologías, metodologías y herramientas a utilizar en el proceso de desarrollo.

Capítulo 2 “Descripción de la arquitectura”: se definen los requisitos no funcionales, se describe la arquitectura del sistema, se realiza un análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados. Además se abordan aspectos importantes como la seguridad del sistema, la vista de despliegue y las estrategias de codificación.

Capítulo 3 “Descripción y análisis de la solución propuesta”: se realiza una valoración del diseño, se muestran los diagramas de clases del diseño y los de secuencia, se describen las clases u operaciones necesarias para el desarrollo, se muestra el modelo de datos y la vista de implementación.

Capítulo 4 “Modelo de Prueba”: se define como método de prueba a utilizar las de caja negra y se diseñan los casos de pruebas correspondientes a los Casos de Usos del Sistema seleccionados.

Capítulo 1. Fundamentación Teórica

El presente capítulo tiene como principal objetivo profundizar en la situación actual del proceso generar reporte del Módulo Admisión del Sistema de Información Hospitalaria alas HIS. Se explican cada uno de los conceptos básicos relacionados con el dominio del problema a resolver y se analizan los sistemas de información existentes en el mundo que están relacionados con el problema actual, para diseñar una solución eficiente. También se realiza un estudio de las tecnologías, herramientas y metodologías que se utilizan en el proceso de desarrollo y se justifica el uso de cada una de ellas.

1.1 Conceptos fundamentales relacionados con el dominio del problema

1.1.1 Reporte

Documento, generado por el sistema, que presenta de manera estructurada y resumida, datos relevantes guardados o generados por la misma aplicación.

1.1.2 Inscripción

La inscripción del paciente es la actividad delantera en todo hospital. Cada paciente tiene que registrarse antes de conseguir cualquier consulta, tratamiento e investigación. La inscripción involucra, información general y demográfica sobre el paciente al que se le asigna un único número de inscripción central y se le entrega una tarjeta de identificación.

1.1.3 Admisión

Existe un proceso estrechamente relacionado con la hospitalización de los pacientes al que generalmente se le llama Admisión. Esta sección es la encargada de almacenar toda la información referente a los pacientes durante su estancia en el hospital, además de manejar y mantener actualizados todos estos datos.

1.1.4 Historia clínica

Sumergida dentro de la inscripción y admisión hospitalarias se encuentra la historia clínica (HC), que es un documento que surge en el contacto entre el equipo de salud y los pacientes. Este es único y válido desde el punto de vista clínico y legal. Es creada en el proceso de inscripción en un hospital y luego es manejada por los médicos para almacenar el comportamiento evolutivo de un paciente durante su estancia en el centro. No se limita a ser una narración o exposición de hechos

simplemente, sino que incluye juicios, documentos y procedimientos; es un documento que se va haciendo con el tiempo, documentando fundamentalmente la relación médico-paciente. (5)

1.1.5 Ingreso

Un ingreso es la admisión de un paciente al hospital, que puede ser por urgencias cuando se da el caso de que el paciente llega por cuerpo de guardia y la dolencia que presenta requiere de un ingreso urgente, o de manera electiva, esto significa que viene por consulta externa y que el ingreso ya estaba planificado. El ingreso se practica en una sala perteneciente a un servicio determinado, al paciente se le asigna una cama de la sala correspondiente al servicio por el que está ingresado, aunque puede suceder que sea ingresado fuera de servicio.

1.1.6 Ingreso fuera de servicio

En el momento del ingreso el paciente debe ser asignado a un servicio especializado para atender las dolencias que presente, puede suceder que las salas de dicho servicio estén llenas y sea urgente el ingreso del paciente, en este caso se realiza lo que se conoce como ingreso fuera de servicio. Esto significa que se ingresa al paciente en un servicio que no es el que le corresponde y se especifica para el que debería ser trasladado en caso de liberarse alguna cama del servicio correspondiente.

1.1.7 Sala

Recinto o lugar del hospital en el que se encuentran habilitadas las camas donde se realizarán los ingresos. Perteneciente a determinado servicio.

1.1.8 Cama

Ubicación física en la que se encuentra el paciente ingresado. Perteneciente a una sala determinada.

1.1.9 Servicio

Tiene asociadas salas y camas. Hace referencia a las diferentes especialidades del hospital, por ejemplo: Nefrología, Dermatología, Cardiología, Neurología, entre otras.

1.1.10 Egreso

Es la salida o Alta del hospital. El paciente egresa porque mejoró totalmente, o por causas de fallecimiento.

1.1.11 Transferencia

Movimiento hospitalario que se lleva a cabo por diversas razones entre las que se encuentran: cuando el paciente presenta problemas de salud que no se pueden solucionar en el servicio en el cual se encuentra y es necesario trasladarlo a otro, por problemas de comodidad o físicos del paciente, o por condiciones físicas de la cama en la que se encuentra ubicado. Cabe destacar que es posible la realización de una transferencia que tenga como destino el mismo servicio en el cual se encuentra el paciente.

1.2 Situación problemática

El Sistema de Información Hospitalaria alas HIS cuenta con un Módulo Admisión, el cual se encarga de gestionar la información referente a los ingresos y la creación de la Historia Clínica Electrónica en las instituciones hospitalarias. También brinda la posibilidad de generar 38 reportes basados en las estadísticas del movimiento hospitalario del paciente desde que ingresa hasta que egresa de la institución. La gran cantidad de reportes puede dificultar el trabajo de los usuarios, al no tener una manera más general, reducida y fácil de manejar la información. De igual manera muchos de los reportes existentes tienen en común varios de los datos que manejan, siendo repetitivos en algunas ocasiones, careciendo además de gráficos que visualicen el comportamiento de la información. Además no es posible imprimir dichos reportes ni disponer de ellos sin depender del sistema.

1.3 Sistemas automatizados existentes vinculados al campo de acción

Actualmente existen numerosos Sistemas de Información Hospitalaria en todo el mundo que poseen módulos relacionados con los procesos que se llevan a cabo en el área de Admisión de las instituciones hospitalarias, pocos de ellos tienen entre sus funcionalidades la generación de reportes y gráficos, que brinden información estadística en esta área de las instituciones sanitarias. A continuación se analizan los de mayor importancia con el objetivo de buscar funcionalidades que puedan enriquecer la solución al problema científico.

1.3.1 Hosix-V

Es uno de los productos desarrollados por la empresa española SIVSA, con más de 15 años de experiencia en el desarrollo de software para el sector de la salud. Utiliza una arquitectura cliente-servidor por lo que es accesible desde un navegador web mediante el protocolo http. Está compuesto por un grupo de módulos orientados a la gestión específica de cada servicio o departamento, cubriendo los aspectos

más importantes de una unidad de salud. Ha sido desarrollado utilizando como lenguaje de programación Microsoft Visual Basic y herramientas que son propiedades de la compañía Microsoft.

Posee un Módulo Hospitalización que permite controlar los procesos correspondientes a los ingresos de los pacientes. Su organización autoriza al sistema a registrar e informar, de cualquier movimiento desde la llegada del paciente hasta su salida, así como de los pre-ingresos existentes. Ofrece una gestión completa de las camas hospitalarias. Anexa la información de la actividad realizada al historial del paciente. Permite ver el nivel de ocupación, las camas libres por totales y por servicio, los traslados, facilitando con todo ello la optimización de recursos del hospital. Facilita la obtención de cualquier tipo de documentación relacionada con el ámbito de la hospitalización, tal como el justificante de asistencia, la hoja estadístico-clínica, etc. Y genera reportes estadísticos oficiales como el libro de registro o el resumen de altas para el instituto nacional de estadística. (6)

1.3.2 Galenhos(R)

El Sistema Integrado de Gestión Hospitalaria Galenhos (R) ha sido diseñado con el propósito de apoyar a los establecimientos de salud en el correcto registro de información, clínica o administrativa, y la generación de información gerencial. Su desarrollo nace de la asistencia técnica prestada por la agencia de Estados Unidos para el desarrollo internacional (USAID), a través del proyecto Partners for Health Reform plus (PHRplus), a un hospital de Perú. Entre las ventajas que ofrece este sistema están:

1. Información estandarizada: favorece un registro adecuado, donde se utilizan todos los estándares establecidos por el ministerio de salud.
2. Bases de datos consolidadas y exportables: eliminan los problemas provenientes de la existencia de sistemas paralelos, incompatibles e incommunicados entre sí.
3. Generación de reportes clínicos de uso gerencial. Facilita la evaluación de su capacidad resolutive y eficiencia en el manejo de los recursos.
4. Altos estándares de seguridad informática: responden a la elevada sensibilidad de la información que se maneja, protegiendo la operación regular del sistema y favoreciendo los controles o auditorias.
5. Diseño modular: considera la ampliación y futuros desarrollos, que se extiendan en toda la complejidad hospitalaria con módulos adicionales.

Entre sus módulos está Hospitalización que considera la filiación de pacientes y registra diagnósticos y procedimientos. Este módulo también brinda la posibilidad de emitir reportes relacionados fundamentalmente con la información del movimientos hospitalario de los pacientes, de forma general se brindan estadísticas de ingresos, transferencias y egresos hospitalarios; de indicadores hospitalarios por departamentos, servicios, y por meses; se generan diversos reportes de camas y sobre la estancia del paciente en el hospital. Genera además, otros reportes de mortalidad hospitalaria combinados de distintas formas por departamentos, especialidad y diagnóstico del paciente. También, cuenta con un submódulo para la administración de camas hospitalarias.

Fue desarrollado en el lenguaje de programación Visual Basic 6.0 y utiliza el Microsoft SQL server como gestor de base de datos. (7) La principal desventaja del sistema Galenhos (R) radica en que no es un sistema multiplataforma y se ejecuta mediante una aplicación de escritorio.

1.3.3 X-HIS

Es una solución global de la compañía ISOFT, del grupo IBA Health, en un entorno abierto que, además de gestionar electrónicamente la historia clínica del paciente, permite la integración con otros sistemas de información. Este software funciona en múltiples plataformas (Unix, Windows, Linux), es independiente de la base de datos y posee una arquitectura cliente-servidor y en 3 capas. Incluye un Módulo Admisión diseñado para registrar y controlar la información administrativa del paciente durante su paso por el área de hospitalización. Los profesionales sanitarios podrán acceder a datos personales del paciente, así como a toda la información clínica recopilada durante su paso por hospitalización. Es software propietario y se comercializa bajo altos precios por lo que no constituye una opción viable para los países subdesarrollados. (8)

Tras la implantación del sistema X-HIS en el Hospital Nuestra Señora de Fátima, los profesionales sanitarios que laboraban en esa institución precisaban de una nueva funcionalidad que facilitase el acceso inmediato a los datos del paciente a través del reconocimiento del habla, funcionalidad que facilita y agiliza los procesos asistenciales, aportándoles una mayor calidad. El sistema de reconocimiento de habla SpeechMagic de Philips se presentaba como la mejor solución, gracias a su alto grado de integración con x-HIS de iSOFT, aportando resultados satisfactorios en la elaboración de reportes. (9)

1.3.4 SIGHO

El Sistema de Información para la Gerencia Hospitalaria (SIGHO) 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. Se apoya de estándares internacionales para el diagnóstico de enfermedades y realización de procedimientos tales como el CIE-10 y CIE9MC.

Cuenta con un Módulo Admisión que permite registrar los datos de los pacientes hospitalizados para después utilizarlos para búsquedas. Brinda información actualizada a los módulos con los que se relaciona como son Consulta externa, Cirugía y Emergencia. Permite generar reporte de admisiones, para conocer los pacientes hospitalizados en la institución, y reporte de seguimiento de hospitalización por paciente, el cual permite conocer las áreas por las cuales un paciente ha transitado durante su estancia u hospitalización. El sistema tiene como inconveniente ser un software propietario que se ejecuta mediante una aplicación escritorio. (10)

1.3.5 Galen Hospital

En Cuba, se han desarrollado también sistemas de gestión hospitalarios, tal es el caso de la empresa SOFTEL que brinda soluciones para la salud y que radica en la Universidad de las Ciencias Informática. Uno de los productos desarrollados por SOFTEL es Galen Hospital, sistema que está orientado hacia la informatización de la gestión de pacientes como elemento básico de control para mejorar la atención médica.

Brinda la información requerida para la actividad gerencial a todos los niveles y la elaboración de reportes estadísticos. Incluye un Módulo Hospitalización que asume los procesos vinculados con el ingreso, traslado y alta de los pacientes, además de llevar un control de las historias clínicas de los mismos. En este módulo, a partir del registro del informe de egreso, se obtienen estadísticas sobre la actividad del centro médico, entre las que se encuentran algunos reportes de morbilidad, reporte por tiempo de estadía, promedio de estadía por diagnóstico, promedio de estadía por servicio, entre otros. (11)

Este sistema utiliza como gestor de bases de datos SQL server 2000 y sus módulos corren sobre los sistemas operativos Windows 2000 y Windows XP, por lo que los principales inconvenientes de este

sistema es que es una aplicación que se ejecuta en el escritorio y además depende de tecnologías que son propiedades de compañías privadas.

La utilización de alguno de estos sistemas no resolvería el problema científico identificado, ya que, el mismo, está centrado en cómo optimizar un grupo de reportes pertenecientes a un módulo de un sistema en específico, por lo que no es necesario integrar las funcionalidades que poseen estos sistemas, sino, utilizar lo aprendido en el estudio de los mismos, para mejorar la calidad y aumentar la eficiencia de la solución.

Este análisis ha sido útil para demostrar que en la optimización de los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS, no pueden faltar reportes como son, el de ingresos, transferencias y egresos, el de movimientos hospitalarios por paciente, el de indicadores por departamentos y servicios. También es importante conocer sobre el tiempo de estancia de los pacientes en el hospital, de los fallecimientos, y del uso de las camas.

Por todo lo antes expuesto, la solución del problema estará centrada en agrupar los reportes existentes de forma tal que no existan ambigüedades entre ellos. Deben ser capaces de brindar información relevante para el análisis de estadísticas hospitalarias, deben ser flexibles, mostrar gráficos que representen la información y sobre todo permitir seleccionar el formato al que se desea exportar el reporte.

1.4 Tendencias y tecnologías actuales a considerar

A continuación se profundiza en las principales características de las tecnologías y herramientas que se utilizan en el proceso de desarrollo, teniendo en cuenta las características del entorno donde se desplegará el sistema en un futuro.

1.4.1 Navegador Web

Un navegador o navegador web (del inglés, web browser) es un programa que permite ver la información alojada en un servidor contenida en una página web. El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos. (12)

1.4.2 Aplicación web

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo. (13)

1.4.3 Arquitectura cliente-servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa denominado servidor, que le da respuesta. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Entre las características fundamentales que presenta la arquitectura cliente-servidor se encuentran las siguientes:

1. El Cliente y el Servidor pueden actuar como una sola entidad o como entidades separadas, y realizar actividades o tareas independientes.
2. Un servidor da servicio a múltiples clientes en forma concurrente.
3. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
4. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra al combinar los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, lo que proporciona el servicio más efectivo para el usuario final. (14)

1.4.4 Patrones de Arquitectura y Diseño

1.4.4.1 Arquitectura en capas

El objetivo esencial de la arquitectura en capas es la separación de la lógica de negocios de la lógica de diseño. La ventaja principal de esta es que, el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles.

En el diseño de sistemas informáticos actuales se suele usar las arquitecturas multinivel o Programación por capas, donde a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables. El diseño más utilizado actualmente es el diseño en tres niveles (o en tres capas), el cual presenta la siguiente estructura:

Capa de presentación: es la que ve el usuario, también se le denomina "capa de usuario", presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de

proceso. También es conocida como interfaz gráfica y debe tener la característica de ser amigable, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o de lógica del negocio porque 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 solicitar al gestor de base de datos almacenar o recuperar datos.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. 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. (15)

1.4.4.2 Modelo Vista Controlador

Modelo Vista Controlador (por sus siglas en inglés MVC), es un patrón de diseño de arquitectura de software que separa la parte lógica de una aplicación de su presentación. Básicamente sirve para separar el lenguaje de programación del HTML lo máximo posible y para poder reutilizar componentes fácilmente. Utilizar este patrón brinda la facilidad de agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de otras capas y facilita el mantenimiento en caso de errores.

Modelo: representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

Vista: es la información presentada al usuario. Una vista puede ser una página web o una parte de una página.

Controlador: actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página. (16)

1.4.5 Tecnologías utilizadas en el proceso de desarrollo

La Tecnología constituye un conjunto de conocimientos técnicos, que permiten diseñar y crear bienes o servicios que facilitan la adaptación al medio y satisfacen las necesidades de las personas. A continuación

se explican las tecnologías que se proponen para el proceso de desarrollo, las cuales se encuentran agrupadas por la capa donde están presentes.

1.4.5.1 Java

Java es un lenguaje de programación orientado a objetos que toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible. El lenguaje Java fue creado con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++. (17)

1.4.5.2 Capa de presentación

1.4.5.2.1 Java Server Faces (JSF)

JSF es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE (17). Este incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Dos librerías de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Beans administrados.

1.4.5.2.2 RichFaces

Es una biblioteca de componentes para JSF y un avanzado framework para la integración de AJAX con facilidad en la capacidad de desarrollo de aplicaciones de negocio. Los componentes RichFaces vienen listos para su uso sin tener que hacer muchas configuraciones, por lo que los desarrolladores pueden ahorrar tiempo y aprovechar las características de los componentes para crear aplicaciones web que proporcionen mejoras en gran medida y más fiabilidad. También incluye un fuerte apoyo para el soporte de temas de aplicaciones JSF y aprovecha al máximo los beneficios de JSF framework incluyendo, la validación y conversión junto con la gestión de recursos estáticos y dinámicos. (19)

1.4.5.2.3 XHTML

El lenguaje extensible de marcado de hipertexto (por sus siglas en inglés XHTML) es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML. Las aplicaciones que contengan XHTML apreciarán las siguientes mejoras:

- Los documentos XHTML son conformes a XML. Como tales, son fácilmente visualizados, editados y validados con herramientas XML estándar.
- Los documentos XHTML pueden escribirse para que funcionen igual o mejor que lo hacían antes tanto en las aplicaciones de usuario conformes a HTML 4.0 como en las nuevas aplicaciones conformes a XHTML 1.0.
- A medida que la familia XHTML evolucione, los documentos conformes a XHTML 1.0 estarán más preparados para interactuar dentro de y entre distintos entornos XHTML.
- Se aseguran la compatibilidad con aplicaciones de usuario pasadas y futuras. (20)

1.4.5.2.4 Seam UI

Serie de controles Seam altamente integrables con JBoss Seam. Están dirigidos a complementar los controles JSF incorporados y los controles de otras bibliotecas externas.

1.4.5.2.5 Facelets

Java Server Facelets es un framework para plantillas centrado en la tecnología JSF, por lo cual se integran de manera muy fácil. Este framework incluye muchas características siendo las más importantes:

- Fácil composición de componentes.
- Creación de etiquetas lógicas a la medida.
- Funciones para expresiones.
- Desarrollo amigable para el diseñador gráfico.
- Creación de librerías de componentes.

1.4.5.2.6 Ajax4jsf

Ajax4jsf es una librería que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Java script. Mediante este framework es posible variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. Ajax4jsf permite dotar a las aplicaciones JSF de contenido mucho más profesional con muy poco esfuerzo. (21)

1.4.5.3 Capa de negocio

1.4.5.3.1 JBoss Seam

Es un framework que combina a dos frameworks: Enterprise JavaBeans (EJB) y JavaServerFaces (JSF). (22) Mediante él se puede acceder a cualquier componente EJB desde la capa de presentación haciendo referencia a su nombre de componente Seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador; inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar del navegador. (23)

Puede automáticamente generarse una aplicación web a partir de una base de datos existente mediante una herramienta de línea de comandos llamada seam-gen incluida con el framework. Seam puede ser integrado con las bibliotecas de componentes JBoss RichFaces.

1.4.5.3.2 JBoss Server

Es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como clustering, caching y persistencia gracias a que es una plataforma certificada J2EE. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, esto hace que el desarrollo de las aplicaciones empresariales sean mucho más simples. (24)

1.4.5.3.3 JBoss tools

Es un conjunto de plugins y mejoras para Eclipse diseñado para ayudar a los desarrolladores de JBoss y J2EE a crear aplicaciones utilizando el menor tiempo y esfuerzo posible. Esta suite tiene herramientas para desarrollar en Seam 2.1, integración con Hibernate Tools, JBoss AS Tools, soporte para Drools y JBPM.

1.4.5.3.4 JasperReports

Es una librería de clases de Java de código abierto diseñada para facilitar el agregar capacidades de reporte a las aplicaciones Java. No es una herramienta por sí sola, por lo que no se puede instalar. Para utilizar JasperReports es necesario añadirlo a las aplicaciones Java por medio de la inclusión de su librería.

Esta librería permite la creación de informes Java libre que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML. Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. JasperReports se usa comúnmente con iReport, una herramienta de código abierto para la edición de informes. (25)

1.4.5.3.5 JFreeChart

Es una librería para gráficos escrita completamente en Java que facilita mostrar gráficos de calidad profesional en las aplicaciones, ya sean web o de escritorio. Constituye un API consistente y bien documentado con soporte para un amplio rango de tipos de gráficos. Presenta un diseño flexible fácilmente extensible, y la posibilidad de ser usado en aplicaciones web y de cliente. JFreeChart es software libre y está distribuido bajo la licencia LGPL, que permite el uso en aplicaciones propietarias.

1.4.5.4 Capa de datos

1.4.5.4.1 Enterprise JavaBeans (EJB)

Los EJB son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE. Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. (26)

1.4.5.4.2 Java Persistence API (JPA)

JPA es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue su diseño es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs). (26)

1.4.5.4.3 Hibernate

Es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU LGPL. (27) Este busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). (28) Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información, Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objetos.

Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Está diseñado para ser flexible en cuanto al esquema de tablas

utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Además ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas. Para Java, Hibernate puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma. (29)

1.4.5.4.4 PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos de código abierto, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios. Además incorpora una estructura de datos array y funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc. Permite la declaración de funciones propias, así como la definición de disparadores, soporta el uso de índices, reglas y vistas, incluye herencia entre tablas. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos y es posible mantener la integridad referencial. (30)

1.5 Herramientas

1.5.1 Eclipse

Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega 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.

Eclipse dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real y a través de plugins libremente disponibles, es posible añadir control de versiones con Subversion e integración con Hibernate. (31)

1.5.2 Visual Paradigm

Es una herramienta CASE (Computer-Aided Software Engineering), es decir, ayuda a desarrollar básicamente. Utiliza el Lenguaje de Modelado Unificado UML (Unified Modeling Language) como lenguaje de modelado y soporta el ciclo de vida completo del desarrollo de software. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar, es compatible entre ediciones. (32)

1.5.3 PgAdmin

Es la plataforma de desarrollo de PostgreSQL y la más avanzada base de datos de código abierto en el mundo. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta la elaboración de complejas bases de datos. La interfaz gráfica soporta todas las características de PostgreSQL y hace fácil la administración. La conexión con el servidor puede hacerse utilizando el protocolo TCP-IP y puede ser encriptado SSL para la seguridad. No se requieren drivers adicionales para comunicarse con el servidor de base de datos. Es desarrollado por una comunidad de expertos de PostgreSQL de todo el mundo y está disponible en más de una docena de idiomas. (33)

1.5.4 iReport

Es un constructor y diseñador visual de informes, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport está además integrado con JFreeChart, una de las bibliotecas gráficas Open Source más difundidas para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, etc. Entre sus características más importantes se encuentran las siguientes:

- 100 por ciento escrito en el lenguaje Java y además Open Source y gratuito.
- Maneja el 98 por ciento de las etiquetas de JasperReports.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreportes.

- Soporta internacionalización nativamente.
- Recopilador y exportador integrados.
- Soporta JDBC y JavaBeans como orígenes de datos (estos deben implementar la interface JRDataSource).
- Incluye asistentes para crear automáticamente informes, generar los subreportes y las plantillas.
- Facilidad de instalación. (34)

1.6 Metodologías de desarrollo de software

1.6.1 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo Software (en inglés RUP, Rational Unified Process), es un marco de desarrollo de software que se caracteriza por su desarrollo iterativo e incremental, por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo. Es guiado por los casos de uso, un caso de uso describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante, los casos de uso guían el diseño construcción y prueba del sistema, esto significa que guían el proceso de desarrollo. RUP está centrado en la arquitectura, lo que le permite a los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas, dejando a un lado los detalles. (35)

RUP es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto establecido. RUP consta de cuatro fases o etapas: Inicio, Elaboración, Construcción y Transición, además contiene flujos de trabajos, los cuales se dividen en flujos de trabajo de desarrollo (Modelado del negocio, Requerimientos, Análisis y Diseño, Implementación y Pruebas) y flujos de trabajo de soporte (Administración de proyecto, Administración de configuración y cambios, Ambiente). El Proceso Unificado utiliza UML como lenguaje de modelado.

1.6.2 Lenguaje Unificado de Modelado (UML)

UML son un grupo de especificaciones de notación orientadas a objeto, las cuales están compuesta por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. Es el

lenguaje de modelado de sistemas de software más conocido en la actualidad; permite la especificación, visualización, construcción y documentación de elementos de la Ingeniería del Software. Los varios tipos de diagramas de UML muestran diferentes aspectos de lo que se quiere representar. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, por lo que no es un proceso.

En este capítulo se realizó un estudio de los Sistemas de Información Hospitalaria existentes en el mundo que tienen entre sus funcionalidades la generación de reportes estadísticos, relacionados con los procesos que se desarrollan en el área de Admisión de las instituciones sanitarias, lo que constituye un factor principal para hacer más robusto y abarcador el Sistema alas HIS en cuanto a funcionalidades, que engloben todos los requisitos que necesitan los clientes y que satisfagan las necesidades de generación de información que se requieren. Se determinó que existen funcionalidades que no pueden omitirse en la optimización de los reportes del Módulo Admisión del Sistema alas HIS, tales como: reporte de movimientos hospitalarios, reporte de movimientos hospitalarios por servicio, reporte de uso de camas y de movimientos hospitalarios por paciente.

Se analizaron las tecnologías, herramientas y metodologías que forman parte de la propuesta tecnológica para el desarrollo de los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS. Y se decidió para el desarrollo de los mismos, debido al manejo eficiente y racional de los recursos de memoria, por estar bajo licencias de código abierto y facilitar el trabajo de los desarrolladores, utilizar RichFaces como biblioteca de componentes web, Hibernate para el mapeo objeto-relacional, Jboss 4.2 GA como servidor de aplicaciones web, Eclipse como entorno de desarrollo integrado, Seam como framework de integración, Visual Paradigm como herramienta de modelado, PostgreSQL como servidor de base de datos y pgAdmin para su administración, iReport para el diseño visual de los reportes, JasperReports para la generación de los mismos y JFreeChart para la generación de gráficos.

Capítulo 2. Descripción de la Arquitectura

En el presente capítulo se realiza una descripción de los requisitos no funcionales de software y de la arquitectura definida para el desarrollo. También se realiza un análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser reusados en el presente trabajo, así como aspectos importantes de seguridad, la vista de despliegue del sistema y las estrategias de codificación.

2.1 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales y son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no funcionales, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. (36)

2.1.1 Requerimientos de usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- Para alcanzar un nivel Elemental asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 20 días de preparación, obteniendo la categoría de Usuarios normales.
- Para alcanzar un nivel Avanzado asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 30 días de preparación, obteniendo la categoría de Usuarios avanzados.

2.1.2 Requerimientos de soporte

Seguridad de acceso y administración de usuarios:

- Se mantendrá seguridad y control entre usuarios, garantizando su acceso solo a los niveles establecidos de acuerdo con la función que realizan. Las contraseñas podrán cambiarse por el propio usuario o por el administrador del sistema.
- Se mantendrá un segundo nivel de seguridad entre estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán

todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.

- El sistema proporcionará un registro de actividades (log) de cada usuario. Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvallas realizadas.

Monitoreo de funcionamiento:

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

Respaldo y recuperación de base de datos:

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

Auditoría:

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

Configuración de parámetros:

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

2.1.3 Restricciones de diseño

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

2.1.4 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.

2.1.5 Requerimientos de interfaz

Interfaces de usuario:

- Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La 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.

2.1.6 Requerimientos de rendimiento

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

2.1.7 Requerimientos de hardware

Estaciones de trabajo:

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se recomienda Internet Explorer 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.

Servidores:

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

2.1.8 Requerimientos de software

El sistema debe funcionar en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

2.2 Descripción de la arquitectura. Fundamentación.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema y consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Para la creación del Sistema de Información Hospitalaria alas HIS se utiliza la arquitectura basada en el patrón de diseño Modelo Vista Controlador, sobre el cual se abordó en el capítulo anterior.

La Vista o capa de presentación está desarrollada básicamente con JSF, se utiliza la librería de componentes Richfaces 3.2.0 G.A., esta integra fácilmente con el framework de integración escogido (Seam) y permite generar vistas no necesariamente basadas en HTML (PDF, etc.), adiciona además el framework de extensión AJAX para los controles JSF básicos Ajax4jsf. Incluye conversión y validación de campos, establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, un modelo orientado a eventos y combinado con Facelets, se elimina la necesidad de dos motores de renderización (uno para JSF y otro para JSP) lo que mejora el rendimiento en general además de brindar la capacidad añadida de la tecnología de plantillas de Facelets. Por su parte los controles para UI de Seam adicionan varias mejoras a JSF, desde validación, expresiones EL extendidas, integración de la navegación en la UI basada en pageflows o procesos del negocio, etc.

La capa de negocio está compuesta por clases controladoras, que definen la lógica del negocio conjuntamente con los datos que se validan en la capa de presentación. Todo esto se desarrolla mediante el framework Seam, un poderoso y moderno framework creado para unificar todas las tecnologías estándares JSF, EJB3, JPA. A estas clases, mediante anotaciones que provee Seam como marco de trabajo, se les puede especificar el contexto en que se encuentran, ya sea conversacional, evento, página, entre otros, los que definen el estado de los datos y las entidades que manejan. Se utiliza la librería JasperReports para la creación de reportes y la librería JFreeChart para la generación de gráficos.

En el Modelo, para el acceso a datos se usa la implementación de JPA de Hibernate 3.3, minimizando por un lado las configuraciones en XML sin chequeo de tipos y por otro lado usando los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam, se elimina gran parte del código infraestructural en cuanto a transacciones, la transmisión del contexto de persistencia, etc. (37)

2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reusados. Estrategias de integración.

La reutilización de código tiene gran importancia en la construcción de sistemas ya que, una de las ventajas que ofrece es la de economizar tiempo, minimizar las redundancias y aprovechar el trabajo anterior. La manera más fácil de llevar a cabo esta reutilización, es copiar total o parcialmente, pero resulta trabajoso mantener múltiples copias del mismo código, por lo que es recomendable mantener el código reusable en un solo lugar, para luego ser llamado en donde sea requerido.

Para el desarrollo del Sistema alas HIS se utilizan varios componentes de forma común en todos los módulos del sistema con el objetivo de lograr uniformidad en el desarrollo y mejorar la calidad del trabajo. Entre los componentes que se reutilizan están: la clase ActiveModule para conocer en que módulo y entidad se encuentra el usuario que está utilizando el sistema, la clase Bitácora para el control de las trazas de todas las acciones que se realizan con la aplicación; la clase User para saber qué usuario está trabajando con la aplicación en tiempo real; la clase ReportManager la cual brinda las funcionalidades más significativas para la generación de los reportes y gráficos permitiendo exportarlos a diferentes formatos de archivo.

También se utiliza la clase UbicacionesManager para el trabajo con las ubicaciones de las camas que participan en los movimientos hospitalarios. Del Módulo Admisión, se utiliza la clase Utiles_admision que permite reutilizar las funcionalidades más usadas en las clases controladoras del módulo, relacionadas con los departamentos, servicios, disponibilidad de camas, entre otras.

2.4 Seguridad

El sistema debe garantizar la confidencialidad, integridad y disponibilidad de la información que se maneja para evitar que esta sea accedida por personal no autorizado. Para ello se propone un control de acceso a nivel de usuarios y contraseñas, así como su diferenciación atendiendo al rol de cada uno para asegurar

que cada usuario pueda acceder solamente a la información que su rol requiera. También se cuenta con la existencia de una bitácora, en la cual se almacenan todas las operaciones llevadas a cabo por el usuario, quedando registrada la fecha, hora, así como la actividad que llevó a cabo el usuario en cuestión. Las contraseñas podrán ser cambiadas por los propios usuarios y por el administrador del sistema, este último es el encargado de otorgar y cambiar los permisos para acceder a los datos.

La información que se intercambia entre el sistema y otros sistemas debe ser cifrada, pues de esta forma se asegura que no se acceda o modifiquen los datos confidenciales que se manejan. Además, todos los datos que se hayan ingresado en el sistema no serán eliminados físicamente de la base de datos.

2.5 Vista de Despliegue

Los diagramas de despliegue muestran la disposición física de los distintos nodos que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Para que sea posible un despliegue exitoso del sistema, el usuario debe conectarse a este mediante una PC cliente a través de un navegador web. Las peticiones a través de protocolo HTTP serán procesadas por el servidor de aplicaciones, el cual se encargará de enviar la respuesta al cliente y hacer peticiones al servidor de base de datos mediante el protocolo TCP-IP en caso de que sea necesario. Además, el usuario puede imprimir informes con sus respectivos gráficos desde la PC cliente que estará conectada a una impresora mediante un puerto USB o LPT.

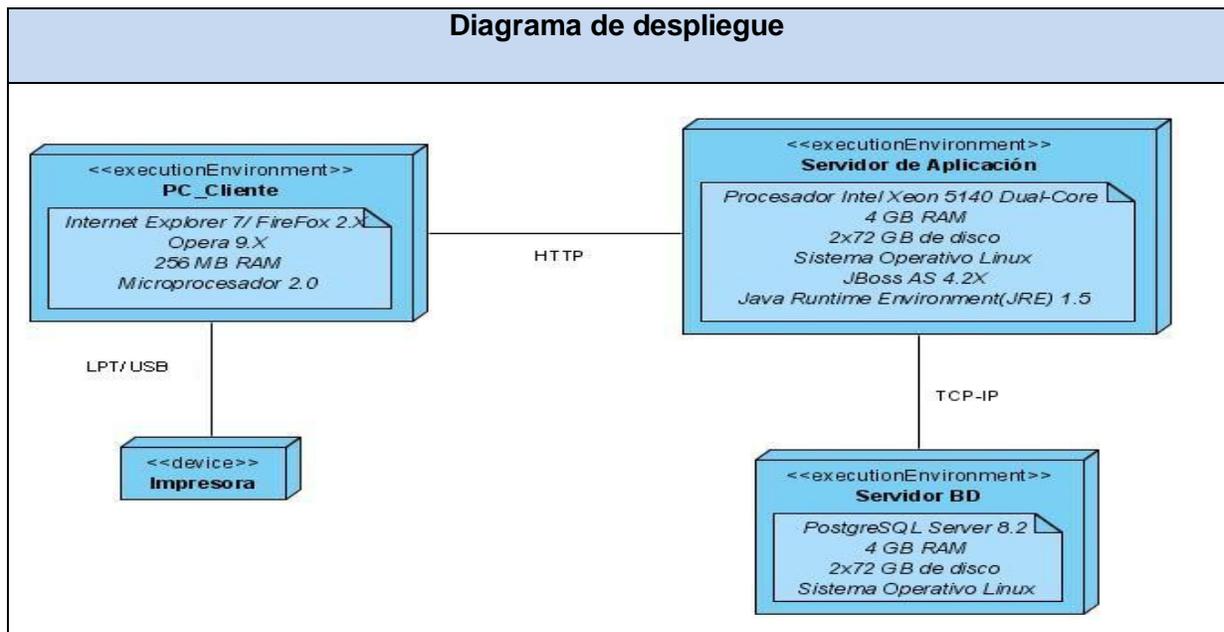


Figura 2.1. Diagrama de despliegue

2.6 Estrategias de codificación. Estándares y estilos a utilizar.

La legibilidad en el código fuente de un sistema es un atributo de calidad de primer orden. La programación es mucho más una tarea de comunicación con otras personas que una comunicación con una computadora, por eso es importante establecer y respetar estándares de codificación y lograr que la estructura del código fuente refleje la estructura lógica del programa. Para la implementación del sistema propuesto se han utilizado varios estándares de codificación como son:

Idioma

- Se debe utilizar el idioma español y las palabras no se acentuarán.

Indentación

- Inicio y fin de bloque: Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.
- Aspectos generales: El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla. Los

inicios ({) y cierre (}) de ámbito deber estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar ({) en la línea de un código cualquiera, esto requiere una línea propia.

Comentarios, separadores, líneas, espacios en blanco y márgenes

- Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.
- Se recomienda comentar al inicio de la clase o función, especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.
- Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
- Se recomienda usar espacios en blanco entre los operadores lógicos y aritméticos, para lograr una mayor legibilidad en el código. Ejemplo: `cama = nomCama`.
- Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.

No se debe usar espacio en blanco en los siguientes casos:

- Después del corchete abierto y antes del cerrado de un arreglo.
- Después del paréntesis abierto y antes del cerrado.
- Antes de un punto y coma.

Variables y constantes

El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamellCasing, la cual sugiere iniciar cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

Ejemplo: `nombrePaciente`

Clases y Objetos

- El objetivo fundamental es nombrar las clases e instancias de forma estándar para todas las aplicaciones.
- Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing, la cual sugiere iniciar cada palabra con letra mayúscula. Ejemplo: MiClase (). Para el caso de las instancias se comenzará con un prefijo que identificara el tipo de dato, este se escribirá en minúscula.
- Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing. Ejemplo: function BuscarPaciente (). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.
- El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

En este capítulo se describió la arquitectura del sistema la cual constituye una base fundamental para la construcción del software. Se explicaron los requisitos no funcionales del sistema con el propósito de lograr un software con calidad, vistosidad y seguro. Se analizaron los aspectos más significativos a tener en cuenta en cuanto a la seguridad del sistema para lograr un producto confiable y robusto. Además se definió la vista de despliegue que especifica la distribución física del sistema.

Capítulo 3. Descripción y análisis de la solución propuesta

En este capítulo se realiza una descripción del diseño del sistema dando a conocer los principales patrones de diseño que se ponen en práctica, además se muestran los diagramas de clases del diseño e interacción que forman parte del modelo de diseño. Se brinda una descripción de las clases u operaciones necesarias, el modelo de datos definido, así como una breve valoración de las técnicas de validación a utilizar.

3.1 Valoración crítica del diseño propuesto por el analista

La etapa de diseño de un sistema contribuye a obtener una arquitectura estable y sólida del mismo, y a crear un plano del modelo de implementación. Una vez que se logra obtener la arquitectura estable y los requisitos bien entendidos, el centro de atención se desplaza hacia la implementación.

En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, que proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debe ser conservada lo más fielmente posible cuando se da forma al sistema.

Para elaborar el diseño se utilizan un grupo de patrones, que no son más que modelos a seguir con el propósito de lograr determinados objetivos. Estos capturan la experiencia existente y probada por los seres humanos para promover buenas prácticas. Cada patrón describe un problema que ocurre una y otra vez en el entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada muchas veces sin hacerlo dos veces de la misma forma.

Existen varios tipos de patrones de diseño, todos encaminados a lograr mejores soluciones. Entre los más usados se encuentran los patrones GRASP, los cuales brindan grandes ventajas ya que, se encargan de asignar responsabilidades a las diferentes clases que se definen en el diseño, logrando como resultado un bajo acoplamiento en el sistema, es decir, existencia de poca dependencia entre las clases, lo que favorece la reutilización de las mismas.

Como parte de este patrón a cada clase se le asignó las tareas que podían ser realizadas, basándose en la información que dichas clases poseían, al mismo tiempo se les otorgó la habilidad de construir

instancias de otras clases que fueran necesarias para llevar a cabo las responsabilidades especificadas, trayendo como resultado la utilización de los patrones Experto y Creador, lográndose la conservación del encapsulamiento de la información.

En el diseño obtenido se evidencia el cumplimiento de los patrones Bajo Acoplamiento y Alta Cohesión, los cuales se encargan respectivamente de que las dependencias entre clases sean mínimas y que cada elemento lleve a cabo una única labor dentro de la aplicación, la cual no es realizada por ninguno de los otros elementos.

Por otra parte, en la capa de datos, para lograr mayor flexibilidad y para conseguir asimilar mejor los cambios que en un futuro se puedan producir en la base de datos, se utilizan los siguientes patrones:

Active record: consiste en que una fila de cualquier tabla de la base de datos se convierte en un objeto, de manera que se asocian filas únicas de la base de datos, con objetos del lenguaje de programación usado.

Identity field: la idea fundamental de este patrón es que guarda el campo identificador de la base de datos en un objeto para mantener la relación entre el objeto cargado en memoria y la fila correspondiente en la base de datos.

Foreign Key Mapping: consiste en mapear una asociación entre objetos por cada relación entre tablas por claves foráneas.

Query Object: mantener un objeto que permita hacer consultas a la base de datos.

Lazy load: el objeto que mapea una fila en la base de datos no necesariamente debe cargar toda la información desde un principio. Debe conocer la forma de obtenerla y cargarla solo cuando sea necesario.

Como parte del modelo de diseño se define una estructura de paquetes y subpaquetes que permite dividir el sistema en fracciones manejables para su futura implementación. Los paquetes fueron definidos por procesos y clases, cada proceso en el sistema se grafica en su paquete individual, utilizando las clases que se encuentran dentro del paquete repositorio, las cuales a su vez se encuentran separadas en 3 subpaquetes:

El subpaquete **Sesiones** está compuesto por las clases controladoras autogeneradas, personalizadas y las controladoras propias del proceso. Las autogeneradas fueron creadas por el entorno de desarrollo, las personalizadas son aquellas que fueron modificadas, y las últimas, aquellas que fueron creadas específicamente para el proceso. En el subpaquete **Entidades** se encuentran el conjunto de entidades autogeneradas que fueron obtenidas mediante el proceso de ingeniería inversa, utilizando el mapeo objeto-relacional de Hibernate. Además contiene las entidades personalizadas, que son aquellas que han sido modificadas. El subpaquete **Vistas** está conformado por el conjunto de clases que representan la interfaz de usuario, además de los ficheros que constituyen las plantillas de los reportes.

A continuación se presenta el diagrama de paquetes del diseño, donde se evidencia lo antes mencionado:

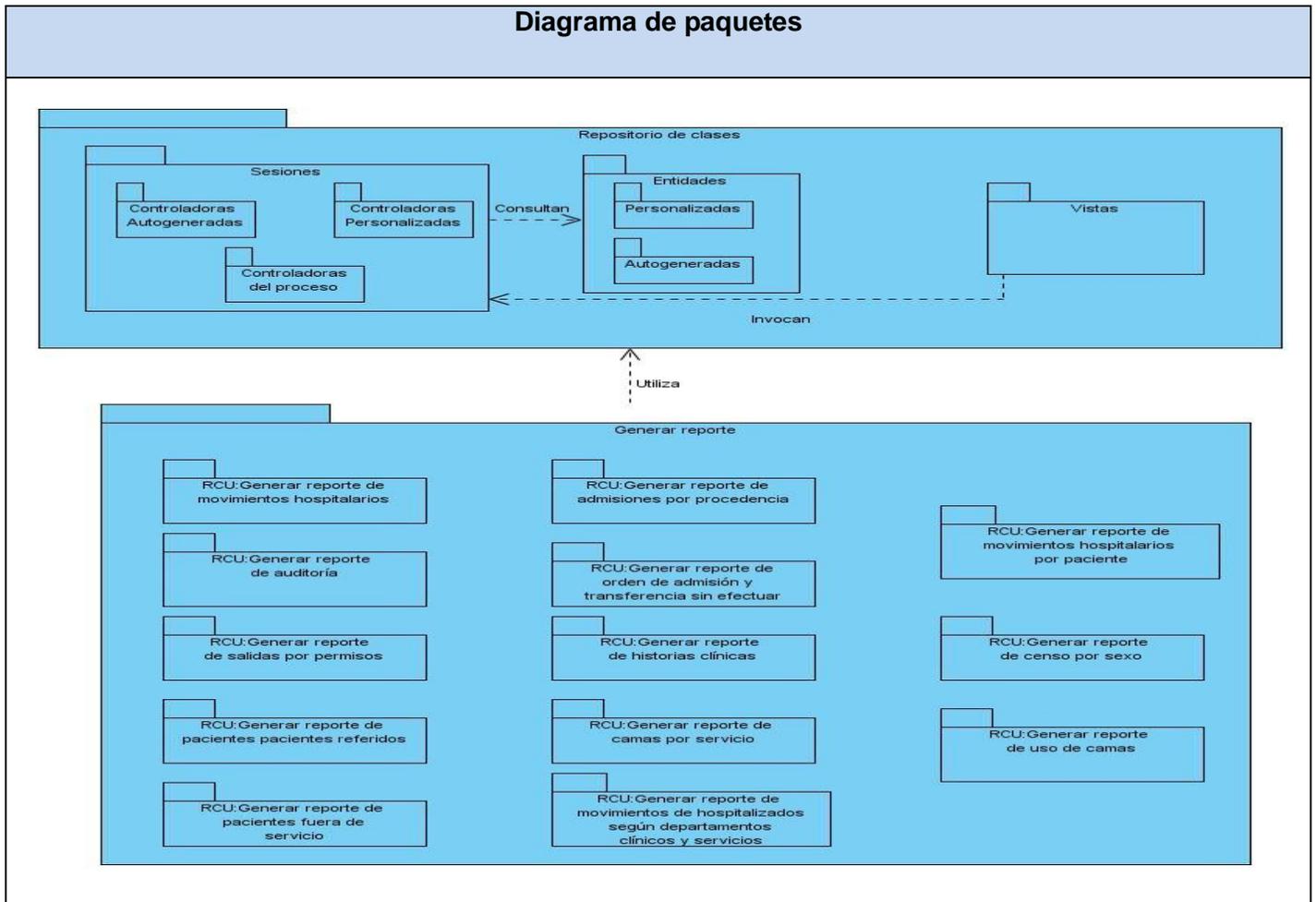


Figura 3.1. Diagrama de paquetes

Entre los elementos que componen el modelo de diseño se encuentran también los diagramas de clases del diseño y los de interacción. Los diagramas de clases del diseño contienen las clases que participan en los casos de uso, aunque algunas de ellas puedan participar en varios, además constituyen abstracciones de clases directamente utilizables en la implementación. Por otra parte en los diagramas de interacción se muestra un patrón de interacción entre objetos; de estos diagramas existen dos tipos, los de secuencia y los de colaboración, ambos basados en la misma información, pero cada uno enfatiza un aspecto particular.

Los diagramas de secuencia son los que generalmente se utilizan en la etapa de diseño, estos muestran una interacción ordenada según la secuencia temporal de eventos. En particular, muestran los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

A continuación se muestran varios diagramas de clases de diseño, y un conjunto de diagramas de secuencia que evidencian la realización de los casos de uso más significativos.

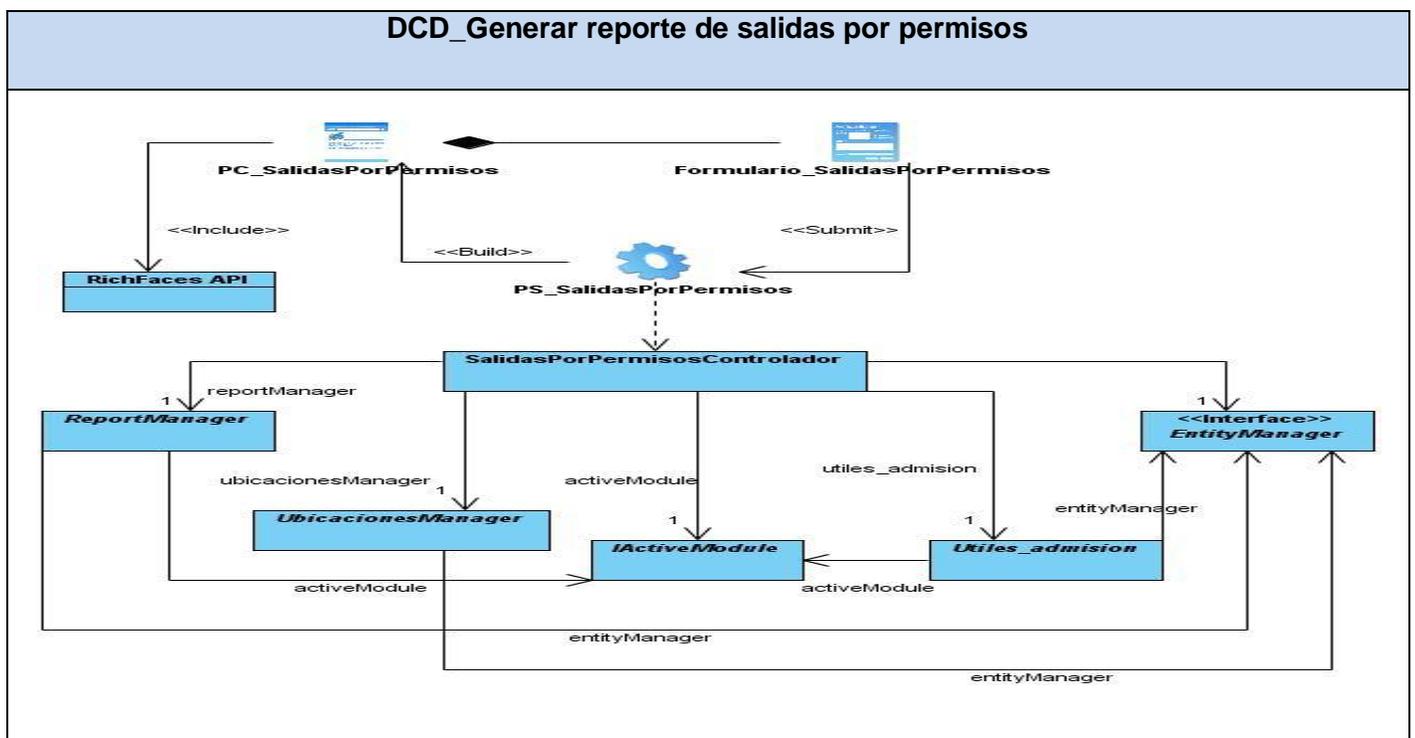


Figura 3.2. DCD_Generar reporte de salidas por permisos

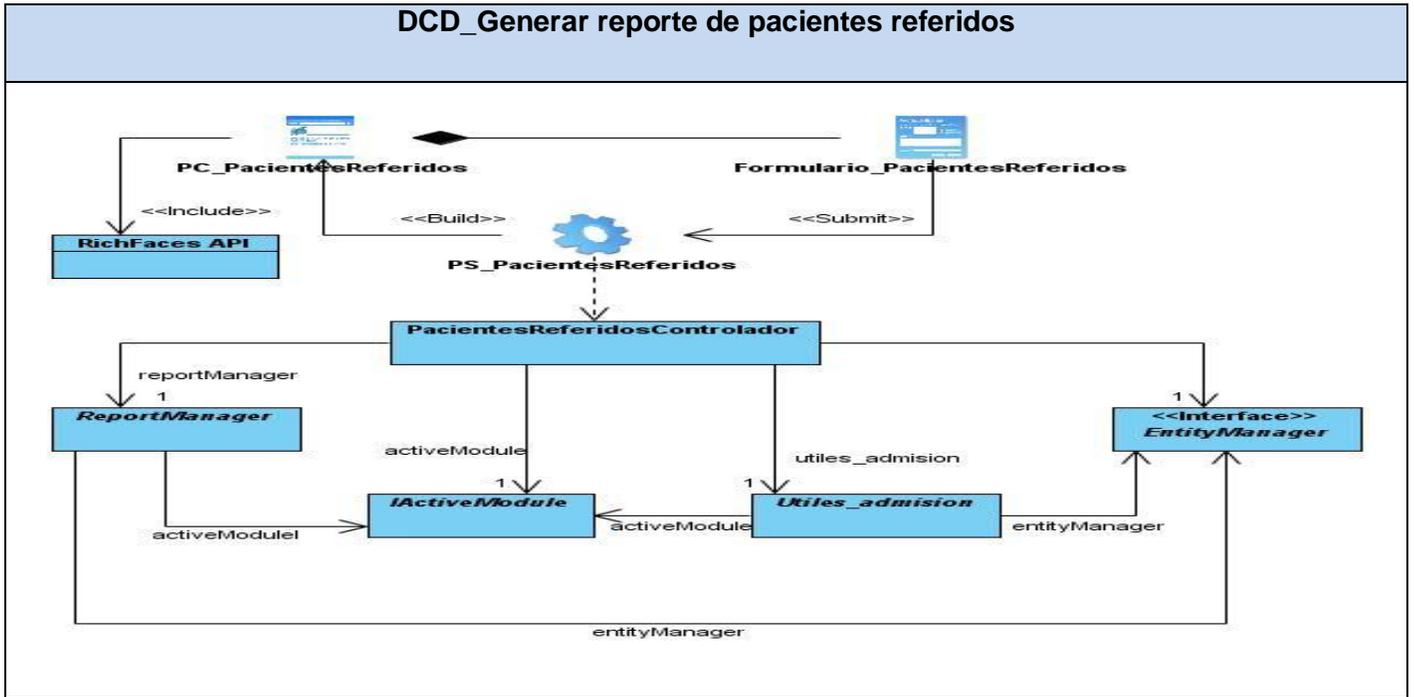


Figura 3.3. DCD_Generar reporte de pacientes referidos

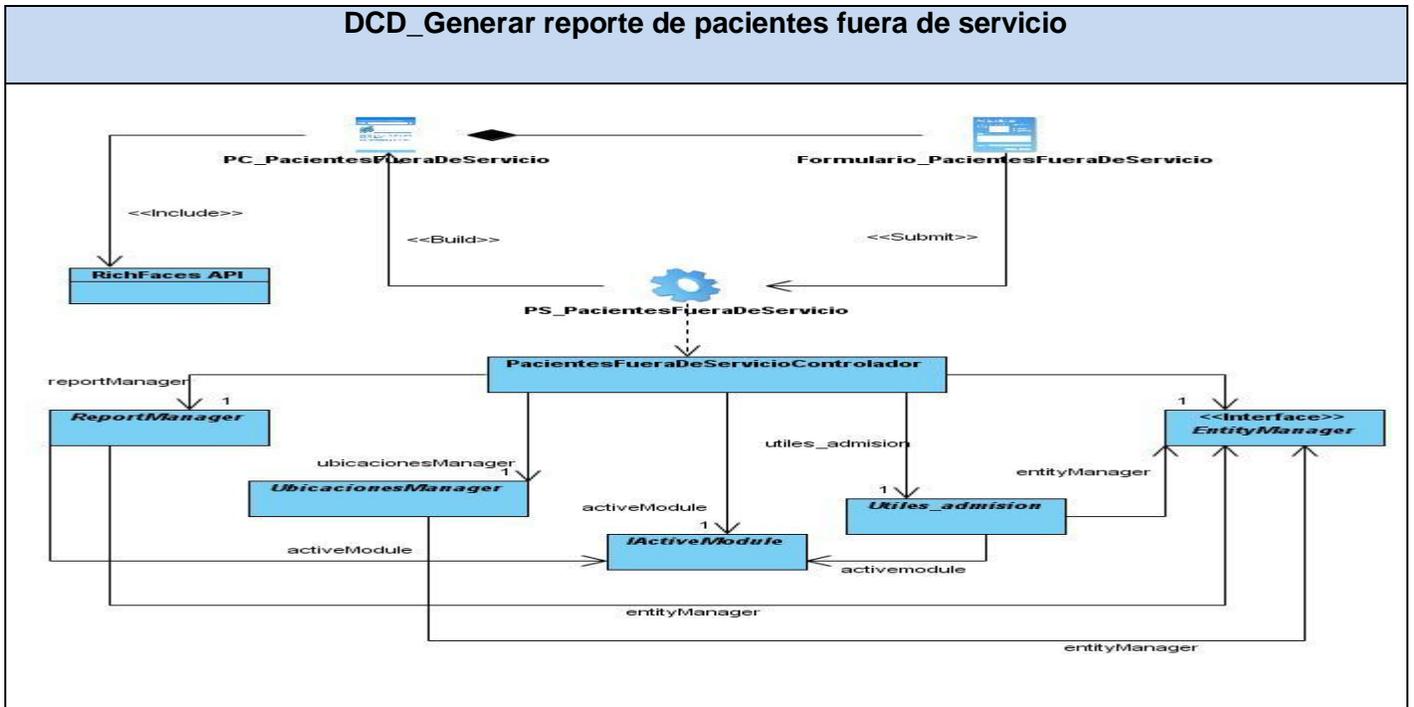


Figura 3.4. DCD_Generar reporte de pacientes fuera de servicio

DCD_Generar reporte de orden de admisión y transferencia sin efectuar

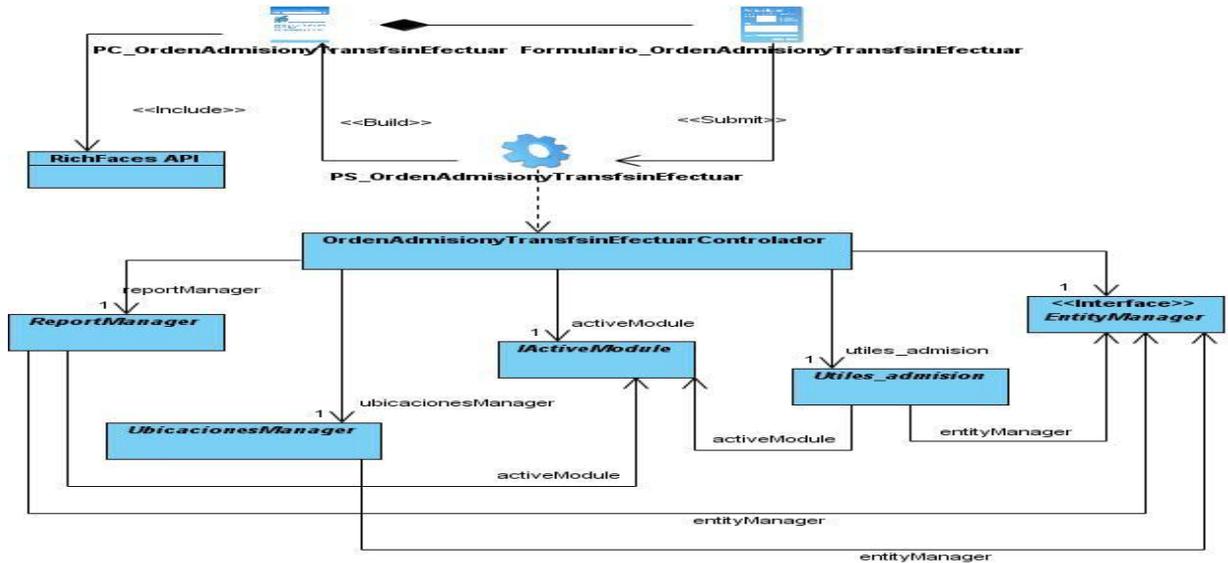


Figura 3.5. DCD_Generar reporte de orden de admisión y transferencia sin efectuar

DCD_Generar reporte de auditoría

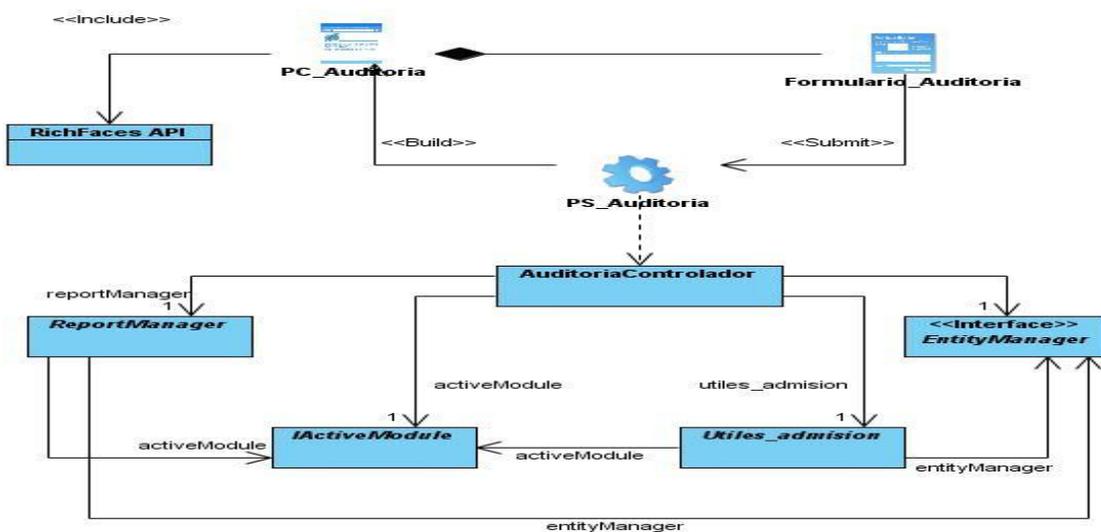


Figura 3.6. DCD_Generar reporte de auditoría

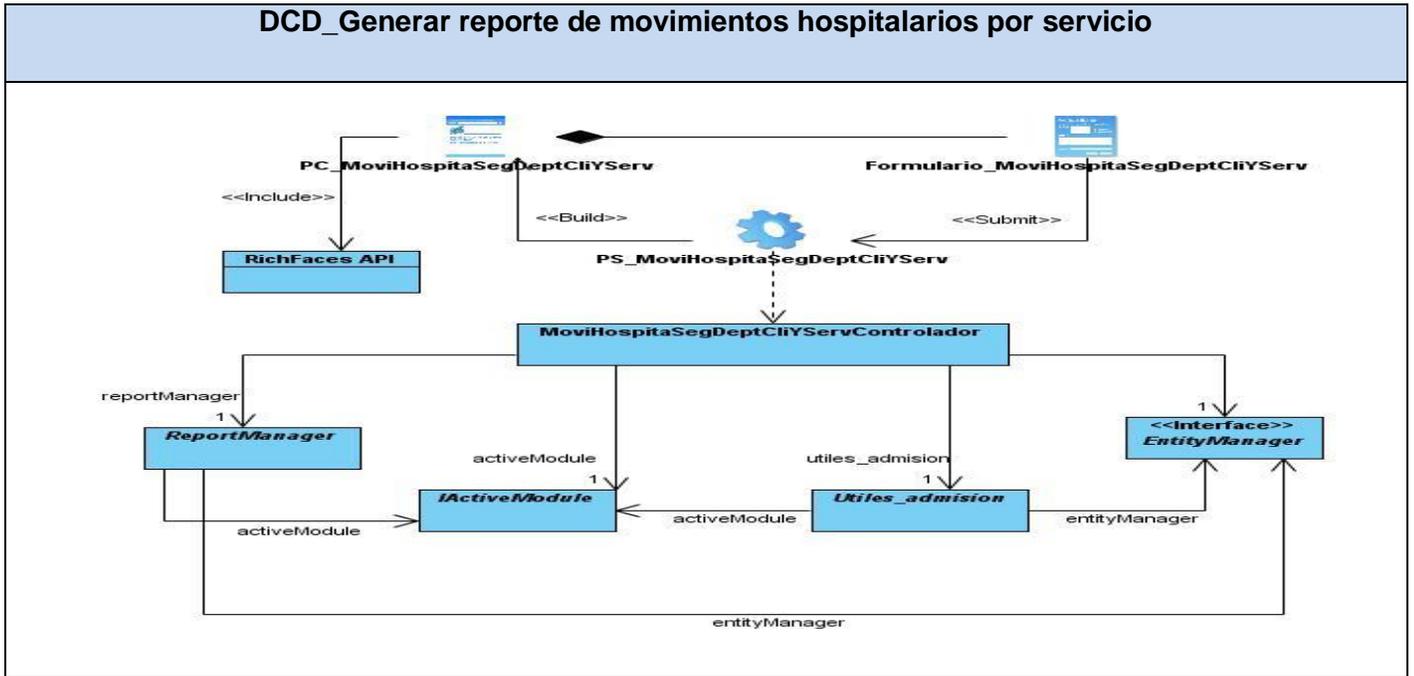


Figura 3.7. DCD_Generar reporte de movimientos hospitalarios por servicio

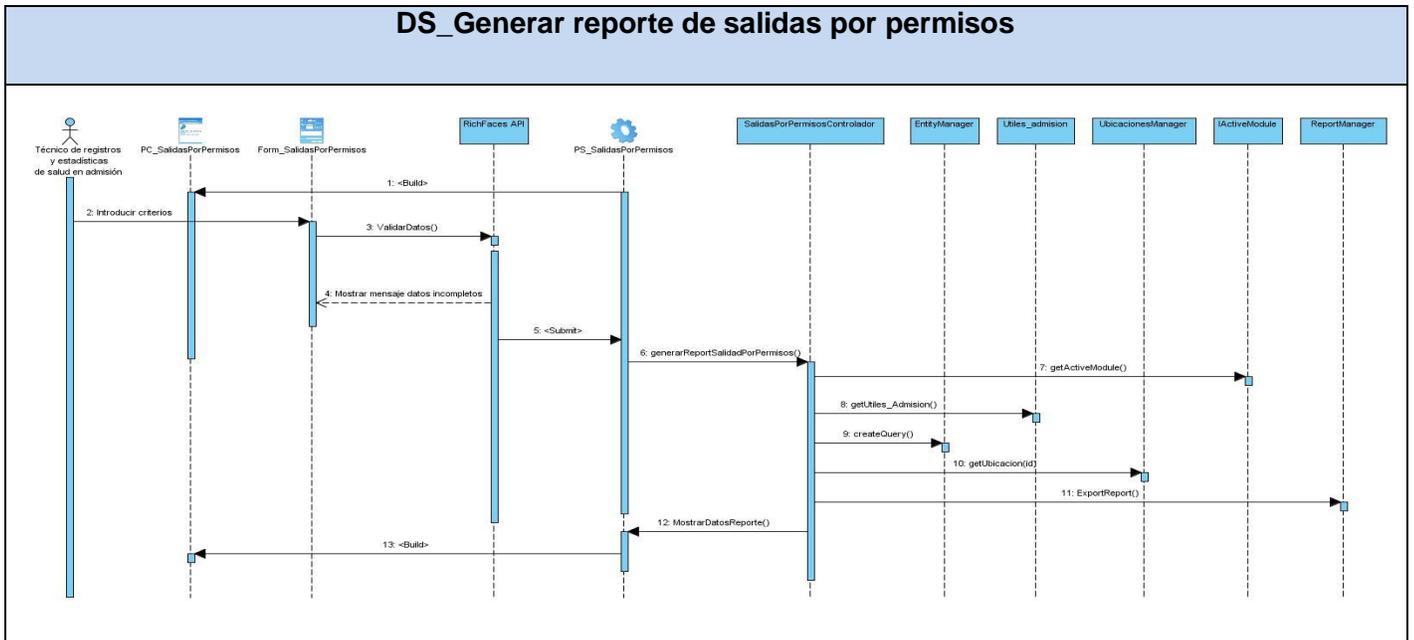


Figura 3.8. DS_Generar reporte de salidas por permisos

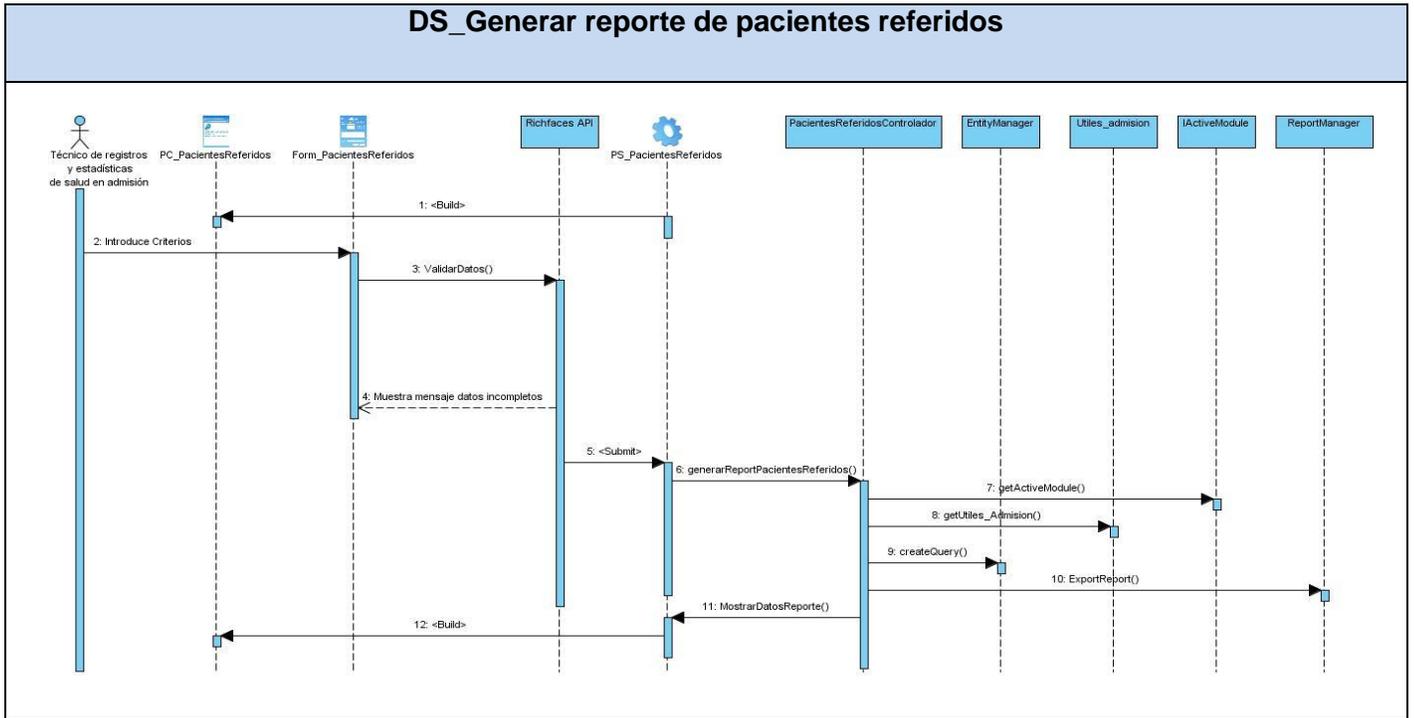


Figura 3.9. DS_Generar reporte de pacientes referidos

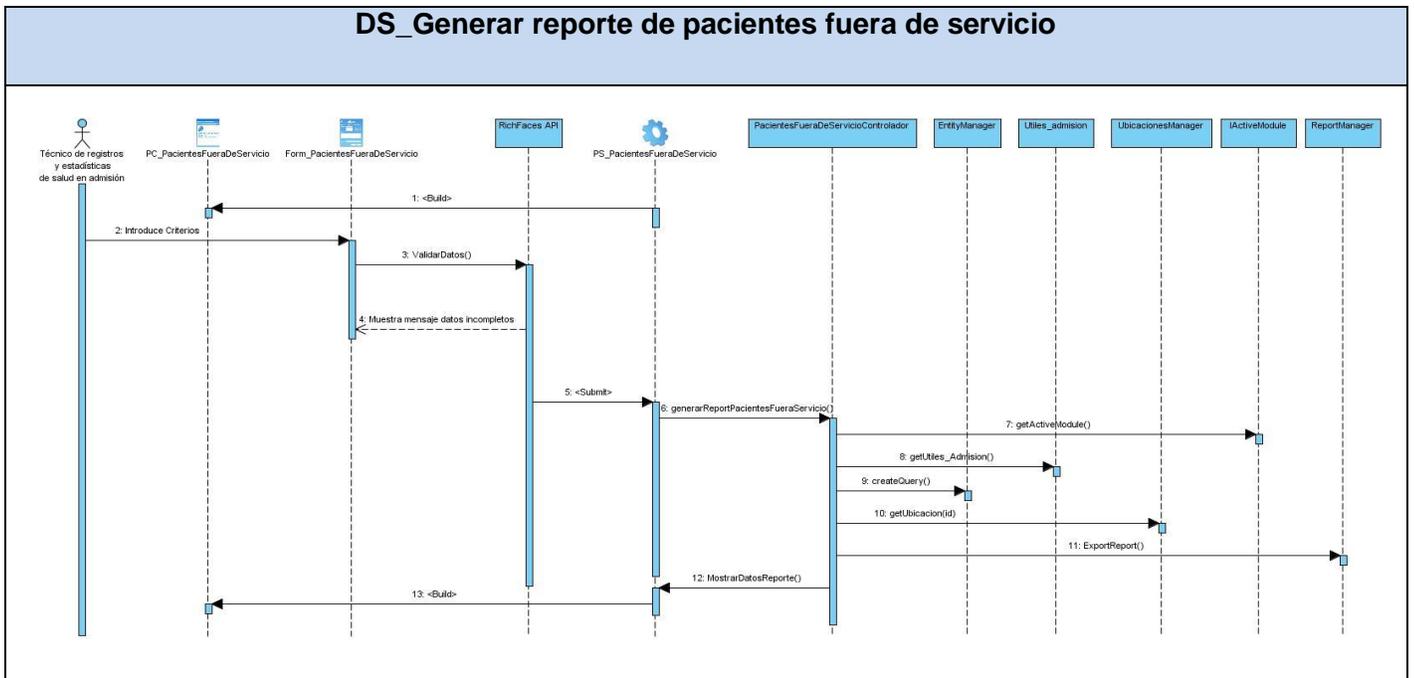


Figura 3.10. DS_Generar reporte de pacientes fuera de servicio

DS_Generar reporte de orden de admisión y transferencia sin efectuar

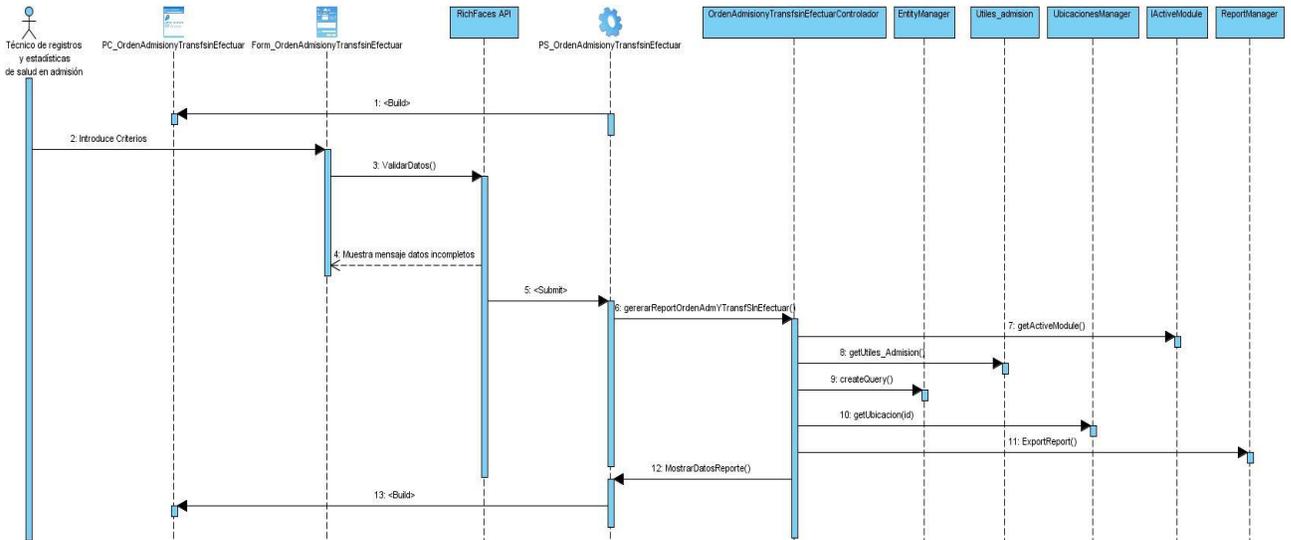


Figura 3.11. DS_Generar reporte de orden de admisión y transferencia sin efectuar

DS_Generar reporte de auditoría

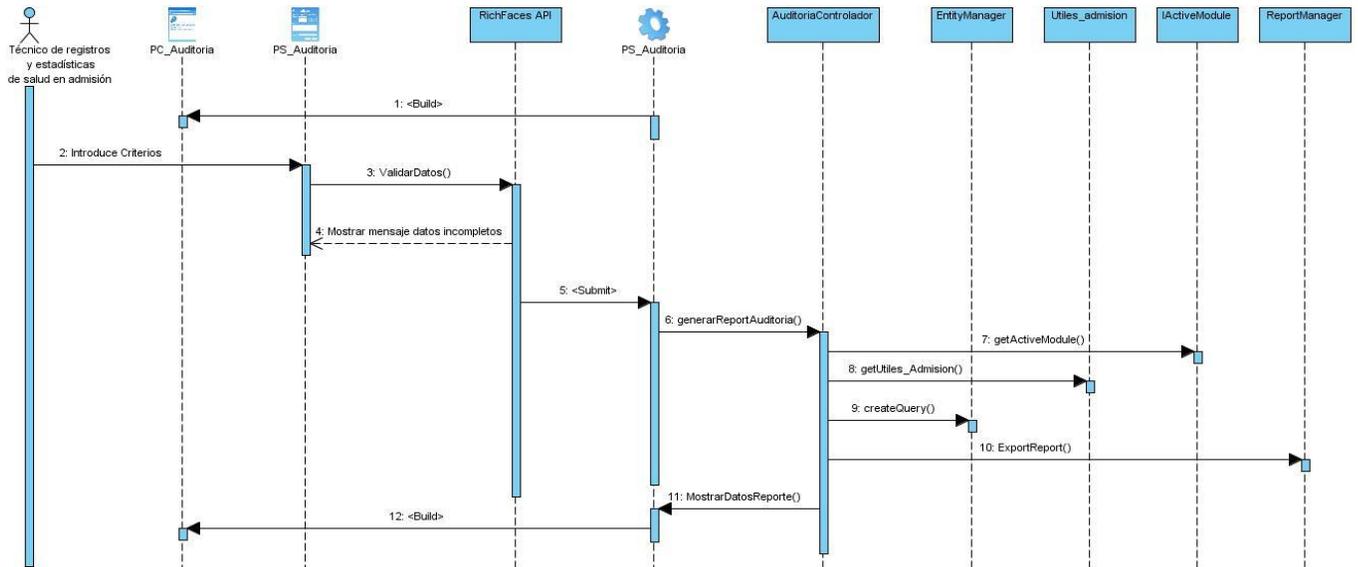


Figura 3.12. DS_Generar reporte de auditoría

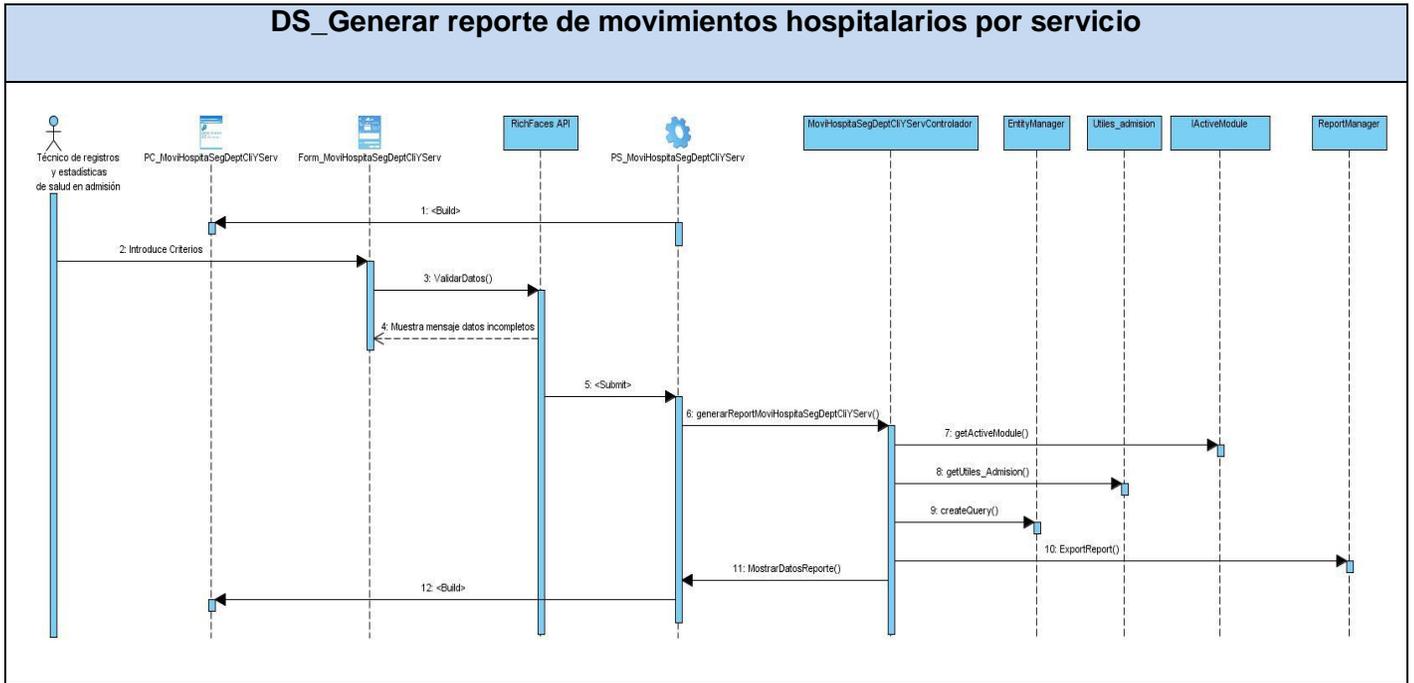


Figura 3.13. DS_Generar reporte de movimientos hospitalarios por servicio

3.2 Descripción de las nuevas clases u operaciones necesarias

Nombre: SalidasPorPermisosControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechaIni	Date
fechaFin	Date
dpto	String
servicio	String
servicios	List<String>
dptos	List<String>
lista	List<SalidaPermisoSource>
flag	Integer
pathToReport	String
parametros	Map
fileformatToExport	String
filesFormatCombo	List<String>
pathExportedReport	String
indicador	String
reportManager	ReportManager
entityManager	EntityManager
activeModule	IActiveModule

utiles_admision	Utiles_admision
ubicacionesManager	UbicacionesManager
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte
Nombre:	listarDepartamentos ()
Descripción:	Lista los departamentos con hospitalización existentes en la entidad actual
Nombre:	listarServicios ()
Descripción:	Lista todos los servicios pertenecientes a un departamento en específico
Nombre:	listDtosVacía ()
Descripción:	Permite conocer si la lista de departamentos (dptos) está vacía
Nombre:	generarReportSalidasPorPermisos ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.1. SalidasPorPermisosControlador

Nombre: PacientesReferidosControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechalni	Date
fechaFin	Date
estado	String
municipio	String
parroquia	String
entidad	String
listEstados	List<String>
listMunicipios	List<String>
listParroquias	List<String>
listEntidades	List<String>
entidadRefiere	String
lista	List<PacientesReferidosSource>
flag	Integer
pathToReport	String
parametros	Map
fileformatToExport	String
filesFormatCombo	List<String>
pathExportedReport	String
titulo	String
reportManager	ReportManager
entityManager	EntityManager

activeModule	IActiveModule
utiles_admision	Utiles_admision
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte
Nombre:	listarEstados ()
Descripción:	Llena una lista con los Estados pertenecientes a la entidad actual
Nombre:	listarMunicipios ()
Descripción:	Llena una lista con los municipios pertenecientes a un estado específico
Nombre:	listarParroquias ()
Descripción:	Llena una lista con las parroquias pertenecientes a un municipio específico
Nombre:	listMunicipioVacía ()
Descripción:	Permite conocer cuándo la lista de municipios debe estar vacía
Nombre:	listParroquiaVacía ()
Descripción:	Permite conocer cuándo la lista de parroquias debe estar vacía
Nombre:	generarReportPacientesReferidos ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.2. PacientesReferidosControlador

Nombre: PacientesFueraServicioControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechaIni	Date
fechaFin	Date
indicador	String
listaIndicadores	List<String>
tipoHC	String
tiposHC	List<String>
dept	String
departamentos	List<String>
servUbi	String
listaServicios	List<String>
parametros	Map
fileformatToExport	String
pathExportedReport	String
pathToReport	String
filesFormatCombo	List<String>
crossTab	List<CrossTabGeneric<Integer, Integer, String>>

flag	boolean
banderaCross	Integer
id	int
reportManager	ReportManager
entityManager	EntityManager
activeModule	IActiveModule
ubicacionesManager	UbicacionesManager
utiles_admision	Utiles_admision
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte y otros atributos de la clase
Nombre:	ListarIndicadores ()
Descripción:	Llena la lista listaIndicadores
Nombre:	Cambiar ()
Descripción:	Llena una lista con los servicios pertenecientes a un departamento
Nombre:	ComprobarTipoHC (String codigop)
Descripción:	Permite comprobar un tipo de historia clínica dado un código
Nombre:	GenerarReporte ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.3. PacientesFueraServicioControlador

Nombre: OrdenAdmisionTransfSinEfectuarControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechaIni	Date
fechaFin	Date
indicador	String
tipoMovim	String
datosPacienteOrigen	List<String>
datosPacienteDestino	List<String>
mapa	Map
parametros	Map
crossTab	List<CrossTabGeneric<Integer, Integer, String>>
flag	Integer
titulo	String
pathToReport	String
fileformatToExport	String
filesFormatCombo	List<String>
pathExportedReport	String

listIndicadores	List<String>
reportManager	ReportManager
entityManager	EntityManager
activeModule	IActiveModule
utiles_admision	Utiles_admision
ubicacionesManager	UbicacionesManager
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte
Nombre:	Listar ()
Descripción:	Llena la lista datosPacienteOrigen con otros datos del paciente
Nombre:	habilitarDeshabilitar ()
Descripción:	Permite conocer si el atributo tipoMovim está vacío con el objetivo de habilitarlo o deshabilitarlo
Nombre:	generarReportOrdenAdmYTransf ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.4. OrdenAdmisionTransfSinEfectuarControlador

Nombre: AuditoriaControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechaIni	Date
fechaFin	Date
flag	boolean
parametros	Map
pathExportedReport	String
fileformatToExport	String
pathToReport	String
filesFormatCombo	List<String>
tipoMov	String
tiposMovimientos	List<String>
tipoHC	String
tiposHC	List<String>
departamento	String
departamentoEmisor	String
departamentoReceptor	String
departamentos	List<String>
servicio	String
servicioEmisor	String
servicioReceptor	String

servicios	List<String>
otrosDatos	List<String>
destinoOtrosDatos	List<String>
lista	List<CrossTabGeneric<Integer, Integer, String>>
transferencia	boolean
egreso	boolean
admission	boolean
banderaServi	boolean
banderaReceptor	boolean
banderaEmisor	boolean
banderaDepartamento	boolean
chutle	boolean
mesesInt	Map
id	int
entityManager	EntityManager
utiles_admission	Utiles_admission
activeModule	IActiveModule
reportManager	ReportManager
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte y otros atributos de la clase
Nombre:	ListarOtrosDatos ()
Descripción:	Permite llenar la lista otrosDatos con otros datos del paciente
Nombre:	ListarTipoMovi ()
Descripción:	Permite llenar la lista tiposMovimientos con los tipos de movimientos hospitalarios existentes
Nombre:	ListarTiposHC ()
Descripción:	Permite llenar la lista tiposHC con los tipos de historias clínicas existentes
Nombre:	ListarDepartamentos ()
Descripción:	Permite llenar la lista departamentos con los departamentos que tienen hospitalización existentes en la entidad actual
Nombre:	ComprobarTipo_HC (String codigop)
Descripción:	Permite comprobar un tipo de historia clínica dado un código
Nombre:	Indocumentado ()
Descripción:	Permite conocer si el tipo de historia clínica seleccionada es Indocumentado
Nombre:	Cambiar ()
Descripción:	Permite cambiar el valor de algunas variables de la clase
Nombre:	CambiarServi (String dpto)
Descripción:	Llena la lista servicios con los servicios pertenecientes a un departamento dado

Nombre:	CambiarServEmisor ()
Descripción:	Llena una lista con los servicios emisores pertenecientes a un departamento emisor
Nombre:	CambiarServReceptor ()
Descripción:	Llena una lista con los servicios receptores pertenecientes a un departamento receptor
Nombre:	Buscar (String parametro)
Descripción:	Permite conocer si una lista contiene un parámetro dado
Nombre:	CalcularDias (Date fechaini, Date fechafin)
Descripción:	Permite calcular la cantidad de días transcurridos entre dos fechas dadas
Nombre:	GenerarReporte ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.5. AuditoriaControlador

Nombre: MovimientosHospitalariosPorServicioControlador	
Tipo de clase: Controladora	
Atributo	Tipo
fechalni	Date
fechaFin	Date
parametros	Map
mesesInt	Map
pathExportedReport	String
fileformatToExport	String
pathToReport	String
filesFormatCombo	List<String>
lista	List<CrossTabGeneric<String, Integer, Long>>
tipoHC	String
tiposHC	List<String>
departamento	String
departamentos	List<String>
sevicio	String
servicios	List<String>
admission	boolean
transferencia	boolean
egreso	boolean
render	boolean
flag	boolean
id	int
entityManager	EntityManager

utiles_admision	Utiles_admision
activeModule	IActiveModule
reportManager	ReportManager
Para cada responsabilidad:	
Nombre:	Inicializar ()
Descripción:	Inicializa la lista filesFormatCombo con los formatos de archivos a los cuales se puede exportar un reporte y otros atributos de la clase
Nombre:	ListarTiposHC ()
Descripción:	Permite llenar la lista tiposHC con los tipos de historia clínica existentes
Nombre:	ListarDepartamentos ()
Descripción:	Permite llenar la lista departamentos con los departamentos con hospitalización existentes en la entidad actual
Nombre:	Cambiar ()
Descripción:	Permite llenar la lista servicios con los servicios pertenecientes al departamento seleccionado
Nombre:	ContarAdmisiones (String nombre_servicio)
Descripción:	Permite conocer la cantidad de admisiones pertenecientes a un servicio dado
Nombre:	ContarEgresos (String nombre_servicio, boolean parametro)
Descripción:	Permite conocer la cantidad de egresos vivos y fallecidos pertenecientes a un servicio dado
Nombre:	ContarTransferencias (String nombre_servicio, boolean parametro)
Descripción:	Permite conocer la cantidad de transferencias emitidas y recibidas pertenecientes a un servicio dado
Nombre:	GenerarReporte ()
Descripción:	Permite generar el reporte de información
Nombre:	exportReportToFileFormat ()
Descripción:	Permite exportar el reporte generado al formato deseado

Tabla 3.6. MovimientosHospitalariosPorServicioControlador

3.3 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Este permite detallar los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí. En general un modelo de datos es la estructura o representación física de las tablas en una base de datos.

Teniendo en cuenta los datos que se recogen en el Módulo Admisión del Sistema de Información Hospitalaria alas HIS y que son imprescindibles para la generación de reportes estadísticos, se representa el siguiente modelo de datos a utilizar:

Modelo de datos

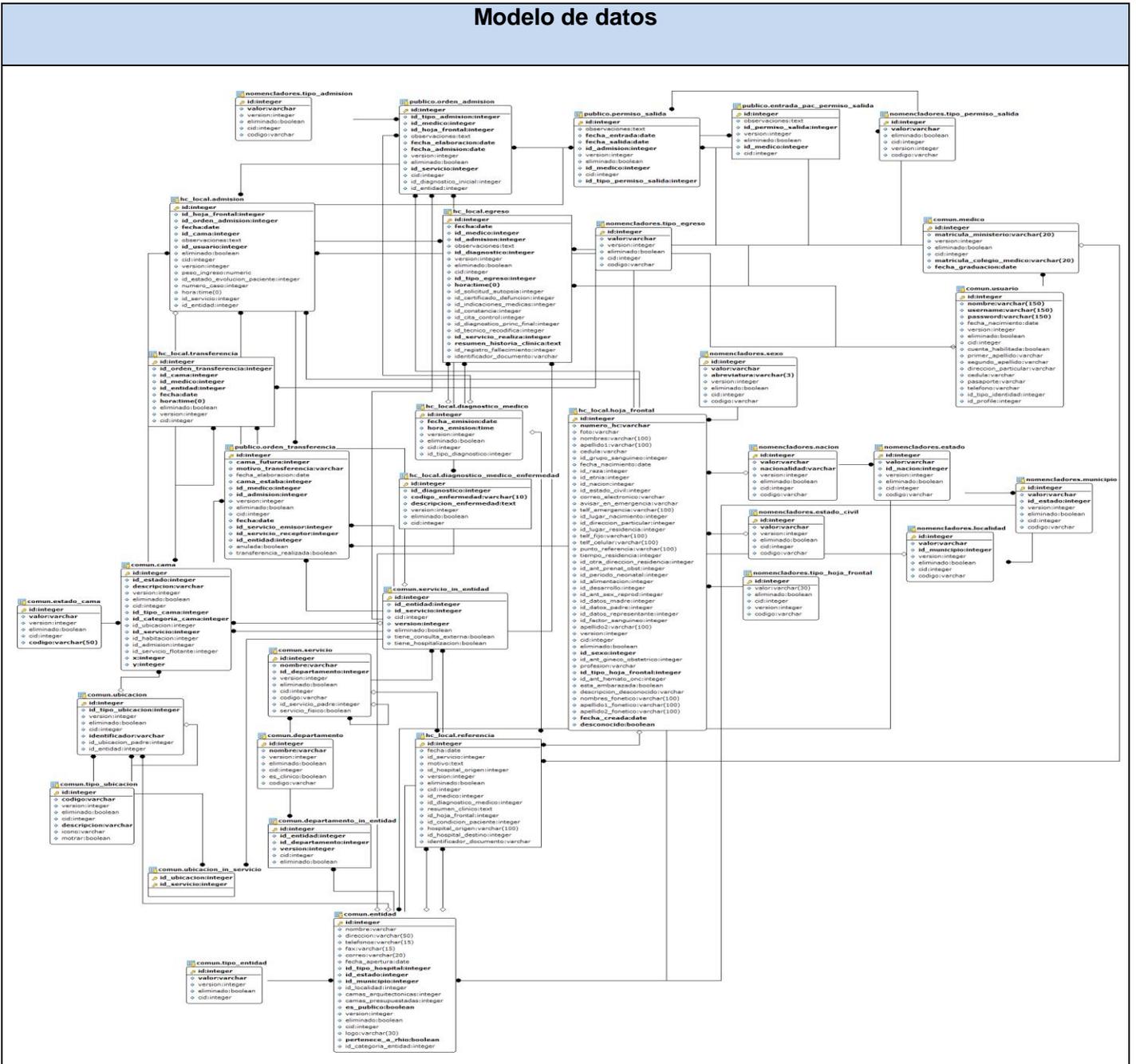


Figura 3.14. Modelo de datos

3.4 Valoración de las Técnicas de validación

La validación de los datos es una parte importante de cualquier aplicación, ya que asegura que los datos en un modelo estén conformes a las reglas de negocio de la aplicación. La validación de los datos antes de enviar actualizaciones a la base de datos, constituye una buena práctica que reduce los errores y la cantidad potencial de acciones de ida y vuelta entre una aplicación y la base de datos.

Hibernate Validator define una única vez las validaciones en los objetos de negocio, evitando encontrar validaciones repetidas en diferentes formularios que se relacionan con los mismos objetos del negocio. JBoss Seam hace uso de esta capacidad y la integra con JSF y se invocan dichas validaciones donde sea necesario. Hibernate Validator está plenamente internacionalizado y las características fundamentales que presenta son las siguientes:

- Se definen las validaciones fácilmente utilizando anotaciones; siendo posible utilizarlas tanto en los atributos como en los getters.
- Trae un conjunto predefinido de validaciones típicas, que se pueden extender con validaciones propias del desarrollador.
- El sistema soporta internacionalización, trae mensajes de error traducidos a diez idiomas. Estos mensajes pueden ser cambiados fácilmente, simplemente escribiendo un fichero propio de propiedades y sobrescribiendo los mensajes que interesen.
- Se integra directamente con Hibernate, y en general con cualquier herramienta de mapeo objeto-relacional, de forma que antes de hacer una inserción o actualización en la base de datos se validarán los objetos.
- Si se utiliza Hibernate, las validaciones indicadas se tendrán en cuenta a la hora de generar el DDL (los scripts de creación de la base de datos).

3.4.1 Descripción de las tablas

En todas las tablas de la base de datos, existen los siguientes atributos:

- **id:** identifica unívocamente cada tupla de la tabla.
- **eliminado:** la eliminación de los datos es lógica, este atributo especifica si un dato fue borrado o no, lo que permite mantener siempre disponible los datos clínicos del paciente.
- **cid:** almacena el identificador de la última acción realizada por un usuario para mantener las trazas del sistema.

- **version:** es utilizado por Hibernate para controlar las versiones de los datos.

Nombre: hc_local.admisión		
Descripción: Almacena todos los ingresos que se realizan en la institución		
Atributo	Tipo	Descripción
id_hoja_frontal	Integer	Identificador de la hoja frontal del paciente
id_orden_admision	Integer	Identificador de la orden de admisión
fecha	Date	Fecha de ingreso del paciente
id_cama	Integer	Identificador de la cama a la que pertenece el paciente
observaciones	Text	Observaciones realizadas cuando ingresó el paciente
id_usuario	Integer	Identificador del usuario que admitió el paciente
peso_ingreso	Numeric	Peso del paciente al ingresar
id_estado_evolucion_paciente	Integer	Identificador del estado evolución del paciente
numero_caso	Integer	Número de caso atendido al que pertenece el paciente
hora	Time(0)	Hora de ingreso
id_servicio	Integer	Identificador del servicio a donde va a ser ingresado el paciente
id_entidad	Integer	Identificador de la entidad

Tabla 3.7. hc_local.admisión

Nombre: hc_local.transferencia		
Descripción: Almacena todas las transferencias que se realizan a los pacientes ingresados en la institución		
Atributo	Tipo	Descripción
id_orden_transferencia	Integer	Identificador de la orden de transferencia
id_cama	Integer	Identificador de la cama a la que pertenece el paciente
id_medico	Integer	Identificador del médico que realiza la transferencia
id_entidad	Integer	Identificador de la entidad
fecha	Date	Fecha en que se realiza la transferencia
hora	Time(0)	Hora en que se realiza la transferencia

Tabla 3.8. hc_local.transferencia

Nombre: hc_local.egreso		
Descripción: Almacena todos los egresos realizados en la institución		
Atributo	Tipo	Descripción
fecha	Date	Fecha en que se realiza el egreso

id_medico	Integer	Identificador del médico que realiza el egreso
id_admision	Integer	Identificador de la admisión a la cual pertenece el egreso
observaciones	Text	Observaciones realizadas al egresar el paciente
id_diagnostico	Integer	Identificador del diagnóstico del paciente
id_tipo_egreso	Integer	Identificador del tipo de egreso
hora	Time(0)	Hora en que se realiza el egreso
id_solicitud_autopsia	Integer	Identificador de la solicitud de autopsia en caso que se realice
id_certificado_defuncion	Integer	Identificador del certificado de defunción en caso que se realice
id_indicaciones_medicas	Integer	Identificador de las indicaciones medicas realizadas al paciente al egresar
id_constancia	Integer	
id_cita_control	Integer	
id_diagnostico_princ_final	Integer	
id_tecnico_recodifica	Integer	
id_servicio_realiza	Integer	Identificador del servicio donde se realiza el egreso
resumen_historia_clinica	Text	Resumen de la historia clínica del paciente
id_registro_fallecimiento	Integer	
identificador_documento	Varchar	

Tabla 3.9. hc_local.egreso

Nombre: hc_local.hoja_frontal		
Descripción: Almacena la información de los paciente		
Atributo	Tipo	Descripción
numero_hc	Varchar	Número de historia clínica
foto	Varchar	Foto del paciente
nombres	Varchar(100)	Nombres del paciente
apellido1	Varchar(100)	Primer apellido del paciente
cedula	Varchar	Cédula del paciente
id_grupo_sanguineo	Integer	Identificador del grupo sanguíneo del paciente
fecha_nacimiento	Date	Fecha de nacimiento del paciente
id_raza	Integer	Identificador de la raza del paciente
id_etnia	Integer	Identificador de la etnia del paciente
id_nacion	Integer	Identificador de la nación a la que pertenece el paciente
id_estado_civil	Integer	Identificador del estado civil del paciente
correo_electronico	Varchar	Correo electrónico del paciente
avisar_en_emergencia	Varchar	Lugar o persona a la que se debe avisar en

		caso de emergencia
telf_emergencia	Varchar(100)	Teléfono al que se debe llamar en caso de emergencia
id_lugar_nacimiento	Integer	Identificador del lugar de nacimiento del paciente
id_direccion_particular	Integer	Identificador de la dirección particular del paciente
id_lugar_residencia	Integer	Identificador del lugar de residencia del paciente
telf_fijo	Varchar(100)	Número de teléfono fijo del paciente
telf_celular	Varchar(100)	Número del teléfono celular del paciente
punto_referencia	Varchar(100)	
tiempo_residencia	Integer	Tiempo en que el paciente reside
id_otra_direccion_residencia	Integer	Identificador de otra dirección de residencia del paciente
id_ant_prenat_obst	Integer	Identificador de antecedentes prenatales obstetricia
id_periodo_neonatal	Integer	Identificador del período neonatal
id_alimentacion	Integer	Identificador de la alimentación que requiere el paciente
id_desarrollo	Integer	
id_ant_sex_reprod	Integer	
id_datos_madre	Integer	Identificador de los datos de la madre del paciente
id_datos_padre	Integer	Identificador de los datos del padre del paciente
id_datos_representante	Integer	Identificador de los datos del representante del paciente
id_factor_sanguineo	Integer	Identificador del factor sanguíneo del paciente
apellido2	Varchar(100)	Segundo apellido del paciente
idsexo	Integer	Identificador del sexo del paciente
id_ant_gineco_obstetrico	Integer	Identificador de antecedentes gineco obstétrico
profesion	Varchar	Profesión del paciente
id_tipo_hoja_frontal	Integer	Identificador del tipo de hoja frontal
id_ant_hemato_onc	Integer	
esta_embarazada	Boolean	Paciente embarazada o no
descripcion_desconocido	Varchar	Descripción del paciente en caso de que sean desconocido sus datos personales
nombres_fonetico	Varchar(100)	Nombre fonético del paciente
apellido1_fonetico	Varchar(100)	Primer apellido fonético del paciente
apellido2_fonetico	Varchar(100)	Segundo apellido fonético del paciente
fecha_creada	Date	Fecha de creación de la hoja frontal

desconocido	Boolean	Paciente desconocido o no
-------------	---------	---------------------------

Tabla 3.10. hc_local.hoja_frontal

Nombre: publico.orden_admision		
Descripción: Almacena los datos de todas las ordenes de admisión realizadas en la institución		
Atributo	Tipo	Descripción
id_tipo_admision	Integer	Identificador del tipo de admisión
id_medico	Integer	Identificador del médico que realiza la orden de admisión
id_hoja_fontal	Integer	Identificador de la hoja frontal del paciente al cual pertenece la orden de admisión
observaciones	Text	Observaciones realizadas al elaborar la orden de admisión
fecha_elaboracion	Date	Fecha de elaboración de la orden de admisión
fecha_admision	Date	Fecha en la que el paciente debe ser admitido
id_servicio	Integer	Identificador del servicio en el cual debe ser admitido el paciente
id_diagnostico_inicial	Integer	Identificador del diagnostico inicial del paciente
id_entidad	Integer	Identificador de la entidad a la que pertenece la orden de admisión

Tabla 3.11. publico.orden_admision

Nombre: publico.orden_transferencia		
Descripción: Almacena los datos de todas las ordenes de transferencias realizadas en la institución		
Atributo	Tipo	Descripción
cama_futura	Integer	Cama hacia la cual va a ser transferido el paciente
motivo_transferencia	Varchar	Motivos por los cuales se transfiere el paciente
fecha_elaboracion	Date	Fecha de elaboración de la orden de transferencia
cama_estaba	Integer	Cama donde estaba el paciente antes de ser transferido
id_medico	Integer	Identificador del médico que realiza la orden de transferencia
id_admision	Integer	Identificador de la admisión para la cual se realiza la orden de transferencia
fecha	Date	Fecha en la que debe efectuarse la

		transferencia
id_servicio_emisor	Integer	Identificador del servicio donde está ingresado el paciente antes de ser transferido
id_servicio_receptor	Integer	Identificador del servicio donde está ingresado el paciente luego de ser transferido
id_entidad	Integer	Identificador de la entidad a la que pertenece la orden de transferencia
anulada	Boolean	Orden de transferencia anulada o no
transferencia_realizada	Boolean	Transferencia realizada o no

Tabla 3.12. publico.orden_transferencia

Nombre: nomencladores.tipo_admision		
Descripción: Contiene los tipos de admisiones que se pueden realizar		
Atributo	Tipo	Descripción
valor	Varchar	Tipo de admisión
codigo	Varchar	Código que identifica el tipo de admisión

Tabla 3.13. nomencladores.tipo_admision

Nombre: nomencladores.tipo_egreso		
Descripción: Contiene los tipos de egresos que se pueden realizar		
Atributo	Tipo	Descripción
valor	Varchar	Tipo de egreso
codigo	Varchar	Código que identifica el tipo de egreso

Tabla 3.14. nomencladores.tipo_egreso

Nombre: comun.departamento		
Descripción: Contiene los departamentos de la institución		
Atributo	Tipo	Descripción
nombre	Varchar	Nombre del departamento
es_clinico	Boolean	Define si un departamento es clínico o no
codigo	Varchar	Código que identifica el departamento

Tabla 3.15. comun.departamento

Nombre: comun.servicio		
Descripción: Contiene los servicios de la institución		
Atributo	Tipo	Descripción
nombre	Varchar	Nombre del servicio
id_departamento	Integer	Identificador del departamento al cual pertenece el servicio
codigo	Varchar	Código que identifica el servicio
id_servicio_padre	Integer	Identificador del servicio padre de un servicio

servicio_fisico	Boolean	Servicio físico o no
-----------------	---------	----------------------

Tabla 3.16. comun.servicio

Nombre: publico.permiso_salida		
Descripción: Contiene todas las salidas por permisos que se han efectuado en la institución		
Atributo	Tipo	Descripción
observaciones	Text	Observaciones realizadas al paciente
fecha_entrada	Date	Fecha de entrada a la institución
fecha_salida	Date	Fecha de salida de la institución
id_admision	Integer	Identificador de la admisión a la cual pertenece la salida
id_medico	Integer	Identificador del médico que autoriza la salida de la institución
id_tipo_permiso_salida	Integer	Identificador del tipo de permiso

Tabla 3.17. publico.permiso_salida

Nombre: publico.entrada_pac_permiso_salida		
Descripción: Contiene todas las entradas pertenecientes a salidas por permisos que se han efectuado en la institución		
Atributo	Tipo	Descripción
observaciones	Text	Observaciones de la entrada del paciente
id_permiso_salida	Integer	Identificador del permiso de salida al cual pertenece la entrada
id_medico	Integer	Identificador del médico que registra la entrada

Tabla 3.18. publico.entrada_pac_permiso_salida

3.5 Vista de Implementación

La vista de implementación describe la descomposición del software en capas y subsistemas de implementación, además muestra la ejecución de los elementos del diseño mediante componentes, así como sus interfaces y dependencias entre componentes. Uno de los diagramas que forma parte de la vista de implementación es el de componentes, el cual permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Específicamente en UML los diagramas de componentes se utilizan para modelar la vista estática de un sistema.

A continuación se muestra el diagrama de componentes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS:

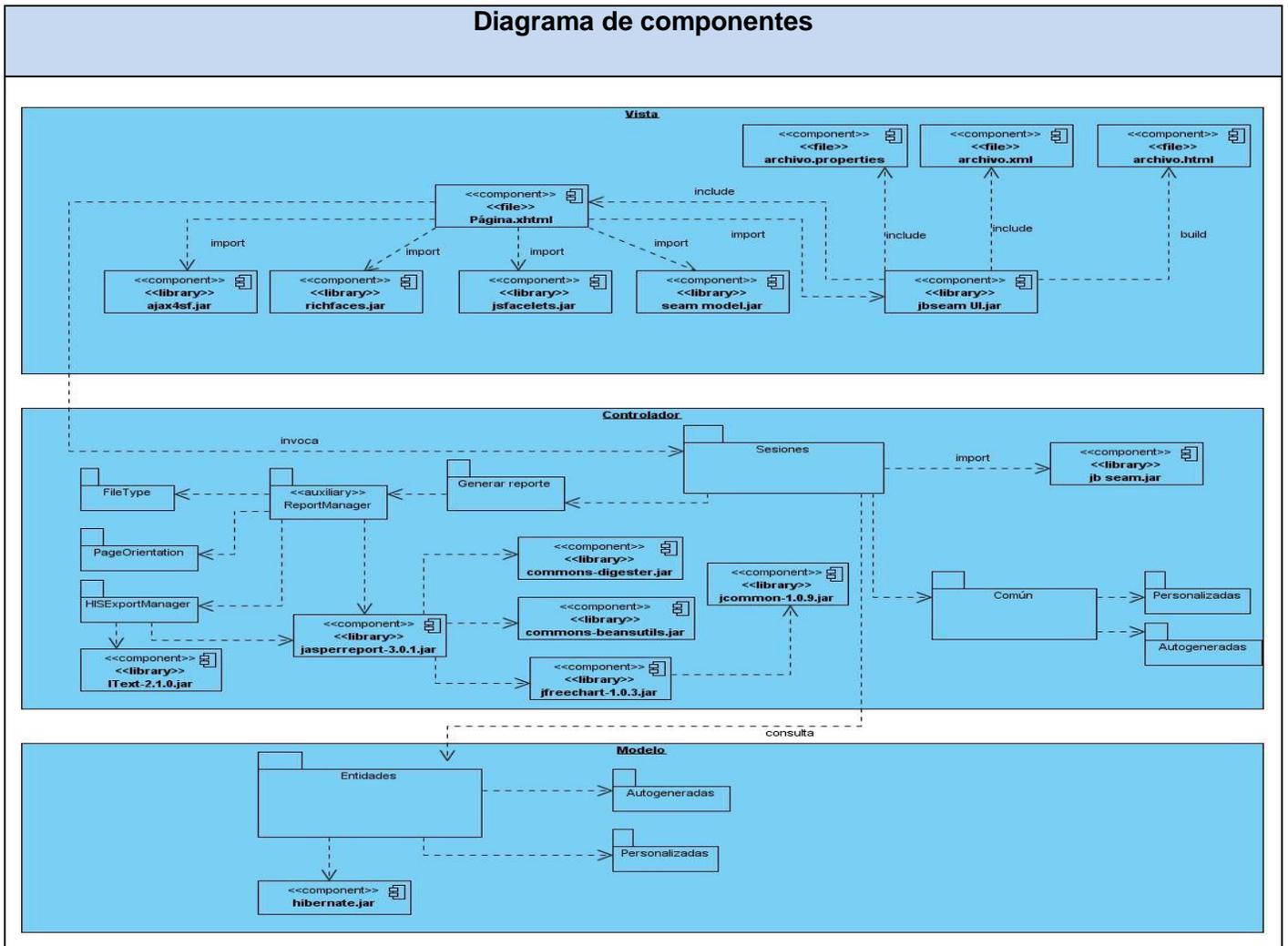


Figura 3.15. Diagrama de componentes

En este capítulo se realizó una descripción del diseño donde se explicaron los principales patrones que se utilizaron, se mostraron los diagramas de clases del diseño y los de secuencia de los casos de uso más significativos. También se describieron las clases u operaciones necesarias en la implementación de la solución, se mostró el modelo de datos que contempla la estructura de la base de datos y se expusieron algunas de las tablas más importantes. Se abordó sobre las técnicas de validación que se utilizan en el sistema, así como la vista de implementación del Módulo Admisión del Sistema de Información Hospitalaria alas HIS. Todo esto de gran importancia para la implementación del sistema.

Capítulo 4. Modelo de prueba

Este capítulo estará dedicado a abordar aspectos significativos acerca de las pruebas de software que se realizan sobre un sistema, con el objetivo de verificar y revelar la calidad del mismo. Se explicarán las características fundamentales del método de prueba utilizado y se mostrarán las descripciones de algunos casos de prueba de los casos de uso seleccionados.

Las pruebas de software constituyen una fase importante en el desarrollo del sistema ya que representan una revisión final de las especificaciones del diseño y de la codificación, además, son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Cuando se está ejecutando una prueba, los componentes del sistema son ejecutados bajo ciertas condiciones o requerimientos especificados; los resultados son observados y registrados, y se realiza una evaluación de algún aspecto del sistema o componente. Existen dos métodos de prueba a través de los cuales puede ser probado cualquier sistema, estos son: las pruebas de caja blanca y las pruebas de caja negra.

4.1 Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. Es decir, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten todos los requisitos funcionales de un programa, sin tener en cuenta el comportamiento interno y la estructura del mismo. Todo esto con el objetivo de demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Cuando se realizan estas pruebas se intenta encontrar errores de distintas categorías como son: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. (38)

Para llevar a cabo las pruebas de caja negra se utilizan varias técnicas de las cuales se derivan un conjunto de casos de prueba que no son más que un grupo de condiciones o variables bajo las cuales se determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Entre las técnicas más utilizadas se encuentran:

- **Particiones de equivalencia**

La partición de equivalencia divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba

que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia que representan un conjunto de estados válidos o no válidos para condiciones de entrada, las cuales típicamente, son valores numéricos específicos, rango de valores, conjunto de valores relacionados o condiciones lógicas; de tal modo que se pueda asumir que una prueba realizada con un valor representativo de cada clase, es equivalente a una prueba realizada con cualquier otro valor de dicha clase. Esto quiere decir que si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error. (39)

- **Análisis de valores límite**

Esta técnica permite elegir los casos de prueba que ejerciten los valores límite, que son aquellos que se encuentran en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por lo tanto, el análisis de valores límite complementa la técnica de partición de equivalencia de manera que en lugar de seleccionar cualquier caso de prueba de las clases válidas e inválidas, se eligen los casos de prueba en los extremos. Y en lugar de centrarse sólo en el dominio de entrada, los casos de prueba se diseñan también considerando el dominio de salida. (39) Por ejemplo, si una condición de entrada o salida exige un rango entre B y R, se realizan los casos de prueba para los valores B y R, y además a los valores que están por encima de ellos, es decir A y S.

- **Grafos causa-efecto**

El uso de grafos de causa-efecto es una técnica de casos de prueba que proporciona una concisa representación de las condiciones lógicas y sus correspondientes acciones. La técnica lista las causas que son las condiciones de entrada y los efectos que constituyen las acciones, asignando un identificador a cada uno de ellos. Se desarrolla un grafo causa-efecto y se obtienen los casos de prueba que verifican que todos los objetos tienen entre ellos las relaciones esperadas. Esta técnica tiene un efecto secundario beneficioso, ya que, precisa estados incompletos y ambigüedades en la especificación.

Todas estas técnicas tienen gran utilidad para determinar errores presentes en las funcionalidades desarrolladas. La técnica partición de equivalencia, es la que se utilizará para obtener los casos de pruebas necesarios, que permitirán probar los casos de uso desarrollados.

4.2 Descripción de los casos de prueba

Escenarios del reporte de salidas por permisos	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de salidas por permisos	Generar el reporte de salidas por permisos satisfactoriamente.	<p>Se selecciona la opción Generar reporte de salidas por permisos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Selecciona la opción Salir.</p> <p>Regresa a la vista anterior.</p>
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	<p>Se selecciona la opción Generar reporte de salidas por permisos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra un indicador sobre los campos incompletos.</p>
EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	<p>Se selecciona la opción Generar reporte de salidas por permisos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."</p>
EC 4: Cancelar operación	Cancelar la opción de generar reporte de salidas por permisos.	<p>Se selecciona la opción Generar reporte de salidas por permisos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Cancelar.</p> <p>Se regresa a la vista anterior.</p>
EC 5: Exportar	Permite exportar el reporte generado.	<p>Se selecciona la opción Generar reporte de salidas por permisos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Se selecciona la opción exportar. Ver DCP Exportar.</p>

Tabla 4.1. CP_Generar reporte de salidas por permisos

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5
Escenario	Generar reporte de salidas por permisos	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar
Variable 1 (Desde)	V	I	I	V	V
Variable 2 (Hasta)	V	I	I	V	V
Variable 3 (Indicadores)	V	I	I	V	V
Variable 4 (Departamento)	V	V	I	V	V
Variable 5 (Servicio ubicación)	V	V	I	V	V
Botón 1 (Generar)	NA	NA	NA		NA
Botón 2 (Cancelar)				NA	
Botón 3 (Exportar)					NA
Botón 4 (Salir)	NA				
Respuesta del Sistema	Muestra: <ul style="list-style-type: none"> • Paciente • Cédula • Sexo 	Muestra un indicador sobre los campos	Muestra el mensaje de información "No existe información que cumpla con los parámetros"	Regresa a la vista anterior	Muestra la interfaz para

	<ul style="list-style-type: none"> • Fecha salida • Fecha entrada • Tipo permiso • Departamento • Servicio ubicación • Ubicación • Médico autoriza 	incompletos.	seleccionados.”		exportar.
Resultado de la Prueba					

Tabla 4.1.1. SC_Generar reporte de salidas por permisos

Escenarios del reporte de pacientes referidos	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de pacientes referidos	Generar el reporte de pacientes referidos satisfactoriamente.	<p>Se selecciona la opción Generar reporte de pacientes referidos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Selecciona la opción Salir.</p> <p>Regresa a la vista anterior.</p>
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	<p>Se selecciona la opción Generar reporte de pacientes referidos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra un indicador sobre los campos incompletos.</p>
EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	<p>Se selecciona la opción Generar reporte de pacientes referidos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra el mensaje de información “No existe información que cumpla con los parámetros seleccionados.”</p>
EC 4: Cancelar operación	Cancelar la opción de generar reporte de pacientes referidos.	Se selecciona la opción Generar reporte de pacientes referidos.

		<p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Cancelar.</p> <p>Se regresa a la vista anterior.</p>
EC 5: Exportar	Permite exportar el reporte generado.	<p>Se selecciona la opción Generar reporte de pacientes referidos.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Se selecciona la opción exportar. Ver DCP Exportar.</p>

Tabla 4.2. CP_Generar reporte de pacientes referidos

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5
Escenario	Generar reporte de pacientes referidos	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar
Variable 6 (Desde)	V	I	I	V	V
Variable 7 (Hasta)	V	I	I	V	V
Variable 8 (Indicadores)	V	I	I	V	V
Variable 9 (Estado)	V	V	I	V	V
Variable 10 (Municipio)	V	V	I	V	V
Variable 11 (Parroquia)	V	V	I	V	V

Variable 12 (Entidad)	V	V	I	V	V
Botón 5 (Generar)	NA	NA	NA		NA
Botón 6 (Cancelar)				NA	
Botón 7 (Exportar)					NA
Botón 8 (Salir)	NA				
Respuesta del Sistema	Muestra: <ul style="list-style-type: none"> • Paciente • Cédula • Sexo • Fecha nacimiento • Fecha referencia • Diagnóstico • Hospital • Tipo hospital • Estado • Municipio • Parroquia • Médico que autoriza 	Muestra un indicador sobre los campos incompletos.	Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."	Regresa a la vista anterior	Muestra la interfaz para exporta.
Resultado de la Prueba					

Tabla 4.2.1. SC_Generar reporte de pacientes referidos

Escenarios del reporte de pacientes fuera de servicio	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de pacientes fuera de servicio	Generar el reporte de pacientes fuera de servicio satisfactoriamente.	<p>Se selecciona la opción Generar reporte de pacientes fuera de servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Selecciona la opción Salir.</p> <p>Regresa a la vista anterior.</p>
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	<p>Se selecciona la opción Generar reporte de pacientes fuera de servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra un indicador sobre los campos incompletos.</p>
EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	<p>Se selecciona la opción Generar reporte de pacientes fuera de servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."</p>
EC 4: Cancelar operación	Cancelar la opción de generar reporte de pacientes fuera de servicio.	<p>Se selecciona la opción Generar reporte de pacientes fuera de servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Cancelar.</p> <p>Se regresa a la vista anterior.</p>
EC 5: Exportar	Permite exportar el reporte generado.	<p>Se selecciona la opción Generar reporte de pacientes fuera de servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Se selecciona la opción exportar. Ver DCP Exportar.</p>

Tabla 4.3. CP_Generar reporte de pacientes fuera de servicio

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5
Escenario	Generar reporte de pacientes fuera de servicio	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar
Variable 13 (Desde)	V	I	I	V	V
Variable 14 (Hasta)	V	I	I	V	V
Variable 15 (Indicadores)	V	I	I	V	V
Variable 16 (Tipo HC.)	V	V	I	V	V
Variable 17 (Departamento)	V	V	I	V	V
Variable 18 (Servicio ubicación)	V	V	I	V	V
Botón 9 (Generar)	NA	NA	NA		NA
Botón 10 (Cancelar)				NA	
Botón 11 (Exportar)					NA
Botón 12 (Salir)	NA				

Respuesta del Sistema	<p>Muestra:</p> <ul style="list-style-type: none"> • Sexo • Fecha admisión • Departamento • Servicio ubicación • Ubicación • Servicio tratante • Médico que admite <p>Si el tipo de historia clínica seleccionada es Indocumentado muestras además:</p> <ul style="list-style-type: none"> • Descripción <p>Si el tipo de historia clínica seleccionada es Pediátrica, Oficial o ninguna de estas muestras además:</p> <ul style="list-style-type: none"> • Paciente • Cédula 	<p>Muestra un indicador sobre los campos incompletos.</p>	<p>Muestra el mensaje de información “No existe información que cumpla con los parámetros seleccionados.”</p>	<p>Regresa a la vista anterior</p>	<p>Muestra la interfaz para exportar.</p>
Resultado de la Prueba					

Tabla 4.3.1. SC_Generar reporte de pacientes fuera de servicio

Escenarios del reporte de orden de admisión y transferencia sin efectuar	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de orden de admisión y transferencia sin efectuar	Generar el reporte de orden de admisión y transferencia sin efectuar satisfactoriamente.	Se selecciona la opción Generar reporte de orden de admisión y transferencia sin efectuar. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte. Selecciona la opción Salir.

		Regresa a la vista anterior.
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	Se selecciona la opción Generar reporte de orden de admisión y transferencia sin efectuar. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra un indicador sobre los campos incompletos.
EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	Se selecciona la opción Generar reporte de orden de admisión y transferencia sin efectuar. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."
EC 4: Cancelar operación	Cancelar la opción de generar reporte de orden de admisión y transferencia sin efectuar.	Se selecciona la opción Generar reporte de orden de admisión y transferencia sin efectuar. Se seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 5: Exportar	Permite exportar el reporte generado.	Se selecciona la opción Generar reporte de orden de admisión y transferencia sin efectuar. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte. Se selecciona la opción exportar. Ver DCP Exportar .

Tabla 4.4. CP_Generar reporte de orden de admisión y transferencia sin efectuar

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5
Escenario	Generar reporte de orden de admisión y transferencia sin efectuar	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar

Variable 19 (Desde)	V	I	I	V	V
Variable 20 (Hasta)	V	I	I	V	V
Variable 21 (Tipo movimiento)	V	I	I	V	V
Variable 22 (Indicadores)	V	V	I	V	V
Variable 23 (Sexo)	V	V	I	V	V
Variable 24 (Estado civil)	V	V	I	V	V
Variable 25 (Nacionalidad)	V	V	I	V	V
Variable 26 (Fecha nacimiento)	V	V	I	V	V
Botón 13 (Generar)	NA	NA	NA		NA
Botón 14 (Cancelar)				NA	
Botón 15 (Exportar)					NA
Botón 16 (Salir)	NA				

Respuesta del Sistema	<p>Muestra:</p> <ul style="list-style-type: none"> • Paciente • Cédula • Médico <p>Si el tipo de movimiento seleccionado es Admisión muestra además:</p> <ul style="list-style-type: none"> • Fecha elaboración de la orden de admisión • Fecha admisión • Tipo admisión • Diagnóstico <p>Si el tipo de movimiento seleccionado es Transferencia muestra además:</p> <ul style="list-style-type: none"> • Fecha elaboración de la orden de transferencia • Fecha transferencia • Motivo transferencia • Servicio emisor • Servicio receptor • Ubicación actual • Ubicación futura <p>Muestra además otros datos que hayan sido seleccionados</p> <ul style="list-style-type: none"> • Sexo • Estado civil • Nacionalidad • Fecha nacimiento 	<p>Muestra un indicador sobre los campos incompletos.</p>	<p>Muestra el mensaje de información “No existe información que cumpla con los parámetros seleccionados.”</p>	<p>Regresa a la vista anterior</p>	<p>Muestra la interfaz para exportar.</p>
------------------------------	---	---	---	------------------------------------	---

Resultado de la Prueba					
-------------------------------	--	--	--	--	--

Tabla 4.4.1. SC_Generar reporte de orden de admisión y transferencia sin efectuar

Escenarios del reporte de auditoría	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de auditoría	Generar el reporte de auditoría satisfactoriamente.	Se selecciona la opción Generar reporte de auditoría. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte. Selecciona la opción Salir. Regresa a la vista anterior.
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	Se selecciona la opción Generar reporte de auditoría. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra un indicador sobre los campos incompletos.
EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	Se selecciona la opción Generar reporte de auditoría. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."
EC 4: Cancelar operación	Cancelar la opción de generar reporte de auditoría .	Se selecciona la opción Generar reporte de auditoría. Se seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 5: Exportar	Permite exportar el reporte generado.	Se selecciona la opción Generar reporte de auditoría. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte.

		Se selecciona la opción exportar. Ver DCP Exportar.
--	--	---

Tabla 4.5. CP_Generar reporte de auditoría

Id del escenario	EC 6	EC 7	EC 8	EC 9	EC 10
Escenario	Generar reporte de auditoría	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar
Variable 27 (Desde)	V	I	I	V	V
Variable 28 (Hasta)	V	I	I	V	V
Variable 29 (Tipo movimiento)	V	I	I	V	V
Variable 30 (Tipo HC.)	V	V	I	V	V
Variable 31 (Departamento)	V	V	I	V	V
Variable 32 (Servicio tratante)	V	V	I	V	V
Variable 33 (Departamento receptor)	V	V	I	V	V
Variable 34 (Servicio receptor)	V	V	I	V	V
Variable 35 (Departamento)	V	V	I	V	V

emisor)					
Variable 36 (Servicio emisor)	V	V	I	V	V
Variable 37 (Sexo)	V	V	I	V	V
Variable 38 (Estado civil)	V	V	I	V	V
Variable 39 (Nacionalidad)	V	V	I	V	V
Variable 40 (Fecha nacimiento)	V	V	I	V	V
Variable 41 (Médico)	V	V	I	V	V
Botón 17 (Generar)	NA	NA	NA		NA
Botón 18 (Cancelar)				NA	
Botón 19 (Exportar)					NA
Botón 20 (Salir)	NA				
Respuesta del Sistema	Si el tipo de movimiento seleccionado es Admisiones muestra: <ul style="list-style-type: none"> Fecha de admisión 	Muestra un indicador sobre los campos incompletos.	Muestra el mensaje de información "No existe información que cumpla con los parámetros	Regresa a la vista anterior	Muestra la interfaz para exportar.

	<ul style="list-style-type: none"> • Servicio tratante • Ubicación • Condición paciente • Impresión diagnóstica <p>Si el tipo de movimiento seleccionado es</p> <p>Transferencias muestra:</p> <ul style="list-style-type: none"> • Fecha transferencia • Tipo transferencia • Servicio emisor • Ubicación del servicio emisor • Servicio receptor • Ubicación del servicio receptor <p>Si el tipo de movimiento seleccionado es</p> <p>Egresos muestra:</p> <ul style="list-style-type: none"> • Fecha admisión • Fecha egreso • Tipo egreso • Servicio ingreso • Servicio egreso • Ubicación • Tiempo permanencia • Tipo estancia <p>Si el tipo de Historia Clínica seleccionada es</p> <p>Indocumentado muestra además:</p> <ul style="list-style-type: none"> • Descripción • Sexo 		seleccionados.”		
--	--	--	-----------------	--	--

	<p>Si el tipo de Historia Clínica seleccionada es Pediátrica, Oficial o ninguna de estas, muestra además:</p> <ul style="list-style-type: none"> • Paciente • Cédula <p>Otros datos que hayan sido seleccionados</p> <ul style="list-style-type: none"> • Sexo • Estado civil • Nacionalidad • Fecha nacimiento • Médico 				
Resultado de la Prueba					

Tabla 4.5.1. SC_Generar reporte de auditoría

Escenarios del reporte de movimientos hospitalarios por servicio	Descripción de la funcionalidad	Flujo Central
EC 1: Generar reporte de movimientos hospitalarios por servicio	Generar el reporte de movimientos hospitalarios por servicio satisfactoriamente.	<p>Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra los datos del reporte.</p> <p>Selecciona la opción Salir.</p> <p>Regresa a la vista anterior.</p>
EC 2: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber datos incompletos muestra un indicador sobre estos datos.	<p>Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio.</p> <p>Se seleccionan los datos correspondientes.</p> <p>Se selecciona la opción Generar.</p> <p>Muestra un indicador sobre los campos incompletos.</p>

EC 3: No existe información que cumpla con los parámetros seleccionados	No existe información que cumpla con los parámetros seleccionados.	Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra el mensaje de información "No existe información que cumpla con los parámetros seleccionados."
EC 4: Cancelar operación	Cancelar la opción de generar reporte de movimientos hospitalarios por servicio.	Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio. Se seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 5: Exportar	Permite exportar el reporte generado.	Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte. Se selecciona la opción exportar. Ver DCP Exportar .
EC 6: Generar gráfico	Permite generar gráfico del reporte.	Se selecciona la opción Generar reporte de movimientos hospitalarios por servicio. Se seleccionan los datos correspondientes. Se selecciona la opción Generar. Muestra los datos del reporte. Se selecciona la opción graficar. Ver DCP Generar gráfico .

Tabla 4.6. CP_Generar reporte de movimientos hospitalarios por servicio

Id del escenario	EC 1	EC 2	EC 3	EC 4	EC 5	EC 6
Escenario	Generar reporte de movimientos hospitalarios por servicio	Existen datos incompletos	No existe información que cumpla con los parámetros seleccionados	Cancelar operación	Exportar	Generar gráfico

Variable 42 (Desde)	V	I	I	V	V	V
Variable 43 (Hasta)	V	I	I	V	V	V
Variable 44 (Tipo de HC.)	V	V	I	V	V	V
Variable 45 (Departamento)	V	V	I	V	V	V
Variable 46 (Servicio)	V	V	I	V	V	V
Variable 47 (Indicadores)	V	V	I	V	V	V
Botón 21 (Generar)	NA	NA	NA		NA	NA
Botón 22 (Cancelar)				NA		
Botón 23 (Exportar)					NA	
Botón 24 (Graficar)						NA
Botón 25 (Salir)	NA					
Respuesta del Sistema	Muestra: <ul style="list-style-type: none"> Servicio(s) Total por indicador Si el indicador	Muestra un indicador sobre los campos	Muestra el mensaje de información "No existe"	Regresa a la vista anterior	Muestra la interfaz para	Muestra la interfaz para generar

	seleccionado es admisiones muestra: <ul style="list-style-type: none"> • Cantidad de admisiones por servicio Si el indicador seleccionado es transferencias muestra: <ul style="list-style-type: none"> • Cantidad de transferencias emitidas por servicio • Cantidad de trasferencias recibidas por servicio Si el indicador seleccionado es egresos muestra: <ul style="list-style-type: none"> • Cantidad de egresos vivos por servicio • Cantidad de egresos fallecidos por servicio 	incompletos.	información que cumpla con los parámetros seleccionados.”		exporta.	gráfico.
Resultado de la Prueba						

Tabla 4.6.1. SC_Generar reporte de movimientos hospitalarios por servicio

En este capítulo se abordó sobre el método utilizado para realizar las pruebas de software, y se expusieron algunas de las técnicas existentes para el diseño de casos de prueba, entre ellas la técnica de partición de equivalencia, que fue la seleccionada para llevar a cabo el desarrollo de los casos de prueba que se utilizarán para examinar las funcionalidades y detectar determinados errores presentes en las mismas. Y por último, se muestra la descripción de algunos casos de prueba diseñados.

Conclusiones

Concluida la presente investigación científica, se determinó que el estudio de los sistemas existentes en el mundo, sirvió para darse cuenta de que existían funcionalidades que no podían ser omitidas en la optimización de los reportes del Módulo Admisión del Sistema alas HIS, tales como:

- Reporte de movimientos hospitalarios.
- Reporte de movimientos hospitalarios por servicio.
- Reporte de uso de camas.
- Reporte de movimientos hospitalarios por paciente.

Por otro lado la aplicación de la arquitectura definida por el Departamento de Sistemas de Gestión Hospitalaria para el desarrollo de sus aplicaciones, facilitó el desarrollo de los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS. Así como la aplicación de las pautas de diseño y codificación, definidas por dicho departamento, posibilitaron obtener interfaces visualmente homogéneas y un código fuente legible y uniforme.

Los reportes del Módulo Admisión del Sistema de Información Hospitalaria alas HIS, que se obtuvieron como resultado de la optimización de la solución existente, permitirán mayor fluidez de las estadísticas en esta área de las instituciones hospitalarias, al posibilitar obtener informes correctamente estructurados y flexibles que permitan tomar decisiones adecuadas en el momento requerido.

Recomendaciones

Sobre la presente investigación, se recomienda:

Agrupar los servicios por el departamento al cual pertenecen en el reporte de camas por servicio y en el reporte de movimientos hospitalarios por servicio, para lograr mayor entendimiento de la información que se muestra.

Adicionar a la clase ReportManager el resto de las funcionalidades presentes en las librerías JFreeChart y JasperReports para la generación y personalización de gráficos y reportes.

Referencias bibliográficas

1. **Salvador Oliván, José Antonio.** *Sistemas de información hospitalarios:el C.M.B.D.* Departamento de Ciencias de la Documentación e Historia de la Ciencia, Universidad de Zaragoza. 1997.
2. **Cerritos, Antonio, Fernández Puerto, Francisco J. y Gatica Lara, Florina.** *Sistema de Información Hospitalaria (Manual de Introducción a la Informática Médica).* Departamento de Servicios de Cómputo, UNAM- Facultad de Medicina. México : s.n., 2003.
3. **Díaz, Josué, Giménez, Iliana, López, Julio, Narváez, Cesar y Rivas, Richard.** *Diseño de un plan de capacitación en el uso de un manejador de base de datos para el personal de recursos humano.* Universidad Nacional Experimental “Simón Rodríguez”. República Bolivariana de Venezuela-Estado Lara : s.n. Trabajo presentado como requisito parcial para optar por el Grado de Licenciado en Administración.
4. *Manual de usuario del Módulo Admisión del Sistema de Información para Gerencia Hospitalaria (SIGHO).*
5. **Ledo Ramírez, Rodney y Ortega Palacios, Alberto.** *Trabajo de Diploma para optar por el título de ingeniero en Ciencias Informáticas: Módulo Admisión del Sistema de Información Hospitalaria alas HIS.* 2009.
6. **Empresa SIVSA .** SIVSA. [En línea] Producto Hosix-V. <http://www.sivsa.com/site/espanha/home.asp>.
7. **Castro, William.** *Manual de Sistema (Versión 1) Sistema Básico de Gestión Hospitalaria (GalenHos).* Noviembre, 2005.
8. *iSOFT una compañía del Grupo IBA Health(x-HIS).* España : s.n.
9. *iSOFT una compañía del Grupo IBA Health-Soluciones extensibles y adaptables a cualquier entorno sanitario.* 2006.
10. **Servicios de Salud de Sinaloa-México.** *SIGHO Sistema de Información para la Gerencia Hospitalaria. Manual de usuario Módulo Admisión.* Disponible en: <http://sigho.ses-gro.gob.mx/wp-content/uploads/MANUALES/SIGHO/Admision.pdf>.

11. **SOFTEL soluciones informáticas.** *Propuesta para la informatización de Centros de Atención Médica.* La Habana : s.n., marzo, 2007. Disponible en: <http://www.softel.cu/doc/CentrosAtencionMedica.pdf>.
12. Pergamino Virtual. [En línea] 2009. [Citado el: 23 de 01 de 2011.] http://www.pergaminovirtual.com.ar/definicion/Navegador_web.html.
13. Diseño+Web. [En línea] [Citado el: 27 de 01 de 2011.] http://boolewebs.com.ar/index.php?option=com_content&task=view&id=20&Itemid=46.
14. Cliente-Servidor. [En línea] <http://www.mitecnologico.com/Main/ClienteServidor>.
15. Programación por capas. [En línea] [Citado el: 27 de 01 de 2011.] <http://www.slideshare.net/CarolRuizP/programacion-por-capas>.
16. **Bascón Pantoja, Ernesto.** *El patrón de diseño Modelo-Vista-Controlador (MVC).* 2004.
17. Tecnología Java. [En línea] [Citado el: 27 de 01 de 2011.] <http://www.slideshare.net/Dryope/tecnologia-java-420996>.
18. **Geary, David y Horstmann, Cay.** *Core JavaServer™ Faces, Second Edition.* 2007. 978-0-13-173886-7.
19. **Red Hat.** *RichFaces Developer Guide.* 2007.
20. XHTML. [En línea] [Citado el: 28 de 01 de 2011.] <http://icesecurity.org/conceptos/xhtmll.htm>.
21. **Red Hat.** *Ajax4jsf Developer Guide.* 2007.
22. **Red Hat.** Seamframework.org. [En línea] [Citado el: 28 de 01 de 2011.] <http://seamframework.org/>.
23. **Juntao Yuan, Michael y Heute, Thomas.** *JBoss Seam Simplicity and Power Beyond Java EE.* 2007. 0-13-134796-9.
24. **Red Hat.** JBoss. [En línea] [Citado el: 28 de 01 de 2011.] <http://www.jboss.com/>.
25. *Introducción a JasperReports e iReport (Primera parte).* 2006.

26. Oracle. [En línea] [Citado el: 28 de 01 de 2011.] <http://www.oracle.com/technetwork/java/index-jsp-140203.html>.
27. **Red Hat**. Hibernate. [En línea] [Citado el: 28 de 01 de 2011.] <http://www.hibernate.org/>.
28. *Hibernate Reference Documentation*.
29. **Bauer, Cristian y King, Gavin**. *Java Persistence with Hibernate*. 2005. 1-932394-88-5.
30. **Gibert Ginestá, Marc y Pérez Mora, Oscar**. *Bases de Datos en Postgres SQL*. Disponible en: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf.
31. EcuRed. [En línea] Eclipse, entorno de desarrollo integrado. http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado.
32. Visual Paradigm. [En línea] [Citado el: 15 de 01 de 2011.] <http://www.visual-paradigm.com/>.
33. pgAdmin. [En línea] [Citado el: 17 de 02 de 2011.] <http://www.pgadmin.org>.
34. iReport. [En línea] [Citado el: 28 de 01 de 2011.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>.
35. *Metodologías de desarrollo de software*.
36. *Introducción a la Disciplina de Requisitos de RUP*. 2010. Conferencia de la Asignatura: Ingeniería de Software 1.
37. **Velázquez Carralero, Alejandro Mario**. *IH-SW-DR-091 ALAS-HIS_Documento de Arquitectura del Sistema*. 2008.
38. **S. Pressman, Roger**. *Ingeniería de Software. Un enfoque práctico*. s.l. : Félix Varela, 2005.
39. **Juristo, Natalia, Moreno, Ana M. y Vegas, Sira**. *Técnicas de evaluación de software*. 2005.

Bibliografía

1. **Akif, Mohammad, y otros.** *Java y XML. s.l.: Anaya.*
2. **Alemán Antelo, Lorena.** *Especificación de Casos de Uso del Módulo Admisión.* 2008.
3. **Allen, Dan.** *Seam in Action.* s.l.: Manning Publications, 2008.
4. **Bauer, Cristian y King, Gavin.** *Java Persistence with Hibernate.* 2005. ISBN 1-932394-88-5.
5. **Becerril C., Francisco.** *Java a su alcance.* ISBN 970-10-1774-9.
6. **Castañero Rodríguez, Dainerys, Fabra Torres, Ángel y García Ramos, Jublar.** *Solución informática para el área de Consulta Externa de las instituciones hospitalarias.*
7. **Castro Espinosa, Osmany y Carrasco Medina, Zadys.** *Trabajo de Diploma para optar por el título de ingeniero en Ciencias Informáticas: Implementación de los procesos de observación y administración del módulo Emergencias del Sistema de Información Hospitalaria alas HIS.* 2010
8. **Delgado Ramos, Ariel y Vidal Ledo, María.** *Informática en la salud pública cubana.*
9. **Díaz Rodríguez, Jorge.** *Trabajo de Diploma para optar por el título de ingeniero en Ciencias Informáticas: Implementación de los procesos del movimiento hospitalario del Sistema de Información Hospitalaria alas HIS.* 2010.
10. **Elliott, James.** *Hibernate: A Developer's Notebook.* 2004. ISBN 0-596-00696-9.
11. **Farley, Jim.** *Practical JBoss Seam Projects.* 2007. ISBN 978-1-59059-863-4.
12. **García Nogueira, Keila.** *IH-SW-DE-033 ALAS-HIS_Admisión_Glosario de Entidades.* 2008.
13. **García Nogueira, Keila.** *IH-SW-DR-079 ALAS-HIS_Estadísticas_Modelo de Casos de Uso del Sistema.* 2008.
14. **Geary, David y Horstmann, Cay.** *Core JavaServer™ Faces.* 2007. 978-0-13-173886-7.
15. *Hibernate Reference Documentation.*

16. Hibernate. [En línea] Red Hat. <https://www.hibernate.org/>.
17. **J. Homes, Barry y T. Joyce, Daniel.** *Object-Oriented programming with Java.*
18. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* 2004.
19. JfreeChart. [En línea] <http://www.jfree.org/jfreechart/>.
20. **Juntao Yuan, Michael y Heute, Thomas.** *JBoss Seam, Simplicity and Power Beyond Java EE.* 2007.
21. **Larman, Craig.** *UML y patrones. Introducción al análisis y diseño orientado a objetos.* 2004.
22. **Ledo Ramírez, Rodney y Ortega Palacios, Alberto.** *Trabajo de Diploma para optar por el título de ingeniero en Ciencias Informáticas: Módulo Admisión del Sistema de Información Hospitalaria alas HIS.* 2009.
23. **Li Guojie, Jackwind.** *UI development with JavaServer Faces.*
24. PostgreSQL. [En línea] PostgreSQL Global Development Group. <http://www.postgresql.org/>.
25. **Red Hat.** *Ajax4jsf Developer Guide.* 2007.
26. **Red Hat.** *RichFaces Developer Guide.* 2007.
27. RichFaces live demo. [En línea] <http://livedemo.exadel.com/richfaces-demo/index.jsp>.
28. **Rubén Cañedo, Andalia, Ramos Ochoa, Raúl y Guerrero Pupo, Julio.** *La Informática, la Computación y la Ciencia de la Información: una alianza para el desarrollo.*
29. **S. Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* s.l. : Félix Varela, 2005.
30. **Sobrevilla, Alfredo.** *Implementación de un sistema de información en salud: GalenHos.*

Glosario de términos

AJAX: Técnica para el desarrollo Web que posibilita la creación de aplicaciones interactivas.

API (Application Programming Interface): Conjunto de funciones y procedimientos que poseen algunas librerías con el objetivo de ser utilizadas por otro software como una capa de abstracción.

Bean: Componente de un software, que tiene la particularidad de ser reutilizable.

Bytecode: Es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.

CIE-10: Es el acrónimo de la Clasificación Internacional de Enfermedades, décima versión. Determina la clasificación y codificación de las enfermedades y una amplia variedad de signos, síntomas, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños y/o enfermedad.

CIE-9-MC: Es un acrónimo de Clasificación Internacional de Enfermedades, novena revisión. Se trata de una clasificación de enfermedades y procedimientos utilizada en la codificación de información clínica derivada de la asistencia sanitaria, principalmente en el entorno de hospitales y centros de atención médica especializada.

Concurrencia: Cualidad de lo simultáneo, es la propiedad de dos eventos que ocurren a la vez, coincidentes en el tiempo.

DDL (Data Definition Language, Lenguaje de Definición de Datos): Es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos, así como de los procedimientos o funciones que permitan consultarlos.

Estándar: Que sirve como tipo, modelo, norma, patrón o referencia por ser corriente.

Framework: Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

GNU LGPL (GNU Lesser General Public License): Licencia de software donde los contratos de licencia de la mayor parte del software están diseñados para jugar con su libertad de compartir y modificar dicho software.

GRASP: Son patrones generales de software para asignación de responsabilidades, es el acrónimo de "General Responsibility Assignment Software Patterns".

Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora): Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

HTML (HyperText Markup Language o Lenguaje de Marcado de Hipertexto): Es el lenguaje de marcado predominante para la elaboración de páginas web.

HTTP (HyperText Transfer Protocol o Protocolo de Transferencia de Hipertexto): Protocolo usado en cada transacción de la Red Global Mundial.

Java EE, Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE): Es una plataforma de programación, parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JavaBeans: Modelo de componentes para la construcción de aplicaciones en Java. Se usan para encapsular varios objetos en un único objeto. Constituyen componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción.

JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web.

JDBC (Java Database Connectivity): Es un API para trabajar con bases de datos desde Java, independientemente de la base de datos a la que se acceda.

JSP (JavaServer Pages): Tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

LPT: Puerto paralelo para conectar periféricos a una computadora.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Open Source (Código abierto): Término con el que se conoce al software distribuido y desarrollado libremente.

Patrón de diseño: Es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, además debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Persistencia: Se refiere a la propiedad de los datos para que estos sobrevivan de alguna manera.

Plataforma: Sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.

Plug-in: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

POJO (Plain Old Java Object): Enfatiza el uso de clases simples y que no dependen de un framework en especial.

SSL (Secure Sockets Layer, Protocolo de Capa de Conexión Segura): Es un protocolo criptográfico que proporciona comunicación segura por una red, comúnmente Internet.

TCP-IP: Conjunto de protocolos de red que permiten la transmisión de datos entre redes de computadoras.

Transacción: Es la ejecución de ciertas instrucciones que acceden a una base de datos compartida.

UI: Interfaz de usuario.

USB (Universal Serial Bus): Puerto que sirve para conectar periféricos a una computadora.

XML (eXtensible Markup Language o Lenguaje de Marcas Extensibles): Es un metalenguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos. No constituye realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.