

Universidad de las Ciencias Informáticas

Facultad 7



Desarrollo del Módulo de Facturación del Sistema de Información Hospitalaria alas HIS

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

Autores: Carlos Javier Infante Ramos
Yuniel García Sánchez

Tutor: Ing. Dismey Saavedra López

Cotutor: Ing. Diuber Estanque Díaz

La Habana, junio de 2011

“Año 53 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 28 días del mes de junio del año 2011.

Carlos Javier Infante Ramos

Autor

Yuniel García Sánchez

Autor

Ing. Dismey Saavedra López

Tutora

Ing. Diuber Estanque Díaz

Cotutor

Datos de contacto

Tutor: Ing. Dismey Saavedra López

Graduada de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Profesora de la Universidad de las Ciencias Informáticas, pertenece al Departamento de Ciencias Básicas de la Facultad 7. En 2 años de labor, se ha desempeñado como analista principal en proyectos del CESIM, este año se encuentra impartiendo la asignatura de Matemática Discreta 1 y en el equipo de desarrollo del Módulo de Facturación perteneciente al Departamento de Sistemas de Gestión Hospitalaria. Cursa maestría en Informática en Salud.

Correo electrónico: dsaavedra@uci.cu

Cotutor: Ing. Diuber Estanque Díaz

Graduado en la Universidad de Ciencias Informáticas (UCI) como Ingeniero en Ciencias Informáticas. Especialista del Centro de desarrollo de Sistemas de Informática Medica (CESIM) y profesor del Departamento Sistemas de Apoyo a la Salud (SAS), líder del equipo de desarrollo del Módulo de Facturación perteneciente al Departamento Sistemas de Gestión Hospitalaria.

Correo electrónico: destanque@uci.cu

Agradecimientos de Carlos Javier Infante Ramos

Agradecer principalmente a mi familia que siempre ha estado a mi lado sin importar las adversidades, tanto en las buenas como en las malas. A mi novia por pasar junto a mí momentos buenos y momentos malos, tanto de día como de madrugada. Agradezco también el apoyo incondicional que me han dado mis amistades, entre ellos Yuniel, Boris, Maikel, Edgar, Indira, Nuria, Sergio, Yisel Morera, Patricia, Yadia, Yenier, Yoel, Eddy Duque, Eddy Elíceo, Frank, Dayanna, y otros muchos que temo dejar de mencionar. Agradecerle a los tutores Dismey y Diuber que hicieron el mayor de los esfuerzos al soportarnos, al oponente por dar siempre el señalamiento oportuno y benéfico. Y por último al tribunal de la tesis en especial a su presidente siempre dispuesto a ayudarnos y a brindarnos un poco más de conocimientos. A todos muchas gracias de corazón.

Agradecimientos de Yuniel García Sánchez

Quisiera agradecerles muy en especial a mis padres por el gran apoyo que siempre me han brindado. También a mis amigos por poder contar con ellos en los momentos difíciles. A mis tutores que me han ayudado a lograr el resultado alcanzado con este trabajo y a la revolución por darme la oportunidad de estudiar en esta universidad de excelencia. Por último y no menos importante a mi compañero de tesis que me ha acompañado en mis aventuras por esta universidad desde primer año. Por las madrugadas de trabajo juntos para realizar este trabajo y por la amistad que nos une.

A mi madre y a mi padre por todo el amor y el cariño que profesan. A mi difunto abuelo, que hubiese querido verme de ingeniero.

Carlos Javier Infante Ramos

Dedico este trabajo a mis padres por el buen ejemplo con que me han educado para que sea un hombre de bien. A mis demás familiares por apoyarme siempre y a mis amigos por siempre estar junto a mi cuando los necesito.

Yuniel García Sanchez

Resumen

El presente trabajo tiene como objetivos principales describir los procesos relacionados con la facturación de servicios médicos en las instituciones hospitalarias y dar solución mediante el desarrollo del Módulo de Facturación para el Sistema de Información Hospitalaria alas HIS a la necesidad de gestionar los cobros de las prestaciones brindadas a un paciente. Este sistema facilitará la gestión de la información en esta área de las instituciones hospitalarias además de minimizar en costo y tiempo los procesos referentes a la facturación.

Para la creación del módulo se utilizó el Proceso Unificado de Desarrollo (RUP) como metodología apoyado en el Lenguaje de Modelado Unificado (UML) y en la Notación para el Modelado de Procesos del Negocio (BPMN). Fue desarrollado mediante el IDE Eclipse; Java como lenguaje de programación y PostgreSQL como Sistema Gestor de Base de Datos. Como arquitectura rectora del sistema se empleó el patrón Modelo Vista Controlador (MVC), incluyendo frameworks, componentes e Hibernate para el Mapeo Objeto Relacional (ORM).

Con este módulo se optimizarán los procesos relacionados con la facturación de servicios médicos. Se agilizará la gestión de facturas, presupuestos y de las relaciones Obra Social-Paciente-Institución Hospitalaria. Además se mejorará el manejo de los datos y la generación de reportes relacionados con la planificación hospitalaria.

Palabras claves: facturación, gestión, Sistema de Información Hospitalaria, factura.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1. CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA	6
1.2. SISTEMAS GESTORES DE LA INFORMACIÓN MÉDICA QUE INCLUYEN UN MÓDULO DE FACTURACIÓN.....	8
1.2.1. Sistemas internacionales.....	8
1.2.1.1. Net Clínicas	8
1.2.1.2. SALUS.....	9
1.2.1.3. Michigan.....	9
1.2.2. Sistemas nacionales	10
1.2.2.1. Sistema de Información Hospitalaria HIS.....	10
1.2.2.2. Sistema Integral Económico Administrativo RODAS XXI	10
1.3. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.....	11
1.3.1 Aplicaciones Web	12
1.3.2 Patrón de arquitectura.....	12
1.3.2.1 Modelo – Vista – Controlador (MVC)	12
1.3.3 Entornos distribuidos	13
1.3.3.1 Modelo Cliente –Servidor.....	13
1.3.4 Tecnologías libres.....	13
1.3.5 Capa de presentación	13
1.3.5.1 Java Server Faces (JSF).....	14
1.3.5.2 RichFaces.....	14
1.3.5.3 Ajax	14
1.3.5.4 Ajax4jsf.....	14
1.3.5.5 Facelets.....	14
1.3.5.6 XHTML	15
1.3.5.7 Seam UI	15

1.3.6	Capa de negocio	15
1.3.6.1	Seam.....	15
1.3.7	Capa de datos	16
1.3.7.1	Hibernate.....	16
1.3.7.2	Enterprise Java Bean 3 (EJB 3).....	16
1.3.8	Sistema Gestor de Base de Datos	17
1.3.8.1	PostgreSQL.....	17
1.3.9	Tecnologías horizontales.....	17
1.3.9.1	Java EE 5.....	17
1.3.9.2	JRE 6.....	18
1.3.10	Herramientas.....	18
1.3.10.1	Eclipse.....	18
1.3.10.2	JBoss Tools.....	18
1.3.10.3	JBoss Server.....	19
1.3.11	Herramienta de modelado	19
1.3.11.1	Visual Paradigm para UML.....	19
1.3.11.2	Proceso Unificado de Desarrollo (RUP).....	19
1.3.11.3	Notación para el Modelado de Procesos del Negocio (BPMN)	20
1.3.11.4	Lenguaje Unificado de Modelado (UML).....	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA		22
2.1	OBJETO DE ESTUDIO.....	22
2.1.1	Flujo actual de los procesos involucrados en el campo de acción	22
2.2	INFORMACIÓN QUE SE MANEJA	23
2.3	MODELADO DEL NEGOCIO	25
2.3.1	Definición de roles	26
2.3.2	Diagramas de procesos del negocio	26
2.4	ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA	33
2.4.1.	Requisitos funcionales	34
2.4.2	Requisitos no funcionales	35

2.4.2.1	Usabilidad	35
2.4.2.2	Seguridad.....	35
2.4.2.3	Rendimiento.....	36
2.4.2.4	Soporte.....	36
2.4.2.5	Hardware.....	37
2.4.2.6	Software	38
2.4.2.7	Restricciones de diseño.....	38
2.4.2.8	Requisitos para la documentación de usuarios en línea y ayuda del sistema	38
2.4.2.9	Requisitos de interfaz.....	38
2.4.2.10	Portabilidad.....	39
2.4.3	Modelo de casos de uso del sistema	39
2.4.3.1	Definición de actores.....	39
2.4.3.2	Vista global de actores del sistema.....	40
2.4.3.3	Diagrama de casos de uso del sistema.....	41
2.4.4	Descripción textual de los casos de uso arquitectónicamente significativos	42
2.4.4.1	Crear factura.....	42
2.4.4.2	Crear presupuesto.....	43
2.4.4.3	Crear convenio.....	44
CAPÍTULO 3: DISEÑO.....		45
3.1	DESCRIPCIÓN DE LA ARQUITECTURA.....	45
3.2	ANÁLISIS DE POSIBLES IMPLEMENTACIONES, COMPONENTES O MÓDULOS YA EXISTENTES Y QUE PUEDAN SER REHUSADOS. ESTRATEGIAS DE INTEGRACIÓN.....	46
3.3	MODELO DE DISEÑO	47
3.3.1	Patrones del diseño	47
3.4	DIAGRAMA DE PAQUETES	49
3.5	ESTRUCTURA DEL DISEÑO	50
3.5.1	Diagrama de clases del diseño	51
3.6	DIAGRAMA DE SECUENCIA	52
3.7	DESCRIPCIÓN DE LAS CLASES Y SUS ATRIBUTOS.....	52

3.8	MODELO DE DATOS	58
3.8.1	Descripción de las tablas.....	59
CAPÍTULO 4: IMPLEMENTACIÓN		63
4.1	MODELO DE DESPLIEGUE.....	63
4.2	DIAGRAMA DE PAQUETES DE COMPONENTES	64
4.2.1	Diagrama de componentes “Crear factura”	66
4.3	VALIDACIONES.....	66
4.4	DESCRIPCIÓN DE ALGORITMOS UTILIZADOS	67
4.5	TRATAMIENTO DE ERRORES	68
4.6	SEGURIDAD.....	69
CONCLUSIONES		70
RECOMENDACIONES		71
REFERENCIAS BIBLIOGRÁFICAS		72
BIBLIOGRAFÍA		75
ANEXO 1: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....		78

Índice de tablas

Tabla 1. Características de sistemas existentes que gestionan la facturación médica.....	11
Tabla 2. Trabajadores del negocio.....	26
Tabla 3. Requisitos funcionales.....	35
Tabla 4. Actores del sistema	39
Tabla 5. Descripción textual del caso de uso “Crear factura”.....	43
Tabla 6. Descripción textual del caso de uso “Crear presupuesto”.....	43
Tabla 7. Descripción textual del caso de uso “Crear convenio”	44
Tabla 8. Descripción de la clase “CrearFacturaControlador”	54

Tabla 9. Descripción de la clase “CrearPresupuestoControlador”	55
Tabla 10. Descripción de la clase “CrearMonedaControlador”	57
Tabla 11. Descripción de la tabla “Factura”	60
Tabla 12. Descripción de la tabla “Moneda”	61
Tabla 13. Descripción de la tabla “Obra Social”	62

Índice de figuras

Figura 1. Diagrama del proceso del negocio “Procesos de facturación”	28
Figura 2. Diagrama del proceso del negocio “Registrar prestaciones ambulatorias”	29
Figura 3. Diagrama del proceso del negocio “Registrar prestaciones de internados”	30
Figura 4. Diagrama del proceso del negocio “Realizar facturación”	31
Figura 5. Diagrama del proceso del negocio “Registrar presupuesto”	32
Figura 6. Diagrama del proceso del negocio “Cierre de caja”	33
Figura 7. Actores del sistema	40
Figura 8. Diagrama de caso de uso del sistema “Crear factura”	41
Figura 9. Diagrama de caso de uso del sistema “Crear presupuesto”	42
Figura 10. Diagrama de paquetes	50
Figura 11. Diagrama de clase del diseño “Gestionar factura”	51
Figura 12. Diagrama de secuencia “Crear factura”	52
Figura 13. Modelo de datos para el Módulo de Facturación.	58
Figura 15. Diagrama de despliegue.	64
Figura 16. Diagrama de paquetes de componentes	65
Figura 17. Diagrama de componentes “Crear factura”	66
Figura 18. Diagrama de flujo para búsqueda lineal.	68
Figura 19. Interfaz “Seleccionar Historia Clínica a facturar”	78

Figura 20. Interfaz “Crear factura”	79
Figura 21. Interfaz “Ver datos de factura”	80
Figura 22. Interfaz “Cerrar caja”	81
Figura 23. Interfaz “Establecer prestaciones entidad”	82

Introducción

Mediante el desarrollo tecnológico el hombre ha intentado automatizar los procesos de trabajo para mejorar la calidad y productividad de sus actividades. La informática ha formado parte de este avance impulsor donde se crean nuevas alternativas menos complejas y capaces de brindar una mejor respuesta a los problemas existentes en la sociedad. Con el surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC) y el aprovechamiento de las mismas surge una nueva era dentro de las soluciones informáticas.

El uso de las Tecnologías de la Información y las Comunicaciones ha impulsado al avance progresivo hacia la automatización de procesos, con la creación de sistemas capaces de almacenar grandes cantidades de información y procesar enormes volúmenes de datos. La salud forma parte de estos avances, y es, dentro de los sectores sociales uno de los más importantes, donde el flujo de pacientes y la calidad del servicio suponen los mayores retos. Son estas dificultades las que han promovido el uso progresivo de sistemas gestores de la información médica.

Los sistemas gestores de la información médica facilitan los procesos internos de las instituciones, aumentan la calidad en la atención a los pacientes y mejoran la labor por parte de los trabajadores en los diferentes niveles de atención. Tienen un fuerte impacto en las instituciones de salud, pues elevan la calidad en la atención y en los servicios brindados. Además aplican la información obtenida a las áreas de investigación, clínica, docente y administrativa con el propósito adicional de disminuir los costos y elevar la productividad.

Desde los inicios, los sistemas de información en la salud se desarrollaron en ambientes hospitalarios e instituciones vinculadas a la salud. Las primeras prácticas de informatización se enfocaron en los procesos administrativos, con énfasis en la gestión contable y financiera de los servicios médicos. Posteriormente estas soluciones informáticas se orientaron a la información clínica centrada en el paciente como eje principal, brindando nuevas opciones al personal médico y al propio paciente.

Los Sistemas de Información Hospitalaria (SIH ó HIS por sus siglas en inglés) son sistemas informáticos destinados a la gestión de la información en instituciones hospitalarias de forma eficiente y robusta. Estos sistemas son diseñados con el objetivo de mejorar la calidad y la eficiencia del trabajo en los centros de

atención médica. Los mismos se encargan de almacenar, procesar, recopilar, recuperar y comunicar los datos obtenidos en la atención al paciente y en la labor administrativa de las instituciones hospitalarias.

Un Sistema de Información Hospitalaria se encuentra conformado por módulos, los cuales responden a cada una de las áreas de las instituciones hospitalarias en las que brindan servicios. Entre sus características principales se encuentran la interacción entre las diferentes áreas de atención, de forma tal que la información generada en cualquiera de las áreas pueda ser almacenada, y al mismo tiempo fluir hasta el resto de los módulos con los que interactúan.

En una institución hospitalaria se pueden encontrar diferentes áreas: por la parte asistencial están Hospitalización, Admisión, Laboratorio, Emergencias, entre otras. Por la parte administrativa, Estadísticas, Facturación y otras. La información generada en cada una de las áreas de la institución, el flujo de datos, los reportes y demás actividades, pueden ser controladas por la administración central del hospital.

Cada uno de los módulos de un HIS trabaja usando como condición necesaria la información proporcionada por el personal que interactúa con el sistema. A nivel mundial existen instituciones donde deben ser pagados los servicios médicos, una de las principales tareas de la facturación de servicios médicos es reflejar la entrega o prestación de un servicio a un cliente y el modo de compensación de los mismos.

Los módulos de facturación surgen dentro de los Sistemas de Información Hospitalaria con el objetivo de integrar un solo sistema de gestión de la información y así obtener entre sus beneficios una mejor gestión de la economía. Se encuentran destinados fundamentalmente al control de las operaciones financieras que se realizan en una institución hospitalaria alrededor de un paciente.

El propio paciente para ejercer el pago tiene varias opciones que pueden fundamentar su pago, entre ellas se encuentran las Obras Sociales encargadas de sufragar los gastos de sus clientes ante una institución hospitalaria. También puede ejercer su pago de forma personal o mediante un tutor que pueda representarlo. Estos sistemas permiten el pago con efectivo, a través de una transferencia bancaria o ejercer una variante de pago a plazos.

Durante la estancia del paciente en el hospital se gestionan las tarifas desde citas médicas, productos farmacéuticos, procedimientos y demás servicios que lo asisten. Los elementos logísticos también forman parte de los costos en los que incurre un paciente. Al crear la factura son agrupados los servicios y

consumos prestados al paciente y conformada la hoja de factura, lista para el pago. La información de la misma es enviada a la obra social y en caso de no tenerla es cargada directamente al paciente o a su tutor.

Los médicos o especialistas de un hospital son los encargados de brindar todas las atenciones para con un paciente, en dependencia de su especialidad, estos pueden recibir honorarios en su pago gracias a los resultados obtenidos. Estos honorarios son también gestionados en el hospital por el sistema de facturación.

En todas las instituciones médicas están presentes los procesos de facturación, para ejercer los mismos es necesario un gran esfuerzo y alta calidad en el trabajo realizado, hoy día gran parte de estos procesos son llevados a cabo utilizando métodos tradicionales que ya resultan poco efectivos. Esto trae como consecuencia que los mismos se realicen de forma ineficiente, se dificulte mucho el trabajo y la manipulación, debido fundamentalmente a la cantidad de información que es gestionada, obteniéndose al final resultados poco productivos y de muy bajo rendimiento en el área de facturación. (1)

Para la gestión de la información que genera la facturación es de suma importancia estar claro que:

“El conocimiento generado a partir de la facturación, puede servir de gran ayuda a las instituciones en el análisis de las gestiones financieras, la planificación y el control de ganancias; pero como todo este conocimiento se encuentra descentralizado, es difícil de procesar y por tanto no se aprovecha al máximo. (...) Además, la información almacenada en formato duro, expuesta a sucesos impredecibles, es susceptible al deterioro y pérdida en la institución. Por ello el período de vida de la información de las facturas en este formato está sujeto a un tiempo limitado.” (2)

Los procesos de facturación son manejados por grupos de especialistas dentro de la entidad hospitalaria, estos son los encargados de recolectar diariamente la información relacionada con los documentos y datos relacionados con los pacientes en las distintas áreas de un hospital. Esta actividad por sí es sumamente agotadora y los errores pueden devenir en un alto grado si el trabajo no se efectúa con sumamente cuidado y precisión.

Realizar un buen proceso de facturación también forma parte de las ventajas aportadas al paciente:

“(...) los extensos y engorrosos procesos en ocasiones conllevan a que existan contradicciones financieras, de las cuales se derivan todo un conjunto de reclamaciones en las que el paciente debe invertir tiempo, dinero y preocupaciones.” (3)

En los casos particulares de instituciones médicas que presten servicios de forma gratuita resulta de mucha utilidad el sistema de facturación, para la institución continúa aportando los antes mencionados beneficios y para el paciente es una vía indiscutible para conocer los gastos a los que fue sometido y le fueron sufragados.

Planteada la situación anterior se define el siguiente **problema a resolver**: ¿Cómo integrar la gestión de la información de los procesos de facturación de los servicios médicos al Sistema de Información Hospitalaria alas HIS?

Este problema se puede ver enmarcado en el **objeto de estudio** concerniente al proceso de gestión de la información en las instituciones hospitalarias, delimitado por el **campo de acción** que comprende los procesos de gestión de la información referentes a la facturación de servicios médicos en las instituciones hospitalarias.

Basado en problema se define como **objetivo general** del presente trabajo: Desarrollar el Módulo de Facturación del Sistema de Información Hospitalaria alas HIS, que facilite la gestión de la información referente a la facturación en las instituciones hospitalarias.

Para dar solución al objetivo antes propuesto se definen las siguientes **tareas de la investigación** a desarrollar:

- ✓ Asimilar el Trabajo de Diploma “Diseño del Módulo de Facturación del Sistema de Información Hospitalaria alas HIS”.
- ✓ Analizar las tendencias actuales de las soluciones informáticas relacionadas con la facturación de servicios médicos en las instituciones hospitalarias.
- ✓ Asimilar la arquitectura y tecnologías definidas para el desarrollo del Sistema de Información Hospitalaria alas HIS.
- ✓ Identificar los requerimientos del software basados en los modelos de procesos del negocio.
- ✓ Obtener la documentación ingenieril del Módulo de Facturación, según la metodología RUP.
- ✓ Desarrollar el Módulo de Facturación del Sistema de Información Hospitalaria alas HIS aplicando las pautas de diseño y guiándose por el modelo de caso de uso.

Beneficios:

- ❖ Facilitar el manejo de la información a facturar, lo que permite el fácil acceso a la información y la gestión de la misma.

- ❖ Garantizar la persistencia de los documentos de facturación, al prolongar el período de vida de las facturas.
- ❖ Lograr exactitud en los cálculos de los procesos de facturación, aprovechando de forma eficiente el conocimiento obtenido en dichos procesos.
- ❖ Mejorar la planificación y predicción financiera en las instituciones hospitalarias, luego de obtener los cálculos y las relaciones entre pacientes, prestaciones e instituciones hospitalarias.

El presente documento se encuentra estructurado en cuatro capítulos, el primero de ellos, **“FUNDAMENTACIÓN TEÓRICA”**, ubica al lector en el ambiente de desarrollo del módulo de facturación, justificándose las tendencias, tecnologías, metodologías y herramientas que fueron utilizadas para el desarrollo de los procesos seleccionados. Seguidamente el segundo capítulo, **“CARACTERÍSTICAS DEL SISTEMA”**, contiene la fundamentación de la arquitectura, la estrategia de integración, los elementos que garantizan la seguridad en el sistema y la especificación de los estándares y estilos de codificación a utilizar.

En el tercer capítulo **“DISEÑO”** se implementan las clases y subsistemas en términos de componentes. Se presenta la propuesta de solución para lograr una gestión más eficiente de los procesos hospitalarios asociados al área en cuestión. El cuarto y último capítulo **“IMPLEMENTACIÓN”**, donde se explica el subsistema en términos de componentes.

Capítulo 1: Fundamentación teórica

La gran cantidad de información que se manejan en las instituciones hospitalarias en los momentos actuales ha condicionado la necesidad de crear los Sistema de Información Hospitalaria (HIS), y con esto la necesidad de incluirle un modulo capaz de facturar los servicios médicos que son llevados a cabo en la institución hospitalaria. Con el presente capítulo se pretende explicar de forma breve las terminologías de uso indispensable en los procesos de facturación. También se describen las distintas herramientas a utilizar para el diseño y desarrollo del mismo, las distintas metodologías de trabajo y los patrones de arquitectura y diseño.

1.1. Conceptos básicos relacionados con el dominio del problema

- **Factura:** Hace referencia al documento oficial entregado al consumidor en un proceso de compraventa o recibo de servicios. El mismo es confeccionado por el proveedor, quien lo ofrece como constancia de la transacción realizada y representa el derecho de cobro de este. La factura incluye datos de los participantes en el acuerdo y las especificaciones de las operaciones. (4)
- **Factura electrónica:** No es más que el conjunto de datos o registros ordenados de modo racional que son almacenados electrónicamente, pudiendo ser leídos por equipos de procesamiento de datos. La misma contiene información de ventas de mercancías determinadas u ofrecimiento de servicios documentados por una organización empresarial o profesional. (5)

Las facturas pueden catalogarse en varias tipologías de acuerdo con las características particulares que estas posean.

- **Factura ordinaria (Tipo A):** aquella que documenta una operación comercial o un servicio. (6)
- **Factura rectificativa (Tipo B):** aquella que documenta correcciones de una o más facturas anteriores, o bien devoluciones de productos, envases y embalajes o comisiones por volumen. (7)
- **Factura recapitulativa (Tipo C):** factura que documenta agrupaciones de facturas en un período de tiempo determinado. Para que esta factura tenga validez fiscal se han de anular las anteriores a las que haga referencia. (8)

Capítulo 1: Fundamentación teórica

En función del valor que poseen las facturas se pueden clasificar en:

- **Pro-forma:** Documenta una oferta, con indicación de la forma exacta que tendrá la factura tras el suministro. No tienen valor contable ni como justificante. Suele incluir la fecha máxima de validez.
- **Copia:** Documenta la operación para el emisor, con los mismos datos que el original. Debe llevar la indicación de copia para permitir distinguirla del original.
- **Duplicado:** Documenta la operación para el receptor, en caso de pérdida del original. La expide el mismo emisor que expidió el original y tiene los mismos datos que el documento auténtico. Debe llevar la indicación de duplicado para permitir distinguirla del original, especialmente para el caso de que reaparezca el documento auténtico. (9)

Otros conceptos también manejados por facturación:

- **Impuesto al Valor Agregado (IVA):** Impuesto al consumo, que se aplica a la venta de activos, a la prestación de servicios y a las importaciones de ciertos bienes. El IVA es un mecanismo para financiar el gasto público mediante el cual el estado redistribuye ingresos desde los sectores más ricos hacia los más vulnerables. Específicamente, los impuestos al consumo gravan el acto de consumir bienes y servicios tanto de alimentos, bebidas, indumentaria, artículos de tocador y perfumería, combustibles, servicios públicos, seguros, entre otros.
- **Gestión de cobros:** Es la acción de administrar el capital existente en la institución, teniendo en cuenta los distintos métodos de cobros y la cancelación de deudas. La misma permite obtener un reporte del balance financiero de la institución, así como la notificación de los cobros pendientes. Para realizar la gestión de cobros, diariamente se debe realizar la apertura y cierre de caja en la organización.
- **Obras Sociales:** Es la reinversión de parte de los beneficios, producto de las actividades de las organizaciones sin ánimo de lucro para el bien común. Las obras sociales son entidades encargadas de organizar la prestación de la atención médica de los trabajadores.
- **Convenio:** Acuerdo entre dos o más grupos sociales o instituciones por el que ambas partes aceptan una serie de condiciones y derechos.
- **Abono o anticipo:** Cantidad de dinero que se paga anticipadamente.
- **Afiliación:** Inscripción de una persona como miembro de un partido, un sindicato u otra asociación.
- **Prestaciones:** Servicio o ayuda que una persona, una institución o una empresa ofrece a otra.

Capítulo 1: Fundamentación teórica

- **Módulo de prestaciones:** Es el conjunto, paquete de servicio o ayuda que una persona, una institución o una empresa ofrece a otra.
- **Presupuesto:** Cálculo de los gastos e ingresos que se producirán en un periodo de tiempo determinado.
- **Titular:** Persona o entidad que da su nombre para que figure como título de algo o para que conste como propietario o sujeto activo de un derecho.
- **Pagador:** Persona encargada de pagar las pensiones, sueldos, créditos, etc., en un banco o en una oficina.
- **Dígito control:** Son un conjunto de dígitos asociados a una cuenta bancaria los cuales se encargan de verificar la validez de la cuenta y de la sucursal bancaria además de establecer su concordancia.
- **Transferencia bancaria:** Operación bancaria que consiste en transferir dinero guardado en una cuenta para otra.
- **Domiciliación:** Medio de pago consistente en dar la orden a un banco de que atienda periódicamente, hasta nuevo aviso, todos los recibos que una empresa, administración o particular le envíe a nombre del cliente, pasando el cobro a la cuenta bancaria con la citada periodicidad.

1.2. Sistemas gestores de la información médica que incluyen un módulo de facturación

En las instituciones hospitalarias uno de los procesos que requiere mayor precisión por el gran número de información útil que se obtiene es el de facturación, por lo que en diversos sistemas informáticos se han incluido soluciones que gestionan en mayor o menor medida este proceso.

1.2.1. Sistemas internacionales

1.2.1.1. Net Clínicas

Este sistema de gestión de la información médica es entre los software españoles uno de los más usados. El mismo tiene una gestión de la información muy eficiente y sencilla, y sobre todo potente. Net Clínicas permite la gestión de innumerables datos dentro de un centro de salud, entre los que se pueden observar el tratamiento de ficheros de pacientes, trabajadores, mutuas entre otros; permite una gestión de agenda y citas muy eficiente. Permite la gestión de ingresos, presupuestos, informes y algunas utilidades que brinda el propio sistema. (10)

Capítulo 1: *Fundamentación teórica*

Net Clínicas es sumamente eficiente en la facturación de prestaciones médicas. Admite la gestión de mutuas y pagos a través de estas, de pagos anticipados y abonos, así como recibos y facturas. Puede realizar el cobro de facturas por domiciliación bancaria.

El software cumple con todas las reglas de protección de datos, es intuitivo y fácil de usar. Está desarrollado con tecnología de punta y dispone de un potente gestor de base de datos. Está basado en la arquitectura cliente/servidor y puede ser utilizado tanto en equipos mono-puestos, como en grandes redes como aplicación multi-puesto.

1.2.1.2. SALUS

El sistema SALUS es un software de origen español, muy usado en la atención médica que brinda un gran número de funcionalidades y es de muy fácil configuración. El mismo opera en tres áreas de gestión de la información, como son la historia clínica, agendas y facturación posibilitando proveer para estas áreas una gran variedad de servicios y funcionalidades entre los que se pueden encontrar administración de productos ó servicios, facturas y control de gastos, compras, cobro de facturas, control de caja entre otros. (11)

1.2.1.3. Michigan

El Michigan es un sistema de gestión de la información médica de origen argentino que incluye variadas opciones dentro de sus módulos. El mismo abarca de una forma muy genérica el tratamiento de la información gestionada en los centros de salud en argentina. Su sistema de facturación se encuentra basado fundamentalmente en el nomenclador argentino INOS que regula el precio de las prestaciones médicas en el territorio nacional. También cuenta con otros servicios como son la contratación, factura ambulatoria, facturas a internados, titulares que son las personas responsables de pagar los gastos de un paciente entre otros. (12)

Este sistema también se especializa en trabajar los procesos de facturación. De modo que incluye en sus funcionalidades distintos módulos dedicados a gestionar esta información. Entre estos se pueden encontrar los de facturación de internados, facturación ambulatoria y administración.

Capítulo 1: Fundamentación teórica

1.2.2. Sistemas nacionales

1.2.2.1. Sistema de Información Hospitalaria HIS

El Sistema de Información Hospitalaria HIS es un producto Web desarrollado mediante la colaboración de ingenieros españoles y la CUJAE. Para su implementación se utilizó PHP como lenguaje de programación. Este software cuenta con un conjunto de módulos como son admisión, hospitalización, consulta, laboratorio, configuración, emergencias, banco sangre, facturación entre otros. El módulo de facturación permite facturar las prestaciones al paciente ya sea de forma integrada a los módulos o de forma separada. También permite la creación de presupuestos y de paquetes de prestaciones además de generar distintos reportes de suma importancia para la institución.

1.2.2.2. Sistema Integral Económico Administrativo RODAS XXI

El Sistema Integral Económico Administrativo Rodas XXI es un software económico administrativo desarrollado por la empresa CITMATEL en el año 1999 utilizando ACCESS como gestor de base de datos y evolucionó hacia la Versión 3.0 sobre SQL en el año 2005. El mismo está conformado por un conjunto de módulos como son finanzas el cual permite el control de los movimientos financieros dentro de la entidad; activos fijos que permite el control de los activos fijos tangibles dentro de la entidad, inventario, almacén, costos además de facturación y otros. El módulo de facturación permite entre otras cosas realizar facturas, pre facturas y re-facturas. También permite realizar la facturación en moneda nacional y en divisas. Genera distintos informes como son: factura, pre factura, productos más vendidos entre otros.

(13)

Estos sistemas en la actualidad se encuentran en explotación y son capaces de gestionar total o parcialmente la facturación en las distintas instituciones hospitalarias. Sin embargo estos sistemas en ocasiones producto a cualidades de los mismos como pago por propiedad y entorno Web o escritorio, como se muestra en la siguiente tabla, se hace ineficiente su uso o implantación en algunas instituciones.

Capítulo 1: Fundamentación teórica

Nombre	Software propietario	Aplicación Web	Aplicación escritorio	Plataforma	Año de lanzamiento al mercado
Net clínicas	X		X	Windows	2009
SALUS	X		X	Windows	1995
Michigan	X		X	Windows	1998
Sistema de Información Hospitalaria HIS		X		Multi-plataforma	2008
Sistema Integral Económico Administrativo RODAS XXI	X		X	Windows	2004

Tabla 1. Características de sistemas existentes que gestionan la facturación médica.

1.3. Tendencias y tecnologías actuales a considerar

No se puede hablar de un software de calidad sin tener en cuenta las tendencias y tecnologías informáticas existentes a nivel mundial. Siendo para ello necesario hacer una correcta selección de las herramientas y técnicas a utilizar para lograr que el producto cumpla con la mayor calidad los requisitos definidos por el equipo.

La tendencia en la actualidad para el desarrollo de software es la utilización de herramientas libres de licencias, donde los sistemas para la facturación no están libres de esta tendencia. También se busca que los mismos sean cada vez más configurables y multiplataforma logrando así su funcionamiento en cualquier sistema operativo con la menor cantidad de cambios posibles.

Capítulo 1: *Fundamentación teórica*

1.3.1 Aplicaciones Web

Una aplicación Web ofrece una gran variedad de ventajas, posibilitando las condiciones necesarias para un buen funcionamiento del proyecto. Estas aplicaciones son sistemas donde los usuarios a través de la red pueden acceder a un servidor utilizando la arquitectura cliente-servidor. La misma permite que cualquier cambio o actualización que se haga en la aplicación y se actualice en el servidor, al mismo tiempo sea vista por los usuarios sin tener que ir actualizando por estaciones de trabajo.

Los procesos de instalación de una aplicación Web son sumamente sencillos ya que pueden realizarse en el servidor y los usuarios mediante el navegador acceden al mismo sin involucrarse en el proceso. Estas aplicaciones no dependen de una plataforma específica posibilitando que si existen variaciones en el sistema no se vea afectada su posterior utilización.

Entre las principales desventajas que presentan las aplicaciones Web se encuentran que necesitan de una conexión veloz y permanente, situación que en algunos casos puede tener dificultad.

1.3.2 Patrón de arquitectura

1.3.2.1 Modelo – Vista – Controlador (MVC)

La arquitectura Model-View-Controller (MVC), está formada por tres niveles donde:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio, el modelo y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HyperText Markup Language (HTTP), consola de comandos, email). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (14)

Capítulo 1: Fundamentación teórica

1.3.3 Entornos distribuidos

1.3.3.1 Modelo Cliente –Servidor

La arquitectura cliente - servidor no es más que un modelo para el desarrollo de sistemas en el cual las transacciones se dividen en procesos independientes que colaboran entre sí. Tiene como principal objetivo intercambiar información, denominándosele cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este tipo de modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada uno. (15)

El modelo tiene como ventaja que los desarrollos son más cortos, existe una mayor duración de las aplicaciones, es posible reconfigurar el servidor o los clientes sin la necesidad de alterar los programas que constituyen la aplicación. Y además permiten un aumento de rendimiento y reducción de gastos, debido a que la aplicación se puede redistribuir por distintas máquinas.

Este modelo tiene como arquitectura básica:

- Interfaz de Usuario: La presentación al usuario, con las entradas de datos y las consultas.
- Reglas de Negocio: El procesamiento de la información.
- Accesos a Datos: El control del almacén de datos.

1.3.4 Tecnologías libres

Entre los aspectos más importantes de la factibilidad a la hora de decidirse por alguna herramienta o tecnología se encuentran el costo de producción, despliegue y utilización del sistema. Si el software es muy costoso generalmente será utilizado por un pequeño grupo de personas, esto está dado por la poca accesibilidad en ocasiones y los costos de las licencias propietarias que los sustentan. Por lo que es recomendable para el desarrollo del software la utilización de herramientas libres, las cuales permitan desarrollar un producto de alta calidad y con costos mínimos.

1.3.5 Capa de presentación

La capa de presentación es la encargada de mostrar la información de los datos mediante los distintos componentes visuales. Siendo esta la encargada de solicitar y recibir los servicios que ofrecen otros componentes independientemente del nivel que pertenezcan haciéndolo de forma elegante y transparente ante el usuario.

Capítulo 1: *Fundamentación teórica*

1.3.5.1 Java Server Faces (JSF)

El Java Server Faces es un framework de desarrollo basado en el Modelo Vista Controlador (MVC). Este framework permite desarrollar de forma sencilla y eficaz interfaces de usuarios muy distintas a las acostumbradas en las aplicaciones Web, además el mismo permite la compatibilidad entre tecnologías como son Java Server Page (JSP), RichFaces y otras más que serán mencionadas a continuación. (16)

1.3.5.2 RichFaces

RichFaces es una librería de código abierto la cual permite añadir capacidades Ajax a aplicaciones desarrolladas con JSF sin la utilización de JavaScript. Esta librería se integra perfectamente con Seam, también incluye ciclo de vida, validaciones, conversores y gestiona los recursos tanto estáticos como dinámicos. (17)

1.3.5.3 Ajax

Asynchronous JavaScript + XML (Ajax) no es un lenguaje de programación si no una forma de utilizar las normas existentes. Su utilización posibilita actualizar la página o una parte de esta sin necesidad de recargar la página el intercambio de datos con un servidor. Se puede utilizar en múltiples plataformas pues está basada en estándares abiertos como JavaScript y Document Object Model (DOM). (18)

1.3.5.4 Ajax4jsf

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código JavaScript.

Ajax4jsf presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF después de que la petición Ajax cambie los datos en el servidor. (19)

1.3.5.5 Facelets

Es un framework ligero cuya función principal es permitir la utilización de plantillas en JSF. El mismo tiene como principales ventajas la construcción de interfaces basadas en plantillas, la rápida creación de componentes por composición y la fácil creación de funciones y librerías de componentes. (20)

Capítulo 1: *Fundamentación teórica*

1.3.5.6 XHTML

Extensible Hypertext Markup Language (XHTML) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en Extensible Markup Language (XML). XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. (21)

1.3.5.7 Seam UI

Son una serie de controles JSF altamente integrables con JBoss Seam que adicionan varias mejoras a JSF, desde validación, expresiones Extended EL, hasta la integración de la navegación en la interfaz de usuario basada en pageflows o procesos del negocio. Contienen propiedades que incluye entre otros, la simplificación de etiquetas y anotaciones de componentes JSF. (22)

1.3.6 Capa de negocio

La capa del negocio es la encargada de recibir las solicitudes y presentar los resultados utilizando la capa de presentación, al mismo tiempo que invoca la capa datos para almacenar o recuperar información.

1.3.6.1 Seam

Seam es una potente plataforma de desarrollo de código abierto realizada para construir aplicaciones Web en Java. La misma integra tecnologías como Ajax, Java Server Faces (JSF), Java Persistence API (JPA), Enterprise Java Beans (EJB 3.0) y Business Process Management (BPM) en un sistema unificado con sofisticadas herramientas. Seam también se integra perfectamente con otros frameworks entre los que se encuentran RichFaces, ICEFaces, MyFaces, Hibernate y Spring. (23)

Seam ha sido diseñado desde cero para eliminar la complejidad, tanto en la arquitectura como los niveles de la Interfaz de Programación de Aplicaciones (Application Programming Interface ó API). Permite a los desarrolladores ensamblar aplicaciones Web complejas utilizando simples clases de Java, un rico conjunto de componentes de interfaz de usuario y muy poco XML. Elimina los errores más comunes que se encuentran en las aplicaciones Web tradicionales. (24)

Capítulo 1: *Fundamentación teórica*

1.3.7 Capa de datos

La capa de acceso a datos contiene principalmente las tecnologías utilizadas para acceder y persistir los datos. Se encuentra conformada por uno o más gestores de bases de datos encargados de almacenar, recibir y recuperar la información a través de los componentes agrupados en la misma.

1.3.7.1 Hibernate

Esta es una herramienta de Object-Relational mapping (ORM) 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; permitiendo establecer estas relaciones. La misma, brinda la posibilidad de generar bases de datos en cualquiera de los entornos soportados: Oracle, DB2, MySQL, además de ser una herramienta de código abierto. (25)

Hibernate 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). 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 (POO).

La herramienta convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Además, 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.

1.3.7.2 Enterprise Java Bean 3 (EJB 3)

Los EJB son una de las Interfaces de Programación de Aplicaciones, cuyo acrónimo en inglés es Application Programming Interface (API). Estos 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 para centrarse en el desarrollo de la lógica de negocio en sí. (26)

El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. Incluida en este estándar se encuentra la Java Persistence API, más conocida por su sigla JPA, es la API

Capítulo 1: *Fundamentación teórica*

de persistencia desarrollada para la plataforma Java Enterprise Edition (Java EE). Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos como sucedía con EJB 2 y permitir usar objetos regulares conocidos como Plain Old Java Object (POJOs), los cuales permiten obtener servicios, ejecutando pequeñas porciones de código en el contenedor EJB.

1.3.8 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por lo que se trata de un software de propósito general. Ejemplos de SGBD son Oracle y SQL Server, PostgreSQL, entre otros. (27)

1.3.8.1 PostgreSQL

Es un Sistema Gestor de Base de Datos (SGBD) que ofrece una potencia adicional al incorporar los conceptos de clases, herencia, tipos y funciones; de modo que los usuarios puedan extender fácilmente el sistema. PostgreSQL incluye reglas, restricciones, disparadores y una integridad transaccional que le aportan flexibilidad y potencia. (28) Estas características incorporan a este gestor en la categoría de las bases de datos identificadas como objeto-relacionales, aunque este incluye algunos detalles pertenecientes al mundo de las bases de datos orientadas a objetos. (29)

1.3.9 Tecnologías horizontales

1.3.9.1 Java EE 5

Esta tecnología incluye varias especificaciones de API, tales como Java Database Connectivity (JDBC), Java Remote Method Invocation (RMI), Java Message Service (JMS), servicios Web, XML, email y define cómo coordinarlos. Java EE 5 también configura algunas especificaciones que incluyen Enterprise JavaBeans, servlets, portlets, JavaServer Pages y varias tecnologías de servicios Web, además de permitir desarrollar y ejecutar aplicaciones en lenguaje de programación Java con arquitectura de N niveles distribuidos. (30)

Esto permite al desarrollador crear una aplicación de empresa portable entre plataformas y escalable, a la vez que se integra con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor

Capítulo 1: *Fundamentación teórica*

de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

1.3.9.2 JRE 6

El Java Runtime Environment (JRE) está destinado a desarrolladores de software y proveedores para la distribución de sus aplicaciones. Este contiene la máquina virtual Java, las bibliotecas de clases en tiempo de ejecución, y el lanzador de aplicaciones Java que son necesarios para ejecutar los programas escritos en este lenguaje de programación. Está compuesto además por las librerías de clases estándar que implementan las API de Java. (31)

1.3.10 Herramientas

1.3.10.1 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto y multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos para proporcionar toda su funcionalidad al frente de la plataforma de los llamados clientes ricos. Esto lo diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software, que adicionalmente permite al Eclipse extenderse, usando otros lenguajes de programación como C/C++, Python y Java. (32)

1.3.10.2 JBoss Tools

Es un conjunto de herramientas para Eclipse que permiten el manejo de diferentes frameworks que facilitan el desarrollo de aplicaciones. JBoss Tools dispone de plugins que proporcionan soporte en Eclipse para Hibernate, JBoss AS, Drools, JBPM, JSF, XHTML, Seam, Smooks, JBoss ESB o JBoss Portal, entre otros. (33)

Capítulo 1: Fundamentación teórica

1.3.10.3 JBoss Server

JBoss Application Server es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado, por ser una plataforma certificada JEE, incluyendo los servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la Web. También soporta EJB 3, lo que proporciona que el desarrollo de las aplicaciones sea mucho más simple. (34)

1.3.11 Herramienta de modelado

1.3.11.1 Visual Paradigm para UML

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. 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. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (35)

1.3.11.2 Proceso Unificado de Desarrollo (RUP)

Es una metodología flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. La misma está basada en las mejores prácticas adoptadas en miles de proyectos en todo el mundo. Evita tener que crear desde cero y favorece la reutilización de los procesos que han tenido éxito para otras organizaciones. (36)

Ofrece una capacidad de los patrones que permiten a los administradores de proyectos rápidamente agregar o quitar trozos reutilizables de los procesos de solución a problemas comunes. Debido a que no hay dos proyectos iguales, permite a los jefes de proyecto modificar el proceso para atender las necesidades de proyectos específicos. (37)

La elección de RUP conlleva a un entorno de desarrollo limpio y de calidad basado en sus 6 principios de desarrollo:

- Adaptar el proceso.
- Equilibrar prioridades.
- Demostrar valor iterativamente.

Capítulo 1: Fundamentación teórica

- Colaboración entre equipos.
- Elevar el nivel de abstracción.
- Enfocarse en la calidad.

1.3.11.3 Notación para el Modelado de Procesos del Negocio (BPMN)

Business Process Modeling Notation (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. La misma proporciona un lenguaje común para que las partes involucradas en el proyecto puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma, BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (BPD). (38)

Basado en una técnica de diagramas de flujo muy similar a los diagramas de actividad de UML. El objetivo principal de BPMN es apoyar la gestión de procesos de negocio tanto para usuarios técnicos como para los usuarios de negocio. (39)

1.3.11.4 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje de modelado para especificar o para describir métodos y procesos. Se utiliza para definir un sistema, para detallar los artefactos en el mismo y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Este se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software tal como RUP, pero no especifica en sí mismo qué metodología o proceso utilizar. (40)

Este lenguaje permite mejorar el soporte a la planeación y el control del proyecto, por consiguiente, posibilita mejorar los tiempos totales de desarrollo. Además, logra encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica y una alta reutilización y minimización de costos.

Con el estudio del presente capítulo se describieron los diferentes conceptos relacionados con la facturación de los servicios médicos. Se llegó a la conclusión de que los sistemas analizados que implementan módulos de facturación o semejantes, no cumplen con las expectativas necesarias pues no son basados en sistemas libres ni multiplataforma.

Para resolver estas dificultades se estudiaron las tecnologías que pueden hacer posible la creación de un Módulo de Facturación del Sistema de Información Hospitalaria alas HIS. Por lo que se decidió utilizar

Capítulo 1: Fundamentación teórica

Java, junto al Sistema Gestor de Base de Datos PostgreSQL. Visual Paradigm como herramienta de modelado y RUP como metodología de desarrollo de software. Estos serán la base de conocimientos necesarios para llevar a cabo la elaboración del módulo.

Capítulo 2: Características del sistema

Fundamentada la problemática se continúa describiendo el trabajo realizado en la facturación desde la prestación médica hasta algunos de sus procesos específicos. El presente capítulo tiene como objetivo describir los principales procesos implicados, así como sus actores, el modelo de negocio y el sistema, plasmando los principales requerimientos tanto funcionales como no funcionales y llegando a una propuesta de solución explicada en términos de casos de uso.

2.1 Objeto de estudio

2.1.1 Flujo actual de los procesos involucrados en el campo de acción

Dentro de una institución hospitalaria el área de facturación es la encargada de gestionar la información de un paciente en términos de contabilidad, a partir de la llegada del mismo al hospital, son registrados los servicios, prestaciones y medios consumidos por él. Estas acciones son agrupadas en procesos mediante los cuales los especialistas de facturación son los encargados de recolectar y registrar la información de las prestaciones ofrecidas por la institución para una posterior facturación.

El proceso de facturación de prestaciones ambulatorias dirige y procesa toda la información de un paciente referente a los servicios brindados al mismo siempre y cuando sean ambulatorios. Se puede tener referencia de los servicios de emergencias, específicamente exámenes, cirugías, transfusiones hasta la posible autopsia al paciente.

El proceso de facturación de internados es dirigido por el especialista encargado de facturación, el mismo recibe y procesa la información de un paciente desde el momento en que es ingresado en la institución hasta su posterior egreso.

Estos servicios prestados al paciente pueden ser evaluados contablemente de dos formas, una de ellas es por lista de precios pudiendo seleccionar independientemente cada uno de las prestaciones por separado y la otra forma sería agrupando por módulo de prestaciones anteriormente definidas por algún convenio previo con una compañía de seguros u Obra Social.

“Luego de registrar toda la información posible del paciente atendido, las prestaciones son enlazadas con un precio ya establecido, de esta manera, se emite una factura final.” (41)

Capítulo 2: Características del sistema

Estos procesos culminan con el envío de la hoja de facturación a la caja del hospital para su posterior aprobación, pago y archivo.

El especialista de facturación es el principal encargado de registrar los convenios entre Obras Sociales y la entidad hospitalaria, este proceso comienza cuando la Obra Social decide establecer en el hospital un grupo de prestaciones o una agrupación de estas, las cuales serán sufragadas por la Obra Social cuando sea atendido un paciente que mantenga un contrato o seguro con dicha Obra Social. Cuando la Obra Social es creada dentro de la institución se crea una afiliación Obra Social - Institución Hospitalaria.

El paciente dentro de la institución tiene la posibilidad de hacer una petición de un presupuesto, el cual semejante a una factura agrupa las prestaciones que en un futuro puede recibir el paciente, estableciendo todos los parámetros para la misma, dígase paciente solicitante del presupuesto, posible pagador de la misma, y el listado ó módulo de prestaciones. El especialista en facturación es el encargado directo de este proceso.

Registrar abonos se encuentra también dentro de los procesos que son manejados directamente por el especialista en facturación. El paciente paga un monto establecido por él y es registrado para un posterior uso en el pago de alguna factura.

El proceso de cierre de caja periódico se encuentra subordinado directamente al especialista de facturación específicamente en el encargado de caja, el cual al finalizar el período establecido para los cambios de turnos comprueba la cantidad de efectivo en dependencia de las facturas que han sido cobradas. Al final de este proceso es generado un reporte de cierre de caja dejando plasmado los detalles del cierre en ese período o fecha.

El encargado de caja es el que debe emitir los reportes referentes a las prestaciones y facturas realizadas en la institución hospitalaria, con el objeto de obtener los datos necesarios para realizar posteriormente una correcta planificación y predicción financiera.

2.2 Información que se maneja

Una vez analizado y descrito el proceso que sigue una factura, se definen las actividades a automatizar y se realiza un análisis donde se establecen las funcionalidades del sistema. Una vez que estos procesos sean automatizados se está garantizando la eficiencia en el trabajo de la facturación dentro de las instituciones hospitalarias. Para esto se propone implementar un sistema capaz de manejar la información referente a la facturación de una forma eficiente, rápida y poco compleja.

Capítulo 2: Características del sistema

“La creación del módulo de Facturación del Sistema de Información Hospitalaria alas HIS, facilitará la gestión a través de facturas electrónicas, la generación de reportes, la automatización de los cálculos, así como el procesamiento de la información requerida.” (42)

Se toma como centro de referencia la factura, que no es más que el elemento legal que se emite luego de finalizada la atención médica. La misma contiene datos como: número de factura, fecha de expedición, nombre del paciente, dirección domiciliaria, carné de identidad, datos del emisor, prestaciones del paciente, precio de las prestaciones, tipo impositivo del Impuesto del Valor Agregado (IVA) y fecha de prestación del servicio. La factura se verá asociada directamente a la Hoja de factura.

La Hoja de factura contiene toda la información personal del paciente obtenida de su Historia Clínica, nombre, apellidos, carné de identidad, dirección domiciliaria. Recoge también, la información de las distintas prestaciones y servicios ofrecidos al paciente durante su estancia en el hospital. Las prestaciones estarán vinculadas a un listado de precio que define el precio para cada una de ellas, con el objetivo de la facturación de las prestaciones del paciente.

El registro de Obra Social es el encargado de almacenar los datos de una obra social y la fecha de la afiliación con la institución hospitalaria. Se encuentra relacionado directamente con el registro de convenios, el cual almacena los datos referentes a la obra social convenida, las prestaciones que estarían regidas bajo ese convenio, ya sea en listado de precios ó módulos de prestaciones que agrupan una serie de prestaciones. También puede establecerse en los convenios el por ciento de aseguramiento para el pago de las prestaciones.

Los presupuestos muy parecidos a las hojas de factura almacenan toda la información referente al paciente y los servicios o prestaciones a las cuales puede llegar a ser sometido en un futuro el paciente. El objetivo del presupuesto viene dado por la necesidad del paciente de conocer cuánto puede costarle una serie de tratamientos, prestaciones o servicios en la institución hospitalaria. Los presupuestos pueden llegar a ser convertidos en hojas de facturas, facilitando la gestión de las mismas.

Los abonos son depósitos de efectivo por parte de un paciente para posteriormente ser utilizado en el pago de facturas. Los mismos pueden contener como datos el monto o cantidad de dinero abonada por el paciente, el número de la cuenta del paciente asociada a ese abono y la fecha de registro del mismo. Estos abonos serán seguidos mediante un registro de uso de los mismos, guardando la información de su utilización, dígame fecha en la que fue utilizado y la cantidad de dinero utilizada en esa oportunidad.

Capítulo 2: Características del sistema

Los registros de cierre de caja tienen como tarea fundamental almacenar la información referente al período entre cambios de turnos en los cuales se contabilizan las facturas emitidas, pagadas y pendientes. Estos registros pueden ser aprobados una vez que se comprueba la cantidad de dinero en caja y la cantidad de dinero en el sistema validando así su aprobación o su desaprobación justificada.

Otras entidades objeto de automatización serían: Reporte de cirugía, contiene la información de las cirugías realizadas, si las mismas han sido emergencias o no, y si el resultado de las mismas ha sido el esperado; Reporte de transfusiones, permite determinar si el paciente se realizó una transfusión y obtener los datos de la misma; Hoja de emergencia, permite conocer de un paciente motivo de consulta, nivel de gravedad y datos personales del acompañante.

También se pueden ver: Indicaciones del médico, permite obtener la información de los servicios ofrecidos al paciente; Reporte de exámenes de anatomía patológica, permite conocer los exámenes patológicos a los que ha sido sometido el paciente así como las especificidades de estos; Stock de enfermería, permite conocer los datos de las indicaciones médicas, el servicio al que pertenece, fecha y hora del cumplimiento del tratamiento. En este último se obtendrán similarmente los medicamentos indicados, dosis entre otros datos de los mismos.

En caso de fallecer el paciente se podrán obtener los detalles del fallecimiento y de la autopsia en caso de haber sido solicitada, el mismo es el Reporte de defunción, contiene fecha y hora de la muerte, número de la autopsia, médico de guardia, persona que autoriza y parentesco.

2.3 Modelado del negocio

Para un mejor entendimiento del funcionamiento de la institución donde será implantado el producto se realiza el flujo de trabajo Modelado del negocio. En análisis exhaustivo de este capítulo es fundamental, pues es aquí donde coexiste el marco común de entendimiento entre los desarrolladores, los clientes y el usuario final.

Para realizar el modelado de los procesos fue usada la notación Business Process Modeling Notation (BPMN), esta notación de modelado garantiza una mejor visibilidad de las actividades que se realizan y ayuda a lograr un mejor entendimiento del flujo de trabajo existente entre las áreas hospitalarias, lo que permite definir con claridad las actividades innecesarias a la hora de automatizar el negocio. (43)

Capítulo 2: Características del sistema

2.3.1 Definición de roles

Un trabajador del negocio representa el rol que juega alguien en específico o algo dentro del negocio y que a la vez realiza alguna actividad dentro del mismo. Un trabajador del negocio colabora con otros trabajadores del negocio, se notifica de acontecimientos del negocio y manipula entidades de negocio para realizar sus responsabilidades.

Trabajadores del Negocio	
Rol	Funcionalidad
Encargado de la facturación ambulatoria	Rol encargado de registrar todos los servicios ofrecidos a un paciente durante la atención ambulatoria, para la posterior valuación de precios y pre-facturación.
Encargado de la facturación en internados	Rol encargado de registrar todos los servicios ofrecidos al paciente a partir del ingreso de este a la institución, hasta su posterior egreso, para realizar la valuación de precios y pre-facturación.
Encargado de caja	Rol encargado de aprobar o anular la factura realizada, luego de la revisión de la misma. Gestiona todas las facturas de la institución, y es responsable por la generación de todos los reportes necesarios para esta.

Tabla 2. Trabajadores del negocio.

2.3.2 Diagramas de procesos del negocio

La Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en ingles) es una notación gráfica común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y procesos públicos o privados, así como conceptos avanzados de modelado. Esta notación permite construir diagramas de procesos de una forma más sencilla y mucho más eficiente.

Capítulo 2: Características del sistema

Dentro de la misma se encuentran eventos que definen inicio y fin de una acción lógica en el flujo de los procesos o subprocesos. Las actividades se relacionan entre sí por un flujo secuencial. Las bifurcaciones determinan la ramificación del flujo a través de decisiones. Y los artefactos definen los documentos o registros que son creados, modificados, enviados o recibidos entre las actividades.

Capítulo 2: Características del sistema

Diagrama del proceso de negocio “Procesos de facturación”

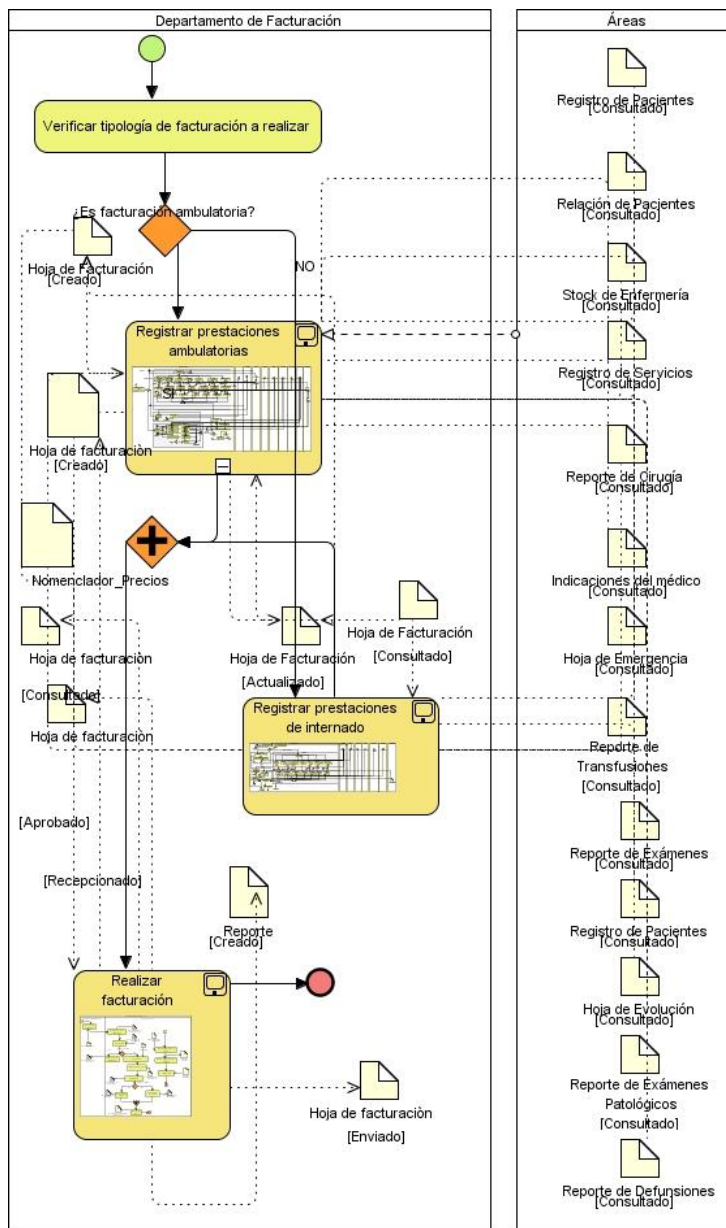


Figura 1. Diagrama del proceso del negocio “Procesos de facturación”.

Capítulo 2: Características del sistema

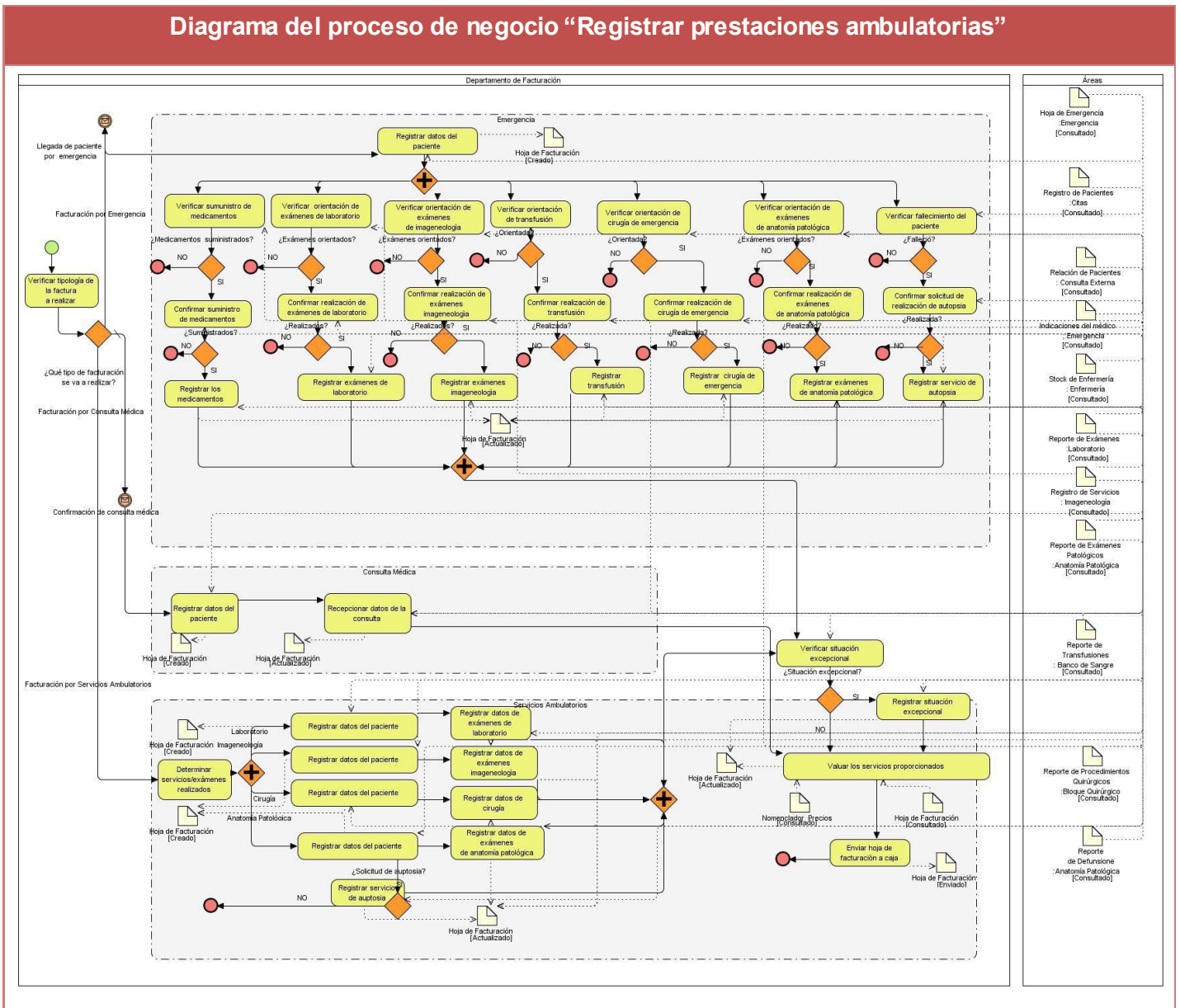


Figura 2. Diagrama del proceso del negocio "Registrar prestaciones ambulatorias".

Capítulo 2: Características del sistema

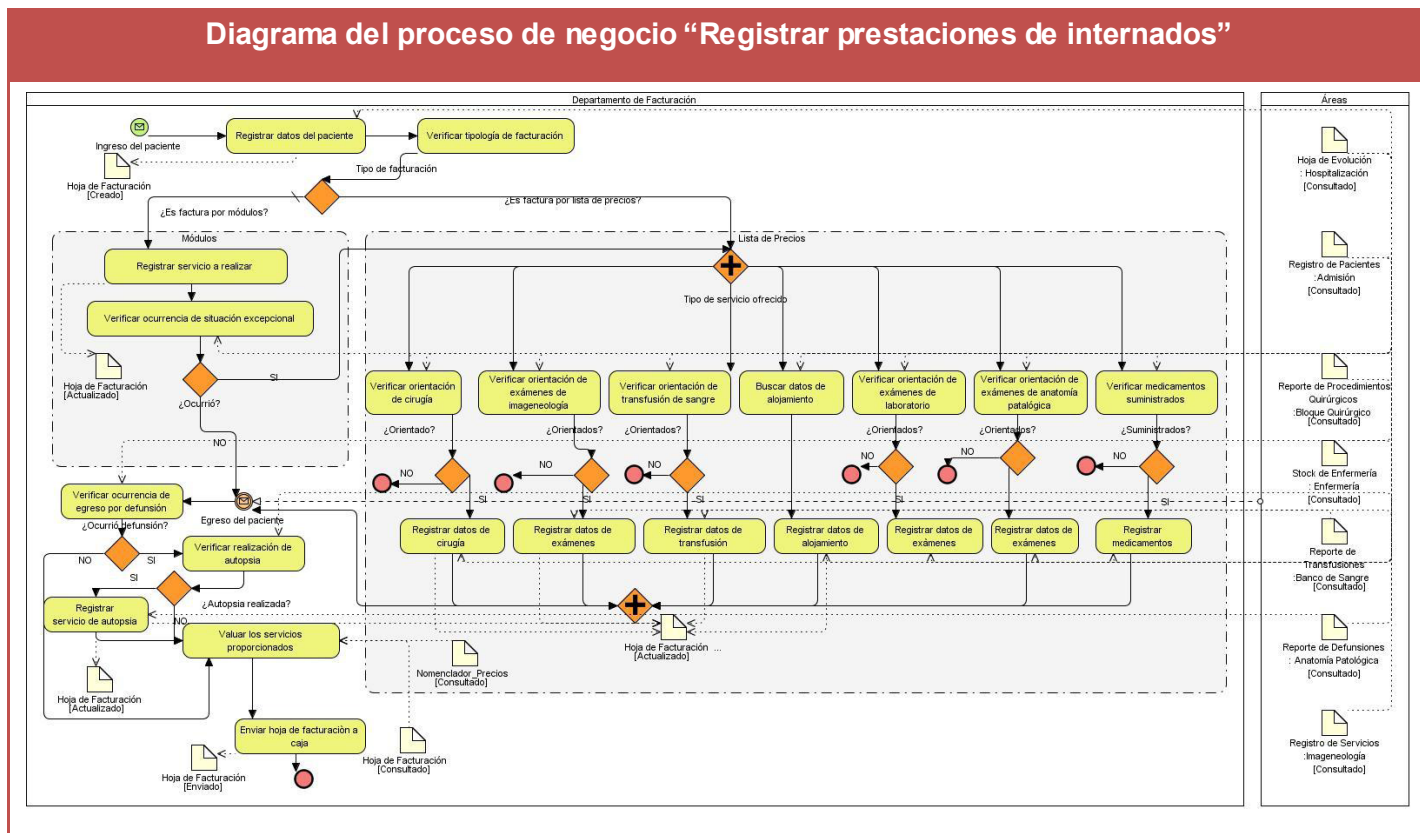


Figura 3. Diagrama del proceso del negocio “Registrar prestaciones de internados”.

Capítulo 2: Características del sistema

Diagrama del proceso de negocio "Realizar facturación"

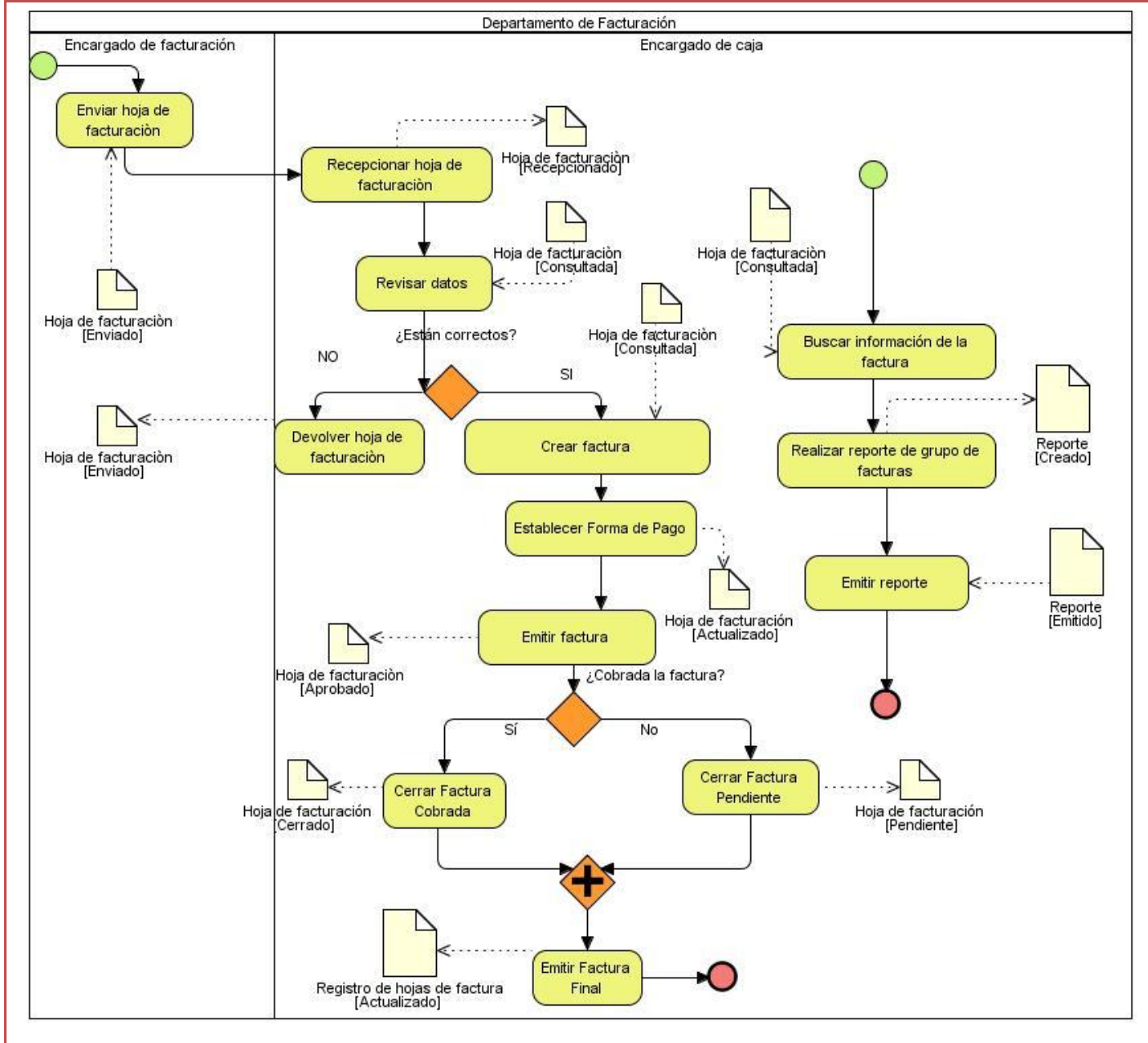


Figura 4. Diagrama del proceso del negocio "Realizar facturación".

Capítulo 2: Características del sistema

Diagrama del proceso de negocio "Registrar presupuesto"

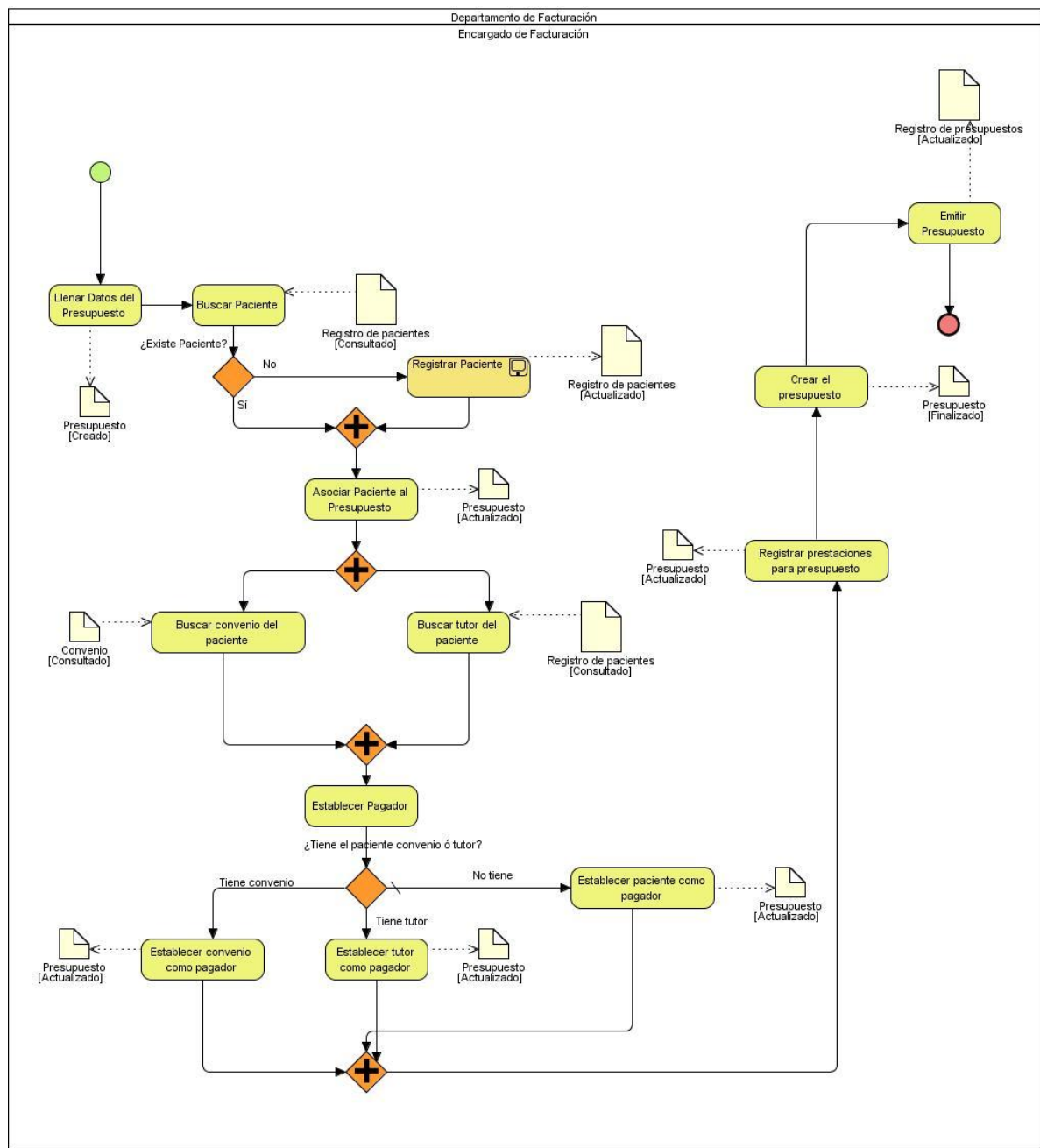


Figura 5. Diagrama del proceso del negocio "Registrar presupuesto".

Capítulo 2: Características del sistema

Diagrama del proceso de negocio “Cierre de caja”

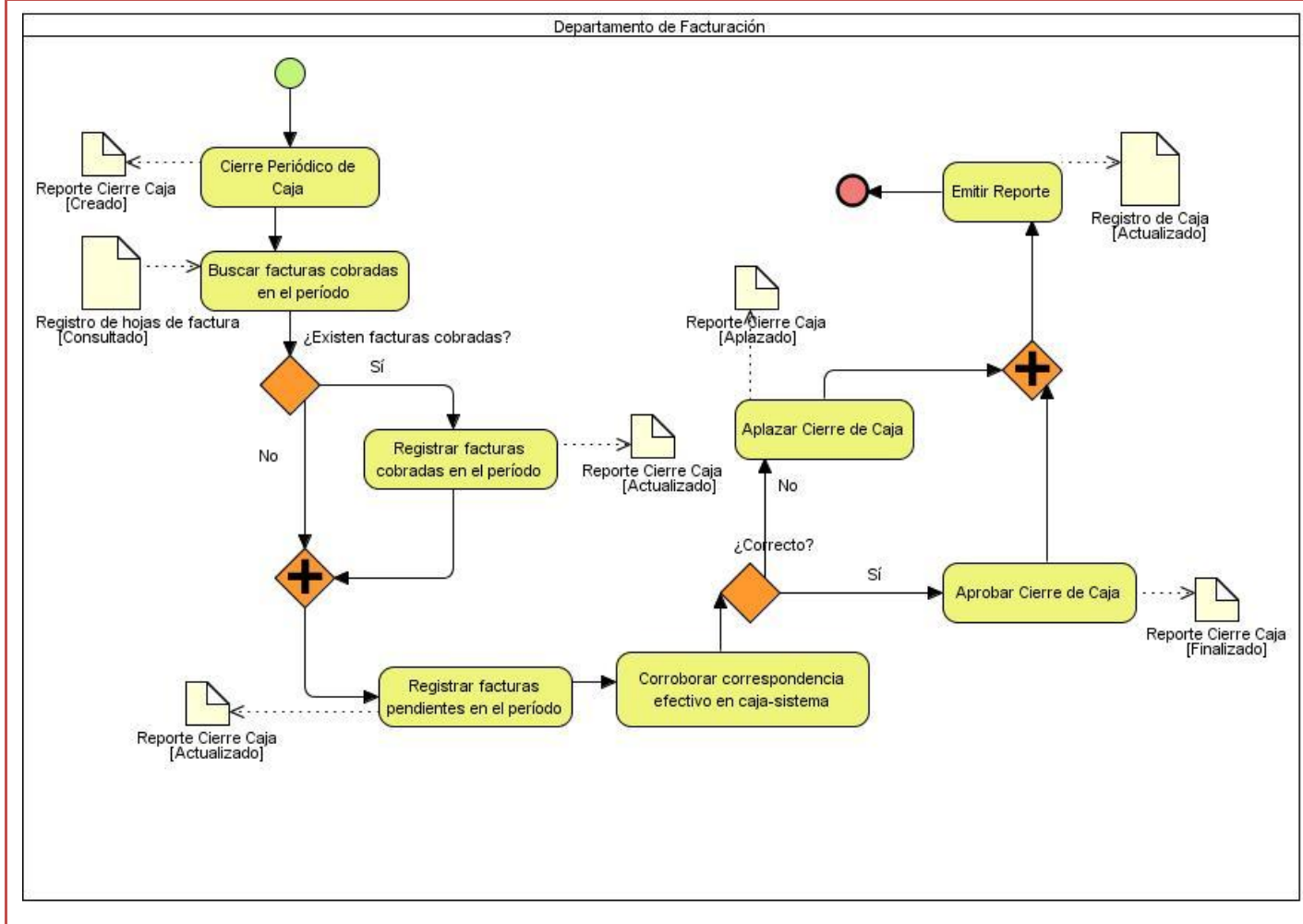


Figura 6. Diagrama del proceso del negocio “Cierre de caja”.

2.4 Especificación de los requisitos del sistema

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El conjunto de todas las necesidades es el fundamento para el consiguiente desarrollo del sistema o componente. (44)

Capítulo 2: Características del sistema

2.4.1. Requisitos funcionales

RF1 Crear forma pago	RF35 Ver datos de cierre de caja
RF2 Ver detalles forma de pago	RF36 Exportar cierre de caja
RF3 Modificar forma de pago	RF37 Crear presupuesto
RF4 Eliminar forma de pago	RF38 Ver detalles presupuesto
RF5 Buscar forma de pago	RF39 Modificar presupuesto
RF6 Crear convenio	RF40 Eliminar presupuesto
RF7 Ver detalles de convenio	RF41 Buscar presupuesto
RF8 Modificar convenio	RF42 Ver dato presupuesto
RF9 Eliminar convenio	RF43 Exportar presupuesto
RF10 Buscar convenio	RF44 Crear factura
RF11 Ver datos del convenio	RF45 Establecer forma de pago de la factura
RF12 Seleccionar listado de prestaciones del convenio	RF46 Ver detalles factura
RF13 Seleccionar módulos de prestaciones convenio	RF47 Modificar factura
RF14 Exportar datos del convenio	RF48 Eliminar factura
RF15 Crear obra social	RF49 Buscar factura
RF16 Ver detalles obra social	RF50 Ver datos factura
RF17 Modificar obra social	RF51 Exportar factura
RF18 Eliminar obra social	RF52 Establecer estado de emisión de una factura
RF19 Buscar una obra social	RF53 Buscar las prestaciones ofrecidas al paciente
RF20 Ver datos obra social	RF54 Buscar otros consumos ofrecidos al paciente
RF21 Exportar datos obra social	RF55 Registrar las prestaciones
RF22 Seleccionar paciente	RF56 Buscar el precio de las prestaciones ofrecidas al paciente
	RF57 Crear moneda de trabajo

Capítulo 2: Características del sistema

RF23 Crear abono	RF58 Ver detalles de una moneda de trabajo
RF24 Ver detalles abonos	RF59 Modificar moneda de trabajo
RF25 Modificar abono	RF60 Eliminar moneda de trabajo
RF26 Eliminar abono	RF61 Buscar moneda de trabajo
RF27 Buscar abono	RF62 Establecer moneda de trabajo
RF28 Crear cierre de caja	RF63 Modificar IVA
RF29 Ver detalles cierre de caja	RF64 Ver detalles de IVA
RF30 Modificar cierre de caja	RF65 Valuar las prestaciones
RF31 Eliminar cierre de caja	RF66 Editar prestaciones
RF32 Buscar cierre de caja	RF67 Registrar otros consumos ofrecidos al paciente
RF33 Buscar facturas cobradas y por cobrar en el periodo	RF68 Generar prestaciones
RF34 Registrar efectivo en caja y dinero electrónico acumulado en el periodo	RF69 Generar prestaciones por área

Tabla 3. Requisitos funcionales.

2.4.2 Requisitos no funcionales

2.4.2.1 Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido. Para los usuarios que requieren de un dominio básico del sistema se propone como máximo un período de aprendizaje de 20 días; mientras que, para un dominio avanzado se propone un período máximo de 30 días.

2.4.2.2 Seguridad

- Se mantendrá seguridad y control a nivel de usuario, que garantice el acceso de los mismos sólo a los niveles establecidos de acuerdo con la función que realizan.
- Las contraseñas podrán cambiarse sólo por el propio usuario o por el administrador del sistema.

Capítulo 2: Características del sistema

- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la base de datos, independientemente de que para el sistema, este elemento ya no exista.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvos realizadas.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- El sistema implementará un control de cambios a determinados campos de información (seleccionados por su importancia), de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.

2.4.2.3 Rendimiento

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

Además, el mismo respetará las buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual de Java como la creación de objetos.

Se deberá usar siempre que sea posible el patrón Singleton para destruir referencias que ya no sean usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

2.4.2.4 Soporte

Seguridad de acceso y administración de usuarios

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

Capítulo 2: Características del sistema

Monitoreo de funcionamiento

Se permitirá la 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, logrando esto mediante un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información sobre el registro modificado.

Configuración de parámetros

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

Réplica

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.

2.4.2.5 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 IE8, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0GHz.

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.

Capítulo 2: Características del sistema

- 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 2GB de memoria y 2x72GB de disco y sistema operativo Linux.

2.4.2.6 Software

El sistema debe correr 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 8, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

2.4.2.7 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.

2.4.2.8 Requisitos para la documentación de usuarios en línea y ayuda del sistema

Se posibilitará el uso de ayudas dinámicas y tutoriales online sobre el funcionamiento del sistema.

2.4.2.9 Requisitos de interfaz

Interfaces de usuarios

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con menús desplegables que faciliten y aceleren su utilización. La entrada incorrecta de datos será detectada e informada al usuario.

Capítulo 2: Características del sistema

Interfaces de comunicación

Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven). El sistema usará el formato estándar WSDL (Web Services Description Language) para la descripción de los servicios Web. El sistema implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos. El sistema utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

2.4.2.10 Portabilidad

El producto podrá ser utilizado bajo los sistemas operativos Linux, Unix o Windows.

2.4.3 Modelo de casos de uso del sistema

2.4.3.1 Definición de actores

Actores del sistema	
Actor del sistema	Funcionalidad
Usuario	Usuario global que permite la autenticación en el sistema y que valida al mismo dándole un rol.
Encargado de facturación	Rol encargado de interactuar con el sistema y de gestionar todas las facturas de la institución, es responsable de la generación de todos los reportes necesarios. Además, se encarga de configurar y gestionar todas las prestaciones y precios de estas.

Tabla 4. Actores del sistema

Capítulo 2: Características del sistema

2.4.3.2 Vista global de actores del sistema

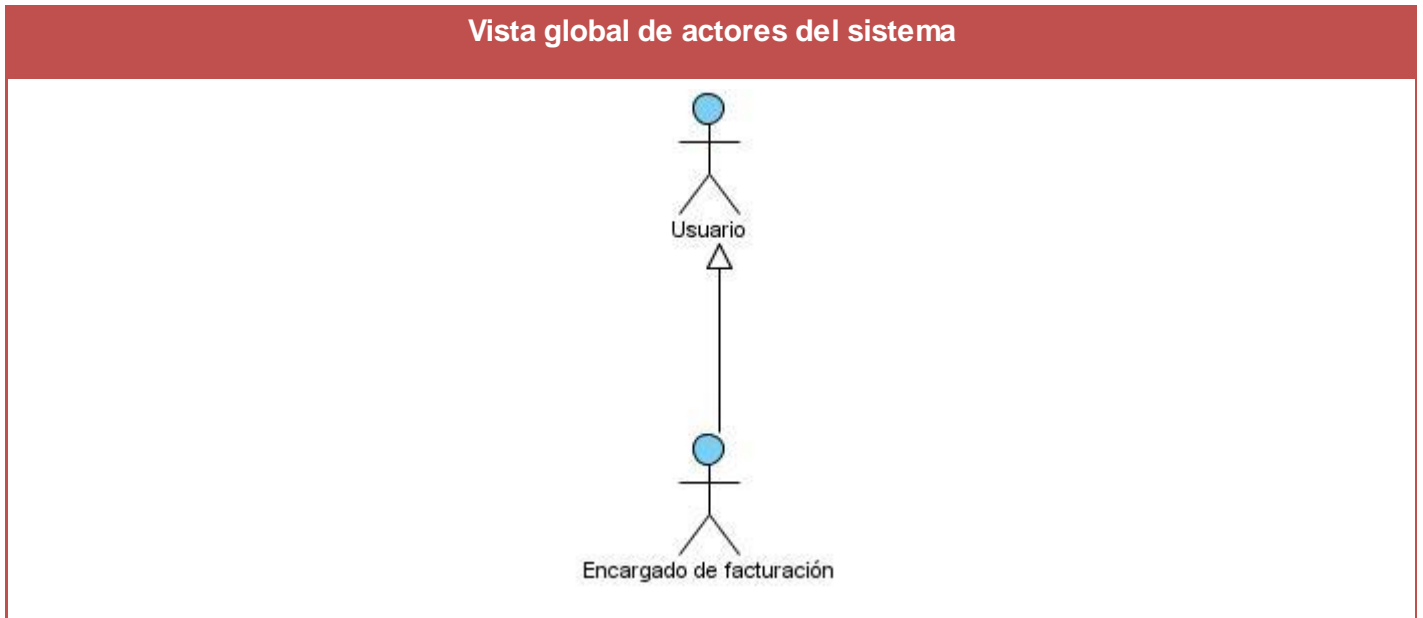


Figura 7. Actores del sistema

Capítulo 2: Características del sistema

2.4.3.3 Diagrama de casos de uso del sistema

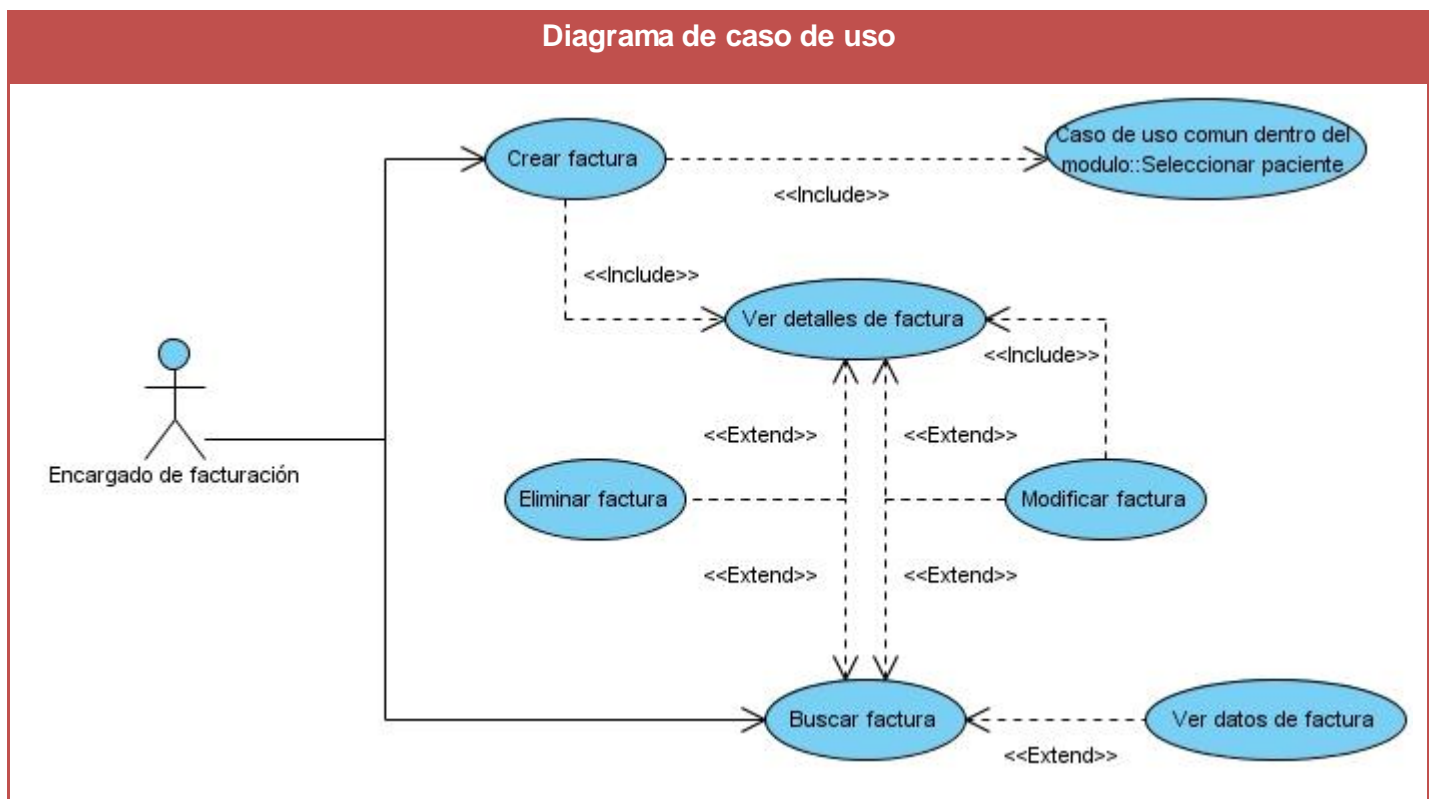


Figura 8. Diagrama de caso de uso del sistema "Crear factura".

Capítulo 2: Características del sistema

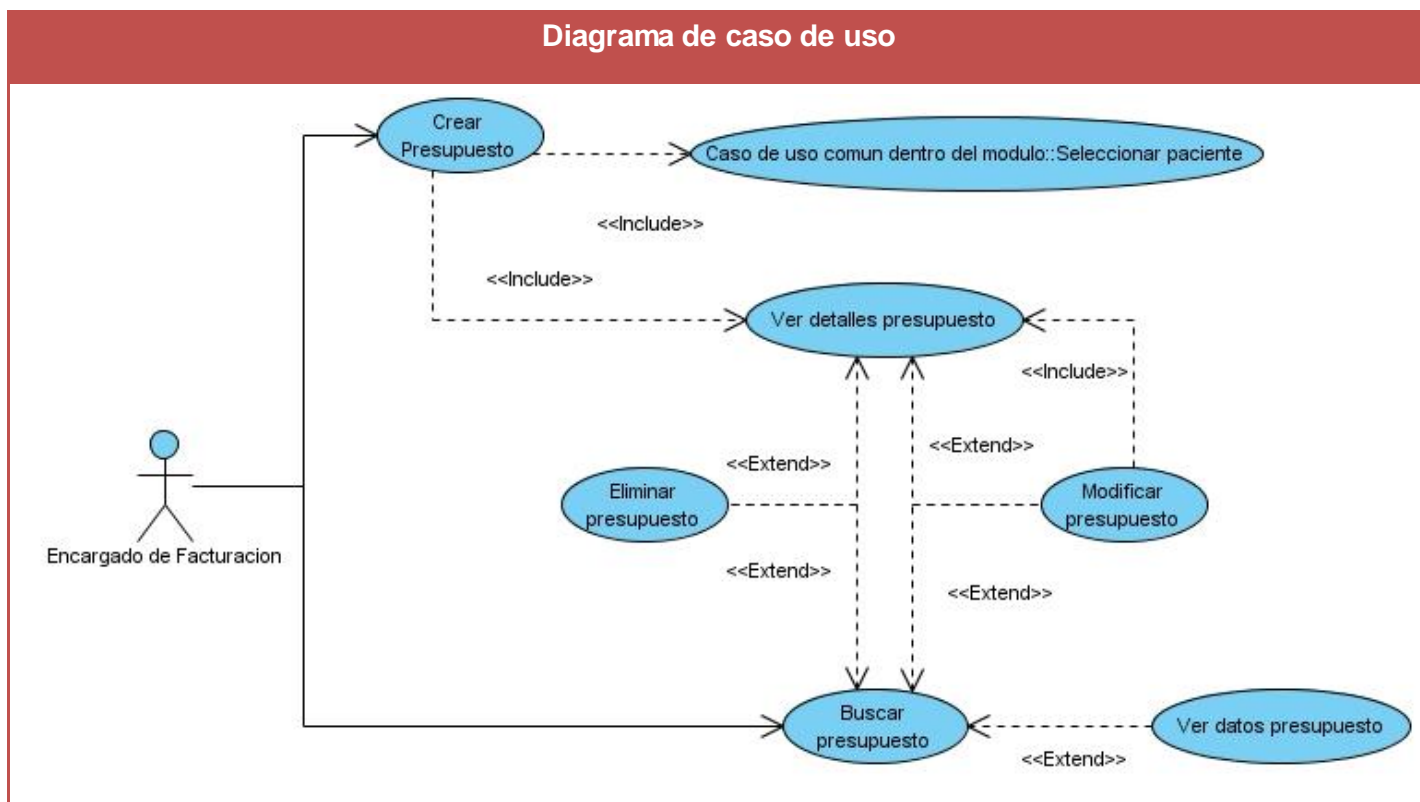


Figura 9. Diagrama de caso de uso del sistema "Crear presupuesto".

2.4.4 Descripción textual de los casos de uso arquitectónicamente significativos

2.4.4.1 Crear factura

CASO DE USO:	Crear factura
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear factura, el sistema brinda la posibilidad de introducir los datos para crear la factura, el actor introduce los datos de la factura, el sistema crea la factura, el caso de uso termina.
Complejidad:	Alta
Prioridad:	Crítico
Precondiciones:	El paciente debe tener registradas prestaciones o servicios que le fueron brindados en la institución.

Capítulo 2: Características del sistema

REFERENCIAS	
Actores:	Encargado de facturación
Requisitos:	RF22, RF44, RF45, RF52, RF53, RF54, RF55, RF56
Entidades:	Factura
Casos de Uso:	Ver detalles de factura Caso de uso común dentro del módulo :: Seleccionar paciente

Tabla 5. Descripción textual del caso de uso "Crear factura".

2.4.4.2 Crear presupuesto

CASO DE USO:	Crear presupuesto
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear presupuesto, el sistema brinda la posibilidad de introducir los datos para crear el presupuesto, el actor introduce los datos del presupuesto, el sistema crea el presupuesto, el caso de uso termina.
Complejidad:	Media
Prioridad:	Crítico
Precondiciones:	No existe.
REFERENCIAS	
Actores:	Encargado de facturación
Requisitos:	RF22, RF37
Entidades:	Presupuesto
Casos de Uso:	Ver detalles de presupuesto Caso de uso común dentro del módulo :: Seleccionar paciente

Tabla 6. Descripción textual del caso de uso "Crear presupuesto".

Capítulo 2: Características del sistema

2.4.4.3 Crear convenio

CASO DE USO:	Crear convenio
Resumen:	El caso de uso inicia cuando el actor accede a la opción Crear convenio, el sistema brinda la posibilidad de introducir los datos para crear el convenio, el actor introduce los datos del convenio, el sistema crea el convenio, el caso de uso termina.
Complejidad:	Media
Prioridad:	Crítico
Precondiciones:	No existe.
REFERENCIAS	
Actores:	Encargado de facturación
Requisitos:	RF6, RF12, RF13, RF55
Entidades:	Convenio
Casos de Uso:	Ver detalles de convenio

Tabla 7. Descripción textual del caso de uso "Crear convenio".

En este capítulo, luego de analizar los diferentes procesos del negocio dentro de la facturación de servicios médicos, se describieron los modelos del negocio que intervienen en la gestión de la información y el flujo de los procesos dentro de la facturación, llegando así a las propuestas de requisitos funcionales. Las mismas serán capaces de ofrecer al Módulo de Facturación características fundamentales para su correcto funcionamiento. También se describieron los requisitos no funcionales necesarios para que la aplicación no presente problemas técnicos de referente a las tecnologías utilizadas. Y por último se describieron los diagramas de casos de uso, así como los casos de uso arquitectónicamente significativos, dando una idea más clara de las características de la aplicación.

Capítulo 3: Diseño

En el capítulo anterior se tratan en detalles las características del sistema, describiendo los principales procesos involucrados en la facturación de servicios médicos. También se define el modelo de negocio, así como el sistema, haciendo énfasis principalmente en los requerimientos funcionales. En el presente capítulo se describe de forma minuciosa el sistema propuesto, se explica la arquitectura definida para el Sistema de Información Hospitalaria alas HIS y finalmente se muestran los principales diagramas de clases y secuencias del diseño que contribuyen futuramente a la implementación del sistema.

3.1 Descripción de la arquitectura

“La arquitectura es la eje central para sistemas de alta complejidad, masivos a gran escala y altamente interoperativos que necesitamos ahora y en el futuro”

Rolf Siegers, Raytheon

La arquitectura es el principal indicador para las cualidades de un sistema, tanto su rendimiento, su maleabilidad, así como la seguridad del mismo no pueden ser determinadas sin una visión unificada de la arquitectura. La arquitectura sirve como línea que une el sistema y quienes lo desarrollan, definiendo que tareas son ejecutadas bajo el diseño y la implementación; provee de un análisis temprano asegurando que el diseño propuesto del sistema genere el rendimiento esperado.

Para el desarrollo del Módulo de Facturación en el Capítulo 1 se proponen las herramientas, componentes, metodologías y tecnologías a utilizar. Teniendo en cuenta esto se define como patrón de diseño Modelo-Vista-Controlador (MVC), como base de la Arquitectura. Este paradigma en la programación es muy utilizado en aplicaciones (como es el caso del presente módulo) Web. Como se detalla en el capítulo 1 este patrón permite la separación de la lógica del negocio, del modelo y la presentación, cada uno cumpliendo tareas específicas, proporcionando un sistema más flexible y potente.

El framework JSF permite entre sus facilidades la implantación del patrón seleccionado, brindando la posibilidad de separar de forma clara y precisa la información que se muestra al usuario de las acciones que el mismo ejecuta. La capa de presentación se encuentra conformada por páginas XHTML, las cuales, integradas por formularios que contienen controles tanto del propio JSF, como de Seam UI y RichFaces que controlan y validan los datos entrados por el usuario. También se enriquece el envío de datos y la

comunicación de forma asincrónica con el modelo mediante la utilización de AJAX4JSF, logrando efectos mucho más agradables y naturales al intercambiar con el sistema.

La capa del negocio se encuentra constituida por clases controladoras que definen la lógica del negocio del módulo, así como el manejo y validación de los datos capturados en la capa de presentación. Mediante las anotaciones que Seam provee, se pueden declarar estas clases en dependencia del contexto en que se encuentren, ya sea conversacional, de tipo pagina, evento e inclusive de sesión, permitiendo definir el estado de los datos y las entidades que manejan.

Por último, la capa de acceso a datos es la encargada de cargar, modificar, eliminar y persistir la información en la base de datos ya sea una vez validada. Mediante el uso del ORM Hibernate el cual permite el mapeo de las tablas en la base de datos como clases, facilitando en gran medida las acciones sobre la misma.

Todas estas tecnologías se encuentran integradas al poderoso y moderno framework que es Seam, el cual unifica las mencionadas tecnologías en un solo ambiente, logrando una vía más fácil para el desarrollo de aplicaciones empresariales complejas. Actualmente Seam se encuentra dentro de las aplicaciones de más alta categoría en la Java EE 5 y entre los frameworks más populares y más usados para Java.

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

La facturación de servicios médicos como su nombre lo indica es el complemento que facilita el cobro de las prestaciones médicas que son registradas una vez a un paciente en particular. El Sistema de Información Hospitalaria alas HIS provee los datos de las prestaciones médicas que le asisten a un paciente registrándose así la información referente a los servicios que se brindan en la institución hospitalaria. Lo más útil sería lograr una integración, en la cual cada una de las áreas que proveen facturación o cobro de prestaciones registre los datos referentes al servicio prestado y al paciente en cuestión. Es de suma importancia aclarar que el propio Módulo de Facturación puede ser capaz de registrar a un paciente las prestaciones o servicios que lo asisten dentro del hospital, pero no sería ni la forma más eficiente, ni la más cómoda de resolver la situación.

Para lograr lo anteriormente dicho, el Módulo de Facturación provee la clase genérica RegistrarPrestación, mediante la cual, los módulos que brindan servicios a facturar pueden registrar de forma simple la prestación ofrecida a un paciente.

Para gestionar las prestaciones dentro de la institución serán utilizadas las tablas que contienen información referente a los servicios de las áreas de Bloque Quirúrgico, Anatomía Patológica, Consulta Externa, Laboratorio, y también las tablas de tratamientos y componentes sanguíneos de Banco de Sangre. Este último no provee servicios activos, por lo que se propone el uso de la tabla de servicios de Banco de Sangre, para una mejor distribución de las prestaciones brindadas para la facturación. También se utiliza la tabla de servicios clínicos para determinar la estancia de un paciente en el hospital, haciendo referencia al área de Hospitalización.

El Módulo de Facturación, para el funcionamiento de varias de sus características principales hace uso de una de las funcionalidades utilizadas en el Módulo de Admisión, la cual, presente en forma de reutilización de código facilita las búsquedas de Historias Clínicas de pacientes, así como, ver los detalles de las mismas.

Para el control de las trazas de las acciones llevadas a cabo por los usuarios que interactúan con el sistema se maneja la clase Bitácora, haciendo uso de la clase IActiveModule para la obtención de los datos referentes al módulo y entidad en la cual se encuentra trabajando el usuario. También se hace uso de la clase User, la cual ofrece los datos del usuario que se encuentra autenticado, entre otras funcionalidades.

3.3 Modelo de diseño

En el capítulo anterior se definen los requisitos funcionales que debe cumplir el módulo, expresados en casos de uso. A continuación se hace una descripción detallada que fija el punto de partida para dar solución a dichos requisitos, pero ya desde una perspectiva mucho más cerca de la solución final. Se modela el sistema de forma tal que sea capaz de cumplir con todos los requisitos, tanto los funcionales como los no funcionales. Además de ver los casos de uso representados en las clases del diseño y sus objetos, a partir de los cuales se forma el diagrama de clases del diseño.

3.3.1 Patrones del diseño

Los patrones son soluciones a problemas existentes y que pueden ser aplicados en varios contextos. Los patrones en si permiten codificar conocimientos, estilos y principios existentes y que se han probado que

son válidos. Entre sus características principales se puede ver un vocabulario de diseño común entre los desarrolladores, ayudan a diseñar los sistemas, independientes del lenguaje en el que se vaya a implementar y pueden llegar a acortar la fase de diseño en un proceso de desarrollo del software.

Entre los patrones usados para el desarrollo del Módulo de Facturación, y definido previamente por la arquitectura del Sistema de Información Hospitalaria alas HIS se encuentra *GRASP*, el cual maneja la asignación de responsabilidades para cada clase. Éste proporciona a una clase la posibilidad de instanciar otras clases en dependencia de las responsabilidades asignadas, poniendo así de manifiesto los patrones *Experto* y *Creador*, donde el primero viene siendo uno de los más usados.

Otro de los patrones utilizados son *Bajo Acoplamiento* y *Alta Cohesión*, permitiendo la asignación de responsabilidades entre clases siempre estableciendo una colaboración entre ellas, que permita dar paso a cualquier proceso de reutilización.

Hibernate como ORM por excelencia de Seam implementa dentro de su estructura patrones de diseño tales como *Abstract Factory*, el cual permite dentro de sus características determinar la subclase a partir de la cual se va a instanciar un objeto en tiempo de ejecución. Visto desde el punto de vista de Hibernate permite crear objetos de tipo *EntityManager* que es el objeto principal de la JPA utilizado para la persistencia de los datos.

Uno de los patrones más importantes que implementa Hibernate es el patrón *Query object*. Este está a encargo de interpretar la estructura de objetos y traducirlos a consultas (queries) SQL. Haciendo uso del mismo se pueden crear consultas referenciando a las clases y sus campos, en lugar de tablas y columnas independientemente del esquema en que puedan estar.

También se implementado por Hibernate el patrón *Active Record* el cual genera una clase a partir de una tabla de la base de datos, creando una asociación única entre las filas de la base de datos y los objetos del lenguaje de programación que se esté utilizando. Patrones como *Identity field*, *Foreign Key Mapping* y *Association Table Mapping*, son utilizados el primero para generar un único ID de forma tal que se defina un valor por el cual se identifique una entidad; el segundo es utilizado para mapeo de relaciones uno a muchos y el tercero para relaciones muchos a muchos.

Y finalmente como patrones de comportamiento implementados por Hibernate se encuentran *Identity map* para evitar tener en memoria dos representaciones distintas del mismo objeto en transacciones de

negocio, y *Lazy Load* para mantener en memoria solo los datos de los objetos que se invoquen en cada momento, evitando problemas como carga inmoderada y dependencias circulares.

3.4 Diagrama de paquetes

Para una mejor organización del modelo del diseño se propone una estructura en paquetes que a su vez agrupan subpaquetes que contienen los diagramas de clases del diseño y los diagramas de secuencia para cada caso de uso. Existe una relación entre paquetes donde se establecen dependencias entre las distintas clases que lo contienen.

Entre los paquetes se encuentra un *Repositorio de clases* que a su vez contiene tres subpaquetes: las *Entidades*, las *Sesiones* y las *Vistas*. Las *Entidades* contiene las entidades Autogeneradas por el ORM y las Personalizadas o creadas para una mejor gestión. Las *Sesiones* agrupan las Clases Controladoras Autogeneradas (Autogeneradas por el entorno de desarrollo), Personalizadas (Autogeneradas que llegan a ser modificadas) y del Proceso (solo las propias del proceso).

Por último el paquete de las *Vistas* está compuesto con contenidos Web referentes a las páginas clientes y los formularios que las componen, además contiene las vistas que interactúan con el usuario, a través de la cual se reciben y se muestran los datos.

Otro de los paquetes son: *Realizar facturación* y *Registrar prestaciones*, donde se crean las realizaciones de los casos de uso.

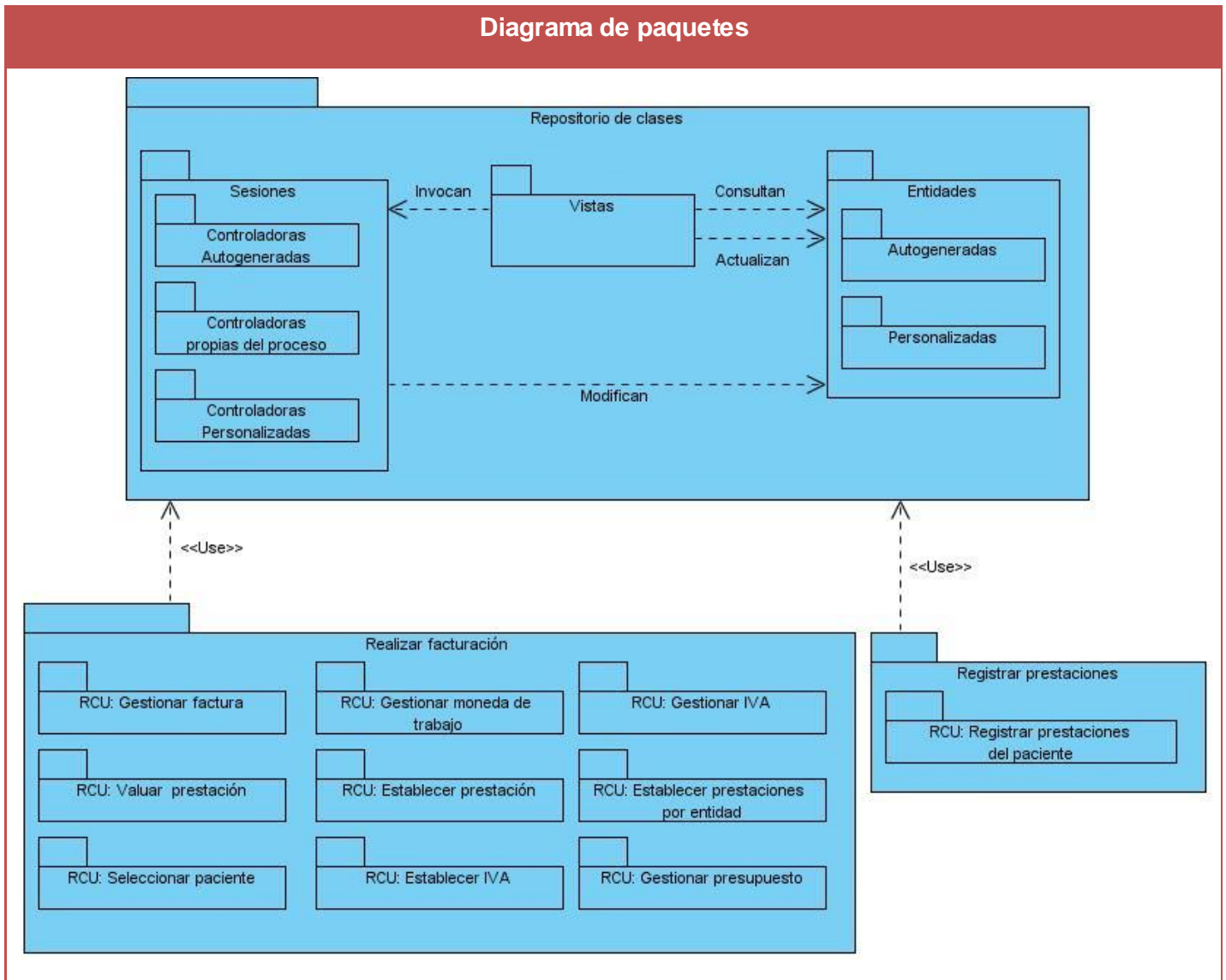


Figura 10. Diagrama de paquetes.

3.5 Estructura del diseño

Para dar una idea de cómo queda conformada la estructura del diseño para el módulo se tienen en cuenta que las páginas servidoras construyen las páginas clientes, las cuales son la interfaz visual mostrada al usuario. Estas contienen formularios que envían la información hacia la página servidora la cual hace uso de las clases y éstas transforman los datos en las entidades. Los formularios al mismo tiempo pueden actualizar o consultar las entidades, logrando así la estructura del patrón Modelo-Vista-Controlador.

3.5.1 Diagrama de clases del diseño

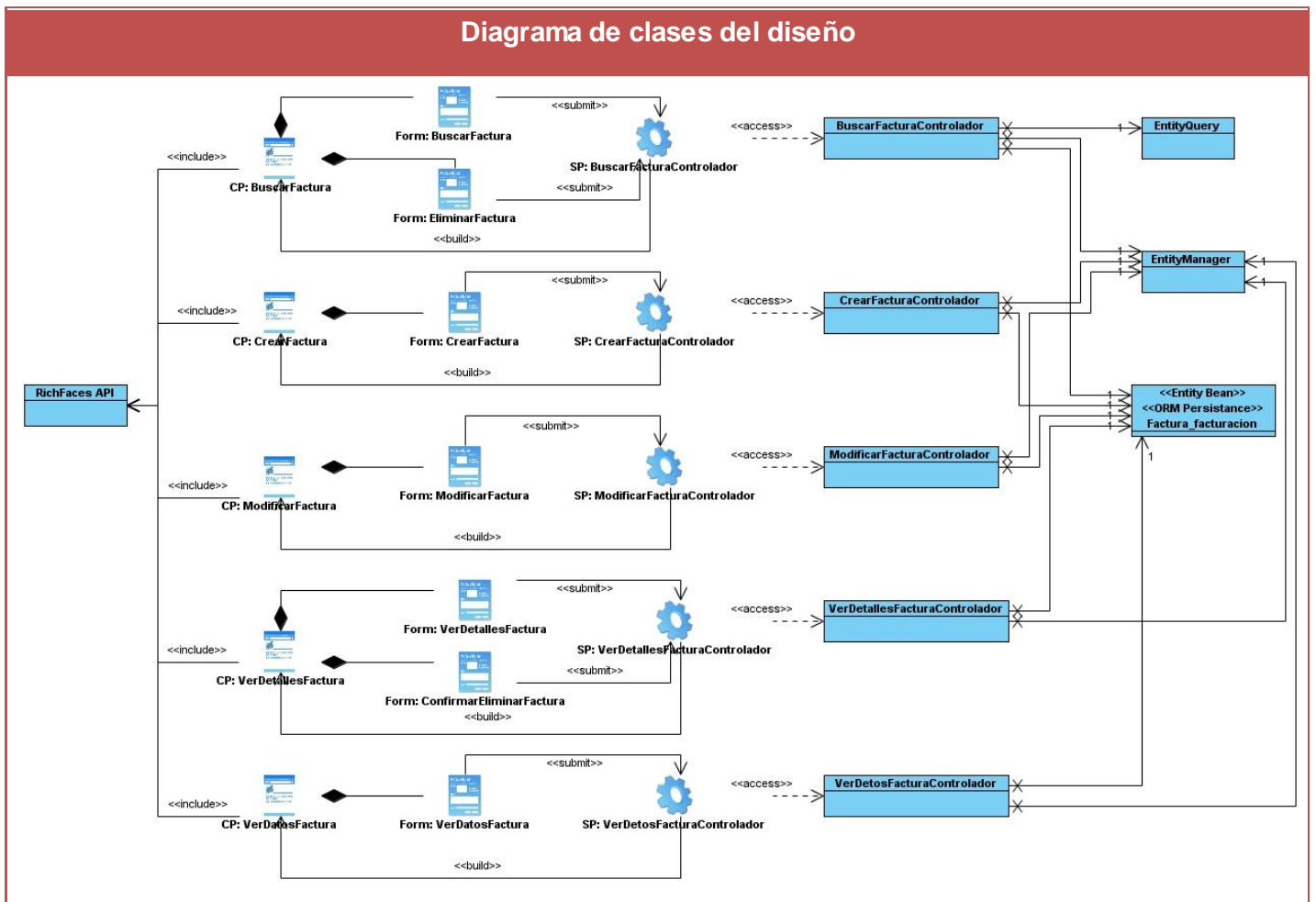


Figura 11. Diagrama de clase del diseño "Gestionar factura".

3.6 Diagrama de secuencia

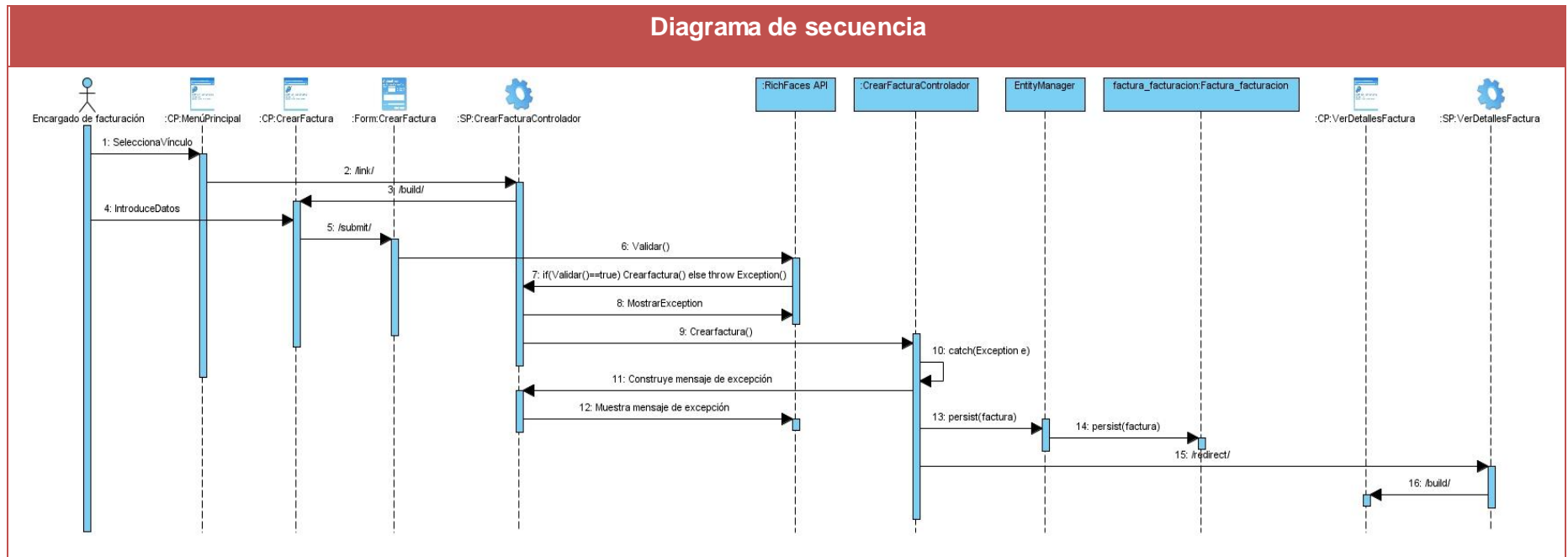


Figura 12. Diagrama de secuencia "Crear factura".

3.7 Descripción de las clases y sus atributos

Con el objetivo de un mejor entendimiento y comprensión de las principales clases identificadas en el diseño, se realiza, a continuación una descripción de las mismas con vistas a su futura implementación.

Nombre: CrearFacturaControlador	
Controladora	
Atributo	Tipo
idHf	Integer
cerrar	Boolean
pagoAbono	Boolean
ppLabSelected	List<PrestacionPacienteLab_facturacion>
ppBqSelected	List<PrestacionPacienteBq_facturacion>
ppHospSelected	List<PrestacionPacienteHosp_facturacion>
ppBsSelected	List<PrestacionPacienteBs_facturacion>
ppAnatpSelected	List<PrestacionPacienteAnatp_facturacion>
ppCeSelected	List<PrestacionPacienteCe_facturacion>
ppEmergSelected	List<PrestacionPacienteEmerg_facturacion>
ppFactSelected	List<PrestacionPacienteFact_facturacion>
idFactura	String
fechaFactura	Date
entityManager	EntityManager
facesMessages	FacesMessages
activeModule	IActiveModule
user	Usuario
Para cada responsabilidad:	

Nombre:	Begin()
Propósito:	Esta operación se encarga de inicializar las conversaciones y al mismo tiempo cargar las prestaciones del paciente.
Nombre:	CrearFactura()
Propósito:	Crea la factura persistiendo los datos de las prestaciones seleccionadas.
Nombre:	SeleccionarPaquetePrestacion(Integer id)
Propósito:	Selecciona las prestaciones que pertenezcan al paquete de prestaciones seleccionado.
Nombre:	GenerarCodigoFactura()
Propósito:	Genera un código consecutivo teniendo en cuenta la fecha actual en la que es creada la factura.
Nombre:	addSelectedPrestacion(Integer id)
Propósito:	Adiciona una prestación al listado de prestaciones a facturar

Tabla 8. Descripción de la clase "CrearFacturaControlador".

Nombre: CrearPresupuestoControlador	
Controladora	
Atributo	Tipo
idHf	Integer
precio	Float
IVA	Double
listPresSelectlabExist	List<PrestacionPacienteLab_facturacion>
listPresSelectBqExist	List<PrestacionPacienteBq_facturacion>

listPresSelectCeExist	List<PrestacionPacienteCe_facturacion>
listPresSelectFacExist	List<PrestacionPacienteFact_facturacion>
listPresSelectEmerExist	List<PrestacionPacienteEmerg_facturacion>
listPresSelectBsExist	List<PrestacionPacienteBs_facturacion>
listPresSelectAnatpExist	List<PrestacionPacienteAnatp_facturacion>
listPresSelectHospExist	List<PrestacionPacienteHosp_facturacion>
entityManager	EntityManager
facesMessages	FacesMessages
activeModule	IActiveModule
user	Usuario
Para cada responsabilidad:	
Nombre:	Begin()
Propósito:	Esta operación se encarga de inicializar las conversaciones y al mismo tiempo cargar las prestaciones del paciente.
Nombre:	selectPrestaciones(Integer id)
Propósito:	Permite seleccionar una prestación para registrar en el presupuesto.
Nombre:	CrearPresupuesto()
Propósito:	Esta operación permite registrar las prestaciones en el presupuesto persistiendo los datos.
Nombre:	End()
Propósito:	Cierra las conversaciones.

Tabla 9. Descripción de la clase "CrearPresupuestoControlador".

Nombre: CrearMonedaControlador	
Controladora	
Atributo	Tipo
entityManager	EntityManager
facesMessages	FacesMessages
activeModule	IActiveModule
codigo	String
simbolo	String
nombre	String
error	Integer
moneda	Moneda_facturacion
Para cada responsabilidad:	
Nombre:	Begin()
Propósito:	Esta operación se encarga de iniciar las conversaciones.
Nombre:	End()
Propósito:	Esta operación se encarga de finalizar las conversaciones.
Nombre:	Validar()
Propósito:	Valida que el nombre de la moneda que se desea crear no se encuentre registrado en la Base de Datos.
Nombre:	MonedaActiva()
Propósito:	Comprueba que no existe una moneda activa.

Nombre:	CrearMoneda()
Propósito:	Crea una nueva moneda y la persiste en la Base de Datos.

Tabla 10. Descripción de la clase "CrearMonedaControlador".

3.8 Modelo de datos

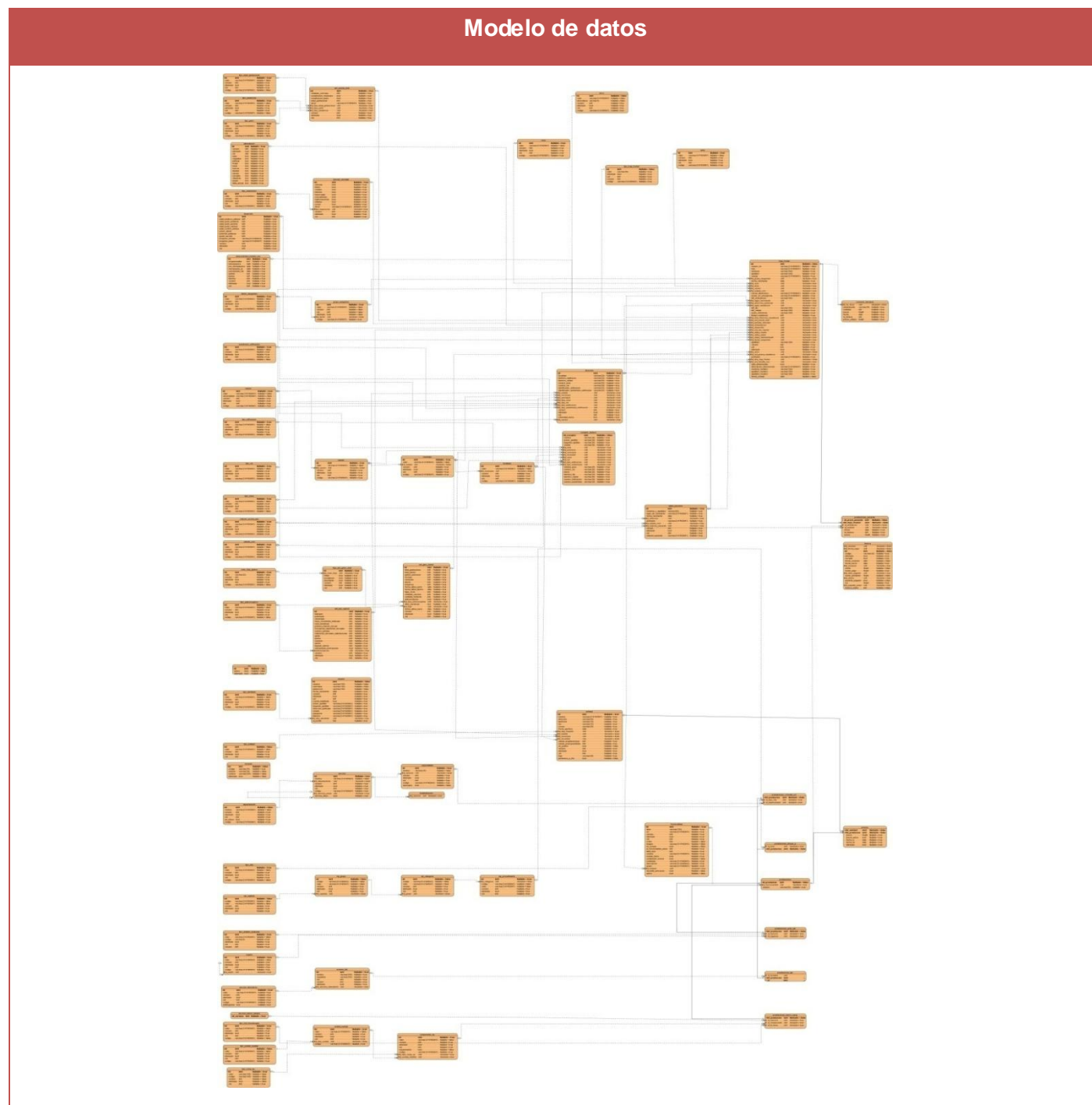


Figura 13. Modelo de datos para el Módulo de Facturación.

3.8.1 Descripción de las tablas

Factura				
Descripción	Contiene la información referente a los gastos del paciente en la institución, por ello incluye el identificador de los pacientes, factura, identificador del receptor de la misma, moneda de trabajo, valor del I.V.A. y el estado de la factura.			
Atributo	Tipo	Descripción	Tipo llave	Nulo
id	Integer	Identificador de la factura que se crea.	Primaria	No
codigo	Varchar(20)	Identificador propio de la factura conformado por la fecha y un número en secuencia.		Si
id_hoja_frontal	Integer	Identificador de la historia clínica del paciente.	Foránea	No
id_moneda	Integer	Identificador de la moneda de trabajo establecida.	Foránea	No
id_user	Integer	Identificador del usuario del sistema que crea la factura.	Foránea	No
id_forma_pago	Integer	Identificador de la forma de pago mediante la cual será cobrada la factura.	Foránea	No
eliminado	Boolean	Determina si la información ha sido eliminada.		No
cerrada	Boolean	Determina si la factura ha sido cerrada y pagada.		No
fecha_creación	Date	Informa la fecha de creación de la factura.		No
fecha_cierre	Date	Informa la fecha de cierre de la factura.		Si
monto	Real	Cantidad de dinero total por el cual es emitida la factura.		No
monto_pago	Real	Cantidad de dinero ya pagado del monto de la		No

		factura.		
monto_pendiente	Real	Cantidad de dinero pendiente del pago de la factura.		No
id_entidad	Integer	Identificador de la entidad a la cual pertenece la factura.	Foránea	No
id_convenio	Integer	Identificador del convenio mediante el cual se efectúa el pago de la factura.	Foránea	Si
paciente_pagador	Boolean	Determina si el paciente es el pagador de la factura.		No
id_tutor_pagador	Integer	Identificador del tutor encargado de pagar la factura.		Si
id_abono	Integer	Identificador del abono mediante el cual se hace pago de la factura.		Si
iva	Real	IVA por el cual fue emitida la factura.		No
id_paquete_prest	Integer	Identificador del paquete de prestaciones facturado en la factura.		Si
Factura_madre	Integer	Identificador de la factura de la cual es hija.		No

Tabla 11. Descripción de la tabla "Factura".

Moneda				
Descripción	Contiene la información de las distintas monedas de trabajo con las que se puede facturar en la institución.			
Atributo	Tipo	Descripción	Tipo llave	Nulo
id	Integer	Identificador de la moneda.	Primaria	No
símbolo	Varchar(20)	Símbolo identificativo de la moneda.		No

Capítulo 3: Diseño

codigo	Varchar(20)	Código identificativo de la moneda.		Si
nombre	Varchar (255)	Nombre de la moneda.		No
eliminado	Boolean	Determina si la información ha sido eliminada.		No

Tabla 12. Descripción de la tabla "Moneda".

Obra Social				
Descripción	Se encuentran contenidos todos los datos presentes en una obra social.			
Atributo	Tipo	Descripción	Tipo llave	Nulo
id	Integer	Identificador de la Obra Social	Primaria	No
id_datos_bancarios	Integer	Identificador de los datos bancarios pertenecientes a la Obra Social	Foránea	Si
id_forma_pago	Integer	Identificador de la forma de pago perteneciente a la Obra Social	Foránea	No
nombre	Varchar(30)	Nombre de la obra social.		No
nombre_fiscal	Varchar(30)	Nombre con el que se encuentra registrado en la entidad fiscal del país.		No
código_ident_fiscal	Varchar(20)	Código fiscal único para cada obra social.		No
teléfono_fijo	Varchar(20)	Número del teléfono fijo.		Si
teléfono_movil	Varchar(20)	Número del teléfono móvil.		Si
fax	Varchar(20)	Número del fax de la empresa.		Si
email	Varchar(30)	Dirección de correo de la obra social.		Si
eliminado	boolean	Indica si la factura está eliminada o no.		No

Id_direccion	Integer	Identificador de la dirección de la Obra Social		No
--------------	---------	---	--	----

Tabla 13. Descripción de la tabla "Obra Social".

Culminado el estudio del diseño del módulo a desarrollar se pueden evidenciar los nuevos artefactos generados para el desarrollo y culminación de la solución propuesta. Se definió la arquitectura a utilizar analizando sus principales ventajas, además se generaron las clases del diseño, diagrama de paquetes y diagramas de secuencia, vitales para el posterior desarrollo. También se estableció el Modelo de Datos basado en la técnica de modelado Entidad – Relación, el cual es punto de partida como contenedor de datos. Como punto final, e inicio del siguiente capítulo queda implementar la solución propuesta.

Capítulo 4: Implementación

Luego de haberse definido el diseño del módulo, además de describir el modelo de datos mediante el siguiente capítulo se dará una perspectiva más amplia sobre las principales características de la implementación del módulo. Se muestran los principales componentes y sus relaciones mediante un Diagrama de componentes. También se exponen las estrategias de seguridad adoptadas así como los tratamientos de errores aplicados.

4.1 Modelo de despliegue

El modelo de despliegue es utilizado para definir la estructuración, descripción y relaciones físicas para el hardware propuesto para el despliegue del sistema.

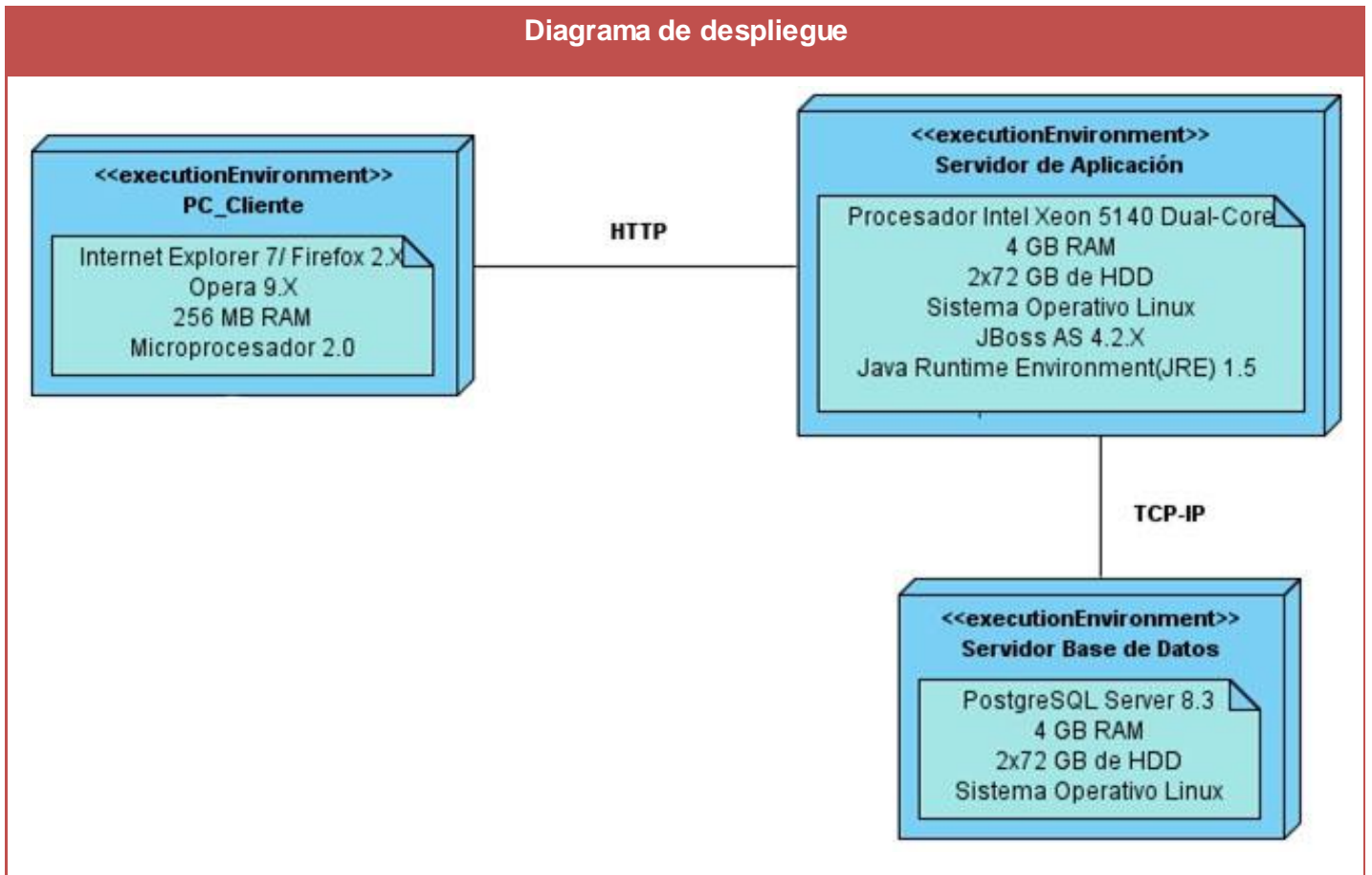


Figura 15. Diagrama de despliegue.

4.2 Diagrama de paquetes de componentes

El diagrama de componentes describe la descomposición física del sistema de software (y, eventualmente, de su entorno organizativo) en componentes, a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se realiza en términos de componentes y de relaciones entre los mismos. (45)

Diagrama de paquetes de componentes

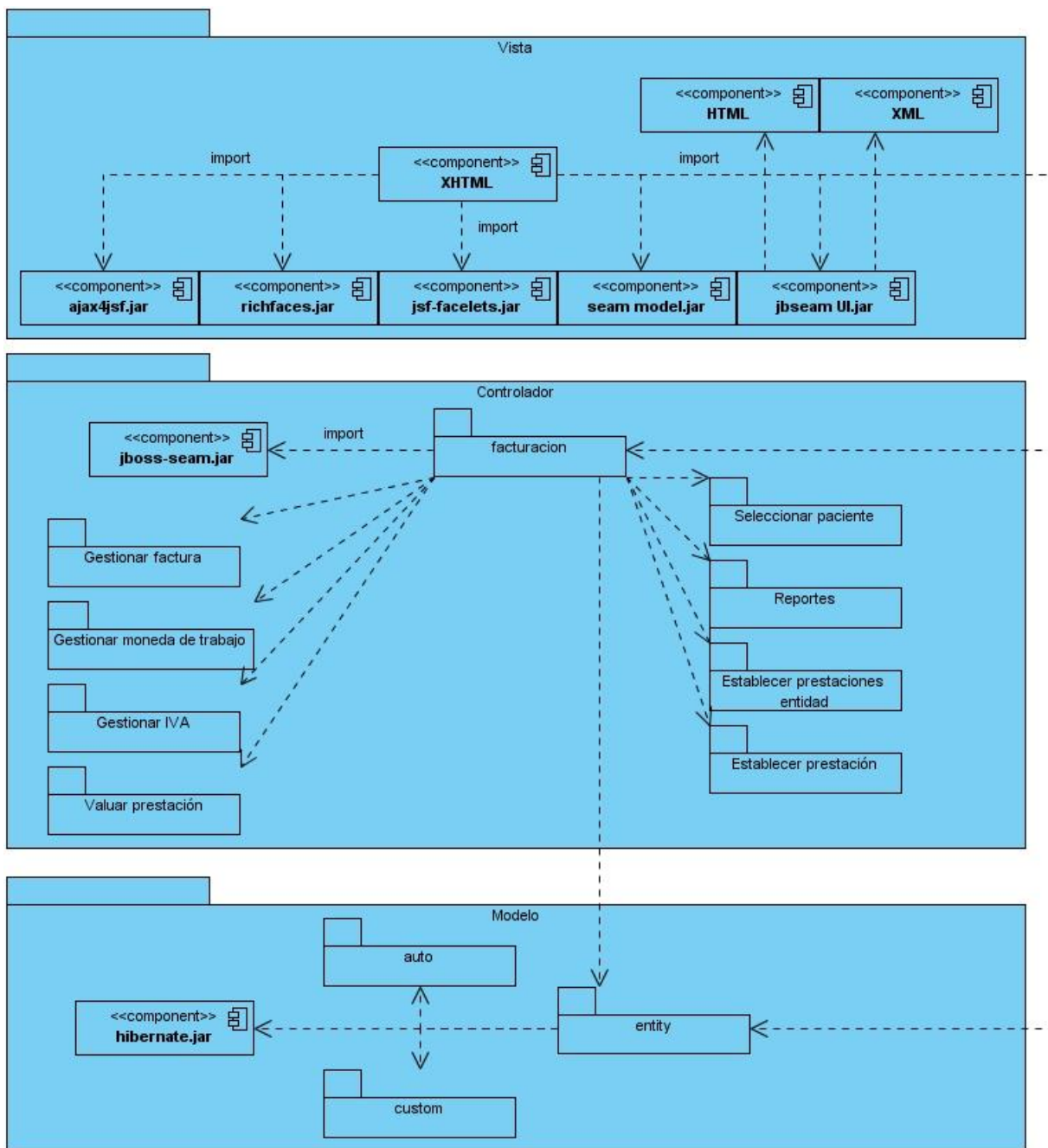


Figura 16. Diagrama de paquetes de componentes.

4.2.1 Diagrama de componentes “Crear factura”

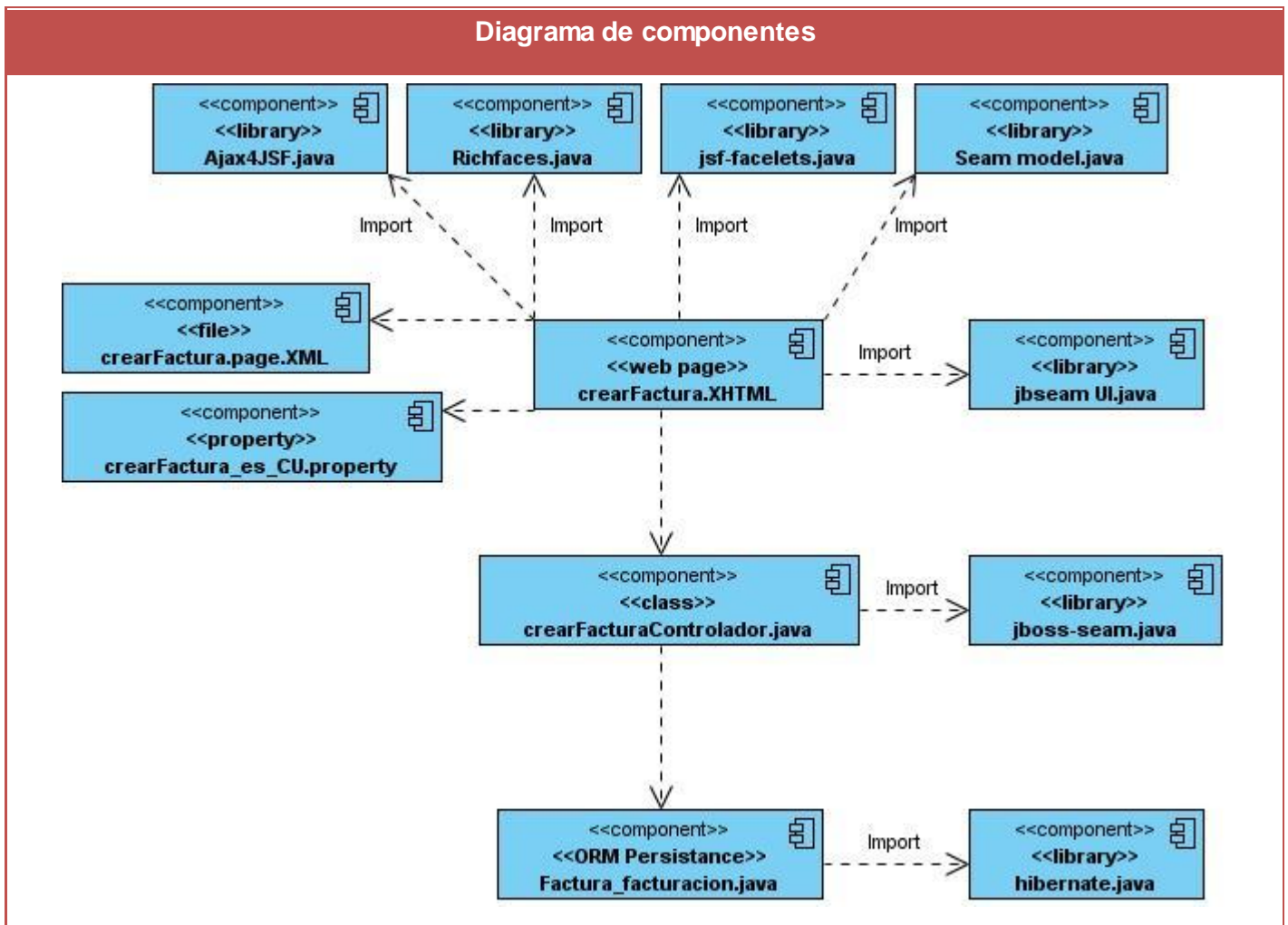


Figura 17. Diagrama de componentes “Crear factura”.

4.3 Validaciones

El proceso de validación se define como un programa documentado que evidencia el correcto funcionamiento del sistema, ofreciendo todas las garantías de calidad para que produzca en forma consistente en conformidad con las especificaciones de calidad previamente establecidas.

Hacer uso de las validaciones es útil dado que el usuario puede ser retroalimentado con información sobre las mismas, asegura que los servicios para los cuales fueron definidos se desarrollen de forma eficiente, estableciendo rangos de usabilidad y por último evita la entrada de errores o intentos de violar la

seguridad del sistema. Entre los problemas principales que se evidencian al validar un sistema se encuentran la declaración múltiple de una misma validación, duplicación de código y múltiples chequeos en tiempo de ejecución.

La solución se encuentra en la utilización del framework JSF, el mismo proporciona utilidades para definir de manera sencilla las validaciones en la vista. Al mismo tiempo es posible definir una única vez las validaciones en el negocio, utilizando Hibernate Validator e invocando dichas validaciones donde sean requeridas. Esto es posible gracias a al framework de integración Seam que agrupa JSF en conjunto con Hibernate Validator. (46)

4.4 Descripción de algoritmos utilizados

El término *algoritmo* se utiliza en informática para describir un método de resolución de un problema que es adecuado para su implementación como programa de computadora. Los algoritmos son la “esencia” de la informática, son uno de los centros de interés de muchas, si no todas, las áreas del campo de la informática. (47)

El Módulo de Facturación comprende varios procesos fundamentales, donde se puede apreciar Crear factura el cual agrupa la mayor cantidad de algoritmos y complejidad. Entre los algoritmos existentes como son los casos de *Ordenamiento*, *Búsqueda*, *Procesamiento de cadenas*, *Geométricos*, *Matemáticos* y *Avanzados*, el módulo hace uso de los *Algoritmos de Búsqueda*, los cuales permiten de forma eficiente encontrar elementos dentro de las estructuras de datos utilizadas. Para ellos se implementan búsquedas lineales conformadas por una o varias iteraciones anidadas las cuales concluyen al encontrar el elemento deseado.

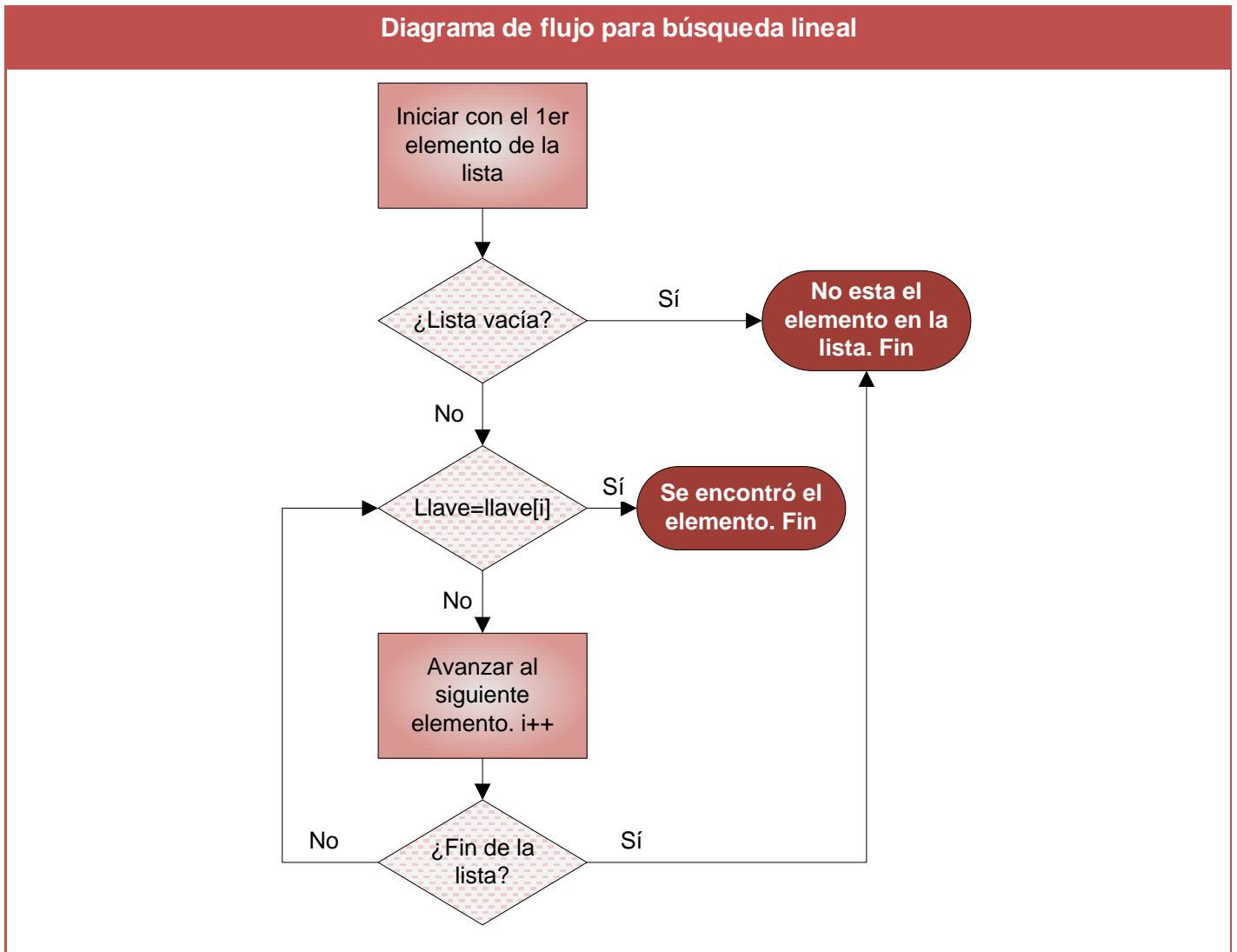


Figura 18. Diagrama de flujo para búsqueda lineal.

4.5 Tratamiento de errores

Las excepciones dentro de un sistema informático son aquellos eventos que ocurren durante la ejecución del programa y que a su vez interrumpen el flujo normal de las acciones. Capturarlas y tratarlas es el mecanismo recomendado para evitar la propagación de errores en la aplicación. Al ocurrir un error dentro de un método en Java, automáticamente se crea el objeto “Exception” el cual es tratado en el sistema de ejecución. El objeto trae consigo varias informaciones referentes a la excepción, como son entre otros: tipo y estado del programa.

El Sistema de Información Hospitalaria alas HIS propone un tratamiento de excepciones basado principalmente en la protección de las zonas críticas, como pueden ser: inserción de datos a la base de datos o la modificación de los mismos, así como los procesos de validación. Al ocurrir una excepción que implica la redirección es manejada mediante las interfaces “*.page.xml”. Las mismas se encargan de capturar de forma general los datos referentes a la excepción y ejecutar las acciones establecidas para cada caso.

Para el control y alerta de las demás excepciones es utilizado el componente FacesMessages que brinda el framework Seam el cual se encarga de mostrar los mensajes que se manejan a través del objeto facesMessages inyectado en las clases controladoras. Estos mensajes son tratados por tipos: error, alerta y notificación.

4.6 Seguridad

En capítulos anteriores se trata el tema de la seguridad como punto fundamental para los requisitos no funcionales del sistema. Pero es de suma importancia hacer énfasis en la seguridad, siempre que se traten temas tan delicados como la salud de las personas. El control se encuentra garantizado mediante un nivel de acceso de usuarios y contraseñas, garantizando el acceso sólo a los niveles establecidos y a las funcionalidades permitidas, de acuerdo con la función que realiza cada usuario.

Luego de analizado el modelo de datos presentado en el capítulo anterior se pudo obtener un acercamiento a la implementación en términos de componentes. Se vieron los diagramas de despliegue para una mejor idea sobre la distribución del sistema en concepto de hardware, y de componentes mostrándose las relaciones existentes entre los componentes de implementación ya de una forma más detallada. El modelo de clases persistentes abarcó todo el modelo de datos ya estudiado pero esta vez como estructuras de clases.

Conclusiones

Con el desarrollo del Módulo de Facturación para el Sistema de Información Hospitalaria alas HIS, se tiene en cuenta las tareas trazadas para la investigación y da solución al problema propuesto se puede arribar a las siguientes conclusiones:

- ✓ Los sistemas de facturación para servicios médicos analizados no pueden ser tomados como solución para el problema planteado, debido a que la mayoría son propietarios, además de no poseer características semejantes al Sistema de Información Hospitalario alas HIS que permita su integración y posterior uso.
- ✓ Las herramientas, tecnologías y patrones seleccionados permitieron crear un módulo adaptable a diferentes entornos de despliegue y capaz de gestionar la información relacionada con la facturación de servicios médicos en las instituciones hospitalarias.
- ✓ La aplicación del patrón de diseño MVC permitió la organización de las funcionalidades en tres componentes de diseño según sus responsabilidades, disminuyendo la complejidad de implementación de los casos de uso.
- ✓ La utilización de las pautas definidas garantizaron la uniformidad visual de todas las interfaces del sistema.
- ✓ El producto obtenido contribuye a que el Sistema de Información Hospitalaria alas HIS sea un producto más completo, mejorando la calidad de servicio en las instituciones de salud y disminuyendo las insatisfacciones producto del burocratismo y la demora de los procesos.

Recomendaciones

Los autores recomiendan:

- ✓ Implementar funcionalidades de configuración que permitan establecer en tiempo de ejecución los módulos del HIS que se deben facturar.
- ✓ Extender el proceso actual de facturación adicionando el concepto de gasto para la institución y la gestión de vales.
- ✓ Incluir la gestión de pagos a galenos por conceptos de consultas brindadas, categoría y especialidad.
- ✓ Permitir la generación de reportes sobre facturas, facturas por paciente y cierres de caja.

Referencias bibliográficas

1. **Estanque Díaz, Diuber.** *Tesis Diseño del Módulo de Facturación del Sistema de Información Hospitalaria alas HIS.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009.
2. Igual a la referencia número 1.
3. Igual a la referencia número 1.
4. **Articulosinformativos.** Articulosinformativos. [En línea] 2006. <http://www.articulosinformativos.com.mx>.
5. **Garrigues, Joaquin.** *Curso de derecho mercantil.* Bogotá : s.n., 1987.
6. **www.modelofactura.net.** Factura ordinaria. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modelofactura.net/factura-ordinaria.html>.
7. —. Factura rectificativa. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modelofactura.net/factura-rectificativa.html>.
8. **www.modelofactura.net.** Factura recapitulativa. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modelofactura.net/factura-recapitulativa.htm>.
9. **Ser autónomo.** Ser autónomo. [En línea] 2008. [Citado el: 12 de Noviembre de 2010.] <http://www.serautonomo.net/tipos-de-factura.html>.
10. **NetClinicas Software.** *Manual de Ayuda del Software de Gestión de Clínicas NetClínicas.* España : s.n.
11. **QSOFT.** *DOSSIER del Sistema de Gestión de la Información de Salud SALUS.* España : s.n., 1995.
12. **Vinar, Juan.** Conferencias sobre el Sistema de Gestión de la Información Hospitalaria MICHIGAN. Argentina : s.n., 2009.
13. **CITMATEL.** RODAS XXI sobre SQL.MANUAL DE USUARIO. 2005.
14. **Zaninotto, François and Potencier, Fabien.** *Guía definitiva de Symfony.* 2008.
15. **Tenambaum, Andrew S.** *Redes de computadoras.* México : Pearson Educación, 2003. 970-26-0162-2.
16. **JSR-314 (JSF 2.0) Expert Group.** JavaServer Faces.org. [En línea] <http://www.java-serverfaces.org/>.

Referencias bibliográficas

17. **Magnolia, SBS, JBoss EAP, and RHEL.** RichFaces. [En línea] <http://www.jboss.org/richfaces>.
18. **Ajax.org.** Ajax.org. [En línea] <http://www.ajax.org/>.
19. **Junta de Andalucía.** Ajax for Java Server Faces. [En línea] 2009. [Citado el: 7 de Diciembre de 2010.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Ajax4JSF>.
20. **Hookom, Jacob.** Facelets - JavaServer Faces View Definition Framework. [En línea] <http://facelets.java.net/nonav/docs/dev/docbook.html#intro>.
21. **World Wide Web Consortium.** Guía Breve de XHTML. [En línea] 2008. [Citado el: 10 de Diciembre de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.
22. **Red Hat Middleware.** Seamframework. [En línea] 2009. [Citado el: 10 de Diciembre de 2010.] <http://seamframework.org/>.
23. Igual a la referencia número 22. [En línea]
24. Igual a la referencia número 22. [En línea]
25. **Magnolia, SBS, JBoss EAP, and RHEL.** Hibernate - JBoss Community. [En línea] <http://www.hibernate.org/>.
26. **Oracle and Sun.** Enterprise JavaBeans Technology. [En línea] <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>.
27. Sistema Gestor de base de datos SGBD. [En línea] 1 de Noviembre de 2004. [Citado el: 10 de Noviembre de 2010.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php..
28. **Postgres Global Development Group.** Manual del usuario de PostgreSQL. [En línea] 1996.
29. Igual a la referencia número 20.
30. **Oracle and/or its affiliates.** *The Java EE 5 Tutorial*. Redwood City, CA 94065 USA. : s.n., 2010. 821-1743-10.
31. —. Java SE 6 Documentation. [En línea] 1993,2011. <http://download.oracle.com/javase/6/docs/>.
32. Programas de desarrollo libres. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www.kubuntues.org/wiki/desarrollo-programacion/programas-desarrollo-libres..>

Referencias bibliográficas

33. **Magnolia, SBS, JBoss EAP, and RHEL.** JBoos Tools. Eclipse Plugins for JBoss technology. [En línea] <http://www.jboss.org/tools>.
34. **Magnolia, SBS, JBoss EAP, and RHEL.** . JBoss Application Server. [En línea] <http://www.jboss.org/jbossas/>.
35. **Object Management Group, Inc.** Object Management Group - UML. [En línea] 1997. [Citado el: 10 de Noviembre de 2010.] <http://www.uml.org/>.
36. Rational Unified Process. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www.rational.com.ar/herramientas/rup.html>.
37. **IBM.** IBM Rational Unified Process (RUP). [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www-01.ibm.com/software/awdtools/rup/>.
38. **BizAgi.** Soluciones de Gestión de Procesos de Negocio, BPM y Automatización de Flujos de Trabajo | BizAgi BPMS. [En línea] 2011. [Citado el: 10 de Noviembre de 2010.] <http://www.bizagi.com/>.
39. **Object Management Group, Inc.** BPMN Information Home. [En línea] 1997. [Citado el: 10 de Noviembre de 2010.] <http://www.bpmn.org/>.
40. **Fowler, Martin and Scott, Kendall.** *UML gota a gota*. México : Addison Wesley Longman de México, 2009. 968-444-364-1.
41. Igual a la referencia número 1.
42. Igual a la referencia número 1.
43. **Sommerville, Ian.** *Ingeniería del software*. Madrid : PEARSON EDUCACIÓN S.A., 2005. 84-7829-074-5.
44. **Campderrich Falgueras, Benet.** *Ingeniería del software*. Barcelona : Editorial UOC, 2003. 84-8318-997-6.
45. Igual a la referencia número 33.
46. **Ferentschik, Hardy and Morling, Gunnar.** *Hibernate Validator. Reference Guide*.
47. **Sedgewick, Robert.** *Algoritmos en C++*. Wilmington, Delaware, E.U.A. : Addison-Wesley Iberoamericana, S.A., 1995. 0-201-62574-1.

Bibliografía

Estanque Díaz, Diuber. *Tesis Diseño del Módulo de Facturación del Sistema de Información Hospitalaria alas HIS.* Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009.

Articulosinformativos. Articulosinformativos. [En línea] 2006. <http://www.articulosinformativos.com.mx>.

Garrigues, Joaquin. *Curso de derecho mercantil.* Bogotá : s.n., 1987.

www.modeloactura.net. Factura ordinaria. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modeloactura.net/factura-ordinaria.html>.

www.modeloactura.net. Factura rectificativa. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modeloactura.net/factura-rectificativa.html>.

www.modeloactura.net. Factura recapitulativa. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://www.modeloactura.net/factura-recapitulativa.htm>.

Ser autónomo. Ser autónomo. [En línea] 2008. [Citado el: 12 de Noviembre de 2010.] <http://www.serautonomo.net/tipos-de-factura.html>.

NetClínicas Software. *Manual de Ayuda del Software de Gestión de Clínicas NetClínicas.* España : s.n.

QSOFT. *DOSSIER del Sistema de Gestión de la Información de Salud SALUS.* España : s.n., 1995.

Vinar, Juan. Conferencias sobre el Sistema de Gestión de la Información Hospitalaria MICHIGAN. Argentina : s.n., 2009.

CITMATEL. RODAS XXI sobre SQL.MANUAL DE USUARIO. 2005.

Zaninotto, François and Potencier, Fabien. *Guía definitiva de Symfony.* 2008.

Tenambaum, Andrew S. *Redes de computadoras.* México : Pearson Educación, 2003. 970-26-0162-2.

JSR-314 (JSF 2.0) Expert Group. JavaServer Faces.org. [En línea] <http://www.java-serverfaces.org/>.

Magnolia, SBS, JBoss EAP, and RHEL. RichFaces. [En línea] <http://www.jboss.org/richfaces>.

Ajax.org. Ajax.org. [En línea] <http://www.ajax.org/>.

Junta de Andalucía. Ajax for Java Server Faces. [En línea] 2009. [Citado el: 7 de Diciembre de 2010.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Ajax4JSF>.

Hookom, Jacob. Facelets - JavaServer Faces View Definition Framework. [En línea] <http://facelets.java.net/nonav/docs/dev/docbook.html#intro>.

World Wide Web Consortium. Guía Breve de XHTML. [En línea] 2008. [Citado el: 10 de Diciembre de 2010.] <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.

Red Hat Middleware. Seamframework. [En línea] 2009. [Citado el: 10 de Diciembre de 2010.] <http://seamframework.org/>.

Magnolia, SBS, JBoss EAP, and RHEL. Hibernate - JBoss Community. [En línea] <http://www.hibernate.org/>.

Oracle and Sun. Enterprise JavaBeans Technology. [En línea] <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>.

Sistema Gestor de base de datos SGBD. [En línea] 1 de Noviembre de 2004. [Citado el: 10 de Noviembre de 2010.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.

Postgres Global Development Group. Manual del usuario de PostgreSQL. [En línea] 1996.

Oracle and/or its affiliates. *The Java EE 5 Tutorial*. Redwood City, CA 94065 USA. : s.n., 2010. 821-1743-10.

Oracle and/or its affiliates. Java SE 6 Documentation. [En línea] 1993,2011. <http://download.oracle.com/javase/6/docs/>.

Programas de desarrollo libres. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres>.

Magnolia, SBS, JBoss EAP, and RHEL. JBoos Tools. Eclipse Plugins for JBoss technology. [En línea] <http://www.jboss.org/tools>.

Magnolia, SBS, JBoss EAP, and RHEL. . JBoss Application Server. [En línea] <http://www.jboss.org/jbossas/>.

Object Management Group, Inc. Object Management Group - UML. [En línea] 1997. [Citado el: 10 de Noviembre de 2010.] <http://www.uml.org/>.

Rational Unified Process. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www.rational.com.ar/herramientas/rup.html>.

IBM. IBM Rational Unified Process (RUP). [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www-01.ibm.com/software/awdtools/rup/>.

BizAgi. Soluciones de Gestión de Procesos de Negocio, BPM y Automatización de Flujos de Trabajo | BizAgi BPMS. [En línea] 2011. [Citado el: 10 de Noviembre de 2010.] <http://www.bizagi.com/>.

Object Management Group, Inc. BPMN Information Home. [En línea] 1997. [Citado el: 10 de Noviembre de 2010.] <http://www.bpmn.org/>.

Fowler, Martin and Scott, Kendall. *UML gota a gota*. México : Addison Wesley Longman de México, 2009. 968-444-364-1.

Sommerville, Ian. *Ingeniería del software*. Madrid : PEARSON EDUCACIÓN S.A., 2005. 84-7829-074-5.

Campderrich Falgueras, Benet. *Ingeniería del software*. Barcelona : Editorial UOC, 2003. 84-8318-997-6.

Ferentschik, Hardy and Morling, Gunnar. *Hibernate Validator. Reference Guide*.

Sedgewick, Robert. *Algoritmos en C++*. Wilmington, Delaware, E.U.A. : Addison-Wesley Iberoamericana, S.A., 1995. 0-201-62574-1.

Emmanuel Bernard, Steve Ebersole, and Gavin King. *Hibernate EntityManager. User guide*. 2009.

Pressman, Rogert S. *Ingeniería del Software. Un enfoque práctico*. 2005.

Koenig, John. *JBoss jBPM. WHITE PAPER*. 2004.

Jboss Community. *RichFaces Developer Guide*. 2010.

Anexo 1: Descripción de la solución propuesta

Proceso de facturación

Entre los procesos fundamentales del Módulo de Facturación se encuentra Crear factura, el cual comienza seleccionando el paciente que se desea facturar, en el listado solo aparecen los pacientes que tiene registrada al menos una prestación a su nombre.

Interfaz “Seleccionar Historia Clínica a facturar”

The screenshot displays the 'Seleccionar paciente' interface. At the top, there is a search bar labeled 'Carné Identidad:' with a 'Buscar' button and a 'Cancelar' button. Below the search bar is a 'Búsqueda avanzada' link. The main area is a table titled 'Listado de pacientes' with the following columns: Foto, Paciente, Carné Identidad, Fecha nacimiento, and Sexo. The table contains five rows of patient data:

Foto	Paciente	Carné Identidad	Fecha nacimiento	Sexo
	Alvin Farmacia Central	32145678987	03/05/2010	Femenino
	Amalia Sosa Lin	09887776652	08/04/1992	Femenino
	Luis Perez Ramop	09209009019	13/07/1976	Masculino
	Jose Sanchez Gomez	84101298746	18/10/2006	Masculino
	Maikel David Ruenes Correa	80121124505	11/12/1980	Masculino

At the bottom of the interface, there is a copyright notice: © Universidad de las Ciencias Informáticas, 2010.

Figura 19. Interfaz “Seleccionar Historia Clínica a facturar”.

Seleccionado el paciente se procede a establecer las prestaciones que se desean facturar mediante la interfaz Crear factura. En la misma se ven agrupadas las prestaciones por área permitiendo Cancelar o Asumir como gasto dichas prestaciones. Al mismo tiempo que permite seleccionar el pagador de la factura, que puede ser el paciente, un tutor y en caso de tener registrada a su nombre una obra social.

Interfaz "Crear factura"

alashis
SISTEMA DE INFORMACIÓN HOSPITALARIA

Root Root
CIMEQ
Durango

Inicio Preferencias Notificaciones (0) Ayuda Salir

Facturación

Crear factura

Datos del paciente No.H.C.: 587469

Nombre: Jose Carné Identidad: 84101298746
 Primer apellido: Sanchez Fecha nacimiento: 18/10/2006
 Segundo apellido: Gomez Sexo: Masculino

Pagador

Paciente Tutor Obra Social

Datos de factura Código factura: 2011062300281

Datos generales

Fecha de creación: 23/06/2011 Moneda: Pesos cubanos Caja: Caja 6

Listado de prestaciones de Laboratorio

Nombre	Fecha	Precio	
Sistema	23/06/2011	78.0	<input type="checkbox"/>
Chagas	23/06/2011	0.0	<input type="checkbox"/>
Urea	23/06/2011	0.0	<input type="checkbox"/>
Orina	23/06/2011	0.0	<input type="checkbox"/>
Subtotal		\$ 0.0	

Listado de prestaciones de Hospitalización

Listado de prestaciones de Banco de Sangre

Listado de prestaciones de Emergencias

Otros datos

Subtotal \$ 0.0
 IVA 0.03
 Total \$ 0.0

© Universidad de las Ciencias Informáticas, 2010.

Figura 20. Interfaz "Crear factura".

Luego de creada la factura los datos pueden ser vistos de forma detallada mediante las interfaces Ver detalles de factura y Ver datos de factura.

Interfaz "Ver datos de factura"

The interface displays the following information:

- Header:** alasHIS SISTEMA DE INFORMACIÓN HOSPITALARIA. User: Root Root, CIMEQ Durango. Navigation: Inicio, Preferencias, Notificaciones (0), Ayuda, Salir.
- Facturación Menú:** Marcadores, Buscar historia clínica, Prestaciones, Gestión de facturas (Crear factura, Buscar factura), Presupuesto, Cierre de caja, Obras sociales, Convenios, Abonos, Configuración.
- verDatosFactura:**
 - Datos generales de paciente:** No.H.C.: 103. Nombre: Paula Carrasco Trujillo. Carné Identidad: 86100769203. Fecha nacimiento: 05/05/2009. Sexo: Femenino.
 - Datos del pagador:** Pagador: Tutor.

Datos generales	Dirección	Datos bancarios
Nombre: Carlos Infante Ramos		
Carné Identidad: 87052825089		
 - Datos de factura:** Código factura: 2011061900216.

Datos generales			
Fecha de creación:	19/06/2011	Moneda emisión:	Pesos cubanos
Forma de pago:	Efectivo	Estado de la factura:	Cerrada
Caja:	Caja 6	Fecha de cierre:	19/06/2011
 - Listado de prestaciones de Laboratorio:**

Nombre	Fecha	Precio
Sistema	14/06/2011	78.0
Subtotal		\$ 78.0
 - Otros datos:**

IVA:	0.02	Monto total:	\$ 79.56	Monto pendiente:	\$ 0.0
Monto pago:	\$ 79.56				

Exportar Salir

© Universidad de las Ciencias Informáticas, 2010.

Figura 21. Interfaz "Ver datos de factura".

Otras interfaces

Interfaz "Cerrar caja"

alás HIS
SISTEMA DE INFORMACIÓN HOSPITALARIA

Root Root
CIMEQ
Durango

Inicio Preferencias Notificaciones (0) Ayuda Salir

Facturación

Menú

- Marcadores
- Buscar historia clínica
- Prestaciones
- Gestión de facturas
 - Crear factura
 - Buscar factura
- Presupuesto
- Cierre de caja
 - Cerrar caja
 - Buscar cierre de caja
- Obras sociales
- Convenios
- Abonos
- Configuración

Crear cierre de caja

Datos generales

Nombre de caja: Caja 6 Fecha último cierre: 21/06/2011 Hora último cierre: 02:38 PM
 Fecha actual: 23/06/2011 Hora actual: 04:04 AM Fondo inicial: 100.0

Listado de facturas cerradas

Código	Fecha de creación	Fecha de cierre	Monto	
2011062300284	23/06/2011	23/06/2011	41.7665	<input checked="" type="checkbox"/>
2011062300285	23/06/2011	23/06/2011	132.922	<input checked="" type="checkbox"/>
2011062300286	23/06/2011	23/06/2011	80.34	<input checked="" type="checkbox"/>
2011062300287	23/06/2011	23/06/2011	80.34	<input checked="" type="checkbox"/>
Subtotal			\$ 335.3685	

Listado de facturas pendientes

Código	Fecha de creación	Monto	Monto pendiente	
2011062300282	23/06/2011	30.9	30.9	<input checked="" type="checkbox"/>
2011062300283	23/06/2011	80.34	80.34	<input checked="" type="checkbox"/>

Datos de cierre

Efectivo en el período: Fondo base: Depositar en banco:
 Dinero electrónico: 85.3685

✔ Satisfactorio

© Universidad de las Ciencias Informáticas, 2010.

Figura 22. Interfaz "Cerrar caja".

Interfaz "Establecer prestaciones entidad"

The screenshot displays the 'Establecer prestaciones entidad' interface. On the left is a 'Menú' sidebar with categories like 'Marcadores', 'Prestaciones', 'Gestión de facturas', 'Presupuesto', 'Cierre de caja', 'Obras sociales', 'Convenios', 'Abonos', and 'Configuración'. The main area is titled 'Establecer prestaciones en entidad' and features a search bar. Below the search bar, there are two panels: 'Entidad' (showing 'CIMEQ') and 'Laboratorio'. The 'Laboratorio' panel includes a search criteria section with a 'Nombre:' input field and 'Buscar'/'Cancelar' buttons. Below this is a table titled 'Listado de prestaciones' with columns for 'Nombre' and 'Visible'. The table contains the following data:

Nombre	Visible
otro nuevo	✗
nuevo	✗
Sistema	✓
Franco	✓
Preuba	✗
Exa	✓
Liane	✗
Lolo	✗
Prueba	✓
Examenc	✓

At the bottom of the interface, there is a copyright notice: '© Universidad de las Ciencias Informáticas, 2010.'

Figura 23. Interfaz "Establecer prestaciones entidad".